TOWARDS AUDIO TO SCORE ALIGNMENT IN THE SYMBOLIC DOMAIN

Bernhard Niedermayer
Department for Computational Perception
Johannes Kepler University Linz, Austria
bernhard.niedermayer@jku.at

ABSTRACT

This paper presents a matrix factorization based feature for audio to score alignment. We show that in combination with dynamic time warping it can compete with chroma vectors, which are the probably most frequently used approach within the last years. A great benefit of the factorization-based feature is its sparseness, which can be used in order to transform it into a symbolic representation. We will show that music to score alignments using the symbolic version of the feature is less accurate but on the other hand reduces the memory required for feature representation and during the alignment process to a fraction of the original amount. This is of special value when dealing with very long pieces of music where the limits of default DTW are reached.

1 INTRODUCTION

The problem of audio to midi alignment is well known and has been of broad interest within the last years. The task is to link information from a score representation to an audio recording of a certain performance of a piece of music. Since symbolic transcriptions of a large number of classical as well as modern pieces are available, alignment can replace the much more complex task of blind audio transcription in most scenarios where the performed piece is known in advance.

This is the case in a number of applications like in the field of computational musicology. Performance analysis for example relies on the exact transcription of various performances in order to describe or compare the styles of different artists. Other applications of audio alignment are pedagogical systems and special query engines as well as intelligent audio editors or players.

State-of-the-art approaches like [2], [4], or [12], just to name a few, use a combination of a local distance measure and a specific kind of dynamic time warping (DTW) or Hidden Markov Models (HMM) in order to find the optimal global alignment between the score and a corresponding audio file. Although the idea of calculating such alignments in the symbolic domain is not new [1], modern approaches avoid transcription of the audio data into a symbolic representation. Instead local distances are calculated from acoustic features extracted from the audio signal on the one hand, and either from a rendering or directly from the score on the other hand.

Amongst these acoustic features chroma vectors seem to be most frequently used. Each vector has 12 elements corresponding to the pitch classes (i.e. C, C#, D,...). The computation is based on a short time Fourier transform (STFT) where each frequency bin is mapped to a musical note. The notes are then folded into a single octave by calculating the average energy of all STFT bin contributions to the same pitch class. A more detailed description can be found in [7]. In [4] this representation was compared to others like Pitch Histograms or MFCC based features in the context of audio matching and alignment. It was shown that chroma vectors perform significantly better than the other features.

Another approach is to use features composed of estimations of the presence of individual pitches instead of pitch classes. The idea of such a feature based on non-negative matrix factorization was initially proposed in [2]. In this work we describe a feature that also represents f_0-observation probabilities for single pitches but amongst other things differs in the optimization criterion and algorithms used in the matrix factorization step. Whereas in [2] a qualitative evaluation is given, we will present a quantitative evaluation on a large data set, comparing our feature to chroma vectors and show that the two features yield comparable results when used in combination with dynamic time warping.

Although this feature is acoustic in nature it can be easily converted into a symbolic form in order to use the advantages of both representations. One could be a reduction of computational costs since local similarities can be computed on note events instead of the much larger number of audio frames.

In Section 2 we will give an overview of the algorithm used to extract the proposed feature. Section 3 then describes the global alignment using this feature in the acoustic domain. An evaluation and comparison with the performance of chroma vectors is given in Section 4 before we show and discuss an analogous alignment method in the symbolic domain in Section 5.

SMC 2009, July 23-25, Porto, Portugal

2 METHODOLOGY

2.1 Pitch decomposition

A feature based on non-negative matrix factorization (NMF) for audio alignment was originally proposed in [2]. The basic idea is that a non-negative input V of the size $m \times n$ is decomposed into two as well non-negative output matrices W and H of size $m \times r$ and $r \times n$ respectively, such that

$$V \approx W \cdot H \tag{1}$$

The quality of a factorization is measured by a cost function over V and $W \cdot H$. Common choices for these functions are the Euclidean distance or the Kullback-Leibler divergence. By minimizing the cost function W and H are learned as a fixed number r of basis vectors and the aggregation of their activation patterns over time.

