An Efficient State-Space Model for Joint Tempo and Meter Tracking

Florian Krebs, Sebastian Böck, and Gerhard Widmer
Department of Computational Perception
Johannes Kepler University, Linz, Austria
florian.krebs@jku.at

ABSTRACT

Dynamic Bayesian networks (e.g., Hidden Markov Models) are popular frameworks for meter tracking in music because they are able to incorporate prior knowledge about the dynamics of rhythmic parameters (tempo, meter, rhythmic patterns, etc.). One popular example is the *bar pointer model*, which enables joint inference of these rhythmic parameters from a piece of music. While this allows the mutual dependencies between these parameters to be exploited, it also increases the computational complexity of the models. In this paper, we propose a new state-space discretisation and tempo transition model for this class of models that can act as a drop-in replacement and not only increases the beat and downbeat tracking accuracy, but also reduces time and memory complexity drastically. We incorporate the new model into two state-of-the-art beat and meter tracking systems, and demonstrate its superiority to the original models on six datasets.

1. INTRODUCTION

Building machines that mimic the human understanding of music is vital for a variety of tasks, such as organising and managing today's huge music collections. In this context, automatic inference of metrical structure from a musical audio signal plays an important role. Generally, the metrical structure of music builds upon a hierarchy of approximately regular pulses with different frequencies. In the centre of this hierarchy is the *beat*, a pulse to which humans choose to tap their feet. These beats are again grouped into bars, with the *downbeat* denoting the first beat of each bar.

Several approaches have been proposed for tackling the problem of automatic inference of meter (or subcomponents such as beats and downbeats) from an audio signal, with approaches based on machine learning currently being the most successful [1, 5, 12, 13, 22]. All of these approaches incorporate probabilistic models, but with different model structures: the systems introduced in [5, 13, 22] decouple tempo detection from the detection of the beat/downbeat phase, which has the advantage of reducing the search space of the algorithms but can be problematic if the tempo detection is erroneous. Others [1, 12] model tempo and beat/downbeat jointly, taking into account their mutual dependency, which leads to increased model complexity.

One popular model that jointly models tempo and bar position is the *bar pointer model*, first proposed in [20]. In addition to tempo and bar position, the model also integrates various rhythmic pattern states. It has been extended by various authors: in [12,14] the benefit of using rhythmic pattern states to analyse rhythmically diverse music was demonstrated, in [18] a simplification for models with multiple rhythmic pattern states was proposed, in [17] the label of an acoustic event was additionally modelled in order to enable a drum robot to distinguish different instruments, and in [6] it was applied to a drum transcription task. These algorithms share the problem of a high space and time complexity because of the huge state-space in which they perform inference. In order to make inference tractable, the state-space is usually divided into discrete cells, with either fixed [1, 6, 12, 14, 17, 20] or dynamic [15, 18, 21] locations in the state-space. While the former approach can be formulated as a hidden Markov model (HMM), which performs best but is prohibitively complex, the latter uses particle filtering (PF), which is fast but performs slightly worse in sub-tasks such as downbeat tracking [15].

In this paper, we propose a modified bar pointer model which not only increases beat and downbeat tracking accuracy, but also reduces drastically time and memory complexity. In particular, we propose (a) a new (fixed grid) discretisation of the joint tempo and beat/bar state-space and (b) a new tempo transition model. We incorporated the new model into two state-of-the-art beat and meter tracking systems, and demonstrate its superiority on six datasets.

2. METHOD

In this section, we describe how we tackle the problem of metrical structure analysis using a probabilistic state-space model. In these models, a sequence of *hidden variables*, which in our case represent the meter of an audio piece, is inferred from a sequence of *observed variables*, which are extracted from the audio signal. For ease of presentation, we now consider a state-space of two hidden variables, the position within a bar and the tempo. Including additional hidden variables, e.g., a rhythmical pattern

state [12, 14, 18, 20, 21] or an acoustic event label [6, 17] is straightforward. In the following, we describe the original bar pointer model [20], its shortcomings, and the proposed improvements.

