Automating Ensemble Performance

Lorin Grubb
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
lgrubb@cs.cmu.edu

Roger B. Dannenberg
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
rbd@cs.cmu.edu

Abstract

This paper describes a system that can provide accompaniment for an ensemble of performers. The system tracks the performance of each ensemble member to determine score position and tempo. The individual score positions and tempi are then combined and resolved in order to estimate position and tempo of the ensemble. Using this information, the system produces an accompaniment synchronized to the live performers. The system is able to interact with the ensemble, even as individual members rest and re-enter the performance, become lost, or make mistakes. Demonstrations of the system have been undertaken with ensembles having as many as four live performers, using a combination of electronic and acoustic instruments.

Keywords: Interactive performance systems, Artificial intelligence in music, Real-time systems, MIDI applications

1 Introduction

When we participate as performers in an ensemble, we listen to many independently controlled lines of music. If all performers were perfect, then coordinating and synchronizing with an ensemble would be no more difficult than performing with a polyphonic instrument such as a piano. In reality, ensemble players are not necessarily very well synchronized, and some players may become lost or consistently drag or rush the tempo. Even when the performance goes well, individual players will rest and rejoin the ensemble as indicated in the score. During contrapuntally active sections of a composition, some performers may play moving lines while others sustain tones. An ensemble player must integrate this information and resolve contradictions in order to form a sense of the true ensemble tempo and score position. This paper describes a working model of an ensemble player capable of performing composed music in synchrony with live musicians.

Several systems for following and accompanying *solo* performers have previously been developed and described in the computer music literature [Dannenberg,1984][Vercoe,1984][Baird *et al.*, 1993]. The problem of solo accompaniment can be partitioned into three distinct subproblems: 1) reliably detecting what the soloist has performed, 2) determining the score position of the soloist from the detected performance, and 3) producing an accompaniment in synchrony with the detected performance. An accompaniment system which attempts to accompany a soloist responsively must address each of these tasks.

Likewise, a system which attempts to responsively accompany an ensemble of live performers must also address each of these tasks. Furthermore, the system's capabilities must be extended beyond those of the solo accompaniment system so that it is able to track multiple performers simultaneously. This corresponds to multiple, simultaneous execution of reliable performance detection and determination of score position. Before taking actions to control production of the accompaniment, an ensemble performer must also integrate the information derived from tracking each performer. This may require resolution of discrepancies, such as different estimates of individual performers' score positions. In this paper, we first provide a detailed consideration of the problems involved in tracking multiple performers and combining the available information in order to control an accompaniment. Next, we describe the approach taken by our ensemble performer in order to solve each of the problems faced by an ensemble accompaniment system. Finally, we offer a qualitative examination of the results of using our ensemble performer to accompany actual ensembles of live performers.

2 Problem Description

The first task faced by an accompaniment system is reliably detecting what has been performed by the live musicians. The goal here is to extract from the performance important musical parameters that can be used to determine the score position and tempo of the performer. Such parameters might include fundamental pitch, note duration, attack, dynamic (relative loudness), and articulation. The precise parameters obtained by a particular system may vary according to the type of performance it tracks, how that performance is represented, and the expense and reliability with which certain parameters may be extracted from the representation. In the simplest case, MIDI messages sent from an electronic keyboard can provide very reliable, clean, and precise information about the pitch, duration, and dynamic of the performance. This information can be inexpensively extracted from such a representation.

In a more difficult case, the representation of the performance might be an audio signal from a microphone. This is often the case when tracking performances from acoustic wind instruments. A system which must extract parameters from a representation like this will need to devote more computation time to analysis of the signal and deal with issues like background noise, and distinguishing between signals received from attacks versus sustained pitches. A yet more difficult case is to distinguish multiple instruments recorded with a single microphone [Kashino and Tanaka, 1993]. For performances from different instruments, the system may need to track different parameters. In the case of tracking vocal performances, information about phonemes can be useful, as might syllables or words if they can be reliably recognized [Inoue *et al.*, 1993]. The performance detection and representation task is common to all accompaniment systems, whether they accompany soloists or ensembles, either by playing from a pre-composed score or by improvising.

