Musical Score Generation in *Valses and Etudes*

David Kim-Boyle
University of Maryland, Baltimore County
Department of Music
1000 Hilltop Circle, Baltimore, MD 21250
1-410-455 8190
kimboyle@umbc.edu

ABSTRACT

The author describes a recent composition for piano and computer in which the score performed by the pianist, read from a computer monitor, is generated in real-time from a vocabulary of predetermined scanned score excerpts. The author outlines the algorithm used to choose and display a particular excerpt and describes some of the musical difficulties faced by the pianist in a performance of the work.

Keywords

Score generation, Jitter.

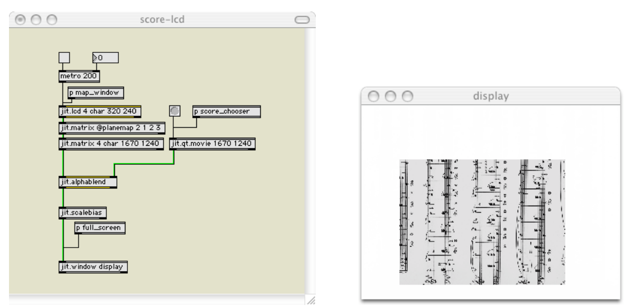
1. INTRODUCTION

In *Valses and Etudes*, a recent work for piano and computer premiered at the 2005 Florida Electroacoustic Music Festival, the score performed by the pianist is generated in real-time from a variety of scanned score excerpts. These excerpts are taken from a variety of existing works including Movement VI of Schoenberg's 6 Kleine Klavierstücke Op. 19, the Second Movement of Webern's Variationen für Klavier Op. 27, and several of Ravel's Valses Nobles et Sentimentales amongst others. A number of additional pieces are played by the computer, from pre-existing recordings, at several points in the work but are not called upon in the score generation process. The score selected for performance is conditioned by Markov chain probabilities and the actual score excerpt displayed for the pianist is determined through a Jitter patch. This excerpt is not fixed, but dynamically varies during the performance. Unlike previous works of the author in which the original materials are extensively processed, in *Valses and Etudes* very little audio processing takes place. Rather, the musical complexity lies in the simultaneous performance of up to twelve or so pieces by the computer, and the interaction within this musical tapestry by the pianist. Aesthetically, the process recalls the mesostics of John Cage where pre-existing works are *read* through in a manner that creates new meanings from the amplification *and* omission of detail. [1]

2. SCORE GENERATION METHOD

Unlike traditional methods of score generation in which algorithms generate a score from which the performer/s learn and then perform the piece, [2] in *Valses and Etudes* the score is generated in real-time from eight predetermined scanned score excerpts. How these excerpts are displayed and ordered is determined in real-time during the performance itself.

The first page of each of the preselected scores was scanned, edited and saved as high resolution jpeg files for further processing in Cycling'74s Jitter environment. [3] In *Valses and Etudes*, Jitter is used to frame particular score excerpts. This framing patch is illustrated in Figure 1a, where a small, random window is generated with the jit.lcd object with values mapped to the alpha channel and then blended with the jpeg score file. Large Jitter matrices are used to maximize the screen resolution. A typical result, as viewed by the pianist, is illustrated in Figure 1b.



a) b)

Figure 1. a) Framing in Jitter, b) Typical framing result.

The windows mapped by Jitter are not fixed, as implied in Figure 1b, but dynamically change during performance. Their size, rate of size change, trajectory across the score page, and speed of movement are all definable. Each of these parameters can also be randomized. Clearly, different rates of change will produce qualitatively different interpretations. Trajectory paths for eight possible score selections are defined with one lcd object and can be determined during or prior to the performance.

The author has also experimented with Thiebaut's *trajectory* object. [4] This object requires a different implementation but facilitates geometrical trajectories which are not easily obtainable otherwise. Unfortunately, it does not allow multiple trajectories to be simultaneously defined which yields a more cumbersome interface.

In *Valses and Etudes*, source scores are selected with a first order Markov Chain algorithm where probabilities are determined with eight multisliders. The interface for this is illustrated in Figure 2.

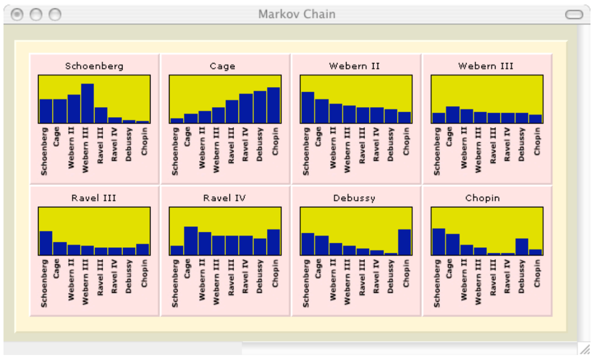


Figure 2. Markov chain interface.

