Recent Work In Real-Time Music Understanding By Computer[1]

Roger B. Dannenberg

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA

Abstract

Music Understanding refers to the recognition or identification of structure and pattern in musical information. Three distinct music understanding projects initiated by the author are discussed. In the first, Computer Accompaniment, the goal is to follow a performer in a score. Knowledge of the position in the score as a function of time can be used to synchronize an accompaniment to the live performer and automatically adjust to tempo variations. In the second project, it is shown that statistical methods can be used to recognize the location of an improvisor in a cyclic chord progression such as the 12-bar blues. This problem is similar to Computer Accompaniment; however, much less information is conveyed by any one note because pitches are not predetermined. The third project, Beat Tracking, attempts to identify musical beats using note-onset times from a live performance. Parallel search techniques are used to consider several hypotheses simultaneously, and both timing and higher-level musical knowledge are integrated to evaluate the hypotheses.

1. Introduction

Music Understanding is the study of methods by which computer music systems can recognize pattern and structure in musical information. One of the difficulties of research in this area is the general lack of formal understanding of music. For example, experts disagree over how music structure should be represented, and even within a given system of representation, music structure is often ambiguous. Because of these difficulties, my work has focussed on fairly low-level musical tasks for which the interpretation of results is usually straightforward.

---

[1]Dannenberg, ‘‘Recent Work In Real-Time Music Understanding By Computer,’’ ***Music, Language, Speech, and Brain***, Wenner-Gren International Symposium Series, Sundberg, Nord, and Carlson, ed., Macmillan, 1991, pp. 194-202.

The following sections will describe three Music Understanding skills. The first two provide the basis for the responsive synchronization of a musical accompaniment. The first skill is to follow a performance in a score, that is, to match performed notes with a notated score in spite of timing variations and performance errors. The second skill is to follow a jazz improvisation for which the underlying chord sequence is known but specific pitches are not known. The third skill is ‘‘foot-tapping’’: to identify the time and duration of beats, given a performance of metrical music. This skill provides the basis for a variety of capabilities that include synchronization and music transcription.

Research has shown that these ‘‘low-level’’ musical tasks are far from trivial to automate. They are similar in that they (1) are well-defined, challenging problems, (2) have fairly objective measures of success, and (3) offer solutions to practical problems of real-time computer music performance. The following three sections will explore the work to date on these projects. The concluding section will summarize the work and suggest directions for future research.

2. Score Following and Computer Accompaniment

A basic skill for the musically literate is to read music notation while listening to a performance. Humans can follow quite complex scores in real-time without having previously heard the music or seen the score. The task of Computer Accompaniment is to follow a live performance in a score and to synchronize a computer performance. Note that the computer performs a precomposed part, so there is no real-time composition involved but rather a responsive synchronization.

Several computer accompaniment systems have been implemented by the author and his colleagues (Dannenberg, 1984, Bloch and Dannenberg, 1985, Dannenberg and Mukaino, 1988). These differ from the accompaniment systems of others (Vercoe and Puckette, 1985, Lifton, 1985) primarily in the algorithms used for score following. Only the score following component developed by the author will be described here. Score following can be considered to have two subtasks as shown in figure 1.

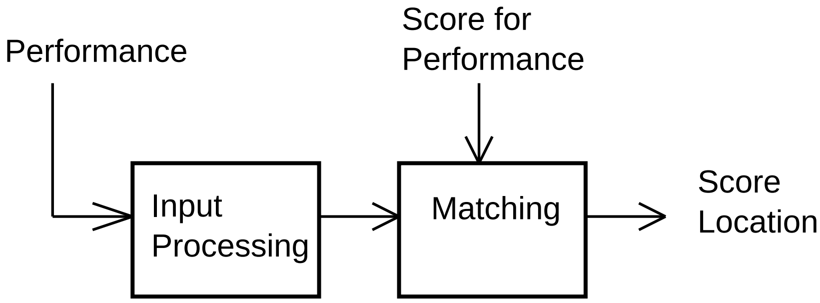

Figure 1: The correspondence between a score and a performance.

