AN ADVENTURE OF GROWING MOTIVES INTO INTERACTIVE PERFORMANCES: A DIGITAL PORTFOLIO DISSERTATION OF SEVEN ELECTROACOUSTIC COMPOSITIONS FOR DATA-DRIVEN INSTRUMENTS

by

FANG WAN

A DISSERTATION

Presented to the School of Music and Dance
and the Graduate School of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Doctor of Musical Arts

June 2020

DISSERTATION APPROVAL PAGE

Student: Fang Wan

Title: An Adventure of Growing Motives into Interactive Performances: A Digital Portfolio Dissertation of Seven Electroacoustic Compositions for Data-driven Instruments

This dissertation has been accepted and approved in partial fulfillment of the requirements for the Doctor of Musical Arts in Data-driven Instruments degree in the School of Music and Dance by:

Jeffrey Stolet	Advisor and Chair
Akiko Hatakeyama	Core Member
Jon P. Bellona	Core Member
Ying Tan	Core Member

and

Kate Mondloch	Interim Vice Provost and Dean of the Graduate School

Original approval signatures are on file with the University of Oregon Graduate School.

Degree awarded June 2020.

DISSERTATION ABSTRACT

Fang Wan

Doctor of Musical Arts

School of Music and Dance

June 2020

Title: An Adventure of Growing Motives into Interactive Performances: A Digital Portfolio Dissertation of Seven Electroacoustic Compositions for Data-driven Instruments

This Digital Portfolio Dissertation focuses on seven performance videos of electroacoustic compositions for data-driven instruments. The text document included in this digital portfolio dissertation explains, analyzes, and discusses the design and implementation of data-driven instruments, the musical challenges and opportunities provided by data-driven instruments, the performance techniques employed, the sound synthesis techniques used, and the data-mapping strategies used for each of the seven compositions. Each of the seven compositions is based on sonic motives. The analysis of the compositions in this dissertation explain how the sonic motives are introduced, modified, and developed throughout the compositions using data-driven instruments.

SUPPLEMENTAL FILES

The supplemental material takes three forms: digital videos of performances of the portfolio compositions, the custom software used to perform the works, and all affiliated files necessary to perform the works.

Video of performance of *Frozen*
Video of performance of *Origin*
Video of performance of *Pulsing*
Video of performance of *Huadan's Whisper*
Video of performance of *Dots Illusion*
Video of performance of *Overlapping Strings*
Video of performance of *Double Shadows*

Custom software and affiliated file required to perform *Frozen*
Custom software and affiliated file required to perform *Origin*
Custom software and affiliated file required to perform *Pulsing*
Custom software and affiliated file required to perform *Huadan's Whisper*
Custom software and affiliated file required to perform *Dots Illusion*
Custom software and affiliated file required to perform *Overlapping Strings*
Custom software and affiliated file required to perform *Double Shadows*

CURRICULUM VITAE

NAME OF AUTHOR: Fang Wan

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene
Xinghai Conservatory of Music, Guangzhou

DEGREES AWARDED:

Doctor of Musical Arts, 2020, University of Oregon
Master of Music, 2016, University of Oregon
Bachelor of Arts, 2011, Xinghai Conservatory of Music

AREAS OF SPECIAL INTEREST:

Data-driven Instrument Performance
Intermedia Music Technology
Sound design
Electroacoustic music

PROFESSIONAL EXPERIENCE:

Graduate Employee, School of Music and Dance, University of Oregon, Eugene, 2018-2020

Teaching Assistant, Computer Music Summer Academy, Music Technology, Eugene, 2015-2020

Sound Technician, 2010 Asian Games Cricket Venue, Guangzhou, 2009-2010

GRANTS, AWARDS AND HONORS:

Graduate Teaching Fellowship, School of Music and Dance, 2018-2020

Best Student Composition, International Computer Music Conference, 2017

Outstanding Graduate Scholar Award, University of Oregon, 2016

Outstanding Student Scholarship, Xinghai Conservatory of Music, 2008-2011

Best Electronic Music Artist, Chinese Golden Melody Awards, 2011

Best Electronic Music Artist, the 11th Chinese Media Music Award, 2010

CONFERENCE PRESENTATIONS:

Society of Electro-Acoustic Music in the United States, 2016, 2017, 2018, 2019, and 2020

