

Managing Gesture and Timbre for Analysis and Instrument Control in an Interactive Environment

William Hsu

San Francisco State University
1600 Holloway Avenue
San Francisco CA, USA
hsu@tlaloc.sfsu.edu

ABSTRACT

This paper describes recent enhancements in an interactive system designed to improvise with saxophonist John Butcher [1]. In addition to musical parameters such as pitch and loudness, our system is able to analyze timbral characteristics of the saxophone tone in real-time, and use timbral information to guide the generation of response material. We capture each saxophone gesture on the fly, extract a set of gestural and timbral contours, and store them in a repository. Improvising agents can consult the repository when generating responses. The gestural or timbral progression of a saxophone phrase can be remapped or transformed; this enables a variety of response material that also references audible contours of the original saxophone gestures. A single simple framework is used to manage gestural and timbral information extracted from analysis, and for expressive control of virtual instruments in a free improvisation context.

Keywords

Interactive music systems, timbre analysis, instrument control.

1. INTRODUCTION

Timbre is an important structural element in non-idiomatic free improvisation [2], especially in the work of saxophonists and other instrumentalists who use extended techniques. For true interactivity, the virtual instruments within a software improvisation system should be able to respond to aspects of an improviser's gestural language, including timbre, that might be perceived as significant by human improvisers.

Our interactive music system [1] is developed in close collaboration with British saxophonist John Butcher, well-known for the complex and crucial role of timbre in his sophisticated musical language [3]. The system works with timbral information, as well as more traditional musical parameters such as pitch, loudness and duration. The design goals for our system were these:

- 1) The system will be used in the context of free improvisation.
- 2) There will be minimal use of looping or sequencing, i.e., the system will behave in unpredictable ways, like an

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

NIME 06, June 4-8, 2006, Paris, France.
Copyright remains with the author(s).

improviser.

- 3) The system will be responsive to timbral variations in the saxophone sound.
- 4) It should work with the range of Butcher's saxophone vocabulary, from extended techniques, to small close-miked sounds, to saxophone-controlled feedback through a sound system.
- 5) The system will not be a purely player paradigm system in the sense of Rowe [4]. That is, there will be options for a human to intervene and influence the larger shape of the system's behavior.
- 6) Overly obvious mappings of saxophone gesture to computer-generated gestures should be minimized.

[1] concentrated on the problems of timbral analysis and classification in our system. This paper details enhancements made in the last six months, especially in tracking, managing and coordinating high and low level gestural information. A few excerpts from our residency at STEIM, using an older version of the system from 2003, are at <http://userwww.sfsu.edu/~whsu/Timbre>.

We continued our work during a residency at ZKM (Karlsruhe) in May 2006, and will have new recordings for audition at NIME 2006.

We will present a selected survey of related work, and describe our system organization, focusing on the components for gesture/timbre capture and management. We will discuss issues of material generation in improvisation, and describe the framework we use for expressive timbre control of our virtual instruments. Finally, we will evaluate our experiences of the system, and discuss future directions.

2. RELATED WORK

Many previous interactive music systems work primarily with pitch and high level gestural characteristics. For example, George Lewis' *Voyager* [5] and Matt Ingalls' *Claire* [Ingalls, personal communication] both use pitch-to-MIDI converters to preprocess the input audio stream. For more examples of systems that work mostly with MIDI, see [4]. Roberto Morales' GRI [6] combines pitch with information from sensors that capture a human improviser's physical gestures on the flute.

In [4], Rowe discusses aspects of Zach Settel's piece *Punjar*, in which timbral characteristics, such as sibilance in the delivery of a vocalist, are used to influence synthesis. In [7], Cort Lippe describes his *Music for Clarinet and ISPW*, and discusses how timbre might be used to control material generation.

Ciufo's *Beginner's Mind* [8] is an improvisation system for use with unspecified instruments. It performs detailed analysis on the input audio stream, using Jehan's MSP external analyzer~ [9]. The real-time data stream from analyzer~ influences the

configuration and behavior of a network of processing modules. In addition, statistics (such as the mean and standard deviation of the pitch, loudness etc) are collected for each phrase to create a *perceptual identity* for the phrase. As will be seen in Section 4, we monitor a larger set of timbral characteristics, and also track their progression over the course of a phrase or gesture. Our improvisation agents may use this information to generate responses that make references to timbral feature contours in the human improviser's performance.

