Beat Detection & MIDI
Carmine Casciato, McGill University

The area of music rhythm detection by computers is a relatively new field, but one might consider it a part of a long history of pattern recognition problems. "A full understanding of a piece of music may require both creativity and extensive training, but it also seems fair to say that a broadly available ability to perceive patterns at various levels of organization lies close to the heart of the matter." (Dannenberg and Mont-Reynaud 1987) Music rhythm detection involves somewhat chaotic signals. There is almost fractal self-similarity on a temporal dimension. Large rhythms are composed of smaller ones. Human performances of those rhythms are not mechanically accurate. This inaccuracy can be developed to imply pulse or syncopation. Despite the signal complexity, all humans are able to tap out a beat from a piece of music without much effort, regardless of style or changes in rhythm or tempo. Using a computer to find the pulse of music that changes in tempo remains a challenging issue.

Due to its ubiquitous use in early music technology research and design, music in the form of MIDI signals was one object of beat tracking algorithms, especially for improvisation. Audio signals are subject to noise and amplitude masking effects. Due to its design as a musical performance protocol, MIDI is relatively easy to analyze for temporal aspects such as note onsets. (In actuality, many beat tracking designs that work with MIDI data require minor changes to work with audio signals(Dixon 2001, Cemgil 2000, Dannenberg 1990, Pardo 2004)). Drawbacks include a maximum sampling rate of 3.125kHz and its focus on piano keyboard input (Moore 1988). Also, the note message does not take into account the perceived onset time, which depends on the rise-time of the instrument sound.

One early attempt at MIDI beat tracking was the real time improvisation follower by Roger Dannenberg and Bernard Mont-Reynaud (1987). Their system concentrated on following a solo on a unnamed instrument over a 12-bar blues progression. Existing solos were analyzed using a probablistic approach. That is, for each eighth note of the 12-bar progression, the likelihood that a given pitch would be played on that beat was calculated and used to provide chordal accompaniment during real time performances. The beat tracking used weighted differences between the onsets of notes to detect and predict pulse, although it was not mentioned whether these were in the form of MIDI data. In 1990, Allen and Dannenberg model beat using states of phase and period parameters. These two approaches, stochastic and oscillatory respectively, re-occur often in future research.

Cemgil et al. (2000) provide an example of using a stochastic approach for a tempo tracker. The authors makes a distinction between a tempo tracker and a beat tracker, as the latter assumes a constant tempo. Tempo is modelled as a dynamical state system corresponding to "a perfect metronome corrupted by Gaussian noise" (Dixon 2001). A Kalman filter estimates the hidden variables of a multiscaling measurement model called to tempogram. The technique can be adapted to MIDI or audio (by changing the tempogram) offline or real time data, and in concert with a score or 'blind' tracking. Raphael (2003) and Takeda et al. (2003) also employ a probabilistic design using Hidden Markov Models. The underlying theme is that beat by note onset is an observed process that corresponds to a tempo estimate by a certain degree of similarity, which can be calculated or approximated by hidden variables in a state model.

Pardo (2004) implements a oscillatory design to compare to Cemgil by using the same test corpus and evaluation heuristics. The design uses a series of inter-related functions to update the period and phase of the tempo estimate, essentially updating the estimate using a windowed sample of past inter-onset intervals. Three function parameters can be varied to change the tracking behaviour of the functions, including the window size and the weight of past IOIs. The system was trained using 99 performance of *Michelle* from the Cemgil corpus for a given parameter combination, and judged according to period error, phase error, and an error score introduced in the earlier work that combines the first two. 5000 parameters were randomly selected from a uniform distribution over the the interval (0,1). The optimal combination was then chosen and applied to the *Yesterday* performances. The average phase error was 23ms, which makes it more or less adequate for real time situations. The Kalman filter design, despite being conceptually and computationally more complex, did not prove to be statistically more accurate.

For a thorough discussion on the history of rhythm perception models, beat tracking and tempo tracking designs, and general issues in the evaluation of these designs, in the context of implementing a offline tempo tracker, the reader is referred to Dixon (2001).

References

Allen, P. and R. Dannenberg. 1990. Tracking musical beats in real time. In *Proceedings of the International Computer Music Conference 1990*: 140–3.

Cemgil, A., B. Kappen, P. Desain, and H. Honing. 2000. On tempo tracking: Tempogram representation and Kalman filtering. *Journal of New Music Research* 29: 259–73.

Dannenberg, R. and B. Mont-Reynaud. 1987. Following an Improvisation in Real Time. In *Proceedings of the International Computer Music Conference 1987*: 241–7.

Dixon, S. 2001. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research* 30 (1): 39–58.

Moore, F. R. 1988. The dysfunctions of MIDI. *Computer Music Journal* 12(1): 19–28

Pardo, B. 2004. Tempo tracking with a single oscillator. *In Proceedings of the International Conference on Music Information Retrieval 2004*.

Raphael, C. 2002. A hybrid graphical model for rhythmic parsing*. Artificial Intelligence* 137: 217–38.

Takeda, H., T. Nishimoto, and S. Sagayama. 2002. Automatic rhythm transcription from multiphonic MIDI signals. In *Proceedings of the International Conference on Music Information Retrieval 2003*.