Automatic Extraction of Tempo and Beat from Expressive Performances

Simon Dixon
Austrian Research Institute for Artificial Intelligence,
Schottengasse 3, Wien 1010, Austria.
Email: simon@oefai.at
Voice: +43-1-533 6112 22
Fax: +43-1-533 6112 77

May 16, 2001

Abstract

We describe a computer program which is able to estimate the tempo and the times of musical beats in expressively performed music. The input data may be either digital audio or a symbolic representation of music such as MIDI. The data is processed off-line to detect the salient rhythmic events and the timing of these events is analysed to generate hypotheses of the tempo at various metrical levels. Based on these tempo hypotheses, a multiple hypothesis search finds the sequence of beat times which has the best fit to the rhythmic events. We show that estimating the perceptual salience of rhythmic events significantly improves the results. No prior knowledge of the tempo, meter or musical style is assumed; all required information is derived from the data. Results are presented for a range of different musical styles, including classical, jazz, and popular works with a variety of tempi and meters. The system calculates the tempo correctly in most cases, the most common error being a doubling or halving of the tempo. The calculation of beat times is also robust. When errors are made concerning the phase of the beat, the system recovers quickly to resume correct beat tracking, despite the fact that there is no high level musical knowledge encoded in the system.

Introduction

The task of beat tracking or tempo following is perhaps best described by analogy to the human activities of foot-tapping or hand-clapping in time with music, tasks of which average human listeners are capable. Despite its apparent intuitiveness and simplicity compared to the rest of music perception, beat tracking has remained a difficult task to define, and still more difficult to implement in an algorithm or computer program. In this paper, we address the problem of beat tracking, and describe algorithms which have been implemented in a computer program for discovering the times of beats in expressively performed music.

The approach taken in this work is based on the belief that beat is a relatively low-level property of music, and therefore the beat can be discovered without recourse to high-level musical knowledge. It has been shown that even with no musical training, a human listener can tap in time with music (Drake et al., 2000). At the same time, it is clear that higher level knowledge aids the perception of beat. Drake et al. (2000) also showed that trained musicians are able to tap in time with music more accurately and require less time to synchronise with the music than non-musicians.

The primary information required for beat tracking is the onset times of musical events, and this is sufficient for music of low complexity and little variation in tempo. For more difficult cases, we show that a simple estimation of the salience of each musical event makes a significant improvement in the ability of the system to find beat times correctly.

Motivation and Applications

There are several areas of research for which this work is relevant, namely performance analysis, perceptual modelling, audio content analysis and synchronisation of a musical performance with computers or other devices.

Performance analysis investigates the interpretation of musical works, for example, the performer's choice of tempo and expressive timing. These parameters are important in conveying structural and emotional information to the listener (Clarke, 1999). By finding the times of musical beats, we can automatically calculate the tempo and variations in tempo within a performance. This accelerates the analysis process, thus allowing more wide-ranging studies to be performed.

Perception of beat is a prerequisite to rhythm perception, which in turn is a fundamental part of music perception. Several models of beat perception have been proposed (Steedman, 1977; Longuet-Higgins and Lee, 1982; Povel and Essens, 1985; Desain, 1992; Rosenthal, 1992; Parncutt, 1994; van Noorden and Moelants, 1999). Although this work is not intended as a perceptual model, it can inform perceptual models by examining the information content of various musical parameters, for example the relationship between the musical salience and the metrical strength of events.

Audio content analysis is important for automatic indexing and content-based retrieval of audio data, such as in multimedia databases and libraries. This work is also necessary for applications such as automatic transcription or score extraction from performance data.

Another application of beat tracking is in the automatic synchronisation of devices such as lights, electronic musical instruments, recording equipment, computer animation and video with musical data. Such synchronisation might be necessary for multimedia or interactive performances or studio post-production work. The increasingly large amounts of data processed in this way leads to a demand for automatisation, which requires that the software involved operates in a “musically intelligent” way, and the interpretation of beat is one of the most fundamental aspects of musical intelligence.

Definitions of Terms

We assign the following meanings to terms used throughout the paper. *Beat*, as a phenomenon, refers to the perceived pulses which are approximately equally spaced and define the rate at which the notes in a piece of music are played. For a specific performance, the beat is defined by the occurrence times of these pulses (*beat times*), which are measured relative to the beginning of the performance.

A *metrical level* is a generalisation of the concept of the beat, corresponding to multiples or divisors of the beat which also divide the meter and any implied subdivision of the meter evenly and begin on the first beat of the meter. For example, we can talk about the *quarter note level* of a piece in 3/4 or 4/4 meter, but not of a piece in 3/8 or 6/8 meter. Similarly, a piece in 4/4 meter has a *half note level* and a *whole note level*, whereas a piece in 3/4 time has a *dotted half note level*. The *primary metrical level* is given by the denominator of the time signature (the *notated level*), which does not necessarily equate to the perceptually preferred metrical level for the beat.

Score time is defined as the relative timing information derived from durations of notes and rests in the score, measured in abstract units such as quarter notes and eighth notes. In conjunction with a metronome setting, score time can be converted to (nominal) note onset times measured in seconds. The term *performance time* is used to refer to the concrete, measured, physical timing of note onsets. We defer discussion of the practical difficulties with the measurement of performance time until the section on evaluation.

A *mechanical* or *metrical performance* is a performance played strictly in score time. That is, all quarter notes have equal duration, all half notes are twice as long as the quarter notes, and so on. An *expressive performance* is any other performance, such as any human performance.

Tempo refers to the rate at which musical notes are played, expressed in score time units per real time unit, for example quarter notes per minute. When the metrical level of the beat is known, the tempo can be represented by the number of beats per time unit (beats per minute is most common), or inversely as the *inter-beat interval*, measured in time per beat. Further, the tempo might be an instantaneous value, such as the inter-beat interval measured between two successive beats, or an average tempo measured over a longer period of time. A measure of central tendency of tempo over a complete musical excerpt is called the *basic tempo* (Repp, 1994), which is the implied tempo around which the expressive tempo varies (not necessarily symmetrically).

Tempo induction is the process of estimating the basic tempo from musical data, and a *tempo hypothesis* is one such estimate generated by the tempo induction algorithm shown in Figure 1. *Beat tracking* is the estimation of beat times based on a given tempo hypothesis. The term *beat phase* is used to refer to the times of musical events (or estimated beat times) relative to actual beat times.

Outline of Paper

In the background section, we review the literature on the modelling and analysis of tempo and beat in music, from score data, symbolic performance data and audio data, and conclude with a brief summary of models of performance timing and how they relate to this work.

The following three sections give a detailed description of the algorithms used in tempo induction, beat tracking and calculating musical salience respectively. For tempo induction from audio data, the onsets of events are found using a time-domain method which seeks local peaks in the slope of the amplitude envelope. For symbolic performance data, the notes are grouped by temporal proximity into rhythmic events, and the salience of each event is estimated. The tempo induction algorithm then proceeds by calculating the inter-onset intervals between pairs of events (not necessarily adjacent), clustering the intervals to find common durations, and then ranking the clusters according to the number of intervals they contain and the relationships between different clusters, to produce a ranked list of basic tempo hypotheses. These hypotheses are the starting point for the beat tracking algorithm, which uses a multiple agent architecture to test the different tempo and phase hypotheses simultaneously, and finds the agent whose predicted beat times match most closely to those implied by the data. The evaluation of agents is based on the assumption that the more salient events are more likely to occur in metrically strong positions. The estimation of musical salience is based on note duration, density, pitch and amplitude.

Evaluation of the system is then discussed, and a number of situations are presented in which the desired behaviour of a perfect beat tracking system is not clear. A practical methodology for evaluation is then described, including a formula for rating overall beat tracking performance on a musical work.

The results section begins with a brief description of the implementation details of the system, and then presents results for tempo induction and beat tracking of audio data, and then for symbolic data, using various measures of musical salience. We show that for popular music, which has a very regular beat, the onset times and amplitudes are sufficient for calculating beat times, but for music with greater expressive variations, note duration becomes an important factor in being able to estimate beat times correctly.

The paper concludes with a discussion of the results obtained, the strengths and weaknesses of the system, and a preview of several possible directions of further work.

Background: Tempo Induction and Beat Tracking

Before discussing the methods, algorithms and results of the beat tracking system, we provide a brief background of previous work in the area. The literature on tempo induction and beat tracking is reviewed here in three parts, based on the type of input data. Firstly we examine models processing mechanical performances or musical scores, then we look at work involving symbolic performance data (usually MIDI), and finally we describe approaches to analysis of audio data. We conclude this section with a review of models of performance timing and discuss their relevance to beat tracking.

Scores and Mechanical Performance Data

For data with no expressive timing, the inter-beat interval is normally a multiple of the shortest duration, and all durations can be expressed in terms of rational multiples of this interval. Research using this type of data usually goes beyond tempo induction, and tries to induce the complete metrical hierarchy.

Steedman (1977) describes a model of perception using note durations to infer accents, and melodic repetition to infer metrical structure. He assumes that the meter is established clearly before any syncopation can occur, and therefore weights information at the beginning of a piece more highly than that which occurs later.

A model of rhythm perception developed by Longuet-Higgins and Lee (1982) predicts beat times and revises the predictions in the light of the timing of events. For example, after the first two onsets are processed, it is predicted that the third onset will occur after an equal time interval so that the 3 events are equally spaced. If this expectation is fulfilled, the next expected interval is double the size of the previous interval. This method successfully builds binary hierarchies, but does not work for ternary meters. An extension of this work (Longuet-Higgins and Lee, 1984) provides a formal definition of syncopation and describes a preferred rhythmic interpretation as one which avoids syncopation.

