Beat Tracking with Musical Knowledge

Simon Dixon [1] and **Emilios Cambouropoulos** [2]

Abstract. When a person taps a foot in time with a piece of music, they are performing beat tracking. Beat tracking is fundamental to the understanding of musical structure, and therefore an essential ability for any system which purports to exhibit musical intelligence or understanding. We present an off-line multiple agent beat tracking system which estimates the locations of musical beats in MIDI performance data. This approach to beat tracking requires no prior information about the input data, such as the tempo or time signature; all required information is derived from the performance data. For constant tempo performances, previous beat tracking systems have proved successful; however, these systems fail when there are large variations in tempo. We examine the role of musical knowledge in guiding the beat tracking process, and show that a system equipped with knowledge of musical salience is able to track the beat of music even in the presence of large tempo variations. Results are presented for a large corpus of expressively performed classical piano music (13 complete sonatas), containing a full range of tempos and much variability in tempo within sections. With the musical knowledge disabled, the beats are tracked about 75% correctly; the inclusion of musical knowledge raises this figure to over 90%.

1 Introduction

The term *beat* refers to a regular pulse perceived when listening to music – this is usually the temporal level at which listeners tap their feet or clap their hands. The perception of the beat is fundamental to the understanding of the timing structure of music, and therefore also a necessary component of "musically intelligent" applications, such as intelligent digital audio editors, music notation software, content-based search engines and interactive musical performance systems. The action of finding an appropriate beat rate (tempo) will be referred to as *beat induction*, whereas the task of following the beat, that is, finding the locations of beats, will be referred to as *beat tracking*. Ordinary human listeners are competent in performing these tasks; equivalent performance on a computer has proved remarkably hard to achieve, except for simple strict tempo cases.

One reason that computer systems are quite weak in emulating human competency in rhythm understanding is that they often do not have access to sufficient musical knowledge to guide the tracking process when significant local variations in the beat occur. Most beat tracking and beat induction systems rely solely on the timing of note onsets or equivalently the time spans between successive onsets [2, 1, 16, 17, 5]. However, it has often been proposed in theoretical work that finding the beat involves matching a regular grid to the *accent* structure of a musical work [13]. Musical accents indicate the relative *salience* of musical events. Musical salience is determined by a number of factors such as note duration, dynamics, pitch, harmony, and cadences. For instance, longer notes and harmonically more stable notes are more salient. The assumption, borne out in traditional Western music, is that beats tend to fall on more accented/salient events. In this paper we show that the use of such musical knowledge can enhance beat-processing applications significantly.

Firstly, we describe a multi-agent beat tracking system that induces and tracks the beat using a multiple competing agent methodology. This system was previously used successfully on music with a steady tempo (e.g. pop music) [5, 6], but it was found that it was not able to track the tempo variations that occur in expressive performances of classical music. By *expressive* performance, we mean that the performers intentionally vary the tempo of the music in order to communicate emotive and structural aspects of the music.

We then describe the incorporation of musical knowledge into the beat tracking system, and show that when musical salience is taken into account, the ability of the system to track the beat of an expressive performance improves significantly. Musical knowledge is used to enhance the system in two ways: firstly by detecting and filtering out non-salient events, and secondly by attaching to each event a salience value that depends on note duration, pitch and dynamics, and guides the evaluation of the beat tracking agents.

Next we outline a series of experiments applying the beat tracking system to expressive performance data stored in MIDI files, a format which contains explicit details of the onsets, offsets, pitch and dynamics of all the performed notes. An evaluation methodology is then described, whereby the correct beats indicated in the musical score are automatically aligned with the beats tracked by the algorithm and a 'goodness of fit' value is calculated. The final section contains the presentation and discussion of results for a large data set of Mozart piano sonatas performed by a professional pianist.

2 The Multi-Agent Beat Tracking System

In previous work [6], we described a multiple agent beat tracking system which processed audio data and found the locations of musical beats, under the assumption of constant tempo. This system was able to track music with minor tempo deviations, such as most popular and dance music. In this section we describe the beat tracking system as it has been adapted for expressively performed music, and then in the following section we describe the incorporation of musical knowledge into the system.