Applying this principle to audio processing, one can use a spectrogram, as obtained by the short time Fourier transform, in order to learn a base set W of weighted frequency groups in an unsupervised manner. In the ideal case these would either represent single pitches played on a certain instrument or pitches that are often found together like the notes of a chord.

However in the context of audio alignment, where the piece and its score are known in advance, we assume the instruments used to be known as well. So there is no absolute need for unsupervised learning of base vectors. Instead a dictionary W of tone models, adjusted to those instruments, can be trained in advance. This leaves us with only H being unknown.

As described by [11] and [8] this reduces the NMF problem to the much simpler decomposition task where each column vector of V can be processed independently, such that equation 1 resolves to

$$v \approx \overline{W} \cdot h \tag{2}$$

where $\overline{W}$ is the fixed set of tone models. v and h represent the spectrogram of a single time frame and the pitch activation respectively. This pitch activation h is the feature vector describing one time frame. In order to find an optimal h, again a cost function is needed. Throughout this work the mean square criterion given as

$$f = \frac{1}{2} \parallel \overline{W}h - v \parallel^2 \tag{3}$$

is chosen and optimized using a standard algorithm for solving non-negative least square problems as described in [6]. Reassembling the activation patterns of all time frames results in a multiple-f_0 estimation over the whole piece of music.

On the other hand extracting feature vectors describing the score is trivial since pitch information can be directly taken from the midi representation.

2.2 Dictionary learning

In order to process the pitch decomposition as described above a dictionary of tone models is required. Each of these models represent one pitch by its weighted frequency components. As pointed out in [8] the exactly same method as used for pitch decomposition in the performing step can be used for model learning in the training step.

Given a database of transcribed audio training samples, the activation patterns of single pitches are known due to the transcription. Therefore H in equation 1 can be fixed to these activation patterns while now W is calculated. Using monophonic training samples where only one pitch is present can simplify the learning step even more. On the one hand W is reduced to a vector and h becomes one scalar at each time frame. On the other hand such training samples can be created with minimal effort. Since the activation energy can be described by the amplitude envelope, the only information required for each sample is the pitch as well as the instrument that has been playing.

3 ALIGNMENT

3.1 Local Distances

Given two sequences of feature vectors a global alignment has to be found that matches each element of one sequence to a corresponding element within the other sequence. In order to measure the similarity between two such elements a local distance function is required. A common choice is the Euclidean distance or the Kullback-Leibler divergence. However two properties of the factorization based feature suggest the use of another distance measure.

In the first place the feature produces a different quantity of deletion (false negative) and insertion (false positive) errors. Especially in high pitch ranges the majority of errors is made up by spurious note detections. Therefore the two types of errors should be treated differently.

Secondly the STFT we use here divides the spectrum into linearly distributed frequency bins. On the contrary musical notes follow a logarithmic frequency scale. So the deeper a tone is, the closer in the spectrogram it is to its immediate neighbors. Additionally higher pitches also exhibit significant energy in the lower frequency bins making it even harder to reliably detect low notes. Therefore local distance calculation should accommodate this fact by relatively tolerant penalizing of missing low notes in the audio feature.

A simple distance measure that combines these ideas and has yielded good results during experimentation is

$$d(h^s, h^p) = \sum_{i=0}^{N-1} diff(h_i^s, h_i^p) \tag{4}$$

with

$$diff(h_i^s, h_i^p) = \begin{cases} h_i^s * \alpha & \text{if } h_i^p = 0 \\ h_i^p * \beta & \text{if } h_i^s = 0 \\ |h_i^s - h_i^p| & \text{else} \end{cases} \quad (5)$$

where h^s represents a feature vector taken from the score and h^p represent one feature vector extracted from the recorded performance. α and β are the weights for missing and spurious notes respectively. Throughout our work 1.2 and 2.0 have proven to yield good results.

Experiments have further shown that alignments can be improved by ignoring missing notes lower than a threshhold around C3 (midi pitch 48). Also taking the square root of d turned out to be advantageous in combination with dynamic time warping as explained in Section 3.2.

