MIREX 2006 Audio Beat Tracking Evaluation: BeatRoot

Simon Dixon
Austrian Research Institute for Artificial Intelligence
Freyung 6/6, Vienna 1010, Austria
simon.dixon@ofai.at

Abstract

BeatRoot is an interactive beat tracking system which has been used for several years in studies of performance timing. In this new version, some of the weaknesses of the original system have been addressed. The original simple onset detection algorithm, which caused problems for beat tracking music without prominent drums, has been replaced with a more robust onset detector. Several new features have been added, such as annotation of multiple metrical levels and phrase boundaries, and improvements in the user interface. Also, the new version has been written entirely in Java, so that it runs on all major platforms. The beat tracking algorithm remains largely unchanged: BeatRoot uses a multiple agent architecture which simultaneously considers several different hypotheses concerning the rate and placement of musical beats, resulting in accurate tracking of the beat, quick recovery from errors, and graceful degradation in cases where the beat is only weakly implied by the data.

Keywords: MIREX, tempo induction, beat tracking.

1. Introduction

Compared with complex cognitive tasks such as playing chess, beat tracking (identifying the basic rhythmic pulse of a piece of music) does not appear to be particularly difficult, as it is performed by people with little or no musical training, who tap their feet, clap their hands or dance in time with music. However, while chess programs compete with world champions, no computer program has been developed which approaches the beat tracking ability of a good musician.

As a fundamental part of music cognition, beat tracking has practical uses in performance analysis, perceptual modelling, audio content analysis (such as for music transcription and music information retrieval), and the synchronisation of musical performance with computers or other devices. The previous version of BeatRoot [1, 2] was used in a large scale study of interpretation in piano performance [3, 4] to create symbolic metadata from audio CDs for automatic analysis of performance timing.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.
© 2006 University of Victoria

In this paper we describe the new version of BeatRoot, a system which models the perception of beat by two interacting processes: the first finds the rate of the beats (*tempo induction*), and the second synchronises a pulse sequence with the music (*beat tracking*). A clustering algorithm finds the most significant metrical units, and the clusters are then compared to find reinforcing groups, and a ranked set of tempo hypotheses is computed. Based on these hypotheses, a multiple agent architecture is employed to match sequences of beats to the music, where each agent represents a specific tempo and alignment of beats with the music. The agents are evaluated on the basis of the regularity, continuity and salience of the onsets corresponding to hypothesised beats, and the highest ranked beat sequence is returned as the solution. The user interface presents a graphical representation of the music and the extracted beats, and allows the user to edit and recalculate results based on the editing. More complete descriptions of the algorithms can be found in [1, 5].

2. BeatRoot Architecture

BeatRoot takes digital audio as input, and processes the data off-line to detect salient rhythmic events. The timing of these events is then analysed to generate hypotheses of the tempo at various metrical levels. The stages of processing are shown in Figure 1, and will be described in the following subsections.

2.1. Onset Detection

Initial processing of the audio signal is concerned with finding the onsets of musical notes, which are the primary carriers of rhythmic information. Earlier versions of BeatRoot used a time-domain onset detection algorithm, which finds local peaks in the slope of a smoothed amplitude envelope. This method is particularly well suited to music with drums, but less reliable at finding onsets of other instruments in a polyphonic setting. In the current version it has been replaced with an onset detector which finds peaks in the spectral flux. This method is described fully in [5].

Spectral flux sums the change in magnitude in each frequency bin where the change is positive, that is, the energy is increasing. First, a time-frequency representation of the signal based on a short time Fourier transform using a Hamming window $w(m)$ is calculated at a frame rate of 100 Hz. If $X(n, k)$ represents the kth frequency bin of the nth frame,

