AN IMPROVED HIERARCHICAL APPROACH FOR MUSIC-TO-SYMBOLIC SCORE ALIGNMENT

Cyril Joder, Slim Essid, Gaël Richard
Institut TELECOM, TELECOM ParisTech, CNRS LTCI
{cyril.joder, slim.essid, gael.richard}@telecom-paristech.fr

ABSTRACT

We present an efficient approach for an off-line alignment of a symbolic score to a recording of the same piece, using a statistical model. A hidden state model is built from the score, which allows for the use of two different kinds of features, namely chroma vectors and an onset detection function (spectral flux) with specific production models, in a simple manner. We propose a hierarchical pruning method for an approximate decoding of this statistical model. This strategy reduces the search space in an adaptive way, yielding a better overall efficiency than the tested state-of-the art method.

Experiments run on a large database of 94 pop songs show that the resulting system obtains higher recognition rates than the dynamic programming algorithm (DTW), with a significantly lower complexity, even though the rhythmic information is not used for the alignment.

1. INTRODUCTION

We address the problem of synchronizing a polyphonic musical score with an audio performance of this score, in the "off-line" version of this task. This allows one to consider the whole recording before estimating the positions of the score notes. We are interested in an alignment at the "symbolic level", which means that the result is the time indexes of the score notes or chords.

Applications of such a system can be a score retrieval from a musical query, or the ability to use both the audio and symbolic (score) content for music indexing. Some musical content analysis tasks, such as motif detection or chord transcription, may indeed be easier on symbolic data than on raw audio files.

While most on-line score following systems use statistical models which can be rather complex [4, 8, 14], many off-line algorithms simply rely on the DTW algorithms or refinements of it [6, 9]. These latter algorithms are often faster than the former and can also be applied to audio-to-audio synchronization.

However, their complexity (in time and space) is quadratic in the number of audio frames. This complexity problem has been addressed in [10], where a "short-time" DTW is proposed, which reduces the memory space requirement, at the cost of a greater time complexity. In [11], Müller *et al* introduce a "multi-scale" DTW (MsDTW) which allows for an efficient pruning strategy in a coarse-to-fine fashion.

To the authors' knowledge, hierarchical approaches have not been used for music synchronization, appart from [11]. In [3], Cont exploits a Hierarchical Hidden Markov Model. However, although its advantage in terms of interpretation, this structure is equivalent to a "flat" HMM [12].

With a dynamic programming framework, the use of different kinds of descriptors can be difficult. Hence such systems use a single feature representation, generally chroma vectors. A notable exception can be found in [6], where a strategy is proposed to combine local distances resulting from chroma vectors and onset features in a DTW scheme. The use of a statistical model makes the fusion of different pieces of information more natural. This structure is often used in real-time systems [2, 5], which model each feature distribution with a Gaussian mixture.

The hidden state model presented here exploits a different model for two different sets of features: a "histogram model" (see Sec. 2.1.1) for chroma vectors and a logistic model (see 2.1.2) for an onset indicator feature. This system obtains a very good alignment precision with a significantly lower complexity than the DTW algorithm.

We also introduce a hierarchical approach for search space reduction, which performs a pruning of the unlikely states in a hierarchical way. We take advantage of structural information given by the score (namely beat and bars), which allows for a meaningful hierarchical segmentation of the music. This method provides an alternative to the commonly used *beam search* strategy, which consists in maintaining only a fixed (small) number of paths at each decoding step. Our approach proves advantageous compared to both beam search and MsDTW, in terms of global search space size and runtime, without affecting the alignment performance in practice.

In the next section we present our baseline models for audio-to-score alignment. Then, a hierarchical pruning method for an approximate decoding of these models is proposed in Section 3. We expose the results of our experiments in Section 4 before suggesting some conclusions in Section 5.

THIS WORK WAS PARTLY SUPPORTED BY THE EUROPEAN COMMISSION UNDER THE OSEO PROJECT QUAERO.

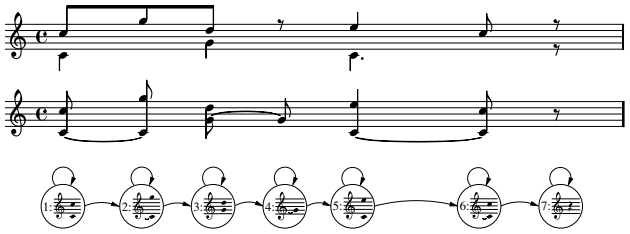

Figure 1. Score Representations. Top: The original graphical score. Middle: The score as a sequence of chord. Bottom: The finite state machine representing the score.

