

# REAL-TIME VISUAL BEAT TRACKING USING A COMB FILTER MATRIX

*Andrew Robertson, Adam M. Stark, Mark D. Plumbley*

Centre For Digital Music  
School of Electronic Engineering and Computer Science  
Queen Mary University of London, England  
andrew.robertson@eecs.qmul.ac.uk

## ABSTRACT

This paper describes an algorithm for real-time beat tracking with a visual interface. Multiple tempo and phase hypotheses are represented by a comb filter matrix. The user can interact by specifying the tempo and phase to be tracked by the algorithm, which will seek to find a continuous path through the space. We present results from evaluating the algorithm on the Hainsworth database and offer a comparison with another existing real-time beat tracking algorithm and offline algorithms.

## 1. INTRODUCTION

In this paper, we are concerned with the problem of beat tracking and its real-time visualisation that might be used by an operator or musician when integrating algorithmic-based effects and processes in live performance. If a piece of music can be characterised as consisting of musical events which are often ‘on the beat’, then we may expect that signal processing methods such as comb filtering and autocorrelation might succeed in locating the beat. The comb filter has been employed for the purposes of beat tracking since Scheirer [10]. There the signal was pre-processed through six band-pass filters to generate envelopes which were used to filter white noise and provide a representation of the music as input to a bank of comb filter resonators. The strongest comb filter output would then be analysed for phase locked behaviour, which thereby provided the beat location. Klapuri et al. [9] extended Scheirer’s approach by making use of probabilistic techniques to track at different metrical levels: the level of the tatum, defined by Bilmes [2] as the lowest metrical level present in a piece, the beat level and the bar level.

Eck [6] has presented work on the Autocorrelation Phase Matrix, APM, in which multiple tempo and phase hypotheses that might be adopted by a listener are represented visually in a matrix structure. Whereas in many beat trackers the estimation of phase is often only undertaken after a tempo induction process, in the APM phase information is preserved and summing each row provides the conventional unbiased autocorrelation function. Whilst autocorrelation is useful for detecting periodicities in signals, it has some properties which may make it not entirely

suitable for beat analysis of musical signals. Autocorrelation measures the cross-correlation of the signal with itself over a range of lags  $k$  and is defined as

$$y(t) = x(t)x(t-k). \quad (1)$$

Thus periodic peaks in the signal will have high autocorrelation values at the appropriate lag. For some musical rhythms, namely those where peaks in the onset detection function occur regularly on the beat, this will work well and often this is the case, such as in the real-time Max/MSP object `Btrack~` by Stark et al. [11] based on the offline model of Davies and Plumbley [4]. However, in principle, autocorrelation requires regular pulses of energy at the beat interval. In cases of syncopation whereby one event occurs at a usually unstressed metrical level, then the correlation will be very low at the beat period for both the syncopated beat and the subsequent beat and alternative lag times corresponding to intervals between the regular beat and the syncopated accent will have strong autocorrelation values.

## 2. ALGORITHM DESCRIPTION

The comb filter is very similar to autocorrelation, but sums energy at a given lag and there is no requirement that there is correlation between consecutive lags. For input  $x_t$  at discrete time-step  $t$ , a comb filter of period  $T$  has its output is determined by the recursive equation

$$y_t = \alpha y_{t-T} + (1 - \alpha)x_t. \quad (2)$$

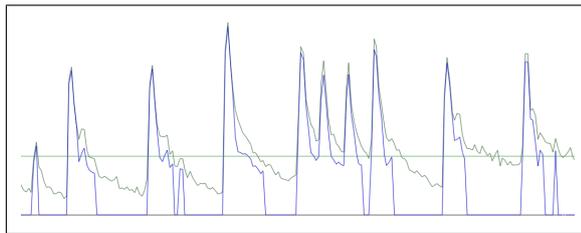
Thus a pulse train of energy at a repetitive period  $\tau$  causes reinforcement or resonance at that period. There is also a lesser degree of reinforcement at whole fraction multiples of the period. If our signal has higher energy at regular intervals which correspond to our perception of beat, then we can expect the output of the comb filter to be highest at this period. At the beat period, a syncopated rhythm will contribute energy at alternative phases which is potentially problematic, but this no longer affects the contribution of a regular beat after the syncopated event.

