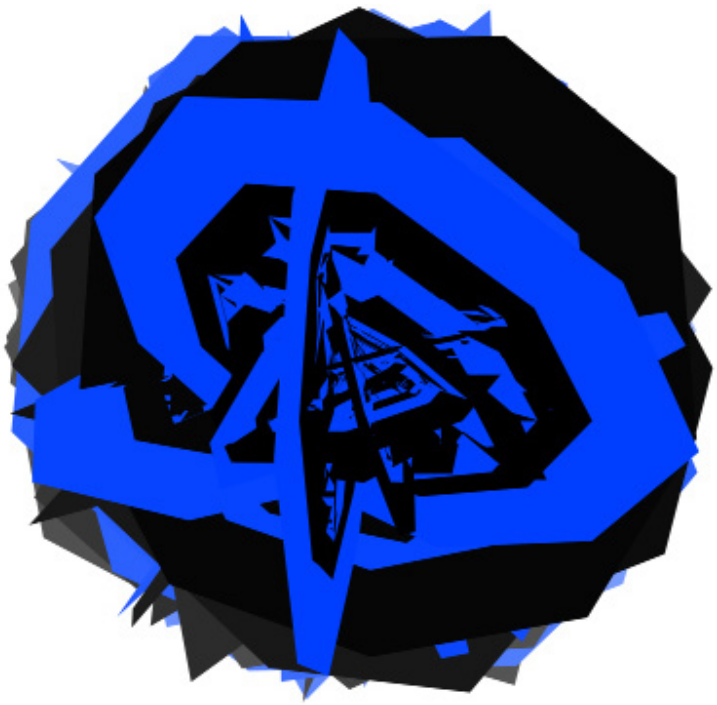
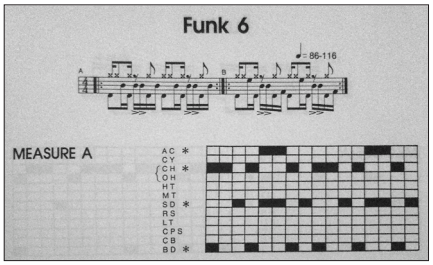


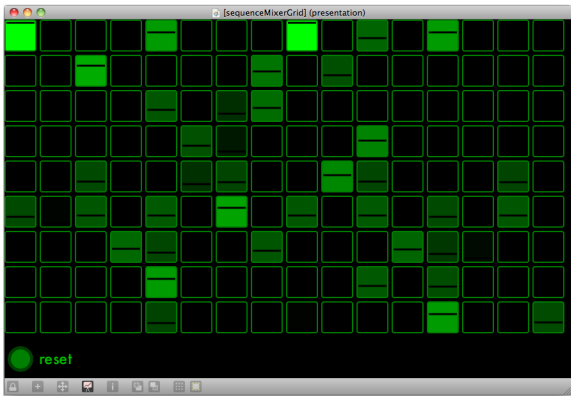
T. A. Gambarotto

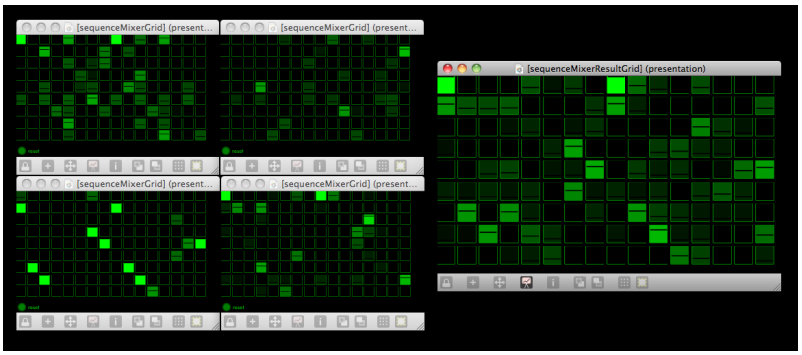
OPERATIONS

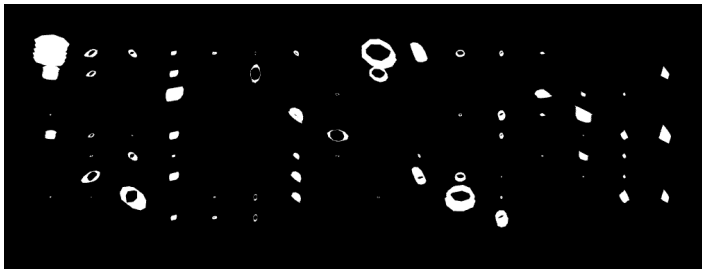
audiovisual performance
2012
project documentation