The second task of an accompaniment system is tracking the score position of performers in real-time. This involves matching sequences of detected performance parameters to a score. The "score" might exist in a variety of forms, including a completely composed piece or simply an expected harmonic progression. Several considerations complicate tracking score location of performers. First, the tracking needs to be accomplished efficiently so that the system is able to control the accompaniment in real-time. The more quickly a system can recognize that a soloist has entered early, for example, the more quickly it will be able to adjust the accompaniment performance to accommodate. Additionally, since a flawless performance is not guaranteed, the tracking process must be tolerant of extraneous parameters as generated by an occasional wrong note, extra note, or omitted note. Finally, the method chosen to track the performer must use, in an appropriate manner, the parameters extracted by the performance analyzer. For example, if the performance analyzer can reliably recognize pitched signals and extract the fundamental, but is not so reliable at recognizing attacks, the tracking system should be appropriately less dependent upon the attack information.

If successive score locations can be accurately identified in the performance and time-stamped, then the accompaniment system may be able to derive accurate tempo predictions. This is accomplished by comparing actual time differences between performed events and expected time differences between corresponding score events. Since it is well-known that performers alter durations of notes for expressive purposes [Desain and Honing, 1992], accompaniment systems must apply some method of averaging successive time difference comparisons to avoid sudden, drastic tempo changes. Conversely, this averaging must not be so extreme as to hinder the system from reacting to actual, expressive tempo changes initiated by the performers. Reacting too slowly or too hesitantly to such legitimate changes can noticeably detract from the overall performance.

Once estimates of individual performers' score locations and tempi are obtained, an ensemble accompaniment system must combine and resolve this information. Before the system can make adjustments to the accompaniment performance, attempting to synchronize with the ensemble, it must have an idea of the overall ensemble score position and tempo. Several considerations affect generation of these estimates. First, performers who are not tracked reliably should not strongly influence the final estimates. This would be the case if a performer's input signal is noisy and difficult to analyze, or for performers who make numerous mistakes (wrong notes, omitted notes, etc.) such that their score position cannot reliably be determined. Second, reliable information obtained from more recently active performers should influence final estimates more strongly. For example, less consideration should be given to performers who have either a sustained note or a rest for a long duration while other performers play active lines. The latter performers are more likely to give the most accurate estimates of current ensemble score location and tempo. Finally, ensemble estimates should be less influenced by information from performers whose score position is relatively distant from the majority of the ensemble. Such performers are presumably lost, possibly having fallen behind or made an entrance too early. As a result, they offer an invalid ensemble score position and are also likely to change their tempo drastically in order to rejoin the group.

Having obtained estimates of ensemble score position and tempo which satisfy these considerations as much as possible, an accompaniment system must then decide when and how to adjust the accompaniment performance. Generally, an accompaniment must be continuous and aesthetically acceptable, yet reactive to the performers' omissions, errors, and tempo changes. If the performers increase or decrease the tempo for interpretive reasons, the accompaniment system should do likewise. If the performers pause or jump ahead in the score, then the accompaniment should follow as much as possible, but should always sound "musical" rather than "mechanical". The system must react to the performers' actions in a generally expected and reasonable fashion. Accomplishing this is, of course, strongly dependent upon the reliability of the tracking system as well as the entire system's ability to operate in real-time.

3 Approach

The ensemble performer we have developed provides one solution to the ensemble accompaniment problem. It is based upon a system for accompanying solo performers previously described in [Dannenberg, 1984]. It assumes the availability of a score that has all parts explicitly and completely written out for all performers.

The system extracts musical parameters from a performance represented by a sequence of MIDI messages. It makes use of pitch information provided in these messages, as well as their arrival time. For electronic instruments, such as MIDI keyboards, this information can be easily and reliably obtained directly from the instrument. In the case of acoustic wind instruments, we use IVL Pitchrider 4000 pitch-to-MIDI converters that transform microphone input into a MIDI message stream. For vocal input, we have developed a software preprocessor that applies some simple, heuristic statistical techniques for cleaning up the attack and pitch information provided by the same pitch-to-MIDI device. This is necessary since these devices are tuned to recognize notes in performances by wind instruments, which tend to have more consistent, easily recognized attacks and a more steady, precise pitch than vocal performances.