3. SYSTEM ORGANIZATION

Our system, implemented in Max/MSP (www.cycling74.com), monitors real-time audio input from an improviser, extracts timbral and gestural characteristics, and uses this information to guide the generation of response material. Figure 1 shows the high-level system organization.

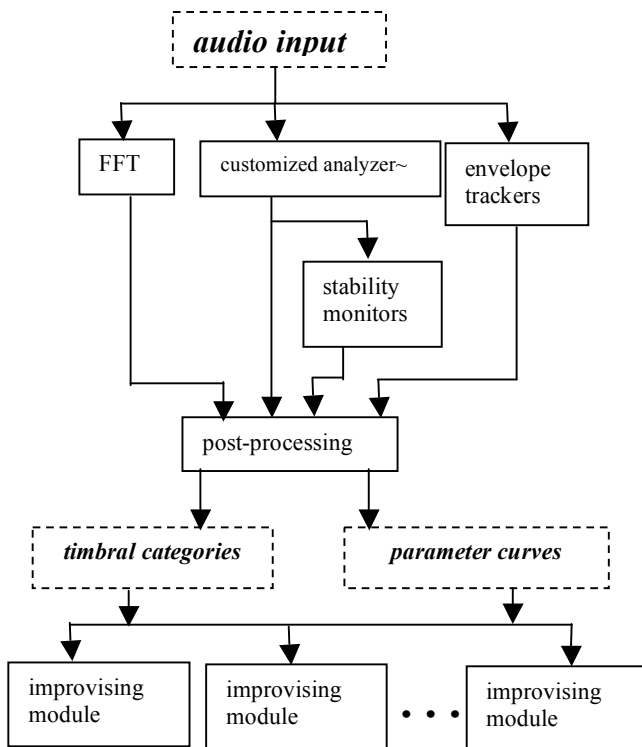


Figure 1: High-level system organization

The audio input stream (Butcher's saxophone sound) is fed into analysis modules. The raw measurements are post-processed to yield broad descriptive categories for timbre, and other performance characteristics. In addition, for each phrase/gesture, the progression of a number of timbral (and performance) parameters are tracked and stored in a repository, to be referenced for generating response material.

4. TIMBRE AND GESTURE ANALYSIS

4.1 Timbre categories

Timbral variation is often an integral component of musical gestures in improvisation. For example, a long saxophone tone might be held, with stable pitch and loudness, but acoustic roughness is slowly increased through embouchure control. An experienced human improviser would perceive and respond to this gestural variation.

Our proposed timbre classification framework attempts to reflect broad perceptual categories from a listener's perspective. [1] described our measurements and strategies for identifying specific timbral categories, which we will briefly summarize. A saxophone tone might be described as

- 1) *noisy* (vs. not noisy); the prominence of breath noise in a tone
- 2) containing *harmonic partials* (vs. inharmonic partials)
- 3) containing a *sharp attack* (vs. no sharp attack)
- 4) containing *multiphonics* (vs. no multiphonics)
- 5) with *flutter* (vs. no flutter); we define flutter to be a periodic fluctuation in the amplitude envelope, like a tremolo. (In [1], we confusingly called this *roughness*. Since we are now working with an additional *acoustic roughness* measure, detailed in Section 4.3, we have renamed this category.)

4.2 Gestures and parameter curves

In addition to the on-the-fly classification of the audio input stream into timbral categories, the system also monitors and records the progression of timbral and other musical parameters for each phrase or gesture.

For our purposes, we define a phrase/gesture as a sustained musical statement, possibly containing multiple note on and off events and short silences, separated from other gestures by significant intervals of silence. Each gesture is divided into a sequence of approximately 200 ms windows. For each window in a phrase, we track a set of measurements. Hence, for each gesture, we have a set of curves; each curve represents the variation of a parameter over the gesture. In addition, we track and store timestamped note on/off's through a phrase.

Figure 2 shows a block diagram of how we organize the per-gesture measurements. For each gesture, our system collects a set of parameter curves, parsed from the audio input stream during real-time performance, and stores them in a repository (within dashed outline in Figure 2). We assume a gesture begins when, after a significant (adjustable) period of silence, the amplitude of the input signal increases past a threshold. Starting from the onset of the gesture, we track loudness, brightness, noisiness, roughness, the pitch estimate and its stability/reliability at 200ms intervals. We stop recording parameters for a gesture when a significant period of silence is detected, or when a maximum gesture length is exceeded.

