DON'T LOOK BACK: AN ONLINE BEAT TRACKING METHOD USING RNN AND ENHANCED PARTICLE FILTERING

Mojtaba Heydari and Zhiyao Duan[1]

Department of Electrical & Computer Engineering
University of Rochester, 500 Wilson Blvd, Rochester, NY 14627, USA
mheydari@ur.rochester.edu, zhiyao.duan@rochester.edu

ABSTRACT

Online beat tracking (OBT) has always been a challenging task. Due to the inaccessibility of future data and the need to make inference in real-time. We propose *Don't Look back!* (DLB), a novel approach optimized for efficiency when performing OBT. DLB feeds the activations of a unidirectional RNN into an enhanced Monte-Carlo localization model to infer beat positions. Most preexisting OBT methods either apply some offline approaches to a moving window containing past data to make predictions about future beat positions or must be primed with past data at startup to initialize. Meanwhile, our proposed method only uses activation of the current time frame to infer beat positions. As such, without waiting at the beginning to receive a chunk, it provides an immediate beat tracking response, which is critical for many OBT applications. DLB significantly improves beat tracking accuracy over state-of-the-art OBT methods, yielding a similar performance to offline methods.

Index Terms— Online beat tracking, particle filtering, Monte Carlo localization, causal inference, rhythm analysis

1. INTRODUCTION

Beat tracking is an important task in Music Information Retrieval (MIR) and is the core block of music rhythm analysis. It also has a vast range of usages for other MIR tasks (e.g., music transcription, music metadata generation, genre classification, music emotion recognition, and event analysis). Online Beat Tracking (OBT) refers to beat tracking in real time or a causal fashion. It enables further applications such as automatic music accompaniment, interactive music processing, and auto DJs.

In general, every music beat tracker has two main blocks. One is extracting some music features that are informative for the task and the second one is to infer beat positions based on the extracted features. For the first block, many approaches [1-2] extract an Onset Strength Signal (OSS) such as signal's spectral flux as the main feature for beat detection. Other methods [3-4-5] take advantage of deep neural network to directly obtain a beat activation function. With the OSS or beat activation function, in the second block, different techniques such as autocorrelation function (ACF) and peak picking [4], resonating comb filter (CF) [24], dynamic programming [8] or more advanced probabilistic models such as hidden Markov models and dynamic Bayesian networks (DBN) [25], Kalman filter [15], and *particle filtering* (PF) [17-18] are used to infer beat locations.

Compared to offline approaches, OBT faces unique challenges including partial data access, processing speed constraints, and the inability to correct previous detections. One category of existing OBT methods use a moving window to model signal regularity over the recent past signal to make beat predictions [7-8-9-10-11-13-14]. In particular, OBTAIN [6] uses cumulative OSS and autocorrelation response to locate beats. However, the cumulative OSS relies heavily on past data and cannot track tempo changes or fluctuations properly. Aubio [13-14] locates beat positions based on the weighted autocorrelation response in a context-dependent model of OSS. Böck and Schedl [7] used beat activations from an RNN with a Bidirectional Long Short-Term Memory (BLSTM) structure in a moving window to estimate local tempo and infer beats based on auto correlation function. Gkiokas and Katsouros [8] utilized the activation functions of a CNN and dynamic programing for a moving window to estimate next beat positions. There are several arguments against the moving window strategy. These include the potential for computational overload, the intrinsic difficulty in adapting these tracking strategies to causal and real-time scenarios, and the lack of continuity between windows [12].

There are a few OBT methods that do not use a moving window. IBT [12] is an OBT model based on the Beatroot [2] multi-agent offline approach, which infers beat positions without using a dynamic window. However, the initialization of the agents can be difficult, needing up to 5 seconds of audio. Shiu et al. [15] utilized Kalman filtering for beat tracking, but the underlying Gaussian assumption of the observation distribution can be too strong.