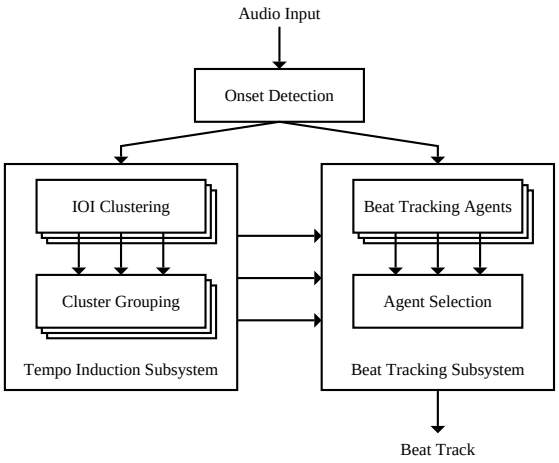

Figure 1. System architecture of BeatRoot

then:

$$X(n,k) = \sum_{m=-\frac{N}{2}}^{\frac{N}{2}-1} x(hn+m)\, w(m)\, e^{-\frac{2j\pi mk}{N}}$$

where the window size $N = 2048$ (46 ms at a sampling rate of $r = 44100$ Hz) and hop size $h = 441$ (10 ms, or 78.5% overlap). The spectral flux function SF is then given by:

$$SF(n) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} H(|X(n,k)| - |X(n-1,k)|)$$

where $H(x) = \frac{x+|x|}{2}$ is the half-wave rectifier function. Empirical tests favoured the use of the L_1-norm here over the L_2-norm used in [6, 7], and the linear magnitude over the logarithmic (relative or normalised) function proposed by Klapuri [8].

2.2. Tempo Induction

The tempo induction algorithm uses the calculated onset times to compute clusters of inter-onset intervals (IOIs). An IOI is defined to be the time interval between any pair of onsets, not necessarily successive. In most types of music, IOIs corresponding to the beat and simple integer multiples and fractions of the beat are most common. Due to fluctuations in timing and tempo, this correspondence is not precise, but by using a clustering algorithm, it is possible to find groups of similar IOIs which represent the various musical units (e.g. half notes, quarter notes).

This first stage of the tempo induction algorithm is represented in Figure 2, which shows the events along a time line (above), and the various IOIs (below), labelled with their corresponding cluster names (C1, C2, etc.). The next stage is to combine the information about the clusters, by recognising approximate integer relationships between clusters.

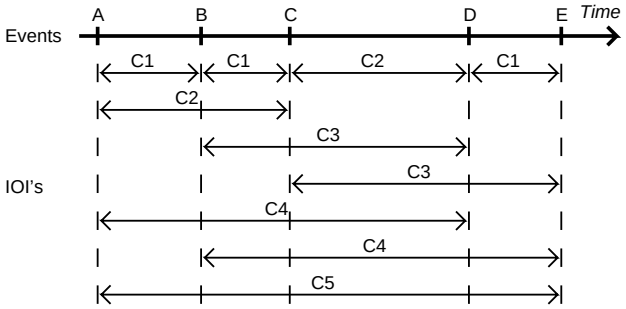

Figure 2. Clustering of inter-onset intervals: each interval between any pair of events is assigned to a cluster (C1, C2, C3, C4 or C5)

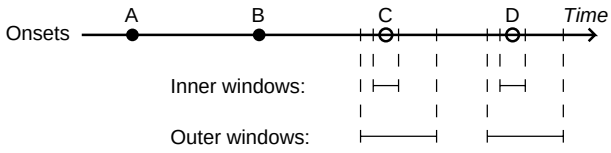

Figure 3. Tolerance windows of a beat tracking agent predicting beats around C and D after choosing beats at onsets A and B

For example, in Figure 2, cluster C2 is twice the duration of C1, and C4 is twice the duration of C2. This information, along with the number of IOIs in each cluster, is used to weight the clusters, and a ranked list of tempo hypotheses is produced and passed to the beat tracking subsystem.

2.3. Beat Tracking

The most complex part of BeatRoot is the beat tracking subsystem, which uses a multiple agent architecture to find sequences of events which match the various tempo hypotheses, and rates each sequence to determine the most likely sequence of beat times. The music is processed sequentially from beginning to end, and at any particular point, the agents represent the various hypotheses about the rate and the timing of the beats up to that time, and make predictions of the next beats based on their current state.

Each agent is initialised with a tempo (rate) hypothesis from the tempo induction subsystem and an onset time, taken from the first few onsets, which defines the agent's first beat time (phase). The agent then predicts further beats spaced according to the given tempo and first beat, using tolerance windows to allow for deviations from perfectly metrical time (see Figure 3). Onsets which correspond with the inner window of predicted beat times are taken as actual beat times, and are stored by the agent and used to update its rate and phase. Onsets falling in the outer window are taken to be possible beat times, but the possibility that the onset is not on the beat is also considered.

