

## Computer Performance in an Ensemble

Lorin Grubb and Roger Dannenberg  
 School of Computer Science  
 Carnegie Mellon University  
 Pittsburgh, PA 15213

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. Even when the performance goes well, players may rest and then rejoin the ensemble. During contrapuntally active sections of a composition, some performers may play moving lines while others sustain tones. An ensemble player must integrate this information to form a sense of tempo and position. Our study presents a working model of an ensemble player that can play composed music, synchronizing with human performers.

Constructing the model has led to a better understanding of what issues arise in ensemble performance and suggests possible strategies adopted by human performers.

### 1. Problem Description

The problem of automating accompaniment for a musical ensemble can be decomposed into four subtasks that the computer must successfully complete. The first of these tasks is detecting what the other ensemble players have performed. This requires the computer to obtain accurate representations of performance parameters, possibly including fundamental pitch, note duration, and dynamic (relative loudness). The precise parameters obtained by a particular system may vary according to the type of performance and the reliability with which those parameters may be extracted from the performance. For example, recognizing phonemes or syllables from a vocal performance might provide an important and useful characterization of that particular performance, providing recognition can be accomplished accurately and efficiently. 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 performance in real-time. This involves matching the detected performance to a score in order to track the performance (i.e., the score position of the player or players). The "score" might exist in a variety of forms, including a completely composed piece or simply an expected harmonic progression. Several factors complicate tracking performances to identify score location of the performer. First, the tracking needs to be accomplished efficiently so that the system is able to control the accompaniment in real-time. The more quickly the 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 or extra note. If successive score locations can be accurately identified and time-stamped, then the accompaniment system may be able to derive accurate tempo predictions by comparing actual time differences between performed events and expected time differences between corresponding score events. Since it is well-known that performers will alter durations of

notes for expressive purposes, the system must also be tolerant of these effects when tracking performance tempo.

Once estimates of individual performers' score location and tempi are obtained, an ensemble accompaniment system must combine this information to predict an ensemble score location and tempo. These ensemble predictions are used to make decisions on how the computer performance should be adjusted in order to best synchronize with the other performers. To obtain ensemble predictions, the accompaniment system must weigh the reliability and value of corresponding predictions for each tracked performer. Generating ensemble predictions sometimes demands resolution of conflicting individual predictions, since individual performers may or may not reliably indicate the activity of the ensemble. For example, as parts enter and drop-out of the performance, the last available predictions for some performers may be less recent (and potentially less characteristic of the ensemble) than others. Performers who become lost or make mistakes will provide less reliable estimates of ensemble score position. Similarly, parts which are less active in some portion of the performance (i.e., have many sustained notes) are less likely to provide tempo change information than parts which are more active during that same portion of the performance.

Automated accompaniment systems must also control performance of the accompaniment (the voices of the score not already performed by other players). The system must make decisions on how to adjust the position and tempo of the accompaniment in real-time. For an ensemble accompaniment system, these decisions must be based partly upon the ensemble score location and ensemble tempo. If the system recognizes that the ensemble's tempo is decreasing (perhaps indicating a *ritardando*), then the accompaniment system will need to adjust the accompaniment performance accordingly. This task requires that the system perform the accompaniment in an aesthetically acceptable manner, reacting to the performers' actions in a generally expected and "reasonable" fashion.

### 2. Approach and Implementation

The system discussed here provides one solution to the ensemble accompaniment problem. It is based upon a system for accompanying solo performers previously described in [Dan84]. Several other systems for accompanying soloists have also been developed, including [Ver84] and [BBZ93]. Our system takes MIDI messages as input and extracts performance parameters (such as pitch and duration) from a sequence of such messages. To generate these messages, we make use of commercially available MIDI keyboards and pitch-to-MIDI converters. The latter convert microphone input into a MIDI message stream. To track individual performers, the system uses a dynamic programming algorithm to match the sequence of input performance parameters against the known score (the expected parameter sequence). The algorithm works exclusively with the pitches of the recognized notes.

In effect, this process attempts to identify the current score location of the performer by maximizing the evaluation function:

$$\text{evaluation} = a * \text{matched notes} - b * \text{omitted score notes} - c * \text{extra performed notes}$$

Versions of this algorithm exist for tracking monophonic performances and polyphonic performances (such as from an electronic piano). A more detailed description of this algorithm can be found in [BID85]. A score position is posited for each performer on every note input received from that performer. The system maintains a buffer of the last several score locations of each performer along with a timestamp of when that prediction was

made. From these real-time timestamps and the corresponding score duration between these positions, the system can estimate the tempo of the performer. Though several possibilities exist for deriving such an estimate, we currently take the ratio of the total real-time duration of the buffer divided by the corresponding score time between the first and last positions in the buffer. This is a computationally inexpensive calculation that prevents the tempo predictions from being too "jerky", as compared to considering only the last two predicted score positions. In practice, it has worked well.