SEAMUS Vol. 28 CD, 2018

New York City Electroacoustic Music Festival, 2020

TURN UP Multimedia Festival, 2020

Music-Acoustica Beijing, 2018 and 2019

International Computer Music Conference, 2017 and 2018

Kyma International Sound Symposium, 2016 and 2018

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my committee: Professors Jeffrey Stolet, Akiko Hatakeyama, Jon Bellona, and Ying Tan for their expertise, advising and encouragement throughout my creative research and work on my dissertation. Having no digital art experience, I was grateful to be taught by Professor Tan, who introduced me to visual aesthetics and video compositional techniques and enabled me to challenge myself by composing intermedia compositions with visual elements and accomplishing all the video recording and editing for my dissertation. Professor Bellona took me on a journey of discovering new ways to analyze and apply data in sound design and composition. Professor Hatakeyama presented me with a wide variety of electronic music styles. She inspires and encourages me to confront and express myself in my compositions. Professor Stolet continually and convincingly transformed me deep in my spirit, inspiriting me to explore data-driven instruments in new musical expression, to work hard on the creation of music of my own, and more importantly, to be a nice and confident person who makes the world a better place. My completion of this dissertation could not have been accomplished without the support of his guidance and persistent help.

A special thanks to Chi Wang, my peer, mentor, and friend. Dr. Wang, you provided support and help throughout my entire time at the University of Oregon.

A very special thanks to my parents Bin Wan and Tiantao Li. Because of your unconditional and complete support and care, I could pursue my dream of electronic music at the highest level of education at the University of Oregon.

TABLE OF CONTENTS

Chapter Page

Chapter Page

SUPPLEMENTAL FILES SUPPLEMENTAL FILES

Video of performance of *Frozen*

Video of performance of *Origin*

Video of performance of *Pulsing*

Video of performance of *Huadan's Whisper*

Video of performance of *Dots Illusion*

Video of performance of *Overlapping Strings*

Video of performance of *Double Shadows*

Custom software and affiliated file required to perform *Frozen*

Custom software and affiliated file required to perform *Origin*

Custom software and affiliated file required to perform *Pulsing*

Custom software and affiliated file required to perform *Huadan's Whisper*

Custom software and affiliated file required to perform *Dots Illusion*

Custom software and affiliated file required to perform *Overlapping Strings*

Custom software and affiliated file required to perform *Double Shadows*

LIST OF FIGURES

Figure Page

Figure Page

LIST OF TABLES

Table Page

Chapter I

INTRODUCTION

The text document that follows is associated with the seven digital videos of performances of original real time electroacoustic compositions that employ data-driven instruments. In this digital portfolio dissertation, all seven compositions use Symbolic Sound's Kyma as the sound design environment. In five of the seven compositions I use custom software created in Max for initial data processing and mapping. For this portfolio of seven compositions I use seven different performance interfaces that each required distinct performance techniques. The seven interactive electroacoustic works for data-driven instruments that comprise the dissertation are:

1. *Frozen*, a multichannel real time interactive composition for two Wiimote Controllers, OSCulator, and Symbolic Sound's Kyma;
2. *Origin*, a multichannel real time interactive composition for Wacom Tablet and Symbolic Sound's Kyma;
3. *Pulsing*, a stereo real time interactive composition for Roli Lightpad Block, custom software created with Cycling74's Max, and Symbolic Sound's Kyma;
4. *Huadan's Whisper*, a multichannel real time interactive composition for Leap Motion, custom software created with Cycling74's Max, and Symbolic Sound's Kyma;
5. *Dots Illusion*, a multichannel real time interactive composition for iPad, custom software created with Cycling74's Max/MSP/Jitter, and Symbolic Sound's Kyma;

6. *Overlapping Strings*, a multichannel real time interactive composition for Gametrak, custom software created with Cycling74's Max, and Symbolic Sound's Kyma;
7. *Double Shadows*, a stereo real time interactive composition for two Blue Air Controllers, custom software created with Cycling74's Max, and Symbolic Sound's Kyma.

For each of the compositions I provide an introduction and describe 1) the design and implementation of each of the data-driven instruments, 2) the musical challenges and opportunities provided by the data-driven instrument, 3) the data mapping strategies used, 4) the musical composition and the sound synthesis techniques used, and 5) performance technique employed. Because all of the seven compositions contained in this Digital Portfolio Dissertation revolve around the presentation of musical motives that are developed over time, I will analyze, describe, and explain my concept of motive as it occurs within each of my seven data-driven instrumental compositions.