2.1 The original bar pointer model

The bar pointer model [20] describes the dynamics of a hypothetical pointer which moves through the space of the hidden variables throughout a piece of music. At each time frame k, we refer to the (hidden) state of the bar pointer as $\mathbf{x}_k = [\Phi_k, \dot{\Phi}_k]$, with $\Phi_k \in \{1, 2, ..., M\}$ denoting the position within a bar, and $\dot{\Phi}_k \in \{\dot{\Phi}_{min}, \dot{\Phi}_{min}+1, ..., \dot{\Phi}_{max}\}$ the tempo in bar positions per time frame. M is the total number of discrete positions per bar, $N = \dot{\Phi}_{max} - \dot{\Phi}_{min} + 1$ is the total number of distinct tempi, $\dot{\Phi}_{min}$ and $\dot{\Phi}_{max}$ are respectively the lowest and the highest tempo. See Fig. 1a for an illustration of such a state space. Finally, we denote the observation features as $\mathbf{y}_k$.

Overall, we want to compute the most likely hidden state sequence $\mathbf{x}^*_{1:K} = \{\mathbf{x}^*_1, \mathbf{x}^*_2, ..., \mathbf{x}^*_K\}$ given a sequence of observations $\{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_K\}$ for each audio piece as

$$\mathbf{x}^*_{1:K} = \arg\max_{x_{1:K}} P(\mathbf{x}_{1:K} \mid \mathbf{y}_{1:K}). \quad (1)$$

with

$$P(\mathbf{y}_{1:K}|\mathbf{x}_{1:K}) \propto P(\mathbf{x}_1) \prod_{k=2}^{K} P(\mathbf{x}_k|\mathbf{x}_{k-1}) P(\mathbf{y}_k|\mathbf{x}_k). \quad (2)$$

Here, $P(\mathbf{x}_1)$ is the *initial state distribution*, $P(\mathbf{x}_k|\mathbf{x}_{k-1})$ is the *transition model*, and $P(\mathbf{y}_k|\mathbf{x}_k)$ is the *observation model*, which we further describe in the bottom of this section. Eq. 1 can be solved using the well-known Viterbi algorithm [16]. Finally, the set of downbeat frames $\mathcal{D}$ can be extracted from the sequence of bar positions as

$$\mathcal{D} = \{k : \Phi^*_k = 1\}, \quad (3)$$

and the set of beat frames can be obtained analogously by selecting the time frames which correspond to a bar position that matches a beat position.

2.1.1 Initial distribution

Here, any prior knowledge (e.g., about tempo distributions) can be incorporated into the model. Like most systems, we use a uniform distribution in this work.

2.1.2 Transition model

The transition model $P(\mathbf{x}_k \mid \mathbf{x}_{k-1})$ can be further decomposed into a distribution for each of the two hidden variables Φ_k, and $\dot{\Phi}_k$ by:

$$P(\mathbf{x}_k \mid \mathbf{x}_{k-1}) = P(\Phi_k \mid \Phi_{k-1}, \dot{\Phi}_{k-1}) \cdot \cdot P(\dot{\Phi}_k \mid \dot{\Phi}_{k-1}). \quad (4)$$

The first factor is

$$P(\Phi_k \mid \Phi_{k-1}, \dot{\Phi}_{k-1}) = \mathbb{1}_x, \quad (5)$$

where $\mathbb{1}_x$ is an indicator function that equals one if $\Phi_k = (\Phi_{k-1} + \dot{\Phi}_{k-1} - 1) \bmod M + 1$, and zero otherwise. The modulo operator makes the bar position cyclic (the last, light grey column in Fig. 1a is identical to the first column).

The second factor $P(\dot{\Phi}_k \mid \dot{\Phi}_{k-1})$ is implemented by
If $\dot{\Phi}_{min} \leq \dot{\Phi}_k \leq \dot{\Phi}_{max}$,

$$P(\dot{\Phi}_k \mid \dot{\Phi}_{k-1}) = \begin{cases} 1 - p_{\dot{\Phi}}, & \dot{\Phi}_k = \dot{\Phi}_{k-1}; \\ \frac{p_{\dot{\Phi}}}{2}, & \dot{\Phi}_k = \dot{\Phi}_{k-1} + 1; \\ \frac{p_{\dot{\Phi}}}{2}, & \dot{\Phi}_k = \dot{\Phi}_{k-1} - 1, \end{cases} \quad (6)$$

otherwise $P(\dot{\Phi}_k|\dot{\Phi}_{k-1}) = 0$.
$p_{\dot{\Phi}}$ is the probability of a tempo change. From Eq. 6 it can be seen that the pointer can perform three tempo transitions from each state (indicated by arrows in Fig. 1a).