To track the score position of individual performers, the system uses a dynamic programming algorithm to match the parameters extracted from the performance against the score. In practice, a prefix of the performance (what has been played up to present) is matched against a complete sequence found in the score. The basic algorithm works exclusively with the pitches of the recognized notes. The objective of this algorithm is to find the "best" match between performance and score according to the evaluation function:

$$\textit{evaluation} = a \times \textit{matched notes} - b \times \textit{omissions} - c \times \textit{extra notes}$$

The matching algorithm is applied on every recognized note in a given individual's performance. Although the number of ways the performed pitches can be matched against the score is exponential in the number of performed notes, dynamic programming allows us to compute the best match in time that is linear in the length of the score, and which gives a result after each performed note. By using a "window" centered around the expected score location, the work per performed note is further reduced to a constant. A more detailed presentation of the matcher's algorithm can be found in [Bloch and Dannenberg, 1985], which also shows how to modify this algorithm to handle polyphonic performance input (e.g., chord sequences played on a keyboard).

A score position is posited for each performer on every note input received from that performer. Each position is recorded in a buffer along with a timestamp indicating the real time when that location was reached by the performer. If successive score positions for a given performer are plotted versus the corresponding real time, the tempo of the performance at any point is given by the slope of the graph (since tempo is the amount of score traversed in a unit of real time). As previously mentioned, it is necessary to apply some form of averaging over the individual tempi found at successive points in the graph. While many averaging techniques are available, we have elected to simply take the slope of the line between the first and last points in the location buffer. Since the buffer size is limited and relatively small, with older entries discarded one at a time once the buffer's capacity is exceeded, this tempo estimation is responsive to actual changes in tempo but less "jerky" than estimates based solely on the two most recent buffer entries. This method of averaging is also more expedient than calculating the true mean tempo or applying linear regression. In practice, it has worked well. If the tracking system detects an error or leap in the performer's score position (i.e., the matcher cannot conclusively identify a score position for the performer), the buffer is emptied and no tempo estimates for that performer are possible until the buffer is replenished.

Since the ensemble accompaniment system must track multiple performers simultaneously, separate instances of a match state and score location buffer are maintained. Performance information from different individuals is identified by different MIDI channels, allowing the ensemble performer to distribute input to the appropriate tracking system. Since score location information for each performer is available, it is possible to estimate each performer's current score location at any time, providing the matcher has been able to follow that performer. For example, consider Figure 1. If at time t_1 the performer is at score location s_1 and maintaining an estimated tempo of 0.5; then at time t_1+2, the performer's expected score location would be s_1+1 (if no intervening input is received from the performer). This enables the ensemble system to estimate score positions for every ensemble member for the same point in time, regardless of when the system last received input from that performer.

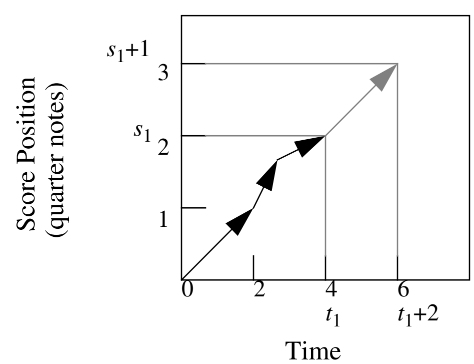

Figure 1, Estimating score position.

The various estimates obtained from each tracking system must be consolidated into a single ensemble score position and tempo. The accompaniment system estimates an ensemble score position and tempo on every input from every performer. To accomplish this in accordance with the criteria previously discussed, each pair of estimates from each tracking system is rated, and a weighted average computed form both score location and tempo estimates. The ratings give more weight to estimates which are more recent and less weight to estimates that do not cluster with estimates from other tracking systems. Figure 2 presents the rating function. The final rating (FR) used for the weighted average is the product of the squares of two independent ratings—a recency rating (RR) and a clustering rating (CR).