2.1. Input Processing

The first task, the Input Processor, translates the human performance (which may be detected by a microphone or by mechanical sensors attached to keys) into a sequence of symbols which typically correspond to pitches. With microphone input, the pitch must be estimated and quantized to the nearest semitone, and additional processing is useful to reduce the number of false outputs that typically arise.

2.2. Matching

The Matcher receives input from the Input Processor and attempts to find a correspondence between the real-time performance and the score. The Matcher has access to the entire score before the performance begins. As each note is reported by the Input Processor, the matcher looks for a corresponding note in the score. Whenever a match is found, it is output. The information needed for Computer Accompaniment is just the real-time occurence of the note performed by the human and the designated time of the note according to the score.

Since the Matcher must be tolerant of timing variations, matching is performed on sequences of pitches only. This decision makes the matcher completely time-independent. One problem raised by this pitch-only approach is that each pitch is likely to occur many times in a composition. In a typical melody, a few pitches occur in many places, so there may be many candidates to match a given performed note.

The matcher described here overcomes this problem and works well in practice. The matcher is derived from the dynamic programming algorithm for finding the longest common subsequence (LCS) of two strings (Sankoff and Kruskal, 1983). Imagine starting with two strings and eliminating arbitrary characters from each string until the the remaining characters (subsequences) match exactly. If these strings represent the performance and score, respectively, then a common subsequence represents a potential correspondence between performed notes and the score (see figure 2). If we assume that most of the score will be performed correctly, then the longest possible common subsequence should be close to the ‘‘true’’ correspondence between performance and score.

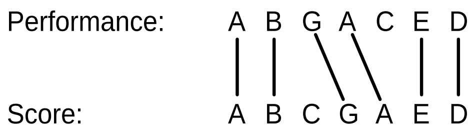

Figure 2: The correspondence between a score and a performance.

In practice, it is necessary to match the performance against the score as the performance unfolds, so only an initial subsequence of the entire performance is available. This causes an interesting anomaly: if a wrong note is played, the LCS algorithm will search arbitrarily far ahead into the score to find a match. This will more than likely turn out *not* to be the best match once more notes are played, but being unreasonably wrong, even momentarily, causes problems in the accompaniment task. To avoid skipping ahead in the score, the algorithm is modified to maximize the number of corresponding notes *minus* the number of notes skipped in the score. Other functions are possible, but this one works well: the matcher will only skip notes when their number is offset by a larger number of matching notes.

The Matcher is an interesting combination of algorithm design, use of heuristics, and outright ad-hoc decisions. Much of the challenge in designing the matcher was to model the matching problem in such a way that good results could be obtained efficiently.

Polyphonic matchers have also been explored. One approach is to group individual notes that occur approximately simultaneously into structures called *compound events*. A single isolated note is considered to be a degenerate form of compound event. By modifying the definition of ‘‘matches’’, the monophonic matcher can be used to find a correspondence between two sequences of compound events. Another approach processes each incoming performance event as it occurs with no regard to its timing relationship to other performed notes. It is important in this case to allow notes within a chord (compound event) in the score to arrive in any order. (Note that the LCS algorithm disallows reordering.) The resulting algorithm is time-independent. This work was performed with Joshua Bloch (Bloch and Dannenberg, 1985) and the reader is referred to our paper for further details.

The Matcher performs a fairly low-level recognition task where efficiency is important and relatively little knowledge is required. When matches are found, they are output for use by an Accompaniment Performance subtask, which uses knowledge about musical performance to control a synthesizer. Several systems have been implemented based on these techniques, and the results are quite good. Recent work has extended the Matcher to handle trills, glissandi, and grace notes as special cases that would otherwise cause problems (Dannenberg and Mukaino, 1988), and this version has been used successfully for several concerts.