2. BASELINE MODELS FOR AUDIO-TO-SCORE ALIGNMENT

Similarly to [15], we segment the musical score into *chords*, which are sets of notes that sound at the same time. Every time a note appears or disappears, a new chord is created. We can then fit a hidden state model to the audio signal, the states of which are defined by the chords of the score. The score is seen as an automaton, as represented in Figure 1.

In this work, we chose not to take into account the rhythmic information given by the score, as we consider that we have no prior knowledge of the tempo. We use the Maximum Likelihood (ML) path in the automaton as the alignment path. Let $\mathbf{y}{=}y_1, \ldots, y_N$ be the feature sequence extracted from the signal. Let S_n be the random variables describing the current state at time n. The ML path $\hat{\mathbf{S}}$, which can be efficiently computed by the Viterbi algorithm, is:

$$\hat{\mathbf{S}} = \underset{\mathbf{S}\in\mathcal{S}}{\operatorname{argmax}}\, P\big(\mathbf{y}\big|\mathbf{S}\big) = \underset{\mathbf{S}\in\mathcal{S}}{\operatorname{argmax}} \prod_{n=1}^{N} P(y_n|S_n), \quad (1)$$

where $\mathcal{S}$ is the set of acceptable paths. We consider as acceptable the paths which go through all the states in the right order. This model is thus a left-right Hidden Markov Model whose only transitions are self-transitions and transitions from one state to the following one. All these transitions have the same probabilities.

2.1 Observation Models

Similarly to [13], two kinds of information are used in this work: the pitch content and the onset information. Thus, we use two types of features in order to take them into account. *Chroma vectors* are used in order to model the pitch content of the signal, and the *spectral flux* is supposed to detect the note onsets.

2.1.1 Chroma Vectors

As observed in [9], *chroma vectors* provide a compact, yet efficient representation of the pitched content of a musical signal for music-to-score matching. A chroma vector is a twelve-dimension vector, each of whose component represent the "power" in all the frequency bands of a chromatic class (from A to G#). The chroma vectors we use are computed according to [16], with a 50 Hz time resolution.

For each state s, a probability distribution $\{\tilde{g}(i)\}_{i=1\ldots12}$ over the 12 chroma components is built, as the superposition of one-note distributions which correspond to the notes that are present in the state. A one-note distribution is a simple Kronecker function $\{\delta(i,j)\}_{i=1\ldots12}$ where j is the pitch class of the considered note. Then, a constant component q is added in order to model noise, and we obtain a distribution g defined by $g(i) = (1-q)\tilde{g}(i) + \frac{q}{12}$. A value of 0.7 has been found satisfactory for the noise parameter q. For example, the distribution values corresponding to the chord $\{\mathrm{C}_3, \mathrm{E}_3, \mathrm{G}_3, \mathrm{C}_4\}$ are (represented in a vector) $\frac{1-q}{4}(0,0,0,2,0,0,0,1,0,0,1,0)+\frac{q}{12}\mathbb{1}$, where $\mathbb{1}$ is a vector of ones.

In order to calculate the likelihood of each state, we use the model exposed in [15]. The values of the chroma vector v extracted from the audio is considered as a histogram of random samples drawn from the distribution g (corresponding to chord c). The probability of having v as a result of such a sampling is:

$$p\big(v|c\big) = Z(v)\prod_{i=1}^{12} g(i)^{\alpha v(i)}. \quad (2)$$

Here, α is a scaling parameter. Since the value of this parameter has no effect on the decoding result, it is fixed to 1. $Z(v)$ is a positive number which only depends on the observation v, hence it is the same for every path and its value is not considered.

2.1.2 Spectral Flux Feature

In order to render the "burst of energy" which appears at a note onset, we exploit the *spectral flux* feature, which has been proven efficient in a beat tracking task [1]. We use this feature for a "probabilistic" onset detector.

The spectral flux values are first normalized so that their maximum is 1. A local threshold is then computed by applying a 67% rank filter of length 200 ms to the output. We then obtain an "onset feature" by subtracting this local threshold to the normalized spectral flux. Finally, a simple logistic model is used in order to calculate the likelihood of an onset. We denote by A the random variable representing the attack (onset) indicator ($\{A = 1\}$ means that there is an attack). For a value f of the onset feature, we have:

$$p\big(A = 1|f\big) = \frac{e^{bf}}{1+e^{bf}} \quad (3)$$

where b is a positive parameter, which controls the "confidence" of the onset detector: when the value increases, the decision is closer to a deterministic detector (with probabilities 0 or 1).

