AIRWAVE: A Gesture-Based Granular Synthesis Controller

Mick Marchan
mickkm@cs.washington.edu

Michael Notter
blueshoe@cs.washington.edu

ABSTRACT

Granular synthesis is a powerful technique for the manipulation and synthesis of sound. However, due to its conceptually complex nature, the technique goes largely unnoticed and unused by most musicians and performers. AIRWAVE provides an easy point of entry for artists of all backgrounds and skill levels to control the rich parameter space of granular synthesis through simple, real-time hand gestures in mid air. In this way, it represents a novel application of this potent type of synthesis which makes it widely accessible to new audiences - from the hobbyist who wishes simply to learn the basics of granular synthesis and explore its potential to the seasoned musician looking for expressive new creative tools.

1. INTRODUCTION

At a high level, granular synthesis can be described as the process of chopping audio into small, enveloped "grains" (typically 10-50ms segments) and recombining them to create one unified cloud of sound. These grains are then fired out either synchronously at a chosen rates (grain rate) or asynchronously at a chosen density (grain density) with the audio inside each grain played back at varying speeds (wave ratio). By manipulating these parameters and by using both synchronous and asynchronous grain triggers, an impressive variety of sounds and textures can be achieved.

Given its wide spectrum of uses - which range from technical applications like resynthesis, time-stretching and pitch-shifting to more aesthetic applications such as artistic composition, polyrhythms, noise and live audio manipulation in performance - one would imagine that granular synthesis would be a familiar concept to any seasoned musician. However, the abstract relationships between its parameters present strange conceptual challenges to anyone trying to master them. Furthermore, the use of traditional tools such as digital plugins and physical interfaces often require that parameter settings are fixed (at best, the user might be able to manipulate one or two parameters at a time), which limits the degree to which an intimate relationship can be formed with granular synthesis as an instrument or performance tool.

AIRWAVE tackles this problem head on by allowing for real-time control over all parameters simultaneously through the use of a Leap Motion controller - which reads 3D positional information from up to two hands in space. Not only does this result in a highly expressive performance tool, but it also invites the user to explore the nuanced relationships between the parameters of granular synthesis (along with a strange and inspiring new soundscape) with nothing more than the wave of a hand.

2. RELATED WORK

Since its inception in the late 1950s, granular synthesis has been put to use in a variety of fields. Notably, the interesting sound textures and gestures that can be generated from granular synthesis have been explored heavily in the realm of Western Art Music. Influential pieces such as Barry Truax's *Riverrun* (1986) [1] and *Wing's of Nike* (1987) [2] are entirely constructed through this technique, and have acted as an aesthetic reference to AIRWAVE. Beyond the art world, granular synthesis has become a staple technique for time signal processing, which aims to manipulate the duration of a sound independently of its pitch or vice versa [3]. Examples of this specific use can be found anywhere from modern DJ equipment for beat matching to speeding up radio or TV advertisement disclaimers to fit within strictly defined time segments.

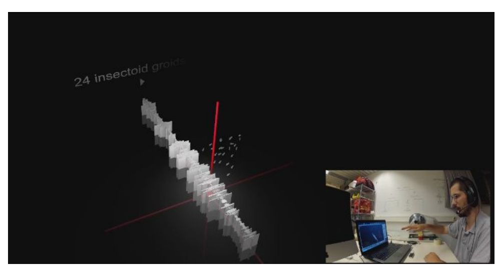

Figure 1: A screenshot of GROIDs, an existing granular synthesis controller

Due to the general frustration and fascination people have towards granular synthesis, there are several existing applications that are similar to AIRWAVE. However, none of them give users total control over granular synthesis parameters. GROIDS, an existing granular synthesis controller that also uses gesture based controls, sought to address an identical problem to AIRWAVE, attacking the complexity of granular synthesis with an easy-to-use controller. Instead of directly manipulating values such as grain rate, grain size, or index, GROIDS made use of algorithmic flocking, where the user's hands would act as the targets towards which swarms of particles would race, with the particles being individual moving grains in the sound. The position of a particle determined what part of the sound was enveloped in its grain. This allowed for interesting visuals and textures, but we did not feel that it was the best approach for both introducing people to granular synthesis and also allowing for deep exploration into the full parameter space [4].

