Tilt: An Aesthetic Interface For An Additive Synthesizer

[Extended Abstract]

Aaron McLeran
Media Arts and Technology Program
University of California
Santa Barbara, CA 93106-9560
amcleran@mat.ucsb.edu

ABSTRACT

This paper presents a software additive synthesizer, Tilt, which uses an aesthetic, abstract interface. The goal of the design was to demonstrate the possibility of creating a synthesizer which provides intuitive and easy parametric control while also giving a positive aesthetic experience. This makes possible a multimodal performance that features both an instruments audio synthesis as well as its visual interface.

Though Tilt's interface is abstract, it intuitively represents the synthesis parameters in a way that is quickly grasped by a performer. Also, because the interface is aesthetically pleasing yet directly represents synthesis parameters, algorithms which automatically modify these parameters can also modify and animate the interface. In this state, the interface takes on the role of an audio visualizer.

The synthesis code was made as a flexible additive synthesizer class using SuperCollider. The graphics were written using Objective-C and Cocoa. The interface and the synthesizer communicate with each other using Open Sound Control.

This paper will discuss details of Tilt's implementation, its aesthetic goals, and discuss the results of a live, interactive performance installation at Santa Barbara's Museum of Art.

Categories and Subject Descriptors

H.5.5 [**Information Interfaces and Presentation**]: Sound and Music Computing—*Methodologies and techniques, Signal analysis, synthesis, and processing* ; D.2.6 [**Software Engineering**]: Programming Environments —*Graphical environments, Interactive environments*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Graphical user interfaces (GUI), Interactive environments*

General Terms

EXPERIMENTATION, DESIGN

Keywords

Electronic Music, Software,Multimodal interaction

1. INTRODUCTION

The paradigm for synthesis interfaces, both in hardware and software, have typically preferred various interface controls such as knobs, sliders, and buttons. This limitation is understandable in hardware systems where there are significant physical limitations. However, in software, the situation is different. Because most people who use software synthesizers have used or have learned from using hardware synthesizers, it is natural for software interfaces to mimic hardware interfaces. But because graphical software is capable of drawing any shape, animate any sequence, and in any number of spatial dimensions, it makes little sense to restrict software interfaces to knobs, sliders, and buttons.

Tilt's interface was created with a goal of creating an interface which is more than knobs, sliders, and buttons. This paper will present a discussion of the parameter space of the additive synthesizer, the interface design, and will discuss the reaction to the synthesizer from an installation at the Santa Barbara Museum of Art.

2. ADDITIVE SYNTHESIS PARAMETER SPACE

Additive synthesis is a synthesis technique based on Fourier Theorem, which states that any periodic function can be represented as a series of sinusoidal terms, each with specific amplitude and phase coefficients [1]. For audio a periodic signal is any signal which has a defined tone, or pitch. Traditionally, in additive synthesis and analysis, sinusoidal frequencies of a given Fourier series are defined according to multiples of a fundamental frequency rather than their individual frequencies. These fundamental multiples are called harmonics. Whole number multiples of a fundamental frequency correspond to pure harmonic timbres (i.e. like a flute) while fractional multiples of a fundamental frequency correspond to inharmonic timbres (i.e. like a bell).

Specifically, for Tilt's additive synthesis, there is an upper bound of 64 possible sinusoidal oscillators. Each oscillator can be added or removed dynamically from the synthesis graph. There are 16 possible harmonics, each of which can be tuned fractionally. Further, each oscillator's amplitude can be controlled as well as its 2-channel spatial location.

3. THE INTERFACE

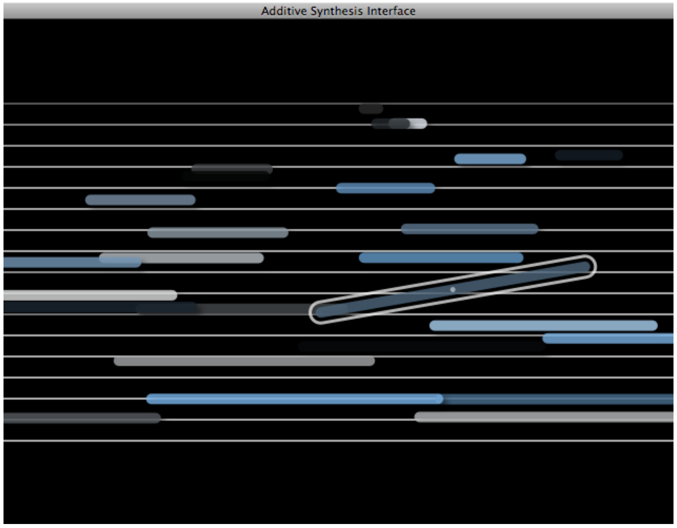

Figure 1: Interface shows an inharmonic timbral configuration with 15 sinusoids.

