

USING TIMBRE IN A COMPUTER-BASED IMPROVISATION SYSTEM

William Hsu

Department of Computer Science
San Francisco State University
San Francisco CA 94132
USA
hsu@tlaloc.sfsu.edu

ABSTRACT

Timbre is an important structural element in non-idiomatic free improvisation [1]. An interactive software system that improvises in such a context should be able to analyze and respond to gestures with rich timbral information. We have been working on a system that improvises with British saxophonist John Butcher, whose complex vocabulary incorporates extended saxophone playing techniques, as well as use of amplification and feedback [19]. Butcher's rich and diverse saxophone sounds are classified into broad perceptual categories that might be useful for improvisers. The general behavior of our system is influenced by real-time timbral characteristics, as well as other musical parameters. As in contexts involving non-idiomatic free improvisation, the emphasis is on working with abstract sound, gesture, and texture, rather than more traditional parameters such as pitch and harmony.

1. INTRODUCTION

Rowe's classification scheme [2] identifies some improvisation-oriented interactive music systems as following a *player paradigm*; such a system behaves like an improvising partner in a performance with a human musician. The system should be able to analyze and "understand" aspects of an improviser's gestural language that might be perceived as significant by other human improvisers.

In non-idiomatic group improvisation, musicians clearly take timbre into account when making performance decisions. This is all the more critical when working with saxophonists or other wind instrumentalists who have made extended techniques and timbre variations important components of their approaches, from the pioneers of the 60s such as Roscoe Mitchell and Evan Parker, to today's virtuosos such as John Butcher and Peter van Bergen.

We have been working on a software improvising system in the Max/MSP environment. This project has been a result of close collaborations with British saxophonist John Butcher. Butcher is commonly regarded as an innovative improviser who has greatly expanded the timbral palate of the saxophone (see, for example, [19]). We felt that many improvisation-oriented computer music systems are limited in their

interactivity, because they do not make sufficient use of timbral information.

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 improviser.
- 3) The system will be responsive to timbral variations in the saxophone sound, plus other performance characteristics.
- 4) It should work with the range of Butcher's saxophone vocabulary, from extended techniques such as multiphonics, to small sounds that are closely miked and amplified, to saxophone-controlled feedback through a sound system.
- 5) The system will not be a purely player paradigm system in the sense of Rowe. That is, while certain components of the system have fair degrees of autonomy and unpredictable behavior, there will be options for a human to intervene and influence the larger shape of the system's behavior.
- 6) While the system should be responsive and engage in cohesive conversation with the saxophonist, overly obvious mappings of saxophone gesture to computer-generated gestures should be minimized.

Our initial plans for the system were formulated in 2001. Software design and implementation has continued since. Butcher and I worked closely together through two residencies at the STEIM studios (Amsterdam) in 2003; in November 2004, our first public concert was at the Kraakgeluiden Werkplaats in Amsterdam. A few excerpts from our STEIM sessions are at <http://userwww.sfsu.edu/~whsu/Timbre>.

This paper will describe the overall organization of the system. We will present a selected survey of related work, overview how we extract timbral categories from the real-time audio stream, and describe how this information shapes the behavior of a virtual improvising ensemble. Finally, we will evaluate the current state of the system, and discuss future directions.

2. RELATED WORK

George Lewis' Voyager [3] is one of the best-known improvisation-oriented software systems. Voyager's

virtual improvising orchestra is driven by a combination of a human’s real-time performance and its own internal processes. A pitch-to-MIDI converter transforms the real-time audio stream (from, say, Lewis’ trombone) to MIDI data; Voyager works with the converted MIDI input, rather than directly with audio.

Matt Ingalls’ interactive system Claire has mostly been heard in performance with his clarinet and bass clarinet. Claire also uses a pitch-to-MIDI converter; MIDI output controls tone modules or a Yamaha Disklavier. Claire is able to effectively evoke pianistic gestures and interactions in a free improvisation context [4].

Both Voyager and Claire are *player paradigm* systems, according to Rowe’s classification. Lewis and Ingalls are both well-known as masters of extended techniques on their respective instruments. Hence, it is interesting that Voyager and Claire have produced wonderful musical results, despite the seemingly severe limitation of using the pitch-to-MIDI converter to parse the input audio.

