Accurate Tempo Estimation based on Recurrent Neural Networks and Resonating Comb Filters

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 tempo estimation algorithm which uses a bank of resonating comb filters to determine the dominant periodicity of a musical excerpt. Unlike existing (comb filter based) approaches, we do not use hand-crafted features derived from the audio signal, but rather let a recurrent neural network learn an intermediate beat-level representation of the signal and use this information as input to the comb filter bank. While most approaches apply complex post-processing to the output of the comb filter bank like tracking multiple time scales, processing different accent bands, modelling metrical relations, categorising the excerpts into slow / fast or any other advanced processing, we achieve state-of-the-art performance on nine of ten datasets by simply reporting the highest resonator's histogram peak.

1. INTRODUCTION

Tempo estimation is one of the most fundamental music information retrieval (MIR) tasks. The tempo of music corresponds to the frequency of the beats, i.e. the speed at which humans usually tap to the music.

In this paper, we only deal with global tempo estimation, i.e. report a single tempo estimate for a given musical piece, and do not consider the temporal evolution of tempo. Possible applications for such algorithms include automatic DJ mixing, similarity estimation, music recommendation, playlist generation, and tempo aware audio effects. Finding the correct tempo is also vital for many beat tracking algorithms which use a two-folded approach of first estimating the tempo of the music and then aligning the beats accordingly.

Many different methods for tempo estimation have been proposed in the past. While early approaches estimated the tempo based on discrete time events (e.g. MIDI notes or a sequence of onsets) [6], almost all of the recently proposed algorithms [4, 7, 8, 17, 23, 28] use some kind of continuous input. Generally, they follow this procedure: they transform the audio signal into a down-sampled feature, estimate the periodicities and finally select one of the periodicities as tempo.

As a reduction function, the signal's envelope [26], band pass filters [8, 17, 28], onset detection functions [4, 8, 23, 28] or combinations thereof are commonly used. Popular choices for periodicity detection include Fast Fourier Transform (FFT) based methods like tempograms [3, 28], autocorrelation [6, 8, 23, 25] or comb filters [4, 17, 26]. Finally, post-processing is applied to chose the most promising periodicity as perceptual tempo estimate. These post-processing methods range from simply selecting the highest periodicity peak to more sophisticated (machine learning) techniques, e.g. hidden Markov models (HMM) [17], Gaussian mixture model (GMM) regression [24] or support vector machines (SVM) [9, 25].

In this paper, we propose to use a neural network to derive a reduction function which makes complex post-processing redundant. By simply selecting the comb filter with the highest summed output, we achieve state-of-the-art performance on nine of ten datasets in the *Accuracy 2* evaluation metric.

2. RELATED WORK

In the following, we briefly describe some important works in the field of tempo estimation. Gouyon et al. [12] give an overview of the first comparative algorithm evaluation which took place for ISMIR 2004, followed by another study by Zapata and Gómez [29].

The work of Scheirer [26] was the first one to process the audio signal continuously rather than working on a series of discrete time events. He proposed the use of resonating comb filters, which are one of the main techniques used for periodicity estimation since then. Periodicity analysis is performed on a number of band pass filtered signals and then the outputs of this analysis are combined and a global tempo is reported.

Dixon [6] uses discrete onsets gathered with the spectral flux method to build clusters of inter onset intervals which are in turn processed by a multiple agent system to find the most likely tempo. Oliveira et al. [23] extend this approach to use a continuous input signal instead of discrete time events and modified it to allow causal processing.

Klapuri et al. [17] jointly analyse the musical piece at three time scales: the tatum, tactus (which corresponds to the beat or tempo) and measure level. The signal is split into multiple bands and then combined into four accent bands before being fed into a bank of resonating comb filters similar to [26]. Their temporal evolution and the relation of the different time scales are modelled with a probabilistic framework to report the final position of the beats. The tempo is then calculated as the median of the beat intervals during the second half of the signal.

