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. 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 improviser in a cyclic chord progression such as the 12-bar blues. 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. The fourth project, the Piano Tutor, identifies student performance errors and offers advice. The fifth project studies human tempo tracking with the goal of improving the naturalness of automated accompaniment systems.

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]Published as: Dannenberg, ‘‘Music Understanding By Computer,’’ ***IAKTA/LIST International Workshop on Knowledge Technology in the Arts Proceedings***, Osaka, Japan: Laboratories of Image Information Science and Technology, pp 41-56 (September 16, 1993).

The following sections will describe a number of Music Understanding skills. The first two provide the basis for the responsive synchronization of a musical accompaniment. One of these skills 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 other synchronization 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. The fourth skill is error diagnosis and remedial feedback to piano students, accomplished in the Piano Tutor. The fifth skill concerns music synchronization: How is it that performers adjust their tempo or score position to synchronize with another player?

The first part of this paper is taken almost verbatim from an earlier report [Dannenberg 91a]. That report is extended here with current information. It should be noted that this paper focuses almost entirely on work by the author with various students and colleagues. It should not be interpreted as a survey of the state of the art. Due to space and time limitations, many interesting research has been ignored.

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 pre-composed 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, Baird 93] 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.

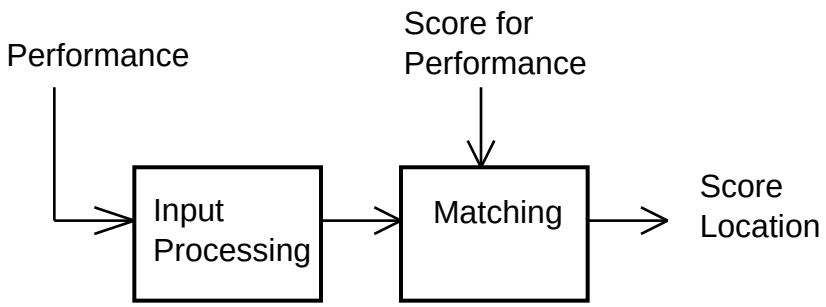

Figure 1: Block diagram of a score following system.

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 occurrence 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 83]. 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.

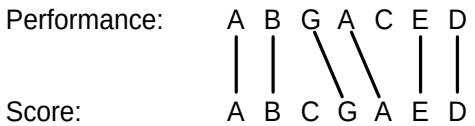

Figure 2: The correspondence between a score and a performance.

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. In contrast to the previously cited accompaniment systems, the matcher designed by the author can easily match sequences with 20 or more pitches, making it very tolerant of errors.

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 85] 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. [Dannenberg 91b] The matcher has been extended 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. A commercial Computer Accompaniment system derived directly from the author’s work was announced in January, 1993, and should be available in the fall of 1993.

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 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. 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 is 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 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.

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

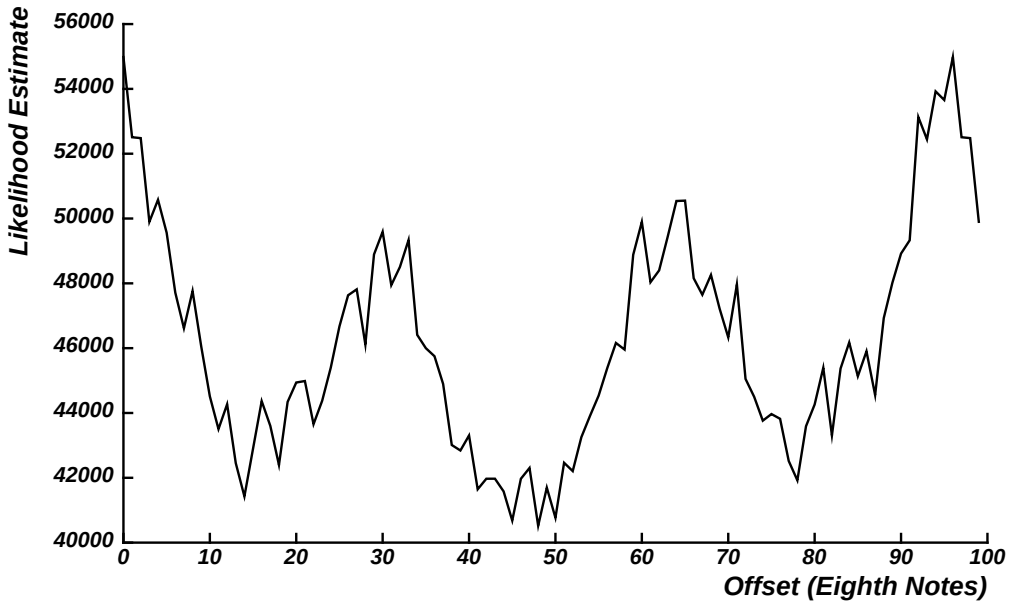

Figure 3: Likelihood estimates of the solo starting at different offsets in a 12-bar blues progression.