We would like to provide a representation of possible tempos to the user in a comprehensible manner. Few beat tracking algorithms offer a real-time visualisation of their

behaviour and tend to be presented as ‘black box’ objects which hide the operational processes and are fully automatic. By visualising aspects of the algorithmic process, we aim to offer more control back to the user, which might also allow their instinctive beat tracking abilities to aid the tempo tracking process.

Rather than just calculating the output of comb filter resonators across a range of tempos, we wish to compute the values over the range of available tempo and phase hypotheses. Following Eck [6], we make use of a matrix structure which retains phase information and can be used to represent multiple tempo and phase hypotheses in a single image. This visualisation is informative about the nature of how the comb filter responds to musical signals and can be used for performance purposes by having the user select the appropriate tempo and phase at which to track. Whilst we provide the results of automatic analysis across a database, the motivation for this work is to generate a real-time algorithm and visualisation whose behaviour can be adapted.

## 2.1. Pre-Processing



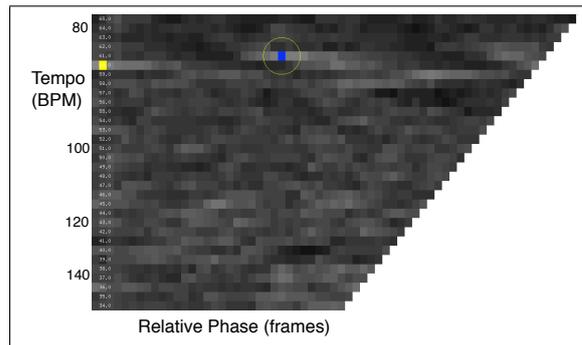
**Figure 1.** Onset detection function obtained processing a recording of drum parts by Led Zeppelin’s John Bonham. The median is the horizontal line through the middle of the onsets. Both the raw detection function and the output of pre-processing as described in section 2.1 are shown.

Approaches to beat tracking tend to be either event based, taking a list on onset times as input, or else they process an onset detection function,  $x_t$ , that acts as a driving function. Bello et al. [1] discuss the range of algorithms that are employed in onset detection functions, which provide a representation of the strength of musical events or onsets at each time step; these include energy, high frequency content, spectral difference, phase-based difference and complex domain onset detection. We made use of Brossier et al.’s *aubio* library to use the onset detection functions as described in Bello et al. [1]. We made use of the complex spectral difference function and processed the resulting signal by setting

$$x(t) = \begin{cases} d(t) & \text{if } d(t) > \text{median} \\ 0 & \text{otherwise} \end{cases},$$

where  $d(t)$  is the onset detection function at time  $t$ . The hopsize was set to 512 samples corresponding to values being output every 11.6ms. The effect of this pre-processing can be seen in Figure 1.

## 2.2. Comb Filter Matrix



**Figure 2.** Matrix of comb filter values across many tempo and phase hypotheses. The grid is with respect to detection function frames and the corresponding BPM is shown on the vertical axis. The peak value is a highlighted square and the currently tracked hypothesis is ringed.

When constructing the comb filter matrix, we require the phase of the current time-frame at each lag  $\tau$ . We denote the value in the cell corresponding to a tempo of period  $\tau$  and phase  $\theta$  by  $X(\tau, \theta)$ . The phase of cells that are vertically aligned with each other do not necessarily correspond to the same frames in onset detection function due to the differences in their period. Input  $x_t$  at discrete time-step  $t$  will contribute to the period  $\tau$  at a specific phase  $\theta_{\tau,t}$ . The comb filter matrix has output determined by the update equation

$$X(\tau, \theta_{\tau,t}) \leftarrow \alpha x_t + (1 - \alpha)X(\tau, \theta_{\tau,t}). \quad (3)$$