Several electronics/saxophone duos work with live sampling and processing. Lawrence Casserley has performed and recorded extensively with saxophonist Evan Parker, using his ISPW-based digital signal processing instrument [16]. Phil Durrant has also developed a body of work with saxophonist John Butcher, using hardware and software effects units to process Butcher’s saxophone sound. Both these systems are closer to Rowe’s *instrument paradigm* category; each behaves like an “extended instrument” controlled directly by a computer musician. The audio stream from the saxophonist is transformed, but timbral information is not extracted for configuration and decision-making.

In [5], Rowe overviews several interactive systems/pieces that work primarily with MIDI data, generated either through a MIDI controller or a pitch-to-MIDI converter. These include Rowe’s own Cypher, Edmund Campion’s *Natural Selection* (for MIDI piano and computer), and Mari Kimura’s *Izquierda e Derecha* (for Zeta violin and computer).

Rowe also discusses aspects of Zach Settel’s piece *Punjar*, which uses analyzers developed by Settel for IRCAM’s Jimmies library [5]. Timbral characteristics, such as sibilance in the delivery of a vocalist, are used to influence synthesis. Cort Lippe describes in [6] his *Music for Clarinet and ISPW*, and discusses how timbre might be used to control material generation; it is not clear from [6] how this is actually organized or implemented in his piece. Puckette and Lippe [7] discuss using timbre from a live audio stream to influence control, but the paper contains few specifics.

Ciufo describes in [8] an interactive improvisation-oriented system for guitar that combines sensors and real-time audio to control processing. Real-time

processing and synthesis are controlled by brightness, noisiness, and other parameters measured through Tristan Jehan’s MSP external analyzer~ [9]. Jehan’s system in [9] uses timbral (and other) characteristics to guide the mapping of control parameters to acoustically meaningful results in cross-synthesis. Our system uses a larger set of timbral categories to coordinate an ensemble of virtual improvisers; see next section for details.

3. SYSTEM ORGANIZATION

Our goal is to construct an interactive computer music system that is able to monitor real-time input from improvisers, extract timbral characteristics, and use timbral changes in its decisions for generating response material. Figure 1 shows the high-level organization of our system.

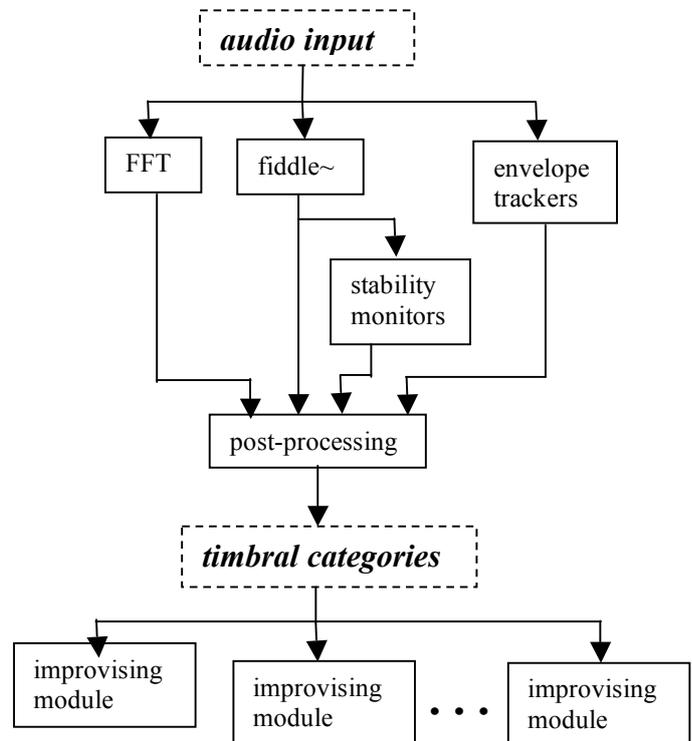


Figure 1: High level organization of system

The audio input stream (in our case, Butcher’s saxophone sound) is fed into a set of analysis modules. The raw measurements are post-processed to yield broad descriptive categories for the timbre of the sound, and other performance characteristics. The timbral categories (and other information) are monitored by a virtual ensemble of improvising modules. Each module “performs”, based on a combination of internal processes and extracted information from the audio input.