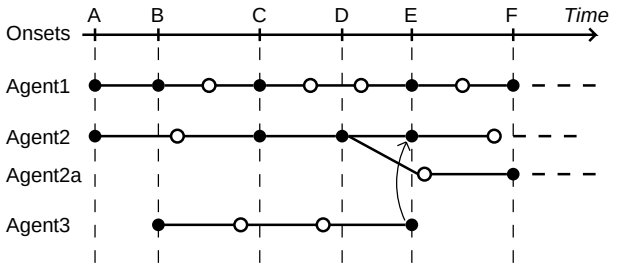

Figure 4. Beat tracking by multiple agents (see text for explanation)

Figure 4 illustrates the operation of beat tracking agents. A time line with 6 onsets (A to F) is shown, and below the time line are horizontal lines marked with solid and hollow circles, representing the behaviour of each agent. The solid circles represent predicted beat times which correspond to onsets, and the hollow circles represent predicted beat times which do not correspond to onsets. The circles of Agent1 are more closely spaced, representing a faster tempo than that of the other agents.

Agent1 is initialised with onset A as its first beat. It then predicts a beat according to its initial tempo hypothesis from the tempo induction stage, and onset B is within the inner window of this prediction, so it is taken to be on the beat. Agent1's next prediction lies between onsets, so a further prediction, spaced two beats from the last matching onset, is made. This matches onset C, so the agent marks C as a beat time and interpolates the missing beat between B and C. Then the agent continues, matching further predictions to onsets E and F, and interpolating missing beats as necessary.

Agent2 illustrates the case when an onset matches only the outer prediction window, in this case at onset E. Because there are two possibilities, a new agent (Agent2a) is created to cater for the possibility that E is not a beat, while Agent2 assumes that E corresponds to a beat.

A special case is shown by Agent2 and Agent3 at onset E, when it is found that two agents agree on the time and rate of the beat. Rather than allowing the agents to duplicate each others' work for the remainder of the piece, one of the agents is terminated. The choice of agent to terminate is based on the evaluation function described in the following paragraph. In this case, Agent3 is terminated, as indicated by the arrow. A further special case (not illustrated) is that an agent can be terminated if it finds no events corresponding to its beat predictions (it has lost track of the beat).

Each agent is equipped with an evaluation function which rates how well the predicted and actual beat times correspond. The rating is based on how evenly the beat times are spaced, how many predicted beats correspond to actual events, and the salience of the matched events, which is calculated from the spectral flux at the time of the onset. At the end of processing, the agent with the highest score outputs its sequence of beats as the solution to the beat tracking problem.

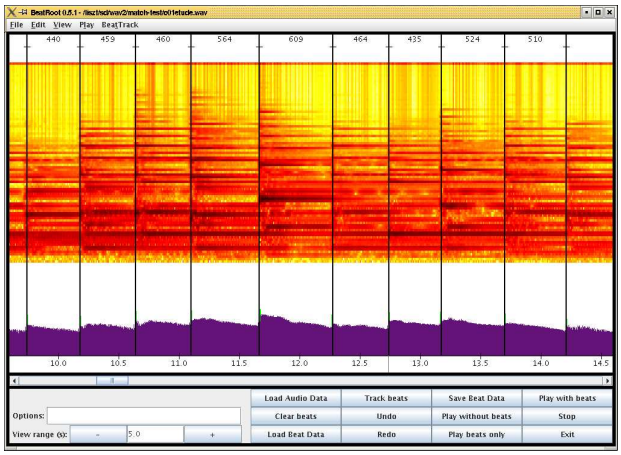

Figure 5. Screen shot of BeatRoot showing a 5-second excerpt from a Chopin piano Etude (Op.10, No.3), with the inter-beat intervals in ms (top), calculated beat times (long vertical lines), spectrogram (centre), amplitude envelope (below) marked with detected onsets (short vertical lines) and the control panel (bottom)

2.4. Implementation

The system described above has been implemented with a graphical user interface which allows playback of the music with the beat times marked by clicks, and provides a graphical display of the signal and the beats with editing functions for correction of errors or selection of alternate metrical levels. The audio data is displayed as a waveform and spectrogram, and the beats are shown as vertical lines on the display (Figure 5).