Loudness, brightness, and noisiness are collected using Jehan's MSP external analyzer~; the average of each parameter over a 200ms window is recorded. Hence, each parameter curve represents a list of 200ms averages of that parameter, over the progression of a specific gesture. For pitch, IRCAM's yin~ pitch estimator gives us 20ms pitch estimates and confidence. We also monitor the stability of the pitch estimate over each 200ms window.

The 200ms windowed averages are clearly imprecise; they only give a reasonable indication of input signal characteristics if the signal is fairly stable. If there are, for example, fast runs with pitch changes, the measured parameters will not be meaningful. Hence, it is useful to monitor the stability and confidence of the pitch estimate in each 200ms window, as well as the presence of multiple note on/off's, to validate the timbral, loudness and pitch measurements. Examples of some saxophone phrases and parameter curves for several of the measurements can be found at <http://userwww.sfsu.edu/~whsu/Timbre>.

While processing measurements from analyzer~ is quite straightforward, handling acoustic roughness entailed significantly more effort, which we will describe in the next section.

4.3 Measuring acoustic roughness

Auditory roughness describes an aural sensation associated with harsh, dissonant sounds [10]. It is one aspect of timbre, and appears to encapsulate, in a quantitative measure, characteristics of some of the timbral categories we discussed earlier, such as flutter, closely spaced inharmonic partials, and prominence of (harsher) multiphonics. Vassilakis [11] has suggested that roughness is correlated with tension/release patterns in Lebanese mijwiz and other non-Western musics; from our experience, it also appears to be a usable approximation in free improvisation.

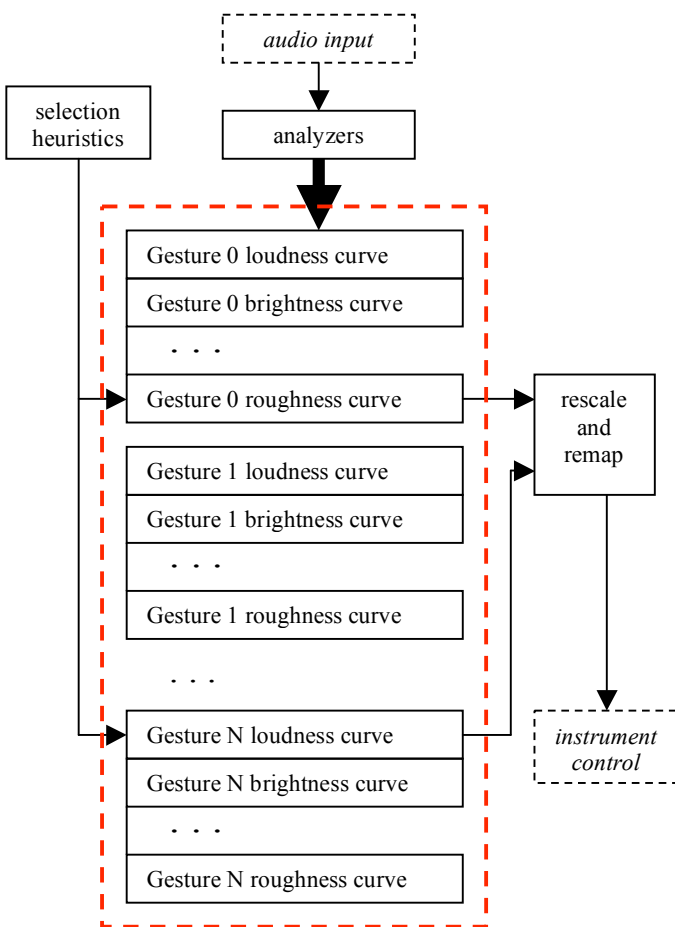


Figure 2. Parameter curve management block diagram

We based our roughness measurement on Vassilakis' model from [10], which is a refinement of an earlier model by Sethares. The basic procedure for estimating acoustic roughness from a complex tone includes these steps: 1) extract frequencies and amplitudes of sinusoidal partials from tone; 2) for each pair of partials, compute roughness contribution based on [10]; 3) sum contributions from all pairs of partials. (See <http://acousticlab.com/rougness/learnmore/MoreModel.html>) for more details.)