Lerdahl and Jackendoff (1983) describe meter perception as the process of finding periodicities in the phenomenal and structural accents in a piece of music. They propose a set of metrical preference rules, based on musical intuitions, which are assumed to guide the listener to plausible interpretations of rhythms. The rules prefer structures where: beats coincide with note onsets; strong beats coincide with onsets of long notes; parallel groups receive parallel metrical structure; and the strongest beat occurs early in the group.

Povel and Essens (1985) propose a model of perception of temporal patterns, based on the idea that a listener tries to induce an internal clock which matches the distribution of accents in the stimulus and allows the pattern to be expressed in the simplest possible terms. They use patterns of identical tone bursts at precise multiples of 200ms apart to test their theory. They do not suggest how the theory should be modified for musical data or non-metrical time.

A theoretical and experimental comparison of the above models is reported by Lee (1991). He concludes that every meter has a canonical accent pattern of strong and weak beats, and that listeners induce meter by matching the natural accent patterns occurring in the music to the canonical accent pattern of possible rhythmic interpretations. In this model, major syncopations and weak long notes are avoided.

Desain and Honing (1999) compare several tempo induction models, integrating them into a common framework and showing how performance can be improved by optimisation of the parameters.

Symbolic Performance Data

Much of the work in machine perception of rhythm has used MIDI files as input (Rosenthal, 1992; Rowe, 1992; Desain, 1993; Large, 1995; Cemgil et al., 2001).

The input is usually interpreted as a series of event times, ignoring the event duration, pitch, amplitude and chosen synthesizer voice. That is, each note is treated purely as an uninterpreted event. It is assumed that the other parameters do not provide *essential* rhythmic information, which in many circumstances is true. However, there is no doubt that these factors provide useful rhythmic cues; for example, more salient events tend to occur on stronger beats.

Notable work using MIDI file input is Rosenthal's emulation of human rhythm perception (Rosenthal, 1992), which produces multiple hypotheses of possible hierarchical structures in the timing, assigning a score to each hypothesis, corresponding to the likelihood that a human listener would choose that interpretation of the rhythm. This technique gives the system the ability to adjust to changes in tempo and meter, as well as avoiding many implausible rhythmic interpretations.

A similar approach is advocated by Tanguiane (1993), using Kolmogorov complexity as the measure of the likelihood of a particular interpretation, with the least complex interpretations being favoured. He provides an information-theoretic account of human perception, and argues that many of the "rules" of music composition and perception can be explained in information-theoretic terms.

Desain (1993) compares two different approaches to modelling rhythm perception, the symbolic approach of Longuet-Higgins (1987) and the connectionist approach of Desain and Honing (1989). Although this work only models one aspect of rhythm perception, the issue of quantisation, and the results of the comparison are inconclusive, it does highlight the need to model expectancy, either explicitly or implicitly. Expectancy is a type of predictive modelling relevant to real time processing, which provides a contextual framework in which subsequent rhythmic patterns can be interpreted with less ambiguity.

Allen and Dannenberg (1990) propose a beat tracking system that uses beam search to consider multiple hypotheses of beat timing and placement. A heuristic evaluation function directs the search, preferring interpretations that have a "simple" musical structure and make "musical sense", although these terms are not defined. They also do not describe the input format or any specific results.

Large and Kolen (1994); Large (1995, 1996) use a nonlinear oscillator to model the expectation created by detecting a regular pulse in the music. The system does not perform tempo induction; the basic tempo and initial phase must be supplied to the system, which then tracks tempo variations using a feedback loop to control the frequency of the oscillator. On improvised melodies, the system achieved a mean absolute phase error of under 10% for most data, which was considered subjectively good.

Another system which uses multiple hypotheses is from Rowe (1992), who discretises the complete tempo range into 123 inter-beat intervals ranging from 280ms to 1500ms in 10ms steps, corresponding to metronome markings of 40–208 beats per minute. Each tempo theory tries to provide a plausible rhythmic interpretation for incoming events, and the most successful theories are awarded points. The system copes moderately well with simple input data, but cannot deal with complex rhythms.

An alternative approach is to model tempo tracking in a probabilistic framework (Cemgil et al., 2001). The beat times are modelled as a dynamical system with variables representing the rate and phase of the beat, and corresponding to a perfect metronome corrupted by Gaussian noise. A Kalman filter is then used to estimate the unknown variables. Since the beat times are not directly observable from the data, they are induced by calculating a probability distribution for possible interpretations of performances. The system parameters are estimated by training on a data set for which the correct beat times are known. The system performs well (over 90% correct) on a large number of performances of a simple arrangement of a popular song. The results are compared with the current system in Dixon (2001).

Audio Data

The earliest work on automatic extraction of rhythmic content from audio data is found in the percussion transcription system of Schloss (1985). Onsets are detected as peaks in the slope of the amplitude envelope, where the envelope is defined to be equal to the maximum amplitude in each period of the high-pass filtered signal, and the period defined as the inverse of the lowest frequency expected to be present in the signal. The main limitation of the system is that it requires parameters to be set interactively. Also, no quantitative evaluation was made; only subjective testing was performed, by resynthesis of the signal.

The main work in beat tracking of audio data is by Goto and Muraoka (1995, 1997a,b, 1998, 1999) who developed two beat tracking systems for popular music, the first for music containing drums and the second for music without drums. The earlier system (BTS) examines the frequency bands centred on the frequencies of the snare and bass drums, and matches the patterns of onset times of these two drum sounds to a set of pre-stored drum patterns. This limits the system to a very specific style of music, but the beat tracking on suitable songs is almost always successful.

Goto and Muraoka's second system makes no assumption about drums; instead, it uses frequency-domain analysis to detect chord changes, which are assumed to occur in metrically strong positions. This is the first system to demonstrate the use of high level knowledge in directing the lower-level beat tracking process. The high level knowledge is specific to the musical style, which is a major limitation of the system. Furthermore, all music processed by the system is assumed to be in 4/4 time, with a tempo between 61 and

120 quarter note beats per minute, chord changes occurring in strong metrical positions (not every beat), and no tempo changes.

Both systems are based on a multiple agent architecture using a fixed number of agents (28 and 12 in the two systems, respectively). Each agent predicts the beat times using different strategies (parameter settings). One feature of this work which does not appear in most beat tracking work is that three metrical levels (quarter note, half note and whole note) are tracked simultaneously. The system also operates in real time, for which it required a multiple-processor computer at the time it was built. (A fast personal computer today has almost the same computing power.)

Scheirer (1998) also describes a system for the beat tracking of audio signals, based on tuned resonators. The signal is split into 6 frequency bands, and the amplitude envelopes in each band are extracted, differentiated and rectified before being passed to a bank of 150 comb filters (representing each possible tempo on a discretised scale). The output of the filters is summed across the frequency bands, and the maximum output gives the tempo and phase of the signal. The system was evaluated qualitatively on short musical excerpts from various styles, and successfully tracked 41 of the 60 examples. One problem with the system is that in order to track tempo changes, the system must repeatedly change its choice of filter, which implies the filters must be closely spaced to be able to smoothly track tempo variations. However, the system applies no continuity constraint when switching between filters.

Two recent approaches which find periodicities in audio data have been proposed (Cariani, 2001; Sethares and Staley, 2001). Cariani (2001) presents a neurologically plausible model called a recurrent timing net (RTN). The audio data is preprocessed by finding the RMS amplitude in overlapping 50ms windows of the signal, and the resulting data is passed to the RTN, which effectively computes a running autocorrelation at all possible time lags in order to find the most significant periodicities in the data. Sethares and Staley (2001) filter the audio signal into 1/3 octave bands, decimate to a low sampling rate and then search for periodicities using the periodicity transform developed previously in (Sethares and Staley, 1999). Although both of these approaches use audio input, they assume constant tempo performances, and so are not directly relevant to the analysis of expressive performance.

Modelling of Performance Timing

An understanding of rules governing expressive timing is advantageous in developing a system to follow tempo changes. Technical advances over the last decades have facilitated the analysis of timing in music performance in ways that were previously infeasible. Clarke (1999) and Gabrielsson (1999) review research in this area and conclude that expressive timing is generated from the performers' understanding of the musical structure and general knowledge of music theory and musical style. However, there is no precise mathematical model of expressive timing, and the complexity of musical structure from which timing is derived, coupled with the individuality of each performer and performance, makes it impossible to capture musical nuance in the form of rules. Attempts to formulate rules governing the relationship between the score and expressive timing (Todd, 1985; Clarke, 1988; Friberg, 1995) are partially successful, as judged by listening tests and by comparison with performance data, but ignore individual performers' interpretation and cover only limited aspects of musical performance.

Tempo Induction

The beat tracking system has two stages of processing, the initial tempo induction stage, which is described in this section, and the beat tracking stage, which appears in the following section. The tempo induction stage examines the times between pairs of note onsets, and uses a clustering algorithm to find significant clusters of inter-onset intervals. Each cluster represents a hypothetical tempo, expressed as an inverse value, the inter-beat interval, measured in seconds per beat. The tempo induction algorithm ranks each of the clusters, with the intention that the most salient time intervals are ranked most highly. The output from this stage (and input to the beat tracking stage) is the ranked list of tempo hypotheses, each representing a particular beat rate, but saying nothing about the beat times (or beat phase), which are calculated in the subsequent beat tracking stage.

The tempo induction algorithm operates on *rhythmic events*, an abstract representation of the performance data as a weighted sequence of time points. A rhythmic event may represent the onset of a single note or a collection of notes played approximately simultaneously. Events are characterised by their onset time and a salience value which is calculated from the parameters of the constituent notes of the event, such as pitch, loudness, duration, and number of constituents. We now describe the various input data formats, and then the generation of the rhythmic event representation of performance data, and finally present the tempo induction algorithm based on this representation.

Input Data

There are two types of input data accepted by the beat tracking system: digital audio and symbolic performance representations, which are described in turn.