Chapter II

CONCEPTUAL BACKGROUND AND THEORETICAL FRAMEWORKS

Before I offer specific descriptions about each of my compositions, I would like to provide some fundamental ideas about data-driven instrument compositions. Among these ideas are: 1) the structure of a data-driven instrument, 2) data mapping, 3) mutability, 4) sound synthesis techniques, 5) sonic motive, and 6) performance space and performance techniques.

The Structure of Data-driven Instruments

Traditional acoustic musical instruments, such as the piano or violin, have been used to play music for hundreds of years. When a performer plays an acoustic instrument, he or she projects energy into the physical body of that instrument in order to create sound. Data-driven instruments, on the other hand, operate on a different concept and method of sound generation and performance. In data-driven instruments, sound is not caused by the physical energy projected into its body. Data-driven instruments replace energy's function with data streams that initiate and control sound-producing algorithms resident in software. Data is first generated when the performer operates a controller. After initial data conditioning, the data is sent to a software program to produce sonic outcomes. According to Jeffrey Stolet, the structure of a data-driven instrument can be considered as a three-part mechanism:

1. A section where data is created or acquired through human operation of some interface device,
2. A software layer where each generated value input to it is mapped to an output value, or where each value input to it is analyzed, and, on the basis of that analysis, new replacement data is created, and then routed as control data to

3. A sound synthesis section capable of receiving and responding to this modified data to control musical parameters in real time.[1]

A data-driven instrument is completed only when all three parts are connected. Stolet also explains there is a fundamental difference between acoustic instruments and data-driven instruments. Stolet explains that, "Whereas traditional instruments are driven by energy exerted into their physical systems, new instruments – data-driven instruments – replace energy's function with data streams. And the way we play these instruments is by generating data through performative actions involving interfaces."[2]

Data Mapping

Data mapping is an indispensable part of data-driven instruments. Since performance interfaces output data in different numerical ranges, and the sonic events generated and changed in the sound-producing layer also require data in different ranges, mapping the original data into new numerical ranges is often necessary. The arrangement of the data can be worked out either in the software layer of the data-driven instrument or in the sound-producing layer. I used several basic data mapping methods in my compositions:

1. Data scaling

   When the desired range of data is different from the generated data streams, the range of the data streams needs to be adjusted. Data scaling is one of the methods I used to accomplish this. For example, the range of the data output from a MIDI

---

[1] Jeffrey Stolet, "Twenty-three and a Half Things about Musical Interfaces," (Lecture Notes, Kyma International Sound Symposium Keynote Address, Brussels, Belgium, September 14, 2013).

[2] Stolet, "Twenty-three and a Half Things."

controller is likely to be 0-127. However, the desired data range might be 0.0-1.0. Data scaling methods can be performed to adjust the range of the data from 0-127 to 0-1. Multiplication and division are often used to accomplish this task.

2. Data offsetting

   When the desired data has the same range as the original, but the minimum value is different, the data needs to be offset. For example, a data stream ranging from 0-100 can be offset into range 1000-1100 or 200-300, depending on the desired musical needs. Addition and subtraction are used to accomplish data offsetting.

3. Data analysis and replacement

   Sometimes simple analytic procedures must be performed on data. The intention of such analytics is to determine whether certain conditions have been met in order to trigger some musical event. Often only one condition needs to be met, but it is not that unusual to require multiple conditions. An example of a single condition being applied is contained in a question like, “Are the values greater than 0.9?” An example of multiple conditions being met is contained in a question like “Are the values greater than 0.9? and also did this eclipsing of 0.9 occur in the first twenty seconds of the sound?”

4. Data smoothing

   In order to reduce differences between contiguous values, data smoothing methods must be applied.[3] Typically such smoothing is accomplished by averaging adjacent values or interpolating new intermediate values between existing values.

---

[3] Chi Wang, “Narrative to Action in the Creation and Performance of Music with Data-driven Instruments,” (DMA diss., University of Oregon, 2018), 14.

5. Data filtering

   Sometimes data is generated at a rate faster than needed. Data filtering helps with controlling the amount of data passing through the system. This scheme can also be used to control the rhythm of musical events.

