

TWO APPROACHES FOR INTERACTION MANAGEMENT IN TIMBRE-AWARE IMPROVISATION SYSTEMS

William Hsu

Department of Computer Science
San Francisco State University
San Francisco CA 94132
USA

ABSTRACT

This paper describes two recent improvisation systems that incorporate timbre as an integral element in performance characterization and decision-making. The first *London* system uses high-level performance descriptors, adapted from recent work in mood detection in music, to coordinate low-level performance events. It was used in performances at LAM 2006 and NIME 2007. After careful evaluation of the results, we addressed the perceived shortcomings in the design of the *ARHS* system, which manages performance statistics hierarchically; this improved short-term responsiveness, as well as the ability to monitor and adapt to long-term performance variations.

1. INTRODUCTION

Timbre plays a relatively minor role in most automatic improvisation systems that have been described in the literature. Since 2002, we have built several interactive music systems that extract in real-time perceptually significant features of a saxophonist's timbral and gestural variations; this information is used to coordinate the performance of an ensemble of virtual improvising agents. In [1], we overviewed a system that recognizes timbral features, such as multiphonics. Timbral contours are tracked and referenced when synthesizing response material. Integrating timbral information into high-level decision logic was the focus of the next design iteration, the *London* system described in [2].

At the 2006 Live Algorithms for Music (LAM) conference, the *London* system participated in duo improvisations with saxophonists John Butcher and Evan Parker. At the 2007 NIME conference, the system worked with percussionist Sean Meehan. Based on these experiences, we identified a number of shortcomings, and undertook a significant re-design of the *London* system. This paper will present detailed evaluations of the *London* system, describe the re-designed *ARHS* system that addresses perceived shortcomings, and some initial test results.

In Section 2, we briefly survey related work in the area. Section 3 presents a short overview of the *London* system, and evaluates in detail its behavior in performances in London and New York. We examine the details of the re-designed *ARHS* system in Section 4, and form preliminary conclusions.

2. RELATED WORK

Interactive improvisation systems tend to focus on pitch, loudness, and rhythm; input from human musicians is usually filtered through a pitch-to-MIDI converter. Timbral information is largely limited to specifying a MIDI patch/instrument. The emphasis tends to be on using machine learning and similar techniques to coordinate response behavior. George Lewis' *Voyager* [3] compiles statistics of human performance captured through a pitch-to-MIDI converter. Dannenberg's system in [4] also takes input from a trumpet through a pitch-to-MIDI converter. Pre-recorded performances train neural networks to distinguish between styles such as "frantic" and "lyrical". More references are in [2].

[5] describes using FFT analysis to provide visual feedback for human improvisers during performance. The analysis data is broken into frequency bands, and mapped to control parameters for software instruments. The system for guitarist and four artificial performers in [6] focuses on event onset detection, pitch and other gestural information. Parameters such as "shyness", "sloppiness" and "keenness" are used to describe the high-level behavior of the artificial performers.

None of the systems surveyed here adequately address the crucial role that timbral events and timbral variation may play in a free improvisation.

3. OVERVIEW OF THE LONDON SYSTEM

Figure 1 shows a block diagram of the *London* system [2]. Timbral and gestural information is extracted from the human improviser's performance, and made available to the improvising agents. The Interaction Management Component (IMC) computes high-level descriptions of the performance, and coordinates the high-level behavior of the ensemble. We will present a brief overview of the IMC, evaluate performances involving the *London* system and three human improvisers, and outline shortcomings that will be addressed in the redesign. For more details, see [2].

3.1. Performance modes and IMC Operation

The Interaction Management Component (IMC) first computes high-level descriptors of the real-time performance; these will be consulted by the virtual ensemble. For our performance descriptors, we adapted some ideas from [7], which detailed a system for tracking the mood of traditional classical music clips.

We chose a simplified triple of features to form our performance mode descriptor: intensity (loud/soft), tempo (fast/slow), and for timbre, we chose an auditory roughness estimate based on [8]. Roughness is estimated by extracting partials from the audio; for each pair of partials, its contribution to the roughness measure is computed, and the roughness contributions of all pairs are summed. In our work with saxophone timbres [2], we have found that roughness is a useful measure that tracks overall timbral variation.