There are a large number of digital audio formats currently in use, which provide various possible representations of the audio data, allowing the user to choose the bit rate and/or compression algorithm suitable for the task at hand. Considering the wide availability of software for converting between formats, we limited the input data format to uncompressed linear pulse code modulated (PCM) signals, as found on compact discs and often used in computer audio applications. The specific file format may be either the MS ".wav" format or SUN ".snd" format, both of which permit choices of sampling rates, word sizes and numbers of channels. For the results reported in this paper, single channel 16 bit linear PCM data was used, with a sampling rate of 44100 Hz. This data was created directly from compact discs by averaging the two channels of the original stereo recordings.

Two symbolic formats may be used, MIDI, the almost universal format for symbolic performance data, and the Match format, a locally developed text format combining the representation of MIDI performance data with the musical score, and associating the corresponding notes in each. The Match format facilitates the automatic evaluation of the beat tracking results relative to the musical score.

Rhythmic Events

Rhythmic information is primarily carried by the onset times of musical components (musical notes and percussive sounds). When these components have onset times which are sufficiently close together, they are heard as a composite event, which we name a rhythmic event. A rhythmic event is characterised by an onset time and a salience. Rhythmic events represent the most basic unit of rhythmic information, from which all beat and tempo information is derived. The process of deriving the rhythmic events from symbolic data is entirely different from that required for audio data, as most of the required information is directly represented in the symbolic data, whereas it must be estimated by an imperfect onset detection algorithm in the audio case.

Symbolic Data

For symbolic representations, the onset time of each musical note is encoded directly in the data. This onset time denotes the beginning of the waveform, not the perceived onset time, which usually falls slightly later, depending on the rise time of the instrument (Vos and Rasch, 1981; Gordon, 1987). With symbolic data, it is not possible to correct for the instrumental rise time, because the waveform is not encoded in the data, and thus is unknown. However, since most rhythmic information is carried by percussive instruments or other instruments with very short rise times, this does not create a noticeable problem.

The first task which must be performed is to group any approximately simultaneous onsets into single rhythmic events, and calculate the salience of each of the rhythmic events. In studies of chord asynchrony in piano and ensemble performance, it has been shown that asynchronies of 30–50ms are common, and much larger asynchronies also occur (Sundberg, 1991; Goebl, 2001). Perception research has shown that with up to 40ms difference in onset times, two tones are heard as synchronous, and for more than two tones, the threshold is up to 70ms (Handel, 1989). In this work, a threshold of 70ms was chosen for grouping near onsets into single rhythmic events.

The second task is to calculate the salience of the rhythmic events. Music theory identifies several factors which contribute to the perceived salience of a note. We particularly focus on note duration, density, dynamics and pitch in this work, since these factors are represented in the MIDI and Match file data.

The precise way in which these factors are combined to give a numerical salience value for each rhythmic event is described later.

Audio Data

Audio data requires significant processing in order to extract any symbolic information about the musical content of the signal. To date, no algorithm has been developed which is capable of reliably extracting the onset times of all musical notes from audio data. Nevertheless, it is possible to extract sufficient information in order to perform tempo induction and beat tracking. In fact, beat tracking may be improved by a lossy onset detection algorithm, as it implicitly filters out the less salient onsets (Dixon, 2000).

The onset detection method is based loosely on the techniques of Schloss (1985), who analysed audio recordings of percussion instruments in order to transcribe performances. The signal is passed through a first order high pass filter, and then smoothed to produce an amplitude envelope. The amplitude envelope is calculated as the average absolute value of the signal within a window of the signal. In this work a window of 20ms was used, with a 50% overlap, so that smoothed amplitude values were calculated at 10ms intervals. A 4-point linear regression is used to find the slope of the amplitude envelope, and a peak-picking algorithm then finds local maxima in the slope of the amplitude envelope. Local peaks are rejected if there is a greater peak within 50ms, or if the peak is below threshold (10% of the average amplitude per 10ms). The default parameter values as used in this work were determined empirically, but all values can be adjusted via command-line parameters.

For tasks such as transcription, it is important that all onsets are found, and the present time-domain technique would not suffice – it would be necessary to use a frequency domain method to detect onsets more reliably. However, for beat tracking, it is advantageous to discover just the most salient onsets, as these are more likely to correspond to beat times. In other words, the onset detection algorithm performs an implicit filtering of the true onsets, removing those with a low salience. The actual salience value for the detected peaks, used later in the beat tracking algorithm, is a linear function of the logarithm of the amplitude envelope value.

The onset detection algorithm has not been tested for music without instruments with sharp rise times; we expect that in this case a frequency domain algorithm would be required to find onsets sufficiently reliably.

Clustering of Inter-Onset Intervals

Once the rhythmic events have been determined, the time intervals between pairs of events reveal their rhythmic structure. The clustering algorithm (Figure 1) uses this data to generate a ranked list of tempo hypotheses, which are then used as the basis for beat tracking. In the literature, an inter-onset interval (IOI) is defined as the time between two successive events. We extend the definition to include compound intervals, that is intervals between pairs of events which are separated by other events. This is important in reducing the effect of any events which are uncorrelated with the beat.

Rhythmic information is provided by IOIs in the range of approximately 50ms to 2s (Handel, 1989). The clustering algorithm assigns each IOI to a cluster of similar intervals, if one exists, or creates a new cluster for the IOI if no sufficiently similar cluster exists. The clusters are characterised by the average of the inter-onset intervals contained in the cluster, which we denote the *interval* of the cluster. Similarity holds if the given IOI lies within a small distance (called the cluster width) of the cluster's interval. The cluster width is kept small (in this work 25ms) so that outlying values do not affect the cluster's interval. Nevertheless, the incremental building of the clusters means that the interval of a cluster can drift as IOIs are added.

Once cluster formation is complete, pairs of clusters whose intervals have drifted together are merged, and the clusters are ranked according to the number of elements they contain, with an adjustment for any related clusters. Two clusters are said to be related if the interval of one cluster is within the cluster width of an integer multiple of that of the other cluster. This reflects the expectation that for a cluster representing the beat, there will also exist clusters representing integer multiples and integer divisions of the beat. The adjustment is applied to the cluster's interval by calculating the average of the related clusters' normalised intervals, weighted by their scores. The ranking of the clusters is also adjusted by adding the scores of related clusters, weighted by a relationship factor $f(d)$, where d is the integer ratio of cluster intervals, given by:

$$f(d) = \begin{cases} 6 - d, & 1 \leq d \leq 4 \\ 1, & 5 \leq d \leq 8 \\ 0, & otherwise \end{cases}$$

The top ranked clusters represent a set of hypotheses as to the basic tempo of the music. At this point it is not necessary to choose between hypotheses; this choice is made later by the beat tracking algorithm. The clustering algorithm is usually successful at ranking the primary metrical level as one of the highest ranking clusters (see results section). What the clustering algorithm does not provide is any indication of the beat times. This task is performed by the beat tracking stage described in the next section.

Example

We illustrate the tempo induction stage with a short example. Consider the sequence of five events A, B, C, D, E shown on the time line in Figure 2. Below the time line, the horizontal lines with arrows represent each of the inter-onset intervals between pairs of events, and these lines are labelled with the name of the cluster to which they are assigned. Five clusters are created, denoted C1, C2, C3, C4 and C5, with C1 = $\{AB, BC, DE\}$, C2 = $\{AC, CD\}$, C3 = $\{BD, CE\}$,

Definitions

Events are denoted by $E_1, E_2, E_3, ...$
The inter-onset interval between events E_i and E_j is denoted by $IOI_{i,j}$
Clusters are sets of inter-onset intervals denoted by $C_1, C_2, C_3, ...$
$C_i.\text{interval} = \frac{\Sigma_{j,k}\{IOI_{j,k} \epsilon C_i\}}{|C_i|}$
$f(n)$ is the relationship factor
i, j, k, m, n are positive integer variables

Algorithm

```
FOR each event E_i
    FOR each event E_j
        IOI_{i,j} = |E_j.onset − E_i.onset|
        Find k such that |C_k.interval − IOI_{i,j}| < ClusterWidth is minimum
        IF k exists THEN
            C_k := C_k ∪ {IOI_{i,j}}
        ELSE
            Create new cluster C_m := {IOI_{i,j}}
        END IF
    END FOR
END FOR

FOR each cluster C_i
    FOR each cluster C_j (j ≠ i)
        IF |C_i.interval − C_j.interval| < ClusterWidth THEN
            C_i := C_i ∪ C_j
            Delete cluster C_j
        END IF
    END FOR
END FOR

FOR each cluster C_i
    FOR each cluster C_j
        IF |C_i.interval − n * C_j.interval| < ClusterWidth THEN
            C_i.score := C_i.score + f(n) * C_j.size
        END IF
    END FOR
END FOR
```

Figure 1: Algorithm for clustering of inter-onset intervals

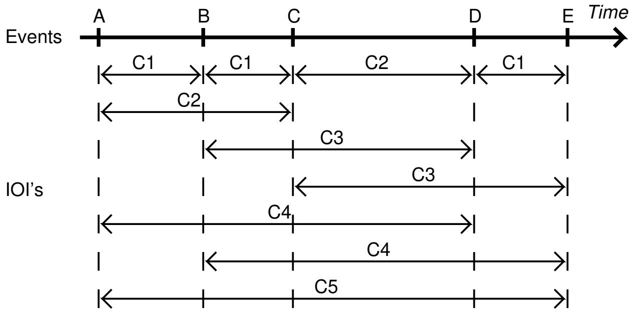

Figure 2: Clustering of inter-onset intervals (IOI's)

C4 $= \{AD, BE\}$ and C5 $= \{AE\}$. The scores for each cluster are calculated as follows:

```
C1.score = 2*3*f(1) + 2*f(2) + 2*f(3) + 2*f(4) + f(5) = 49
C2.score = 3*f(2) + 2*2*f(1) + 0 + 2*f(2) + 0         = 40
C3.score = 3*f(3) + 0 + 2*2*f(1) + 0 + 0              = 29
C4.score = 3*f(4) + 2*f(2) + 0 + 2*2*f(1) + 0         = 34
C5.score = 3*f(5) + 0 + 0 + 0 + 2*1*f(1)              = 13
```

Therefore the clusters are ranked in the following order: C1, C2, C4, C3, C5.