3.2 Global Optimization

Using the local distance measure a similarity matrix SM comparing each frame of one sequence to each frame of the other sequence can be built. Mapping corresponding frames together is done by finding a minimal cost path through this similarity matrix. A path through SM_{ij} is then equivalent to the alignment of frame i of the score feature sequence to the performance feature sequence's frame j. Dynamic time warping (DTW) is a well-established dynamic programming based algorithm that finds such optimal paths. A detailed tutorial can be found in [9].

In order to get meaningful results a path has to meet several constraints. The constraint of continuity forces a path to proceed through adjacent cells within the similarity matrix. Jumps would be equal to skipping frames without considering the costs of this operation. The constraint of monotonicity in both dimensions guarantees that the alignment has the same temporal order of events as the reference sequence. And finally the end-point constraint forces the ends of the path to be the diagonal corners of the similarity matrix. In doing so it assures that the alignment covers the whole sequences.

The determination of the optimal path according to DTW works in two steps. The forward step starts at the point $[0, 0]$ with the cost SM_{ij} and recursively calculates the minimized path cost of any partial alignment ending with frame i of the score being aligned to frame j of the recorded performance according to

$$Accu(i,j) = min \begin{cases} Accu(i-1, j-1) + SM_{ij} * w_d \\ Accu(i-1, j) + SM_{ij} * w_s \\ Accu(i, j-1) + SM_{ij} * w_s \end{cases} \quad (6)$$

The three options correspond to a diagonal step, a step upwards, and a step to the right within the similarity matrix respectively. Accordingly w_d and w_s are weights for diagonal and straight steps. We have chosen the values 1.4 and 1.0 giving diagonal steps a preference over straight ones. In our implementation we do this calculation in place, i.e. overwriting the values SM_{ij} by $Accu(i, j)$ in order to save memory space.

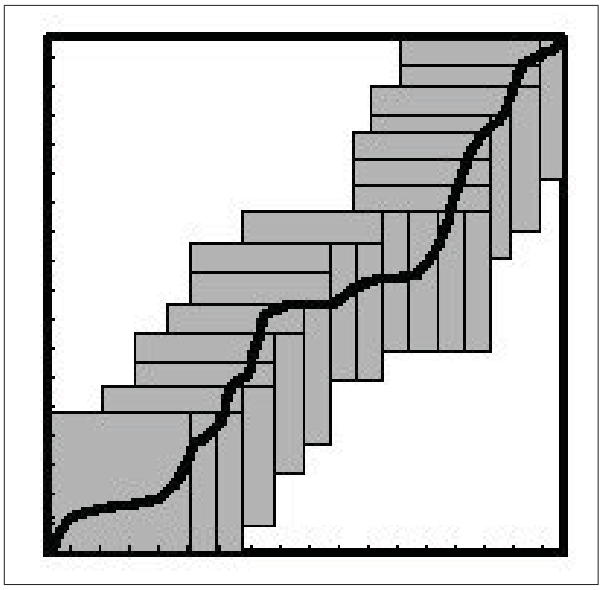

Figure 1. Refinement of the global alignment: The estimated path leads the search area through a similarity matrix computed using a higher time resolution.

Additional to the accumulated path cost a second matrix is used in order to memorize whether the last step leading to a point $[i, j]$ was diagonal, upwards, or to the right. As soon as all values $Accu(i, j)$ have been calculated this information is used to trace the complete path back from $[N-1, M-1]$ to $[0, 0]$.

3.3 Alignment of very long sequences

This algorithm is of complexity $O(n^2)$ in time as well as in space. For very long pieces of music it is impossible to keep a reasonable time resolution of features and still compute a global alignment by DTW. Several improvements have been proposed in order to reduce the complexity, including path pruning where only promising partial paths with costs below an adaptive threshold are further expanded or Shortcut Path where only the alignments of frames corresponding to note on- and offsets are stored [12]. Another approach to handle very long pieces of music is to use online algorithms like proposed in [3] instead of processing the whole piece at once.

Another approach reducing time and space complexity to $O(n)$ is multiscale DTW ([10]). Here an initial estimation of the optimal path is calculated using a low time resolution and then refined iteratively. Since each iteration increases

time resolution but only considers paths near the one found during the last step there is no guarantee that the optimal path is really found. It may happen that low resolution features are misleading in such a strong way that the actual optimum is out of the search radius.