In contrast, Particle Filtering (PF) is a more general nonparametric approach for sequential decision-making problems. Duan and Pardo [16] used PF for online audio score alignment. Cemgil and Kappen [17] applied PF in tempo tracking and rhythm quantization given a prior on quantization locations. Hainsworth and Macleod [18] used it in tempo tracking. According to [17-18], a critical component of PF methods is the choice of an appropriate input feature and the related observation model. Another practical consideration is

[1] Thanks to National Science Foundation Grant 1846184.

reducing the number of particles to speed up inference for real-time applications.

In this paper, we propose a novel OBT model that integrates a unidirectional Recurrent Neural Network (RNN) for feature extraction and particle filtering for online decision making. In particular, the RNN predicts a beat activation function for each incoming audio frame following [25] . For the particle filtering, we utilize an efficient state space model [21] and propose a new observation model. We further propose informative priors to improve the performance. Experiments on the GTZAN dataset show that our proposed model outperforms state-of-the-art methods and can be utilized in situations with weaker processing capacity.

2. APPROACH

In this section, we describe our DLB approach to music OBT. The name of the model reflects the fact that it does not require the beat activations of past frames to make an inference about the current frame. Here are two points to clarify though. 1) No need to look back is an intrinsic property of PF models, not limited to our system. However, we highlight this fact because such a characteristic can be advantageous for OBT applications. 2) By saying no need for past activations, we specifically refer to the inference stage that unlike moving window approaches, it does not need previous activations. We do recognize that activation generation process itself uses an RNN to implicitly model long-term dependencies in past data.

2.1. Pre-processing

RNN structures have been an interesting choice for many time series applications, since they consider the relationship between adjacent frames of data. In particular, many recent works in related fields take advantage of RNNs with BLSTM neurons. The main advantage of BLSTM over LSTM is its better performance, which originates from leveraging future data consideration in addition to past data. However, given that causality (which makes the future data inaccessible) is one of the main conditions in OBT applications, we chose to use an LSTM neural net trained to extract beat activations as the input of our PF inference model. In the preprocessing module, a Hann window is applied to the current frame of the audio signal with a size of 46 ms and hop size of 10 ms. Then, after obtaining the magnitude spectrogram, a logarithmically spaced filters ranging from [30, 17k] Hz with 12 bands per octave, corresponding to semitone resolution is applied. The first-order temporal differences are concatenated to the log-scaled spectrogram and then, the total spectrogram is fed to an ensemble of 8 pretrained LSTM nets. Their output is averaged to get current frame's beat activation. Note that to obtain the activation from the ensemble of LSTMs, pretrained models obtained from Madmom python library [19] are used. After getting the activation of the current frame, we feed it to our PF inference block to classify it to either the beat or the non-beat class.

2.2. Inference

In this paper, we approach the problem from the PF localization perspective, which comprises of two steps of motion and correction. According to the importance sampling principle, a high dimensional probability distribution $p(x) = p'(x)/K$ with an unknown model, parameters, and normalization constant K, can be represented by a large number (N) of independent samples $x^{(i)}$ from a known arbitrary proposal distribution π(x). Even though there is no strict limitation in choosing π(x), choosing a closer one to the distribution of interest, leads to a more efficient representation of $p(x)$. Thus:

$$p(x) = \frac{p\prime(x)}{K} = \frac{p\prime(x)}{K\,\pi(x)}\pi(x) = \frac{p\prime(x)}{K\,\pi(x)} \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} \delta(x - x^{(i)})$$

$$= \lim_{N\to\infty} \sum_{i=1}^{N} \frac{\omega_i}{\sum_{i=1}^{N}\omega_i}\, \delta(x - x^{(i)}), \qquad (1)$$

where $\omega_i = p'(x^{(i)})/\pi(x^{(i)})$ is each particle i's importance weight.