Mutability

One of the unique characteristics that data-driven instruments have is the ability to be mutable during a performance – this is called *instrumental mutability*.[4] Acoustic instruments usually have fixed mapping mechanisms, which is not optimized for change in real time during performance. For example, the piano keyboard has an established way of associating its keys to strings in its sound-producing mechanism. The lower notes are associated with keys on the left side of the piano keyboard, and the higher pitches are associated with keys on the right. However, with data-driven instruments, changing such a set of associations during the performance is completely manageable and can be achieved many times during the course of a composition. Typically, such changes occur in the software portions of the instrument where the data mapping is executed or in the part of the data-driven instrument that holds the sound-producing algorithm.

Sound Synthesis Techniques

The sound-producing mechanism in all my seven compositions resides in Symbolic Sound's Kyma. Symbolic Sound's Kyma is a powerful sound synthesis

---

[4] Stolet, "Twenty-three and a Half Things."

environment for sound design and music composition.[5] Among the sound synthesis techniques that I employ with Kyma are:

1. Sampling synthesis

   In Symbolic Sound's Kyma, there are many Sound objects that use sampling synthesis. The two Sound objects I used the most in my sound design are the *Sample* and *SampleCloud* Sound objects.[6]

2. Subtractive synthesis

   Subtractive synthesis is a synthesis technique that is based on the removal or attenuation of specified frequencies by a filter. In my compositions, the *HighPassFilter* and *LowPassFilter* Sound objects are used most often depending on the musical context.

3. Additive synthesis

   Additive synthesis is a technique based on the combining of simple waveforms (usually sine waves) at various frequencies, amplitudes, and phases to create a more complex waveform. In Symbolic Sound's Kyma, *Oscillators* and *Mixers* can be used for combining various simple waveforms at various frequencies, amplitudes, and phases to achieve sound design in an additive approach.

4. Analysis and resynthesis

   Analysis and resynthesis is a fairly widespread type of sound creation process. According to Stolet, "Analysis and resynthesis is a two-part process that first

---

[5] Carla Scaletti, "Computer Music Languages, Kyma, and the Future," *Computer Music Journal* 26, no. 4 (2002): 69-82, https://doi.org/10.1162/014892602320991392.

[6] As a convention in this dissertation document, I will italicize the names of all Kyma Sound objects and Max objects.

involves the execution of an analytic procedure where spectral content of a sound is derived. This data is then used as the basis to drive and control other synthesis methodologies."[7] The *SumOfSines* Sound object is one object that I use to execute this technique. Among the special outcomes that analysis and resynthesis can achieve is the transposition of the sound's frequency without changing its duration. Another analysis and resynthesis technique offered by Kyma is *TAU*, the Time Alignment Utility, which "is a proprietary analysis and resynthesis procedure where temporally coordinated amplitude, frequency, format, and bandwidth parameters can be exploited to produce stunning and unusual sonic results,"[8] according to Wang.

Sonic Motive

A motive is a small identifiable musical unit usually made up of a series of pitches that has a distinct contour and rhythmic pattern.[9] During the 20th century, when the techniques of musique concrète moved into compositional use, sonic attributes such as timbre, texture, or musical dynamics became available to be used in manners similar to the traditional concept of the motive.[10] Because every sound possesses frequency components at specific amplitudes, and is articulated a specific duration and timbre, all of

---

[7] Jeffrey Stolet, *Kyma and the SumOfSines Disco Club* (Morrisville: Lulu Press, 2012), 150**.**

[8] Wang, "Narrative to Action," 25.

[9] William Drabkin, "Motif," Grove Music Online, accessed April 28, 2020. https://www.oxfordmusiconline.com/grovemusic/view/10.1093/gmo/9781561592630.001.0001/omo-9781561592630-e-0000019221.

