The WaveSaw: A Flexible Instrument for Direct Timbral Manipulation

Luke Dahl
CCRMA, Department of Music
Stanford University
Stanford, California, USA
lukedahl@ccrma.stanford.edu

Nathan Whetsell
CCRMA, Department of Music
Stanford University
Stanford, California, USA
whetsell@ccrma.stanford.edu

John Van Stoecker
CCRMA, Department of Music
Stanford University
Stanford, California, USA
stoecker@stanford.edu

ABSTRACT
In this paper, we describe a musical controller – the WaveSaw – for directly manipulating a wavetable. The WaveSaw consists of a long, flexible metal strip with handles on either end, somewhat analogous to a saw. The user plays the WaveSaw by holding the handles and bending the metal strip. We use sensors to measure the strip's curvature and reconstruct its shape as a wavetable stored in a computer. This provides a direct gestural mapping from the shape of the WaveSaw to the timbral characteristics of the computer-generated sound. Additional sensors provide control of pitch, amplitude, and other musical parameters.

Keywords
Musical controller, Puredata, scanned synthesis, flex sensors.

1. INTRODUCTION
Most commercial electronic instruments limit the control of sound to one-dimensional controls, such as knobs or faders, whose settings are mapped through various levels of abstraction to create a resulting waveform or timbre. The WaveSaw is inspired by a desire to control sound in a direct and physical way. We want to touch a sound, to manipulate it with our hands as if it is a physical object. The WaveSaw is an instrument whose physical shape is mapped directly to the shape of a waveform or spectrum, and by changing the shape of the instrument we change the sound.

The WaveSaw is made of a long, flat, saw-like strip of flexible metal with wooden handles on each end with which the user can bend, twist, and rotate the instrument. The shape of the blade is measured by flex sensors [1] along its length. The flex sensor values are sent via a microcontroller to a computer, where a custom Puredata (Pd) [2] object recreates the lengthwise shape of the saw blade as a table. This table is then used as the basis of either scanned synthesis [3] or spectral filtering. In the case of scanned synthesis, the table is used as a wavetable that is scanned at audio rates to generate a pitched tone whose waveform, and hence spectrum, varies with the shape of the saw blade. Similarly, the table can be used as the spectral shape of a multi-band filter through which any signal can be passed.

Additionally, the WaveSaw has flex sensors oriented width-wise on the saw that are used to measure the amount of twist applied to the blade, an accelerometer for sensing orientation in space, and a pressure sensitive resistor on one handle to measure how hard the handle is squeezed.

2. DESIGN AND CONSTRUCTION
The WaveSaw blade is made of a 22'' by 4.5'' piece of thin aluminum, which is bolted into slots in the wooden handles. Experiments with early prototypes of the saw revealed that by manipulating the handles it is possible to induce a maximum of three lengthwise bends in the saw blade (a shape similar to one and a half periods of a sinusoid).

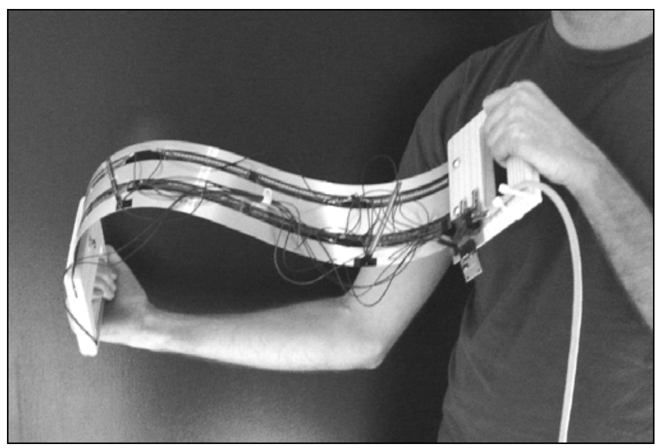

Figure 1. The WaveSaw

We use four flex sensors in order to capture up to three bends and possible "phase variations" thereof. However, each flex sensor can sense bend in only one direction, so we use four pairs of opposite-facing flex sensors in parallel along the length of the blade. Each flex sensor is a 4.5" inch long device whose electrical resistance varies proportionally to the amount of bend. For each pair, one sensor's resistance increases as its partner's decreases. We can therefore arrange each pair electrically as a voltage divider, allowing us to measure only one voltage for each pair.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Nime'05, May 26-28, , 2005, Vancouver, BC, Canada.
Copyright remains with the author(s).