Overview









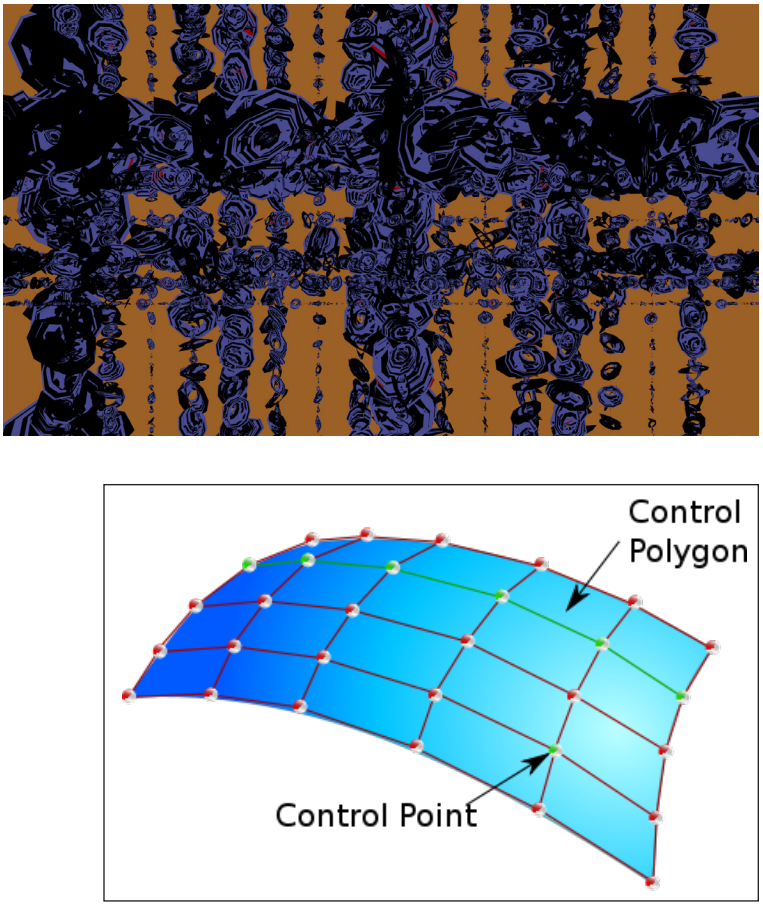
Top to bottom: traditional drum machine grid; Operations source matrix; source matrices added to create master grid; projected representation of master grid.

Operations is a computer generated audiovisual solo performance piece, created on the Max/MSP/Jitter multimedia platform.

The starting point for this piece is the programming of traditional drum machine grid patterns, a type of musical score. This grid is translated into a series of Jitter matrices on multiple mixer channels. Each of these source matrices can be interpolated through nine different sets of values. The source matrices may then be further transformed via inversion or randomization before finally being added together to create a master grid. The master grid is then projected on the screen as a visual representation of the currently playing pattern. However, this pattern is never static, as on a traditional drum machine, but can be continually and seamlessly altered via the mixer.

This projected grid is then subjected to a series of visual transformations, some automated and synchronized to the beat of the audio program, some as a consequence of performer control. The resultant images accumulate on the screen, creating a series of “paintings” that depict the virtual space of the drum pattern through time. These scenes

are organized in a database, such that the performer may control the timing and flow of the performance.



Top: transformations of master grid. Bottom: three-dimensional NURB showing control points.