$$\text{Ensemble Score Position} = \frac{\sum_{i=1}^{n} FR(i) \times pos(i)}{\sum_{i=1}^{n} FR(i)}$$

$$FR(i) = (RR(i))^2 \times (CR(i))^2 + c$$

FR(i) = Final rating for estimate from tracking system i
RR(i) = Recency rating for estimate from tracking system i
CR(i) = Clustering rating for estimate from tracking system i
c = Very small constant to prevent FR from reaching zero

$$RR(i) = \begin{matrix} 1 - \dfrac{(rtime - ltime(i))}{3} \\ 0 \end{matrix} \quad \text{if} \quad \begin{matrix} rtime - ltime(i) \le 3 \\ rtime - ltime(i) > 3 \end{matrix}$$

rtime = Current time for which estimates are made
ltime(i) = Time of last match made by tracking system i

$$CR(i) = 1 - \frac{\left(\sum_{j=1}^{n} |pos(i) - pos(j)|\right) + |acc - pos(i)|}{n \times (pos(max) - pos(min))}$$

n = Number of active tracking systems
pos(i) = Score position for tracking system i
pos(j) = Score position for tracking system j
acc = Score position for accompaniment
pos(max) = Maximum of all pos(i), pos(j), and acc
pos(min) = Minimum of all pos(i), pos(j), and acc but NOT pos(max)

Figure 2, Function to calculate ensemble score position.

The recency rating (RR) for each tracking system decays from a value of one to zero during a three-second interval. If the score position buffer of the tracking system is empty, then the recency rating is zero. This value is squared in the final rating product, causing the final rating to decay more rapidly (in a quasi-exponential fashion) over the three-second interval. This rating is designed to give preference to the most recently active performers, thereby making the accompaniment performance more reactive to recent changes in the score position and tempo.

The clustering rating (CR) characterizes the relative separation of voices. It is the ratio of the summed distance of the i'th voice from all other voices, divided by the maximum possible summed distance at the time the rating is generated. It indicates, on a scale from zero to one, how close a particular performer's score location lies to the location of the other performer's (i.e., the rest of the live ensemble and the accompaniment). If all performers (including the accompaniment) are at the exact same score position, all will have a clustering rating of one. As the score positions of the performers start to vary, their clustering ratings will fall below one. If their relative distances from one another remain similar, their clustering ratings will also remain similar. If one performer's distances from the others are much larger relative to their distances from one another (i.e., all but one form a relatively tight cluster), then the clustering ratings of the "cluster" members will remain relatively similar while the rating of the other performer will be significantly lower. If the cluster members in this case all have the exact same score position, then the other performer's clustering rating will be zero. The clustering rating is designed to discount information obtained from a performer whose score position is abnormally distant from the rest of the ensemble. Note that the current accompaniment position is considered by the clustering rating. This provides a slight bias toward performers who are currently synchronized with the accompaniment when the performers' ratings would otherwise be very similar. The clustering rating, like the recency rating, is squared in the final rating so as to give an even stronger preference to the tightly clustered performers.

The ensemble score position and tempo are calculated as a weighted average of the tracking system estimates. Each tracking system's estimates influence the ensemble estimates according to their relative satisfaction of the previously discussed criteria, compared to the estimates from the other tracking systems. Each estimate is weighted by its final rating. The final rating is a product of the squares of the recency and clustering ratings and is guaranteed to be less than or equal to the minimum of the individual squares (since the recency and clustering ratings range from zero to one). Thus, as the criteria characterized by the component ratings fail to be satisfied, the final rating decreases. For example, consider the score excerpt presented in Figure 3. As the first performer proceeds, the recency rating of the other, sustained or resting voices will decay. The tempo and score position estimated by the first performer's tracking system will quickly dominate the ensemble average, in turn causing the accompaniment to more closely follow the first performer.

Once the ensemble player has calculated ensemble score position and tempo estimates, it applies a set of accompaniment rules to adjust the accompaniment performance. These rules correspond to studies of how live accompanists react to similar situations encountered during a performance [Mecca, 1993]. The rules consider the time difference between the ensemble score position and the current accompaniment score position. If the time difference is less than a pre-determined "noise" threshold, then only the accompaniment tempo is modified to agree with the ensemble tempo. The noise threshold prevents excessive jumping and tempo alterations, since performers do make subtle alterations in note placement [Bilmes, 1992]. If the performer is ahead of the accompaniment by a difference at least as great as the noise threshold, the accompaniment will either jump to the ensemble score position or play at an abnormally fast tempo to catch up. The technique applied depends on the magnitude of the time difference. If the accompaniment is ahead of the ensemble by a time difference at least as great as the noise threshold, then the accompaniment will pause until the ensemble catches up. To prevent the accompaniment from continuing too far ahead of the performers, an input expectation point is maintained. If this point is passed without additional input from any performer, the accompaniment system pauses until additional input arrives.

