Computer Accompaniment and Music Understanding[1]

Roger B. Dannenberg

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA
Email: dannenberg@cs.cmu.edu

Abstract

Music Understanding is the recognition or identification of structure and pattern in musical information. Five music understanding projects are discussed. In the first, Computer Accompaniment of Melodic Instruments, the goal is for a computer system to listen to and play along with a live performer, using a predetermined score. In the second project, the technique is extended to handle polyphonic keyboard input so that a computer can accompany a pianist. In the third project, accompaniment is extended to a situation, a blues improvisation, where the score is not known in advance, so an understanding of deeper musical structures is required. The fourth project, Beat Tracking, attempts to identify musical beats in a live performance without a score. The tempo and a transcription of the performance are by-products of Beat Tracking. The final project is the Piano Tutor, a computer-based intelligent teaching system.

1. Introduction

Music Understanding is a relatively new area of research, but it is an important one for the future of computer music. As music systems become more sophisticated, it becomes more difficult to use them effectively. Artificial intelligence techniques offer hope that complex systems can be made easier to use and therefore reach a broader audience of potential users. A key component of intelligent, easy-to-use systems, is Music Understanding.

Music Understanding is the study of techniques by which computer music systems can recognize pattern and structure in musical information. Understanding is a prerequisite for many interesting tasks. For example, keyboard performers are limited in their ability to produce multiple timbres at once, mainly because there is a limit to how much information is conveyed by a keyboard. Therefore, it is difficult to simultaneously produce trumpet, trombone, and saxophone sounds, even if the appropriate synthesizers are on hand. However, a Music Understanding system that could listen to chords might be able to act as a real-time musical arranger, selecting appropriate notes for a section of synthetic brass players. Music

---

[1]Originally published as: Roger B. Dannenberg, ‘‘Computer Accompaniment and Music Understanding,’’ in ***Proceedings of the 1991 KlangArt Kongress***, B. Enders, ed., Universitat Osnabrueck, Germany.

Understanding systems might be able to control reverberation and audio equalization to enhance live or synthetic intruments the way a recording engineer does today. The possibilities are many, and the field is still new.

The following sections briefly describe five projects related to Music Understanding. The first two provide the basis for the responsive synchronization of a musical accompaniment in a live performance of a precomposed score. The third project examines how to follow a jazz improvisation for which the underlying chord sequence is known but specific pitches are not known. The fourth project concerns ‘‘foot-tapping’’: identifying 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. The final project is the Piano Tutor, a system for teaching beginners to play the piano. The Piano Tutor relys on Music Understanding to diagnose student errors.

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 five sections will introduce and explore the work to date on these projects. The concluding section will summarize the work and suggest future directions.

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 84, Bloch 85, Dannenberg 88]. These differ from the accompaniment systems of others [Vercoe 85, Lifton 85] primarily in the algorithms used for score following. Computer Accompaniment can be considered to have two tasks (see Figure 1). The first task is to follow the live performance in the score, and the second is to generate the accompaniment in synchronization to the live performer.

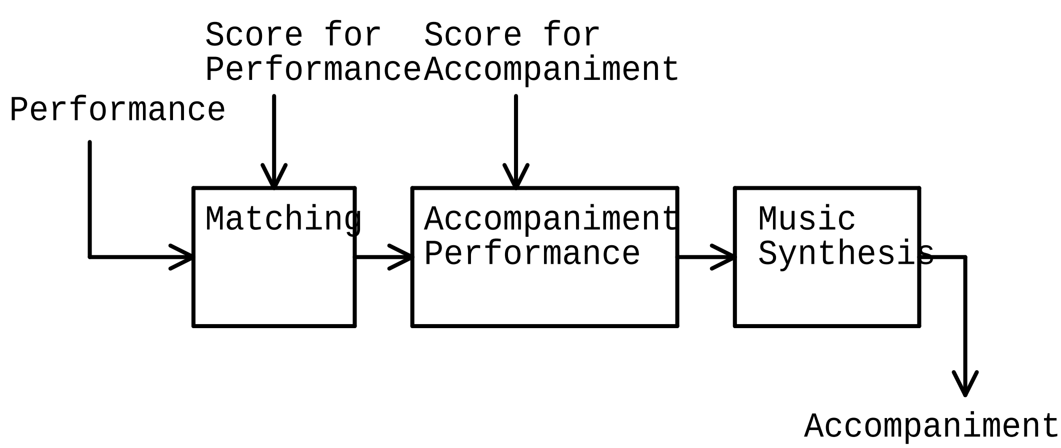

Figure 1: The correspondence between a score and a performance.

2.1. Score Following

The first task of accompaniment is to know, at each point in time, where the performer is located in the score. This is accomplished by matching notes performed live with notes that are in the score. The problem is to find matches even when the performer changes tempo and plays wrong notes.