2.1.3 Observation model

In this paper, we use two different observation models: The first one uses *recurrent neural networks* to derive a probability of a frame being a beat or not [1]. The second one models the observation probabilities with Gaussian mixture models from a two-dimensional onset feature [12, 14]. As the focus of this paper lies on the state discretisation and the tempo transition model, the reader is referred to [1, 12, 14] for further details.

2.2 Shortcomings of the original model

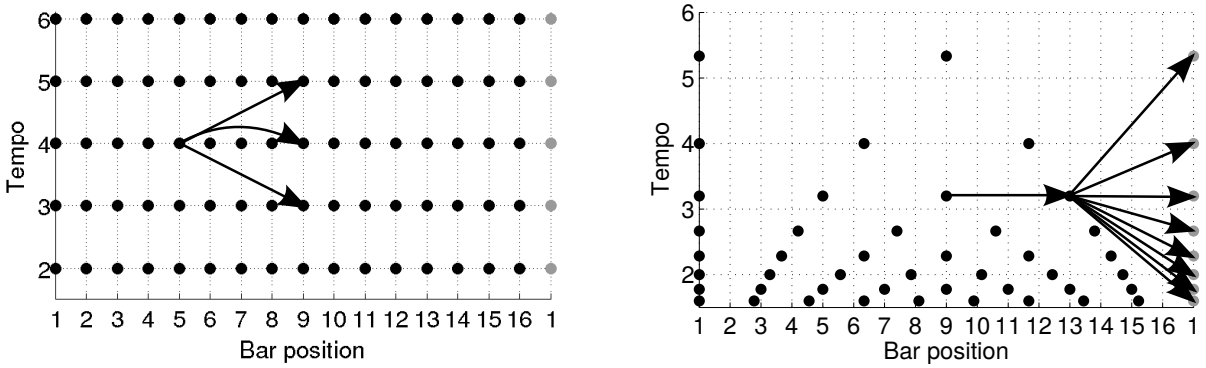
Previous implementations of the bar pointer model [1, 2, 6, 12, 14, 17] followed [20] in dividing the tempo-position state space into equidistant points, with each point aligned to an integer-valued bar position and tempo (see Fig. 1a). This discretisation has a number of drawbacks, which are further explained in the following.

2.2.1 Time resolution

As shown in Fig. 1a, the number of position grid points per bar is constant across the tempi. This means that the grid of a bar played at a low tempo has a lower time resolution than of a bar played at high tempo, because both are divided into the same number of cells. In contrast, there are more observations available for a bar at a low tempo than for a bar at a high tempo, since the observations are extracted at a constant frame rate. This causes a mismatch between the time resolution of the feature extraction and the time resolution of the discretised bar position.

2.2.2 Tempo resolution

As shown in Fig. 1a, the distance between two adjacent tempo grid points is constant across the grid. This is inconsistent with tempo sensitivity experiments on humans, which have shown that the human ability to notice tempo changes is proportional to the tempo, with the JND (just noticeable difference) being around 2-5% of the inter beat interval [4]. Therefore, in order to get a sufficiently high tempo resolution at lower tempi, a huge number of tempo states has to be chosen.



(a) Original discretisation [20]

(b) Proposed discretisation

Figure 1: Toy example with $M = 16$ and $N = 6$: Each dot corresponds to a (hidden) state in the tempo-bar-position state-space. The arrows indicate examples of possible state transitions.

2.2.3 Tempo stability

As the tempo model (see Eq. 6) forms a first-order Markov chain, the current tempo state is independent of all tempo states given the past tempo state. This means that the tempo model is not able to reflect any long term dependencies between tempo states, which may result in unstable tempo trajectories.

2.3 Proposed model

This section introduces a solution to the problems described above. To simplify notation we assume a bar has four beats. Extending to other time signatures [20] or modelling beats instead of bars [1] is straightforward.