Instead of a multi-band approach as used in [17, 26], Davies and Plumbley [4] process an autocorrelated version of a complex domain onset detection function with a shift invariant comb filter bank to get the beat period. Although this method uses only a single dimensional input feature, it performs almost as good as the competing algorithms in [12] but has much lower computational complexity.

Gainza and Coyle [8] use a multi-band decomposition to split the audio signal into three frequency bands and then perform a transient / onsets detection (with different onset detection methods). These are transformed via autocorrelation into periodicity density functions, combined, and weighted to extract the final tempo.

Gkiokas et al. [9] utilise harmonic / percussive source separation on top of a constant-Q transformed signal in order to extract chroma features and filter bank energies from the separated signal respectively. Periodicity is estimated for both representations with a bank of resonating comb filters for overlapping windows of 8 seconds length and the resulting features are combined before a metrical level analysis is performed to report the final tempo. In a consecutive work [10] they use a support vector machine (SVM) to classify the music into tempo classes to better predict the tempo to be reported.

Elowsson et al. [7] also use harmonic / percussive source separation to model the speed of music. They derive various features like onset densities (for multiple frequency ranges) and strong onset clusters and use a regression model to predict the tempo of the signal.

Percival and Tzanetakis [25] use a "traditional" approach by first generating a spectral flux onset strength signal, followed by a stage which detects the beat period in overlapping windows of approximately 6 seconds length (via generalised autocorrelation with harmonic enhancement) and a final accumulating stage which gathers all these tempo estimates and uses a support vector machine (SVM) to decide which octave the tempo should be in.

Wu and Jang [28] first derive an unaltered and a low pass filtered version of the input signal. Then they obtain a tempogram representation of a complex domain onset detection function for both signals to obtain tempo pairs. A classifier is then used to report the final most salient tempo.

3. ALGORITHM DESCRIPTION

Scheirer [26] found it beneficial to compute periodicities individually on multiple frequency bands and then subsequently combine them to estimate a single tempo. Klapuri et al. [17] followed this route but Davies and Plumpley argued that is is enough to have a single – musically meaningful – feature to estimate the periodicity of a signal [4].

Given the fact that beats are the musically most relevant descriptors for the tempo of a musical piece, we take this approach one step further and do not use the pre-processed signal directly – or any representation that is strongly correlated with it, e.g. an onset detection function – as an input for a comb filter, but rather process the signal with a neural network which is trained to predict the positions of beats inside the signal. The resulting beat activation function is then fed into a bank of resonating comb filters to determine the tempo.

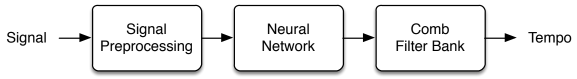


Figure 1: Overview of the new tempo estimation system.

Figure 1 gives general overview over the different steps of the tempo estimation system, which are described into more detail in the following sections.

