音楽情報科学 23－8
（1997．12．15）

Two Highly Integrated Real-Time Music and Graphics Performance Systems

Robert Rowe *
Eric L. Singer**

Media Research Laboratory, New York University*
robert.rowe@nyu.edu
Media Research Laboratory, New York University**
esinger@cat.nyu.edu

Abstract

We describe two systems coupling animated graphic displays with real-time music analysis. One of these systems is for stage performance involving an ensemble of six instrumentalists and live video projection. The other allows users to perform with members of an animated virtual jazz band. The stage work, A Flock of Words, generates real-time animation of graphic objects in response to analysis of musical input, coupled with the simultaneous presentation of video, lighting, large-scale holograms, and algorithmically generated computer music. The Interactive Virtual Musicians project presents animated jazz musicians that can improvise with users performing on MIDI instruments.

2つの高度に統合された実時間音楽・画像演奏システム

ロバート・ロー *
エリック L. シンガー**

ニューヨーク大学 メディア・リサーチ・ラボラトリー*
robert.rowe@nyu.edu
ニューヨーク大学 メディア・リサーチ・ラボラトリー**
esinger@cat.nyu.edu

実時間音楽解析と動画を組み合わせた二つのシステムについて述べる。一つは6人の演奏家によるアンサンブルとライブ・ビデオ映像を伴う舞台演奏の為のものである。もう一つはユーザが動画化された仮想ジャズバンドのメンバーと一緒に演奏できるというものである。舞台作品、「A Flock of Woods」では、入力された音楽を分析し、それに呼応して実時間で動くグラフィック・オブジェクトを生成する。このグラフィック・オブジェクトは、同時に映写されるビデオ、照明、大型のホログラム、そしてアルゴリズムによって生成されるコンピュータ音楽と組み合わされる。インタラクティブ仮想ミュージシャン・プロジェクトはMIDI楽器を演奏するユーザと共に即興演奏を行なう動くジャズ・ミュージシャンを提示する。

1 A Flock of Words

A Flock of Words is a collaboration of composer Robert Rowe and video artist/holographer Doris Vila, with programming and animation design by Eric Singer. The work was premiered at New York

University in the spring of 1995. The instrumental ensemble for the piece consists of violin, viola, cello, MIDI keyboard and two percussionists. Their performance is tracked in real time through the MIDI keyboard and a percussion sensor.

The elements of the technical setup of the piece are as follows: 3 Apple Macintosh computers, 3 video projectors, 3 robotic lighting elements, a MIDI-controlled light board, a laser, 2 large holograms and sound synthesis equipment. Two of the computers are used to cue up and play back video clips and to generate animations in real time. One of these (video computer 1) sends an identical signal to two video projectors, each projecting onto large scale holograms on either side of the instrumental ensemble. The second video computer projects a different set of videos and animations onto a large screen at the back of the stage.

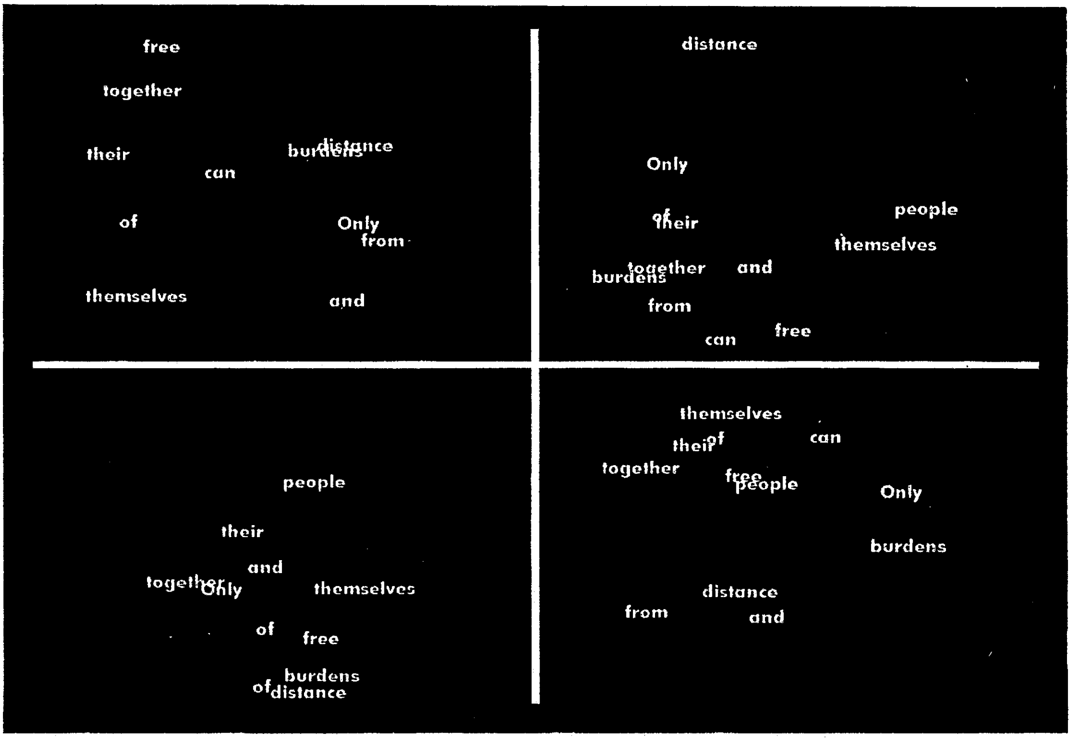
All of the software for A Flock of Words is written in C by the authors. Performance information from the ensemble is analyzed by a program running on the master computer and looking for musical attributes such as register, density and articulation. Software on the video computers is responsible for receiving control messages from the analysis software and sending video and animation to the display projectors. The analysis and video machines communicate through MIDI connections and a set of MIDI messages which we define. The master computer also generates a stream of MIDI-encoded music which is sent to a synthesizer and effects processor to accompany the live ensemble and control lighting effects through a dedicated show controller. Video, animations and lights are projected onto the screen behind the ensemble as well as the holograms flanking the ensemble, changing their appearance as the piece progresses.