Beat Tracking

The Beat Tracking Architecture

The tempo induction algorithm computes the approximate inter-beat interval, that is, the time between successive beats, but does not calculate the beat times. In Figure 2, for example, one hypothesis might be that C2 represents the inter-beat interval, but it does not determine whether events A, C and D are beat times or whether B, E and the midpoint of BE are beat times. That is, tempo induction calculates the beat rate (frequency), but not the beat time (phase).

In order to calculate beat times, a multiple hypothesis search is employed, with an evaluation function selecting the hypothesis that fits the data best. Each hypothesis is handled by a beat tracking agent, which is able to predict beat times and match them to rhythmic events, adjust its hypothesis of the current beat rate and phase, create a new agent when there is more than one reasonable path of action, and cease operation if it is found to be duplicating the work of another agent.

Each agent is characterised by its *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 times selected to date by the agent. The agents can also assess their performance, by evaluating the goodness of fit of the tracking decisions to the data.

The system is designed to track smooth changes in tempo and small discontinuities; the choice of a single best agent based on its cumulative score for beat tracking the whole piece means that a piece which changes its basic tempo significantly will not be tracked correctly. In future work we plan to examine a real time approach to beat tracking, using an incremental tempo induction algorithm; at each point in time the best agent is chosen based on a combined score for its tempo and the tracking of music up to that time, thus allowing sudden changes in tracking behaviour when the previous best agent ceases to be able to track the data correctly.

The Beat Tracking Algorithm

The beat tracking algorithm is given in full in Figure 3. This algorithm is now explained in detail, with reference to the example shown in Figure 4.

Initialisation

For each hypothesis generated by the tempo induction phase, a group of agents are created to track the piece at this tempo. Based on the assumption that at least one event in the initial section of the music coincides with a beat time (normally there will be many events satisfying this condition), an agent is created for each event in the initial section, with its first beat time coinciding with that of the respective event. Using this approach, it is usually the case that there is an agent that begins with the correct tempo and phase.

The initial section, as defined by the constant StartupPeriod in the algorithm, was set to be the first 5 seconds of the music. In some cases, for example when a piece has a free-time introduction, it is possible that no agent starts with the correct tempo and phase. However, an agent with approximately the correct tempo will be able to adjust its tempo and phase in order to synchronise with the beat.

In Figure 4, a simplified example illustrates the operation of the beat tracking algorithm. The rhythmic events (denoted A, B, C, D, E and F) are represented on the time line at the top of the figure. The beat tracking behaviour of each agent is represented by the horizontal lines connecting filled and hollow circles. The filled circles represent beat times which correspond to a rhythmic event, and the hollow circles are beat times which were interpolated because no rhythmic event occurred at that time.

The figure illustrates 2 tempo hypotheses from the tempo induction stage. The faster tempo, with an inter-beat interval approximately equal to the time interval between events A and B, is the tempo hypothesis of Agent1. The slower tempo, with inter-beat interval approximately equal to the interval between C and D, is the tempo hypothesis of the other agents. For the initialisation stage, assume that only events A and B are in the initial section, and then nominally it is expected that 4 agents would be created, one for each (tempo

```
Initialisation
FOR each tempo hypothesis T_i
    FOR each event E_j such that E_j.onset<StartupPeriod
        Create a new agent A_k
        A_k.beatInterval:= T_i
        A_k.prediction:= E_j.onset +T_i
        A_k.history:= [E_j]
        A_k.score:= E_j.salience
    END FOR
END FOR
Main Loop
FOR each event E_i
    FOR each agent A_j
        IF E_i.onset−A_j.history.last > TimeOut THEN
            Delete agent A_j
        ELSE
            WHILE A_j.prediction+Tol_post < E_i.onset
                A_j.prediction := A_j.prediction+A_j.beatInterval
            END WHILE
            IF A_j.prediction+Tol_pre ≤ E_i.onset≤ A_j.prediction+Tol_post THEN
                IF |A_j.prediction −E_i.onset| > Tol_inner
                    Create new agent A_k := A_j
                END IF
                Error:= E_i.onset−A_j.prediction
                A_j.beatInterval:= A_j.beatInterval+Error/CorrectionFactor
                A_j.prediction := E_i.onset+A_j.beatInterval
                A_j.history:= A_j.history+E_i
                A_j.score := A_j.score+(1−relativeError/2) ∗ E_i.salience
            END IF
        END IF
    END FOR
    Add newly created agents
    Remove duplicate agents
END FOR
Return the highest scoring agent
```

Figure 3: Beat Tracking Algorithm

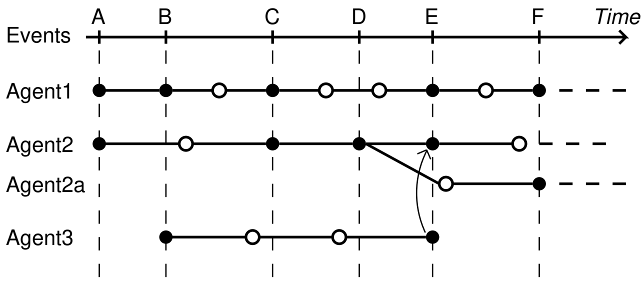

Figure 4: Beat tracking by multiple agents

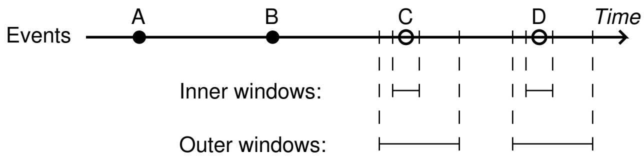

Figure 5: Tolerance windows for beat tracking

hypothesis, initial event) pair. In fact, only 3 agents are created, Agent1 with the faster tempo, and Agent2 and Agent3 with the slower tempo. This is because the system recognises that a fast tempo agent beginning on event B would be redundant, since Agent1 also predicts B as a beat time.

Main Loop

In the main body of the beat tracking algorithm, each event is processed in turn by allowing each agent the opportunity to consider the event as a beat time. Each agent has a set of predicted beat times, which are generated from the most recent beat time by adding integer multiples of the agent's current inter-beat interval. These predicted beat times are surrounded by two-level windows of tolerance, which represent the extent to which an agent is willing to accept an alteration to its prediction (see Figure 5). The inner window, set at 40ms either side of the predicted beat time, represents the deviations from strict metrical time which an agent is willing to accept. The outer window, with a default size of 20% and 40% of the inter-beat interval respectively before and after the predicted beat time, represents changes in tempo and/or phase which an agent will accept as a possibility, but not a certainty. The asymmetry reflects the fact that expressive reductions in tempo are more common and more extreme than tempo increases (Repp, 1994).

There are three possible scenarios when an agent processes an event, illustrated in Figure 4. The simplest case is when the event falls outside the tolerance windows of predicted beat times, and the event is ignored. For example, in the figure, event D is ignored by Agent1 and Agent3, and event B is ignored by Agent2. The second case is when the event falls in the inner tolerance window of a predicted beat time, so that the event is accepted as a beat time. Event B with Agent1 and events D and E with Agent2 are examples of this case. If the event is not in the first predicted beat window, then the missing beats are interpolated by dividing the time interval into equal durations, as shown by the hollow circles in the figure. For example, this occurs at events C, E and F for Agent1 and event E for Agent3. The agent's tempo is then updated by adjusting the tempo hypothesis by a fraction of the difference between the predicted and chosen beat times, and the score is updated by adding the salience of the event, which is also adjusted (downward) according to the difference between predicted and chosen beat times. The third and most complex case is when the event falls in one of the outer tolerance windows. In this case, the agent accepts the event as a beat time, but as insurance against a wrong decision, also creates a new agent that does not accept the event as a beat time. In this way, both possibilities can be tracked, and the better choice is revealed later by the agents' final scores. This is illustrated in Figure 4 at event E, where Agent2 accepts the event and at the same time creates Agent2a to track the possibility that E is not a beat time.

Complexity Management

Since each agent's future beat tracking behaviour is entirely based on its current state (tempo and phase) and the input data, any two agents that agree on the tempo and phase will exhibit identical behaviour from that time on, wasting computational resources. In order to increase efficiency, the duplicate agents are removed at the earliest opportunity. Theoretically, such an operation should make no difference to the results produced by the system, but one complication arises, that the tempo and phase variables are continuous, so equality is too strong a condition to use in comparing agents' states. Therefore thresholds of approximate equality were chosen, conservatively, at 10ms for the inter-beat interval (tempo) difference and 20ms for the predicted beat time (phase) difference.

When the decision to remove a duplicate agent is made, it is important to consider which agent to remove. The agents have different histories (otherwise the duplicate would have been removed sooner), and therefore different evaluation scores. Since the evaluation is calculated based only on the relationship between predicted beat times and the rhythmic events, and not on any global measure of consistency, it is always correct to retain only the agent with the higher current score, since it will also have the higher total score at the end of beat tracking. In Figure 4, Agent3 is deleted after event E, because it agrees in tempo and phase with Agent2. This is indicated by the arrow between Agent3's and Agent2's event E.

Assessment

The comparison of the agents' beat tracking is based on three factors: how evenly spaced the chosen beat times are, how often the beat times match times of rhythmic events, and the salience of the matched rhythmic events. As stated above, the evenness of beat times is not calculated via a global measure, but from the local agreement of predicted beat times and rhythmic events.