3. SOLUTION

AIRWAVE is composed of two main components: a Python script that processes and translates information obtained from the Leap Motion API and a set of SuperCollider scripts which handle synthesis and visualizing the state of the synthesis parameters. These two components communicate through Open Sound Control (OSC), a protocol designed for real-time control of sound and media over a network. This allows AIRWAVE to be extremely modular, which is very advantageous for experimenting with different control systems (e.g. replacing the Leap Motion with a device that tracks the orientation of the body) or different synthesis techniques (e.g. moving to FM synthesis instead of granular).

The user launches AIRWAVE by running both the Python tracking script and the main SuperCollider script, which will launch the application and open the visualizer. From here, the user is able to explore the parameter space of granular synthesis with a predefined audio file by moving their hands over the Leap Motion controller. In general, the gesture controls are the only way the user interacts with AIRWAVE other than looking at the visualizer for spatial reference.

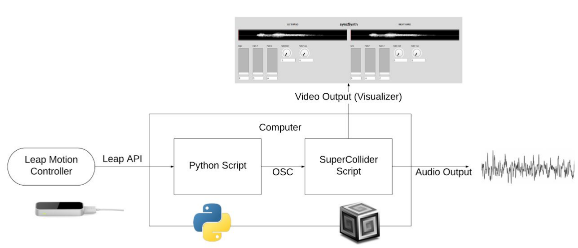

Figure 2. A diagram of the components and data-flow of AIRWAVE

3.1 LEAP API & PYTHON SCRIPT

Within AIRWAVE, our Python script acts as the controller, bridging information from the Leap Motion to SuperCollider. The first challenging question we encountered when writing this script was this: Among the dozens of metrics on hand position, shape and velocity provided by the Leap, which should be captured? Which hand movements would be intuitive for a user to learn and use? Which of them would be most meaningful in the context of granular synthesis and the kind of audio manipulation we had in mind? After much testing and experimentation, we decided to limit our input to x, y and z position of the hand, palm roll, yaw, and grip strength (the degree to which the fist is open or closed). Later we also made use of the velocity feature when writing our own custom gestures (see "Controls and Gestures" below).

With these features in hand (no pun intended), we then also chose to preprocess the raw data in Python before sending it along to SuperCollider. Additionally, we wrote in logic for recognizing our custom gestures here - this way SuperCollider can simply interpret gestural triggers rather than needing to recognize the gestures themselves.

3.2 SYNTHESIS & VISUALIZATION

On the other end of AIRWAVE is the synthesis and visualization component built with SuperCollider. This consists of a driver script that loads and executes function scripts for defining commonly used functions (e.g. the calculating 2-channel stereo positioning using the sine-cosine panning law), synthesis scripts which define how synths are built and how they should react to hand gestures, and a visualization script that loads the visualizer for the performer.

The driver script acts as the core of AIRWAVE, connecting every other component of the application. Most importantly, it defines what audio file the user will perform granular synthesis over and handles synthesis switching, where the user can move between different synths through our custom gestures.

The synthesis scripts focus on loading synthesis and OSC definitions. This is where AIRWAVE defines what hand motions map to what synthesis parameters and how those parameters should ultimately impact the audio output. Due to the subjective nature of performance and sound art, we thought it best to make this portion of AIRWAVE easily customizable to those familiar with SuperCollider. To do this, we emphasized modularity with the synthesis definitions. For example, all a user needs to do to add a new synth to AIRWAVE is write up (or edit) a SuperCollider script with the proper components (a synth definition and an OSC definition) that is named "*_synth.scd," add its name to the driver script, and it will automatically be playable the next time the application is launched.

The visualizer script builds a simple GUI for the user to monitor the state of the various parameters they are manipulating. Due to our interface being invisible and providing no haptic feedback, it is very important that the user is given a sense of the boundaries of the gesture tracker. To do this, we provide simple representations of the hand-data:

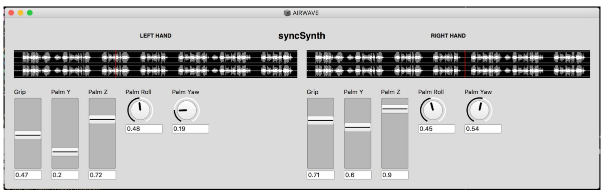

Figure 3. A screenshot of the AIRWAVE visualizer.