The ensemble performer is implemented using the CMU MIDI Toolkit [Dannenberg, 1993] which provides MIDI message handling, real-time scheduling, and performance of MIDI sequences. It is possible to adjust the position and tempo of a sequence (score) performance on-the-fly as part of processing input or generating output. The ensemble system consists of four software components, implemented in an object-oriented programming style. Figure 4 diagrams their interconnection. The matcher receives performance input and uses dynamic programming to determine score location. The estimator maintains the score location buffer, calculates tempi, and generates estimates on request. A matcher-estimator combination forms a single performance tracking system. The ensemble performer can instantiate multiple tracking systems at initialization according to user-supplied specifications. The voter rates and combines multiple score location and tempo estimates to form the ensemble estimates. The scheduler uses these estimates to change the accompaniment performance according to the accompaniment rules. Only one voter and scheduler object are ever present, regardless of the number of tracking systems and live performers.

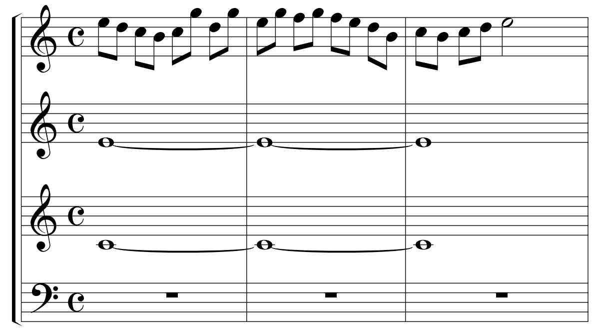

Figure 3, Score excerpt

4 Results

We have used the ensemble performer with live ensembles consisting of from one to four players, using both MIDI keyboards and acoustic wind instruments. The pieces performed range in difficulty from a simple canon on "Row, row, row your boat" to excerpts from Handel's *Water Music* and orchestral pieces by Mozart. In the case of accompanying a single performer, the system is highly reactive to tempo changes and tolerant of omitted notes, wrong notes, and extra notes. If too many mistakes, omissions, or extra notes appear in the performance (as in a heavily embellished rendition), the matcher becomes unable to correctly recognize the part. The occasional mistake from a competent performer does not present a problem. In general, for the case of following a soloist, the ensemble player performs much like the solo accompaniment system upon which it is based.

In the case of larger ensembles, the ensemble performer is able to simultaneously track all live performers. As certain members of the ensemble rest and reenter the performance, the system synchronizes well with the remaining, active members. Tempo changes of the latter performers are recognized by the system—readily so once the silent performers' recency ratings have decayed. If one performer should skip ahead or fall behind, or make an inappropriate entrance, the ensemble system will continue to synchronize with the other performers—ignoring the lost performer until he

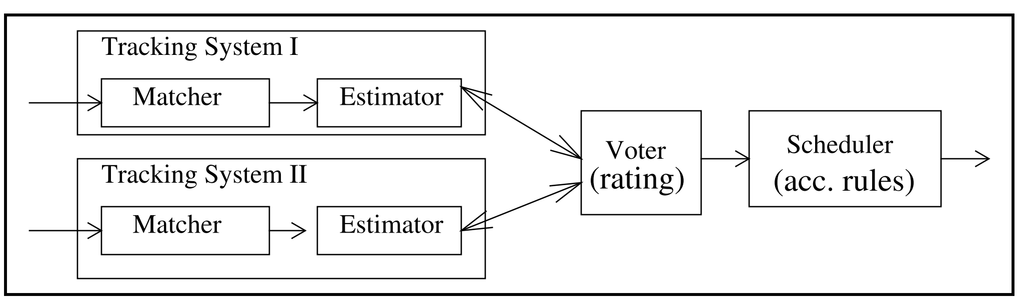

Figure 4, Components of the ensemble performer.

or she rejoins the ensemble or until the other performers rest. If all performers skip ahead in synchrony, then the accompaniment does likewise once the matchers recognize the new score location of the performers they track. When tracking multiple performers, the accompaniment is less affected by a single performer making mistakes, providing the other performers are accurate and in synchrony.