The title *Operations* derives from Lev Manovich's *The Language of New Media*:

> Regardless of whether a new media designer is working with quantitative data, text, images, video, 3-D space or combinations of them, she employs the same techniques--cut, copy, paste, search, composite, transform, filter... I will call these typical techniques of working with computer media operations... While operations are embedded in software, they are not tied to it. They are employed not only within the computer but also in the social world outside it. As we work with software and use the operations embedded in it, these operations become part of how we understand ourselves, others, and the world. (p.118)

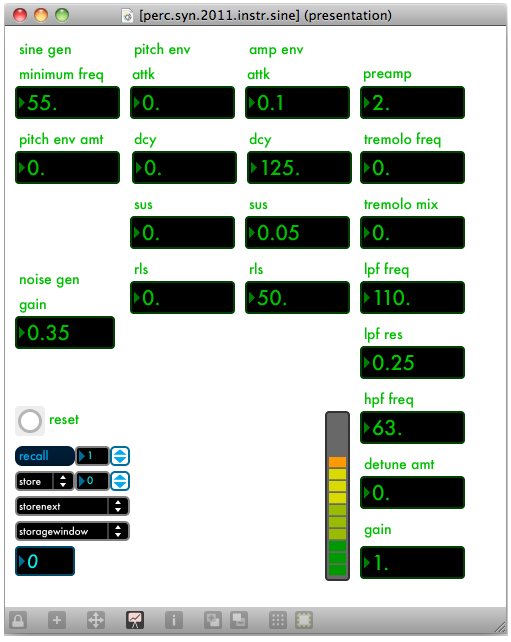
Image

The animation is generated in real time. No fixed video is employed. The animation is rendered entirely using OpenGL's implementation of three-dimensional non-uniform rational basis splines (NURBs), a mathematical model used in computer graphics for generating and representing curved surfaces. This type of model allows for the creation of complex, organic shapes with relatively few defined control points. It is thus open to complex

real-time manipulation; for example, scene transitions in *Operations* are accomplished via morphing rather than hard edits.

Additionally, the generated soundtrack of the animation, in its form as a linear stream of digital audio data, is added to the control points, such that the resultant surfaces continually ripple and pulse with organic motion. Each cell of the master grid is rendered as a single NURB, and it's size relative to the other cells corresponds to its volume level in the current mix.

Sound

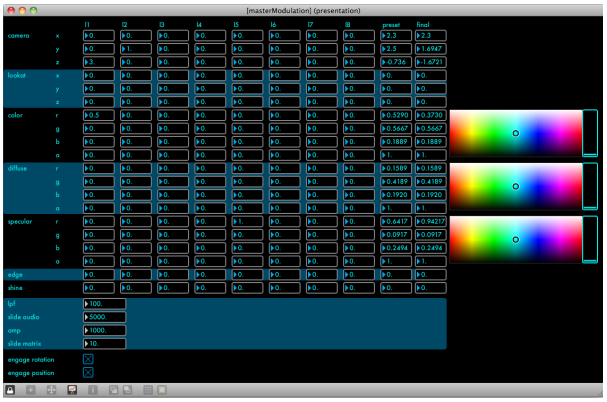


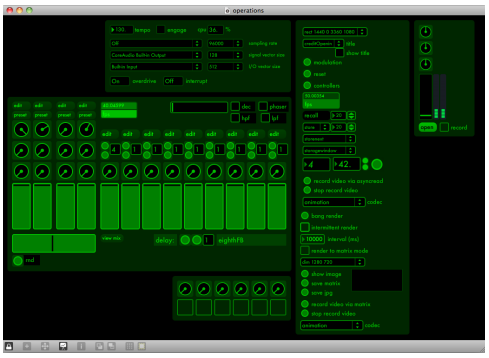
Operations drum synthesis module.

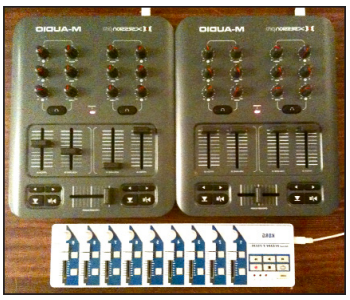
Like a traditional drum machine, each row of the programmed grid corresponds to a single percussion sound. These sounds are synthesized in real time as well; no pre-recorded samples are used. Subtractive, FM and noise synthesis techniques are used to re-create classic electronic drum sounds, with some new variations.

Many of the drums sounds are pitched such that modal harmonies can develop in performance. The performer is able to change the timbral character of each sound, control the overall mix, and apply global effects to the audio stream.