For each beat time at which a rhythmic event occurs, a fraction of the event's salience is added to the agent's score. The fraction is calculated from the relative error of the predicted beat time; that is the difference in predicted and chosen beat times, divided by the window half-width, which is then halved and subtracted from 1. This gives a score between 0.5 and 1.0 times the salience for each event. The beat tracking algorithm then returns the agent with the greatest score.

Estimating Musical Salience

In earlier work (Dixon, 2000), where it was assumed that expressive variations in tempo were minimal, it was found that no specific musical knowledge was needed by the system in order to perform beat tracking successfully. That is, by searching for a regularly spaced subset of the events with few gaps and little variation in the spacing, the system was able to find the times of beats. As the system was tested with more expressive musical examples, it was found that the search had to allow greater variation in the beat spacing, which led to an increase in the number of choices, and therefore the number of agents. Without further musical knowledge, it was not possible for the system to choose correctly between the many possible musical interpretations offered by the various agents. In this section we describe how knowledge of musical salience was added to the system in order to direct the system to the more plausible musical interpretations (Dixon and Cambouropoulos, 2000).

When comparing beat tracking results for audio and MIDI versions of the same performances, it was discovered that the onsets extracted from audio data, although unreliable, provided a better source of data for beat tracking than the onset times extracted without error from the MIDI data. It was postulated that this was due to an implicit filtering of the data. That is, only the more significant onsets had been extracted; the less salient onsets remained undetected, and had no influence on the beat tracking system. The reason this was advantageous is that the more salient events are more likely to correspond to beat times than the less salient events, and hence the search had been narrowed by the onset detection algorithm.

This hypothesis is tested by incorporating knowledge of musical salience into the system, and measuring the corresponding performance gain. This is done using MIDI data, since parameters such as duration, pitch and volume, which are important determiners of salience, are all directly available from the data.

Observations From Music Theory

The tendency for events with greater perceptual salience to occur in stronger metrical positions has been noted by various authors (Longuet-Higgins and Lee, 1982; Lerdahl and Jackendoff, 1983; Povel and Essens, 1985; Lee, 1991; Parncutt, 1994). The factors influencing perceived salience have also been studied, although no model has been proposed which predicts salience based on combinations of factors, or which gives more than a qualitative account of the effects of parameters.

Lerdahl and Jackendoff (1983) classify musical accents into three types: phenomenal accents, which come from physical attributes of the signal such as amplitude and frequency; structural accents, which arise from perceived points of arrival and departure such as cadences; and metrical accents, points in time which are perceived as accented due to their metrical position. We only concern ourselves with the first type of accent here, since the higher level information required for the other types is not available to the beat tracking system. Lerdahl and Jackendoff list the following types of phenomenal accent (which they consider incomplete): note onsets, sforzandi, sudden dynamic or timbral changes, long notes, melodic leaps and harmonic changes. However, they give no indication as to how these factors might be compared or combined, either quantitatively (absolute values) or qualitatively (relative strengths).

In other models of meter perception, the main factor determining salience is note duration, which is usually taken to mean inter-onset time rather than perceived or physical sounding time of a note. For example, Povel and Essens (1985) describe three scenarios in which a note receives a perceived accent relative to other notes of identical pitch, amplitude envelope and physical duration. In their model, notes which are equally spaced in time form groups, subject to the condition that there are no notes outside the group which are closer to any note in the group. Then the notes which receive accents are: notes which do not belong to any group, the second note of any group of two notes, and the first and last note of any group of three or more notes. All of these accents, except the first note in groups of 3 or more, fall on notes with a long duration relative to their context. Other authors (Longuet-Higgins and Lee, 1982; Parncutt, 1987; Lee, 1991) also state that longer notes tend to be perceived as accented.

All of the models based on inter-onset times were developed in a monophonic context, and are difficult to interpret when considering the polyphonic context of most musical performances. For example, one would intuitively expect that a long note in a melody part should not lose its accent due to notes in the accompaniment which follow shortly after the onset of the melody note. However, if we were to observe the inter-onset times only, this would be the result of such models. To adapt to polyphonic music, a model of auditory streaming (Bregman, 1990) could be applied, so that interactions between streams are removed, but that raises the difficult question of how the metrical perception of the various streams could be combined into a single percept. A simpler approach is to use the physical durations of notes rather than inter-onset times, even though these are difficult to estimate from audio data.

Combining Salience Factors

For this work, the factors chosen as determiners of musical salience were note duration, simultaneous note density, note amplitude and pitch, all of which are readily available in the symbolic representation of the performance. The next task was the combination of these factors into a single numerical value, representing the overall salience of each rhythmic event. None of the above-mentioned work has investigated or proposed a method of combining salience, so it was decided to test two possible salience models by their influence on the beat tracking results. The two models presented are a linear combination $s_{add}(d, p, v)$ of duration d, pitch p and amplitude v, and a nonlinear, multiplicative function $s_{mul}(d, p, v)$ of the same parameters. A threshold function is used to restrict the values of p to a limited range, and constants are used to set the relative weights of the parameters. The salience of a group of notes combined as a single rhythmic event was calculated using the longest duration, the sum of the dynamic values and the lowest pitch of all the notes in the group.

The salience functions were defined as follows:

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

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

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

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

The following weights for the parameters were determined empirically:

$$\begin{aligned} c_1 &= 300 \\ c_2 &= -4 \\ c_3 &= 1 \\ c_4 &= 84 \\ p_{min} &= 48 \\ p_{max} &= 72 \end{aligned}$$

These values make duration the most significant factor, with dynamics and pitch being useful primarily to distinguish between notes of similar duration. This salience calculation is not sufficient for all possible MIDI files. In particular, it would not work well for non-pitched percussion such as drums, where the salience is clearly not so strongly related to duration. Such instruments should be treated separately as a special case of the salience calculation.

We will now describe the task of evaluating the beat tracking system, before presenting the results in the following section.

Evaluation

We know of no precise definition of beat for expressively performed music. The definitions given in the first section of the paper do not uniquely define the relevant quantities or give a practical way of calculating them from performance data. In this section, we describe the difficulties with formalisation and evaluation of beat tracking models and systems, and then describe the evaluation methodology used in this work.

Problems with Evaluation

The tempo induction and beat tracking algorithms described in this paper were not designed for any particular style of music. They operate in the same way for classical music played from a score as for improvised jazz. This immediately creates a difficulty for evaluation, in that we cannot assume that a musical score is available to define which notes coincide with beats and which do not. It is laborious (but possible) for a trained musician to transcribe a performance in enough detail to identify which notes occur on the beat. But even when a score or transcription exists, there is no error-free, automatic way to associate the score timing with the performance timing of audio data. That is, given a score, we can establish which notes in the score correspond to beat times, but we do not know the absolute times of score notes, and the beat tracking system returns its results as absolute times. The accurate extraction of onset times in polyphonic music is an unsolved problem, so we are forced to rely on hand-labelled or hand-corrected data for evaluation purposes. This issue is discussed in much greater detail by Goto and Muraoka (1997a), who also use a similar approach to that which we describe below.

Further problems exist which apply equally to audio and symbolic performance data. There are many cases in which the beat is not uniquely defined by performance data. In the simplest case, consider a chord which occurs on a beat according to the score. In performance, it has been observed that the notes of a chord are not necessarily played simultaneously, and these asynchronies may be random or systematic (e.g. melody lead) (Palmer, 1996; Repp, 1992; Goebl, 2001). The problem is more pronounced in ensemble situations, where there are often systematic timing differences between performers (participatory discrepancies) (Keil, 1995; Prögler, 1995; Gabrielsson, 1999).

It is not obvious how the beat time should then be defined, whether at the onset of the first note of the chord, or of the last, the lowest, or the highest, or at the (weighted) average of the onset times, or alternatively expressed as a time interval. The problem is exacerbated in situations where timeless events are notated, such as grace notes and arpeggios. Performers may interpret grace notes in different ways, sometimes to precede the beat, and sometimes to coincide with the beat. In some cases hearers do not even agree on which interpretation they think the performer applied.

It is reasonable to question whether beats necessarily correspond to event times. A beat percept induced by previous events might be stronger than that of an event which nominally coincides with the beat, and in this case the event is perceived as anticipating or following the beat rather than defining the beat time.

The point of this section is to show that some subjective judgement is necessary in evaluating the results produced by a beat tracking program. Our aim is to keep this subjectivity to a minimum, and define an evaluation methodology which gives repeatable results. We distinguish between two approaches to beat tracking: predictive (perceptual) and descriptive beat tracking. An algorithm is said to be *causal* if its output at time t depends only on input data for times $\leq t$. Predictive beat tracking models perception using causal algorithms which predict listeners' expectations of beat times, necessarily smoothing the performance expression. Descriptive beat tracking models musical performance non-causally, giving beat times with a more direct correspondence to the performance data than predictive beat tracking. For a perceptual model of beat tracking, detailed perceptual studies are required to resolve the issues discussed above. A descriptive approach, as taken in the current work, allows the data to define beat times more directly.

Informal Evaluation: Listening Tests

Before describing the formal evaluation methods, we briefly describe an informal, subjective method used to ascertain whether the results appear to make musical sense. We have already expressed the importance of objective evaluation, but since we are dealing with the subjective medium of music, it is also important that the evaluation is musically plausible. This is done by creating an audio file consisting of the original music plus a click track – a percussion instrument playing on the beat times estimated by the beat tracking algorithm. The click track is synthesised from a sample of a chosen percussion instrument, and can be added as a separate channel (for separate volume control) or mixed onto the same channel as the music (for use with headphones, to avoid streaming due to spatial separation).