2.2 *Chord* and *Onset* Models

Two structures of HMMs are evaluated in this work. In the *Chord* structure, a chord is represented by a single state, and only the pitch information (described by the chroma vectors) is taken into account. The spectral flux is not considered.

The *Onset* model is a refinement of the previous structure which takes the onset information into account. In this model, a lower "level of hierarchy" is added in order to model two possible phases of a chord: *attack* and *sustain*. Earch chord corresponding to an onset is split into two successive *phase* states: attack ($A = 1$) and sustain ($A = 0$),

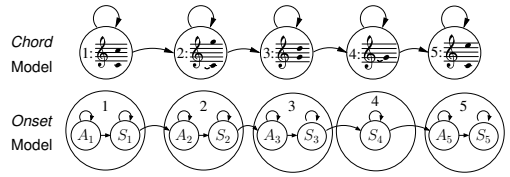

Figure 2. State Structure of the *chord* and *onset* models for the previous score (A and S stand for respectively *attack* and *sustain*).

which share the same chroma vector model. The two types of features are supposed to be independent. The chroma feature is assumed to depend only on the *chord* state while the onset feature only depends on the *phase* state. Hence, if we assume an uninformative prior about the *phase* state, *i.e.* $p(A = a) = \frac{1}{2}$, we have:

$$p\big(v, f|c, a\big) \propto p\big(v|c\big)p\big(A = a|f\big) \tag{4}$$

where these other probabilities are expressed in (2) and (3). Each state of the model is then the combination of a chord and a phase: we write $S_n = (C_n, A_n)$. Equation (1) is then:

$$\hat{\mathbf{S}} = \underset{\mathbf{S}\in\mathcal{S}}{\operatorname{argmax}} \prod_{n=1}^{N} p(v_n|C_n)p(A_n|f_n). \tag{5}$$

Six different values are tested for the parameter b of eq. (3): 0, 0.1, 1, 10, 50 and 100.

The structures of the two models are compared in Fig. 2. For the *Onset* model, the lower level states are represented inside the "chord super-states". In the example, the fourth super-state contains only a *sustain* state, because the transition from the previous chord to the fourth one does not correspond to an onset, but to the extinction of one note.

3. A NOVEL HIERARCHICAL PRUNING APPROACH

In order to speed up the decoding phase, we use a hierarchical pruning approach, inspired by the multi-scale Dynamic Time Warping (MsDTW) algorithm [11]. The idea is to first obtain a coarse alignment and then use the result to prune the search space at a more precise level.

For these coarse alignments, we take advantage of higher musical structures than what we call chords, namely *beat* and *bars*. These structures, given by the score, allow for a meaningful hierarchical segmentation of the music. At each of these levels, a HMM can be built, whose states correspond respectively to the beats and to the bars of the score. As the considered temporal units are larger and the precision needed at these levels is lower, the observations used for the alignment are calculated over longer windows, with a smaller time resolution. Figure 3 illustrates the construction of the automata and the calculation of the observations, at the three levels of hierarchy.

The algorithm proceeds as follows: on the highest level automaton, we calculate for every state s and every frame n, the maximum likelihood that can be obtained by going

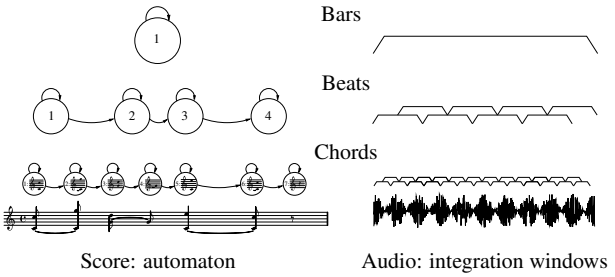

Figure 3. Finite state machines (modeling the score) and integration windows (over which are calculated the observations) at the three considered levels of hierarchy.

through state s at time n. This value is written

$$\bar{P}(s, n) = \max_{\mathbf{S}\in\mathcal{S}, S_n = s} \big\{P(\mathbf{y}|\mathbf{S})\big\} \tag{6}$$

where $\mathcal{S}$ is the set of acceptable paths and $\mathbf{y}$ is the observation sequence. This calculation can be done by a "forward-backward version" of the Viterbi algorithm. It is very similar to the forward-backward algorithm, and can be deduced from it by replacing the `sum` operation by a `max`. This algorithm allows for the calculation of the optimal path $\hat{\mathbf{S}} = \hat{S}_1, \ldots, \hat{S}_N$ at the same time.