2.3.1 Time resolution

We propose making the number of discrete bar positions M dependent on the tempo by using exactly one bar position state per audio frame (and thus per observation feature value). The number of observations per bar (four beats) at a tempo T in beats per minute (BPM) is

$$M(T) = \text{round}(\frac{4 \times 60}{T * \Delta}) \tag{7}$$

with Δ being the audio frame length. Using Eq. 7, we compute the number of bar positions of the tempo limits $M(T_{min})$ and $M(T_{max})$.

2.3.2 Tempo resolution

We can now either model all N_{max} tempi that correspond to integer valued bar positions in the interval $[M(T_{max}), M(T_{min})]$, with

$$N_{max} = M(T_{min}) - M(T_{max}) + 1, \tag{8}$$

or select only a subset of N tempo states. In Section 3, we evaluate the performance of the transition model for various numbers of tempo states. For $N < N_{max}$, we choose the tempo states by distributing N states logarithmically across the range of beat intervals, trying to mimic the JNDs of the human auditory system [4].

2.3.3 Tempo stability

To increase the stability of the tempo trajectories we only allow transitions at beat positions within a bar. This is illustrated in Fig. 1b with the arrows showing examples of possible state transitions. In contrast to the original model which allows three tempo transitions at every time step, we allow transitions to each tempo, but only at beat times. The new tempo transition model then becomes:

If $\Phi_k \in \mathcal{B}$,

$$P(\dot{\Phi}_k|\dot{\Phi}_{k-1}) = f(\dot{\Phi}_k, \dot{\Phi}_{k-1})$$

else

$$P(\dot{\Phi}_k|\dot{\Phi}_{k-1}) = \begin{cases} 1, & \dot{\Phi}_k = \dot{\Phi}_{k-1}; \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

$\mathcal{B}$ is the set of bar positions that corresponds to beats, and $f(\cdot)$ is a function that models the tempo change probabilities. We experimented with various functions (Gaussian, Log-Gaussian, Gaussian mixtures), but found this exponential distribution to be performing best:

$$f(\dot{\Phi}_k, \dot{\Phi}_{k-1}) = exp(-\lambda \times |\frac{\dot{\Phi}_k}{\dot{\Phi}_{k-1}} - 1|) \tag{10}$$

where the rate parameter $\lambda \in \mathbb{Z}_{\geq 0}$ determines the steepness of the distribution. A value of $\lambda = 0$ means that transitions to all tempi are equally probable. In practice, for music with roughly constant tempo, we set $\lambda \in [1, 300]$. Fig. 2 shows the tempo transition probabilities for various values of λ.

2.4 Complexity of the inference algorithm

In this section, we investigate time and memory complexity of the bar pointer model, considering only the complexity of the (Viterbi) inference and ignoring the contribution of computing the observation features and observation probabilities.

Both time and space complexity depend on the number of states of the model. The number of states, in turn, depends on the number of bar positions, the tempo ranges, the audio frame length, and the tempo resolution that we chose to model. Let us assume that we have a model with

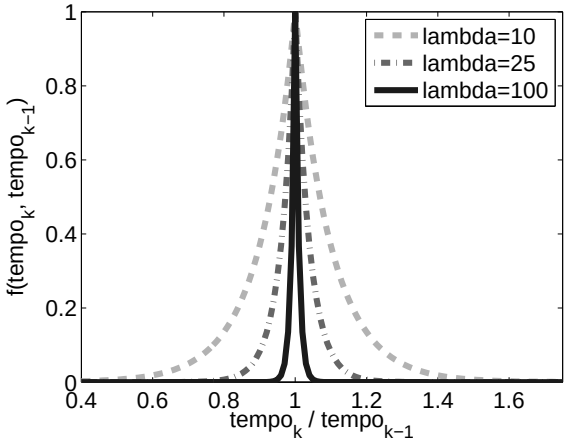


Figure 2: Tempo change probability density (Eq. 10) for various values of λ.

S hidden states, T possible state transitions per frame, and an audio excerpt with K frames. The memory requirement of the algorithm is then simply $S \times K$, as we have to store the best predecessor state for each of the S states for each time frame during Viterbi decoding. The time complexity, on the other hand, is $T \times K$, as we have to compute T transitions at each time step. In Table 1 we show the values of S and T of the models used in this paper.