Figure 2 illustrates the problem of matching performed pitches (denoted by letters) with a sequence of pitches in the score. An efficient computer algorithm has been developed to perform this task [Dannenberg 84], and in practice, the process is very reliable even when faced with frequent wrong, extra, or omitted notes.

```
Performance:  A B G A C E D
              | |  \ \  | |
              | |   \ \ | |
Score:        A B C G A E D
```

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

The matching process is applied only to pitches, and durations and times of notes are ignored in this first task. This makes the matching process more reliable in the face of tempo changes (where matching is most important). The result of the match is simply the knowledge of the position of the performer in the score.

2.2. Accompaniment

When matches are found, they are output for use by an Accompaniment task, which uses knowledge about musical performance to control a synthesizer. The basic operation of the Accompaniment task is quite simple: when a match indicates that the accompaniment is behind the task speeds up, and when the match indicates the accompaniment is ahead, the task slows down.

In practice, the Accompaniment task estimates the speed of the performer by keeping track of recent times and positions at which matches were reported. With a speed and location estimate, the accompaniment can more rapidly reach a synchronization with the live performer. Of course, synchronizing too rapidly can sound very unmusical, so the Accompaniment task uses a number of rules to provide appropriate solutions to various situations. For example, if the Accompaniment task must skip ahead by several seconds, a rule says to skip the intervening notes of the accompaniment. This avoids a rapid, mechanical, and unmusical flight through the score to catch up.

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 88], and this version has been used successfully for several concerts.

3. Polyphonic Accompaniment

The accompaniment system just described works with melodic instruments capable of playing only one pitch at a time. (Incidentally, there is no restriction on the computer-performed accompaniment, which may have many instruments playing a complex polyphonic texture.) The next step in the pursuit of computer accompaniment systems was to consider keyboard input, that is, input that includes chords or even polyphony.

Polyphonic accompaniment required the development of a new matching strategy, and in fact, two matchers were developed. One approach is to group individual notes that occur approximately simultaneously into structures called *compound events*. By modifying the definition of ‘‘matches’’, the monophonic matcher can be used to find a correspondence between two sequences of compound events. This approach is somewhat time-dependent because it uses time to group notes into 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. The resulting algorithm is time-independent. This work was performed with Joshua Bloch [Bloch 85] and the reader is referred to our paper for further details.

Aside from the matcher, a polyphonic system works like a monophonic accompaniment system: matches report score locations to an Accompaniment task, which uses musical rules to synchronize the accompaniment to the live performance. We will see other uses for polyphonic score following and matching below.

4. Following Improvisations

Another Music Understanding problem involves listening to 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?

To explore this question, I set out to develop a computer system that could listen to a 12-bar blues improvisation played on a trumpet. The computer should be able to synchronize an accompaniment by deducing the tempo and the location within the 12-bar blues chord progression. The ‘‘location’’ is not arbitrary: a blues player improvises melodies that are consistent with a particular progression of chords (or harmonies) that repeats every 12 measures (or 48 beats). Once the computer establishes a location and tempo, it can join in with drums, bass, and piano accompaniment.

Like the first computer accompaniment project, this improvisation understanding system can be divided into two tasks: first, finding the location and tempo, and second, synchronizing and generating an accompaniment. The first task required new techniques to be developed, while the second task is essentially the same as in the first computer accompaniment system.

After lengthy discussions with Bernard Mont-Reynaud, who developed beat-tracking software for an automatic music transcription system [Chafe 82], we decided to collaborate in the design and implementation of a ‘‘blues follower’’ program [Dannenberg 87]. 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. My first approach to this problem was not successful, but still seems like a viable approach, so the basics are outlined here. 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, pitches are more likely to occur in some locations than in others. A single note gives us very little information, but it is possible to combine the information from many notes by statistical methods to achieve a reasonably clear summary. The goal then is to compute the most likely location in the progression, given a large number of ‘‘hints.’’ These hints are in the form of pitches that are more likely to occur in some locations and less likely to occur in others.

Figure 3 illustrates a typical graph of this likelihood estimate vs. starting point. (See [Dannenberg 87] 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.

