A MULTI-MODEL APPROACH TO BEAT TRACKING CONSIDERING HETEROGENEOUS MUSIC STYLES

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

ABSTRACT

In this paper we present a new beat tracking algorithm which extends an existing state-of-the-art system with a multi-model approach to represent different music styles. The system uses multiple recurrent neural networks, which are specialised on certain musical styles, to estimate possible beat positions. It chooses the model with the most appropriate beat activation function for the input signal and jointly models the tempo and phase of the beats from this activation function with a dynamic Bayesian network. We test our system on three big datasets of various styles and report performance gains of up to 27% over existing state-of-the-art methods. Under certain conditions the system is able to match even human tapping performance.

1. INTRODUCTION AND RELATED WORK

The automatic inference of the metrical structure in music is a fundamental problem in the music information retrieval field. In this line, *beat tracking* deals with finding the most salient level of this metrical grid, the *beat*. The beat consists of a sequence of regular time instants which usually invokes human reactions like foot tapping. During the last years, beat tracking algorithms have considerably improved in performance. But still they are far from being considered on par with human beat tracking abilities – especially for music styles which do not have simple metrical and rhythmic structures.

Most methods for beat tracking extract some features from the audio signal as a first step. As features, commonly low-level features such as amplitude envelopes [20] or spectral features [2], mid-level features like onsets either in discretised [8,12] or continuous form [6,10,16,18], chord changes [12,18] or combinations thereof with higher level features such as rhythmic patterns [17] or metrical relations [11] are used. The feature extraction is usually followed by a stage that determines periodicities within the extracted features sequences. Autocorrelation [2,9,12] and comb filters [6,20] are commonly used techniques for this task. Most systems then determine the most predominant tempo from these periodicities and subsequently determine the beat times using *multiple agents* approaches [8,12], *dynamic programming* [6,10], *hidden Markov models (HMM)* [7,16,18], or *recurrent neural networks (RNN)* [2]. Other systems operate directly on the input features and jointly determine the tempo and phase of the beats using *dynamic Bayesian networks (DBN)* [3,14,17,21].

One of the most common problems of beat tracking systems are "octave errors", meaning that a system detects beats at double or half the rate of the ground truth tempo. For human tappers this generally does not constitute a problem, as can be seen when comparing beat tracking results at different metrical levels [6]. Hainsworth and Macleod stated that beat tracking systems will have to be style specific in the future in order to improve the state-of-the-art [14]. This is consistent with the finding of Krebs et al. [17] who showed on a dataset of Ballroom music that the beat tracking performance can be improved by incorporating style-specific knowledge, especially by resolving the octave error. While approaches have been proposed which combined multiple existing features for beat tracking [22], no one has so far combined several models specialised on different musical styles to improve the overall performance.

In this paper, we propose a multi-model approach to fuse information of different models that have been specialised on heterogeneous music styles. The model is based on the *recurrent neural network* (RNN) beat tracking system proposed in [2] and can be easily adapted to any music style without further parameter tweaking, only by providing a corresponding beat-annotated dataset. Further, we propose an additional *dynamic Bayesian network* stage based on the work of Whiteley et al. [21] which jointly infers the tempo and the beat phase from the beat activations of the RNN stage.

2. PROPOSED METHOD

The new beat tracking algorithm is based on the state-of-the-art approach presented by Böck and Schedl in [2]. We extend their system to be able to better deal with heterogeneous music styles and combine it with a dynamic Bayesian network similar to the ones presented in [21] and [17].

The basic structure is depicted in Figure 1 and consists of the following elements: first the audio signal is pre-processed and fed into multiple *neural network* beat track-

ing modules. Each of the modules is trained on different audio material and outputs a different beat activation function when activated with a musical signal. These functions are then fed into a module which chooses the most appropriate model and passes its activation function to a *dynamic Bayesian network* to infer the actual beat positions.