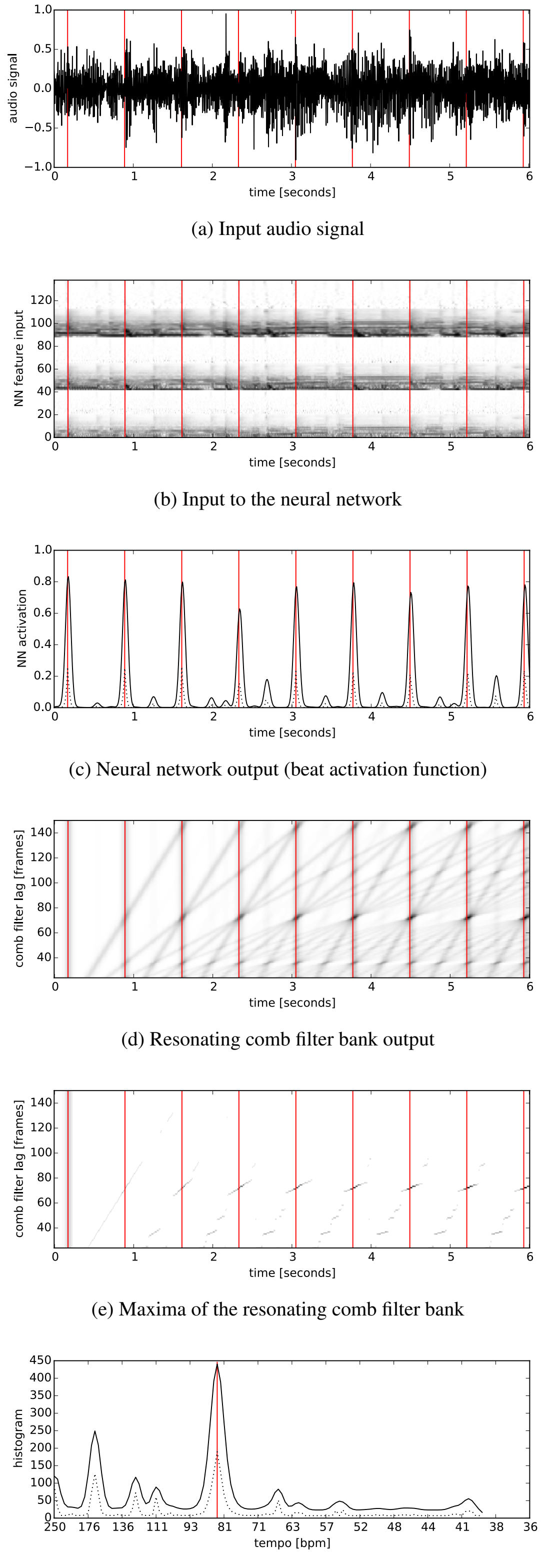
3.1 Signal Pre-Processing

The proposed system processes the signal in a frame-wise manner. Therefore the audio signal is split into overlapping frames and weighted with a Hann window of same length before being transferred to a time-frequency representation by means of the Short-time Fourier Transform (STFT). Two adjacent frames are located 10 ms apart, which corresponds to a rate of 100 fps (frames per second). We omit the phase portion of the complex spectrogram and use only the magnitudes for further processing. To reduce the dimensionality of the signal, we process it with a logarithmically spaced filter which has three bands per octave and is limited to the frequency range [30, 17000] Hz. To better match the human's perception of loudness, we scale the resulting frequency bands logarithmically. As the final input features for the neural network, we stack three spectrograms and their first order difference calculated with different STFT sizes of 1024, 2048 and 4096 samples, a visualisation is given Figure 2b.

3.2 Neural Network Processing

As a network we chose the system presented in [1], which is also the basis for the current state-of-the-art in beat tracking [2, 18]. The output of the neural network is a beat activation function, which represents the probability of a frame being a beat position. Instead of processing the beat activation function to extract the positions of the beats, we use it directly as a one-dimensional input to the bank of resonating comb filters.

Using this continuous function instead of discrete beats is advantageous since the detection is never 100% effective und thus introduces errors when inferring the tempo directly from the beats. This is in line with the observation that recent tempo induction algorithms use onset detection functions or other continuously valued inputs rather than discrete time events.



(a) Input audio signal

(b) Input to the neural network

(c) Neural network output (beat activation function)

(d) Resonating comb filter bank output

(e) Maxima of the resonating comb filter bank

(f) Weighted histogram with summed maxima

Figure 2: Signal flow of a 6 second pop song excerpt: *(a)* input audio signal, *(b)* pre-processed input to the neural network, *(c)* its raw (dotted) and smoothed (solid) output, *(d)* corresponding comb filter bank response, *(e)* the maxima thereof, *(f)* resulting raw (dotted) and smoothed (solid) weighted histogram of the summed maxima. The beat positions and the tempo are marked with vertical red lines.

We believe that the learned feature representation (at least to some extent) incorporates information that otherwise would have to be modelled explicitly, either by tracking multiple time scales [17], processing multiple accent bands [26], modelling metrical relations [9], dividing the excerpts into slow / fast categories [7] or any other advanced processing. Figure 2c shows an exemplary output of the neural network. It can be seen that the network activation function has strong regular peaks that do not always coincide with high energies in the network's inputs.

