MULTI-AGENT MULTIMODAL PERFORMANCE ANALYSIS

Arne Eigenfeldt
School for the Contemporary Arts,
Simon Fraser University,
Burnaby, BC Canada

Ajay Kapur
California Institute of the Arts,
Valencia, CA, USA

ABSTRACT

This paper describes a custom built system which extracts high-level musical information from real-time multimodal gesture data. Data is collected from sensors during rehearsal using one program, GATHER, and, combined with audio analysis, is used to derive statistical coefficients which are used to identify three different playing styles of North Indian music: *Alap*, *Ghat*, and *Jhala*. A real-time program, LISTEN, uses these coefficients in a multi-agent analysis to determine the current playing style.

1. INTRODUCTION

Interactive and real-time computer music is becoming more complex, and the requirements placed upon the software are equally increasing. Composers, hoping to derive more understanding about a performer's actions, are looking not just at incoming audio, but also to sensor data for such information. We have created a real-time system for such analysis, and have chosen to focus upon a high-level musical cognition - playing style - by analyzing synchronized audio and sensor data.

1.1. Related Work

Rowe describes a computer system, which can analyze, perform and compose music based on traditional music theory [18]. Automatic music transcription is a well-researched, but unsolved, area [14, 16]. Automatic pitch extraction is also well researched [6]. There has been extensive work in automatic tempo tracking on audio signals [11, 19].

A variety of research has been undertaking using machine-learning techniques to classify specific gestures based on audio feature analysis. The extraction of control features from the timbre space of the clarinet is explored in [1]. Deriving gesture data from acoustic analysis of a guitar performance is explored in [2, 3, 4]. Gesture extraction from drums is explored in [5].

Multiple-agent architectures have been used to track beats within acoustic signals [10, 8]. It has also been used in Auditory Scene Analysis (ASA) [17] in order to extract individual sounds from mixed audio streams, as well as MPEG-7 analysis [15].

Our approach is different from, albeit informed by, the above research, in that we are using multimodal data from a performing artist and an agent-based approach to classify performance style.

1.2. Goals

The long-term goal of this research is to create a system which can extract high-level musical information from a performing musician, using a combination of sensor and audio analysis. This information will then be used to control a real-time generative/interactive program [9], in a way that emulates a human operator who can recognize high-level musical change.

The initial goal was to recognize different sections within the improvising musicians' performance, specifically a free improvisatory section (*Alap*), a melodic section (*Ghat*), and a faster, rhythmic section (*Jhala*). No attempt is made to create a general-purpose style-analysis system.

Section 2 describes the framework used to structure our research, and how data is collected in rehearsal; Section 3 describes how the data is analysed and evaluated; Section 4 describes how the program was used in performance; Section 5 posits our evaluation of the system; Section 6 suggests our future directions; Section 7 discusses our conclusions.

2. FRAMEWORK

2.1. Data Collection

Our multimodal system captured data from sensors on the instrument and on the human performer, as well as features derived from the audio signal.

Several recordings were made of each of the three classes (*Alap, Ghat, Jhala*), as well as a fourth class of "silence", in which the performer did not actually make sound, but generated sensor data.

2.1.1. Sensors on Musical Instrument

The ESitar [13] is an instrument which gathers gesture data from a performing artist using sensors embedded on the traditional instrument. A number of different performance parameters are captured including fret detection (based on position of left hand on the neck of the instrument), thumb pressure (based on right hand strumming), and 3-dimensions of neck acceleration.

2.1.2. Sensors on Human Body

A KiOm [12] was attached to the Sitar performer's head to capture body movement during performance. The KiOm has an embedded 3-D accelerometer sensor.

In both cases, data is converted to MIDI information, and streamed at a sample rate of 10 Hz.

2.1.3. Audio Feature Extraction

Audio features were computed using recent ChucK implementations [20]. The five features included energy

derived by Root Mean Square (rms), spectral centroid, spectral flux, spectral rolloff at 85%, and spectral rolloff at 50%. All data was sampled and streamed at 10 Hz.

2.2. GATHER

Data was sent via OSC [21] to a second computer. A program, GATHER, written in Max/MSP[1], records all incoming data; this program displays the data graphically as it is received, providing a helpful tool for viewing sensor data while performing (see fig. 1).

The incoming audio data is further analyzed using Tristan Jehan's FFT-based perceptual analysis MSP objects[2]: loudness~, brightness~, and noisiness~. MIDI data from the ESitar and KiOm is scaled to a 0.0 to 1.0 range. A sample & hold function is used to convert incoming control data from 10 Hz to 44.1 kHz: all data is then stored as audio signals and saved to disk for future analysis. Since both audio and sensor data are recorded as audio signals, they can be played back later to replicate the performance situation.