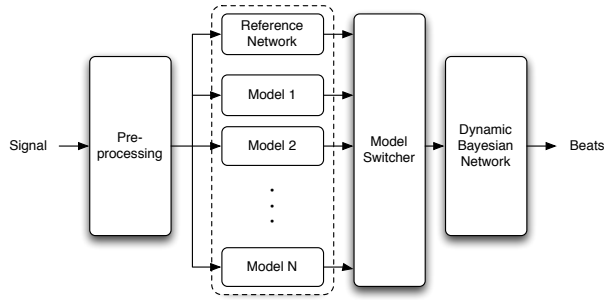

Figure 1. Overview of the new multi-model beat tracking system.

Theoretically, a single network large enough should be able to model all the different music styles simultaneously, but unfortunately this optimal solution is hardly achievable. The main reason for this is the difficulty to choose an absolutely balanced training set with an evenly distributed set of beats over all the different dimensions relevant for detecting beats. These include rhythmic patterns [17, 20], harmonic aspects and many other features. To overcome this limitation, we split the available training data into multiple parts. Each part should represent a more homogeneous subset than the whole set so that the networks are able to specialise on the dominant aspects of this subset.

It seems reasonable to assume that humans do something similar when tracking beats [4]. Depending on the style of the music, the rhythmic patterns present, the instrumentation, the timbre, they apply their musical knowledge to chose one of their "learned" models and then decide which musical events are beats or not. Our approach mimics this behaviour by learning multiple distinct models.

2.1 Signal pre-processing

All neural networks share the same signal pre-processing step, which is very similar to the work in [2]. As inputs to the different neural networks, the logarithmically filtered and scaled spectrograms of three parallel *Short Time Fourier Transforms (STFT)* obtained for different window lengths and their positive first order differences are used. The system works with a constant frame rate f_r of 100 frames per second. Window lengths of 23.2 ms, 46.4 ms and 92.9 ms are used and the resulting spectrogram bins of the discrete Fourier transforms are filtered with overlapping triangular filters to have a frequency resolution of three bands per octave. To put all resulting magnitude values into a positive range we add 1 before taking the logarithm.

2.2 Multiple parallel neural networks

At the core of the new approach, multiple neural networks are used to determine possible beat locations in the audio signal. As outlined previously, these networks are trained on material with different music styles to be able to better detect the beats in heterogeneous music styles.

As networks we chose the same *recurrent neural network (RNN)* topology as in [2] with three bidirectional hidden layers with 25 *long short-term memory (LSTM)* units per layer. For training of the networks, standard gradient descent with error backpropagation and a learning rate of $1e^{-4}$ is used. We initialise the network weights with a Gaussian distribution with mean 0 and standard deviation of 0.1. We use early stopping with a disjoint validation set to stop training if no improvement over 20 epochs can be observed.

One reference network is trained on the complete dataset until the stopping criterion is reached for the first time. We use this point during the training phase to diverge the specialised models from the reference network.

Afterwards, all networks are fine-tuned with a reduced learning rate of $1e^{-5}$ on either the complete set or the individual subsets (cf. Section 3.1) with the above mentioned stopping criterion. Given N subsets, $N + 1$ models are generated.

The output functions of the network models represent the beat probability at each time frame. Instead of tracking the beats with an autocorrelation function as described in the original work, the beat activation functions of the different models are fed into the next model-selection stage.

2.3 Model selection

The purpose of this stage is to select a model which outputs a better beat activation function than the reference model when activated with a signal. Compared to the reference model, the specialised models produce better predictions on input data which is similar to that used for fine-tuning, but worse predictions on signals dissimilar to the training data. This behaviour can be seen in Figure 2, where the specialised model produces higher beat activation values at the beat locations and lower values elsewhere.

Table 1 illustrates the impact on the *Ballroom* subset, where the relative gain of the best specialised model compared to the reference model ($+1.7\%$) is lower than the penalties of the other models (-2.3% to -6.3%). The fact that the performance degradation of the unsuitable specialised models is greater than the gain of the most suitable model allows us to use a very simple but effective method to choose the best model.