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. [Dannenberg 91b] When the harmonic structure is not so obviously outlined, the correlation peaks are not so distinct. The peaks become sharper as more measures of input are analyzed, but requiring many measures of input makes the technique unsuited to real-time performances.

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 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.

Paul Allen and I used a more elaborate approach to overcome this problem of losing synchronization. [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 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.

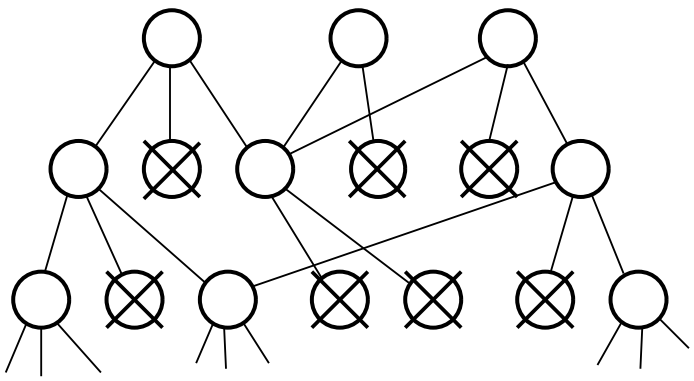

Figure 4: Three levels of beam search.

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). These heuristics are implicit in previous foot tappers, which only consider one interpretation and need not give ratings to alternatives.

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.) In addition to the interpretation with the highest rating, the beam search retains and continues to explore alternatives with lessor ratings. 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.

Although this is not intended as a cognitive model, it was introspection that guided us to this approach. When listening to performances, it seems to the author that the rhythmic interpretation is sometimes ambiguous, and that sometimes it is necessary to reinterpret previous notes in the context of new information. This ability to consider multiple interpretations is the key idea behind our new approach.

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.

New directions that we think are promising include trying to refine the heuristics used to evaluate an interpretation. Learning and classification techniques might be used here. Another promising direction is to use harmonic or other information to help rate various interpretations.

5. The Piano Tutor

The projects described above concern basic music listening skills. In the Piano Tutor project [Dannenberg 90], we attempted to capture the knowledge and skills of a piano teacher, a problem that is in some ways much more difficult, but in other ways actually simpler.

The Piano Tutor is a research project undertaken by Marta Sanchez, Annabelle Joseph, Peter Capell, Ronald Saul, Robert Joseph and the author at Carnegie Mellon University. The Piano Tutor combines an expert system with

multimedia technology to form an interactive piano teaching system. [Sanchez 90] Important elements of the Piano Tutor are:

- The use of score-following software to interpret student performances,
- The use of extensive multimedia to create a natural dialog with the student,
- An expert system to analyze student mistakes and give pertinent multimedia feedback, and
- The use of Instructional Design theory to develop an extensive curriculum that can be tailored automatically to individual student needs.

Figure 5 illustrates a block diagram of the system. In normal operation, the Piano Tutor presents new information to the student; that is, it ‘‘teaches’’ something. Then, the student is asked to apply the new knowledge or skill in a musical exercise. The system compares the student performance to a model performance and develops a response. The response indicates what (if anything) the student did wrong and what to do next. The student performs the exercise again, and this interaction repeats until the exercise is mastered or the system decides the student needs to work on some easier material.

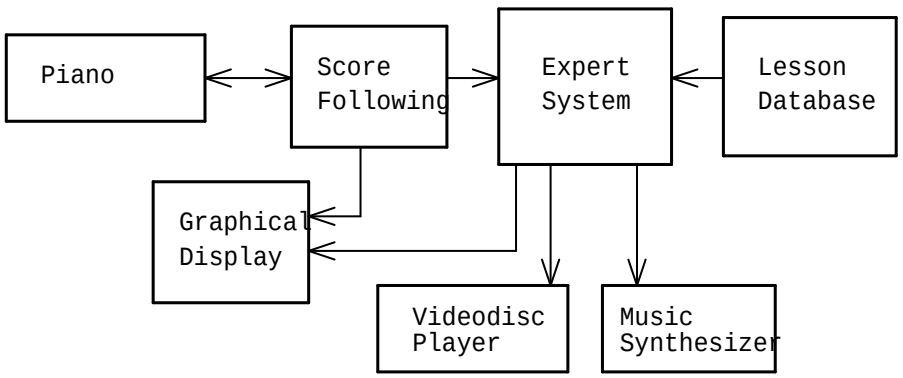

Figure 5: The Piano Tutor.