Now, consider the unknown distribution of interest $p(x_k|y_k)$ in our decoding problem, where x_k and y_k are beat hidden states and the signal's observations at each step k, respectively. Note that y_k values do not require normalization. By considering Bayes filter and chain rule for the 1st order Markovian process, the problem is framed in PF as follows:

$$\omega_k^{(i)} = \frac{p'\left(x_{0:k}^{(i)}\middle|y_{0:k}\right)}{\pi\left(x_{0:k}^{(i)}\middle|y_{0:k}\right)} = \frac{p(y_{0:k}|x_k^{(i)})p(x_k^{(i)}|x_{0:k-1}^{(i)}, y_{0:k-1})}{\pi(x_k^{(i)}|x_{k-1}^{(i)}, y_{0:k})}\,\omega_{k-1}^{(i)}\,. \quad (2)$$

In formula (2), by choosing the proposal distribution equal to the already known transition probability, the problem boils down to a sequential Monte Carlo posterior calculation, i.e., let

$$\pi(x_k^{(i)}|x_{k-1}^{(i)}, y_{0:k}) = p(x_k^{(i)}|x_{0:k-1}^{(i)}, y_{0:k}) \quad , \qquad (3)$$

then

$$\omega_k^{(i)} = p(y_k|x_{0:k-1}^{(i)})\,\omega_{k-1}^{(i)}\,. \qquad (4)$$

Formulae (3) and (4) state that if we sample from a transition distribution (motion) as a proposal distribution of π, we can update the weights simply by multiplying them with the likelihood of the current frame. Thus, to infer beat positions using the posterior (1) estimation for each frame, the following algorithm will be used:

*1- Sample particles from proposal distribution (3) which in our case is transition probability. (**motion**)*

2- Compute the new importance weights (4) based on observation probability derived from LSTM Beat activations.

*3- Resample based on new normalized weights that discard unlikely hypotheses and generate more rational ones. (**correction**)*

4- Take the median of the positions of all particles and classify the frame as beat if it is within the beat boundary and is far from previous beat with a dynamic time threshold equal to half of the median of all particle's tempo.

2.3. State space and transition model

One popular state space model for such problems is the bar pointer model [20] which jointly models tempo and position within the bar as hidden variables. However, for the state space and transition model in this paper, rather than using the common bar pointer model, we implemented the model that is described in [21]. According to that, at each time frame k, the hidden state is referred to as x_k=[$\Phi_k, \dot{\Phi}_k$], with $\Phi_k \in \{1,2,\ldots,M\}$ denoting the position within the bar and $\dot{\Phi}_k \in \{\dot{\Phi}_{min}, \dot{\Phi}_{min}+1,\ldots,\dot{\Phi}_{max}\}$ denoting the jumping intervals representing tempo of the current state. State space and transition probability are then defined as follows:

$$M = \lfloor \frac{60}{T\times\Delta} \rfloor \ , \qquad (5)$$

$$p(\dot{\Phi}_k|\dot{\Phi}_{k-1}) = \begin{cases} exp(-\lambda \times |\dfrac{\dot{\Phi}_k}{\dot{\Phi}_{k-1}} - 1|) & for\ \Phi_k \in B \\ \dot{\Phi}_k = \dot{\Phi}_{k-1} & for\ \Phi_k \notin B \end{cases} . \quad (6)$$

Equation (5) relates each tempo in BPM denoted by T to its corresponding integer jumping interval M, and Δ denotes frame hop size in seconds.. Each row of the 2D state space shown in Fig.1 represents a different M. ΔNote that we model beat state space rather than bar state space. Equation (6) is the transition probability where λ is the parameter to decide the sampling width, B is beat states collection (yellow dots in Fig. 1). To specify the B boundaries that determine beats/non-beat states, we propose three models as shown in Fig. 1. Model (a) sets a vertical line as beat/non-beat boundary. Model (b) considers a constant number of states at each row as beat states. Model (c) provides a smooth Gaussian transition between b_k and γ.

We believe that our described model is very suitable for PF applications for two main reasons: 1- It needs fewer number of states compared to the original beat pointer model (→fewer number of particles can represent the state). 2- Allowing tempo shifts only in beat positions lowers computational cost. Instead of sampling in every step, we do it only in beat positions (B states), and for the rest of states, given that the tempo is the same, we only shift particles.