Displayed video consists of prerecorded video clips and real-time generated animation. The video clips are stored and played back from QuickTime movie files. The animation is based on Craig Reynolds Boids algorithm from an implementation by Simon Fraser. To create the Flock of Words animation, a set consisting of 10 to 30 words is selected from the text of Crowds and Power by Elias Canetti and animated using the flocking algorithm. The center point of each word (or Woid) is treated as the center of a Boid and animated under real-time control. Numerous parameters are used to change the flocking attributes and influence the look of the flock. These include the speed and acceleration of the Woids; their tendency to stay close to the center of the flock; to avoid each other; to follow a point on the screen; and to avoid the edges of the screen.

Figure 1 shows four characteristic Woid flocks in mid-flight. The flock in the lower left, for example, is moving toward a point of attraction centered near the bottom of the screen. The flock above shows the Woids in a pattern of greater dispersal due to a change of direction or increase in neighbor avoidance. Because the objects are words, they continually enter into new grammatical relationships with each other as they fly, sometimes aligning as in the original text, other times garbling the syntactic order entirely. The continual fluctuation between linear sense and nonsense was one of the motivations for this approach: the force of Canetti s original text is always present through the normal-order presentation in the video clips and balances with the non-linear presentation in the animation.

By linking various flocking parameters to messages sent by the analysis software, which are in turn derived from musical gestures of the performers, we achieve the effect of the flock being controlled by the performers. The full set of video controls available to the analysis software includes selection of video clips; selection of word sets from the text; color of the Woids and background screen; and twelve flocking parameters.

Figure 1: Screen displays of Woid flocks



2 Interactive Virtual Musicians

The Interactive Virtual Musicians (IVM) system consists of two major software subsystems: the IMPROV software and the IVM control software. IMPROV enables authors to create real-time behavior- based character animation. Characters in IMPROV, known as virtual actors, are artificially intelligent,autonomous and directable. They are endowed with a library of animated actions, movements and gestures as well as individual personalities created by a programmer or animator using a scripting system. Characters are then able to generate their own animation sequences based on external input and influences (such as user input, musical analysis software and the actions of other characters) and in accordance with their personality traits.

The IVM control software is the main focus of our development work. It extends the functionality of the virtual actors to include music performance, thereby turning them into virtual musicians. IVM is responsible for receiving and interpreting the various forms of input, generating and playing the musical performance and directing the graphical performance of IMPROV.

Various modes of input are used to influence the musical accompaniment and improvisation performed by the virtual musicians as well as their animated performance. For example, a human musician can play chord changes while a virtual saxophonist improvises along, or a person can pick up the conductor's baton and lead a jazz trio. The mood of the performance is controlled by the user and is reflected by the virtual musicians in the tone of their musical performance and the body language of their animated performance. In this way, the IVM system presents an integrated visual and musical environment.

Input can be received from MIDI sources, including keyboards, electronic drums, and pitch-to-MIDI converters, analyzed in real-time, and used to direct or influence the performance by the virtual musicians. The analysis software is based on Cypher, a music analysis/generation interactive system previously developed by Robert Rowe [Rowe 1993]. IVM extends Cypher in many ways including better recognition of chords, contexts to include jazz harmony, and incorporation of real-time pattern recognition to find important melodic and rhythmic units [Rowe & Li 1995].

IVM performs a two-stage analysis of the musical input. The first stage is a lower-level analysis, yielding such properties as velocity, duration, register, harmony and chord density. The second-stage analysis uses the low-level properties to derive more abstract, subjective attributes. These might include properties like swingness, grooviness or drunkenness.