3. Following Improvisations

Knowledgeable listeners can often identify a popular song even when the melody is not played. This is possible because harmonic and rhythmic structures are present even without the melody. Even the improvisations of a single monophonic instrument can contain enough clues for a listener to discern the underlying harmonic and rhythmic structure. Can a computer system exhibit this level of music understanding?

Although many different problems involving improvisation might be posed, a particular task was chosen for study and implementation. The task involves listening to a 12-bar blues improvisation in a known key and played by a monophonic instrument. The goal is to detect the underlying beat of the improvisation and to locate the start of the cyclical chord progression. This is enough information to, for example, join in the performance with a synthesized rhythm section consisting of piano, bass, and drums.

This improvisation understanding task can be divided into two subtasks: finding the beat and finding the harmonic progression. After lengthy discussions with Bernard Mont-Reynaud, who developed beat-tracking software for an automatic music transcription system (Chafe *et. al.*, 1982), we decided to collaborate in the design and implementation of a ‘‘blues follower’’ program (Dannenberg and Mont-Reynaud, 1987). Dr. Mont-Reynaud designed the beat follower or ‘‘foot tapper’’, and I designed the harmonic analysis software.

Since ‘‘foot tapping’’ is the subject of the next section, we will proceed to the problem of harmonic analysis. One of the difficulties of understanding an improvisation is that virtually any pitch can occur in the context of any harmony. However, given a harmonic context many notes would only be used in certain roles such as a chromatic passing tone. This led to the idea that by searching for various features, one might assign functions to different notes. Once labeled with their function, it might be possible after a few notes to unambiguously determine the harmonic context by the process of elimination.

So far, this approach has not been fruitful, so a more statistical approach was tried. In this approach, it is assumed that even though any pitch is possible in any context, there is a certain probability distribution associated with each time position in the 12-bar blues form. For example, in the key of C, we might expect to see a relatively frequent occurrence of the pitch B in measure 9 where B forms the important major third interval to the root of the dominant chord (G). We can calculate a ‘‘correlation’’ between the expected distribution and the actual solo to obtain a figure of merit. This not a true numerical correlation but a likelihood estimate formed by the product of the probabilities of each note of the solo. Since we wish to find where the 12-bar blues form begins in the solo, we compute this estimate for each possible starting point. The point with the highest likelihood estimate indicates the most likely true starting point.

Figure 3 illustrates a typical graph of this likelihood estimate vs. starting point. (Since only the relative likelihood is of interest, the computed values are not normalized to obtain true probabilities, and the plotted values are the direct result of integer computations. See Dannenberg and Mont-Reynaud (1987) for details.) Both the graph and the 12-bar blues form are periodic with a period of 96 eighth notes. Slightly more than one period is plotted so that the peak at zero is repeated around 96. Thus the two peaks are really one and the same, modulo 12 bars. The peak does in fact occur at the right place. There is also a noticeable 4-bar (32 eighths) secondary periodicity that seems to be related to the fact that the 12-bar blues form consists of 3 related 4-bar phrases.

The probability distribution used for the ‘‘correlation’’ can be obtained from actual performances. The beat and starting point of the 12-bar form are recorded along with the notes of the performance or are manually indicated later. The distribution used in figure 3 was obtained in this way and combines the pitches of about 40 repetitions of the 12-bar chord progression. The data were obtained from the recorded output of a real-time pitch detector. An interesting question is whether it matters if the distribution and the solo are created by the same soloist. If so, can this technique be used to identify a soloist? These questions have not yet been studied.

The ‘‘foot tapper’’ and a real-time implementation of the correlation approach were integrated into a real-time improvisation understanding program for further experimentation. The results are interesting, but not up to the level required for serious applications. The tempo estimation software tends to start reliably but eventually loses synchronization with the performance unless the performance is very precise. The ‘‘correlation’’ software can locate the beginning of the blues form only when the performance is very obvious in outlining the harmonic structure. This is less of a problem when the likelihood estimation is performed over many measures, but then the technique becomes totally unsuited to real-time performances.