The resolution of the comb filter matrix (CFM) is dictated by the hopsize of the onset detection function. In this case, our chosen resolution corresponds to a difference of between 1.3 and 4 BPM between consecutive tempo bins over the range 160 to 80 BPM. The resulting comb filter outputs across the range of tempo and phase hypotheses can be seen in Figure 2 with the parameter  $\alpha$  set to 0.1. Since there will be noise due to detection function resolution, tempo and phase shifts and expressive timing, we modify this comb filter update equation so that high detection function values also contribute to the hypotheses in surrounding bins but to a lesser degree. The current detection function sample corresponds to a particular phase offset, denoted  $\theta_{\tau,t}$  at each period  $\tau$ . Then

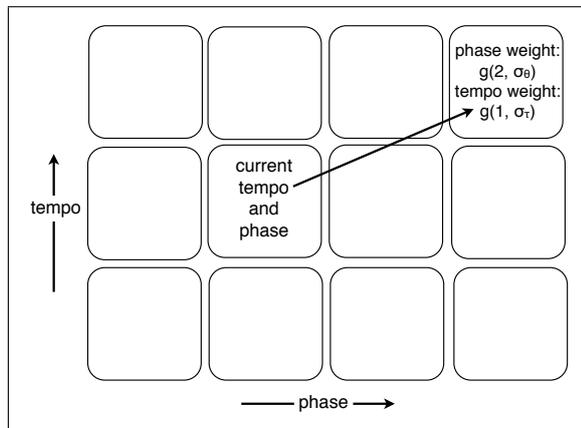
$$X(\tau, \theta_{\tau,t}) \leftarrow \alpha x(t) + (1 - \alpha) \max\{c_\tau(\tau_m)c_\theta(\theta_n)X(\tau_m, \theta_n)\} \quad (4)$$

where  $c_\tau(\cdot)$  and  $c_\theta(\cdot)$  are weighting functions for transition across tempo and phase hypotheses. These are defined using the Gaussian-shaped function,  $g$ , such that

$$g(t, m, \sigma) = e^{-\frac{(t-m)^2}{2\sigma^2}}. \quad (5)$$

Then the weighting functions are defined as  $c_\tau(\tau_m) = g(\tau, \tau_m, \sigma_\tau)$ , where  $\tau_m$  is the period of an adjacent tempo, and  $c_\theta(\theta_n) = g(\theta, \theta_n, \sigma_\theta)$ , where  $\theta_n$  is an

alternative phase offset. The use of these weighting functions is illustrated in Figure 3. We choose the values 3.5 and 6 for  $\sigma_\tau$  and  $\sigma_\theta$ , the standard deviation of the weighting functions for tempo and phase respectively.



**Figure 3.** Weighting function for transitions across tempo and phase as described in Equation 10.

### 2.3. Entropy and Rayleigh weighting

In order to favour those tempos which are preferred by humans, we follow Davies and Plumbley [4] in making use of a Rayleigh weighting centred at 120 BPM. The Rayleigh function is defined as

$$r(\tau) = \frac{\tau}{\beta^2} e^{-\frac{\tau^2}{2\beta^2}} \quad (6)$$

where  $\tau$  is the comb filter delay in DF frames and  $\beta$  is a parameter that sets the strongest point of attenuation. Here, we choose  $\beta$  to be 43 frames, corresponding to a tempo of 120 BPM. We expect a dense concentration of higher matrix values, peaking about the correct tempo and phase hypothesis. This would result in higher Shannon entropy than a tempo at which the matrix values are more consistent across all phase hypotheses. The entropy is defined as

$$h(\tau) = \sum_{\theta=0}^{\tau-1} -p_X(\tau, \theta) \log p_X(\tau, \theta), \quad (7)$$

where

$$p_X(\tau, \theta) = \frac{X(\tau, \theta)}{\sum_{\theta=0}^{\tau-1} X(\tau, \theta)}. \quad (8)$$

and  $p_X(\tau, \theta)$  can be interpreted as a probability distribution of energy at period  $\tau$ . We then look to track tempo and phase across the resulting matrix

$$Y(\tau, \theta) = \frac{X(\tau, \theta)r(\tau)}{h(\tau)} \quad (9)$$

Then the maximum value of  $Y$  indicates the most likely tempo and phase hypothesis for beat tracking. However, in order for the beat tracker to be stable during syncopated rhythms and other less straightforward musical passages, we wish to follow our current hypothesis rather

than simply pick the maximum value in the matrix at every timestep.