[10] Pierre Schaeffer, *Treatise on Musical Objects: An Essay Across Disciplines* (California: University of California Press, 2017), 237.

the attributes of sound may be used in motivic manners. These ideas are first articulated in the writings of Pierre Schaeffer and later by such authors as Leigh Landy in his book *Making Music with Sounds*.[11] For example, a short recording of a pin dropping sound can be used as a motive in an electroacoustic composition. The electroacoustic motive, rather than being predicated on pitch and rhythmic elements, may be predicated on its amplitude envelope. For instance, a sound containing three pins dropping has sharp amplitude attacks and quick decay envelopes that are easily recognizable and that can be musically developed much in the same way traditional motives are developed. The sound may be presented at the beginning of the piece in its original form. As the music unfolds, the motive can be transformed into different variants that employ differences in durations and directions (forwards or backwards), transpositions, spatial locations, or sound synthesis techniques. The way of performing the motive and its transformations may also be different depending on the context of the presentation of the original motive. The identification, development, compositional approaches, and performance techniques developed out of the motives become the essential compositional focus of the seven data-driven instrumental works in my dissertation.

Performance Space and Performance Techniques

The performance interfaces of each data-driven instrument influence the way a musician may compose, shape, and perform a piece. For example, the Roli's Lightpad Block button-based interface is quite tiny and creates a small region in which discontinuous performative actions are encouraged. On the other hand, the Gametrak

---

[11] Leigh Landy, *Making Music with Sounds* (New York: Routledge, 2012), 138-142.

offers a completely a different performance space. The Gametrak's performance space is created by two retractable cables that can be withdrawn and extended to up to ten feet in both horizontal and vertical dimensions, thus producing a large performance area. The fact that the cables are rather "fader-like" and invite the causation of continuous data streams the performance space is radically different.[12] In contrast, the Leap Motion offers a much smaller performance space,[13] the Wacom tablet offers a virtually flat performance space (almost no vertical dimension),[14] and the Blue Air offers only a single vertical beam.[15] These differences profoundly affect how each piece was composed and performed.

---

12. Stolet, "Twenty-three and a Half Things."

13. Alex Colgan, "How Does the Leap Motion Controller Work," Leap Motion, August 9, 2014, http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/.

14. Michael Zbyszynski et al., "Ten Years of Tablet Musical Interfaces at CNMAT," In *Proceedings of the 7th International Conference on New Interfaces for Musical Expression*, 100–105. New York, USA, 2007.

15. Jeffrey Stolet and Andrew Lane, *Blue Air Manual and Documentation* (University of Oregon, 2004), 4.

CHAPTER III

PORTFOLIO COMPOSITIONS

CHAPTER III.1

FROZEN

Overview

Frozen is a multichannel interactive composition for two Nintendo® Wiimote controllers, OSCulator, and Kyma. The piece was based on audio recordings collected during the time of an ice storm in Eugene, Oregon in December 2016. The sound materials of *Frozen* included the sounds of hitting iced tree branches with chopsticks, the shaking of an iced tree, and scraping ice. In some respects, this composition was a Wiimote duet. The piece focused on the antiphonal interplay between the two controllers that musically and visually interact with each other creating and controlling music in real time.

Design and Implementation of the Data-driven Instrument used in *Frozen*

The Wiimote is a game controller produced by Nintendo®. Unlike some of the controllers, such as the Gametrak and Leap Motion that primarily output continuous controls, or other devices such as Monome that only possess button mechanisms, the Wiimote has four continuous controllers, including *Pitch*, *Roll*, *Yaw*, and *Accelerometer*,[16] and eleven buttons that make the Wiimote controller powerful and versatile when simultaneously controlling many musical parameters.[17] An image of the

---

[16] To differentiate from the musical parameter pitch, I will italicize and capitalize *Pitch* data as well as other data streams *Roll*, *Yaw*, and *Accelerometer* coming from the Wiimote controller.

[17] During the composition of *Frozen*, I did not employ Yaw control in any capacity.

Wiimote Controller is shown in Figure 1.

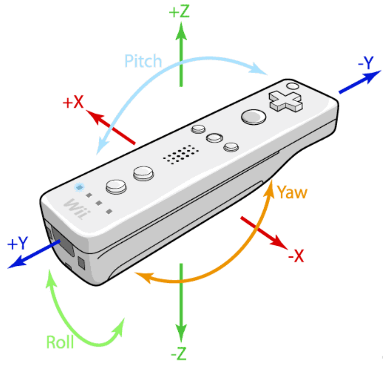


Figure 1. *Pitch*, *Roll*, and *Yaw* axes from Wiimote (image credit: *OSCulator*).[18]

The data-driven instrument used in *Frozen* was comprised of three parts: 1) two Wiimote controllers, which output data when I operated them, 2) OSCulator, which translated the initial data output from Wiimote controllers to data that could be understood by Kyma, and 3) Kyma, which received the transformed data, performed additional data mapping, and responded to it by producing sonic results. This overall structure is depicted below in Figure 2.