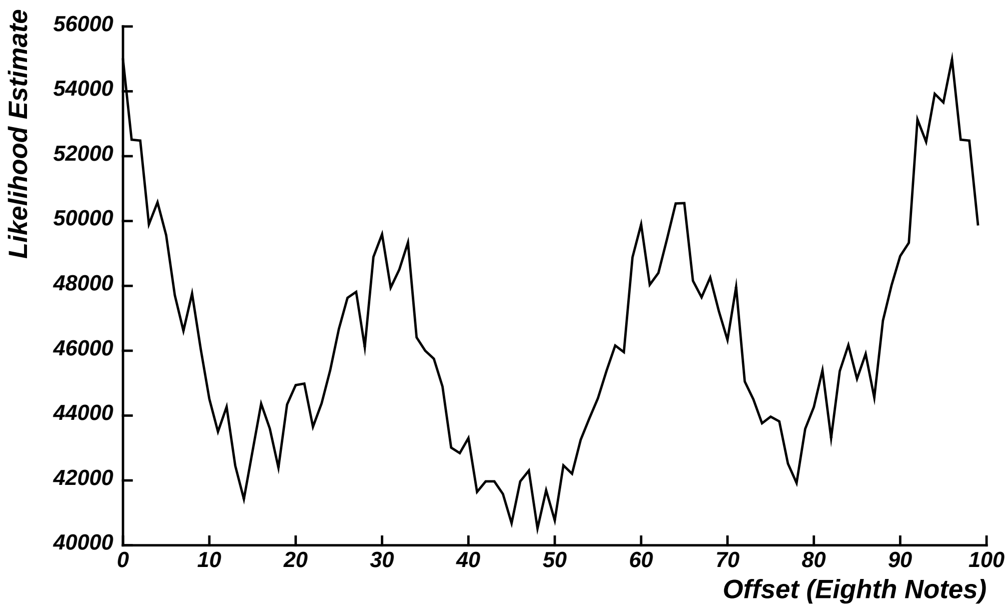

Figure 3: Likelihood estimates (arbitrary units) of the solo starting at different offsets in a 12-bar blues progression.

The ‘‘foot tapper’’ and a real-time implementation of this likelihood 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. It is obvious that human listeners are much more sophisticated than this simple computer listener, and more sophisticated approaches are needed before computer systems will be considered musically competent listeners.

Even if the approach outlined here is not powerful enough for live performance, it suggests some interesting possibilities for analysis. Can these techniques be used to characterize a performer or a blues style? The statistical methods employed allow us to compare two performances or to build a statistical summary of many performances, and there are many types of analysis that could then be applied.

5. 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 (the times at which notes begin) 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 82].

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.

With Paul Allen, I developed a new approach that should be more reliable in tracking a tempo and yet more responsive to tempo change [Allen 90]. 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 row represents a set of new interpretations generated for a new note onset.

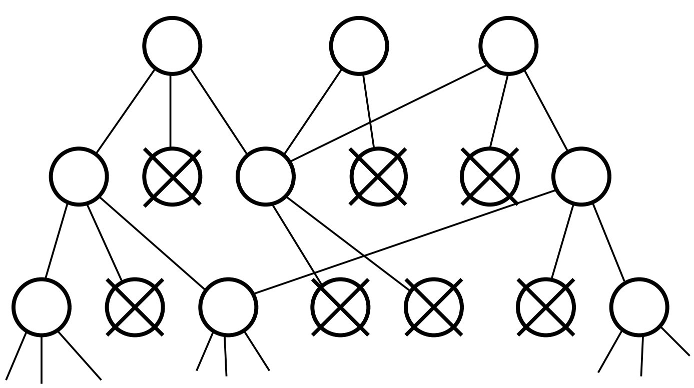

Figure 4: Three levels of beam search.

Just as with previous foot tappers, it is critical not to throw away the correct interpretation. As interpretations unfold with each new note, some lead to situations where beats do not correspond to new note onsets, and some interpretations give good predictions of note onsets. Interpretations that do not make good predictions are discarded, and interpretations that make good predictions are retained. Although the figure illustrates only a few interpretations at each level, hundreds of interpretations may be computed for each note onset in practice.

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).

A real-time implementation of the foot tapper is running and the initial results show that the system sometimes 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.

6. The Piano Tutor

The Piano Tutor is an intelligent computer system for teaching beginners to play the piano. The system consists of an electronic piano interfaced to a computer, a videodisc player, a computer graphics display, and a video display. In a typical interaction, the Piano Tutor selects a lesson and gives the student a corresponding presentation. Lesson presentations usually explain a new skill such as finding and playing a G with the left hand. A task is then given to the student to allow him or her to practice and demonstrate this new skill. The task is usually the performance of a piece of music. After the performance, the Piano Tutor analyses the performance and offers suggestions if the student has made mistakes. When the student successfully completes the task, a new lesson is selected and the process repeats.

The Piano Tutor uses score following to follow along as the student plays, allowing the system to ‘‘turn pages’’ on the computer screen at the proper time. Computer accompaniment is also used to provide a reward to the student once a new piece is mastered. In addition to these uses for score following, the Piano Tutor uses score following as the basis for evaluating student performances. A by-product of score following is a match between each note of the performance and each note of the score. It is then a simple matter to determine where there were wrong notes, missed notes, and extra notes, and where notes were too short, too long, early, or late.