The beat tracking system has two stages of processing. The first stage is beat induction, in which the data is processed to extract the locations of musical events, and then the intervals between event locations are clustered to form initial hypotheses of the tempo. The second stage of processing is beat tracking, in which the location of each beat is determined, and thus tempo fluctuations are tracked.

[1] Austrian Research Institute for Artificial Intelligence, Schottengasse 3, A-1010 Vienna, Austria, email: simon@ai.univie.ac.at
[2] Austrian Research Institute for Artificial Intelligence, Schottengasse 3, A-1010 Vienna, Austria, email: emilios@ai.univie.ac.at

2.1 Beat Induction

The beat induction algorithm is based on the clustering of *inter-onset intervals* (IOI's). In the literature, an IOI is defined as the time between the onsets of two successive events, but we extend the definition to include times between onsets of pairs of events that are separated by intervening event onsets. An IOI object is created for each pair of onsets that are separated in time by between 0.025 and 2.5 seconds, and each IOI object is then assigned to precisely one cluster, according to the clustering algorithm shown below. When an IOI object is created, the cluster with an average IOI closest to the size of the new IOI is found, and if it is sufficiently close, the new IOI is added to the cluster. The IOI is considered sufficiently close if its value is within $Delta$ of the average IOI in the cluster, where $Delta$ is a constant, set to 25ms in this work. If no sufficiently close cluster exists, a new cluster is created and the new IOI placed in it. After all IOIs have been assigned to a cluster, any pair of clusters whose average IOI value differs by less than $Delta$ is merged to form a single cluster containing the IOIs from both clusters.

IOI Clustering Algorithm

For each pair of onset times t_i, t_j (with $t_i < t_j$)
  If $0.025 < t_j - t_i < 2.5$
    Let $I = t_j - t_i$
    Find cluster C_k such that $|Average(C_k) - I|$ is minimum
    If k exists and $|Average(C_k) - I| < Delta$ then
      $C_k := C_k \cup \{I\}$
    Else
      Create new cluster $C_m := \{I\}$
    End If
  End If
End For
For each pair of clusters C_s, C_t
  If $|Average(C_s) - Average(C_t)| < Delta$
    $C_s = C_s \cup C_t$
    Delete cluster C_t
  End If
End For

For example, Figure 1 shows clustering for five events (A, B, C, D, E) into intervals of similar size. Cluster C1 contains the intervals AB, BC and DE, while cluster C2 contains AC and CD. Each cluster is identified by its average interval size.

After clustering is completed, a score is calculated for each cluster, based on the number of IOIs in the cluster. The highly ranked clusters usually correspond to either the *inter-beat interval* (IBI), that is, the time between successive beats, or small integer multiples or fractions of the IBI. For example, suppose that C2 represents the IBI; then C1 represents half the IBI and C4 represents double the IBI. Clusters related in this way are awarded extra points, and the IOI hypotheses are sorted into their final ranked order.

In previous work [4, 5], it was found that for pop music, the correct tempo can be induced from a 5-10 second excerpt of the music with 90% reliability, and by using multiple (or longer) excerpts, the reliability quickly approaches 100%. In this work, we rely on the multiple agent architecture described in the following subsection to examine the multiple hypotheses generated by the clustering algorithm and choose the most likely candidate at that time. In other words, it is not necessary to commit to a particular tempo hypothesis at this stage of processing.