The values $\bar{P}(s, n)$ are then used to prune the low-score paths. We do not use the posterior probabilities $P(S_n|\mathbf{y})$ instead of $\bar{P}(s, n)$ for the pruning process since we are interested in the path's scores and not in the states'. Since a state probability is the sum of the probabilities of the path going through this state, there is a risk that some states containing many average-score paths may be favored compared to a state containing an isolated high-score path.

This "pruning score" $\bar{P}(s, n)$ constitutes an important difference with the beam search strategy. Indeed, beam search operates directly at the low level and it uses the partial Viterbi score

$$\tilde{P}_n(s, n) = \max_{\mathbf{S}\in\mathcal{S}, S_n = s} \big\{P(y_1, \ldots, y_n|S_1, \ldots, S_n)\big\} \tag{7}$$

in order to prune the low-score path. Hence it only considers the observation up to the current frame, whereas our approach takes into account the whole signal.

The structure of the automaton is left-right, thus the relation defined on the set of states by: $s \leq s'$ iff there is a path from s to s', is a total order. It is then possible to define the "furthest admissible states" S_n^- and S_n^+ for each time n by:

$$S_n^- = \min\big\{s \;\big|\; \tilde{P}(s, n) \geq \frac{P(\mathbf{y}|\hat{\mathbf{S}})}{\eta}\big\} \tag{8}$$

$$S_n^+ = \max\big\{s \;\big|\; \tilde{P}(s, n) \geq \frac{P(\mathbf{y}|\hat{\mathbf{S}})}{\eta}\big\}, \tag{9}$$

where η is a parameter which controls the minimum likelihood of the paths that are kept in the pruning process. We define the *tolerance radii* δ_- and δ_+ as the maximum number of states that separate respectively S_n^- from $\hat{S}_n$ and $\hat{S}_n$ from S_n^+, for $n \in \{1, \ldots, N\}$.

These tolerance radii specify a set of states around the alignment path, which allows for a reduction of the search

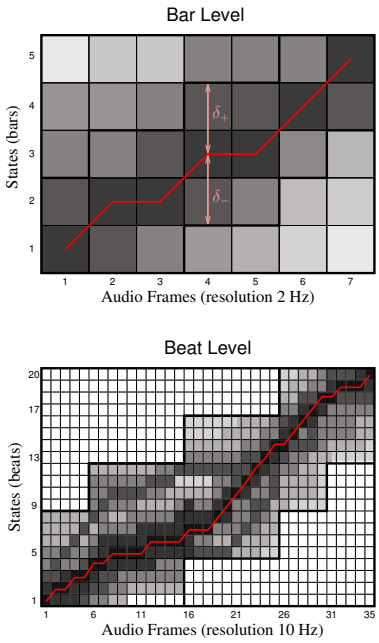

Figure 4. Principle of the hierarchical pruning method. The grey scale of a cell correspond to the maximum likekihood of the paths going through this cell. At the beat level, only the domain delimited by the black lines is explored.

space at the lower level. Hence, the alignment at the lower level is calculated by exploring only this domain. Figure 4 illustrates this pruning process. The same procedure is repeated at each level.

The observations used in the higher levels are integrated (moving average) versions of the chroma vectors, with a lower time resolution. This use of averaged observations is musically justified since the harmony (and thus the chroma information) is in general homogeneous over a whole beat (or bar) duration. The spectral flux is not considered in these levels. The integration windows are chosen in order to take into account the fastest reasonable tempi. For the beat level, this duration is 200 ms, corresponding to a very fast tempo of 300 beats per minute. For the bar level, the integration window is 1 s, that is a four-four time with a tempo of 240 bpm. A 50% overlap is used, yielding time resolutions of respectively 10 Hz and 2 Hz. The histogram model exposed in Sec. 2 is used and the distribution g corresponding to a state (beat or bar) is the superposition of the distributions associated to the chords that it contains, weighted by their theoretical durations (in beat).