3. EXPERIMENTAL SETUP

In this section, we evaluate the proposed model with real-world music data in two experiments [1]. In the first experiment, we investigated the effect of the number of tempo states N and the rate parameter λ of the tempo transition function on the meter tracking performance on a training set. We evaluated only the beat tracking performance, as this is the most fundamental task that we wanted to solve. In the second experiment, we integrated the proposed model with the parameters determined in Experiment 1 into two state-of-the-art systems and compared the meter tracking performance in terms of accuracy and complexity with the original models. Below, we describe the datasets, the evaluation metrics, and the meter tracking models.

3.1 Datasets

In this work, we used seven test datasets, one for Experiment 1 and the remaining six for Experiment 2. For more details about each datasets, see the corresponding references:

Experimental dataset: This dataset is a subset of the *1360-songs* dataset [8] excluding the Hainsworth dataset, because it was used in Experiment 2. In total, it includes 1139 excerpts (total length 662 minutes).

Ballroom dataset [9]: A dataset of 698 30-second excerpts of ballroom dance music (total length 364 minutes). It was annotated with beat and downbeat times in [14].

Hainsworth dataset [10]: A dataset with 222 pieces (total length 199 minutes), covering a wide spectrum of genres.

SMC dataset [11]: A dataset with 217 pieces which are considered difficult for meter inference (total length 145 minutes). This set is also part of the MIREX evaluation.

Greek dataset [12]: 42 full songs of Cretan leaping dances in $2/4$ meter (total length 140 minutes).

Turkish dataset [12]: 82 one-minute excerpts of Turkish Makam music (total length 82 minutes).

Indian dataset [19]: The same subset of 118 two-minute long pieces (total length 235 minutes) as used in [12].

3.2 Evaluation metrics

To assess the ability of an algorithm to infer metrical structure, we used five evaluation metrics - four for beat tracking and one for downbeat tracking.

F-Measure (F): computed from the number of true positives (correctly detected beats within a window of ± 70ms around an annotation), the false positives, and the false negatives.

CMLt: quantifies the percentage of correctly tracked beats at the correct metrical level. In order to count a beat as correct, both previous and next beats have to match an annotation within a tolerance window of $\pm 17.5\%$ of the annotated beat interval.

AMLt: the same as CMLt, but the detected beats are also considered to be correct if they occur on the off-beat or at double or half of the ground-truth tempo.

Cemgil (Cem): places a Gaussian function with standard deviation of 40 ms around the annotations and computes the average likelihood of the corresponding beat closest to each annotation. In contrast to the other measures with hard decision boundaries (due to rectangular tolerance windows), this measure is also sensitive to small timing differences between annotated and detected beats.

Information Gain (D): measures the deviation of the beat error distribution from a uniform distribution by computing the Kullback-Leibler divergence.

Downbeat F-Measure (DB-F): is the same F-measure as used for beats, but considers only downbeats.

We implemented the evaluation metrics according to [3] with standard settings. To make them comparable with other work, we excluded the first five seconds in Experiment 2 when comparing with the model from [12] but did not exclude them when comparing with the results from [1].

3.3 Meter tracking models

To compare the proposed to the original model, we tested its performance with two state-of-the-art meter tracking systems:

RNN-BeatTracker [1]: This model uses a *recurrent neural network* to compute the probability of a frame being a beat. This probability is used as an observation probability for an HMM which jointly models tempo and the position

[1] Additional information as well as the code to reproduce the results of this paper are available at `http://www.cp.jku.at/people/krebs/ismir2015/`

within a beat period. We used the same MultiModelBeatTracker model as described in [1]. The model uses a frame length of 10 ms. Only beats are detected with this model.

GMM-BarTracker [12, 14]: Gaussian Mixture Models (GMMs) are used to compute the observation probabilities for an HMM that jointly models tempo, position within a bar and a set of rhythmic bar-patterns. For Experiment 1, the GMMs were trained on the *Ballroom*, the *Beatles* [3], the *Hainsworth* and the *RWC_Popular* [7] datasets, using three rhythmic patterns that correspond to the time signatures $2/4, 3/4$ and $4/4$. Pieces with other time signatures were excluded. For Experiment 2, we used an updated [2] version of the model described in [12]. The model uses a frame length of 20 ms and integrates eight rhythmic pattern states, one for each of the rhythmic classes. It outputs beats and downbeats.