4. TIMBRE ANALYSIS AND CLASSIFICATION

We first developed a framework for analyzing the timbre of an instrument in real-time, and forming broad

classifications that a human musician might perceive and respond to, in a performance situation. In real-time improvisation, decisions need to be made promptly; a human improviser is more interested in whether a tone is rough versus smooth, rather than how a rough tone is produced. Our emphasis is on broad perceptual categories; we approach timbre largely from a listener's perspective.

We also monitor higher level non-timbral parameters such as pitch range, gesture length, silence between gestures, density of note onsets, etc. This paper will focus primarily on how timbral information is used in our system.

4.1. Timbral gestures and categories

Improvising reed players such as John Butcher have developed sophisticated approaches in which extended techniques and timbral variation are integral components. For example, a long tone might be held, with fairly stable pitch and loudness, but the intensity of multiphonics is slowly increased through embouchure control. An experienced human improviser would perceive and respond to this gestural variation. (See for example [10] for a classic study of extended techniques for woodwinds.)

We have studied a large number of recordings of John Butcher and other saxophone players who are adept at extended techniques. We propose the following timbre categories as a starting point for our descriptive framework. A saxophone tone might be described as:

noisy (vs. not noisy)
containing harmonic partials (vs. inharmonic partials)
containing a sharp attack (vs. no sharp attack)
containing multiphonics (vs. no multiphonics)
rough (vs. smooth)

We will describe the measurements made by our system, and how they are used to identify timbral characteristics.

4.2. Measurements and post-processing

Our system was constructed in the Max/MSP environment. Several existing MSP externals were used in combination with the standard MSP objects; in most cases, extensive post-processing of the raw measurements was necessary to extract usable information and produce the timbral categories that we needed.

At the time of the initiation of our project, Jehan's analyzer~ object [9] was not yet publicly available. In addition to the standard MSP objects for FFTs, we mostly leveraged Miller Puckette's fiddle~ for pitch estimation and partial tracking [11]. While fiddle~ (and similar pitch and partial trackers) work reasonably well

for clean, sustained saxophone tones, its results can be rather unreliable near the attacks and decays of a tone. The pitch and partial estimations are also unreliable and less usable when the saxophone tone itself is noisy or has a complex and rapidly changing spectrum. We monitor the stability of the pitch estimation from fiddle~, over several analysis windows; this turned out to be very useful in characterizing several of the categories. Since the behavior of strong partials tend to mask the behavior of weak ones, we configured fiddle~ to produce the twelve partials that are lowest in frequency, and sorted them to select the six strongest ones for further analysis.

Additional raw measurements made by our system include: relative spectral centroid (absolute spectral centroid divided by the estimated pitch), zero-crossings, peak energy distribution (strength of top ten FFT peaks relative to overall energy in signal), and the presence of very sharp onsets (produced by techniques such as slap tongue). We also monitored the mean and variance of most measurements, over several analysis windows.

Since most of the measurements tend to be unreliable when the audio signal is at extremely low levels, the measured data is only considered "meaningful" to the analyzer modules when the energy in a frame is above a tunable threshold. Measurements made when the input signal is below the threshold are not reported.

4.3. Identifying timbre categories

Noisiness refers to the prominence of breath noise in a saxophone tone. This is often identified through a combination of zero crossing counts and amplitude thresholds. For the saxophone tones we are working with, we found that there were examples with high zero-crossing counts from the presence of extreme high partials, but where breath noise seemed less dominant. Two other measurements helped to identify noisiness. Noise usually results in an extremely unstable pitch estimate from a pitch tracker like fiddle~. In addition, the energy of a noisy tone tends to be widely distributed across its spectrum, rather than concentrated at a few spectral peaks. If our measurement of peak energy distribution is below a threshold, and the pitch estimate is extremely unstable, we classify the audio input as noisy.

Prominence of inharmonic partials is detected by comparing the frequencies of the six strongest partials obtained from fiddle~. The pitch estimate should be relatively stable.

Presence of sharp attacks corresponds to techniques such as slap tongue or amplified key clicks. These are identified by steep rises in the amplitude envelope.

