

Real-Time Tempo Tracking Using Rules to Analyze Rhythmic Qualities

Anthonie Driesse

Department of Computing and Information Science
Queen's University
Kingston, Canada

Abstract

This paper presents a tempo tracking system adaptable to different styles of music. The system analyzes tonal music having a rhythmic structure based on a simple meter. The focus is on tracking gradual tempo changes during the live performance of a piece of music.

A musician using the system plays an instrument connected to the computer through a MIDI interface. The system analyzes every note played according to a set of rules which identify temporal, melodic, harmonic and expressive qualities that contribute to the rhythmic structure in the music. This establishes the location of the note with respect to the beat, and allows another set of rules to predict the time of the next beat. The predicted beat can then be used to synchronize programmed accompaniment.

The flexibility of the system stems from the use of distinct rules which can be prioritized. High priorities are assigned to rules which identify strong rhythmic qualities; low priority rules become effective in the absence of such qualities. Furthermore, the priorities can be tailored to suit a specific style, or a particular piece of music by identifying which qualities are most relevant to the rhythmic structure.

1. Introduction

Tempo variations in live performances add expression to music. Because these variations hinder computer accompaniment and transcription systems, they create a need for tempo tracking or beat following algorithms to determine the relationship between actual time and musical time. Several researchers have developed real-time computer accompaniment systems [Bloch/Dannenberg, 1985; Dannenberg et. al., 1984, 1987, 1988, 1989; Vercoe, 1984, 1985]. Although the research does not always focus on tempo tracking algorithms, the systems usually include them. Only one research team has recently chosen to concentrate on the tempo tracking problem [Allen/Dannenberg 1990].

Our research investigates the possibility of tracking tempo changes by applying musical rules. Whereas Allen and Dannenberg [1990] analyze note onset times only, our system includes rules that evaluate other aspects of the performance, such as pitch and dynamics. Music theorists have already formulated such rules to identify rhythmic features in music and define their contribution to the perceived beat or pulse [Lerdahl/Jackendoff, 1983; Lester,

This research was partially funded by the Natural Sciences and Engineering Research Council of Canada.

1986]. Using some of the rules proposed by Lerdahl and Jackendoff [1983], Miller et. al. [1989] have implemented a non-real-time system to generate a metric structure, but it also analyzes only note onset times.

In theory, a rule-based tempo tracking system could be used with any kind of music as long as appropriate generalizations could be made concerning the rhythm of the music. Because this is difficult for some modern music, we consider only the domain of western tonal music. The music must have a regular pulse, corresponding to a regular time signature.

The hardware used for our system comprises a velocity sensitive electronic keyboard connected to a Macintosh computer through a MIDI interface. The system analyzes the music input and creates a MIDI synchronization signal corresponding to the beats in the music. This signal controls a standard MIDI drum machine playing a preprogrammed pattern to demonstrate how well the system performs. For testing and demonstration purposes, a MIDI sequencer can provide alternative input.

This paper describes the component parts of our tempo tracking system and presents some possible rules. It also discusses a problem with testing real-time tempo tracking systems.

2. Breaking down the problem

Vercoe [1985] presents his real-time accompaniment system as a synthetic performer able to perform three tasks: listen, perform and learn. Dannenberg [1989] calls the first two tasks the 'analytical' and the 'musical' part of his computer accompaniment system, but leaves out the learning. Our system has a similar structure: The listening task consists of analyzing the music input and detecting when the beats occur; the performing task involves predicting future beats and generating a synchronization signal. When we create new rules or modify priorities the system 'learns' to accomodate different styles of music.

2.1. Detection

Techniques for detecting beats can be divided into two distinct categories: 1) known score and 2) unknown score. If a detector knows the score being played, it must match the notes in a performance to those represented by the score. The detector locates a beat in the performance when the score indicates that the matched note occurs on a beat. In the second category, the detector must decide on the basis of some general musical knowledge which notes occur on beats. Our system belongs in the second category: It can track the tempo of an improvising musician.

2.2. Prediction

Prediction of future beats is the same for both of the above categories since scores do not usually contain exact tempo information (and if so, the musician may well ignore it). To accomplish this we must model the behaviour of tempo changes. This model requires that tempo changes are gradual, since there is no way to predict instantaneous changes. Consequently, the initial beats of a piece are unpredictable as well. We solve this problem by counting down before the music starts, as musicians commonly do.

3. Where do the rules fit in?

Having divided the problem into detection and prediction, we can now show how these tasks are performed and how they interact. Specifically, we will show how they are performed according to a set of rules.

3.1. Rules for detection

As stated earlier, the detector must decide on the basis of some general musical knowledge which notes occur on beats. The detector knows or can gain access to the note pitches (key number), note onset times, note durations and attack dynamics (key velocity). The musical knowledge rules use this information to identify rhythmic qualities of notes. Each rule scores the strength of the rhythmic feature as a value ranging from 'absent' to 'strong', which can be interpreted as the likelihood that the note landed on a beat.