3.2.1 Network Training

We train the network on the datasets described in Section 4.2 which are marked with an asterisk (*) in an 8-fold cross validation setting based on a random splitting of the datasets. We initialise the network weights and biases with a uniform random distribution with range $[-0.1, 0.1]$ and train it with stochastic gradient decent with a learning rate of 10^{-4} and a momentum of 0.9. We stop training if no improvement of the cross entropy error of the validation set can be observed for 20 epochs. All adjustable parameters of the system are tuned to maximise the tempo estimation performance on the validation set.

3.2.2 Activation Function Smoothing

The beat activation function of the neural network reflects the probability that a given frame is a beat position. However, it can happen that the network is not sure about the exact position of the beat if it falls close to the border between two frames and hence splits the reported probability between these two frames. Another aspect to be considered is the fact that the ground truth annotations used as targets for the training are sometimes generated via manual tapping and thus deviate from the real beat position by up to 50 ms. This can result also in blurred peaks in the beat activation function. To reduce the impact of these artefacts, we smooth the activation function before being processed with the filter bank by convolving it with a Hamming window of length 140 ms. [1]

3.3 Comb Filter Periodicity Estimation

We use the output of the neural network stage as input to a bank of resonating comb filters. As outlined previously, comb filters are a common choice to detect periodicities in a signal, e.g. [4, 17, 26]. The advantage of comb filters over autocorrelation lays in the fact that comb filters also resonate at multiples, fractions and simple rationales of the filter lag. This behaviour is in line with the perception of humans, which do not necessarily consider double or half tempi wrong. We use a bank of resonating feed backward comb filters with different time lags (τ), defined as:

$$y(t, \tau) = x(t) + \alpha * y(t - \tau, \tau). \qquad (1)$$

Each comb filter adds a scaled (by factor α) and delayed (with lag τ) version of its own output $y(t)$ to the input signal $x(t)$ with t denoting the time frame index.

[1] Because of this smoothing the beat activations do not reflect probabilities any more (and they may exceed the value of 1), but this does not harm the overall interpretation and usefulness.

3.3.1 *Lag Range Definition*

For the individual bands of the comb filter bank we use a linear spacing of the lags with the minimum and maximum delays calculated as:

$$\begin{aligned} \tau_{min} &= \lfloor 60 * fps/bpm_{max} \rfloor \\ \tau_{max} &= \lceil 60 * fps/bpm_{min} \rceil \end{aligned} \qquad (2)$$

with fps representing the frame rate of the system given in frames per second and the minimum and maximum tempi bpm_{min} and bpm_{max} given in beats per minute. We found the tempo range of $[40, 250]$ bpm to perform best on the validation set.

3.3.2 *Scaling Factor Definition*

Scheirer [26] found it beneficial to use different scaling factors $\alpha(\tau)$ for the individual comb filter bands. He defines them such that the individual filters have the same half-energy time. Klapuri [17] also uses filters with exponentially decaying pulse response, but sets the scaling factor such that the response decays to half after a defined time of 3 seconds.

Contrary to these findings, we use a single value for all filter lags, which is set to $\alpha = 0.79$. The reason that a single value works better for this system may lay in the fact that we sum all peaks of the filters. With a fixed scaling factor, the resonance of filters with smaller lags tend to decay faster, but they also produce more peaks, hence leading to a more "balanced" histogram.

3.3.3 *Histogram Building*

After smoothing the neural network output and processing it with the comb filter, we build a weighted histogram $H(\tau)$ from the output $y(t, \tau)$ by simply summing the activations of the individual comb filters (over all frames) if this filter produced the highest peak at the given time frame:

$$\begin{aligned} H(\tau) &= \sum_{t=0}^{T} y(t,\tau) * I(\tau, \arg\max_{\tau} y(t,\tau)) \\ I(a,b) &= \begin{cases} 1 & \text{if } a \equiv b \\ 0 & \text{otherwise} \end{cases} \end{aligned} \qquad (3)$$

with t denoting the time frame index, T the total number of frames, and τ the filter delays.