The main difference between the MsDTW pruning approach and ours is that the tolerance widths δ are not given as a parameter, but they are computed from the data in an adaptive way, controlled by the parameter η. It is often more advantageous to set the tolerance in terms of likelihood (parameter η) than in terms of deviation from the alignment path (parameter δ). Indeed, it is possible that a wrong path obtains a slightly higher score than the right one at a coarse level (for example a path following a different repetition of a musical phrase). If this wrong path is far (in terms of states) from the right alignment, the latter one will be discarded by the fixed-radii pruning process. On the other hand, it is reasonable to suppose that the "real" alignment path always obtains a high likelihood, and thus is not pruned out by our method.

4. EXPERIMENTS

4.1 Database and Evaluation Measure

The database used in this work comprises 94 songs of the RWC-pop database [7]. These songs are polyphonic multi-instrumental pieces of length 2 to 6 minutes, most of which contain percussion. The alignment ground-truth is given by the synchronized MIDI files provided with the recordings. The same MIDI files are exploited as target scores. However, as we intend to be able to handle any type of score, in particular scores with missing or unreliable tempo indications, we artificially introduce (rather extreme) tempo modifications in these MIDI files: every 4 bars, a random tempo change (between 40 and 240 bpm) is added.

The chosen evaluation measure is the recognition rate, defined as the fraction of onsets which are correctly detected less than $\theta = 300$ ms away from the real onset time. This threshold is based on the MIREX'06 contest [1] .

4.2 Reference System: DTW

We compare our alignment models to a reference DTW (Dynamic Time Warping) system. The DTW algorithm searches for the alignment between two sequences which minimizes the cumulative costs along the alignment path.

This method is used to synchronize the sequence of observations (chroma vectors and spectral flux) extracted from the audio with a sequence built from the score. This "pseudo-synthesis" is performed by associating to each chord a chroma vector template (having the same values as the probability distributions of Sec. 2.1.1) and a duration given by the score. The obtained sequence is then linearly stretched so that its length is the same as the recording. For the onset detection feature, the reference sequence is a sequence of zeros and ones, the ones correponding to the onset locations in the "pseudo-synthesis".

For this system, the spectral flux sequence is locally normalized so that its maximum is 1 on a 2-s sliding window. The local distance between the observation $(v, \bar{f})$ (respectively chroma vector and locally normalized spectral flux) and the template counterpart (g, a) is given by:

$$D\Big((v, f), (g, a)\Big) = \frac{v \cdot g}{\|v\|\|g\|} + w\,|f - a|\,, \qquad (10)$$

where $\cdot$ denotes the inner product and w is a non-negative parameter which controls the weight given to the onset detection feature. Between three different values which have been tested $\{\frac{1}{2}, 1, 2\}$, the value $w = 1$ has been found the most efficient on our database. A DTW system which considers only the chroma observation (corresponding to $w = 0$) is also evaluated.

[1] Music Information Retrieval Evaluation eXchange 2006, score following task: http://www.music-ir.org/mirex/2006/index.php/Score_Following_Proposal

System	Recognition Rate	Search Space
DTW (only chroma)	78.77%	100%
DTW (chroma+onset)	**86.07**%	
Chord	64.49%	16.2%
Onset ($b = 0$)	69.70%	26.3%
Onset ($b = 0.1$)	70.49%	
Onset ($b = 1$)	73.14%	
Onset ($b = 10$)	82.90%	
Onset ($b = 50$)	**87.16**%	
Onset ($b = 100$)	84.71%	

Table 1. Recognition rate and mean search space (fraction of the DTW algorithm search space) as a function of the alignment system.

4.3 Performances of the Baseline Systems

The recognition rates and average search space of several settings are summed up in Table 1. The search space is the number of explored cells (state/frame pairs, or audio frame/pseudo-synthesis frame pairs, depending on the system) over the total number of cells required for the DTW algorithm (the square of the number of audio frames).

First, it can be seen that the DTW which considers only the chroma observations performs better than the *chord* model. This is easily explained by the fact that the former system implicitly models the note durations in the pseudo-synthesis stage, whereas the statistical models do not take them into account. This increases the precision, but also the search space (from 16.2% to 100%).

However, the use of the onset information allows the *onset* model to overcome this shortcoming and to obtain a slighty better precision than the DTW systems, with a still lower complexity. Indeed, a recognition rate of 87.16% is obtained with a value of $b = 50$, against 86.07% for the DTW system wich takes into account the onset observation, whereas the mean search space is 26.3%.