Considering the transition model, to detect tempo changes, no jitter or particle noise is required. For double/half tempo investigation, some works add a Metropolis-Hastings step to the algorithm. However, it was found in [18] that this step was counterproductive. In this paper we tried two different models for double/half tempo investigation. One was adding some probability terms to $\dot{\Phi}_k/2$ and $\dot{\Phi}_k \times 2$ in the first line of the transition probability (6). However, since the transition probability affects all the particles, it caused the model to diverge. The other method was adding double/half tempo investigation only to the median of tempi of all particles (interpreted as current frame's tempo). So, we feed the information of the median of tempi to the resampling block as informative priors.

2.3. Observation model

In this paper, we present a new observation model in which, localization is conducted only using the presence of landmarks as follows:

$$p(y_k|x_k) = \begin{cases} b_k & x_k \in B \\ \gamma & x_k \notin B \end{cases} , \quad (7)$$

where b_k is the beat activation of frame k, and γ is a small constant number (e.g., $\gamma = 0.03$). This means that in our approach, rather than using non-beat activations (which are the second output of the softmax in the RNN and fixed to be complementary to the beat activations) as clues in the likelihood function, we only rely on beat activations as landmarks and set all non-beat states probabilities to a small constant number. This is because the softmax output of the RNN is the posterior probability of beat/non-beat instead of the likelihood. Our experiments also showed that low beat activation (→high non-beat activation) does not necessarily mean that the frame is not a beat frame. In many cases, due to reasons like low percussive content or low frame energy, beat activations are weak. In traditional approaches, the complement of these activations encourages the model to classify the frame as s non-beat [21-25]. Figure 2 demonstrates the whole PF inference process. A demo of the performance of our model can be found in following footnote[2].

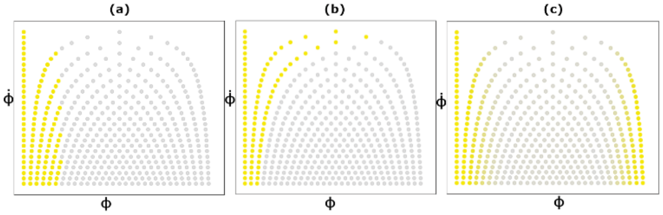

Fig. 1 Different beat/non-beat boundary models in state space. (a): fractional beat/non-beat discriminator (b): constant number discriminator. (c): Gaussian soft discriminator.

3. EVALUATION

In order to evaluate the performance of our online model, we compare it with state-of-the-art online methods as well as state-of-the-art offline methods.

3.2. Evaluation metrics

In line with most other beat tracking publications, F-measure (F_1) with a tolerance window of ±70 (ms) is reported as the main performance criteria. To investigate the required time for each method to become initialized, we evaluate with and without discarding the predictions of the first 5 seconds of each recording.

3.2. Dataset

We used GTZAN dataset [22-23], a large dataset containing 1000 music pieces with a duration of 30 seconds each covering 10 different music genres (e.g., blues, classic). We used the entire dataset as the test set to illustrate the performance of different methods in a more general fashion. It is important to note that this dataset is not used to train the RNN.

3.3. Evaluation results

The DLB model does not require informative priors in the initialization step. Therefore, to initialize early particles we used a random uniform distribution that gives the same chance of receiving a hypothesis to each state. N=1000 is the number of particles that are used in Table 1 evaluations. The transition parameter is λ=30, and for the observation model, the fractional discriminator with the parameter of 1/60 is used. As shown in table 1, for both 5SecSkip and NoSkip, The DLB method outperforms all state-of-the-art online beat tracking models. Comparison between two F_1 columns in table 1 provides insights about the performance quality of different methods at the beginning, comparing to their general performance. Shifting from 5SecSkip to the NoSkip setting, the performance of ACF and IBT models decrease dramatically, while DLB and Aubio decline mildly. However, one important point to mention is that the IBT model's small number of agents must be initialized using the first 5 seconds of audio. It does not deliver any output during this time period. Similarly, the ACF model needs to wait for 6 seconds at the beginning to calculate the autocorrelation function to estimate beat positions. Note that in this evaluation, ACF is implemented in a causal fashion. Also, the lowest performing

