STOCHGRAN ON SILICON GRAPHICS IRICS

Mara Helmuth
University of Cincinnati
College-Conservatory of Music Center for Computer Music - (ccm)2
PO Box 210095, Cincinnati, OH 45221-0095
mara@ping.ccm.uc.edu, Mara.Helmuth@uc.edu

Abstract:

StochGran, the granular synthesis composition program, has been ported from NeXTstep to the Silicon Graphics platform, redesigned and expanded. Flexibility of sound design and clarity of representation are central concerns. The new version is written in RapidApp using Motif under X windows. Included new features are 1) display of function data on the central graph for a more integrated display, 2) granular sampling, 3) graphical definition of synthesis waveforms and grain envelopes, and 4) additional functions for more flexible musical gesture definition. While the application is still primarily oriented toward the composition of complex non-real time events, real time options have been implemented for improvisational situations and demonstrations.

1. Introduction

Granular synthesis and sampling are becoming important and widely used techniques for synthesis, time-stretching and compressing sound, and for obtaining new sounds which may be transformations of acoustic sound. Xenakis was the first composer to describe the idea in Formalized Music (Xenakis, 1971). Barry Truax (Truax, 1988), Curtis Roads (Roads, 1978), Jean Piche, Cort Lippe (Lippe, 1993). Jon Nelson and others have implemented and used these techniques in composition. In 1991-1992 I programmed the NeXTstep version of StochGran, a granular synthesis application based on a Cmix instrument (Helmuth, 1991), and wrote other Cmix instruments for granular sampling. The interface contained a set of graphs for easy editing of functions to shape musical gestures. The port to the Silicon Graphics platform, with its extensions, is the subject of this paper. StochGran maintains an emphasis on a compositional approach which prioritizes control over real time performance, giving the composer the power to shape music gestures with probability distributions and shape of change functions to control grain parameters. However, real time performance has been implemented, for a more spontaneous way of creating sound.

The superior audio, graphical and programming capabilities of the Silicon Graphics made this platform an excellent choice for a music composition program with graphical sound representation. The program is available on IRIX 6.x and 5.3. The Rapidapp Interface Building development tools generated Motif and C++ code for the X windows-based system. The OpenGL Graphics Library was used for the graphical displays. Instruments were written in Paul Lansky’s Cmix music programming language (Lansky, 1990) to create sound.

2. Granular Synthesis

The strategy for synthesis has been described previously, (Helmuth, 1996). Unlike some other granular software which generates a continuous stream of one or more layers of grains, StochGran actuates the composer’s selected grain rates and durations, modified by probability distributions which express points of preference and amounts of preference for certain values. Consequently the densities may be widely varying, from moments with thousand of overlapping grain layers, to very sparse times which contain mostly silence, even within the same event. Events contain changing parameter distributions, that form gestures with rising and falling frequencies, which may cause timbral changes within the event, accelerations and decelerations of grain rates, which may actually sound like frequencies rising and falling, when at audio level rates, and durations, which cause density changes. The composer can shape these changes rather precisely and graphically in StochGran, creating and reforming sound until happy with the results. In addition to grain rate, duration and frequency, in the synthesis program, location can be specified for stereo files, with the same
gestural control and probability distributions as in the other parameters. The composer selects from several tabs of a tabbed deck to edit particular parameters of the event. The main tab shows four graphs of shape of change functions for grain parameters. Illustration 1 shows three of the four grain parameters in the main display tab: the unique shape of change functions for grain rate, duration and frequency for a synthesis event. Grain rates zoom jaggedly up and down throughout the event. Grain durations move first in similar motion with the rates, and then move out of synchrony in opposing directions. The frequency curve moves independently of the rate and duration curves in rounded curves between its extreme values. The definition of each parameter's extreme values (zero and one) are defined by slider/
form fields in the zoom window tab display, where each curve can also be edited in more detail. In the grain tab, grain waveform (for synthesis) and grain envelopes are graphically defined, as well as timing data for the output sound.

The default situation is non-real time performance, so a Cmix soundfile for the included “sgrain” instrument and soundfile are written to the disk before StochGran plays the sound. The scorefile can be edited and run later outside of StochGran if needed, and the soundfile can be used in other applications. A graph of each sound can now be saved as well as a soundfile as a compositional aid, since the lengthy parameter list can be difficult to interpret without hearing the sound. StochGran can also now read in scorefiles it has written out, if the format has not been substantially altered.