To select the best performing model, all network outputs of the fine-tuned networks are compared with the output of the reference network (which was trained on the whole training set) and the one yielding the lowest *mean squared difference* is selected as the final one and its output is fed into the final beat tracking stage.

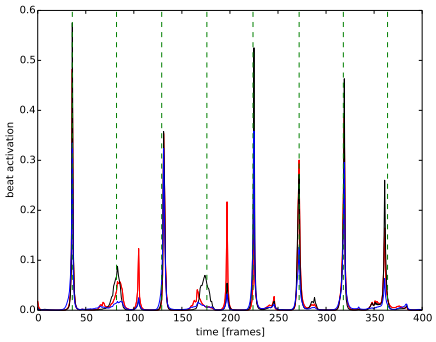

Figure 2. Example beat activations for a 4 seconds ballroom snippet. Red is the reference network's activations, black the selected model and blue a discarded one. Green dashed vertical lines denote the annotated beat positions.

	F-measure	Cemgil	AMLc	AMLt
SMC *	0.834	0.807	0.664	0.767
Hainsworth *	0.867	0.839	0.694	0.793
Ballroom *	0.904	0.872	0.777	0.853
Reference	0.887	0.855	0.748	0.831
Multi-model	0.897	0.866	0.759	0.841

Table 1. Performance of differently specialised models (marked with asterisks, fine-tuned on the *SMC*, *Hainsworth* and *Ballroom* subsets) on the *Ballroom* subset compared to the reference model and the network selected by the multi-model selection stage.

2.4 Dynamic Bayesian network

Independent of whether only one or multiple neural networks are used, the approach of Böck and Schedl [2] has a fundamental shortcoming: the final peak-picking stage does not try to find a global optimum when selecting the final locations of the beats. It rather determines the dominant tempo of the piece (or a segment of certain length) and then aligns the beat positions according to this tempo by simply choosing the best start position and then progressively locating the beats at positions with the highest activation function values in a certain region around the pre-determined position. To allow a greater responsiveness to tempo changes, this chosen region must not be too small. However, this also introduces a weakness to the algorithm, because the tracking stage can easily get distracted by a few misaligned beats and needs some time to recover from this fault. The activation function depicted in Figure 2 has two of these spurious detections around frames 100 and 200.

To circumvent this problem, we feed the output of the chosen neural network model into a *dynamic Bayesian network (DBN)* which jointly infers tempo and phase of a beat sequence. Another advantage of this new method is that we are able to model both beat and non-beat states, which was shown to perform superior to the case where only beat states are modelled [7].

The DBN we use is closely related to the one proposed in [21], adapted to our specific needs. Instead of modelling whole bars, we only model one beat period which reduces the size of the search space. Additionally we do not model rhythmic patterns explicitly and leave this higher level analysis to the neural networks. This finally leads to a DBN which consists of two hidden variables, the tempo ω and the position ϕ inside a beat period. In order to infer the hidden variables from an audio signal, we have to specify three entities: A *transition model* which describes the transitions between the hidden variables, an *observation model* which takes the beat activations from the neural network and transforms them into probabilities suitable for the DBN, and the *initial distribution* which encodes prior knowledge about the hidden variables. For computational ease we discretise the tempo-beat space to be able to use standard hidden Markov model (HMM) [19] algorithms for inference.

2.4.1 Transition model

The beat period is discretised into $\Phi = 640$ equidistant cells and $\phi \in \{1, ..., \Phi\}$. We refer to the unit of the variable ϕ (position inside a beat period) as *pib*. ϕ_k at audio frame k is then computed by

$$\phi_k = (\phi_{k-1} + \omega_{k-1} - 1) \bmod \Phi + 1. \tag{1}$$

The tempo space is discretised into $\Omega = 23$ equidistant cells, which cover the tempo range up to 215 beats per minute (BPM). The unit of the tempo variable ω is *pib per audio frame*. As we want to restrict ω to integer values (to stay within the ϕ grid at transitions), we need a high resolution of ϕ in order to get a high resolution of ω. Based on experiments with the training set, we set the tempo space to $\omega \in \{6, ..., \Omega\}$, where $\omega = 6$ is equivalent to a minimum tempo of $6 \times 60 \times f_r/\Phi \approx 56$ BPM. As in [21] we only allow for three tempo transitions at time frame k: It stays constant, it accelerates, or it decelerates.