The increase of accuracy induced by the onset observation is smaller in the DTW system than in the statistical models. This is probably due to the difficulty of modeling the spectral flux process. Indeed, this onset detection function is not very well modeled by our binary templates, and the logistic model of (3) seems to be more relevant to this process than the local distance of (10).

The increase of search space in the *onset* model is beneficial to the alignment precision. Indeed, even with a value of $b = 0$ (which means that the onset information is not used) the recognition rate increases from 64.49% (*chord* model) to 69.70%. The explanation lies in the fact that most chords are then represented by two states. Thus the minimum duration of each chord is two frames instead of one, which prevents the system from rapidly skipping several states and leads to a smoother alignment path.

4.4 Pruning Evaluation

This hierarchical pruning method is run on the RWC popular music database. The lowest-level model uses the *onset* structure with parameter $b = 50$. Several values of the pruning parameter η have been tested and the experimental results are summed up in Table 2. The mean search space sizes are displayed for each pruning setting at the beat and chord level, as the fraction of explored cells over the total number of cells used by the DTW algorithm. At the bar level, it is 0.16% for MsDTW and 0.04% for all the other systems. For each setting, the total run-time is also presented, as well as the number of "pruning errors" on the whole database (94 songs). A pruning error occurs when a part of the ground truth alignment path is discarded by the pruning process. The implementation of the algorithms is in MATLAB, and was run on a Intel Core2, 2.66 GHz with 3.6 Go RAM under Linux.

System	Search Space		Run time (in s)	Errors (nb)
	Beats	Onsets		
Onset $b = 50$	–	26.26%	482	0
BS $N_h = 700$	–	5.74%	733	0
MsDTW $\delta = 150$	2.24%	14.02%	1180	0
$\delta = 60$	0.81%	7.93%	362	0
$\eta = 1000$	0.42%	4.53%	300	0
$\eta = 200$	0.35%	4.07%	276	0
$\eta = 100$	0.33%	3.82%	265	0
$\eta = 50$	0.30%	3.59%	256	0
$\eta = 20$	0.26%	3.22%	240	0
$\eta = 10$	0.23%	2.97%	229	2
$\eta = 5$	0.19%	2.59%	215	2

Table 2. Performance of our implementation of the alignment algorithm using different settings of the hierarchical pruning method.

The performance of three additional reference systems is displayed. This first one is the baseline onset model with no pruning. The second reference system uses beam search (BS). This algorithm performs the decoding of the statistical model similarly to the Viterbi algorithm, but maintains only the best N_h paths, according to the partial Viterbi score of (7). The minimum value of N_h for which the decoded path is the same as without pruning is $N_h = 700$.

The third reference system is a MsDTW (multi-scale DTW) system [11]. This system performs a DTW alignment, but it uses a coarse-to-fine pruning process in order to keep only a fixed neighborhood around the alignment path, at each level. The same three levels as in 3 are used. The deviation parameter value $\delta = 150$ is the minimum value yielding no pruning error on our database.

Finally, the last one uses constant tolerance radii $\delta_- = \delta_+ = 60$. This value δ is the lowest one for which no pruning errors are made.

In terms of alignment precision, all the systems which do not make pruning errors obtain the same scores as the reference system (87.16%). Thus, the reduction of the search space does not affect the alignment precision.

The results show the benefits of this pruning method, since the search space and run time of all the tested systems which use it are lower than the reference system (without pruning). As expected, the explored space decreases with the value of η. No pruning error occurs until a value of $\eta = 20$, whose corresponding run-time is half of the reference system (240 s against 484 s).

The benefit of this method compared to a fixed radius

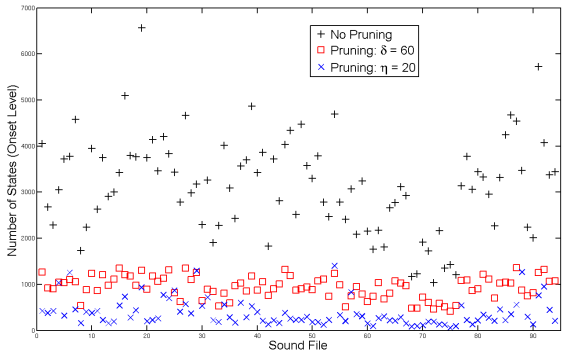

Figure 5. Number of explored states per audio frame at the onset level for each song of the database, without and with pruning (*onset* model with $b = 50$).