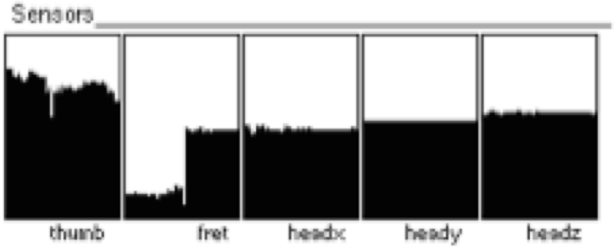

Figure 1. Incoming sensor data in GATHER.

3. ANALYSIS OF DATA

3.1. Statistical Analysis

Within GATHER, each data stream is analysed for minimum, maximum, mean, and standard deviation values. From these, averages are continuously calculated for each signal (see fig. 2).

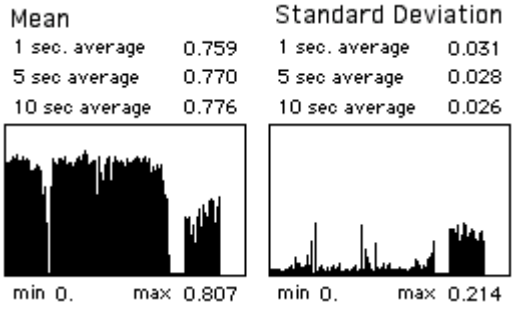

Figure 2. Example statistical analysis of thumb sensor for *Ghat*, averaged over time.

These statistics are stored in a continuously updated array, which can be saved to disk at any point. This allows for the creation of many files of averages, or statistical windows, pulled from varying points of the source recording.

[1] www.cycling74.com

[2] http://web.media.mit.edu/~tristan/

3.2. Separating classes through analysis

Averaged class data was accumulated in a spreadsheet program, and graphs were created comparing the different classes. Classes could be visually separated through different statistical analyses: for example, comparing the standard deviation of thumb sensor data clearly distinguished *Jhala* from the other three classes (see fig. 4), while the mean fret sensor data identified *Ghat* (see fig. 5).

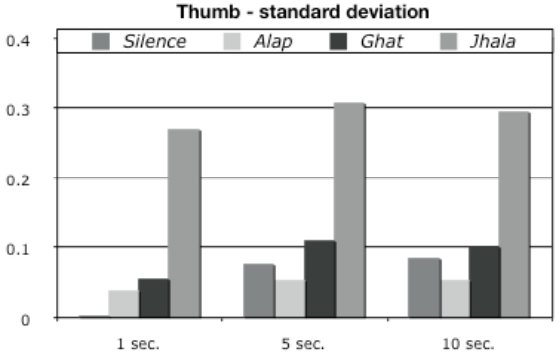

Figure 3. Comparing standard deviation of thumb sensor data averaged over time indicated, between the four classes. *Jhala* is clearly distinguishable over all three averaging periods.

Although some differences were discovered between one, five, and ten second windows (see fig. 4 and 5), the data averaged over one second was similar to that averaged over ten seconds. Therefore, it was predicted that incoming data in real-time, averaged over one second, should show similar consistency.

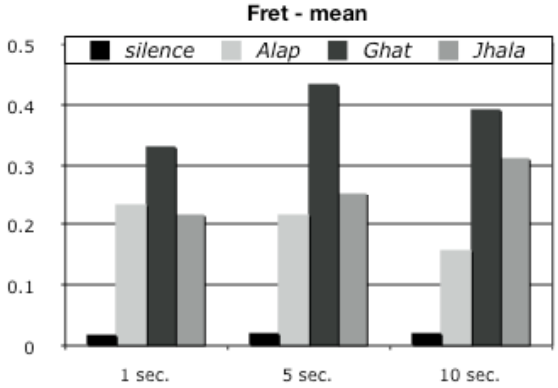

Figure 4. Mean fret sensor data showing different averaging times, clearly distinguishing *Ghat* in each case.

4. REAL-TIME ANALYSIS

Nine statistical analyses were deemed to provide clear distinctions between classes: the mean and standard deviation of both thumb and fret data; the mean and standard deviation of loudness and brightness, and the standard deviation of noisiness.

4.1. Multi-agent analysis

Another program, LISTEN, was written in Max/MSP, in which individual agents – based upon the nine

analysis types - were created to compare incoming performance data to prior analysis data in real-time.

At program initialization, all prior analyses of class recordings were loaded and averaged, thereby creating an overall average of each class: these values were then sent to the individual analysis agents. A normalizing coefficient was calculated for each agent, so as to maximize the potential differences between classes (see table 1). Incoming parameter values were normalized to the same value, and evaluated using fuzzy logic.