$$\omega_k = \begin{cases} \omega_{k-1}, & P(\omega_k|\omega_{k-1}) = 1 - p_\omega \\ \omega_{k-1} + 1, & P(\omega_k|\omega_{k-1}) = \frac{p_\omega}{2} \\ \omega_{k-1} - 1, & P(\omega_k|\omega_{k-1}) = \frac{p_\omega}{2} \end{cases} \tag{2}$$

Transitions to tempi outside of the allowed range are not allowed by setting the corresponding transition probabilities to zero. The probability of a tempo change p_ω was set to 0.002.

2.4.2 Observation model

Since the beat activation function a produced by the neural network is limited to the range $[0, 1]$ and shows high values at beat positions and low values at non-beat positions, we use the activation function directly as state-conditional observation distributions (similar to [7]). We define the observation likelihood as

$$P(a_k|\phi_k) = \begin{cases} a_k, & 1 \leq \phi_k \leq \frac{\Phi}{\lambda} \\ \frac{1-a_k}{\lambda - 1}, & otherwise. \end{cases} \tag{3}$$

$\lambda \in [\frac{\Phi}{\Phi - 1}, \Phi]$ is a parameter that controls the proportion of the beat interval which is considered as beat and non-beat

location. Smaller values of λ (a higher proportion of beat locations and a smaller proportion of non-beat locations) are especially important for higher tempi, as the DBN visits only a few position states of a beat interval and could possibly miss the beginning of a beat. On the other hand, higher values of λ (a smaller proportion of beat locations) lead to less accurate beat tracking, as the activations are blurred in the state domain of the DBN. On our training set we achieved the best results with the value $\lambda = 16$.

2.4.3 Initial state distribution

The initial state distribution is normally used to incorporate any prior knowledge about the hidden states, such as tempo distributions. In this paper, we use a uniform distribution over all states, for simplicity and ease of generalisation.

2.4.4 Inference

We are interested in the sequence of hidden variables $\phi_{1:K}$ and $\omega_{1:K}$, that maximise the posterior probability of the hidden variables given the observations (activations $a_{1:K}$). Combining the discrete states of ϕ and ω into one state vector $\mathbf{x}_k = [\phi_k, \omega_k]$, we can compute the maximum a-posteriori state sequence $\mathbf{x}^*_{1:K}$ by

$$\mathbf{x}^*_{1:K} = \arg\max_{\mathbf{x}_{1:K}} p(\mathbf{x}_{1:K}|a_{1:K}). \quad (4)$$

Equation 4 can be computed efficiently using the well-known Viterbi algorithm [19]. Finally the set of beat times $\mathcal{B}$ are determined by the set of time frames k which were assigned to a beat position ($\mathcal{B} = \{k : \phi_k < \phi_{k-1}\}$). In our experiments we found that the beat detection becomes less accurate if the part of the beat interval which is considered as beat-state is too large (i.e. smaller values of λ). Therefore we determine the final beat times by looking for the highest beat activation value inside the beat-state window $\mathcal{W} = \{k : \phi_k \leq \frac{\Phi}{\lambda}\}$.

3. EVALUATION

For the development and evaluation of the algorithm we used some well-known datasets. This allows for highest comparability with previously published results of state-of-the-art algorithms.

3.1 Datasets

As training material for our system, the datasets introduced in [13–15] are used. They are called *Ballroom*, *Hainsworth* and *SMC* respectively. To show the ability of our new algorithm to adapt to various music styles, a very simple approach of splitting the complete dataset into multiple subsets according to the original source was chosen. Although far from optimal – both the *SMC* and *Hainsworth* datasets contain heterogeneous music styles – we still consider this a valid choice, since any "better" splitting would allow the system to adapt even further to heterogeneous styles and in turn lead to better results. At least the three sets have a somehow different focus regarding the music styles present.