We found the *presence of multiphonics* to be strongly correlated with a stable pitch estimate, a concentration

of energy in relatively few spectral peaks, and a high normalized centroid. This has been one of the more difficult categories to characterize, especially with tones containing rich harmonics such as from a tenor saxophone. While our measurements work well to identify tones with multiphonics that are relatively distant from the fundamental pitch, we do obtain occasional “false positives” when rich tenor tones without perceived multiphonics are encountered. Also, Butcher and other saxophone players are able to produce multiphonics that are close to the fundamental, within a major 2nd; the cluster of closely-spaced low frequency partials can produce confusing spectrum measurements, as well as result in a relatively low normalized centroid. In some situations, this type of multiphonics is identified (not unreasonably) as a form of amplitude fluctuation or roughness. We are currently working on improving our identification of closely-spaced multiphonics.

Several studies have attempted to quantify the acoustic sensation of *roughness*; see for example the work of Sethares [12], or Hutchinson and Knopoff [13]. The emphasis has been on measures for the roughness of synthetic tones with well-defined and relatively stable spectra.

Acoustic saxophone tones can be quite complex, with unstable spectra and rapidly changing relationships between partials. We decided instead to base our roughness estimate on measurements of the fluctuation of waveform amplitude envelopes, after the work of Pantelis Vassilakis [14]. Our roughness category is thus limited to effects of amplitude fluctuation. A “rough” tone can be produced by techniques such as flutter-tongue and throat tremolo. We place in this category tones whose amplitude envelopes fluctuate periodically, with a deviation of more than 10% about its average value, at frequencies of about 10 to 50 Hz.

5. MATERIAL GENERATION

5.1. Language and materials in free improvisation

Free improvisation has emerged as a cohesive movement since the 60s [1]. While the choice of material is very open, the general practice is to avoid references to established idioms. Derek Bailey in [1] acknowledged the influence of Schoenberg’s pre-serial free atonal period and Webern’s late music, as well as early electronic music:

... I thought ... that intervallic manipulation of pitch is less restricting and more productive than other ways of pitch management, and that the very clearly differentiated changes of timbre which characterised some early electronic music was the sort of thing which could assist in assembling a language that would be literally disjointed, whose constituents would be

unconnected in any causal or grammatical way and so would be more open to manipulation. A language based on malleable, not pre-fabricated, material. Generally I was looking... to utilise those elements which stem from the concepts of unpredictability and discontinuity, of perpetual variation and renewal first introduced into European composition at the beginning of the 20th century.

In free improvisation, the role of pitch tends to be downplayed or obscured; greater weight is placed on loudness, duration, and timbre. (Hence, there are relatively few pianists working in free improvisation.) Pitch choice is likewise of secondary importance in our system. Greater effort is placed on managing duration and, especially, timbre.

The use of large gestures that may draw undue attention to themselves is always carefully managed by free improvisers. In a similar spirit, our system works more with smaller tactile gestures that incorporate nuanced timbral changes. Drones and thicker textures can also be generated, with parameters that are influenced by the audio input. With the decreased importance of pitch, timbre and the rate of change of timbral parameters become more important structural elements. Gestures whose features evolved slowly are perceived very differently from gestures whose characteristics undergo abrupt and rapid modifications. For example, consider a noise source passed through a resonance filter with a narrow bandwidth. A 0.5 Hz oscillation in the filter cutoff might be perceived as similar to a pitchglide; a 30 Hz oscillation might evoke a sensation of roughness. Gesture generation in our system involves the pseudo-random selection of a number of parameters, within tunable ranges, and their rates of change. Many of these parameters might also be influenced by the real-time audio input.

5.2. Choice of improvising agents

We organized the generative components of our system as a small virtual ensemble of improvisers. Each agent controls a module that transforms the input audio stream from the saxophone, or “plays” a virtual instrument. Each module receives a stream of messages describing the timbre and other characteristics of the input sound (in our case, John Butcher’s saxophone). Each also has a set of internal rules that governs its behavior.

A module may be chosen to join or leave the ensemble at any time. It may generate material solely according to its internal processes, or may “perform” when it detects specific combinations of timbral or performance characteristics in the audio input. The gestural parameters of the generated material might be influenced by the current or past timbre of the saxophone sound.

Both sound transformation modules and sound synthesis modules were selected for our ensemble. One module is a classic effects chain with a comb filter, flanger/chorus, and pitchshifter. Another is a granular synthesis module from Nathan Wolek's granular toolkit [18]; this turned out to be a useful component for working with the saxophone's clean sound, during simple melodic playing.