The bins of the weighted histogram correspond to the time lags τ and the bin heights represent the number of frames where the corresponding filter has a maximum at this delay, weighted by the activations of the comb filter. This weighting has the advantage that it favours filters which resonate at lags which correspond to intervals with highly probable beat positions (i.e. high values of the beat activation function) over those which are less probable. Figure 2d illustrates the output of the comb filter bank, Figure 2e the weighted maxima which are used to build the weighted histogram shown as the dotted line in Figure 2f.

3.3.4 *Histogram Smoothing*

Music almost always contains tempo fluctuations – at least with regard to the frame rate of the system. Even stable tempi result in weights being split between two or more histogram bins. Therefore we combine bins before reporting the final tempo.

Our approach simply smooths the histogram by convolving it with a Hamming window with a width of seven bins, similar to [25]. Depending on the bin index (corresponding to the filter lag τ), a fixed width results in different tempo deviations, ranging from -7% to $+8\%$ for a lag of $\tau = 24$ (corresponding to 250 bpm) to -2% to $+2.9\%$ for a lag of $\tau = 40$ (i.e. 40 bpm). Although this allows a greater deviation for higher tempi, we found no improvement over choosing the size of the smoothing window as a function of the tempo. Figure 2f shows the smoothed histogram as the solid line.

3.3.5 *Peak Selection*

The histogram shows peaks at the different tempi of the musical piece. Again, previous works put much effort into this stage to select the peak with the strongest perceptual strength, ranging from simple rules driven by heuristics [25] over GMM regression based solutions [24] to utilising a support vector machine (SVM) [10, 25] or decision trees [25]. In order to keep our approach as simple as possible, we simply select the highest peak of the smoothed histogram as our final tempo.

4. EVALUATION

To assess the performance of the proposed system we compare it to an autocorrelation based tempo estimation method as described in [1], which operates on the same beat activation function obtained with the neural network described in Section 3.2. The algorithms of Gkiokas [9], Percival [25], Klapuri [17], Oliveira [23], and Davies [4] were chosen as additional reference systems based on their availability and overall performance.

For a short description of these algorithms, please refer to Section 2.

All of the algorithms were used in their default configuration, except the system of Oliveira [23], which we operated in offline mode with an induction length of 100 seconds, because it yielded significantly better results. [2] It should be noted however, that this mode results in a reduced tempo search range of 81-160 bpm, which can lead to biased results in favour of datasets in this tempo range.

Following [29] and [25] we perform statistical tests of our results compared to the others with McNemar's test using a significance value of $p < 0.01$.

4.1 Evaluation Metrics

Since humans perceive tempo and rhythm subjectively, there is no single best tempo estimate. For example, the perceived tempo can be a multiple or fraction of the tempo given by the score of the piece. This is also known as

[2] This corresponds to: `ibt -off -i auto-regen -t 100`

the tempo octave problem. Therefore, two evaluation measures are used in the literature: *Accuracy 1* considers only the single annotated tempo for the evaluation, whereas *Accuracy 2* also includes integer multiples or fractions of the annotated tempo. Since the data that we use also contains music in ternary meter, we do not only add double and half tempo annotations, but also triple and third tempo. In line with most other publications we report accuracy values which denote the algorithms' ability to correctly estimate the tempo of the musical piece with less than 4% deviation form the annotated ground truth.

4.2 Datasets

We use a total of ten datasets to evaluate the performance of our algorithm. Table 1 lists some statistics of the datasets. Datasets marked with an asterisk (*) were used to train the neural networks with 8-fold cross validation as described in Section 3.2.1.

For all sets with beat annotations available (Ballroom, Hainsworth, SMC, Beatles, RWC, HJDB), we generated the tempo annotations as the median of the inter beat intervals. For the HJDB set (which is in 4/4 meter), we first derived the beat positions from the downbeat annotations before inferring the tempo ground truth. For all other sets we use the provided tempo annotations and – where applicable – the corrected annotations from [25].