3.2 Performance measures

In line with almost all other publications on the topic of beat tracking, we report the following scores:

F-measure : counts the number of true positive (correctly located beats within a tolerance window of $\pm$70 ms), false positive and negative detections;

P-score : measures the tracking accuracy by the correlation of the detections and the annotations, considering deviations within 20% of the annotated beat interval as correct;

Cemgil : places a Gaussian function with a standard deviation of 40 ms around the annotations and then measures the tracking accuracy by summing up the scores of the detected beats on this function normalising it by the overall length of the annotations or detections, whichever is greater;

CMLc & CMLt : measure the longest continuously segment (CMLc) or all correctly tracked beats (CMLt) at the correct metrical level. A beat is considered correct if it is reported within a 17.5% tempo and phase tolerance, and the same applies for the previously detected beat;

AMLc & AMLt : like CMLc & CMLt, but additionally allow offbeat and double/half as well as triple/third tempo variations of the annotated beats;

D & $\mathbf{D_g}$: the information gain (D) and global information gain (D_g) are phase agnostic measures comparing the annotations with the detections (and vice-versa) building a error histogram and then calculating the Kullback-Leibler divergence w.r.t. a uniform histogram.

A more detailed description of the evaluation methods can be found in [5]. However, since we only investigate offline algorithms, we do not skip the first five seconds for evaluation.

3.3 Results & Discussion

Table 2 lists the performance results of the reference implementation, Böck's *BeatTracker.2013*, and the various extensions proposed in this paper for all datasets. All results are obtained with 8-fold cross validation with previously defined splittings, ensuring that no pieces are used both for training or parameter tuning and testing purposes. Additionally, we compare our new approach to published stat-of-the-art results on the *Hainsworth* and *Ballroom* datasets.

3.3.1 Multi-model extension

As can be seen, the use of the *multi-model* extension almost always improves the results over the implementation it is based on, especially on the *SMC* set. The gain in performance on the *Ballroom* set was expected, since Krebs et al. already showed that modelling rhythmic patterns helps to increase the overall detection accuracy [17]. Although we did not split the set according to the individual rhythmic patterns, the overall style of ballroom music can be considered unique enough to be distinct from the other music

	F-measure	P-score	Cemgil	CMLc	CMLt	AMLc	AMLt	D	D_g
Ballroom									
BeatTracker.2013 [1,2]	0.887	0.863	0.855	0.719	0.795	0.748	0.831	3.404	2.596
— Multi-Model	0.897	0.875	**0.866**	0.740	0.814	0.759	0.841	**3.480**	**2.674**
— DBN	0.903	0.876	0.838	0.792	0.825	0.873	0.915	3.427	2.275
— Multi-Model + DBN	**0.910**	**0.881**	0.845	**0.800**	**0.830**	**0.885**	**0.924**	3.469	2.352
Krebs et al. [17]	0.855	0.839	0.772	0.745	0.786	0.818	0.865	2.499	1.681
Zapata et al. [22] †	0.767	0.735	0.672	0.586	0.607	0.824	0.860	2.750	1.187
Hainsworth									
BeatTracker.2013 [1,2]	0.832	0.843	0.712	0.618	0.756	0.655	0.807	2.167	1.468
— Multi-Model	0.832	0.847	**0.716**	0.617	0.761	0.652	0.809	2.171	**1.490**
— DBN	**0.843**	**0.867**	0.711	**0.696**	**0.808**	0.759	**0.883**	2.251	1.481
— Multi-Model + DBN	0.840	0.865	0.707	**0.696**	0.803	**0.760**	0.881	**2.268**	1.466
Zapata et al. [22] †	0.710	0.732	0.589	0.569	0.642	0.709	0.824	2.057	0.880
Davies et al. [6]	-	-	-	0.548	0.612	0.681	0.789	-	-
Peeters & Papadopoulos [18]	-	-	-	0.547	0.628	0.703	0.831	-	-
Degara et al. [7]	-	-	-	0.561	0.629	0.719	0.815	-	-
Human tapper [6] ‡	-	-	-	0.528	0.812	0.575	0.874	-	-
SMC									
BeatTracker.2013 [1,2]	0.497	0.598	0.402	0.238	0.360	0.279	0.436	1.263	0.416
— Multi-Model	0.514	0.617	**0.415**	0.257	0.389	0.296	0.467	1.324	0.467
— DBN	0.516	0.622	0.404	0.294	0.415	0.378	0.550	1.426	0.504
— Multi-Model + DBN	**0.529**	**0.630**	**0.415**	**0.296**	**0.428**	**0.383**	**0.567**	**1.460**	**0.531**
Zapata et al. [22] †	0.369	0.460	0.285	0.115	0.158	0.239	0.397	0.879	0.126