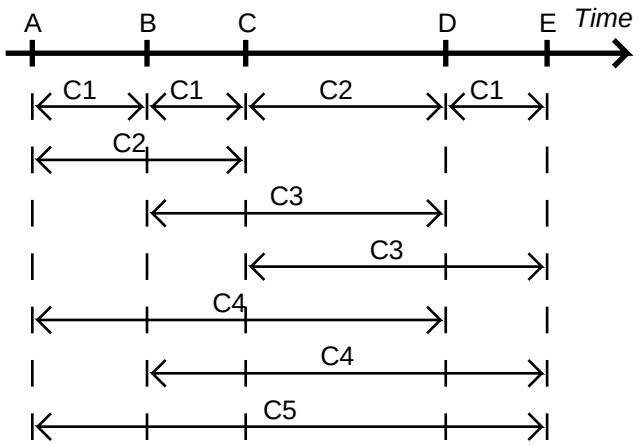

Figure 1. Clustering of inter-onset intervals

2.2 Beat Tracking

The beat induction algorithm computes hypotheses concerning the inter-beat interval, but does not calculate the location of the beat for these hypotheses. In Figure 1, one hypothesis might be that C2 represents the inter-beat interval, but this does not determine whether events A, C and D are beat locations or whether B and E are beat locations. By analogy with wave theory, we say that the beat induction hypotheses concern the *frequency* but not the *phase* of the beat.

The beat locations are determined by an agent-based architecture which simultaneously examines multiple hypotheses about the frequency and phase of the beat throughout the music. The agents are characterized by their *state* and *history*. The state is the agent's current hypothesis of the beat frequency and phase, and the history is the sequence of beat locations selected so far by the agent. Each agent is evaluated on the basis of its history, with higher scores being awarded for greater regularity in the spacing between events, greater salience of chosen events, and fewer gaps in the sequence.

Initially, a number of agents are created for each of the hypotheses from the beat induction stage; for each tempo, one agent is created for each of the first few events in the piece, with its phase set to 0 at the time of the event onset. A simple example is shown in Figure 2, where there are two tempo hypotheses, and two starting locations, events A and B. Agents 1 and 2 start with the same phase, but different tempo, while Agent 3 starts with a different phase, but same tempo as Agent 2. Note that there is no need to start an agent with the tempo of Agent 1 and phase of Agent 3, since Agent 1 covers event B itself.

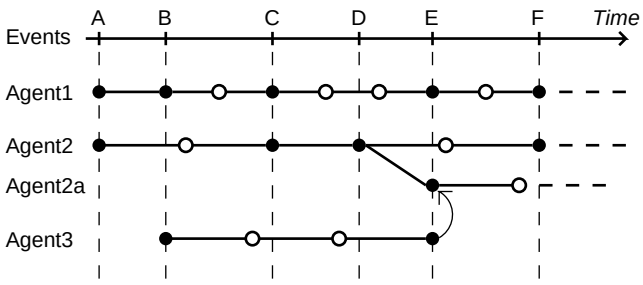

Figure 2. Beat tracking agents (see text for details)

The main loop of the beat tracking section passes each event to each agent, which compares the event's onset time with the predicted beat location. The agents have two windows of tolerance (shown in Figure 3): an inner window, within which the agent is sure that the event corresponds to the predicted beat location, and an outer window, within which the agent is unsure if the event should be accepted as a beat location. The inner window is symmetrical and has a fixed size of 70ms, the outer window is asymmetrical and its size varies with the IBI. If the event falls in the inner window, it is added to the agent's history, and the agent's tempo and phase hypotheses are updated. If the event falls in the outer window around the predicted location, the agent creates a clone which accepts the event as a beat location, while the current agent rejects the event. In Figure 2, Agent 2 creates Agent 2a when event E falls in its outer window of expected beat locations. This guards against the current beat location being lost due to a rogue event, whilst also allowing for moderate deviations in tempo and phase to occur. When an accepted event is more than one beat from the previous beat location determined by the agent, the missing beats are filled in by interpolation (shown by hollow circles in Figure 2).

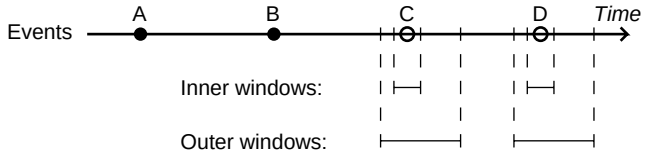

Figure 3. Beat prediction windows for a single agent: A and B are reported beat positions; C and D are the next two predicted beat locations

As the agents track the beats, a confidence value is maintained for each agent. This value is increased each time an event is accepted as a beat location. The amount of increase depends on the salience of the event (described in section 3) and its proximity to the predicted beat location. The event salience value is multiplied by a number between 0 and 1 representing the distance between the predicted and actual beat locations relative to the prediction window, and this value is added to the agent's confidence value.