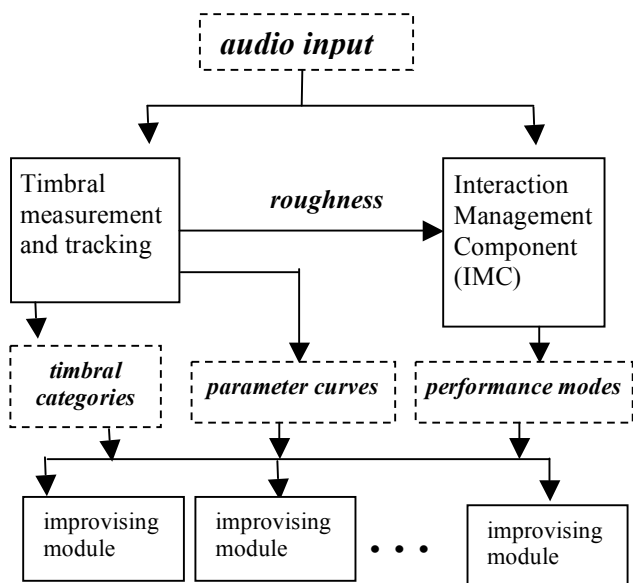


Figure 1: High-level organization of London system

The computation of roughness requires large analysis windows (>100 ms); a fast run of notes is often classified as *rough* because of note transitions and instability within the window. Hence, audio segments classified as *fast* are not assigned a roughness descriptor.

Figure 2 shows the detailed operation of the IMC. The audio input is divided into 2-second windows. The tempo estimate is computed by segmenting the audio input into “note” events, based on pitch estimates and amplitude envelope shapes. A roughness estimate is also computed. A feature vector comprising intensity (loud/soft), tempo (fast/slow), and timbre (rough/smooth) is derived. Each 2-second clip is classified into one of seven performance modes: silence, slow/soft/smooth, slow/soft/rough, slow/loud/smooth, slow/loud/rough, fast/soft, fast/loud. The current performance mode influences the high-level behavior of each improvising agent. For example, an agent may decide to support the human performance by matching the human’s performance mode, or perform in a contrasting mode. The actual gestures performed by an agent is based on its own repertoire.

Excerpts from the LAM 2006 performances with Evan Parker and John Butcher are available at <http://userwww.sfsu.edu/~whsu/Protected/> (password is

saxophone). The IMC is controlling an agent playing filtered noise, with roughness mapped to noise tremolo.

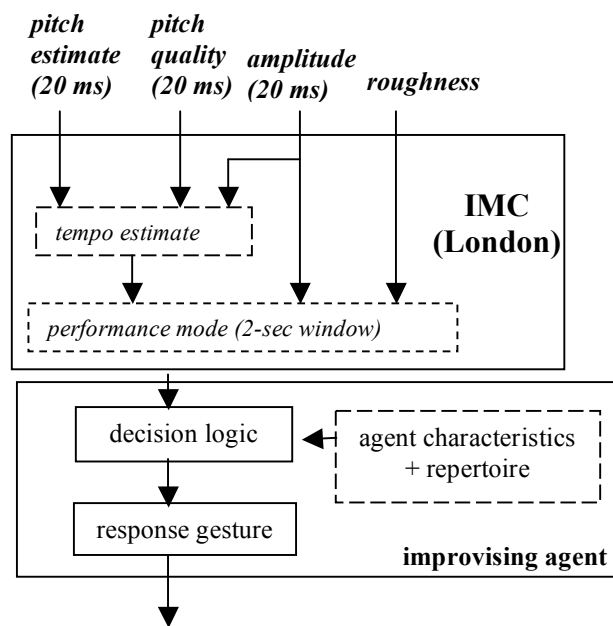


Figure 2: IMC (London) Operations

3.2. Performances at LAM and NIME

The IMC has great difficulty computing note segmentations for some of Parker’s hallmark soprano saxophone gestures: envelope shape heuristics are defeated by fast legato runs, and pitch estimate heuristics are defeated by pitch inflections. Here, fast runs were often described as being slow. However, there was enough ebb and flow in Parker’s playing, and enough autonomy in the agent’s decision logic, that the result is not too predictable or mechanical.

In his performance with the system, Butcher developed his ideas slowly at the start. The performance modes computed by the system adequately describe the saxophone performance, and timbral variations are tracked properly.

At the NIME 2007 performance, percussionist Meehan used mostly friction-related techniques on snare drum and cymbals. There was relatively little variety in gestural shape or tempo; the system mostly worked with changes in loudness and timbre, and the more gradual rates of change seemed appropriate in context.

3.3. Evaluations

In its three public performances so far, the system was able to produce some musically acceptable results. However, on closer monitoring, there seemed to be too many segments where a “soft gray” quality of interaction dominated.

In an improvisation that we characterize as responsive/tight, it often seems that the *timing* of a particular performance action, such as gesture start/end,

or a small timbral/gestural adjustment, is tied to observed events in a very small time window. The 2-second window for characterizing performance mode seemed both too long and too short; we need information from a larger window (for broader material-oriented decisions), and a smaller window for determining response timing. (Voyager [3] also uses multiple window sizes to manage event statistics.)