BeatRoot is written in Java and is available from:
`http://www.ofai.at/~simon.dixon/beatroot`

3. Results

3.1. Testing

BeatRoot was tested on a range of different musical styles, including classical, jazz, and popular works with a variety of tempi and meters. The following results were obtained with the previous version of BeatRoot, using test data consisting of a set of 13 complete piano sonatas, a large collection of solo piano performances of two Beatles songs and a small set of popular, jazz and latin songs. In each case, the system found an average of over 90% of the beats [1], and compared favourably to another (then) state-of-the-art tempo tracker [9]. Tempo induction was in most cases correct, with the most common error being the choice of a musically related metrical level such as double or half the subjectively chosen primary rate. The calculation of beat times is also quite robust; when the system loses synchronisation

Contestant	P-Score (average)	Run-time
Dixon	0.407	639
Ellis	0.401	498
Klapuri	0.395	1218
Davies	0.394	1394
Brossier	0.391	139

Table 1. Results of the MIREX 2006 Audio Beat Tracking Evaluation

with the beat, it usually recovers quickly to resume correct beat tracking, despite the fact that the system has no high level knowledge of music to guide it. Some audio examples are available at:
`http://www.ofai.at/~simon.dixon`

3.2. MIREX 2006 Results

BeatRoot performed best of the 5 systems submitted for the MIREX 2006 Audio Beat Tracking Evaluation, as shown in Table 1. The test data consisted of 140 files from a wide range of musical styles, which had been annotated by around 40 people per file by tapping in time with the music. Although this is a slightly different task than off-line beat tracking (see [10] for a discussion), it is a reasonable approach for this evaluation, especially considering the difficulty of creating or obtaining ground-truth data.

3.3. Discussion

Since the results have been summarised as a single score, we do not know if the difference in performance between systems is significant, nor whether the systems' choice of metrical levels was a deciding factor in these results. BeatRoot is not programmed to select the metrical level corresponding to the perceived beat, nor to a typical tapping rate; it tends to prefer faster rates, because they turn out to be easier to track, in the sense that the agents achieve higher scores. More detailed results and analysis would be very interesting. An interesting task for future years would be to test beat tracking performance for a given metrical level (e.g. given the first two beats or the initial tempo). It would also be interesting to know the P-scores of the annotators (tappers), measured on the basis of the other tappers' data, to see how close this year's entries are to human beat tracking ability.

4. Acknowledgements

This work was supported by the Vienna Science and Technology Fund, project CI010 *Interfaces to Music*, and the EU project S2S^2. OFAI acknowledges the support of the ministries BMBWK and BMVIT. Thanks to the MIREX team for conducting the MIREX evaluation.

References

[1] S. Dixon, "Automatic extraction of tempo and beat from expressive performances," *Journal of New Music Research*, vol. 30, no. 1, pp. 39–58, 2001.

[2] S. Dixon, "An interactive beat tracking and visualisation system," in *Proceedings of the International Computer Music Conference*, 2001, pp. 215–218.

[3] G. Widmer, "Machine discoveries: A few simple, robust local expression principles," *Journal of New Music Research*, vol. 31, no. 1, pp. 37–50, 2002.

[4] G. Widmer, S. Dixon, W. Goebl, E. Pampalk, and A. Tobudic, "In search of the Horowitz factor," *AI Magazine*, vol. 24, no. 3, pp. 111–130, 2003.

[5] S. Dixon, "Onset detection revisited," in *Proceedings of the 9th International Conference on Digital Audio Effects*, 2006, pp. 133–137.

[6] C. Duxbury, M. Sandler, and M. Davies, "A hybrid approach to musical note onset detection," in *Proceedings of the 5th International Conference on Digital Audio Effects*, 2002, pp. 33–38.

[7] J.P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. Sandler, "A tutorial on onset detection in musical signals," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 1035–1047, 2005.

[8] A. Klapuri, "Sound onset detection by applying psychoacoustic knowledge," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Phoenix, Arizona, 1999.

[9] S. Dixon, "An empirical comparison of tempo trackers," in *Proceedings of the 8th Brazilian Symposium on Computer Music*. 2001, pp. 832–840, Brazilian Computing Society.

[10] S. Dixon, W. Goebl, and E. Cambouropoulos, "Perceptual smoothness of tempo in expressively performed music," *Music Perception*, vol. 23, no. 3, pp. 195–214, 2006.