It often occurs that two or more agents come to the same conclusion about the current tempo and beat phase. Since these are the only variables that determine the agent's state (and therefore its future behavior), it is computationally advantageous to remove all but one of these agreeing agents, retaining only the agent with the highest confidence (that is, with the best scoring history). In Figure 2, Agent 3 is terminated when its state coincides with that of Agent 2a, as indicated by the arrow at event E.

After the last event is processed, the highest scoring current agent is selected, and its history is output as the beat tracking "solution". It is also possible to view a trace of the agents and their scores at each event during processing. For aural testing and demonstrations of the system, the music can also be played back or saved to file with a click track added to it, that is, a percussion track indicating the positions of beats as reported by the beat tracking agent.

3 Musical Knowledge: Salience Calculation

In the previous system, under the assumption of an almost constant tempo, it was sufficient to evaluate the beat tracking agents on the basis of their ability to find a sequence of regularly spaced events with few gaps and little variation in the spacing. The level of musical knowledge possessed by the agents in this system was extremely limited, but it was sufficient for relatively constant tempo performances. One of the keys to the success of the system was that it used audio input, for which the onset detection algorithm was only able to detect *some* of the musical events, which tended to be the more salient ones, in the sense of having a sharp increase in amplitude at the note onset. In other words, the onset detection algorithm effectively filtered the events, implicitly selecting only those which were more salient. According to music theory, the more salient events are also more likely to occur in rhythmically strong positions, such as on beat locations. It was found that the inclusion of less salient events actually hinders the performance of the beat tracking system, as it greatly enlarges the search space without adding more correct solutions. In this work, we use MIDI input, which represents all events explicitly, including many non-salient events, so the system must choose explicitly between the many extra possible solutions.

We claim that the system requires some form of musical knowledge to guide its selection of alternative possible solutions. It is expected that among the many beat tracking agents the one that detects the largest number of salient events is most likely to be correct. We now define musical *events* and describe how musical knowledge can be used to generate a salience value for each musical event.

In the context of these experiments, musical events are taken to be either individual notes or chords (synchronous collections of notes). Empirical research has shown that two tones are heard as being synchronous if their onset times differ by less than roughly 40ms; for more than two tones this threshold is up to 70ms [11]. We have used the threshold of 70ms to convert the MIDI data into musical events (notes and chords).

At the next stage, a salience strength is calculated for each musical event. In these experiments the following three salience factors have been taken into account: the duration of notes in each event, the dynamic value (MIDI velocity) and the pitch value (MIDI note number). According to music theory, notes with longer durations, higher dynamic values or lower pitches are considered to be more salient for communicating the rhythmic structure of the music. The salience value for a chord is calculated using the longest duration, the sum of the dynamic values and the lowest pitch of all the notes in the chord.

Although the qualitative effect of each of these parameters on musical salience is known, no quantitative measure has been proposed by which the three factors can be combined into a single salience value for each event. Two types of salience functions were tested, an additive function $s_{add}(d, p, v)$, which uses a linear combination of the parameters, and a multiplicative function $s_{mul}(d, p, v)$ which is non-linear. A threshold function is used to restrict the range of the pitch parameter, and constants are used to set the relative weights of the three parameters, with the sign of the constants determining whether the relationship is direct or inverse.

$$s_{add}(d, p, v) = c_1.d + c_2.p[p_{min}, p_{max}] + c_3.v$$

$$s_{mul}(d, p, v) = d.(c_4 - p[p_{min}, p_{max}]).log(v)$$

where:

c_1, c_2, c_3 and c_4 are constants,
d is duration in seconds,
p is pitch (MIDI number),
v is dynamic value (MIDI velocity), and

$$p[p_{min}, p_{max}] = \begin{cases} p_{min}, & p \leq p_{min} \\ p, & p_{min} < p < p_{max} \\ p_{max}, & p_{max} \leq p \end{cases}$$

After a little experimentation, we chose the following values: $c_1 = 300; c_2 = -4; c_3 = 1; c_4 = 84; p_{min} = 48; p_{max} = 72$. The parameters were set to make the duration of notes the most significant factor in the salience calculation, with the dynamics and pitch factors becoming more influential when notes have relatively similar durations. The precise values of parameters do not make a large overall difference to the results across the entire data set, but can have a significant effect on particular examples. A more thorough investigation of parameter values is a subject for further research.