This procedure requires fairly accurate extraction of partials from the audio stream. Earlier versions of our system relied on Puckette's fiddle~ [12] for partial extraction. However, we need

to separate partials that are close together in frequency, below 10 Hz. It is possible to increase the window size for fiddle~ to achieve this resolution; however, the significantly higher computation load means that the system will no longer run satisfactorily in real-time on our development platform (an Apple G4 iBook).

We switched our analysis components to Jehan's analyzer~ [9], which uses more efficient FFTs and can operate reasonably well with much larger window sizes. However, we found that some preprocessing of the signal in fiddle~, before the FFT and partial extraction, is left out in analyzer~; a possible consequence is analyzer~'s less reliable measurements observed for both partial frequencies and amplitudes.

After some investigation, we rewrote the partial extraction component in analyzer~, using an algorithm based on Smith and Serra's PARSHL [13]. Our customized analyzer~ object now uses PARSHL's parabolic partial estimation algorithm to extract fairly accurate characteristics of the 20 strongest spectral peaks. Neighboring partials should be at least three FFT bins apart, for good results. We currently resolve partials that are approximately 8 Hz apart.

Information from the partial extraction is finally fed into the roughness model equation based on [10]. Our roughness estimator reports a moving window average of the measured roughness, over a tunable number of overlapping windows (default six windows, or about 1.1 seconds).

In summary, our system is able to capture a set of timbral and musical contours for recent gestures or phrases played by the human improviser. Each gesture is essentially represented by a set of parameter curves of loudness, brightness, noisiness, roughness, pitch and pitch stability, and timestamped note on/off events; they represent the progression of these characteristics from start to finish of the gesture. The curves are stored in a repository to be accessed for instrument control.

5. RESPONSE GENERATION

5.1 Choice of materials

In free improvisation, the choice of material is fairly open, though improvisers generally avoid references to established idioms. The role of pitch tends to be downplayed or obscured; greater weight is placed on loudness, duration, and timbre. Likewise, our system emphasizes managing timbre (and loudness and duration) over pitch.

Smaller gestures with nuanced timbral variations are favored when supporting or engaging in dialog with the human improviser, over larger gestures (such as drones and thick textures) that may take up too much of the sonic space. (The latter are also possible, but should be carefully managed.) At the simplest level, gesture generation in our system involves the pseudo-random selection of a number of parameters, within tunable ranges, their rates of change, and how they might be influenced by audio input.

While we wish to avoid a delay/echo effect in our generated responses, human musicians do make references to each other's materials when improvising. Hence, as the human improviser's gesture choices change over the course of a piece, our system should be able to adjust its behavior to reflect these changes.

The tracking of timbral curves over each gesture, described in Section 4, gives us a variety of gestural materials to work with. By remapping one parameter curve from the input to a different parameter in a future generated gesture, we preserve a general impression of gestural shape and cross-referencing of materials,

but without the rigid delay/echo effect that we are trying to avoid. For example, the human saxophonist may play a sequence of tones that increase gradually in roughness. The system may respond with filtered noise whose cutoff frequency increases gradually through the gesture.

5.2 Virtual instrument control

In our system, an ensemble of agents, each “playing” a virtual instrument, responds to the real-time audio input. Each agent monitors the characteristics of the saxophone sound; a combination of internal processes and external stimuli determine the material being generated and performed. Agents may act independently, or form coordinated subunits, with a user making some high level organizational and structural choices (see [1] for more details.)

Each agent/instrument has a set of predetermined gestures that it can choose from. In addition, each agent has access to the parameter curves for recent gestures that have been collected in the repository. As seen in Figure 2, when generating response material, an agent may choose to use one or more parameter curves from the repository of recent saxophone gestures (or use its predetermined gesture set). Any parameter curve can be chosen from one or more recent gestures, rescaled in an appropriate way, time-stretched or otherwise manipulated, and mapped to the same or a different timbral or gestural characteristic to create a new gesture. The same framework for representing a saxophone gesture (a set of timestamped contours of pitch, pitch stability, note on/off, loudness, brightness, noisiness and roughness) is used to organize most aspects of a generated gesture. Hence, as the gestural language of the human improviser evolves over a performance, the gestural vocabulary available to the virtual instruments will also reflect the changing shapes of the saxophone phrases.