In *Valses and Etudes* each multislider in the above interface represents one score performed by the pianist. The eight sliders of each multislider represent the probability that another score will follow. For example, the top left multislider determines the probabilities that another score will follow an instance of the sixth movement of Schoenberg's 6 Kleine Klavierstücke. There is around a 50 percent probability that the Schoenberg score will be followed by the Schoenberg score or the Cage score, a slightly higher probability that it will be followed by the two Webern scores, and a decreasing probability that it will be followed by the Ravel, Debussy, or Chopin scores. The engine behind this interface system is based on the Max *prob* object. In *Valses and Etudes* one *prob* object is used to store all 64 possible transitions. The Markov chain process is able to lend the work a spontaneity that remains nevertheless musically coherent.

3. PERFORMANCE DIFFICULTIES

One of the obvious difficulties the pianist faces in performing the work is the need to learn eight individual pieces. This prospect is made somewhat less daunting given that only the first page of each score for each piece need actually be learned and that the pieces chosen are not too technically demanding.

The fact that the pianist has no knowledge of which piece will follow another affects their interpretation in a more significant way. Winkler notes a similar issue with his real-time score generation technique. [5] That the windowing process might also be different from score to score, for example with different trajectories, window sizes and speeds, adds another layer of complexity which further disrupts interpretative continuity from work to work.

The most obvious challenge for the pianist, as mentioned, is the effect of the windowing process on the interpretation of each of the source pieces. This is particularly challenging as it goes against much in the way of traditional performance practice. To be faced with a score that is dynamically changing during performance or where only a fragment of the score may be visible, or even to be faced with a trajectory that moves from the bottom of the page to the top, forces the pianist to abandon, to a certain extent, traditional interpretative concepts of form and development. The most musically effective solutions have involved simply performing coherent segments of a score in short phrases. This lends the pianist's performance a somewhat episodic quality which nevertheless blends in seamlessly with the constantly varying computer part.

4. FUTURE DEVELOPMENT

In the original form of the work, the order in which the piano works were played back by the computer was fixed. In more recent explorations, the author has begun experimenting with more open form Max patches that determine work selections based on an extension of the first order Markov chain process outlined in Section 2. The author is also interested in experimenting with the use of MIDI files rather than actual recordings in the computer part. This could enable interesting morphing patterns to occur between works. [6] Also being explored is a more responsive musical process whereby the performer is not simply responding to events determined by the computer but becomes more of a musical instigator. [5] This will necessarily involve gesture recognition [7][8] and score following [9][10] techniques to enable the computer to accurately determine which scores are being played and how they are being interpreted.

5. REFERENCES

[1] Cage, J. *I-IV*. Cambridge, MA: Harvard University Press, 1990.

[2] Maurer, J. A. A Brief History of Algorithmic Composition. Available at <http://ccrma-www.stanford.edu/~blackrse/algorithm.html>. Winter 1999.

[3] Zicarelli, D. An Extensible Real-Time Signal Processing Environment for MAX. In *Proceedings of the 1998 International Computer Music Conference*. Ann Arbor, MI: International Computer Music Association, 463-466.

[4] Thiebaut, J.-B. Available at <http://www.mshparisnord.org/cicm/dl_en.htm>. April 2004.

[5] Winkler, G. E. The Realtime Score. A Missing Link in Computer Music Performance. In *Proceedings of the 2004 Sound and Music Computing Conference*. Paris. Available at <http://smc04.ircam.fr/scm04actes/P3.pdf>.

[6] Anderson, C. "Audible Interfaces Festival." (trans. P. Castine). In *Computer Music Journal*, Vol. 26 No. 4, Winter 2002.

[7] Camurri, A. and G. Volpe (Eds.). *Gesture-Based Communication in Human-Computer Interaction*. Berlin: Springer-Verlag, 2004.

[8] Traube, C., P. Depalle, and M. Wanderly. Indirect Acquistion of Instrumental Gesture Based on Signal, Physical and Perceptual Information. In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03)*. Montreal. Available at <http://www.nime.org>.

[9] Dannenberg, R. An On-Line Algorithm for Real-Time Accompaniment. In *Proceedings of the 1984 International Computer Music Conference*. Paris: International Computer Music Association, 193-198.

[10] Orio, N., S. Lemouton, and D. Schwarz. Score Following: State of the Art and New Developments. In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03)*. Montreal. Available at <http://www.nime.org>.