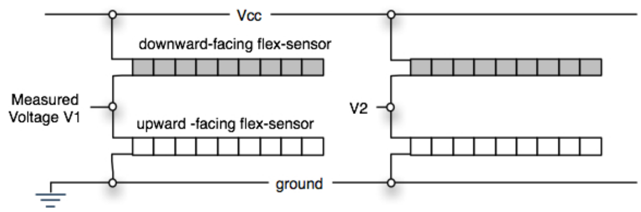

Figure 2. Complementary Flex Sensors as Voltage Divider

An Atmel AVRMini V4.0 AVR Development Board [4] is used to send control voltages from each sensor on the WaveSaw to a computer. We use functions in the AVRlib C-language library [5] for A/D conversion and extensions of AVRlib [6] to transmit the digitized sensor values as Open Sound Control (OSC) messages [7] over the AVRMini's Universal Serial Bus (USB) port.

3. RECREATING THE WAVESHAPE

Once the sensor values have been digitized and sent to a computer as OSC messages, we use Pd to process these values and generate sound. We employ a custom Pd object that reconstructs the shape of the WaveSaw from the digitized sensor values, converts this shape into a wavetable of arbitrary length, and ensures that the audio produced by this wavetable is free from artifacts and glitches.

To reconstruct the shape of the WaveSaw, we assume that the sensors measure the curvature of the metal strip at infinitesimally small points and that the overall shape of the WaveSaw is sinusoidal. Without loss of generality, if we suppose that the vertical displacement (with respect to the handles) of a point on the WaveSaw is $y(t) = \sin t$, then the curvature of the WaveSaw at this point is

$$\frac{d^2}{dt^2} y(t) = \frac{d^2}{dt^2} \sin t = -\sin t = -y(t) \; .$$

Thus, at least in theory, we should be able to negate the sensor values to obtain the vertical displacement at a particular point on the saw blade. This simple method gives surprisingly good results in practice.

At this stage of the process we have obtained the vertical displacement at each point on the saw where a sensor is located. In other words, we have as many values in our wavetable as we have sensors on the saw. We use a method described in [8] to interpolate segments of user-controlled curvature between these values to calculate the final wavetable. To ensure glitch-free production of sound, we update the wavetable with new sensor data only when the complete table has been output as audio.

4. SYNTHESIS AND MAPPINGS

As with any digital musical device, hardware controls do not become an instrument until they are joined by some mapping process to the sound production software. Preferably, this mapping should suggest to the user a conceptual framework that allows them to structure their interaction with the instrument in a reliable and at least somewhat predictable way. The WaveSaw is based on the simple concept of mapping the physical shape of the device to some "shape" in the sound.

The simplest and most intuitive mapping used in the WaveSaw is to let the analogous wavetable in Pd be used as a single cycle of a waveform that is scanned at audio rates. The scanning rate, and hence the pitch, is controlled by the orientation of the instrument with respect to gravity, as measured by the accelerometer.

To this basic mapping we add various extensions. We allow the performer to continuously vary the curvature of each segment in the wavetable from concave to convex by squeezing the sensor on the handle.