Dataset	# files	length	annotations
Ballroom [12, 19] *	685 [3]	5h 57m	beats
Hainsworth [13] *	222	3h 19m	beats
SMC [16] *	217	2h 25m	beats
Klapuri [17]	474	7h 22m	beats
GTZAN [25, 27]	999	8h 20m	tempo
Songs [12]	465	2h 35m	tempo
Beatles [5]	180	8h 9m	beats
ACM Mirum [21, 24]	1410	15h 5m	tempo
RWC Popular [11]	100	6h 47m	beats
HJDB [15]	235	3h 19m	downbeats
total	4987	63h 17m	

Table 1: Overview of the datasets used for evaluation.

4.3 Results & Discussion

Table 2 lists the results of the proposed algorithm compared to the reference systems. The results (of our algorithm) reported on the Ballroom, Hainsworth and SMC set are obtained with 8-fold cross-validation, since these datasets were used to train the neural network. Although this is a technically correct evaluation, it can lead to biased results, since the system knows, e.g. about ballroom music and its features in general and thus has an advantage over the other systems. It is thus no surprise that the proposed system outperforms the others on these sets.

Nonetheless, the new system outperforms the autocorrelation based tempo estimation method operating on the very same neural network output in almost all cases. This clearly shows the advantage of the resonating comb filters, which are less prone to single missing or misaligned peaks in the beat activation function, due to their recurrent nature and the fact that they also resonate on fractions and multiples of the dominant tempo.

The results for the other datasets reflect the algorithm's ability to estimate the tempo of a completely unknown signal without tuning any of the parameters. It can be seen that no single system performs best on all datasets. Our proposed system performs state-of-the-art (i.e. no other algorithm is statistically significantly better) in all but the HJDB set w.r.t. *Accuracy 2*. We even outperform most of the other methods in *Accuracy 1*, which highlights the algorithm's ability to not only capture a meaningful tempo, but also choose the correct tempo octave.

An inspection of incorrectly detected tempi in the HJDB set showed that the algorithm's histogram usually has a peak at the correct tempo but that this peak is not the highest. The reason lays in the fact that this set contains music with breakbeats and strong syncopation. Unfortunately, the neural network often identifies these syncopated notes as beats. Contrary to single or infrequently misaligned beats, the comb filter is not able to correct regularly recurring misalignments. E.g. in drum & bass music, where the bass drum usually falls on the offbeat between the third and fourth beat, this leads to additional peaks in the histogram corresponding to 0.5 and 1.5 times the beat interval, and a much lower peak at the correct position. Since we do not perform intelligent clustering of the histogram peaks, often the rate of the downbeats is reported, which results in a tempo which is not covered by the *Accuracy 2* measure any more.

4.4 MIREX Evaluation

We submitted the algorithm to last year's MIREX evaluation. [4] Performance is tested on a hidden set of 140 files with a total length of 1 hour and 10 minutes. The tempo evaluation used for MIREX is different, because for each song the two most dominant tempi are annotated. MIREX uses the following three evaluation metrics: *P-Score* [22] and the percentage of files for which *at least one* or *both* of the annotated tempi was identified correctly within a maximum allowed deviation of $\pm 8\%$ from the ground truth annotations. Since MIREX requires the algorithms to report two tempi with a relative strength, we adopted the peak-picking strategy outlined in Section 3.3.5 to simply report the two highest peaks.

Table 3 gives an overview of the five best performing algorithms (of different authors) over all years the MIREX tempo estimation task is run, together with results for algorithms also used for evaluation in the previous section.

Our algorithm ranked first in last year's MIREX evaluation and achieved the highest *P-Score* and *at least one tempo reported correctly* performance ever. The best per-

[3] We removed the 13 duplicates identified by Bob Sturm: http://media.aau.dk/null_space_pursuits/2014/01/ballroom-dataset.html