Table 2. Performance of the proposed algorithm on the *Ballroom* [13], *Hainsworth* [14] and *SMC* [15] datasets. *BeatTracker* is the reference implementation our *Multi-Model* and *dynamic Bayesian network (DBN)* extensions are built on. The results marked with † are obtained with Essentia's implementation of the multi-feature beat tracker. [1] ‡ denotes causal (i.e. online) processing, all listed algorithms use non-causal analysis (i.e. offline processing) with the best results in bold.

styles present in the other sets and the salient features can be exploited successfully by the multi-model approach.

3.3.2 Dynamic Bayesian network extension

As already indicated in the original paper [2] (and described earlier in Section 2.4), the original *BeatTracker* can be easily distracted by some misaligned beats and then needs some time to recover from any failure. The newly adapted dynamic Bayesian network beat tracking stage does not suffer from this shortcoming by searching for the globally best beat locations. The use of the DBN boosts the performance on all datasets for almost all evaluation measures. Interestingly, the Cemgil accuracy is degraded by using the DBN stage. This might be explained by the fact that the discretisation grid of the beat period beat positions becomes too coarse for low tempi (cf. Section 2.4.4) and therefore yields inaccurate beat detections, which especially affect the Cemgil accuracy. This is one of the issues that needs to be resolved in the future, especially for lower tempi where the penalty is the highest.

3.3.3 Comparison with other methods

Our new system set side by side with other state-of-the-art algorithms draws a clear picture. It outperforms all of them considerably – independently of the dataset and evaluation measure chosen. Especially the high performance boosts of the CMLc and CMLt scores on the *Hainworth* dataset highlight the ability to track the beats at the correct metrical level significantly more often than any other method.

Davies et al. [6] also list performance results of a human tapper on the same dataset. However it must be noted that these were obtained by online real-time tapping, hence they cannot be compared directly to the system presented. However, the system of Davies et al. can also be switched to causal mode (and thus being comparable to a human tapper). In this mode it achieved performance reduced by approximately 10% [6]. Adding the same amount to the reported tapping results of 0.528 CMLc and 0.575 AMlc suggests that our system is capable of performing as good as humans when continuous tapping is required.

On the *Ballroom* set we achieve higher results than the particularly specialised system of Krebs et al. [17]. Since our DBN approach is a simplified variant of their model, it can be assumed that the relatively low scores of the Cemgil accuracy and the information gain are due to the same reason – the coarse discretisation of the beat or bar states. Nonetheless, comparing the continuity scores (which have higher tolerance thresholds) we can still report an average increase in performance of more than 5%.

4. CONCLUSIONS & OUTLOOK

In this paper we have presented a new beat tracking system which is able to improve over existing algorithms by incorporating multiple models which were trained on different music styles and combining it with a dynamic Bayesian

[1] `http://essentia.upf.edu`, v2.0.1

network for the final inference of the beats. The combination of these two extensions yields a performance boost – depending on the dataset and evaluation measures chosen – of up to 27% relative, matching human tapping results under certain conditions. It outperforms other state-of-the-art algorithms in tracking the beats at the correct metrical level by 20%.