Many of our virtual instruments were chosen for ease of control of a range of timbral characteristics, especially brightness, noisiness and roughness. For example, for a filtered noise generator, brightness is controlled by the lowpass filter cutoff frequency, noisiness by the filter resonance, and roughness by a tremolo envelope. For a metallic-sounding comb filter excited by a noise source, brightness is controlled by varying the bandwidth of the noise source, noisiness by the feedback coefficient of the comb filter, and roughness by either detuning the harmonics of the comb filter, or with a tremolo envelope. Similarly, the brightness, noisiness and roughness of a waveguide bass clarinet can be adjusted by changing the embouchure, the mix of noise in the excitation, and a tremolo envelope for the excitation or embouchure. While not all our virtual instruments have the full range of timbral variations (see [1] for a more detailed description), a significant number of them are adaptable to working with the parameter curves from the captured gestures. Some examples of prerecorded saxophone gestures, their parameter curves, and synthesized gestures using those curves can be found at <http://userwww.sfsu.edu/~whsu/Timbre>.

6. EVALUATION AND FUTURE WORK

Our initial tests and experiences with the recent enhancements to the system, using recorded saxophone material, have been reasonably satisfactory. A proper evaluation is possible only with the live participation of John Butcher (or another saxophonist). We will work with this system extensively at our residency at ZKM in May, and will prepare recordings for audition at NIME 2006.

With the recent enhancements described in this paper, we seem to have found a usable approach for increasing the adaptability of the system to gestural and timbral variation in the improviser’s real-time performance. A single simple framework is used to manage information from analysis for use in instrument control. Future directions include more sophisticated monitoring of both the improviser’s and the generated performance, codifying tension/release patterns in performance, and role-oriented coordination of the improvising agents.

7. ACKNOWLEDGEMENTS

Thanks to Miller Puckette, Tristan Jehan and Pantelis Vassilakis for source code and detailed discussions. This work was also generously supported by STEIM, Anne LaBerge at Kraakgeluiden Werkplaats, Chris Burns (formerly) at CCRMA, and ZKM (Karlsruhe).

8. REFERENCES

- [1] Hsu, W., Using timbre in a computer-based improvisation system. In *Proceedings of the ICMC* (Barcelona, Spain, Sept. 5-9, 2005).
- [2] Bailey, D., *Improvisation: Its nature and practice in music*. Da Capo Press, 1993.
- [3] Keenan, D., Mining echoes. In *The Wire*, November 2004. (See also <http://www.johnbutcher.org.uk/>)
- [4] Rowe, R., *Machine Musicianship*. The MIT Press, Cambridge, Massachusetts, 2001.
- [5] Lewis, G., Too Many Notes: Computers, Complexity and Culture in *Voyager*. In *Leonardo Music Journal*, Vol. 10, 2000.
- [6] Morales R. et al., Combining audio and gestures for a real-time improviser. In *Proceedings of the ICMC* (Barcelona, Spain, Sept. 5-9, 2005).
- [7] Lippe, C., A Composition for Clarinet and Real-Time Signal Processing: Using Max on the IRCAM Signal Processing Workstation. In *Proceedings of the 10th Italian Colloquium on Computer Music*, Milan, Italy, 1993.
- [8] Ciuflo, T., Beginner’s mind: an environment for sonic improvisation. In *Proceedings of the ICMC* (Barcelona, Spain, Sept. 5-9, 2005).
- [9] Jehan, T., Schoner, B., An Audio-Driven Perceptually Meaningful Timbre Synthesizer. In *Proceedings of the ICMC*, 2001.
- [10] Vassilakis, P., Auditory roughness estimation of complex spectra – roughness degrees and dissonance ratings of harmonic intervals revisited. *Journal of Acoustical Society of America*, 110(5/2).
- [11] Vassilakis, P., An improvisation on the Middle-Eastern mijwiz: auditory roughness profiles and tension/release patterns. *Journal of Acoustical Society of America* 117(4/2).
- [12] Puckette, M., Apel, T., Zicarelli, D., Real-time audio analysis tools for Pd and MSP. In *Proceedings of the ICMC* (San Francisco, USA, 1998).
- [13] Smith, J.O. and Serra, X., PARSHL: an analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation. In *Proceedings of the ICMC* (Champaign-Urbana, USA, 1987).