While the system works well with small ensembles, several difficulties must be addressed in order for the system to perform with larger groups. At the input level, for example, individual microphones in large ensembles will experience cross-talk, making pitch estimation more difficult. Obtaining reliable signals from individual members of such an ensemble may require highly directional, close-fitting microphones or more complicated techniques for analyzing the signals. Also, since MIDI only permits sixteen logical channels, the current system can only distinguish input from a maximum of sixteen performers. One way to deal with this might be to use a hierarchical input system and modify the message format. Intermediary processing nodes would monitor sixteen channels each and communicate with the ensemble performer using a message format that uniquely identifies each input source across all intermediary nodes. Alternatively, intermediary nodes might undertake some pre-processing on a section by section basis, reporting to the ensemble accompaniment system position and tempo of the first trumpet section, for example, rather than of each individual performer. Although handling large ensembles requires further investigation, we have no reason to believe that either of the two problems mentioned here is insurmountable.

Required computation time is another consideration, and is also important to scaling. Processing each input from each performer requires time linear in the ensemble size (since the estimates from every tracking system must be re-rated). In the worst case, if all parts simultaneously play a note, the amount of work completed before performance of the next note in the score is quadratic in the ensemble size. When running the ensemble performer on the slowest PC we have available, handling a single input for an ensemble of one requires 1.4 msec. The expense of recomputing one rating (for larger ensembles) is 0.3 msec. Based on these numbers, a conservative estimate indicates that we can process 16 inputs in 100 msec. A sixteenth note of 100 msec. duration implies a tempo of 150 quarter notes per minute. This is a fast tempo. If we were to update the voter once every 100 msec. instead of on every input, we could handle hundreds of instruments in real time with current processor technology. For large acoustic ensembles, computation time is likely to be dominated by signal processing of acoustic input.

5 Conclusions

We have presented the design of an ensemble accompaniment system that has been implemented and tested with small ensembles of electronic and acoustic instruments. This system provides a solution to each of the four problems faced by an ensemble performer: obtaining reliable performance input, tracking score position and tempo of individual performers, combining individual position and tempo estimates to form an ensemble score position and tempo, and considering ensemble estimates when deciding how to generate an aesthetically acceptable performance of the accompaniment.

Developing the ensemble performer has helped to define important criteria and considerations for following an ensemble performance. When generating score location and tempo estimates for an ensemble, it is useful to consider both the recency of input from individual performers and the relative proximity (or "clustering") among their score positions. This information helps to distinguish the active and reliable performers from the inactive or lost ensemble members, whose predictions do not accurately indicate the score position and tempo of the ensemble.

Testing of this system has revealed a trade-off between reactivity and stability of an accompaniment. The ensemble performer currently attempts to be reactive to the live performers. For example, if a significant majority of the ensemble suddenly skips ahead in the score, the ensemble performer will do likewise once it is able to identify their score position. This high reactivity could be questioned, since in some cases maintaining a stable accompaniment might be preferred. For example, if the majority of the ensemble is consistently dragging the tempo, as opposed to changing tempo for expressive purposes, the ensemble performer should perhaps maintain the original tempo more insistently rather than succumbing to the majority. This trade-off must be considered when designing both the rating functions used to calculate ensemble position and tempo, and the rules that control performance of the accompaniment.

Based upon the capabilities of the current ensemble performer, several avenues for additional research now exist. As previously mentioned, attempting to scale to tracking larger ensembles is one possibility. This will require development of methods for efficiently and reliably obtaining performance input, particularly with respect to acoustic instruments. Also, the only acoustic instruments participating in our ensemble testing have been wind instruments. Reliably obtaining performance input from string, percussion, or vocal ensembles may necessitate additional signal analysis techniques or handling of different performance parameters (such as phonemes or words, in the case of vocal performances).

Pre-performance analysis of the score may help the ensemble performer to develop appropriate performance expectations. Annotations in scores can provide useful performance hints to the scheduler [Dannenberg and Bookstein, 1991]. As a simple consideration, for example, if it is clear from the score that a particular performer should be inactive at present, then perhaps that performer's estimates should be ignored. This might make the system more immediately reactive to the contrapuntally active performers, as opposed to waiting for the inactive performer's recency rating to decay. This type of knowledge-based analysis might help the ensemble performer achieve a more subtle and precise control of the accompaniment performance. Automating this type of analysis process would, however, require significant musical knowledge pertinent to interpreting scores and performer actions.