This is the least precise but perhaps the most convincing way to demonstrate the capabilities of the system. Apart from being subjective, a further problem with listening tests is the amount of time required to perform testing. When it is desired to systematically test the effects of a series of changes to the system, it is impractical to listen to every musical example each time. It is also difficult to compare the number and types of errors made by different versions of algorithms or by the same algorithm with different parameter values.

Beat Labelling

For pre-recorded audio data it is necessary to perform the labelling of beat times subjectively. This is done using software which provides both audio and visual feedback, so that beat times can be selected based on both the amplitude envelope and the sound. For popular music, it is sufficient to interpolate some of the beat times in sections of effectively constant tempo, thus greatly accelerating the beat labelling process. This technique was also used to determine beat times which could not be accurately estimated by other methods.

In the case of the symbolic performance data, it had already been matched to a digital encoding of the musical scores, thus allowing automatic evaluation of the system by comparing the beat tracking results (the reported beat times) with the onset times of events which are on the beat according to the musical score (the notated beat times). For beats with multiple notes, we took the beat time to be the interval from the first to the last onset of events which are nominally on the beat.

We refer to the beats calculated in either of these ways as the "correct" beat times, and use them as the basis for evaluation.

Evaluation Formula

The reported beat times are then matched with the correct beat times by finding the nearest correct beat time and recording a match if they are within a fixed tolerance, and no match if the tolerance is exceeded. This creates three result categories: matched pairs of reported and correct beat times, unmatched reported beat times (false positives) and unmatched correct beat times (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 note simultaneity, which is 70ms either side of the beat time. 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 (Cemgil et al., 2001).

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

Metrical Levels

The final aspect of evaluation is that of metrical levels. As discussed earlier, it is possible to track beats at more than one level, and different listeners will feel natural tapping along at different levels. 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 (Desain and Honing, 1999). Various authors report preferred inter-beat intervals around 500ms to 700ms (Parncutt, 1987, 1994; Todd and Lee, 1994; van Noorden and Moelants, 1999), with possible inter-beat intervals falling within the range of 200ms to 1500ms (van Noorden and Moelants, 1999). In performances of 13 Mozart piano sonatas (see results section), the inter-beat interval at the notated metrical level ranged from 200ms to 2000ms. Therefore it is clear that the metrical levels of the perceived beat and the notated beat are not necessarily the same.

The formula given in the previous subsection gives a reasonable assessment of correctness only when the metrical level of beat tracking equals the metrical level of assessment. When the metrical levels fail to coincide, it is tempting to interpolate or decimate the labelled beat times in order to bring the levels into agreement. However, this is incorrect, because it doesn't take phase into account. It is generally easier to track music at lower (i.e. faster) metrical levels, and harder at higher (slower) metrical levels, because the likelihood of phase errors is much higher at the higher levels. An agent which tracks popular music in 4/4 time at the half note level is much more likely to be 50% out of phase (tracking beats 2 and 4) than an agent at the preferred quarter note level (tracking every off-beat). We revisit this issue in the results section.

Implementation and Results

Implementation Details

The beat tracking system is implemented on a Linux platform in C++, and consists of approximately 7000 lines of code in 18 classes. On a 500MHz Pentium computer, audio beat tracking takes under 20% of the length of the music (i.e. a 5 minute song takes less than one minute to process) and beat tracking of symbolic data is much faster, taking between 2% and 10% of the length of the music, depending on the note density.

Audio Data

The audio data used in these experiments is listed in Table 1, which also provides the letters used to identify the pieces in the text. The pieces were chosen to represent various styles of music, and are listed in order of subjective beat tracking difficulty.

The first 3 pieces (I,D,U) are standard modern pop/rock songs, characterised by a very steady tempo, which is clearly defined by simple and salient drum patterns, similar to the data used in the early audio beat tracking work of Goto and Muraoka (1995). In these songs the performed beat is very regular, with only small deviations from metrical timing. It is assumed that this is the simplest type of data for beat tracking.

ID	Title (Artist)	CD (Number)	Style (Date)
I	I Don't Remember A Thing (Paul Kelly and the Coloured Girls)	Under the Sun (Mushroom CD 53248)	Pop/rock (1987)
D	Dumb Things (Paul Kelly and the Coloured Girls)	Under the Sun (Mushroom CD 53248)	Pop/rock (1987)
U	Untouchable (Paul Kelly and the Coloured Girls)	Under the Sun (Mushroom CD 53248)	Pop/rock (1987)
S	Superstition (Stevie Wonder)	Talking Book (Motown 37463-03192-9)	Motown (1972)
Y	You Are The Sunshine of My Life (Stevie Wonder)	Talking Book (Motown 37463-03192-9)	Motown (1972)
O	On A Night Like This (Bob Dylan)	Planet Waves (Columbia CD 32154)	Country (1974)
P	Piano Sonata in C (Movt 3, Sec 1) (Wolfgang Mozart)	[Synthesised from MIDI]	Classical (1775)
R	Rosa Moreña (João Gilberto Trio)	Samba and Bossa Nova (Jazz Roots CD 56046)	Bossa nova (1964)
M	Michelle (Béla Fleck and the Flecktones)	Flight of the Cosmic Hippo (Warner 7599-26562-2)	Jazz swing (1991)
J	Jitterbug Waltz (James Morrison)	Snappy Doo (WEA 9031-71211-1)	Jazz waltz (1990)

Table 1: Details of audio data used in experiments

The next 2 pieces (S, Y) have a Motown/Soul style, characterised by more syncopation, greater tempo fluctuations (5-10% in these examples), and more freedom to anticipate or lag behind the beat. It is expected that these examples are more difficult for a beat tracking system, but only of medium difficulty.

The remaining pieces were chosen as having particular characteristics which make beat tracking difficult. The Bob Dylan song (O) is made difficult by the fact that the drums are not prominent, and there is a much lower correlation between the beat and the events than in the other styles, due to his idiosyncratic style of singing and playing against the rhythmic context.

The classical piece (P), the first section of the third movement of Mozart's Piano Sonata in C major (KV279), was synthesised from the MIDI data used in the beat tracking experiments using symbolic input data. This piece was chosen as an example of a piece with significant tempo fluctuations.

The next piece (R) is a live bossa nova performance with syncopated guitar and vocals, and very little percussion to indicate beat times. Sections of this piece are difficult for humans to beat track.

The two jazz pieces (M,J) were chosen for their particularly complex, syncopated rhythms, which are difficult even for musically trained people to follow. These pieces also provided examples of a different meter and swing eighth notes.

Tempo Induction Results

The tempo induction algorithm for audio data was tested on short segments of the above pieces to determine how reliably the ranked tempo hypotheses agreed with the measured tempo. (Piece P was not available at the time of this experiment.)

The tempo induction algorithm was applied to segments of 5, 10, 20 and 60 seconds duration of each piece, starting at each 1 second interval from the beginning of the piece. A tempo hypothesis was considered correct if the inter-beat interval was within 25ms of the measured inter-beat interval. Table 2 shows the results for 10 second excerpts, showing the position that the correct tempo hypothesis was ranked by the tempo induction algorithm. The results are presented as percentages of the total number of segments. The cumulative sums of rankings are shown in Figure 6.

Table 3 shows the effect of length of the segments on tempo induction. Each column represents a different segment size, and the entries show the percentage of segments for which the correct tempo was included in the top 10 ranked clusters. Even with 5 second segments, the tempo induction algorithm is quite reliable, and reliability increases with segment size.

The tempo induction stage provides a solid foundation for the beat tracking agents to work from, and is robust even for highly syncopated pieces of music. We now present the beat tracking results, first for audio data, and then the MIDI-based experiments.

ID	Ranking of Correct Tempo										Sum of
	1	2	3	4	5	6	7	8	9	10	top 10
I	85.9	13.1	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
D	91.0	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
U	96.3	3.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
S	31.2	40.5	18.6	7.0	0.5	0.0	0.0	1.4	0.5	0.0	99.5
Y	86.7	11.7	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	99.2
O	83.9	12.9	2.4	0.0	0.0	0.8	0.0	0.0	0.0	0.0	100.0
R	58.7	18.2	7.6	3.6	1.3	2.7	0.9	0.9	0.0	0.9	94.7
M	16.9	25.9	16.9	12.9	4.7	6.5	1.8	1.8	1.8	1.1	90.3
J	61.5	13.4	9.2	3.1	4.2	0.4	0.4	0.4	0.0	0.0	92.4

Table 2: Beat induction of 10 second segments of songs. All figures are percentages of the total number of segments

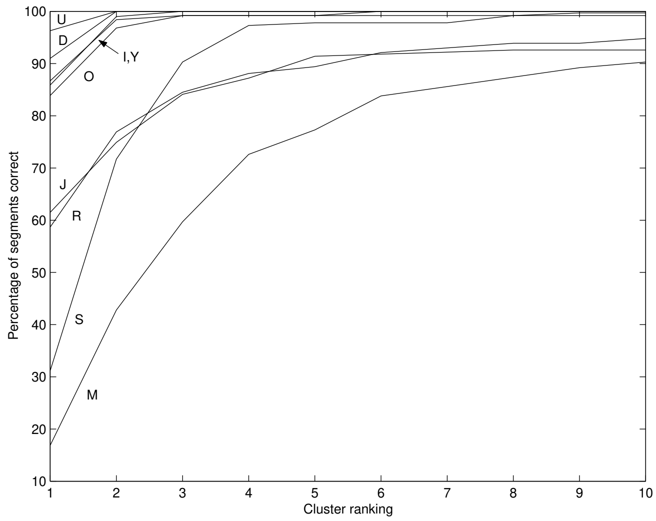

Figure 6: Tempo induction results for 10 second segments shown as cumulative sums of percentages from Table 2

Piece	Correct Tempo in Top 10			
ID	5s	10s	20s	60s
I	100.0	100.0	100.0	100.0
D	100.0	100.0	100.0	100.0
U	100.0	100.0	100.0	100.0
S	93.5	99.5	99.5	100.0
Y	96.9	99.2	100.0	100.0
O	95.2	100.0	100.0	100.0
R	84.9	94.7	100.0	100.0
M	84.9	90.3	95.3	95.9
J	79.2	92.4	97.6	99.1

Table 3: Effects of segment length on tempo induction: for each segment length, the percentage of segments with the correct tempo in the top 10 hypotheses is shown

ID	Tempo range	Meter	Results
I	139-142	4/4	100%
D	151-154	4/4	100%
U	145-146	4/4	100%
S	96-104	4/4	96%
Y	127-136	4/4	92%
O	136-140	4/4	79%
P	120-150	2/4	90%
R	128-134	4/4	95%
M	180-193	3/4	92%
J	155-175	3/4	77%

Table 4: Beat tracking test details and results

Audio Beat Tracking

In Table 4, we present the results for beat tracking of the audio data. The rightmost column of the table indicates the percentage of beat times which were calculated correctly by the beat tracking system, a simpler measure of system performance than that given in the previous section. This does not indicate the nature of the errors made during beat tracking. For each of the songs listed, the tempo was estimated correctly. That is, the highest scoring agent was an agent with the correct tempo hypothesis. The results column can be considered as the percentage of the song for which this agent was in phase with the beat.