In our implementation we only use two iteration steps. In the first one a standard DTW is computed on features extracted from a spectrogram. The window as well as the hop size was chosen to be 4096 samples ($\sim$93 ms). This allows processing pieces of lengths up to more than 25 minutes. The second step is the refinement, calculated on features based on another spectrogram where the hop size was reduced to 512 samples ($\sim$12 ms). As illustrated by Figure 1 the path estimation from the first step leads the search in the second step so that only similarity values and path costs within an area of radius r frames needs to be calculated. In this way memory requirements can be kept low by just storing similarity measures of the currently processed row or column and path costs for the last $2r$ rows and columns, leading to constant space complexity of this part of the algorithm.

4 EXPERIMENTAL RESULTS

In order to evaluate the factorization based feature we test its performance on several pieces of classical piano music. The database used consists of 13 Mozart sonatas played by a professional pianist on a computer monitored Bösendorfer SE290 grand piano, giving us a precise ground truth of played notes in midi representation. The data set consists of more than 100.000 notes and represents a performance time of almost 4 hours.

The single pieces have lengths from 12 minutes up to more than 26 minutes [1] . This is longer than test pieces used in most other publications. On the one hand this leaves us with issues of computational expenses, which are handled as described in Section 3.3. On the other hand stronger deviations from a strictly diagonal alignment path are expected since the different movements of a sonata are played using different tempi, which prohibits the use of additional alignment constraints like the Itakura Parallelogram [5].

The tone models used during factorization were learned from recordings of single tones played on the same piano. Since such a recording was only available for every fourth midi pitch, the missing models were generated by means of parabolic interpolation.

4.1 Feature Evaluation

In the evaluation process an alignment was calculated for each of the test pieces using the audio recording and a midi file containing the mechanical score without any expressive timing. The resulting note onset times were compared to the ground truth data. Sections where more than 10 consecutive notes had been misplaced by more than 3 seconds were classified as 'unaligned'. Throughout the test set 31 such regions were found containing 2438 notes, which accounts for 2.4% of the overall number of notes. Further investigation showed that such regions where alignment failed are likely to be sections played very softly and with increased use of pedal. This causes the spectrogram that is used as basis for the feature calculation to blur and makes it very hard to distinguish partials belonging to a certain pitch.

For the remaining aligned notes we compute the absolute displacement relative to the ground truth data. In Table 1 we give the median difference, the third quartile and the limit covering 95% of all displacements.

The largest value found within the test set was an error of 8.631 seconds. Although we counted unaligned sections separately, sporadic values of that magnitude are plausible. Since we are dealing with whole sonatas it can happen that the alignment places single notes played at the end of one movement at the beginning of the next one and inversely. Pauses of one or more seconds between movements as well as fermatas and long sustaining of notes at the ends of movements lead to such dramatic values.

In the evaluation of onset detection algorithms, an error threshhold of 50 ms up to which a note onset is classified as correctly detected is quite common in literature. Therefore we also give the percentage of notes satisfying this criterion which is about 50% on average in Table 1.

4.2 Feature Comparison

In the context of audio matching and alignment, a comparison of several acoustic features is given by Hu et al. ([4]). They show that chroma vectors perform significantly better than the other features including variations of an MFCC based approach and Pitch Histograms. This is relevant to our work since Pitch Histograms as defined in [13] also rely on a multiple f_0-estimation. However they are computed by an algorithm based on autocorrelation.

Using the same testing environment as described above we also compare the factorization based feature against the performance of chroma vectors. We found that chroma vectors are more robust, leaving only 687 notes unaligned which accounts for 0.7% of the overall number of notes. However the evaluation of the precision of all aligned notes as given in Table 1 is comparable to the results yielded by the factorization feature. Also the value of the largest absolute error being more than 9 seconds is even worse.