Note that the difference between *original* and *proposed* lies only in the definition of the hidden states and the transition model; both use the same observation model, initial distribution, and tempo ranges.

4. RESULTS AND DISCUSSION

4.1 Experiment 1

In this experiment, we evaluated the influence of two parameters of the proposed transition model on the meter tracking performance. These parameters are the width of the tempo change distribution parametrised by the rate λ (Section. 2.3.3, Fig. 3) and the number of tempo states N (Section 2.3.1, Fig. 4). We chose to display the *Cemgil* accuracy in Figs. 3 and 4, because it is the only measure that makes a soft decision to count a beat as correct by using a Gaussian window and thus also takes into account small timing variations. Generally, the plots for the other measures were similar.

Fig. 3 shows the effect of the parameter λ on the *Cemgil* beat tracking accuracy for both the *RNN-BeatTracker* and the *GMM-BarTracker* on the *experimental* dataset, using the maximum number of tempo states N_{max}. The maximum *Cemgil* values were obtained with $\lambda = 125$, and $\lambda = 95$ respectively.

Using these settings for λ, we investigated the effect of the number of tempo states N on the beat tracking performance, which is shown in Fig. 4. As the two systems use a different audio frame rate, the maximum number of tempo states N_{max} is different too (see Section 2.3.1). Using a tempo range of $[55, 215]$ BPM as in [1], the *RNN-BeatTracker* has at most $N_{max} = 82$ tempo states, while for the *GMM-BarTracker* $N_{max} = 41$. As can be seen from Fig. 4, the *Cemgil* accuracy converges at ≈ 75 tempo states for the *RNN-BeatTracker* and at ≈ 40 for the *GMM-BarTracker*. This finding suggests that the *BarTracker* might also benefit from a higher audio frame rate and therefore a higher number of tempo states. In addition, the number of tempo states is a suitable parameter to select a trade-off between speed and accuracy.

[2] http://www.cp.jku.at/people/krebs/ismir2014/

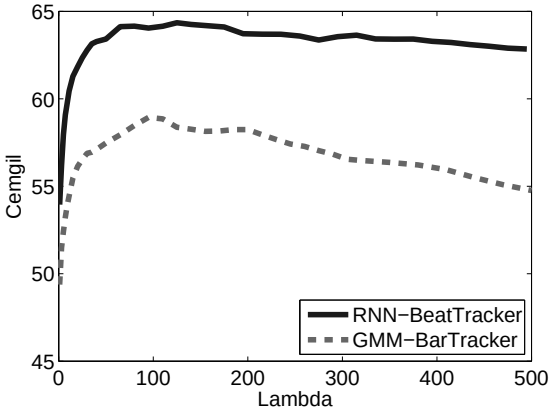


Figure 3: Effect of parameter λ on beat tracking *Cemgil* metric on the *experimental* dataset.

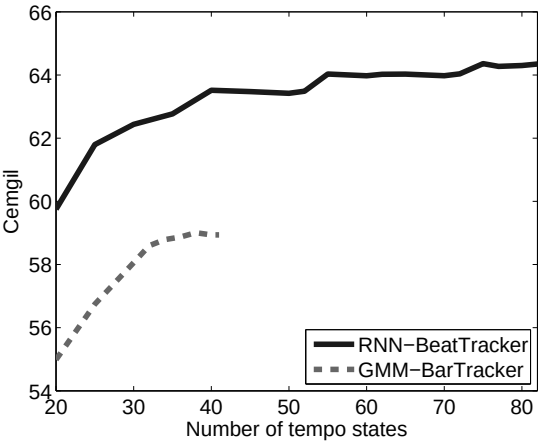


Figure 4: Effect of the number of tempo states on beat tracking *Cemgil* metric on the *experimental* dataset.

4.2 Experiment 2

In this experiment, we integrated the proposed model into two state-of-the-art meter tracking systems (Section 3.3) and compared them to the original models. The beat and downbeat accuracy scores of the original [1, 12] and proposed models, together with the number of states and transitions, are shown in Table 1. The proposed model used the parameters λ and N obtained in Experiment 1.