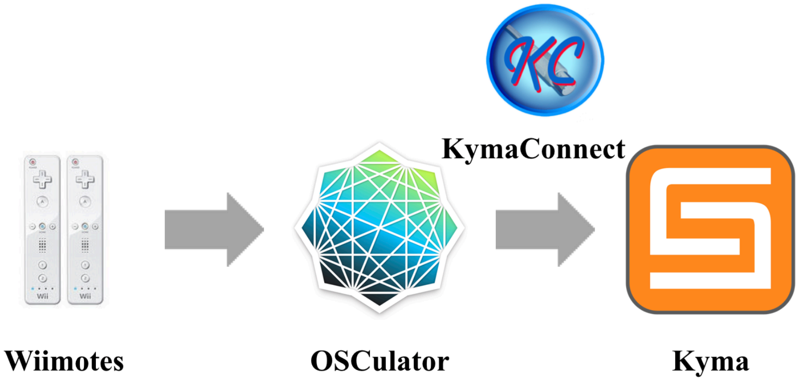
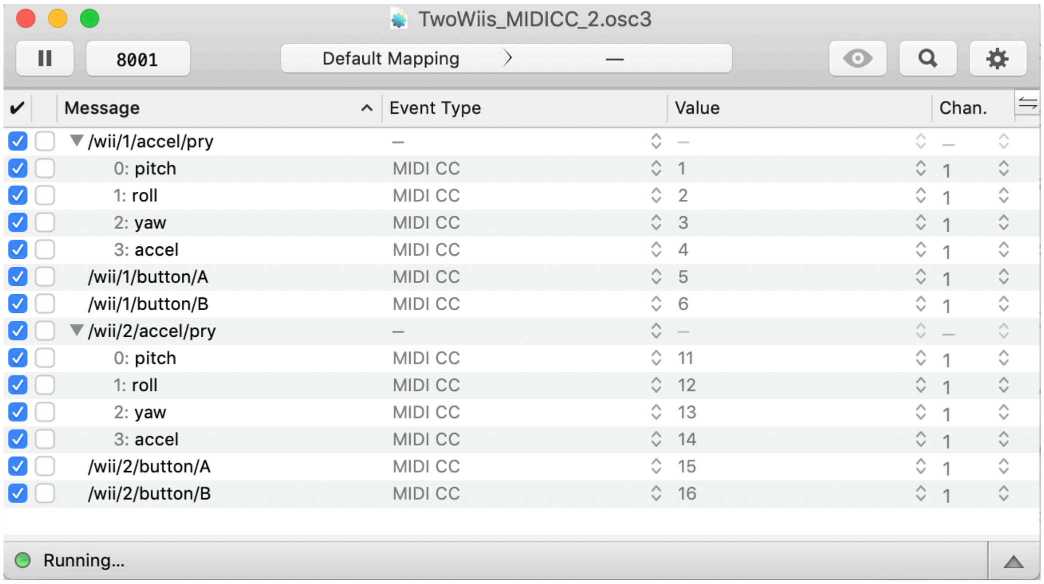


Figure 2. The overall structure of the data-driven instrument used in *Frozen*

---

[18] "What are Pitch, Yaw, and Roll?" Wiimote FAQ, OSCulator, accessed March 9, 2020, https://osculator.net/doc/faq:wiimote.

In *Frozen*, I used two Wiimote controllers as the performance interface. The OSCulator software receives data streams from two Wiimote controllers through the Bluetooth connection and translates that data into MIDI continuous control messages. That matrix is shown in Figure 3.