For each synthesis module, we were interested not so much in a huge range of timbres within one module, but the ability to generate a relatively well-defined class of sounds with timbral gradations and variants. We wanted modules that evoked wind instrument sounds, to blend with the saxophone. To work effectively with Butcher's performance range, these should be capable of a variety of gestural and timbral nuances. One module is a noise generator with a resonance filter, with easily controlled brightness, roughness, envelope shapes etc. We also implemented a waveguide-based bass clarinet module, with a variety of envelope and embouchure effects; it evokes a bass clarinetist versed in some extended techniques.

For timbral contrast, we implemented a modal synthesis module that simulates the sounds of resonating metallic objects, such as bells and woks. Clusters of the resonance peaks can be controlled independently, to provide natural-sounding timbral variations. The model may be "struck" or "bowed".

We also coded a simple acoustic guitar player based on the mandolin~ object from Trueman and Dubois' Percolate library of physical modeling objects [17]. A finite state machine provides high-level control of the guitar performance. Each state constitutes a gesture type (for example, strummed chords, bursts of short notes, or long tones); transition between states is influenced by the module's internal algorithm, along with timbre and performance information from analyzing the audio input stream. From our observations of acoustic guitars who play in free improvisation contexts (such as Derek Bailey, John Russell and Roger Smith), and our experiences with using plucked string synthesis, we found that pitch must be carefully controlled; inadvertent references to, for example, conventional harmonic progressions can upset the balance in an improvisation. Care was taken to obscure pitch information where necessary, through extreme pitchbends or fast damping.

5.3. Interaction design and coordination

Most previous studies on coordinating interactive improvisation focused on established idioms such as jazz piano, with pitch-choice being probably the overriding consideration. For example, Walker in [20] uses conversation analysis as a basis for a software improviser; the role of a participant in the improvisation is usually very well-defined (solo, bass line, comping etc), and the transitions between roles are also relatively

structured. In successful non-idiomatic free improvisation, the interactions are more open-ended, and roles are much more fluid, but there are still strong connections between sounds and gestures.

We designed our modules such that they may act independently of each other, or form coordinated subunits within the ensemble. For example, the granular pitchshifter has the option of obtaining input from the real-time audio stream, or from the metallic modal filter. The guitar player, noise generator, bass clarinet and other agents may coordinate to form clouds of short gestures; the density, frequency range and other parameters of these clouds may be influenced by the timbre of the audio input.

In our initial design discussions, John Butcher emphasized that there should be options for a human to influence at a higher level the behavior of the virtual ensemble. While each improvising module is able to swiftly respond to the changes it perceives in the audio input, a user should be able to make some organizational and structural choices through a performance. In this respect, our system has similarities in concept to Butch Morris' *Conductions*.

Morris described a conduction as a "conducted improvisation"; he has "developed a vocabulary of gestural information to extend the language of both musician and conductor..." to construct a real-time composition [15]. In our system, a user can choose the combination of modules that will participate in a performance, the timbral categories and other performance characteristics each agent may respond to, types of gestures and parameter ranges that might be used, and the manner of coordination between some groups of modules. These parameters and strategies can be changed in the middle of a performance, using the simple user interface, or a MIDI fader box. In general, a user does not directly control the gestures of each improvising module, only the larger overall shapes of the improvisation. (S/he may also participate directly in the improvisation by "playing" a module with a controller such as a Wacom tablet, but this is a separate and independent option.)

6. OBSERVATIONS AND FUTURE WORK

In our residencies at STEIM, and rehearsals and concert at the Kraakgeluiden Werkplaats, John Butcher and I have found that our system is evocative of a responsive improvising ensemble. In performance, we have the option of selecting a few timbral starting points (for example, closely miked saxophone sounds, with clouds of small generated gestures). Thereafter, the improvisation takes its own course, and becomes largely unpredictable. It is also possible to just begin improvising without a prior discussion of material choice; the system is capable from startup of monitoring

the audio input stream, and activating a selection of modules to begin playing. The behavior of each agent is responsive to changes in timbre, at a speed that is clearly impossible if we rely solely on direct human control in real-time.