These low-level errors are examined to find explanations. For example, a number of early notes might indicate that the student is increasing the tempo. If the first note is in the wrong octave, the Piano Tutor will tell the student to find the correct octave and start over. If a wrong note is related to a concept or skill that was just taught, the Piano Tutor can repeat the explanation. In many ways, the Piano Tutor exhibits the deepest understanding of any of the systems discussed here. This is possible in part because the domain is very specific: we know in advance what concepts are being taught, what errors students are likely to make, and what pieces of music are to be played. All this makes it easier for the Piano Tutor to analyse student mistakes and come up with plausible reactions.

7. Summary and Conclusions

Five 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. The tasks addressed by these examples are score following of melodic and polyphonic (keyboard) instruments, identifying the correlation between a jazz improvisation and a chord progression, beat tracking, and analysing student performance errors.

These are but a few of many tasks that might be automated, and there is certainly room for improvement in the areas already mentioned. The problem of following an ensemble has not been studied, and following vocal music has not been studied carefully. Voices tend to exhibit much more variation in pitch and articulation than instruments. The incorporation of learning into these and other Music Understanding systems is an important future direction that has not yet been explored.

Music Understanding is an important factor in the process of developing computer-based music systems. We are reaching a point where computer music systems are limited by their human interface, and more sophisticated interfaces can only develop when computer systems understand music concepts. Without understanding capabilities, computer music systems can only automate the mechanical aspects of music making, and the full potential of computers in music will not be realized.

Music Understanding systems also have an important role in developing and studying theories of cognition. The goal of developing music understanding systems encourages us to explore our own cognitive capacities and mechanisms. Also, computers provide an objective way to test theories and models of music cognition. Finally, formal studies of music understanding lead us to more formal models of music. This may lead to interesting new music theories.

In summary, Music Understanding is a new but important facet of computer music research today. Research in Music Understanding promises improved interfaces to computer music systems, new capabilities, advances in cognitive psychology, and new developments in music theory.

8. Acknowledgments

This paper is based on a talk prepared for the International Wenner-Gren Symposium on Music, Language, Speech, and Brain. 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. The Piano Tutor concept is due to Marta Sanchez and Annabelle Joseph, and the Piano Tutor was developed with the efforts of Peter Capell, Ron Saul (who implemented most of the analysis system), and Robert Joseph. Software was also contributed by John Maloney, and Hal Mukaino offered many design suggestions. Hal also implemented the best polyphonic accompaniment system to date.

This work has been made possible largely through the Carnegie Mellon University School of Computer Science and was partially supported by Yamaha (computer accompaniment) and the Markle Foundation (the Piano Tutor).

References

[Allen 90] Allen, P. E. and R. B. Dannenberg. Tracking Musical Beats in Real Time. In S. Arnold and G. Hair (editor), *ICMC Glasgow 1990 Proceedings*, pages 140-143. International Computer Music Association, 1990.

[Bloch 85] Bloch, J. J. and R. B. Dannenberg. Real-Time Computer Accompaniment of Keyboard Performances. In B. Truax (editor), *Proceedings of the International Computer Music Conference 1985*, pages 279-290. International Computer Music Association, 1985.

[Chafe 82] Chafe, Chris, Bernard Mont-Reynaud, and Loren Rush. Toward an Intelligent Editor of Digital Audio: Recognition of Musical Constructs. *Computer Music Journal* 6(1):30-41, Spring, 1982.

[Dannenberg 84] Dannenberg, R. B. An On-Line Algorithm for Real-Time Accompaniment. In W. Buxton (editor), *Proceedings of the International Computer Music Conference 1984*, pages 193-198. International Computer Music Association, 1984.

[Dannenberg 87] Dannenberg, R. B. and B. Mont-Reynaud. Following an Improvisation in Real Time. In J. Beauchamp (editor), *Proceedings of the 1987 International Computer Music Conference*, pages 241-248. International Computer Music Association, San Francisco, 1987.

[Dannenberg 88] Dannenberg, R. B. and H. Mukaino. New Techniques for Enhanced Quality of Computer Accompaniment. In C. Lischka and J. Fritsch (editor), *Proceedings of the 14th International Computer Music Conference,* pages 243-249. International Computer Music Association, San Francisco, 1988.

[Lifton 85] Lifton, J. Some Technical and Aesthetic Considerations in Software for Live Interactive Performance. In B. Truax (editor), *Proceedings of the International Computer Music Conference 1985*, pages 303-306. International Computer Music Association, 1985.

[Longuet-Higgins 82]
Longuet-Higgins, H. C. and C. S. Lee. The Perception of Musical Rhythms. *Perception* 11:115-128, 1982.

[Vercoe 85] Vercoe, B. and M. Puckette. Synthetic Rehearsal: Training the Synthetic Performer. In B. Truax (editor), *Proceedings of the International Computer Music Conference 1985*, pages 275-278. International Computer Music Association, 1985.

Table of Contents

List of Figures