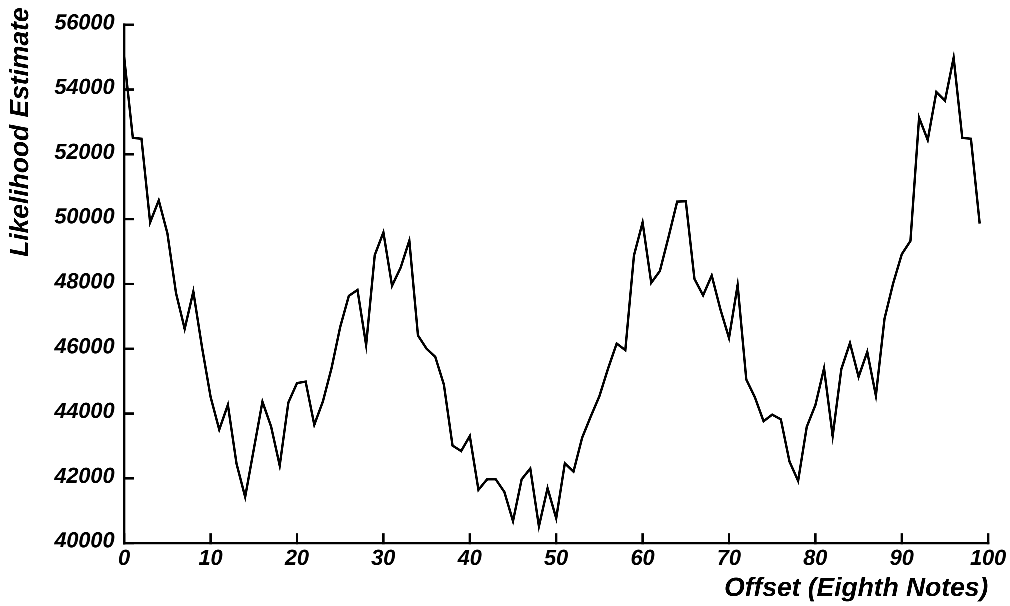

Figure 3: Likelihood estimates of the solo starting at different offsets in a 12-bar blues progression.

Even though some of the simplest approaches tried thus far have been the most successful, it seems obvious that human listeners bring together a panoply of techniques and knowledge in order to follow a blues solo and interpret it correctly. Further research is needed to explore new approaches and to examine ways in which results from different approaches can be integrated.

4. Rhythm Understanding

The ‘‘foot tapping’’ problem is to identify the location and duration of beats in metrical music. Conceptually, foot tapping is easy. One assumes that note onsets frequently occur on regularly spaced beats. The problem then is to find a slowly varying tempo function that predicts beats in correspondence with the observed note onsets. If a beat prediction occurs just before a note onset, then the estimated tempo is assumed to be slightly fast and the estimate is decreased. If a note onset occurs just before a beat prediction, then the estimated tempo is assumed to be too slow, and the estimate is increased. In this way, the predicted beats can be brought to coincide with note onsets and presumably the ‘‘true’’ beat (Longuet-Higgins and Lee, 1982).

In practice, straightforward implementations of this approach are not very reliable. In order to make the foot tapper responsive to tempo changes, it must be capable of large tempo shifts on the basis of the timing of one or two notes. This tends to make the foot tapper very sensitive to ordinary fluctuations in timing that do not represent tempo changes. On the other hand, making the foot tapper less sensitive destroys its ability to change tempo. Furthermore, once the foot tapper gets off the beat, it is very difficult to lock back into synchronization.