The basis for music understanding in the Piano Tutor is the score-following technology described in Section 2. In the Piano Tutor, score following is used to match student performances against a stored model performance. Once the scores are matched, the Piano Tutor can estimate the student’s tempo, and from that calculate the duration of each note in beats (as opposed to seconds). A discrimination network is used to identify the most significant error(s) in the performance and to develop a high-level explanation for the error. For example, if a note is held for two beats instead of one, the analysis will discover that the note is held too long. This error (‘‘too long’’) is refined to a more specific error (‘‘two beats instead of one’’) as the analysis continues.

Music understanding in the Piano Tutor is an essential component of the system. A key element of good teaching is the idea of ‘‘active learning’’: a student who is actively engaged performing a learning task will learn faster than a student who is passive. Activity on the part of the student necessitates an understanding and analysis capability on the part of the teacher. (The difficulty of understanding and analyzing the performance of a group of students is one reason why classroom instruction tends to be passive and less successful than private instruction.) Music understanding in the Piano Tutor allows the system to support active learning where the student is given helpful feedback during, or immediately after, each performance.

One of the interesting elements of interaction in the Piano Tutor is the sort of ‘‘conversation’’ that develops between teacher and student. [Dannenberg 92] This is not a natural language dialog, but it is nevertheless a two-way interaction. The student communicates to the computer teacher by performing specified tasks. The machine responds with multimedia presentations, most often pre-recorded voice and highlighted notes in the graphical music display. The fact that the Piano Tutor is responding to specific actions of the student gives him or her a strong impression that the Piano Tutor is quite intelligent.

Our experience gives us some idea of ‘‘how smart’’ a piano teaching system must be to engage the student in a meaningful and effective dialog. On the one hand, the system must be able to relate student performances to model performances and detect the differences. It must also decide which errors are most important and worth pointing out to the student. Finally, it must give feedback to the student within the context of the task at hand; for example, relating the error to the previous examples or avoiding terminology that has yet to be taught.

On the other hand, the system does not need to be intelligent on a human scale. Since lessons provide a very specific context, the performance input will tend to deviate from the score in limited and fairly predictable ways. When there is a large deviation, it is acceptable to simply ask the student to try again. This is not an adversary game where the computer tries to out-think a human, but a cooperative dialog, where the student and computer have compatible goals. Another simplifying factor is that lessons are selected by the Piano Tutor rather than the student. Since the Piano Tutor is generally in control, it always understands the context in which the human-computer dialog is taking place.

The way in which lessons are selected is also interesting. Lessons have *prerequisite skills*, which the student should have before taking the lesson, and *objective skills*, which the lesson teaches. Normally, the objectives of one lesson will be prerequisites to other lessons. The Piano Tutor maintains a *student model* which reflects the skills that the student is believed to possess. The student model is used to find lessons that the student is prepared to take, and the model is updated as the student masters lessons. This approach [Capell 93] turns out to have little to do specifically with music, and we are starting to build a tutoring system for Computer Science using the same representation and lesson selection mechanism.

6. The Psychology of Musical Accompaniment

Throughout the years of building computer systems to follow human performers, a nagging question has been: How do human accompanists behave? There has been very little research relevant to this question, so Michael Mecca and I set out to find some answers. It is ironic that to learn how to make machines follow humans, we decided to have humans follow machines. To be specific, we asked human accompanists to follow machine-sequenced music which we carefully altered in order to study how humans respond to tempo changes and other timing deviations.

Thus far, we have conducted only a pilot study [Mecca 93], so there is much more work to be done, but even the preliminary results are quite interesting. Five experiments were conducted:

- Playing scales. This experiment characterized the accuracy with which humans could play along with a steady stream of quarter notes.
- Catching up. After a long rest, the computer comes in early to see how human accompanists catch up.
- Tempo change before a rest. The computer changes tempo before a rest and we observe how the human's tempo continues to change.
- Displaced notes. A few notes are displaced in time from their nominal position and the human's response is observed.
- Large tempo change. The computer changes tempo by a large amount, and the human's response is observed.