4 Beat Tracking Experiments

We now describe the various beat tracking experiments performed. The results are presented in section 6.

4.1 Experiment 1: Constant Salience

The beat-tracking algorithm was applied on the untreated MIDI files taking into account only the event onset times. Each event was assigned a constant salience value for use in the agents' evaluation function. As expected, this experiment did not produce particularly good results, as the agents have many equal-strength event options from which to choose possible beat locations. This experiment set the base level from which we were able to measure the performance gain obtained by the application of musical knowledge in the beat tracking system, as described in the following two experiments.

4.2 Experiment 2: Filtered MIDI files

In this second experiment, MIDI files were pre-processed to discard non-salient events. This was achieved by keeping only events with an additive salience exceeding a certain threshold, and applying the beat tracking algorithm to the remaining events, using a constant salience value as in experiment 1. For this experiment, the threshold was set to an intuitively meaningful value, that is, the additive salience value of a note of average pitch, average velocity and duration 200ms, which is just below half of the modal inter-beat interval.

4.3 Experiment 3: Agents with Salience Functions

In the previous experiment, we used a binary notion of salience: an event is either salient and is thus preserved, or non-salient and thus omitted. In the final experiments, we allowed multi-valued salience strengths to be incorporated in the beat-tracking process for the selection of the best agent. Instead of counting the number of events that an agent has traced, which is approximately what occurs when a constant salience function is used, a progressive confidence score (sum of adjusted salience values for all the events traced by an agent) was computed for each agent and used to select the final solution. Two experiments were performed: experiment 3a used the multiplicative salience function $s_{mul}(d, p, v)$ and experiment 3b used the additive salience function $s_{add}(d, p, v)$, defined in section 3. Before presenting the results of the experiments in section 6, we briefly describe our approach to evaluation.

5 Evaluation

Much work on beat tracking has worked from notated music rather than performance data [14, 15, 3], and those using performance data have often used simple pieces such as nursery rhymes [12] or short excerpts of the melody lines of pieces [16], with the exception of Goto [7, 9, 8, 10], who used audio CD's of popular music. Apart from the difficulty of beat tracking the tempo variations in performed music, a further problem with using performance data is that the results must be evaluated by the subjective location of beat positions [8, 6].

A great advantage with the current work is that the evaluation of results is not based on a subjective measure of beat locations, since a large corpus of performance data together with the corresponding musical notation is available. This enables the direct evaluation of the system by comparing the beat tracking results (the reported beats) with the beat specified in the musical score (the notated beats). The testing is in fact performed automatically, since the musical scores are available in a symbolic format.

To evaluate the beat tracking of a section of music, we first match the reported beat locations with the musical score data. Each reported beat is matched with the nearest notated beat, unless the gap is greater than a fixed tolerance value, in which case the beats are left unmatched. This creates three result categories: matched pairs of reported and notated beats, unmatched reported beats (false positives) and unmatched notated beats (false negatives). These are combined using the following formula:

$$Evaluation = \frac{n}{n + F^+ + F^-}$$

where n is the number of matched pairs, F^+ is the number of false positives, and F^- is the number of false negatives. In this work, the tolerance window for matching beats was chosen to match the window for determining when notes are simultaneous, that is, 70ms. A less strict correctness requirement would allow the matching of pairs over a larger time window, with partial scores being awarded to "near misses", and the numerator of the equation being replaced by the sum of these partial scores.

The evaluation function yields a value between 0 and 1, which we express as a percentage. The values are intuitively meaningful: if the only errors are false positives, the value is the percentage of reported beats which are matched with notated beats; if the only errors are false negatives, the value is the percentage of notated beats which were reported.