The pieces expected to be easy to beat track (I,D,U) were tracked without error. The medium difficulty pieces (S,Y) were tracked with some phase errors, from which the agent recovered quickly. In piece (O), there were two sections in which the agent lost synchronisation and tracked the off-beats (i.e., it continued at the correct tempo but half a beat out of phase), but eventually recovered to the correct phase. Even in this case, 79% of the piece was tracked correctly. In (P), the agent lost synchronisation several times, due to large tempo variations and the agent's lack of musical knowledge for distinguishing between beats and off-beats. In this case, the errors amounted to only 10% of the piece. Piece (R) was expected to be difficult because of the syncopation, but it was tracked correctly except for a few phase errors, from which it recovered within a few beats. In the case of the jazz piece (M) the results were surprisingly good (92%), probably due to the fast tempo, which reduces the likelihood of phase errors. The second jazz piece (J) scored lowest (77%), but still was correctly tracked for more than three quarters of the piece. An average person would probably perform no better on the last two pieces.

MIDI Beat Tracking

A series of experiments was performed to test the hypothesis that musical salience is useful in guiding the beat tracking process. In experiment 1, the performance of the beat tracking algorithm without the use of salience was established. In terms of the evaluation function used for the agents, a constant value was used for the salience of all rhythmic events. This set the base level for the measurement of the performance gain due to the salience calculations.

In the second experiment, the behaviour of the audio beat tracking system was simulated, by applying the salience function $s_{add}(d, p, v)$ to the events and deleting those events with a salience below a fixed threshold value. The beat tracking algorithm was then applied to the remaining events, but using a constant salience value in the evaluation functions as for experiment 1.

The third experiment tested the use of the salience values in the agents' evaluation functions directly. In this case the agents accumulated a progressive score consisting of the sum of adjusted salience values for the events tracked by the agent. The two different salience functions were tested: experiment 3a tested the non-linear function $s_{mul}(d, p, v)$ and experiment 3b tested the linear function $s_{add}(d, p, v)$.

Relative tempo	Number of sections			
	Exp1	Exp2	Exp3a	Exp3b
0.5	10	9	10	10
1.0	121	143	146	137
1.5	16	0	1	5
2.0	40	40	41	42
3.0	4	3	3	4
4.0	23	23	20	22
other	8	2	1	2
fail	0	2	0	0

Table 5: Beat tracking rates relative to the primary metrical level

The data consisted of 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 first set of results shows the tapping rate chosen by the highest scoring agent, relative to the rate of the primary metrical level. For almost all sections, a musically plausible metrical level was chosen, with slightly worse performance in experiment 1 where salience was not used. Table 5 shows the number of sections which were tracked at various multiples of the tempo of the primary metrical level. The rows labelled *other* and *fail* represent the cases where the tempo was not related to the primary metrical level, and when beat tracking failed to produce a solution at all, respectively. The majority of sections were tracked at the notated level or otherwise 2 or 4 times the notated level.

To test the correspondence of these results to human perception of beat, the results were compared with the metrical levels chosen by a human (the author) tapping in time with each of the sections. In the majority of cases (155), there were two possible metrical levels chosen for the beat; in 49 cases, three levels were considered possible, and in the remaining 18 cases, only one rhythmic level was considered reasonable. Table 6 shows the relationship between the system's choices and the "perceptually reasonable" metrical levels. The errors (cases where the system's tapping rate was not one of those chosen as a reasonable rate) are divided into three types: double tempo errors, where the system chose a rate of double the fastest human tapping rate, which in each case was a musically possible alternative; half tempo errors, where the system chose to tap in quarter notes, but the piece was in compound time, which is musically incorrect; and other errors, which were mostly the system tapping in dotted quarter notes for pieces in simple time, which again is musically wrong.

Table 7 shows the results of evaluating the beat tracking of the sections which were tracked at the primary metrical level (which in each case also corresponded to a perceptually reasonable metrical level for the beat). Evaluation

	Number of sections			
	Exp1	Exp2	Exp3a	Exp3b
correct	170	194	200	189
error: double	17	15	10	15
error: half	10	9	10	10
error: other	25	4	2	8

Table 6: Correspondence of beat tracking rates relative to human tapping rates

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%	

Table 7: Evaluation of beat tracking at the primary metrical level

was performed using the formula given in the previous section. For each experiment we show the number of sections (n) and the percentage of sections which achieved various minimum scores.

Experiment 1 gives the base level performance of the system without musical knowledge. Approximately one third of the sections were tracked correctly, with a further third scoring over 70%. At the bottom of the table, the results for each experiment are summarised in a single value, the weighted average of the beat tracking evaluation results, weighted by the number of beats. For experiment 1, without musical salience, the system found 75% of beat times.

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 more difficult for the beat tracking system to determine some of the beat times. Nevertheless, the net result of this experiment was positive, with 88.8% of the sections scoring over 70%, and a total of 85% of the beats being found.

The third experiment shows a further 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 $s_{mul}(d, p, v)$. For these two functions the weighted averages were 91.1% and 88.5% respectively, which shows a clear performance gain due to the inclusion of musical knowledge in the form of salience calculations in the system.

Discussion and Conclusion

We have described a beat tracking system which analyses musical data, detects the onsets of rhythmic events and their salience and then determines the tempo and beat times using a multiple hypothesis search. The system successfully calculates the tempo for most musical situations, and tracks the beat with occasional phase errors. The system's performance is robust, in that it recovers from errors and resumes correct tracking quickly, a capability reported to be lacking in earlier systems (Dannenberg, 1991). Informal listening tests demonstrate that the system captures some part of human musical ability in the way that it tracks tempo variations.

One design goal for the system was that it be as general purpose as possible, that is, not focussed on any particular style of music. To date, the system has been tested on a large corpus of expressively performed classical music, and a range of Western popular music, with positive results. We speculate that the system will work equally well with other styles of music, subject to the following two restrictions. First, the system assumes that the music has a beat, with no large discontinuities; it does not answer the question of whether or not a piece of music has a beat. Second, although the tempo induction and beat tracking algorithms are independent of the instrumentation, the calculation of rhythmic events is not. For MIDI input, the salience function lacks a special case computation of the salience of drum sounds. For audio input, the onset detection algorithm assumes the presence of notes with a sharp attack, for example, piano, guitar or drums; in the absence of such instruments it is likely that a frequency domain onset detection algorithm would need to be developed. Nevertheless, the results presented in the previous section indicate that beat tracking accuracy is not dependent on musical style directly, but rather on rhythmic complexity (Dixon, 2001).

There are a large number of parameters which can be varied in order to tune the behaviour of the system. Most parameter values and knowledge used in the system are quite low-level, being derived from knowledge of human perception. The system was designed to work autonomously, and the results which have been presented were generated without fine-tuning the parameters (with the exception of the constants used in the salience calculation). Many of the parameter values are not critical to the behaviour of the system, in a global (average) sense, although they may lead to different results on a local level. The weights attached to factors which support the various competing hypotheses are necessarily somewhat arbitrary. In complex music there are competing rhythmic forces, and higher level knowledge of the musical structure makes the correct interpretation clear to a human listener. The beat tracking agents do not make use of such high level knowledge, and therefore their decisions are influenced by more arbitrary factors such as the numerical values of parameters.

Despite the beat tracking system's lack of higher level musical knowledge, such as notions of off-beats or expected rhythmic patterns, it still exhibits an apparent musical intelligence, which emerges from patterns and structure *in the data*, rather than from high-level knowledge or reasoning (Brooks, 1991).

This makes the system simple, robust and general. In order to disambiguate more difficult rhythmic patterns, it was shown that the use of simple musical knowledge in addition to the timing of events can be used to improve performance considerably. Further improvement can be achieved, at the expense of generality, by programming high-level knowledge of stylistic expectations.

There are many avenues open for further work, in the form of applications, improvements and further investigations. One application which is currently under development is the implementation of an interactive beat tracking system that allows correction of errors via a graphical interface, and restarting the beat tracking from any point in the data (Dixon et al., 2001). This is being developed as a tool for use in the analysis of expressive performances, as it can generate data such as tempo curves semi-automatically.

It is clear that beat tracking can be improved if the system is given information about the music it is tracking. In the performance analysis application, for example, the score is usually available, so by modifying the agents' evaluation to favour interpretations which match the patterns expected from the score, a fully automatic timing analysis system could be created.