The results of these experiments are summarized below.

In any human performance, there will be some amount of variation due to ''noise'' of the motor and nervous system. The first experiment helps us to estimate this variation by giving the subjects a simple accompaniment task. Standard deviation in timing ranged from 5.4ms to 94ms, and lower deviations were correlated with musical training.

In the remaining experiments, songs by Schubert were used. Subjects were given the music to practice, and all subjects could play the accompaniment without great difficulty. Subjects were asked to play the piano accompaniment while the melody was performed by a computer sequencer. The performances were recorded via MIDI and were analyzed afterward.

For the second experiment, the computer enters early after a four-measure rest. When the accompanist discovers the melody is ahead, he or she chooses one of two basic strategies: the *speedup catchup* strategy in which the accompanist races ahead to catch up with the melody, and the *skip and search* strategy, in which the accompanist stops, finds the correct location in the score, and begins to play at the new location.

We found that the *speedup catchup* strategy was preferred by more skilled players and when the time discrepancy is small. If the player had less skill or the melody was farther ahead, the *skip and search* strategy was used. The majority of the subjects used *speedup catchup* for a time difference of 667ms, while the

majority used *skip and search* for a time difference of 1333ms.

The third experiment was intended to measure an accompanist's tendency to continue an acceleration started by the melody. The melody tempo was increased or decreased slightly before a rest, forcing the accompanist to guess how to continue the tempo. When tempo increased, the accompanists would initially fall behind. Rather than catching up to the new tempo, accompanists would pick a new tempo between the new one and the original one, as if the subjects half-expected the tempo to return. Alternatively, the accompanists might be choosing a new tempo using sort of long-term average, resulting in the observed intermediate tempo. A similar behavior was observed in the decreasing tempo case.

In the fourth experiment, notes were displaced in time. This is indistinguishable from a momentary tempo change, and as might be expected, subjects responded with a tempo change between the original and the new implied tempo.

The fifth experiment examined the human response to large instantaneous tempo changes. We expected either a rapid jump to the new tempo implied by note inter-onset timing or some sort of critically-damped rapid convergence to the new tempo. What we observed instead is a slow oscillation around the new tempo as the accompanist repeatedly overshoots the target tempo and then overcorrects. Figure 6 illustrates data from one subject, a trained piano accompanist. This behavior is common across the highly skilled and less skilled players in our experiment.

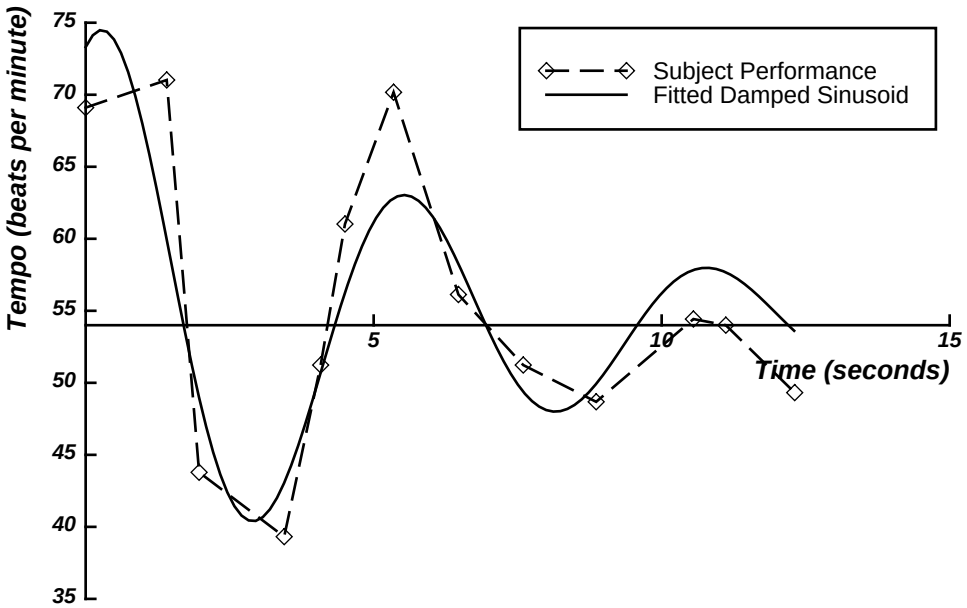

Figure 6: Accompanist's tempo variation in response to an instantaneous melody tempo change.