δ can also be seen: the tested system with $\delta = 60$ (the minimum value for no pruning error) runs in 362 s, and requires more space than our "adaptive" pruning strategy.

The hierarchical strategy allows our approach to be more effective than beam search (3.53% search space against 5.74%). Hence, considering the whole signal (although at a coarse level) seems to reduce the risk of following a promising path which will eventually come to a "dead-end". This problem could be adressed by estimating a tempo process in a beam search approach, such as in [15] or [4]. However the complexity of these models would be much higher.

In Fig. 5 are displayed the numbers of explored states per audio frame in each song of our database, for three different pruning strategies: the reference system (without pruning), the system using a fixed radius $\delta = 60$ and the system using our adaptive approach with $\eta = 20$. Both pruning strategies achieve a significant reduction of the search space on all the songs. More interestingly, we can see that the search space width obtained with our pruning strategy can greatly vary from songs to songs, whereas it is more or less constant with a fixed δ (only affected by the number of onset states in a beat). This variability is uncorrelated to the original number of states in the score, indicating that our approach manages to adapt the pruning process to the data. Thus, whereas in some cases, the width obtained with our method is greater than with a constant δ, it is most of the time significantly smaller.

5. CONCLUSION

In this paper, we show that a novel hierarchical pruning approach for the approximate decoding of a hidden state model leads to a good precision in our alignment task, with a low complexity. In our experiments, we find that the recognition rate is even higher than a DTW system when a description of note onsets is used additionally to the chroma vectors, while keeping a lower complexity than this algorithm in the decoding phase.

The proposed hierarchical pruning method further reduces the complexity without affecting the accuracy of the system. The main advantage of this strategy compared to the one used in [11] is that the tolerance radii can adapt to the data, yielding a better overall efficiency.

In the continuation of this work, we will address the use of a more elaborate model at the lowest level, which is now feasible thanks to the pruning strategy. We will also try to further reduce the number of states in the model, by taking advantage of the repetitions in the musical structure.

6. REFERENCES

[1] M. Alonso, G. Richard, and B. David. Extracting note onsets from musical recordings. In *Proc. of ICME*, 2005.

[2] P. Cano, A. Loscos, and J. Bonada. Score-performance matching using hmms. In *Proc. of the ICMC*, 1999.

[3] A. Cont. Realtime audio to score alignment for polyphonic music instruments using sparse non-negative constraints and hierarchical hmms. In *Proc. of ICASSP*, 2006.

[4] A. Cont. A coupled Duration-Focused architecture for Real-Time Music-to-Score alignment. *IEEE Trans. on PAMI*, 32(6):974–987, June 2010.

[5] A. Cont, D. Schwarz, and N. Schnell. Training ircam's score follower. In *Proc. of ICASSP*, 2005.

[6] S. Ewert, M. Müller, and P. Grosche. High resolution audio synchronization using chroma onset features. In *Proc. of ICASSP*, 2009.

[7] M. Goto. Rwc music database: Popular, classical, and jazz music databases, 2002.

[8] L. Grubb and R. Dannenberg. A stochastic method of tracking a vocal performer. In *Proc. of ICMC*, 1997.

[9] N. Hu, R. Dannenberg, and G. Tzanetakis. Polyphonic audio matching and alignment for music retrieval. In *Proc. of WASPAA*, 2003.

[10] H. Kaprykowsky and X. Rodet. Globally optimal short-time dynamic time warping: Application to score to audio alignment. In *Proc. of ICASSP*, 2006.

[11] M. Müller, H. Mates, and F. Kurth. An efficient multiscale approach to audio synchronization. In *Proc. of ISMIR*, 2006.

[12] K. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. UC Berkeley, july 2002.

[13] N. Orio and D. Schwarz. Alignment of monophonic and polyphonic music to a score. In *Proc. of ICMC*, 2001.

[14] C. Raphael. Automatic segmentation of acoustic musical signals using hidden markov models. *IEEE Trans. on PAMI*, 21:360–370, 1999.

[15] C. Raphael. Aligning music audio with symbolic scores using a hybrid graphical model. *Machine Learning Journal*, 65:389–409, 2006.

[16] Y. Zhu and M. Kankanhalli. Precise pitch profile feature extraction from musical audio for key detection. *IEEE Trans. on Multimedia*, 8(3):575–584, 2006.