	Silence	*Alap*	*Ghat*	*Jhala*
Original	0.460	0.742	0.647	0.379
Normalized	0.558	0.9	0.785	0.46

Table 1. Normalized agent values (to 0.9 rather than 1.0 to allow for predicate roll-off). In this case, mean thumb data.

4.2. Fuzzy logic rating

Fuzzy logic predicates are created around the class coefficients for the agent's particular statistic (see fig. 7). These predicates are scaled according to their inverse proximity to adjacent values.

For example, the thumb sensor's standard deviation clearly distinguishes *Jhala*; therefore, in evaluating membership in the class *Jhala*, the standard deviation thumb agent would be quite confident (*a* in fig. 6). The same agent, however, would be less confident in evaluating other classes since their predicate centers are less discrete (*b* in fig. 6).

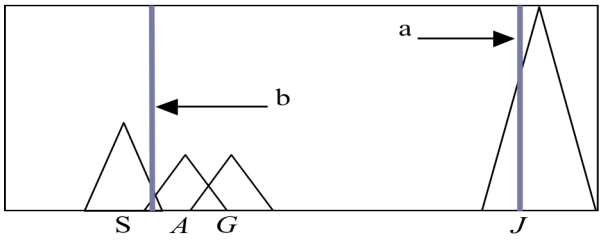

Figure 5. Fuzzy logic predicates for silence (S), *Alap* (A), *Ghat* (G), and *Jhala* (J) for standard deviation thumb data. Incoming sensor data (a, b) is indicated with the gray lines. (a) is clearly *Jhala*, while (b) could be Silence or *Alap*.

Each agent communicates a membership rating ("truth value") for each class (see table 2) once per second; these values are then accumulated globally.

	Silence	*Alap*	*Ghat*	*Jhala*
(a)	0.	0.	0.	0.7
(b)	0.2	0.12	0.	0.

Table 2. Example membership ratings for fig. 7, standard deviation Thumb sensor. (a) is "most likely" *Jhala*, while (b) is "possibly" Silence or *Alap*.

A global evaluator accumulates all agents' membership ratings, and determines the most likely class (see fig. 6).

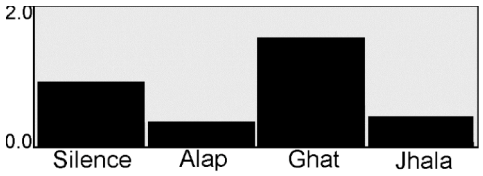

Figure 6. Agent evaluations are accumulated, and the highest rated class is selected, in this case *Ghat*.

Inertia is added to the system at the agent level, so as to avoid rapid changes in evaluation: this is essentially a low-pass filter to logarithmically smooth out data.

5. SYSTEM EVALUATION

5.1. Multi-agent success rating

Live input from the ESitar, playing *Jhala*, was analysed by the multi-agent model, using the statistical analyses averaged over one second. This multi-agent model produced an accurate and consistent analysis over one minute (see fig. 7).

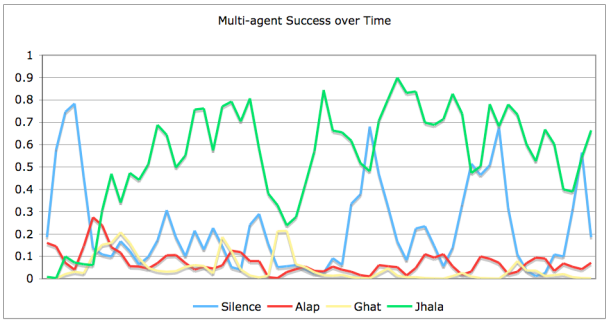

Figure 7. Multi-agent evaluation over one minute of *Jhala*, showing a consistently accurate evaluation. Due to the inertia in the system, the initial evaluation of silence took several seconds to be overcome.

The system was able to successfully determine the playing style. This multi-agent model can be easily strengthened and improved by adding more feature detections (see section 6).

6. FUTURE DIRECTIONS

The agents are rather simple at the moment, not social, and only have limited proactiveness. In the next implementation, higher-level agents will be assigned by class, rather than analysis type – for example, a *Ghat* agent. This would allow for the agent to draw upon ratings from the current, more low-level agents, to provide cumulative and contrasting analyses as needed in order to reaffirm its evaluation.

7. CONCLUSIONS

There is tremendous potential in using sensor data analysis to augment audio analysis as a method of intelligently determining musical activity. In this paper, we have shown that combining sensor analysis with audio feature extraction can determine high-level musical features, specifically performance style.