One aspect of evaluation which has yet to be addressed in this work is the problem that it is possible to track beats at more than one level. For example, in a piece that has a very slow tempo, it might be natural to track the beat at double the rate indicated by the notation [3]. The perceived beat and notated beat are not necessarily the same. The formula shown above gives meaningful results only when the rhythmic level of beat tracking coincides with the notated beat.

6 Results and Discussion

The beat tracking system was tested on 13 complete piano sonatas by Mozart (KV279-KV284, KV330-KV333, KV457, KV475 and KV533), played by a professional pianist. This totals several hours of music, and over 100000 notes. The files were divided into sections as notated in the music, and beat tracking was performed separately on each file (222 files in all). The software is written in C++ and runs on a Linux system, taking under 5 minutes to process all 13 sonatas.

The first set of results shows the rhythmic level chosen by the highest scoring agent relative to the musical score. For almost all sections, a musically plausible rhythmic level was chosen, with slightly worse performance in experiment 1 where no musical knowledge was used. Table 1 shows the number of sections which were tracked at each rhythmic level (expressed as a multiple of the tempo). The rows labelled *other* and *fail* represent the cases where a musically unrelated

rhythmic level was chosen, and when beat tracking failed to produce a solution at all, respectively. The majority of pieces were tracked at the notated level, with others being tracked at double, four times or half the notated level, as would often be done by human listeners.

Table 1. Rhythmic levels of beat tracking solutions

Rhythmic level	Number of sections			
	Exp1	Exp2	Exp3a	Exp3b
1	121	143	146	137
2	40	40	41	42
3	4	3	3	4
4	23	23	20	22
0.5	10	9	10	10
1.5	16	0	1	5
other	8	2	1	2
fail	0	2	0	0

Table 2 shows the results of evaluating the beat tracking of the sections tracked at the notated rhythmic level, using the evaluation formula from section 5. For each experiment we show the number of sections (n) and the percentage of sections which achieved various minimum scores. Experiment 1 provides the base level performance of the system without musical knowledge. Experiment 2 gave mixed results, since the removal of events which were deemed to be non-salient also removed many events which occurred on beats, making it impossible for the beat tracking system to determine beat positions. Nevertheless, the net result of this experiment was positive. The third experiment shows a significant improvement in performance due to the use of salience in the beat tracking process, with the additive salience function $s_{add}(d, p, v)$ performing slightly better than the multiplicative function $s_{mul}(d, p, v)$.

Table 2. Evaluation of beat tracking at rhythmic level 1

Result	Exp. 1		Exp. 2		Exp. 3a		Exp. 3b	
Range	n	%	n	%	n	%	n	%
100%	42	34.7	17	11.9	54	37.0	59	43.1
$\geq 95\%$	46	38.0	50	35.0	71	48.6	82	59.9
$\geq 90\%$	57	47.1	87	60.8	99	67.8	105	76.6
$\geq 85\%$	63	52.1	105	73.4	116	79.5	118	86.1
$\geq 80\%$	68	56.2	115	80.4	130	89.0	127	92.7
$\geq 70\%$	81	66.9	127	88.8	137	93.8	130	94.9
$\geq 50\%$	100	82.6	136	95.1	143	97.9	136	99.3
$\geq 0\%$	121	100.0	143	100.0	146	100.0	137	100.0
Average	75.4%		85.0%		88.5%		91.1%	

At the bottom of Table 2 we show as a summary of the beat tracking experiments, the weighted average of the beat tracking evaluation results (weighted by the number of beats in each section). This gives a clear measure of the performance gain from the inclusion of musical knowledge (in the form of salience values) into the system.

We have endeavoured to keep the system as general as possible, by not encoding specific details of the musical style of the test data. The parameter values and knowledge used in the system are quite low-level, derived mostly from the human perception literature. We expect the system to perform equally well with other musical styles, but we do not have the test data needed to verify this claim. A version of the system that uses audio input performs very well with various styles of pop music [6]. Not a great deal of effort was spent in 'tweaking' parameters; the overall performance of the system is stable with respect to small changes in parameter values. Further improvement in the system by adjustment of parameters would probably only tune the system to the test data set.