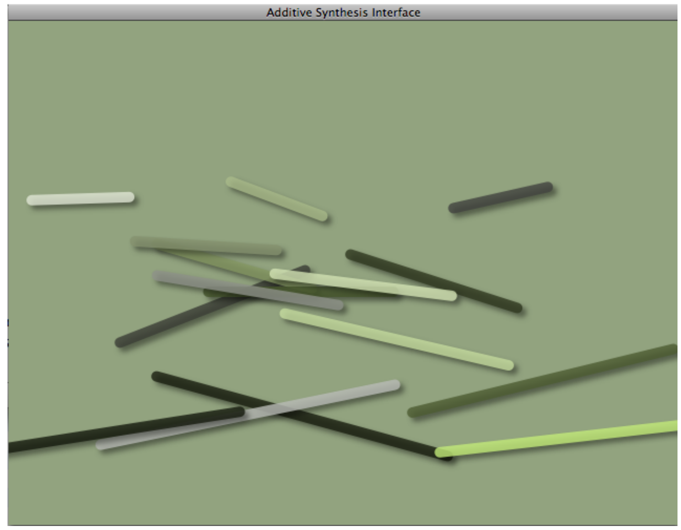

Figure 2: Interface shows an inharmonic timbral configuration with 15 sinusoids.

Fig. (1) shows screenshot of Tilt. Tilt represents additive synthesis parameters by the shape, tilt, and location of the colored rounded rectangles where each rounded rectangle corresponds to a separate oscillator. Each rectangle has the same width but has variable lengths and are centered anywhere between a rectangularly bounded region, which is evenly divided into 16 horizontal strips. The length of each rounded rectangle corresponds to the amplitude of the oscillator it represents. So for a short length, its oscillator is relatively quiet, while for a long length, its oscillator is relatively loud.

The 16 strips correspond to whole number multiples, or harmonics, of the fundamental frequency. The fundamental frequency is defined elsewhere by a midi keyboard input. When the center of a rounded rectangle enters a particular horizontal strip, say for example, the third strip from the bottom, the oscillator the rectangle represents is given a frequency which is the fundamental frequency times the whole number, in this case 3, represented by that region.

The tilt of each rounded rectangle corresponds to a fraction which is either added or subtracted (determined by the slope) to the whole number given by one of the horizontal regions. The tilt is limited within a range of +/- 45 degrees and is mapped to a fractional number of +/- 0.5. So for example, if the center of a rounded rectangle is in the 8th region and is tilted at 45 degrees, the frequency of oscillator will be $8.5 * f$, where f is the fundamental frequency.

Finally, the location of the center of rectangle in the horizontal direction corresponds to the position of the oscillator in a stereo pan. So, for example, if the rectangle is to the far left of the bounded region, the oscillator will be panned to the left.

Fig. (1) shows an additive synthesis state where 24 oscillators, with one of them shown as selected, are randomly distributed across the bounded region. All of the oscillators are harmonic (i.e. no tilt) except for the selected oscillator in the 8th horizontal region with a positive slope tilt. A user interacts with the synthesizer by first selecting a rectangle with the mouse. Then the user can choose to move the oscillator around within the bounded region, change it's tilt, or change its length. This is done in real-time and the effect of the interaction is heard immediately in the sound.

Fig. (2) shows another screenshot of Tilt with the rectangles randomly distributed in space as well as a randomly determined tilt. Fig. (2) also shows the interface with the region division indicators removed.

4. ALGORITHMIC TRANSFORMATIONS

Tilt also has a set of predefined transformation algorithms that animate a given set of parameters. For example, one algorithm applies a stochastically determined low-freqeuncy oscillator (LFOs) to the horizontal position of each of the rectangles. This produces a spatially shimmering effect and is mesmerizing to watch in the interface. Another algorithm applies LFOs to the tilt of each of the rectangles.

There are also a set of commands which produce randomizations of all the synthesis parameters. One command will produce a harmonic randomization, where a random number of rectangles will be distributed across the bounded region, each with zero tilt. Another randomization command will produce inharmonic randomization, distributing the rectangles with variable tilt.

Finally, there is an algorithm which periodically, and rhythmically randomizes the location and optionally, the tilt, of each of the rectangles.

5. REFERENCES

[1] C. Roads. *Computer Music Tutorial*. The MIT Press, 1996.