[4] http://nema.lis.illinois.edu/nema_out/mirex2014/results/ate/

	NEW	Böck [1]	Gkiokas [9]	Percival [25]	Klapuri [17]	IBT [23]	Davies [4]
Accuracy 1							
Ballroom [12, 19]	**0.950**†	0.639†−	0.625−	0.653−	0.642−	0.651−	0.709−
Hainsworth [13]	**0.847**†	0.541†−	0.667−	0.721−	0.752−	0.698−	0.739−
SMC [16]	**0.512**†	0.442†	0.346−	0.267−	0.189−	0.166−	0.152−
Klapuri [17]	**0.789**	0.502−	0.741	0.732	0.768	0.724−	0.692−
GTZAN [25]	0.668	0.601−	0.716−	**0.754**+	0.704+	0.599−	0.582−
Songs [12]	0.477	0.570+	0.570+	**0.611**+	0.585+	0.486	0.424
Beatles [5]	**0.850**	0.700−	0.778	0.811	0.789	0.767	0.761−
ACM Mirum [21, 24]	**0.741**	0.540−	0.725	0.733	0.679−	0.621−	0.646−
RWC Popular [11]	0.600	0.450	**0.900**+	0.810+	0.770	0.750	0.770+
HJDB [14]	0.796	0.434−	0.783	0.285−	0.494−	**0.911**+	0.706
Dataset average	**0.721**	0.543	0.563	0.638	0.636	0.637	0.617
Total average	**0.734**	0.560−	0.685−	0.677−	0.658−	0.623−	0.618−
Accuracy 2							
Ballroom [12, 19]	**1.000**†	0.997†	0.981	0.953−	0.921−	0.921−	0.974
Hainsworth [13]	**0.941**†	0.910†	0.887	0.901	0.869	0.802−	0.878
SMC [16]	**0.673**†	0.599†	0.512−	0.438−	0.438−	0.359−	0.415−
Klapuri [17]	0.937	0.907−	**0.954**	0.937	0.918	0.880−	0.924
GTZAN [25]	**0.950**	0.942	0.938	0.925−	0.923−	0.841−	0.922−
Songs [12]	**0.933**	0.918	0.910	0.865−	0.910	0.791−	0.875−
Beatles [5]	0.983	0.967	0.978	**0.989**	0.928	0.883	0.978
ACM Mirum [21, 24]	0.976	0.958−	**0.979**	0.972	0.967	0.915−	0.975
RWC Popular [11]	0.950	0.940	**1.000**	**1.000**	0.990	0.980	**1.000**
HJDB [14]	0.868	0.851	0.911	**1.000**+	0.864	0.991+	**1.000**+
Dataset average	**0.919**	0.899	0.916	0.896	0.871	0.837	0.893
Total average	**0.946**	0.929−	0.935−	0.923−	0.909−	0.861−	0.923−

Table 2: *Accuracy 1* and *Accuracy 2* results for different datasets and algorithms, with best results marked in bold and + and − denoting statistical significance compared to our results. † denote values obtained with 8-fold cross validation.

Algorithm	P-Score	≥1 tempo	both tempi
NEW	**0.876**	**0.993**	0.629
Elowsson [7]	0.857	0.943	**0.693**
Gkiokas [9]	0.829	0.943	0.621
Wu [28]	0.826	0.957	0.550
Lartillot [20]	0.816	0.921	0.571
Klapuri [17]	0.806	0.943	0.614
Böck [1]	0.798	0.957	0.564
Davies [4]	0.776	0.929	0.457

Table 3: Results on the McKinney test collection used for the MIREX evaluation.

forming algorithm for the *both tempi correct* evaluation was the one submitted by Elowsson [7] in 2013, which explicitly models the speed of the music and thus has a much higher chance to report the two annotated tempi which are inferred from human beat tapping.

5. CONCLUSION

The presented tempo estimation algorithm based on recurrent neural networks and resonating comb filters is able to perform state-of-the-art or outperforms existing algorithms on all but one datasets investigated. Based on the high *Accuracy 2* score, which also considers integer multiples and fractions of the annotated ground truth tempo, it can be concluded that the system is able to capture a meaningful tempo in almost all cases.

Additionally, we outperform many existing algorithms w.r.t. *Accuracy 1* which suggests that it is advantageous to use a musically more meaningful representation than just the onset strength of the signal – even if split into multiple accent bands – as an input for a bank of resonating comb filters.