As can be seen, the proposed transition model outperforms the original model with respect to all performance metrics on all datasets (except *AMLt* (-0.2%) on the Ballroom dataset), with the added advantage of drastically reduced complexity. The *CMLt* metric in particular seems to benefit from the proposed model, with up to 20% relative improvement on the *Greek* dataset. Apparently, the restriction to change tempo only at beat times results in higher stability and therefore better performance in measures that are sensitive to continuity, such as *CMLt* and *AMLt*.

A comparison of the state-space sizes of the original and proposed models shows that the latter uses far fewer states and transitions. This is particularly apparent for the *GMM-*

	F	Cem	CMLt	AMLt	D	DB-F	States	Transitions
RNN-BeatTracker								
Ballroom								
Original [1] (20 tempo states)	0.910	0.845	0.830	0.924	3.469	-	11 520	33 280
Proposed (82 tempo states)	0.919	0.880+	0.854	0.922	3.552	-	5 617	8 343
Proposed (55 tempo states)	0.917	0.878	0.848	0.921	3.536	-	3 369	4 496
Hainsworth								
Original [1] (20 tempo states)	0.840	0.707	0.803	0.881	2.268	-	11 520	33 280
Proposed (82 tempo states)	0.851	0.730	0.805	0.885	2.337	-	5 617	8 343
Proposed (55 tempo states)	0.851	0.729	0.791	0.886	2.332	-	3 369	4 496
SMC								
Original [1] (20 tempo states)	0.529	0.415	0.428	0.567	1.460	-	11 520	33 280
Proposed (82 tempo states)	0.540	0.430	0.460	0.613	1.579	-	5 617	8 343
Proposed (55 tempo states)	0.543	0.431	0.458	0.613	1.578	-	3 369	4 496
GMM-BarTracker								
Greek								
Original [12] (18 tempo states)	0.916	0.810	0.778	0.952	2.420	0.777	133 200	376 800
Proposed (35 tempo states)	0.956	0.850	0.935+	0.965	2.625	0.812	26 716	41 708
Indian								
Original [12] (18 tempo states)	0.799	0.684	0.613	0.845	1.988	0.476	133 200	376 800
Proposed (35 tempo states)	0.850+	0.737+	0.703	0.942+	2.415+	0.515	26 716	41 708
Turkish								
Original [12] (18 tempo states)	0.861	0.679	0.694	0.840	1.431	0.617	133 200	376 800
Proposed (35 tempo states)	0.877	0.689	0.732	0.877	1.575	0.632	26 716	41 708

Table 1: Performance of the original and proposed transition model on the *Ballroom*, *Hainsworth*, *SMC*, *Greek*, *Indian*, and *Turkish* dataset. The + symbol denotes significant ($p < 0.05$) improvement over the result in the row above, using a one-way analysis of variance (ANOVA) test of significance.

BarTracker, which has *a priori* a larger state space because it models (a) bars instead of beats and (b) eight rhythmic patterns. With the original *GMM-BarTracker*, processing a four-minute piece (12 000 frames at 50 fps), required remembering 1.60×10^9 state ids in the Viterbi algorithm, which needs 6.39 GB stored as 32-bit integers. In contrast, using the proposed model, only 0.32×10^9 states must be stored - a demand that can be met using 16-bit integers in only 0.64 GB of memory. With a MATLAB implementation on an Intel Core i5-2400 CPU with 3.1 GHz, we can therefore reduce the computation time for the *Turkish* dataset from 45.8 minutes to 4.2 minutes, including the computation of the audio features (which takes only 18 seconds). Additionally, as already shown in Experiment 1, we can further reduce the number of tempo states from 82 (the maximum number of tempo states as computed in Section 2.3.1) to 55 with the *RNN-BeatTracker*, with only marginal performance decrease. Compared to the original model, this implies a reduction of the numbers of states and transitions by factors of three and seven, respectively. Since in the proposed model most position states are needed to model lower tempi, the lower tempo limits mainly determine the size of the state space.