### 2.4. Tracking a tempo hypothesis

Supposing that we have a current tempo hypothesis,  $(\tau_f, \theta_f)$  for the beat and we then wish to follow this hypothesis through subtle changes in tempo and phase. In order to do so, we require a balance between the flexibility to change to nearby tempo and phase hypotheses and stability at the current hypothesis. This balance is maintained by restricting transitions from the currently followed tempo and using a weighting function to favour the current hypothesis. The weighting of a transition from current hypothesis  $\tau_f$  and  $\theta_f$  to new hypothesis  $\tau_{new}$  and  $\theta_{new}$  is given by the product of two Gaussian-shaped functions

$$w(\tau_{new}, \theta_{new}) = g(|\tau_f - \tau_{new}|, \sigma_\tau)g(|\theta_f - \theta_{new}|, \sigma_\theta), \quad (10)$$

Through testing, we have chosen the standard deviations of the Gaussians,  $\sigma_\tau$  and  $\sigma_\theta$ , to be 4 and 10 respectively. Given a currently tracked hypothesis at tempo  $\tau_f$  and phase  $\theta_f$ , we employ the rule that if  $w(\tau_{new}, \theta_{new})Y(\tau_{new}, \theta_{new}) > Y(\tau_f, \theta_f)$  then we set  $\tau_f$  and  $\theta_f$  to  $\tau_{new}$  and  $\theta_{new}$ .

### 2.5. Visualisation

The visualisation system provides feedback to the user which helps them when selecting between hypotheses for tracking. The output of the comb filter matrix is represented as a grid of squares where the intensity of each square corresponds to the value in the matrix at the respective tempo and phase. Bright spots on the grid represent likely locations of an appropriate tapping rate. We represent beats through squares flashing for the currently followed hypothesis (the output of the beat tracker) and the peak value across all tempo and phase possibilities.

## 3. EVALUATION

We tested the three algorithms across the Hainsworth database [8] of 222 musical pieces. The database consists of the following numbers of excerpts of each genre: dance (40), rock (68), folk (22), jazz (40), classical (30) and choral (22). The results are generally poor for all algorithms on choral and classical music where harmonic information plays a more crucial role in locating the musical pulse and there are wide variations in tempo and phrasing. When presenting the results of evaluation, we also provide figures on the subsets of dance, rock and jazz, which are the more rhythmic types of music which we expect our beat tracking method to be suited to. We compared our comb filter matrix algorithm to another real-time algorithm by Stark et al.[11] and an offline implementation of the same algorithm by Davies and Plumbley. We also compared against offline algorithms by Ellis [7], Klapuri [9] and the BeatRoot algorithm by Simon Dixon [5]. These represent the state of the art in Music Information Retrieval (MIR), although they are not available as

real-time implementations and thus cannot be used in live performance. Initialisation of the beat tracker was to the peak value once the matrix values reflected regularities in the signal.