Lester [1986] provides rule-like descriptions of various types of accents, some of which can establish meter. Similarly, Lerdahl and Jackendoff [1983] list many rules, some to generate grouping structures and some to generate metric structures. Five rules chosen for implementation are:

- 1) a relatively long note likely begins on a beat
- 2) a relatively loud note likely begins on a beat
- 3) a bass note more likely begins on a beat than a higher note
- 4) several nearly simultaneous notes likely indicate a beat
- 5) a change in the harmonic relationship between currently sounding notes likely indicates a beat

The rules are applied in combination with their relative priorities to calculate the likelihood that a note landed on a beat. Both Lerdahl and Jackendoff [1983] and Lester [1986] rank rules to indicate that some rhythmic features are intrinsically stronger than others. If the evidence is strong enough, we conclude that the beat coincided with the note. When several notes happen to land on the same beat, we choose the note with the strongest evidence.

3.2. Rules for prediction

A successful prediction algorithm must extrapolate from the timing of the detected beats without abruptly changing the sequence of predicted beats. Rather than attempt to model the tempo changes mathematically, we formulate a number of rules that impose limitations on the possible tempo changes. These rules must also account for potential timing errors by the musician.

The predictor has available the times of all predicted and detected beats up to the current beat, although it analyzes only the last few measures played. First, the predictor calculates a weighted average of the time interval between consecutive detected beats (beat periods) for the past measure or two in order to reduce the effect of random timing errors. This average beat period is then added to the time of the last detected beat to make an initial prediction. Because the averaging causes a sluggish response to tempo changes, the predictor also adds the average of the difference between beat periods.

Several additional rules apply to the predictor: if the beat proposed by the detector leads to an abrupt change, the predictor discards it; if the detector fails to locate a beat, we assume the previous prediction was correct and; if we predict an abrupt change, we limit the magnitude of that change. Another rule could compensate if the computer consistently leads or lags. We could even adjust the prediction in response to analysis of local dynamics or note density to apply the observation, for example, that an increase in volume often accompanies an increase in tempo.

4. Testing tempo tracking systems

Tempo tracking algorithms usually focus on the task of detecting what the musician is doing, and predicting what (s)he will do next. Demonstrations of these systems are often live performances of musicians accompanied by computer. However, the discussion by other researchers so far does not consider the musician's reaction to the computer's actions. In a live accompaniment system we are dealing with a closed loop, and the behaviour of the

system depends on the behaviour of both components: the computer and the musician. We suspect that good results are sometimes achieved because the musician compensates for misjudgments on the part of the computer.

To better demonstrate the behaviour of the tracking algorithm itself, this loop must be opened. This can be done by tracking the tempo of a performance recorded on a sequencer (without metronome) or by isolating the musician from the accompaniment. Note that both open and closed loop configurations have useful applications, but we must avoid the assumption or implication that if one works well, the other also works well. We have tested our system in both configurations and provide demonstrations of both.

References

- Allen, Paul E. and Roger B. Dannenberg [1990]
"Tracking Musical Beats in Real Time," *Proceedings of the 1990 International Computer Music Conference*, pp. 140-143.
- Bloch, Joshua J. and Roger B. Dannenberg [1985]
"Real-Time Computer Accompaniment of Keyboard Performances," *Proceedings of the 1985 International Computer Music Conference*, pp. 279-289.
- Dannenberg, Roger B. [1984]
"An On-Line Algorithm for Real-Time Accompaniment," *Proceedings of the 1984 International Computer Music Conference*, pp. 193-198.
- Dannenberg, Roger B. and Bernard Mont-Reynaud [1987]
"Following an Improvisation in Real Time," *Proceedings of the 1987 International Computer Music Conference*, pp. 241- 248.
- Dannenberg, Roger B. and Hirofumi Mukaino [1988]
"New Techniques for Enhanced Quality of Computer Accompaniment," *Proceedings of the 1988 International Computer Music Conference*, pp. 243- 249.
- Dannenberg, Roger B. [1989]
"Real-Time Scheduling and Computer Accompaniment," in *Current Directions in Computer Music Research*. MIT Press, Cambridge, 1989.
- Lerdahl, Fred and Ray S. Jackendoff [1983]
A Generative Theory of Tonal Music. MIT Press, Cambridge, 1983.
- Lester, Joel [1986]
The Rhythms of Tonal Music. Southern Illinois University Press, Carbondale and Edwardsville, 1986.
- Miller, B., D. Scarborough and D. Jones [1989]
"Rule-Based versus Constraint-Satisfaction Approaches to the Perception of Meter in Music," *Proceedings of the Second International Workshop on Artificial Intelligence and Music* at IJCAI-89. Detroit, Michigan, Aug. 1989.
- Vercoe, Barry [1984]
"The Synthetic Performer in the Context of Live Performance," *Proceedings of the 1984 International Computer Music Conference*, pp. 199-200.
- Vercoe, Barry [1985]
"Synthetic Rehearsal: Training the Synthetic Performer," *Proceedings of the 1985 International Computer Music Conference*, pp. 275-278.