✔		Message	Event Type	Value	Chan.
☑	☐	▼ /wii/1/accel/pry	–	–	–
☑	☐	0: pitch	MIDI CC	1	1
☑	☐	1: roll	MIDI CC	2	1
☑	☐	2: yaw	MIDI CC	3	1
☑	☐	3: accel	MIDI CC	4	1
☑	☐	/wii/1/button/A	MIDI CC	5	1
☑	☐	/wii/1/button/B	MIDI CC	6	1
☑	☐	▼ /wii/2/accel/pry	–	–	–
☑	☐	0: pitch	MIDI CC	11	1
☑	☐	1: roll	MIDI CC	12	1
☑	☐	2: yaw	MIDI CC	13	1
☑	☐	3: accel	MIDI CC	14	1
☑	☐	/wii/2/button/A	MIDI CC	15	1
☑	☐	/wii/2/button/B	MIDI CC	16	1

Figure 3. Data streams received in OSCulator from two Wiimote controllers

Instead of using all the eleven buttons and four continuous controllers, only two buttons, one in the front and one in the back, and four continuous data streams, *Pitch*, *Roll*, *Yaw*, and *Accelerometer*, were used for each Wiimote controller. These data were sent out as messages on MIDI channel 1. To distinguish the different data streams, each of them was assigned to a different controller number. For Wiimote number 1 the six data streams were assigned to controller numbers 1 to 6. The data streams from Wiimote number 2 were assigned to controller numbers 11 to 16.

Through the assistive software KymaConnect, which helped pass data from OSCulator to Kyma, Kyma received the twelve data streams and responded to the data by triggering audio files of sounds (e.g., the striking and shaking of iced tree branches) and by modifying musical parameters (e.g., frequency and amplitude).

Musical Challenges and Opportunities

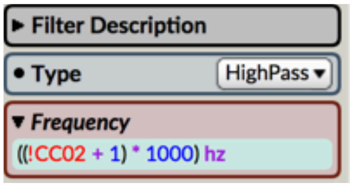
The Wiimote controller's capability of generating both triggers and continuous data streams provided many opportunities as a composer and performer to realize a wide variety of musical results. By pressing buttons, I could start musical events. By tilting and rotating the Wiimotes in space I could generate ongoing control signals that could shape my sounds depending on how the data streams were mapped to musical parameters.

However, as a performer of a data-driven instrument, there was a particular problem related to making the performance more observable. This problem exists most particularly with data-driven instruments because the audience (the observer) almost certainly knows far less about how sounds in the instrumental apparatus are generated than they know about traditional acoustic instruments. To solve this problem, constituting clearer, observable performative actions could help the audience better accrue understanding and appreciation of the instrument. For instance, pressing the buttons on the Wiimote was one of the performative actions that was difficult to observe. This difficulty derived from the button's smallness and their location on the Wiimote surface. To resolve this problem, I combined the actions of pressing buttons with other more observable performative actions. For example, a sound was started by pressing one button. At the same time, I also raised my right hand to emphasize the action of pressing a button (see supplemental video at 0:24). After these two actions occurred back to back, I also rotated the Wiimote's *Roll* continuous control to modify the frequency of the sound. As a positive collateral outcome, data created by depressing the button simultaneously occurred with data produced by rotating the Wiimote to simultaneously support my desired musical outcomes.

Another challenge of using the controller was that the data streams it produced were entangled and were not independent of one another. This means that when I moved the *Pitch* of the Wiimote upward and downward, the *Accelerometer* data also changed in accordance to, but not fully in parallel with, the *Pitch* data. Therefore, when designing the data mapping architecture, I decided to either use one data stream at a time or to embrace the entanglement of multiple data streams. For example, at the beginning of the piece, only the *Pitch* data was used to control the frequency of a sound. In a later section of the piece, I mapped *Accelerometer* data to turn sounds on and off in which the big performative actions could be seen. At the same time, I mapped the *Pitch* continuous control data to the frequency parameter of the harmonic resonator. In this case, I took advantage of the multiple continuous control data streams.

Data Mapping Strategies

Frozen employed a number of data mapping strategies including data scaling and offsetting. For example, as Figure 4 shows, the *Roll* data, which appeared in Kyma as continuous controller number 2 (!CC02), was mapped to control the cutoff frequency of the highpass filter. The original *Roll* data, which ranged from 0-1, was first offset to 1-2 by adding 1 and then scaled to enlarge the effective frequency range by multiplying by 1000. As a result, I was able to control the cutoff frequency of a highpass filter between the frequencies of 1000 and 2000 Hz.