In early versions of the system, listeners sometimes detected an undesirable characteristic in the interactions between the saxophonist and the system: when the saxophonist paused for a short period, the system eventually became silent, because timbral categories were not broadcast when the input signal level is very low. To alleviate this problem, we gave more autonomy to the generative algorithms in some of the agents. In addition, we added "memory" to the broadcast stream of timbral information; when the saxophonist pauses, the timbral description for the past few seconds is looped and still available to the agents. An agent may choose to continue playing through the pause, according to information from the recent past. Since the timbral categories are looped and made available, each agent continues to "hear" similar sounds, and respond in a consistent manner.

This is very much a work in progress. We continue to build on the system and fine-tune its behavior. We are working on more transformation and synthesis modules to expand the sonic palate of the system. We are looking into more sophisticated coordination amongst modules, perhaps extending Walker's use of conversation analysis [20] beyond the traditional jazz context. We also plan to further investigate Morris' techniques for conduction, and design a more user-friendly interface for conducting improvisations.

7. ACKNOWLEDGEMENTS

We would like to thank STEIM (Amsterdam), Chris Burns at CCRMA, and Anne LaBerge at Kraakgeluiden Werkplaats (Amsterdam), for their support through the development of our project.

8. REFERENCES

- [1] Bailey, D. *Improvisation: Its nature and practice in music*. Da Capo Press, 1993.
- [2] Rowe, R. *Interactive Music Systems*. The MIT Press, Cambridge, Massachusetts, 1993.
- [3] Lewis, G. "Too Many Notes: Computers, Complexity and Culture in *Voyager*", *Leonardo Music Journal*, Vol. 10, 2000.
- [4] Ingalls, M. *Personal communication*, 2001.
- [5] Rowe, R. *Machine Musicianship*. The MIT Press, Cambridge, Massachusetts, 2001.
- [6] Lippe, C. "A Composition for Clarinet and Real-Time Signal Processing: Using Max on the IRCAM Signal Processing Workstation", *Proceedings of the 10th Italian Colloquium on Computer Music*, Milan, Italy, 1993.
- [7] Puckette, M. and Lippe, C. "Getting the Acoustic Parameters from a Live Performance", *Proceedings of the 3rd International Conference for Music Perception and Cognition*, Liege, 1994.
- [8] Ciuffo, T. "Design Concepts and Control Strategies for Interactive Improvisational Music Systems", *Proceedings of MAXIS International Festival/Symposium of Sound and Experimental Music*, Leeds, UK, 2003
- [9] Jehan, T., Schoner, B. "An Audio-Driven Perceptually Meaningful Timbre Synthesizer", *Proceedings of the International Computer Music Conference*, 2001.
- [10] Bartolozzi, B. *New Sounds for Woodwind*, Oxford University Press, 1967
- [11] Puckette, M., Apel, T., Zicarelli, D. "Real-time audio analysis tools for Pd and MSP", *Proceedings of the International Computer Music Conference*, San Francisco, 1998.
- [12] Sethares, W. *Tuning, Timbre, Spectrum, Scale*, Springer Verlag, London, 1998
- [13] Hutchinson, W., Knopoff, L. "The acoustical component of western consonance", *Interface*, 7,
- [14] Vassilakis, P. *Perceptual and Physical Properties of Amplitude Fluctuation and their Musical Significance*, Doctoral dissertation, University of California, Los Angeles, 2001.
- [15] Morris, B. *Current Trends in Racism in America*, program note, Sound Aspects LC 8883, 1985.
- [16] Casserley, L. "A Digital Signal Processing Instrument for Improvised Music", *Journal of Electroacoustic Music*, Vol. 11. See also http://www.chiltern.demon.co.uk/DSP_Inst.html
- [17] Trueman, D., Dubois, L. *PeRColate* website, <http://www.music.columbia.edu/PeRColate>
- [18] Wolek, N. "A Granular Toolkit for Cycling74's Max/MSP", *Proceedings of SEAMUS 2002*.
- [19] Keenan, D. "Mining echoes", *The Wire*, November 2004. (See also <http://www.johnbutcher.org.uk/>)
- [20] Walker, W. "Applying ImprovisationBuilder to Interactive Composition with MIDI Piano", *Proceedings of 1996 International Computer Music Conference*, Hong Kong, 1996.