Listening to the performance of the beat tracking system demonstrates that it tracks tempo variations in a way that could be described as 'musically intelligent'. The system is not perfect, but a score of 100% is not possible, as some parts of the music are played in free time, without any notion of beat. It would be interesting, but is beyond the scope of this project, to perform beat tracking experiments with human listeners to compare the results of these experiments with human beat tracking ability.

ACKNOWLEDGEMENTS

This research is part of the project Y99-INF, sponsored by the Austrian Federal Ministry of Education, Science and Culture in the form of a START Research Prize and support to the Austrian Research Institute for Artificial Intelligence. We also wish to thank the L. Bösendorfer Company, Vienna, for use of the performance data, and Gerhard Widmer for comments on an earlier draft of the paper.

REFERENCES

[1] P. Desain, 'A connectionist and a traditional ai quantizer, symbolic versus sub-symbolic models of rhythm perception', *Contemporary Music Review*, **9**, 239–254, (1993).

[2] P. Desain and H. Honing, 'Quantization of musical time: A connectionist approach', *Computer Music Journal*, **13**(3), (1989).

[3] P. Desain and H. Honing, 'Computational models of beat induction: the rule-based approach', in *Proceedings of the IJCAI'95 Workshop on Artificial Intelligence and Music*. International Joint Conference on Artificial Intelligence, (1995).

[4] S. Dixon, 'Beat induction and rhythm recognition', in *Proceedings of the Australian Joint Conference on Artificial Intelligence*, pp. 311–320, (1997).

[5] S. Dixon, 'A beat tracking system for audio signals', in *Proceedings of the Diderot Forum on Mathematics and Music*, pp. 101–110. Austrian Computer Society, (1999).

[6] S. Dixon, 'A lightweight multi-agent musical beat tracking system', in *Proceedings of the AAAI Workshop on Artificial Intelligence and Music: Towards Formal Models for Composition, Performance and Analysis*. AAAI Press, (2000). To appear.

[7] M. Goto and Y. Muraoka, 'A real-time beat tracking system for audio signals', in *Proceedings of the International Computer Music Conference*. Computer Music Association, San Francisco CA, (1995).

[8] M. Goto and Y. Muraoka, 'Issues in evaluating beat tracking systems', in *Issues in AI and Music – Evaluation and Assessment: Proceedings of the IJCAI'97 Workshop on AI and Music*. International Joint Conference on Artificial Intelligence, (1997).

[9] M. Goto and Y. Muraoka, 'Real-time rhythm tracking for drumless audio signals – chord change detection for musical decisions', in *Proceedings of the IJCAI'97 Workshop on Computational Auditory Scene Analysis*. International Joint Conference on Artificial Intelligence, (1997).

[10] M. Goto and Y. Muraoka, 'An audio-based real-time beat tracking system and its applications', in *Proceedings of the International Computer Music Conference*. Computer Music Association, San Francisco CA, (1998).

[11] S. Handel, *Listening: An Introduction to the Perception of Auditory Events*, Bradford, MIT Press, 1989.

[12] E.W. Large, 'Beat tracking with a nonlinear oscillator', in *Proceedings of the IJCAI'95 Workshop on Artificial Intelligence and Music*. International Joint Conference on Artificial Intelligence, (1995).

[13] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*, MIT Press, 1983.

[14] H.C. Longuet-Higgins and C.S. Lee, 'The perception of musical rhythms', *Perception*, **11**, 115–128, (1982).

[15] H.C. Longuet-Higgins and C.S. Lee, 'The rhythmic interpretation of monophonic music', *Music Perception*, **1**(4), 424–441, (1984).

[16] D. Rosenthal, 'Emulation of human rhythm perception', *Computer Music Journal*, **16**(1), 64–76, (1992).

[17] R. Rowe, 'Machine listening and composing with cypher', *Computer Music Journal*, **16**(1), 43–63, (1992).