The results are given for a new measure that combines the total number of correct beats at allowed metrical levels, as in Davies and Plumbley [4], with a measure for localisation to the annotated beat as introduced by Cemgil et al. [3]. This makes use of a Gaussian window with a standard deviation of 40 msec around the annotated beat location to assign a score to each beat. The Allowed Metrical Level considers beat output at twice and half the annotated tempo as permissible and by using the sum of continuous segments, there is no requirement for continuity. Tests indicate that the highest scores achieved using this measure are around 90%. The results for selected genres

	Overall	Dance	Rock	Jazz
Comb Filter Matrix	<b>60.5</b>	<b>77.5</b>	<b>70.3</b>	<b>64.0</b>
Stark et al.: Btrack~	63.7	82.0	74.9	70.2
Davies and Plumbley	66.6	86.2	78.0	74.4
Ellis	61.3	64.1	69.1	56.4
Klapuri et al.	67.2	80.6	79.4	65.9
BeatRoot	63.0	84.1	74.2	69.6

**Table 1.** Results (as a percentage) obtained by the beat tracking models across the complete Hainsworth database and restricted to the genres of rock, dance and jazz.

are displayed in Table 1. The score for beats at the allowed metrical level is 78% for dance and 70% for rock genres. The algorithm’s performance is close but slightly short of Stark et al.’s Btrack~. Whilst some of the other algorithms from the MIR field also have higher results (e.g. Klapuri et al. and Davies Plumbley’s algorithm), the difference is not as much as we might expect for a real-time algorithm competing offline over a database. One important point to consider is that this method of evaluation does not take into account the interactive functionality of our algorithm whereby the user can select the tempo and phase to be tracked. Our intention here is the provision of a beat tracking algorithm that may be used to synchronise lighting, video or generative audio, which provides functionality to the user and a stability of output which is valuable for aesthetic reasons.

#### 4. CONCLUSION AND FURTHER WORK

We introduce the comb filter matrix for the visual representation of a real-time beat tracker and describe an algorithm for the consistent tracking of any chosen tempo and phase hypothesis. This provides an intuitive interface to see the relative strength of multiple tempo and phase hypotheses which might prove valuable for musicians wishing to incorporate such algorithms into performance. In the future, we aim to investigate creative uses of the algorithm and whether such practices can be criti-

cally evaluated within the context of musical interaction. The algorithm is available for download as an external for MaxMSP and in openFrameworks <sup>1</sup>.

#### 5. ACKNOWLEDGEMENTS

Thanks to Stephen Hainsworth for making available his audio database and to Anssi Klapuri his beat tracking model.

#### 6. REFERENCES

- [1] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. Sandler, “A tutorial on onset detection in music signals,” *IEEE Transactions on Speech and Audio In Processing*, vol. 13, no. 5, Part 2, pp. 1035–1047, 2005.
- [2] J. Bilmes, “A model for musical rhythm,” in *Proc. ICMC*, 1992, pp. 207–210.
- [3] A. T. Cemgil, H. J. Kappen, P. Desain, and H. Honing., “On tempo tracking: Tempogram Representation and Kalman filtering,” *Journal of New Music Research*, vol. 28, no. 4, pp. 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*, vol. 15, no. 3, pp. 1009–1020, 2007.
- [5] S. Dixon, “Evaluation of the audio beat tracking system beatroot,” *Journal of New Music Research*, vol. 36, no. 1, pp. 39–50, 2007.
- [6] D. Eck, “Beat tracking using an autocorrelation phase matrix,” in *Proc. ICASSP*, 2007, pp. 1313–1316.
- [7] D. P. W. Ellis, “Beat tracking by dynamic programming,” *Journal of New Musical Research*, vol. 36, no. 1, pp. 51–60, 2007.
- [8] S. Hainsworth, “Techniques for the automated analysis of musical audio,” Ph.D. dissertation, Cambridge University Engineering Department, 2003.
- [9] 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*, vol. 14, no. 1, pp. 342–355, 2006.
- [10] E. D. Scheirer, “Tempo and beat analysis of acoustic musical signals,” *The Journal of the Acoustical Society of America*, vol. 103, no. 1, pp. 588–60, 1998.
- [11] A. M. Stark, M. E. P. Davies, and M. D. Plumbley, “Real-time beat-synchronous analysis of musical audio,” in *Proc. DAFX-09*, 2009, pp. 299–304.

<sup>1</sup><http://www.eecs.qmul.ac.uk/~andrewr>