In Figure 6, we found the best fit of a damped sinusoid to the implied tempo curve. From the fitted curve, we obtain interesting parameters. The half-life of the curve, the time it takes for the oscillation to decay to one-half of its amplitude, is

4.44s, and the period of oscillation is 5.25s.

These are surprisingly large numbers. How does this tempo variation translate into absolute synchronization? Consider the negative-going dip in tempo between 2 and 4 seconds in Figure 6. This dip represents a transition from being ahead by some amount to being behind. The dip has an area of approximately 0.36 beats, indicating the accompanist slowed from being, say 0.2 beats ahead, to 0.16 beats behind. At a tempo of 54 beats/minute (the melody tempo), an error of 0.2 beats corresponds to 222ms. If the exponential model is correct, it will take about 13 more seconds for an oscillation of 222ms to decay to 30ms, which is at the level of normal random timing variation.

This study is only preliminary and raises as many questions as it answers. It has led to the hypothesis that accompanists have one cognitive resource for tempo following and tempo generation. That would explain the observation that musicians can count off a tempo and begin playing in tight synchrony, yet cannot quickly adjust to tempo changes. The explanation is that the ‘‘tempo’’ resource is available during the countoff before the performance begins, but it is in constant use during a performance and cannot be used for listening. Some other cognitive mechanism seems to be required for accompaniment where listening and performing must be simultaneous, and this is the source of the oscillations. These ideas are pure speculation and should be tested experimentally.

7. Summary and Conclusions

We have discussed a number of systems for Music Understanding. Each one has been designed to recognize pattern or structure in music in order to perform a musical task. Systems have been described that provide automatic synchronization of accompaniment, accompaniment of jazz improvisations, beat and tempo identification, and remedial feedback to student performers. We have also considered some research on human performance.

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. The foot tapping problem is far from being 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 aesthetics.

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. The study of human accompaniment has turned out to be far more interesting than expected, and many new experiments are needed to characterize and understand this aspect of human behavior. Many other musical tasks remain to be studied. Perhaps they hold many more interesting discoveries.

8. 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. Michael Mecca ran and analyzed the experiments on human accompaniment. This work has been made possible largely through the Carnegie Mellon University School of Computer Science and was partially supported by Yamaha.

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.

[Baird 93] Baird, B., D. Blevins, N. Zahler. Artificial Intelligence and Music: Implementing an Interactive Computer Performer. *Computer Music Journal* 17(2):73-79, Summer, 1993.

[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.

[Capell 93] Capell, P. and R. B. Dannenberg. Instructional Design and Intelligent Tutoring: Theory and the Precision of Design. *Journal of Artificial Intelligence in Education* 4(1):95-121, 1993.

[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.

[Dannenberg 90] Dannenberg, R. B., M. Sanchez, A. Joseph, P. Capell, R. Joseph, and R. Saul. A Computer-Based Multi-Media Tutor for Beginning Piano Students. *Interface* 19(2-3):155-73, 1990.

[Dannenberg 91a] Dannenberg, R. B. Recent work in real-time music understanding by computer. *Wenner-Gren International Symposium Series, Vol. 59. Music, Language, Speech and Brain.* In J. Sundberg, L. Nord, and R. Carlson, Macmillan, London, 1991, pages 194-202.

[Dannenberg 91b] Dannenberg, R. B. Computer Accompaniment and Following an Improvisation. *The ICMA Video Review.* International Computer Music Association, San Francisco, 1991. (Video).

[Dannenberg 92] Dannenberg, R. B. and R. L. Joseph. Human-Computer Interaction in the Piano Tutor. *Multimedia Interface Design.* In Blattner, M. M. and R. B. Dannenberg, ACM Press, 1992, pages 65-78.

[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.

[Mecca 93] Michael T. Mecca. Tempo Following Behavior in Musical Accompaniment. Master's thesis, Department of Logic and Philosophy, Carnegie Mellon University, 1993.

[Sanchez 90] Sanchez, M., A. Joseph, R. B. Dannenberg, P. Capell, R. Saul, and R. Joseph. The Piano Tutor. *ACM Siggraph Video Review.* Volume 55. *CHI '90 Technical Video Program - New Techniques.* ACM Siggraph, c/o 1st Priority, Box 576, Itasca, IL 60143-0576, 1990. (Video).

[Sankoff 83] Sankoff, David and Joseph B. Kruskal, editors. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison.* Addison-Wesley, Reading, Mass., 1983.

[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.