[2] https://www.youtube.com/watch?v=u2Ee6WsNzoU

deference with and without skip belongs to Aubio. The reason is that unlike the DLB model, which initializes particles randomly and delivers results immediately (figure.2 a), Aubio is more patient and by considering relations between a few early beats, comes up with more careful results at the beginning.

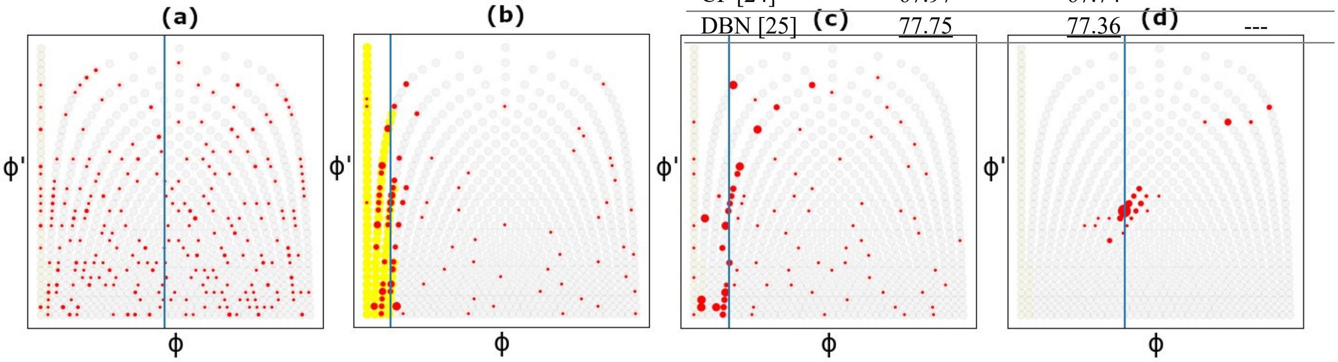

Fig. 2 Proposed PF inference process. (a): particles are initialized randomly and start to move right one step per frame (b): particles within the beat boundary gain weight while many others get discarded, when the first strong beat activation arrives. (c): significant gatherings move right with different paces. (d): Upon the next beat activation's arrival, many gatherings are discarded and the one with the correct tempo survives; a few double tempo investigators are also added. Blue line is the median of particles' positions.

In addition to online models, in the remainder of Table 1, performances of two offline methods are reported as well. The objective of considering offline methods is to compare between their inference and that of particle filtering. The compared offline models are resonating comb filters [24] and a dynamic Bayesian network [5], which is the state-of-the-art for offline beat tracking. It is important to note that even though the original offline models used BLSTM neurons in their RNN structure (which led to better performance in offline fashion), we replaced them with LSTM neurons in order to draw a fair comparison between DLB inference and those inferences. So, the input of all three models is common. Even considering DLB's causal setup, its performance is close to the non-causal DBN model. In addition to performance, the other very important factor in real-time usage is processing speed. PF models are slow in general. However, since our model uses an efficient state and transition model which only requires resampling in beat positions (for the rest of states, particles just shift forward by one step per frame) and uses stochastic universal sampling in the correction block (which only needs one random number generation rather than N random numbers for N particles), then our model using the default setting of N=1000 is already works in real time. However, to make it usable in situations with weaker processing capacity (e.g. microcontrollers), it is better to use a smaller number of particles.

Table 1: F-measure report of online/offline beat tracking models and initialization time for online models (GTZAN dataset).