The accompaniment system generates an ensemble score position and tempo prediction on every input received from every performer. It then uses these predictions in conjunction with a set of accompaniment control rules to adjust the performance parameters of the accompaniment (including score position and tempo). The ensemble predictions are generated by taking a weighted average of the individual predictions for each performer. The system maintains the most recent predictions made by each performer's tracking system. The weight assigned to each pair of predictions from each performer is the product of two independent ratings: a recency rating and a clustering rating. The recency rating indicates, on a scale of 0 to 1 (right now), how recent the prediction is. This value is designed to decay in a quasi-exponential fashion over a short period of time (3-5 secs.). Its effect is to weight more strongly those predictions which are very recent and to virtually ignore those predictions close to or exceeding a pre-specified time threshold, usually 3-5 secs. This allows the system to follow active performers and ignore performers who are less active or resting.

The clustering rating indicates, on a scale of 0 to 1, how close a particular performer's score location is to those of the other performer's (i.e., the rest of the ensemble and the accompaniment). This value is related to the relative distance between performers. If all performers are at the exact same position, all will have a clustering rating of 1. As the score positions of the performers start to vary, their clustering ratings will fall below 1. If their relative distances from one another remain relatively similar, their clustering ratings will also remain similar. If one performer's distances from the others is much larger than their distances from one another (i.e., all but one form a relatively tight cluster in terms of score location), 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 all have the exact same score position, then the other performer's clustering rating will be 0. The clustering rating is designed to discount predictions obtained from performers who appear to be abnormally distant from the ensemble in terms of score location. Note that when calculating this rating for each performer's predictions, the accompaniment position is considered by the rating function. This is done to add a slight bias toward performers who are currently synchronized with the accompaniment when the performers' ratings would otherwise be very similar.

### 3. Results

The ensemble accompaniment system has been used to perform a variety of compositions with a variety of ensembles. The ensembles have consisted of from one to three performers, with instrumentation including MIDI keyboards and wind instruments. The compositions used have ranged from canons based on simple melodies to short orchestral works by Handel and Mozart. In the case of accompanying a soloist, the system is highly reactive to tempo changes and tolerant of omitted notes, wrong notes, and extra notes. In the case of accompanying multiple performers, the system is able to simultaneously track all performers. As individual performers drop-out or become lost, the

system is able to continue accompanying the other performers and to respond to their tempo changes. Actions taken by the system generally seem reasonable. Only in planned, degenerate cases (generally where people would recognize that the ensemble has "fallen apart") does the system behave in a "surprising" way. By creating and analyzing these situations, then modifying the system, we are working to give it a more human-like, expert musical behavior.

### 4. Conclusions

Developing this system has helped to define important criteria and considerations for automated ensemble accompaniment. When generating score location and tempo predictions 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, whose predictions are more desirable. The system has also shown that there exists a trade-off between the reactivity of an accompaniment system (how quickly it responds to performers' tempi and position changes) and its stability (how strongly the accompaniment maintains its current position and tempo). Testing has indicated that while the accompaniment should adjust to tempo and position changes, it should not attempt to follow all changes made by every performer. For example, if one performer in an ensemble of two (excluding the computer) starts to "drag" the tempo, the system should not attempt to follow this performer, but rather maintain the current tempo and stay with the correct performer. This trade-off must be carefully considered when constructing a method of integrating and resolving performance information obtained from multiple performers. This work shows that a skilled ensemble player must be fairly sophisticated to integrate conflicting information from multiple sources. While our current work offers a plausible model for this decision-making process, future work will continue to evaluate and improve the model. It would be interesting to refine the model further by measuring human performance in controlled ensemble performance experiments.

### References

- [BBZ93] B. Baird, D. Blevins, and N. Zahler. Artificial intelligence and music: implementing an interactive computer performer. *Computer Music Journal*, 17(2): 73-79, 1993.
- [BID85] J. J. Bloch and R. B. Dannenberg. Real-time computer accompaniment of keyboard performances. *Proceedings of the 1985 International Computer Music Conference*, 279-90, 1985
- [Dan84] R. B. Dannenberg. An on-line algorithm for real-time accompaniment. *Proceedings of the 1984 International Computer Music Conference*, 193-8, 1984.
- [Ver84] B. Vercoe. The synthetic performer in the context of live performance. *Proceedings of the 1984 International Computer Music Conference*, 199-200, 1984.
- Grubb and Dannenberg, "Computer Performance in an Ensemble," in 3rd International Conference for Music Perception and Cognition *Proceedings*, Liege, Belgium. July 23-27, 1994. Ed. Irene Deliège. Liege: European Society for the Cognitive Sciences of Music Centre de Recherche et de Formation Musicales de Wallonie, 1994. pp. 57-60, 1994.