We showed that the specialisation on a certain musical style helps to improve the overall performance, although the method for splitting the available data into sets of different styles and then selecting the most appropriate model is rather simple. For the future we will investigate more advanced techniques for the selection of suitable data for the creation of the specialised models, e.g. splitting the datasets according to dance styles as performed by Krebs et al. [17] or applying unsupervised clustering techniques. We also expect better results from more advanced model selection methods. One possible approach could be to feed the individual model activations to the dynamic Bayesian network and let it choose among them.

Finally, the Bayesian network could be tuned towards using a finer beat positions grid and thus reporting the beats at more appropriate times than just selecting the position of the highest activation reported by the neural network model.

5. ACKNOWLEDGMENTS

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

6. REFERENCES

[1] MIREX 2013 beat tracking results. `http://nema.lis.illinois.edu/nema_out/mirex2013/results/abt/`, 2013.

[2] S. Böck and M. Schedl. Enhanced Beat Tracking with Context-Aware Neural Networks. In *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11)*, pages 135–139, Paris, France, September 2011.

[3] A. T. Cemgil, H. Kappen, P. Desain, and H. Honing. On tempo tracking: Tempogram Representation and Kalman filtering. *Journal of New Music Research*, 28:4:259–273, 2001.

[4] N. Collins. Towards a style-specific basis for computational beat tracking. In *Proceedings of the 9th International Conference on Music Perception and Cognition (ICMPC9)*, pages 461–467, Bologna, Italy, 2006.

[5] M. E. P. Davies, N. Degara, and M. D. Plumbley. Evaluation methods for musical audio beat tracking algorithms. Technical Report C4DM-TR-09-06, Centre for Digital Music, Queen Mary University of London, 2009.

[6] M. E. P. Davies and M. D. Plumbley. Context-dependent beat tracking of musical audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3):1009–1020, March 2007.

[7] N. Degara, E. Argones-Rúa, A. Pena, S. Torres-Guijarro, M. E. P. Davies, and M. D. Plumbley. Reliability-informed beat tracking of musical signals. *IEEE Transactions on Audio, Speech and Language Processing*, 20(1):290–301, January 2012.

[8] S. Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30:39–58, 2001.

[9] D. Eck. Beat tracking using an autocorrelation phase matrix. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*, volume 4, pages 1313–1316, Honolulu, Hawaii, USA, April 2007.

[10] D. P. W. Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 2007:51–60, 2007.

[11] A. Gkiokas, V. Katsouros, G. Carayannis, and T. Stafylakis. Music tempo estimation and beat tracking by applying source separation and metrical relations. In *Proceedings of the 37th International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012)*, pages 421–424, Kyoto, Japan, March 2012.

[12] M. Goto and Y. Muraoka. Beat tracking based on multiple-agent architecture a real-time beat tracking system for audio signals. In *Proceedings of the International Conference on Multiagent Systems*, pages 103–110, 1996.

[13] 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, September 2006.

[14] S. Hainsworth and M. Macleod. Particle filtering applied to musical tempo tracking. *EURASIP J. Appl. Signal Process.*, 15:2385–2395, January 2004.

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

[16] A. Klapuri, A. Eronen, and J. Astola. Analysis of the meter of acoustic musical signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):342–355, January 2006.

[17] 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 2013)*, pages 227–232, Curitiba, Brazil, November 2013.

[18] G. Peeters and H. Papadopoulos. Simultaneous beat and downbeat-tracking using a probabilistic framework: Theory and large-scale evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1754–1769, 2011.

[19] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.

[20] E. D. Scheirer. Tempo and beat analysis of acoustic musical signals. *The Journal of the Acoustical Society of America*, 103(1):588–601, 1998.

[21] 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 2006)*, pages 29–34, Victoria, BC, Canada, October 2006.

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