5. CONCLUSIONS

In this paper, we have proposed a new discretisation and tempo transition model that can be used as a drop-in replacement for variants of the *bar pointer model*. We have shown that our model outperformed the original one in 32 of 33 test cases, while substantially reducing space and time complexity. We believe that this is an important step towards lightweight, real-time capable, high-performance meter inference systems.

As part of future work, we plan to investigate whether changing tempo only at beat positions also stabilises the particle filter versions of the *bar pointer* model [15, 18], which would further facilitate reducing computational complexity.

6. ACKNOWLEDGMENTS

This work is supported by the European Union Seventh Framework Programme FP7 / 2007-2013 through the Austrian Science Fund (FWF) project Z159 and the GiantSteps project (grant agreement no. 610591). Thanks to Ingrid Abfalter for proofreading.

7. REFERENCES

[1] S. Böck, F. Krebs, and G. Widmer. A multi-model approach to beat tracking considering heterogeneous music styles. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, 2014.

[2] T. Collins, S. Böck, F. Krebs, and G. Widmer. Bridging the audio-symbolic gap: The discovery of repeated note content directly from polyphonic music audio. In *AES 53rd International Conference on Semantic Audio*, London, 2014. Audio Engineering Society.

[3] M. Davies, N. Degara, and M. Plumbley. Evaluation methods for musical audio beat tracking algorithms. *Queen Mary University of London, Tech. Rep. C4DM-09-06*, 2009.

[4] C. Drake and M. Botte. Tempo sensitivity in auditory sequences: Evidence for a multiple-look model. *Perception & Psychophysics*, 54(3):277–286, 1993.

[5] S. Durand, J. Bello, D. Bertrand, and R. Gael. Downbeat tracking with multiple features and deep neural networks. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, 2015.

[6] G. Dzhambazov. Towards a drum transcription system aware of bar position. In *Proceedings of the AES 53rd International Conference on Semantic Audio*, London, 2014.

[7] M. Goto. AIST annotation for the RWC music database. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, pages 359–360, Victoria, 2006.

[8] F. Gouyon. *A computational approach to rhythm description-Audio features for the computation of rhythm periodicity functions and their use in tempo induction and music content processing*. PhD thesis, Universitat Pompeu Fabra, 2005.

[9] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1832–1844, 2006.

[10] S. Hainsworth and M. Macleod. Particle filtering applied to musical tempo tracking. *EURASIP Journal on Applied Signal Processing*, 2004:2385–2395, 2004.

[11] A. Holzapfel, M. Davies, J. Zapata, J. Oliveira, and F. Gouyon. Selective sampling for beat tracking evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(9):2539–2548, 2012.

[12] A. Holzapfel, F. Krebs, and A. Srinivasamurthy. Tracking the odd: Meter inference in a culturally diverse music corpus. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, 2014.

[13] F. Korzeniowski, S. Böck, and G. Widmer. Probabilistic extraction of beat positions from a beat activation function. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, 2014.

[14] F. Krebs, S. Böck, and G. Widmer. Rhythmic pattern modeling for beat and downbeat tracking in musical audio. In *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, Curitiba, 2013.

[15] F. Krebs, A. Holzapfel, A. T. Cemgil, and G. Widmer. Inferring metrical structure in music using particle filters. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(5):817–827, 2015.

[16] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[17] U. Şimşekli, O. Sönmez, B. Kurt, and A. Cemgil. Combined perception and control for timing in robotic music performances. *EURASIP Journal on Audio, Speech, and Music Processing*, 2012(1):1–20, 2012.

[18] A. Srinivasamurthy, A. Holzapfel, A. Cemgil, and X. Serra. Particle filters for efficient meter tracking with Dynamic Bayesian networks. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, Malaga, 2015.

[19] A. Srinivasamurthy and X. Serra. A supervised approach to hierarchical metrical cycle tracking from audio music recordings. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5217–5221, Florence, 2014.

[20] N. Whiteley, A. Cemgil, and S. Godsill. Bayesian modelling of temporal structure in musical audio. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, Victoria, 2006.

[21] N. Whiteley, A. Cemgil, and S. Godsill. Sequential inference of rhythmic structure in musical audio. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages IV–1321, Honolulu, 2007.

[22] J. Zapata, M. Davies, and E. Gómez. Multi-feature beat tracking. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 22(4):816–825, 2014.