I am currently working with Paul Allen on a system that is designed to be both reliable and responsive to tempo change (Allen and Dannenberg, 1990). Our observation was that simpler foot tappers often came to a situation where the interpretation of a note onset was ambiguous. Did the tempo increase such that the note onset is on a downbeat (one interpretation), or did the tempo decrease such that the note onset is before the downbeat (an alternative interpretation)? Once an error is made, a simple foot tapper tends to make further mistakes in order to force its estimates to fit the performance data. Foot tappers seem to diverge from, rather than converge to, the correct tempo.

To avoid this problem, we implemented a system that keeps track of many alternative interpretations of note onsets using the technique of *beam search*. Beam search keeps track of a number of alternative interpretations, where each interpretation consists of an estimated beat duration (the tempo) and an estimated beat phase (current position within the beat). In figure 4, circles represent interpretations. As each new note arrives, new interpretations are generated in the context of each stored alternative. In the figure, each successive row represents the set of interpretations generated for a new note onset, and lines show the context in which the new interpretation is made. The least promising new interpretations are discarded to avoid an exponential growth of alternatives, as indicated by the diagonal crosses. Although the figure illustrates only a few interpretations at each level, hundreds of interpretations may be computed for each note onset in practice.

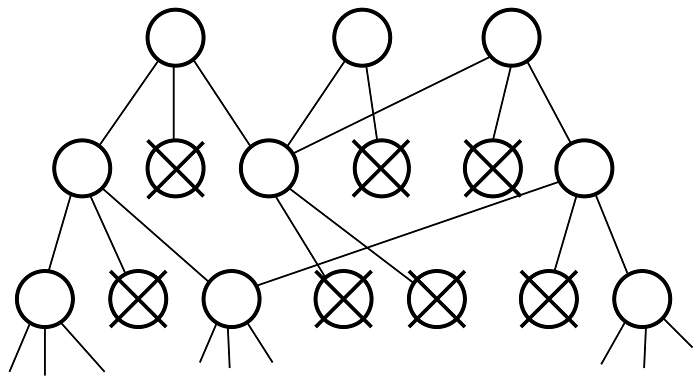

Figure 4: Three levels of beam search.

Just as with previous foot tappers, it is critical not to throw away the correct interpretation. We use a number of heuristics to give ratings to the generated interpretations. For example, interpretations are penalized if they require a large tempo change or if they result in a complex rhythm. Certain rhythmic combinations, such as a dotted eighth note on a downbeat followed by a quarter note triplet, are not allowed at all (even though they may be theoretically possible).

One of the difficulties we encountered was that the search tends to become congested with a large number of very similar alternatives representing slight variations on what is essentially one interpretation of the data. This uses resources that could otherwise be searching truly different interpretations. We avoid congestion of this sort by coalescing interpretations that have the same beat and beat phase and very nearly the same tempo estimates. If the ratings differ, only the best rating is retained.

The output of this foot tapper is based on the interpretation with the highest rating. Typically, this will be the correct interpretation, but occasionally the highest rating will go to an incorrect one. (If this never happened, there would be no need for searching.) The beam search continues to explore other likely alternatives; if one of these is in fact the correct interpretation, then it is likely to provide better predictions of musical events in the long run, and its rating will eventually become the highest one. In this way, the foot tapper can avoid being permanently thrown off course by a single wrong decision.

A real-time implementation of the foot tapper is running and the initial results show that the system improves upon simpler approaches. The system can track substantial tempo changes and tolerate the timing variations of amateur keyboard players. The quality and reliability of tracking is, however, dependent upon the music: steady eighth notes are much easier to follow than highly syncopated music. Further characterization of the system is needed, and an understanding of its limitations will lead to further improvements.

5. Summary and Conclusions

Three systems for Music Understanding have been illustrated. The goal in each case has been to recognize pattern or structure in music in order to perform a low-level musical task. Systems have been described that provide automatic synchronization of accompaniment, accompaniment of jazz impovisations, and beat and tempo identification.