Online beat tracking methods			
Method	*F-measure (5 Sec. skip)*	*F-measure (without skip)*	*Initialization elapse Time*
DLB	73.77	71.44	0 seconds
ACF [7]	64.63	51.74	6 seconds
IBT [12]	68.99	62.75	5 seconds
Aubio [13]	57.09	55.91	3 consistent beat periods
Offline beat tracking methods			
CF [24]	67.97	67.74	---
DBN [25]	77.75	77.36	---

Figure 3 demonstrates the performance of DLB model on the GTZAN dataset for different number of particles. We can see that for greedy filtering setups such as N=300, DLB performs almost as well as the default setting. For very greedy setups such as N=100, it is still in the same performance range with state-of-the-art methods. The reason for the performance robustness against a low number of particles is that by using the efficient space state (rather than the normal bar pointer state space), positions and tempi can be represented with a smaller number of states. Also, contrary to the double/half tempo investigator's ineffectiveness in the high number of particles, they somewhat assist the performance in the settings with low number of particles, by adding a wider viewpoint to monitor.

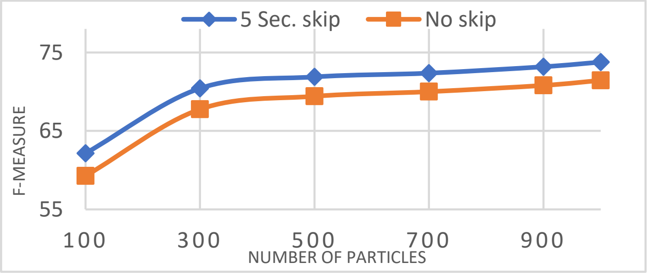

Fig. 3 DLB performance for different number of particles.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a novel online beat tracking method that leverages the beat activations of a unidirectional RNN in a sequential Monte Carlo particle filtering paradigm to infer music beat positions. By utilizing an optimized structure, we improved particle filtering performance in online beat tracking application. The proposed model is efficient and significantly outperforms state-of-the-art online beat tracking models. Also, we found that the performance of our casual inference model is close to that of non-casual state-of-the-art offline models. One direction for future work is to consider more rhythmic information in the model such as downbeat inference and meter analyses.

5. REFERENCES

[1] B. Dalton, D. Johnson, and G. Tzanetakis, "Daw-integrated beat tracking for music production," *Proc. Sound Music Comput. Conf.*, pp. 7–11, 2019.

[2] S. Dixon, "Automatic extraction of tempo and beat from expressive performances," *Int. J. Phytoremediation*, vol. 21, no. 1, pp. 39–58, 2001, doi: 10.1076/jnmr.30.1.39.7119.

[3] M. E. P. Davies and S. Böck, "Temporal convolutional networks for musical audio beat tracking," *Eur. Signal Process. Conf.*, vol. 2019-Septe, 2019, doi: 10.23919/EUSIPCO.2019.8902578.

[4] S. Böck, M. E. P. Davies, and P. Knees, "Multi-task learning of tempo and beat: Learning one to improve the other," *Proc. 20th Int. Soc. Music Inf. Retr. Conf. ISMIR 2019*, pp. 486–493, 2019.

[5] B. Di Giorgi, M. Mauch, and M. Levy, "Downbeat tracking with tempo-invariant convolutional neural networks," pp. 216–222, ISMIR 2020.

[6] A. Mottaghi, K. Behdin, A. Esmaeili, M. Heydari, and F. Marvasti, "OBTAIN: Real-Time Beat Tracking in Audio Signals Index Terms—Onset Strength Signal, Tempo estimation, Beat onset, Cumulative Beat Strength Signal, Peak detection," *Adv. Commun. Res. Inst.*, 2017.

[7] S. Böck and M. Schedl, "Enhanced beat tracking with context-aware neural networks," *Proc. 14th Int. Conf. Digit. Audio Eff. DAFx 2011*, pp. 135–140, 2011.