In future, we want to investigate methods of perceptually clustering the peaks of the histogram to report the most relevant tempo, as this has been identified to be the main problem of the new algorithm when dealing with very syncopated music. We believe that this should increase the *Accuracy 1* performance considerably.

The source code and additional resources can be found at: http://www.cp.jku.at/people/boeck/ISMIR2015.html.

6. 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. We would like to thank the authors of the other algorithms for sharing their code or making it publicly available.

7. REFERENCES

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

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

[3] A. T. Cemgil, B. 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] 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, 2007.

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

[7] A. Elowsson, A. Friberg, G. Madison, and J. Paulin. Modelling the speed of music using features from harmonic/percussive separated audio. In *Proc. of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, Curitiba, Brazil, 2013.

[8] M. Gainza and E. Coyle. Tempo detection using a hybrid multiband approach. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(1):57–68, 2011.

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

[10] A. Gkiokas, V. Katsouros, and G. Carayannis. Reducing Tempo Octave Errors by Periodicity Vector Coding And SVM Learning. In *Proc. of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 301–306, Porto, Portugal, 2012.

[11] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC Music Database: Popular, Classical, and Jazz Music Databases. In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR 2002)*, pages 287–288, Paris, France, 2002.

[12] 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.

[13] S. Hainsworth and M. Macleod. Particle filtering applied to musical tempo tracking. *EURASIP Journal on Applied Signal Processing*, 15:2385–2395, 2004.

[14] J. Hockman and I. Fujinaga. Fast vs slow: Learning tempo octaves from user data. In *Proc. of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 231–236, Utrecht, Netherlands, 2010.

[15] J. Hockman, M. E. Davies, and I. Fujinaga. One in the jungle: Downbeat detection in hardcore, jungle, and drum and bass. In *Proc. of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 169–174, Porto, Portugal, 2012.

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

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

[18] F. Korzeniowski, S. Böck, and G. Widmer. Probabilistic extraction of beat positions from a beat activation function. In *Proc. of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pages 513–518, Taipei, Taiwan, 2014.

[19] F. Krebs, S. Böck, and G. Widmer. Rhythmic pattern modeling for beat and downbeat tracking in musical audio. In *Proc. of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, pages 227–232, Curitiba, Brazil, 2013.

[20] O. Lartillot, D. Cereghetti, K. Eliard, W. J. Trost, M.-A. Rappaz, and D. Grandjean. Estimating tempo and metrical features by tracking the whole metrical hierarchy. In *Proc. of the 3rd International Conference on Music & Emotion (ICME)*, Jyväskylä, Finland, 2013.

[21] M. Levy. Improving perceptual tempo estimation with crowd-sourced annotations. In *Proc. of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 317–322, Miami, USA, 2011.

[22] M. F. McKinney, D. Moelants, M. E. P. Davies, and A. Klapuri. Evaluation of Audio Beat Tracking and Music Tempo Extraction Algorithms. *Journal of New Music Research*, 36(1):1–16, 2007.

[23] J. Oliveira, F. Gouyon, L. G. Martins, and L. P. Reis. IBT: a real-time tempo and beat tracking system. In *Proc. of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, Netherlands, 2010.

[24] G. Peeters and J. Flocon-Cholet. Perceptual tempo estimation using GMM-regression. In *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies*, pages 45–50, 2012.

[25] G. Percival and G. Tzanetakis. Streamlined tempo estimation based on autocorrelation and cross-correlation with pulses. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12):1765–1776, 2014.

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

[27] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[28] F.-H. F. Wu and J.-S. R. Jang. A supervised learning method for tempo estimation of musical audio. In *22nd Mediterranean Conference of Control and Automation (MED)*, pages 599–604, Palermo, Italy, 2014.

[29] J. Zapata and E. Gómez. Comparative evaluation and combination of audio tempo estimation approaches. In A. E. Society, editor, *AES 42nd Conference on Semantic Audio*, Ilmenau, Germany, 2011. Audio Engineering Society, Audio Engineering Society.