3.3 CONTROLS & GESTURES

While AIRWAVE makes use of both synchronous granular synthesis (loosely, the more “rhythmic” and tonally “clean” variety) and asynchronous granular synthesis (loosely, the more “randomized” and tonally “noisey” variety), the mapping between the user’s hand movements and specific synthesis parameters are identical for both. Our mapping is as follows:

Palm x-coordinate	→	Grain Index
Palm y-coordinate	→	Grain Rate / Density
Palm z-coordinate	→	Gain
Palm Roll	→	Wave Ratio
Palm Yaw	→	Panning
Grip Strength	→	Grain Size

Here we will give a brief description of the aural effect each parameter has on the final audio output:

1. Grain Index - this describes the position in the audio file from which a new grain will be “extracted”.
2. Grain Rate - this describes how quickly grains will be “fired”. In other words, a high grain rate results in a very dense, noisy texture while a low rate results in the pulsing of distinct “pieces” of sound.
3. Gain - this corresponds to the volume of the overall output.
4. Wave Ratio - this is determines how fast the audio within each grain will be played and outwardly corresponds to the pitch of the resulting sound. High wave ratios raise the pitch of the audio (e.g. Mickey Mouse) while low wave ratios lower the pitch (e.g. Darth Vader).
5. Panning - this controls speaker-wise panning in the traditional sense.
6. Grain Size - the duration, or size, of a grain. Long grains are big enough for the listener to pick out and recognize specific characteristics of the enveloped audio, while smaller grains begin to sound like miniscule “blips” of sound.

We mentioned above that AIRWAVE makes use of both synchronous and asynchronous granular synthesis - and that these two approaches yield different sonic results. For our final prototype, we wanted the user to be able to switch seamlessly between these two styles on the fly. To do this, we hoped to use the Leap Motion’s built-in recognized gestures such as tapping and scrolling with a single finger. Unfortunately, upon trying to access these features, we found that they had been deprecated in the latest SDK! So we decided to build our own. Making use of the aforementioned palm velocity metric offered by the Leap, we were able to write a concise code block to recognize closed-fist swiping and tapping gestures for the left hand. We then used these gestures to allow the user to select between synthesis styles and confirm selections in real-time. While these simple gestures may not seem especially complex, writing the logic to create them from scratch with hundreds of frames flying by each second was quite a challenge!

4. CONCLUSION

Overall, we feel AIRWAVE has been a successful exploration into this exciting new space of gesture-controlled granular synthesis. Our final prototype is quite stable, its performance is smooth and relatively glitch-free and our custom gestures work well. Most importantly, AIRWAVE does a fantastic job of allowing the performer to experiment with and more fully understand the relationships between these abstract parameters by simply moving their hands in mid air. This alone represents a major advance toward bringing granular synthesis to the masses.

5. ACKNOWLEDGEMENTS

We would like to thank Joseph Andersen and Daniel Peterson for providing our initial introduction to the concept of granular synthesis in the University of Washington DXARTS 46x series; and also to Daniel specifically for his early-stage advice on how best to approach implementing AIRWAVE in SuperCollider. We would also like to thank Bruce Hemingway and Ying-Chao (Tony) Tung for their guidance and encouragement (not to mention for the freedom to get weird) during our work in the Sound & Media Capstone.

REFERENCES

[1] Truax, Barry. *Riverrun*. https://www.sfu.ca/~truax/river.html. 2007.

[2] Truax, Barry. *Wing's of Nike*. https://www.sfu.ca/~truax/nike.html. 2003.

[3] Dodge, Charles, and Jerse, Thomas. "Granular Synthesis." *Computer Music: Synthesis, Composition, and Performance*, 2nd ed., Schirmer, 1997, pp. 262–276.

[4] Woschitz, Gregor. *GROIDS - live granular synthesis controlled with leap motion.* http://www.gorgis.org/groids-live-granular-synthesis-controlled-with-leap-motion/. 2014.