The evaluation criterion used in this work was different from the one in [4]. So the results are not directly comparable. In any event, we can not confirm that chroma vectors perform significantly better than features computed by multiple f_0-estimation except for robustness, where the amount

[1] Note that we align complete sonatas. That is, the pieces were not cut into individual movements.

piece	# notes	duration	chroma vectors				factorization based			
			50% ≤	75% ≤	95% ≤	% ≤ 50ms	50% ≤	75% ≤	95% ≤	% ≤ 50ms
kv279	7387	16:21	35 ms	65 ms	327 ms	64.7%	32 ms	61 ms	408 ms	68.6%
kv280	6070	15:04	41 ms	75 ms	432 ms	59.7%	38 ms	69 ms	399 ms	61.6%
kv281	6395	14:37	40 ms	63 ms	193 ms	61.7%	40 ms	67 ms	169 ms	60.1%
kv282	5564	14:59	60 ms	145 ms	532 ms	41.9%	70 ms	222 ms	808 ms	38.2%
kv283	7884	17:35	38 ms	76 ms	316 ms	59.8%	40 ms	80 ms	500 ms	57.8%
kv284	12762	26:07	37 ms	65 ms	262 ms	64.1%	39 ms	65 ms	260 ms	63.2%
kv330	7589	18:36	38 ms	70 ms	262 ms	60.2%	35 ms	66 ms	358 ms	64.8%
kv331	11580	22:51	276 ms	415 ms	508 ms	9.6%	277 ms	407 ms	493 ms	10.4%
kv332	8744	18:02	51 ms	92 ms	338 ms	49.4%	52 ms	89 ms	302 ms	48.9%
kv333	8833	20:34	58 ms	89 ms	244 ms	44.1%	59 ms	93 ms	261 ms	44.0%
kv457	6915	18:22	44 ms	97 ms	525 ms	54.7%	48 ms	96 ms	919 ms	52.0%
kv475	3871	12:05	79 ms	198 ms	718 ms	32.5%	76 ms	148 ms	579 ms	32.9%
kv533	8611	22:27	46 ms	86 ms	208 ms	52.9%	47 ms	86 ms	178 ms	52.4%
all	102205	3:57:40	50 ms	106 ms	449 ms	50.0%	50 ms	104 ms	459 ms	50.2%

Table 1. Comparison between chroma vectors and the factorization based feature in combination with DTW

of aligned notes was 99.3% instead of 97.6%.

5 THE SYMBOLIC DOMAIN

Most current methods for audio to score alignment including the approaches described above work in the acoustic domain, avoiding the step of explicitly transcribing the audio data. Such a transcription would bring some benefits.

- Whereas acoustic features will result in large arrays of data, symbolic representations are much more compact, using just a small fraction of the original memory space.
- While computing alignments using DTW-like algorithms the number of frames per sequence can be dramatically reduced from a fixed ratio of frames per time unit to one frame each time a note onset or offset occurs.
- Using a transcription in midi format obvious errors of the feature extraction process can be recognized and handled prior to the actual alignment step. Examples for such obvious errors are detected notes with pitches never played during the current piece or detected chords that are never used. This might also eliminate incorrect notes played by the performer in certain cases.

We have also done experiments using the same factorization method as described above to extract an audio feature in midi-format by just setting a note on-event each time the activation energy h_i^p of pitch i becomes greater than zero and setting a note off-event each time h_i^p falls back to zero. This is not just exploiting the sparseness of the factorization result but also strong pruning since note velocities are set to a default value and the actual values of the activation patterns during the note sustain time are dropped.

	50% ≤	75% ≤	95% ≤	% ≤ 50ms
acoustic	32 ms	61 ms	408 ms	68.6%
symbolic	205 ms	370 ms	905 ms	18.9%

Table 2. Comparison of alignments using the factorization based feature in its original version and pruned to a symbolic representation

Applied to the recording of Mozart's piano sonata kv279 the resulting midi representation contains 6275 notes using less than 150 kB of memory. This is a little more than 7.5% of the space needed to store the chroma vectors calculated at a time resolution of 50 frames per second. For the original acoustic representation of the factorization result this relation is even more drastic. The activation patterns of 58 pitches (concerning to the pitch range used in the kv279) require 11MB of memory which is more than 70 times the space needed for the symbolic version of the feature.