There are many directions to take in future research. The problem of following an ensemble, as opposed to an individual performer, has not been addressed, and little effort has been made to address the problems of vocal music where note onsets and pitches are not as easy to detect as with most instruments. Several problems relating to following improvisations have already been mentioned: can analysis be used to identify soloists? Can features be used to improve the recognition of chord progressions implied by a solo line? Listening to improvised keyboard performances should be easier than monophonic instruments because chords tend to be played, but this has yet to be studied. Finally, the foot tapping problem is far from solved, and new search strategies are needed. Humans learn about a work of music as they listen to it, and it seems likely that this is an important factor in rhythm perception. The incorporation of learning into Music Understanding systems is an important future direction.

Even present-day systems illustrate that Music Understanding can have a profound effect on the way musicians and computers interact. As the level of understanding increases, the applicability of computers to musical problems will grow. With the flexibility and adaptability made possible by Music Understanding, computer-based systems seem likely to take an increasingly important role in music composition, performance, and in the development of musical esthetics.

One can also study to what extent automated Music Understanding systems model the cognitive activities of human musicians. Introspection has been a useful technique for designing Music Understanding systems, indicating that what seems to work for humans often works for machines. This is not at all the same as saying that what works for machines *is* what works for humans. However, research in Music Understanding can provide some testable models of what *might* be going on in the mind. In any case, it is hoped that the identification and exploration of well-defined musical tasks may lead to a better and more formal understanding of music.

6. Acknowledgments

I would like to thank Frances Dannenberg for suggesting a number of improvements to an earlier draft. In addition, this work could not have been carried out without major contributions from a number of colleagues. Joshua Bloch co-designed and implemented the first polyphonic computer accompaniment system. Bernard Mont-Reynaud designed and implemented the beat tracker for the jazz improvisation understanding system, and Paul Allen co-designed and implemented the foot tapper program in addition to evaluating many alternative designs. This work has been made possible largely through the Carnegie Mellon University School of Computer Science and was partially supported by Yamaha.

References

Allen, P. E., and R. B. Dannenberg. (1990). Tracking Musical Beats in Real Time. In *Proc. of the 1990 International Computer Music Conf.*, Computer Music Association, (to appear).

Bloch, J. J. and R. B. Dannenberg. (1985). Real-Time Computer Accompaniment of Keyboard Performances. In *Proc. of the 1985 International Computer Music Conf.*, Computer Music Association, 279-290.

Chafe, C., B. Mont-Reynaud, and L. Rush. (1982). Toward an Intelligent Editor of Digital Audio: Recognition of Musical Constructs. *Computer Music Journal* **6**(**1**), 30-41.

Dannenberg, R. B. (1984). An On-Line Algorithm for Real-Time Accompaniment. In *Proc. of the 1984 International Computer Music Conf.*, Computer Music Association, 193-198.

Dannenberg, R. B. and B. Mont-Reynaud. (1987). Following an Improvisation in Real Time. In *Proc. of the 1987 International Computer Music Conf.*, Computer Music Association, 241-248.

Dannenberg, R. B. and H. Mukaino. (1988). New Techniques for Enhanced Quality of Computer Accompaniment. In *Proc. of the 1988 International Computer Music Conf.*, Computer Music Association, 243-249.

Lifton, John. (1985). Some Technical and Aesthetic Considerations in Software for Live Interactive Performance. In *Proc. of the 1985 International Computer Music Conf.*, Computer Music Association, pages 303-306.

Longuet-Higgins, H. C. and C. S. Lee. (1982). The Perception of Musical Rhythms. *Perception*, ***11***, 115-128.

Sankoff, D. and J. B. Kruskal (eds.). (1983). *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley, Reading, Mass.

Vercoe, B. and M. Puckette. (1985). Synthetic Rehearsal: Training the Synthetic Performer. In *Proc. of the 1985 International Computer Music Conf.*, Computer Music Association, 275-278.