IVM has several methods for generating, playing, processing and outputting musical performances. Score files or sequencesin the form of Standard MIDI Files can be read, processed and played. Sequence processing methods include altering and transposing pitches, embellishing lines, adding inflections such as pitch bend, changing accenting and time-shifting notes.

IVM is also capable of generating improvisation. It can improvise constrained random lines and patterns based on chord changes and can also generate Markov-based improvisations. Generation methods based on Cypher are incorporated as well. To output musical material, IVM drives both MIDI synthesizers and software synthesizers such as Csound. Software synthesizers are played by means of messages sent to the synthesis programs via network communication.

Character animation is controlled by IVM, which sends messages to IMPROV via the network. Characters are controlled on several levels simultaneously. Low-level commands control specific physical actions of the characters such as moving the fingers of a virtual saxophonist or the hands of a virtual drummer. Higher-level commands communicate information about the musical performance, user input and other environment variables. This information influences the animated performance in various ways based on each character s programmed personality. Among their many capabilities, virtual musicians can groove along to the music, tap their feet in time and raise their horns in the air during their solos.

The IVM control software is written in C++ and runs on a Macintosh. IMPROV is implemented in Java and VRML and runs under any VRML2-compliant browser. Currently, we use CosmoPlayer on an SGI. IVM communicates with IMPROV and other programs across a local-area network. The system makes extensive use of network communication (Telnet/TCP and UDP connections), which we have found to be a versatile and effective way to tie together system components.

3 Conclusion

The common thread through both of these projects is a tight coupling between real-time musical analysis and the presentation of graphics and animation. Computing the ongoing behavior of musical attributes makes it possible for direct analogs between performance and image to be established both during stage performance and in unstructured installations. The challenge in realizing these projects lies in implementing the analysis and distributing the computation in such a way that the whole can be accomplished in real time. Because we use multiple computers, communications schemes become an integral part of the architecture. We have successfully used both MIDI and Telnet protocols as semaphores between music analysis/generation and graphics machines. Both of these projects demonstrate the salience of music/graphic interaction that can be achieved with real-time analysis.

4 Acknowledgments

The authors gratefully acknowledge the contributions of the following people: Doris Vila, who was responsible for the artistic conception and realization of the graphics in A Flock of Words; Ken Perlin and Athomas Goldberg, for creating and developing IMPROV; Clilly Castiglia, for directing the IVM project; Tamara Smyth, for her help with Java; Sabrina Liao, Agnieszka Roginska and Andrea Bonotto, for their contributions to the IVM project; and the rest of our colleagues at the NYU Center for Advanced Technology, NYU Media Research Laboratory and The University of University of Sco Paulo, Brazil. A Flock of Words was partially funded by the NYU School of Education Research Challenge Fund. The Interactive Virtual Musicians project is funded by the NYU Center for Advanced Technology. Portions of this paper were previously published in the Proceedings of the 1997 International Computer Music Conference.

References

[1] Canetti, E. 1984. Masse und Macht. English title Crowds and Power; translated from the German by Carol Stewart. New York: Farrar Straus Giroux.

[2] Perlin, K. 1995 Real Time Responsive Animation with Personality In IEEE Transactions on Visualization and Computer Graphics: New York: IEEE

[3] Perlin, K.; and Goldberg, A. 1996 IMPROV: A System for Scripting Interactive Actors in Virtual Worlds In Proceedings of SIGGRAPH 96,, Annual Conference Series: New York: ACM SIGGRAPH

[4] Reynolds, C. W. 1987 Flocks, Herds, and Schools: A Distributed Behavioral Model , In Computer Graphics, SIGGRAPH 87 Conference Proceedings: New York: ACM SIGGRAPH

[5] Rowe, R. 1993. Interactive Music Systems: Machine Listening and Composing. Cambridge, Mass.: The MIT Press.

[6] Rowe, R.; and Li, T.C. 1995. Pattern Processing in Music In Proceedings, Fifth Biennial Symposium for Arts and Technology: Center for Arts and Technology at Connecticut College

[7] Rowe, R.; Singer, E.L.; and Vila, D. 1996. A Flock of Words: Real-Time Animation and Video Controlled by Algorithmic Music Analysis In Visual Proceedings, SIGGRAPH 96: New York: ACM SIGGRAPH.

[8] Singer, E.L.; Perlin, K.; and Castiglia, C. 1996. Real-Time Responsive Synthetic Dancers and Musicians In Visual Proceedings, SIGGRAPH 96: New York: ACM SIGGRAPH.

[9] Singer, E.L.; Goldberg, A.; Perlin, K.; Castiglia, C.; and Liao, S. 1997 Improv: Interactive Improvisational Animation and Music ISEA 96 Proceedings, Seventh International Symposium on Electronic Art: Rotterdam, Netherlands: ISEA96 Foundation