8. ACKNOWLEDGEMENTS

The authors would like to thank Ge Wang and Rebecca Fiebrink for the implementing audio features in ChucK. Peter Elsea for his Lobjects used in the fuzzy logic, Tristan Jehan for his spectral analysis objects, and Jasch for his statistical objects.

9. REFERENCES

[1] Aimi, R. "New Expressive Percussion Instruments", M.S. Thesis, MIT Media Laboratory, 2002.

[2] Alkon, D.L. "Memory Storage and Neural Systems", *Scientific America*, pp. 42-50, 1989.

[3] Alonso, M., Richard, G., and Bertrand, D. "Tempo and Beat Estimation of Musical Signals", *International Conference on Music Information Retrieval (ISMIR)*, Barcelona, Spain, 2004.

[4] Atkeson, C. G., Hale, J., Pollick, F., Riley, M., Kotosaka, S., Schaal, S., Shibata, T., Tevatia, G., Vijayakumar, S., Ude, A., and Kawato, M. "Using Humanoid Robots to Study Human Behavior," *IEEE Intelligent Systems: Special Issue on Humanoid Robotics*, vol. 15, pp. 46-56, 2000.

[5] Aucouturier, J., Pachet, F., and Hanappe, P. "From Sound Sampling to Song Sampling," *ISMIR*, Barcelona, Spain, 2004.

[6] Boersma, P. "Accurate Short-Term Analysis of the Fundamental Frequency and the Harmonics-to-Noise Ratio of a Sampled Sound," *Proceedings of the Institute of Phonetic Sciences*, Amsterdam, 1993.

[7] Chouvel, J., Bresson, J., Agon, C. "L'analyse musicale différentielle: principes, représentation et application à l'analyse de l'interprétation", *EMS Network*, http://www.ems-network.org/spip.php?article294

[8] Dixon, S. "A lightweight multi-agent musical beat tracking system", *Pacific Rim International Conference on Artificial Intelligence*, Melbourne, Australia, 2000.

[9] Eigenfeldt, A. "*Drum Circle:* Intelligent Agents in Max/MSP", *Proceedings of the International Computer Music Conference (ICMC)*, Copenhagen, Denmark, 2007.

[10] Goto, M., Muraoka, Y. "Beat Tracking based on Multiple-agent Architecture - A Real-time Beat Tracking System for Audio Signals", *Proceedings of The Second International Conference on Multiagent Systems*, Kyoto, Japan, 1996.

[11] Gouyon, F., Klapuri, A., Dixon, S., Alonso, M., Tzanetakis, G., Uhle, C., and Cano, P. "An Experimental Comparison of Audio Tempo Induction Algorithms", *IEEE Transactions on Speech and Audio Processing*, vol. 14, 2006.

[12] Kapur, A., Lazier, A., Davidson, P., Wilson, R. S., and P. R. Cook. "The Electronic Sitar Controller", *Proceedings of the 2004 Conference on New Interfaces for Musical Expression (NIME)*, Hamamatsu, Japan, 2004.

[13] Kapur, A., Tindale, A, Benning, M. S. and P. Driessen. "The KiOm: A Paradigm for Collaborative Controller Design", *ICMC*, New Orleans, USA, 2006.

[14] Klapuri, A. "Automatic Music Transcription as we Know it Today", *Journal of New Music Research*, vol. 33, pp. 269-282, 2004.

[15] Li, M., Wei, G., Petrushin, V., Sethi, I. "Developing Audio Processing Agents for Multi-Agent MPEG-7 Enabled Environment", *4th Intl. Workshop on Multimedia Data Mining*, Washington, DC, 2003.

[16] Loscos, A., Wang, Y., and Boo, W. "Low Level Descriptors for Automatic Violin Transcription", *ISMIR*, Victoria, BC, 2006.

[17] Nakatani, T., Okuno, H., Goto, M., Ito, T. "Multiagent Based Binaural Sound Stream Segregation", *Computaional Auditory Scene Analysis*, Lawrence Erlbaum Associates, Philadelphia, PA, 1997.

[18] Rowe, R**.** *Machine Musicianship*. MIT Press, Cambridge, MA, 2004.

[19] Schierer, E. "Tempo and Beat Analysis of Acoustic Musical Signals", *Journal of the Acoustical Society of America* vol. 103, pp. 588-601, 1998.

[20] Wang G. and Cook, P. R. 2003. "ChucK: A Concurrent, On-the-fly Audio Programming Language", *ICMC*, Singapore, 2003.

[21] Wright, M., OpenSound Control Specification, 2002. http://www.cnmat.berkeley.edu/OSC/OSC-spec.html