We were also dissatisfied with the long-term behavior of the system. It is certainly easy to randomize agent response to an observed performance mode. We would like to incorporate better justification for how an agent chooses to behave, with some reference to the musical intent of the human improviser. These considerations led us to redesign the IMC, resulting in the ARHS (for Adaptive Real-time Hierarchical Self-monitoring) system.

4. ARHS SYSTEM WITH NEW IMC

The ARHS system differs from the London system mainly in the IMC, shown in Figure 3. We replace the 2-second analysis window with two levels of windows, and introduce the concept of *potential trigger events*. To improve long-term adaptivity, we introduced a simple connectionist-inspired mechanism that we call an *adaptive performance map*.

4.1. Timing and responsiveness

To improve responsiveness, we decreased the event window size to one second. In addition, performance information is managed hierarchically. Performance modes are reported once per second, while event statistics are also summarized over a sliding 8-second window.

Potential trigger events, based on short-term transitions in performance modes, represent “ear-catching” events that may likely cause a change in agent behavior. They are reported once per second and affect the timing of agent actions. In the current version of the ARHS system, these events include the start of a gesture after silence, the start of silence at the end of a gesture, a sudden change in performance mode after a period of stability, a long period of a stable performance mode that indicates high intensity, and a “timeout” event after a set period of agent inactivity. The sudden occurrence of timbral features, for example a sharp attack caused by a slap-tongue, can also act as potential trigger events. Depending on its internal processes, an agent may respond to a potential trigger event by starting or stopping a phrase, changing some timbral characteristic, etc.

Performance mode statistics are also accumulated over the previous eight seconds, and are consulted to shape response materials. Within an 8-second window, we count the number of one-second windows that contain each performance mode, and the number of silent windows. We also report the observed frequency

of each class of tempo, loudness, and roughness, each quantized into three levels; more details in Section 4.2.

4.2. Intent and adaptation

Regarding the long-term behavior of the system, we are interested in several questions. With the human improviser working in particular performance modes, how does an agent select a high-level response mode? The human improviser may or may not like working with the current combination of musical materials. Is the system able to sense this and adjust its behavior? In a longer multi-section improvisation, interactive music systems often produce vaguely similar responses; they do not produce the sharply-etched contrasts in approach and texture that we may expect from good human improvisers. We would like to avoid simulating long-term behavioral evolution with occasional human intervention, or some kind of ad hoc periodic randomization of high-level behavior parameters.

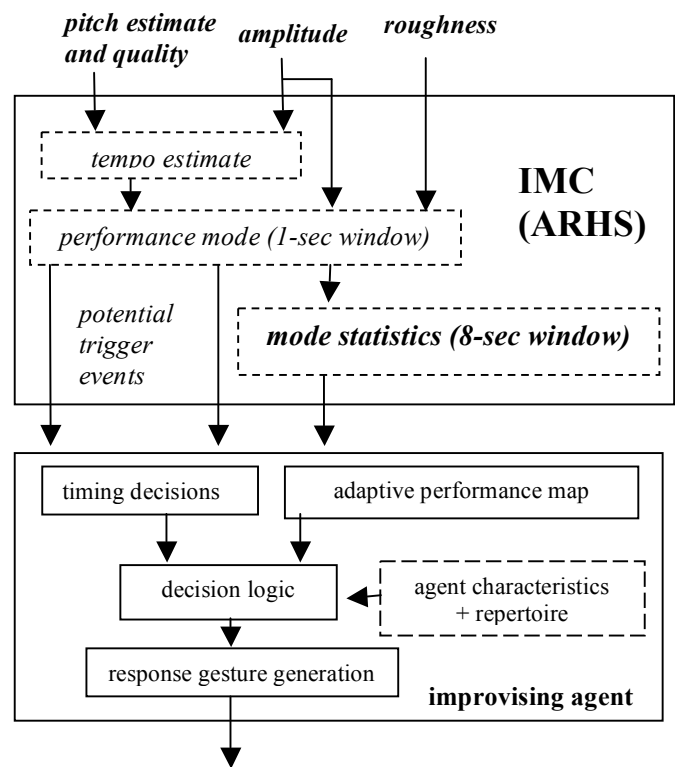


Figure 3: IMC (ARHS) Operations

The problem is it is difficult to infer the intent of the human improviser and her/his reaction to the machine response, solely from observing the input audio stream and the response audio stream. (Not even humans can do this accurately!) How should an agent make an adaptive and dynamic evaluation regarding the compatibility of combinations of musical materials?