[8] A. Gkiokas and V. Katsouros, "Convolutional neural networks for real-time beat tracking: A dancing robot application," *Proc. 18th Int. Soc. Music Inf. Retr. Conf. ISMIR 2017*, pp. 286–293, 2017.

[9] G. Peeters, "Template-based estimation of time-varying tempo," *EURASIP J. Adv. Signal Process.*, vol. 2007, 2007, doi: 10.1155/2007/67215.

[10] D. P. W. Ellis, "Beat tracking by dynamic programming," *J. New Music Res.*, vol. 36, no. 1, pp. 51–60, 2007, doi: 10.1080/09298210701653344.

[11] A. P. Klapuri, A. J. Eronen, and J. T. Astola, "Analysis of the meter of acoustic musical signals," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 14, no. 1, pp. 342–355, 2006, doi: 10.1109/TSA.2005.854090.

[12] J. L. Oliveira, F. Gouyon, L. G. Martins, and L. P. Reis, "IBT: A real-time tempo and beat tracking system," *Proc. 11th Int. Soc. Music Inf. Retr. Conf. ISMIR 2010*, pp. 291–296, 2010.

[13] P. M. Brossier, "Automatic Annotation of Musical Audio for Interactive Applications," no. August, 2006.

[14] M. E. P. Davies, P. M. Brossier, and M. D. Plumbley, "Beat tracking towards automatic musical accompaniment," *Audio Eng. Soc. - 118th Conv. Spring Prepr. 2005*, vol. 2, pp. 751–757, 2005.

[15] Y. Shiu, N. Cho, P. C. Chang, and C. C. J. Kuo, "Robust on-line beat tracking with Kalman filtering and probabilistic data association (KF-PDA)," *IEEE Trans. Consum. Electron.*, vol. 54, no. 3, pp. 1369–1377, 2008, doi: 10.1109/TCE.2008.4637629.

[16] Z. Duan and B. Pardo, "A state space model for online polyphonic audio-score alignment," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, pp. 197–200, 2011, doi: 10.1109/ICASSP.2011.5946374.

[17] A. T. Cemgil and B. Kappen, "Monte Carlo methods for tempo tracking and rhythm quantization," *J. Artif. Intell. Res.*, vol. 18, pp. 45–81, 2003, doi: 10.1613/jair.1121.

[18] S. W. Hainsworth and M. D. Macleod, "Particle filtering applied to musical tempo tracking," *EURASIP J. Appl. Signal Processing*, vol. 2004, no. 15, pp. 2385–2395, 2004, doi: 10.1155/S1110865704408099.

[19] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "Madmom: A new python audio and music signal processing library," *MM 2016 - Proc. 2016 ACM Multimed. Conf.*, pp. 1174–1178, 2016, doi: 10.1145/2964284.2973795.

[20] N. Whiteley, A. T. Cemgil, and S. Godsill, "Bayesian modelling of temporal structure in musical audio," *ISMIR 2006 - 7th Int. Conf. Music Inf. Retr.*, no. December 2014, pp. 29–34, 2006.

[21] F. Krebs, S. Böck, and G. Widmer, "An efficient state-space model for joint tempo and meter tracking," in *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015*, 2015, pp. 72–78.

[22] U. Marchand and G. Peeters, "Swing ratio estimation," *DAFx 2015 - Proc. 18th Int. Conf. Digit. Audio Eff.*, no. part 2, pp. 1–6, 2015.

[23] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, pp. 293–302, 2002, doi: 10.1109/TSA.2002.800560.

[24] S. Böck, F. Krebs, and G. Widmer, "Accurate tempo estimation based on recurrent neural networks and resonating comb filters," *Proc. 16th Int. Soc. Music Inf. Retr. Conf. ISMIR 2015*, pp. 625–631, 2015.

[25] S. Böck, F. Krebs, and G. Widmer, "A multi-model approach to beat tracking considering heterogeneous music styles," *Proc. 15th Int. Soc. Music Inf. Retr. Conf. ISMIR 2014*, no. Ismir, pp. 603–608, 2014.