Alternatively, we are also interested in experimenting with learning through rehearsal, possibly by using techniques similar in philosophy to those presented in [Vercoe and Puckette, 1985]. Ideally, an accompaniment system should be able to improve by practicing a piece with an ensemble and noting where to expect consistent tempo changes, embellishment, or performer error. Such practicing could be done with a single score played by a single ensemble, as well as with a single score performed by multiple ensembles. Since even two expressive performances by the same ensemble may vary greatly, the challenge will be to extract reliable characterizations of multiple ensemble performances and use them to enhance the accompaniment in successive performances, beyond what the naive and score-independent expectations permit. With the availability of the current ensemble accompaniment system, all of these techniques can now be empirically as well as theoretically examined.

So far, the ensemble system has been used only to fill in the missing parts of compositions written for an ensemble consisting exclusively of live musicians. Having demonstrated the system with this type of composition, we now believe that the system can offer composers the possibility of constructing very intricate and highly creative compositions intended to be performed by a group of live musicians and automated ensemble accompaniment system. Since the system is able to simultaneously track and respond to multiple performers, compositions scored for such an ensemble will allow live performers to apply a higher degree of freedom and expressiveness to their performance. At the same time, these compositions can take advantage of the rich source of sounds and techniques available only through computer performance. Composers can orchestrate more subtle and precise interactions and dialogue between the live performers and the computer, since the ensemble performer is able to make entrances and tempo changes based upon the entire performance of the live ensemble, as opposed to relying on elapsed time or simple cues. We are presently beginning to explore these possibilities.

Acknowledgments

This material is based upon work supported under a National Science Foundation Graduate Research Fellowship.

References

[Baird *et. al.*, 1993] B. Baird, D. Blevins, and N. Zahler. Artificial intelligence and music: implementing an interactive computer performer. *Computer Music Journal* 17(2): 73-9, 1993.

[Bilmes, 1992] J. Bilmes. A model for musical rhythm. In Proceedings of the 1992 International Computer Music Conference, 207-10, 1992.

[Bloch and Dannenberg, 1985] J. Bloch and R. Dannenberg. Real-time computer accompaniment of keyboard performances. In Proceedings of the 1985 International Computer Music Conference, 279-90, 1985.

[Dannenberg, 1984] R. Dannenberg. An on-line algorithm for real-time accompaniment. In Proceedings of the 1984 International Computer Music Conference, 193-8, 1984.

[Dannenberg, 1993] R. Dannenberg. *The CMU MIDI Toolkit.* Pittsburgh, Carnegie Mellon University, 1993.

[Dannenberg and Bookstein, 1991] R. Dannenberg and K. Bookstein. Practical aspects of a MIDI conducting program. In Proceedings of the 1991 International Computer Music Conference, 537-40, 1991.

[Desain and Honing, 1992] P. Desain and H. Honing. Tempo curves considered harmful. In *Music, Mind, and Machine: Studies in Computer Music, Music Cognition, and Artificial Intelligence*, Amsterdam, Thesis Publishers, 25-40, 1992.

[Inoue *et al.*, 1993] W. Inoue, S. Hashimoto, and S. Ohteru. A computer music system for human singing. In Proceedings of the 1993 International Computer Music Conference, 150-3, 1993.

[Kashino and Tanaka, 1993] K. Kashino and H. Tanaka. A sound source separation system with the ability of automatic tone modeling. In Proceedings of the 1993 International Computer Music Conference, 248-55, 1993.

[Mecca, 1993] M. Mecca. Tempo following behavior in musical accompaniment. Master's thesis, Carnegie Mellon University, 1993.

[Vercoe, 1984] B. Vercoe. The synthetic performer in the context of live performance. In Proceedings of the 1984 International Computer Music Conference, 199-200, 1984.

[Vercoe and Puckette, 1985] B. Vercoe and M. Puckette. Synthetic rehearsal: training the synthetic performer. In Proceedings of the 1985 International Computer Music Conference, 275-78, 1985.