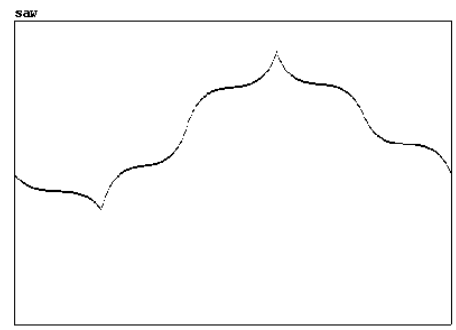

Figure 3. Wavetable with Curvature in Pd

As a further extension of scanned synthesis, we add to the end of the reconstructed waveshape multiple versions of itself, each reflected about the horizontal axis of the wavetable, creating a sort of "compound wavetable." Mapping either twist or squeeze to the number of reflections allows for drastic control of timbre.

By default, the curve reconstruction algorithm assumes that the endpoints of the waveshape are at zero amplitude. If we remove this constraint the waveshape can become discontinuous, creating sounds with higher harmonics.

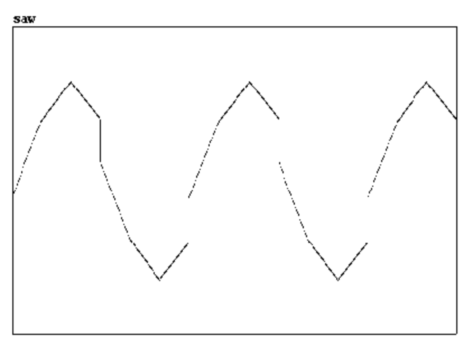

Figure 4. Discontinuities and Reflections in Pd

Additionally, rather than use the waveshape directly as a wavetable, we may map the waveshape to amplitudes of the frequency bins of a spectral filter. Using this mapping in conjunction with the compound wavetable allows for even more extreme spectral manipulation.

5. CONCLUSIONS AND FUTURE WORK

While the instrument lends itself well to timbral gestures, it is difficult to control pitch reliably and independently of other parameters. Since, for our mappings, the shape of the saw controls the volume level of the output signal, the instrument has a wide dynamic range. Compression or other dynamic processing might help retain expressivity while not endangering the audience's hearing.

The electrical connections to the flex sensors on the saw blade undergo mechanical stress during the use of the instrument and often fail. We need to explore new construction techniques to increase the robustness of the instrument.

In conclusion, the WaveSaw is a new instrument based on the idea of mapping the shape of the instrument to the “shape” of a sound. We have used the shape of the saw blade to control the shape of a wavetable for scanned synthesis or the shape of a filter response for spectral filtering. Further variations on this theme might include mapping the saw shape to the shape of time envelopes, or using the shape of the saw to generate MIDI sequences of analogous pitch contour.

A video of the WaveSaw in use can be seen at `http://ccrma.stanford.edu/~lukedahl/250/WaveSaw.mov`.

6. ACKNOWLEDGMENTS

We are grateful for the input and insight of Bill Verplank, Michael Gurevich, and Juan Pablo Cáceres.

7. REFERENCES

[1] Images SI, Inc. Flex Sensor. `http://www.imagesco.com/sensors/flex-sensor.html`.

[2] Puckette, M. Puredata (Pd). University of California San Diego, La Jolla, CA. `http://www-crca.ucsd.edu/~msp/software.html`.

[3] Matthews, M., Shaw, R., and Verplank, B. Scanned synthesis. In *Proceedings of the 2000 International Computer Music Conference (ICMC 2000)*. Berlin, Germany, Aug. 27-Sep. 1, 2000.

[4] Atmel Corporation. AVRMini V4.0 AVR Development Board. `http://www.atmel.com/products/AVR/overview.asp`.

[5] Stang, P. *Procyon AVRlib*. Santa Clara University, Santa Clara, CA. `http://hubbard.engr.scu.edu/embedded/avr/avrlib/index.html`.

[6] Gurevich, M., Ju, W., Verplank, B. et al. Open Sound Control (OSC) extensions for AVRlib.

[7] Open Sound Control (OSC). `http://opensoundcontrol.org/`.

[8] ffitch, J. et al. GEN16. `http://www.csounds.com/manual/html/GEN16.html`