Conversely, another useful tool would be that of a score extraction system for MIDI performance data. This would involve extending the system to perform quantisation of all rhythmic events, as well as other tasks such as note spelling, part separation and induction of the key and time signatures (Cambouropoulos, 1996, 2000).

A further extension is the idea of converting the system to operate in real time, such as the systems of Goto and Muraoka (1995, 1998, 1999). The current approach is fast enough for a real time implementation, but the algorithm is not causal and would need to be modified in order to create a real time system.

This work also suggests several possible modifications to the system. Currently, the agents are assessed wholistically, that is, on the basis of their performance for the complete input data. The agents could improve their results by self-analysis, finding any inconsistencies or times where the evaluation function is low, and searching for better solutions based on the high-scoring parts of their results. Tempo induction is also performed on a larger scale than necessary, and the two algorithms could be modified so that tempo induction is calculated on a more local level and communicated to the agents as they perform beat tracking. Some of the parameter values were chosen arbitrarily, and the system could be improved by analysing musical data and extracting parameter values that correspond better to performance data.

Finally, although the system is not a model of human perception, a comparison between the correctness and accuracy of the system and of human subjects would be interesting, and would shed light on the more difficult evaluation issues, perhaps leading to a clearer understanding of beat tracking. It is not known what the limit of beat tracking performance is; it would be interesting to compare the current results with human beat tracking ability on the same pieces.

Acknowledgements

This research is part of the project Y99-INF, sponsored by the Austrian Federal Ministry of Education, Science and Culture (BMBWK) in the form of a START Research Prize. The BMBWK also provides financial support to the Austrian Research Institute for Artificial Intelligence. The author wishes to thank Gerhard Widmer, Emilios Cambouropoulos and Werner Goebl for suggestions and contributions to this work; Roland Batik and also the L. Bösendorfer Company, Vienna, particularly Fritz Lachnit, for the Mozart performance data; and the anonymous reviewers for suggested improvements to this manuscript.

References

Allen, P. and Dannenberg, R. (1990). Tracking musical beats in real time. In *Proceedings of the International Computer Music Conference*, pages 140–143. International Computer Music Association, San Francisco CA.

Bregman, A. (1990). *Auditory Scene Analysis: The Perceptual Organisation of Sound*. Bradford, MIT Press.

Brooks, R. (1991). Intelligence without reason. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 569–595.

Cambouropoulos, E. (1996). A general pitch interval representation: Theory and applications. *Journal of New Music Research*, 25(3):231–251.

Cambouropoulos, E. (2000). From MIDI to traditional musical notation. In *Proceedings of the AAAI Workshop on Artificial Intelligence and Music: Towards Formal Models for Composition, Performance and Analysis*, pages 19–23. AAAI Press.

Cariani, P. (2001). Temporal codes, timing nets and music perception. *Journal of New Music Research*. To appear.

Cemgil, A., Kappen, B., Desain, P., and Honing, H. (2001). On tempo tracking: Tempogram representation and Kalman filtering. *Journal of New Music Research*. To appear.

Clarke, E. (1988). Generative principles in music performance. In Sloboda, J., editor, *Generative Processes in Music*, pages 1–26. The Clarendon Press.

Clarke, E. (1999). Rhythm and timing in music. In Deutsch, D., editor, *The Psychology of Music*, pages 473–500. Academic Press.

Dannenberg, R. (1991). Recent work in real-time music understanding by computer. *Proceedings of the International Symposium on Music, Language, Speech and Brain*, pages 194–202.

Desain, P. (1992). A (de)composable theory of rhythm perception. *Music Perception*, 9:439–454.

Desain, P. (1993). A connectionist and a traditional AI quantizer: Symbolic versus sub-symbolic models of rhythm perception. *Contemporary Music Review*, 9:239–254.

Desain, P. and Honing, H. (1989). Quantization of musical time: A connectionist approach. *Computer Music Journal*, 13(3):56–66.

Desain, P. and Honing, H. (1999). Computational models of beat induction: The rule-based approach. *Journal of New Music Research*, 28(1).

Dixon, S. (2000). A lightweight multi-agent musical beat tracking system. In *PRICAI 2000: Proceedings of the Pacific Rim International Conference on Artificial Intelligence*, pages 778–788. Springer.

Dixon, S. (2001). An empirical comparison of tempo trackers. In *Proceedings of the 8th Brazilian Symposium on Computer Music*. To appear.

Dixon, S. and Cambouropoulos, E. (2000). Beat tracking with musical knowledge. In *ECAI 2000: Proceedings of the 14th European Conference on Artificial Intelligence*, pages 626–630. IOS Press.

Dixon, S., Goebl, W., and Cambouropoulos, E. (2001). Beat extraction from expressive musical performances. In *2001 Meeting of the Society for Music Perception and Cognition (SMPC2001), Kingston, Ontario*. To appear.

Drake, C., Penel, A., and Bigand, E. (2000). Tapping in time with mechanically and expressively performed music. *Music Perception*, 18(1):1–23.

Friberg, A. (1995). *A Quantitative Rule System for Musical Performance*. PhD thesis, Royal Institute of Technology, Stockholm.

Gabrielsson, A. (1999). The performance of music. In Deutsch, D., editor, *The Psychology of Music*, pages 501–602. Academic Press, 2nd edition.

Goebl, W. (2001). Melody lead in piano performance: Expressive device or artifact? *Journal of the Acoustical Society of America*. To appear.

Gordon, J. (1987). The perceptual attack time of musical tones. *Journal of the Acoustical Society of America*, 82(1):88–105.

Goto, M. and Muraoka, Y. (1995). A real-time beat tracking system for audio signals. In *Proceedings of the International Computer Music Conference*, pages 171–174. Computer Music Association, San Francisco CA.

Goto, M. and Muraoka, Y. (1997a). 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*, pages 9–16. International Joint Conference on Artificial Intelligence.

Goto, M. and Muraoka, Y. (1997b). 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*, pages 135–144. International Joint Conference on Artificial Intelligence.

Goto, M. and Muraoka, Y. (1998). An audio-based real-time beat tracking system and its applications. In *Proceedings of the International Computer Music Conference*, pages 17–20. Computer Music Association, San Francisco CA.

Goto, M. and Muraoka, Y. (1999). Real-time beat tracking for drumless audio signals. *Speech Communication*, 27(3–4):331–335.

Handel, S. (1989). *Listening: An Introduction to the Perception of Auditory Events*. Bradford, MIT Press.

Keil, C. (1995). The theory of participatory discrepancies: a progress report. *Ethnomusicology*, 39(1):1–19.

Large, E. (1995). Beat tracking with a nonlinear oscillator. In *Proceedings of the IJCAI'95 Workshop on Artificial Intelligence and Music*, pages 24–31. International Joint Conference on Artificial Intelligence.

Large, E. (1996). Modelling beat perception with a nonlinear oscillator. In *Proceedings of the 18th Annual Conference of the Cognitive Science Society*.

Large, E. and Kolen, J. (1994). Resonance and the perception of musical meter. *Connection Science*, 6:177–208.

Lee, C. (1991). The perception of metrical structure: Experimental evidence and a model. In Howell, P., West, R., and Cross, I., editors, *Representing Musical Structure*, pages 59–127. Academic Press.

Lerdahl, F. and Jackendoff, R. (1983). *A Generative Theory of Tonal Music*. MIT Press.

Longuet-Higgins, H. (1987). *Mental Processes*. MIT Press.

Longuet-Higgins, H. and Lee, C. (1982). The perception of musical rhythms. *Perception*, 11:115–128.

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

Palmer, C. (1996). Anatomy of a performance: Sources of musical expression. *Music Perception*, 13(3):433–453.

Parncutt, R. (1987). The perception of pulse in musical rhythm. In Gabrielsson, A., editor, *Action and Perception in Rhythm and Music*, number 55, pages 127–138. Royal Stockholm Academy of Music.

Parncutt, R. (1994). A perceptual model of pulse salience and metrical accent in musical rhythms. *Music Perception*, 11(4):409–464.

Povel, D. and Essens, P. (1985). Perception of temporal patterns. *Music Perception*, 2(4):411–440.

Prögler, J. (1995). Searching for swing: Participatory discrepancies in the jazz rhythm section. *Ethnomusicology*, 39(1):21–54.

Repp, B. (1992). Diversity and commonality in music performance: An analysis of timing microstructure in Schumann's "Träumerei". *Journal of the Acoustical Society of America*, 92(5):2546–2568.

Repp, B. (1994). On determining the basic tempo of an expressive music performance. *Psychology of Music*, 22:157–167.

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

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

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

Schloss, W. (1985). *On the Automatic Transcription of Percussive Music: From Acoustic Signal to High Level Analysis*. PhD thesis, Stanford University, CCRMA.

Sethares, W. and Staley, T. (1999). Periodicity transforms. *IEEE Transactions on Signal Processing*, 47(11).

Sethares, W. and Staley, T. (2001). Meter and periodicity in musical performance. *Journal of New Music Research*. To appear.

Steedman, M. (1977). The perception of musical rhythm and metre. *Perception*, 6:555–569.

Sundberg, J. (1991). *The Science of Musical Sounds*. Academic Press.

Tanguiane, A. (1993). *Artificial Perception and Music Recognition*. Springer.

Todd, N. (1985). A model of expressive timing in tonal music. *Music Perception*, 3:33–58.

Todd, N. and Lee, C. (1994). An auditory-motor model of beat induction. In *Proceedings of the International Computer Music Conference*, pages 88–89. International Computer Music Association.

van Noorden, L. and Moelants, D. (1999). Resonance in the perception of musical pulse. *Journal of New Music Research*, 28(1):43–66.

Vos, J. and Rasch, R. (1981). The perceptual onset of musical tones. *Perception and Psychophysics*, 29(4):323–335.