3. Granular Sampling

Granular sampling is the use of sampled acoustic sound within the grain, rather than synthesized waveforms. This type of processing has the same advantage of granular synthesis in that an incredible variety of sounds are possible, but, the sounds can retain a relationship to the acoustic sound. This means that the sound families which can be created from acoustic sound are expanded and varied in sometimes unimaginable ways. When an input soundfile is opened, StochGran makes a couple of changes to the grain parameters: 1., the “frequency” specifications now refer to transposition of the input sound, and 2., input sound data such as start time, duration and input grain rates become active. A different Cmix instrument, “stgrain” is called, and the scorefile is written out for that instrument before the sound is played. Again, this score is editable and runnable at any point afterwards with the Cmix program.

Sound 2: Granular sampling of sigh sounds

4. Probability Distributions and a More Flexible Option for Parameter Shaping

One limitation of the synthesis algorithm described previously is that the same shape of change curve applies to all four aspects of a probability distribution. The four aspects (or values) of the distribution are low and high, which define the limits of the allowable range, the mid point, which is the preferred point, and the “tightness”, which defines the amount of preference for the mid point. To calculate one grain, the low, high, mid and tightness for each parameter at that point in the event is used to generate the actual grain values. For example, if the current frequency low is 100 Hz, high is 700 Hz, mid is 600 Hz, and tightness is 3, a grain frequency somewhere close to 600 Hz will be randomly generated. These low, high, mid and tightness values for the current grain come from the location on the shape of change function for that parameter at that point in the event, between the zero and one extreme distributions. If the zero distribution has a range between 100 Hz and 1000 Hz, and the one distribution has a range spanning 1000 Hz to 2000 Hz, and the shape of change function moves linearly from zero to one, half-way through the event the low and high values will allow grains between 550 Hz and 1500 Hz.

While this algorithm has many degrees of freedom, occasionally the composer will want more control, and to have different shape of change functions for the low, high, mid or tightness parameters. An option for independent control of these four curves for each parameter makes use of the expanded “fsgrain” instrument which has eighteen function curves. In the fsgrain tab display one can edit the four curves for each parameter independently, if the fsgrain option is selected. For example, one may decide to make the frequency mid point function moving erratically up and down, while the high value function defining the upper limit of the range slopes gradually down. This would create a sound which has the frequency at which most of the grains occur moving up and down, while grains occurring near the upper limit of the range would occur at lower and lower frequencies (closer to the mid point) toward the end of the event.

5. Real time Granular Performance

Barry Truax implemented real time granular performance earlier (Truax, 1988), but with the dense layering I often prefer I kept StochGran non-real time, and explored real time capabilities only with ISPW Max (Helmut, 1993). Because of the recent dramatic increases in processor speed, and for improvisational situations and demonstrations, as well as composition, real time strategies are now being implemented in StochGran. A revision of the sgrain instrument
is being written in RTCMIX (Garton, Topper, 1997), Brad Garton and Dave Topper’s extensions to Lansky’s Cmix. The goal is not to duplicate the original capability of layering up to thousands of grains if necessary, but to focus on the composer’s ability to manipulate the sound in real time. To this end, on the live “jam” tab display, one can graphically select one or a number of parameters, indicate initial and target points for some duration, and then hear the parameters change over time. This is somewhat more flexible than applications which allow only one parameter to change at a time.

6. Algorithmic Score Generation

For the moments when the composer wants to surrender control over the details and explore a world of new sound surprises, StochGran now has several programs included which have previously been used standalone. StochGran’s algorithmic score generators will create any number of scores for synthesis or sampling. The scores are run and sounds are created in a directory of numbered sounds which can then be auditioned for interesting material. This is a compositional method I have found enormously useful in pieces such as *Abandoned Lake in Maine* (1997), based on sampled loon sounds, and *Dragon of the Nebula* (1991), a synthesis piece.

7. Conclusion and Future Work Planned

In the port to Silicon Graphics, StochGran has been improved in several areas. First, the main interface has been simplified and extended for a clearer view of data and functions. More importantly, sampling has been incorporated into the main interface rather than existing in several separate programs, as has algorithmic score generation. The Silicon Graphics Audio Library is being explored as a very efficient method of synthesis or processing sound from a microphone. A Linux version is planned, as well as better data generation for FCurve (Helmuth, Ibrahim, 1995), a program which takes granular synthesis data and creates animation data for Softimage models.

StochGran is available at http://meowing.ccm.uc.edu/ under music software.

References