Top to bottom: object-oriented deployment of bpatcher; preparation interface; software performance interface; hardware performance interface.







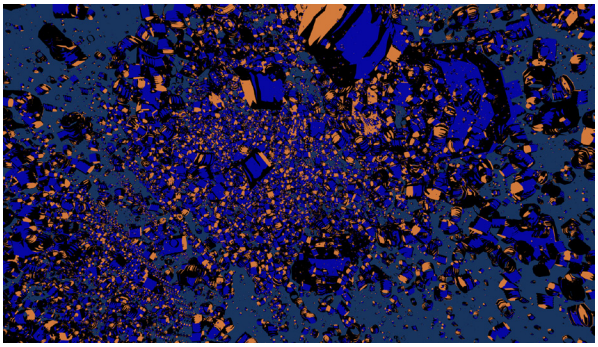
Software

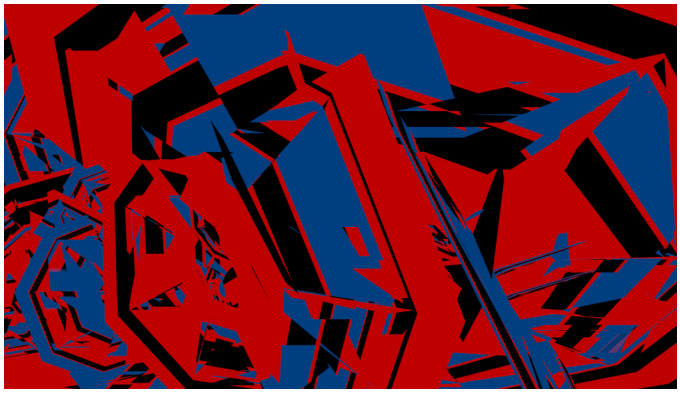
Object-oriented programming techniques were employed in the writing of the software. Max's *bpatcher* object allows for the reuse of interface objects. Individual patches were written once and then deployed multiply, initialized with different parameters, radically increasing the potential for sensory complexity and performance control.

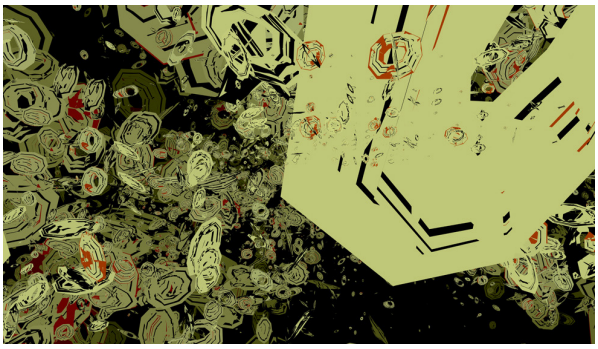
The software has an interface that is used prior to performance to design the audio and visual programs. These programs are saved via database. During performance, however, the performer need not engage with the computer's screen; all performance controls are available via the hardware interfaces. The player is then able to fully immerse themselves in the audiovisual results rather than be involved with the computer, increasing the performative value for themselves and the audience.

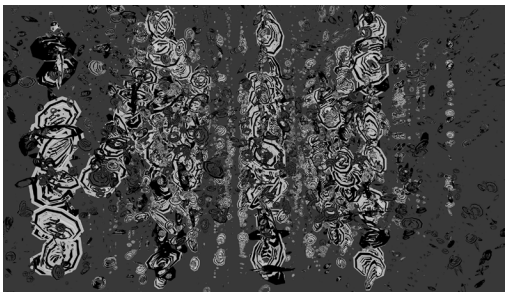
Requirements

Operations may be projected at up to 1080p resolution and with stereo sound. The performance is variable in length and may last as long as 45 minutes, according to the requirements of the program.









Hardware

Laptop, 2 x M-Audio X-Session Pro, Korg nanoKontrol, Presonus Firebox.

Watch

For more information and to watch video excerpts and documentation, visit **foolskool.com/operations**

Contact

T. A. Gambarotto
rocky@foolskool.com
+1 416 703 9536

Representation

Pentimento Gallery, Toronto
pentimento.ca