The ARHS system attempts to “guess” the evaluation of the human improviser from transitions in the human’s

performance mode statistics. Suppose the human improviser is observed to play somewhat consistently in mode H1 over an extended period, while an agent is playing in mode A1. One might reasonably assume that the H1/A1 combination is considered desirable by the human. If the human considers H1/A1 to be a musically unacceptable clash of activity, one might expect the human to change her/his performance mode to adjust to the undesirable situation. Hence, the agent is “encouraged” to continue behavior in mode A1, if it continues to observe the human playing in mode H1. However, if the agent has been playing in mode A1, and observes the human switching to mode H2, the H1/A1 combination is discouraged.

As described earlier, statistics for the human improviser’s performance are collected for the previous eight seconds. For each class of tempo (slow/fast), loudness (soft/loud) and roughness (smooth/rough), an approximate 3-level frequency level is logged: seldom (0), moderate (1), and often (2). For instance, a score of 0 for tempo, 2 for loudness, and 1 for roughness indicates that very few of the one-second performance modes were fast, most of the modes were loud, and a moderate number were rough.

The *adaptive performance map* in Figure 3 is a data structure that maps each set of observed (human) performance mode statistics to a response (agent) mode. A score is kept for each mapping, and updated to indicate a preference for that mapping. The response modes with the highest scores are used to guide the agent’s performance. This quasi-connectionist approach ensures that the agents behave in a complex and adaptive fashion.

In summary, an agent monitors the previous 8 seconds of performance; its adaptive performance map suggests a preferred performance mode, which shapes its response. As with human improvisers, workable practices can be quickly “learned” during performance, and are adjusted through an extended improvisation.

5. EVALUATIONS AND FUTURE DIRECTIONS

At the time of writing, we have not been able to test the ARHS system in a live situation. We have however run tests using recordings from Butcher; some clips can be found at <http://userwww.sfsu.edu/~whsu/Protected> (password is *saxophone*). The saxophone solo recording is in the right channel, with a single agent “playing” filtered noise in the left channel. The generated gestures are fairly simple for demo purposes and have not gone through the fine-tuning that is usual before a performance. Also, there is no “feedback” from the agent performance into the saxophone performance. The first excerpt begins with a number of abrupt transitions, which are potential trigger events; note how the ARHS system uses these events as jumping-off points for its own gestural adjustments. As the saxophone gestures become stable and extended, the system interprets this to mean that a compatible combination of musical

materials has been found; the system also gradually plays with fewer abrupt changes.

The functionality we set out to implement in the ARHS system seems to work correctly. From preliminary tests, it appears we were able to improve the responsiveness of the system with our design changes. The adaptive performance mode map also seems to be functionally correct; a proper evaluation must of course wait until we have had more time to observe the system in performance situations with human improvisers. Compared with the London system, it coordinates more effectively high and low level performance information, and seems capable of some musically interesting behavior with little human intervention. Future directions include improving note-segmentation for tempo estimation, and exploring more machine learning techniques.

6. REFERENCES

- [1] Hsu, W., “Managing Gesture and Timbre for Analysis and Instrument Control in an Interactive Environment”, *Proceedings of the International Conference on New Interfaces for Musical Expression*, Paris, France, 2006.
- [2] Hsu, W., “Design Issues in Interaction Modeling for Free Improvisation”, *Proceedings of the International Conference on New Interfaces for Musical Expression*, New York, USA, 2007.
- [3] Lewis, G., “Too Many Notes: Computers, Complexity and Culture in *Voyager*”, *Leonardo Music Journal*, Vol. 10, 2000.
- [4] Dannenberg, R. et al., “A Machine Learning Approach to Musical Style Recognition”, *Proceedings of ICMC*, Thessaloniki, Greece, 1997.
- [5] Young, M. and Lexer, S., “FFT analysis as a creative tool in live performance”, in *Proceedings of the 6th International Conference on Digital Audio Effects*, ISBN 0-904-18897-3, 2003.
- [6] Collins, N., *Towards Autonomous Agents for Live Computer Music: Realtime Machine Listening and Interactive Music Systems*, Ph.D. Thesis, University of Cambridge, Cambridge, UK, 2006.
- [7] Lu, L., et al., “Automatic Mood Detection and Tracking of Music Audio Signals”, in *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 14, No. 1 (Jan. 2006), 5-18.
- [8] Vassilakis, P., “Auditory roughness estimation of complex spectra – roughness degrees and dissonance ratings of harmonic intervals revisited”, in *Journal of Acoustical Society of America*, 110(5/2), 2001.