The actual alignment is again done using dynamic time warping. In doing so from the score as well as the audio feature slices containing unchanged numbers of notes are extracted (i.e. splitting the piece at each note on- and offset). The resulting number of feature vectors are comparable to those obtained in the first estimation step of our multiscale DTW implementation as described in section 3.3. For the piece kv279 there were no unaligned regions for both feature representations. But as can be seen from Table 2, the accuracy yielded by the symbolic feature can not compete

with the original version. However, a maximum displacement error of 7.95 seconds indicates that stability is not decreased.

A deterioration of accuracy by a factor 7 (concerning the median displacement) may be an acceptable compromise, at least in some applications. It has to be considered that the compactness of feature representation was increased by a factor of 70 and the time of computation in the costly alignment step was reduced to about one tenth because, because no refinement step is needed.

6 CONCLUSIONS

In this paper we have explained a way to extract f_0-estimation features from spectrograms. We then used dynamic time warping in order to align such feature sequences to midi representations of the corresponding score. Since we used whole piano sonatas for our experiments a multiscale DTW approach had to be used in order to tackle complexity issues. Evaluations showed that the extracted feature can compete with other state-of-the-art features.

The actual benefit of the feature described here as well as the one proposed by [2] is that unlike others they are very sparse in nature. So they can easily be converted into a symbolic representation. Using additional pruning the accuracy is reduced significantly but on the other hand data reduction concerning the feature representation as well as during the alignment process is remarkable. Since we have demonstrated the capabilities of our original feature, the modification can be seen as a tradeoff between accuracy and computational costs.

A median displacement error of about 200 ms is too much for applications like performance analysis. But applications like content query engines might profit from such compact features. Especially in the context of huge databases fast and memory-saving routines can be of advantage over methods yielding the highest accuracy.

7 ACKNOWLEDGMENTS

This research is supported by the Austrian Fonds zur Förderung der Wissenschaftlichen Forschung (FWF) under project number P19349-N15.

8 REFERENCES

[1] Bloch, J. J. and Dannenberg, R. B. “Real-Time Accompaniment of Polyphonic Keyboard Performance”, *Proceedings of the 1985 International Computer Music Conference*, pp. 279–290, Burnaby, BC, 1985.

[2] Cont, A. “Realtime Audio to Score Alignment for Polyphonic Music Instruments Using Sparse Non-negative constraints and Hierarchical HMMs”, *IEEE International Conference in Acoustics and Speech Signal Processing (ICASSP)*, Toulouse, 2006.

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

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

[5] Itakura, F. “Minimum Prediction Residual Principle applied to Speech Recognition”, *IEEE Transactions on Acoustics, Speech, and Signal Processing* vol. 23, pp. 52–72, 1975.

[6] Lawson, C. L. and Hanson, R. J. “Solving least squares problems”, *Prentice Hall*, Lebanon, Indiana, 1974.

[7] Müller, M.; Kurth, F. and Clausen, M. “Audio Matching via Chroma-based Statistical Features”, *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, London, 2005.

[8] Niedermayer, B. “Non-negative Matrix Division for the Automatic Transcription of Polyphonic Music”, *Proceedings of the 9th International Symposium on Music Information Retrieval (ISMIR)*, Philadelphia, PA, 2008.

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

[10] Salvador, S. and Chan P. “FastDTW: Toward accurate Dynamic Time Warping in Linear Time and Space”, *Intelligent Data Analysis*, vol. 11/5, pp. 561–580, Amsterdam, 2004.

[11] Sha, F. and Saul, L. “Real-time pitch determination of one or more voices by nonnegative matrix factorization”, *Advances in Neural Information Processing Systems 17*. Saul, K.; Weiss, Y. and Bottou, L. Eds., MIT Press, Cambridge, MA, 2005.

[12] Soulez, F.; Rodet, X. and Schwarz D. “Improving Polyphonic and Poly-Instrumental Music to Score Alignment”, *Proceedings of the 4th International Symposium of Music Information Retrieval (ISMIR)* Baltimore, MD, 2003.

[13] Tzanetakis, G.; Ermolinskyi, A. and Cook, P. “Pitch Histograms in Audio and Symbolic Music Information Retrieval”, *Proceedings of the 3rd International Symposium of Music Information Retrieval (ISMIR)* Paris, 2002.