▶ Filter Description

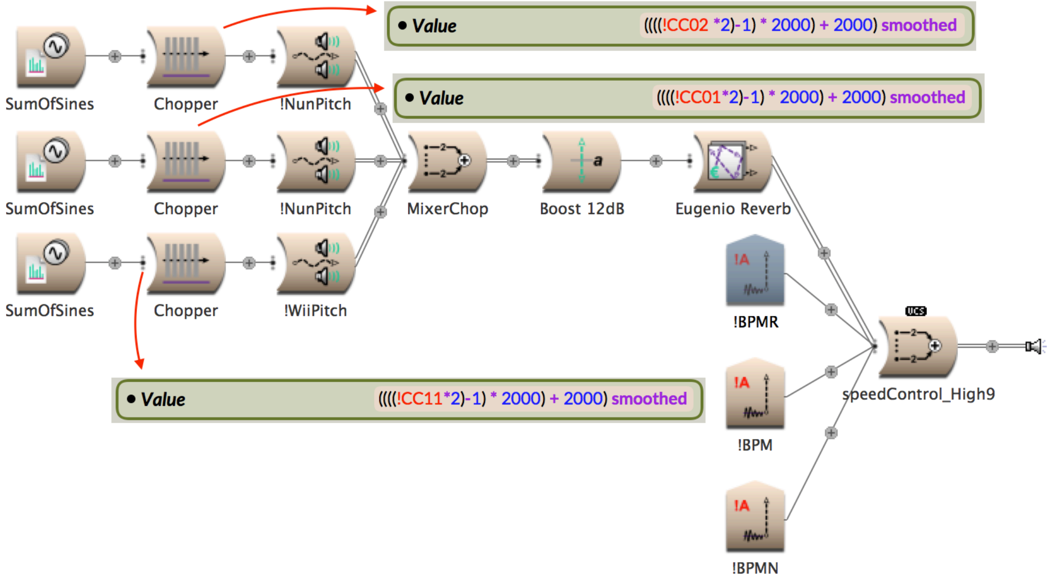
● Type HighPass ▾

▼ Frequency

((!CC02 + 1) * 1000) hz

Figure 4. Using data offsetting and scaling to control timbre

In addition to controlling the timbre of sounds in real time, data offsetting and scaling were also used to control how fast the sounds were being amplitude modulated. As Figure 5 shows, the rate of the amplitude modulation of the three sounds in the signal flows were controlled by three distinct continuous controls. After using the same data offsetting and scaling operations all three continuous controls streams produced chopped speeds within a range of 0-4000 beats per minute.



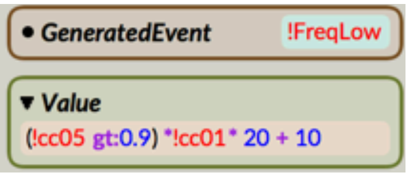
● Value (((!CC02 *2)-1) * 2000) + 2000) smoothed

● Value (((!CC01*2)-1) * 2000) + 2000) smoothed

● Value (((!CC11*2)-1) * 2000) + 2000) smoothed

Figure 5. Applying data offsetting and scaling to control BPM

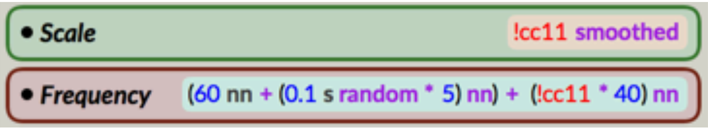
Since the Wiimote controller provided both on-off buttons and continuous controls, these two types of data were combined to control musical parameters. As Figure 6 shows, the *Pitch* data, which was shown as continuous controller number 1 (!cc01), changed the frequency parameter of a sound only when the Button A on the Wiimote was depressed.



```
• GeneratedEvent    !FreqLow
▼ Value
(!cc05 gt:0.9) *!cc01* 20 + 10
```

Figure 6. Using both on-off buttons and continuous controls data

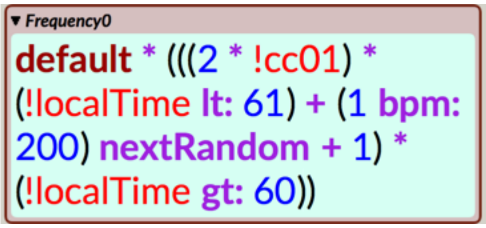
In addition to applying more than one data streams to one parameter field, I also assigned one data stream to multiple destinations. Figure 7 shows that the *Pitch* data stream, appearing as continuous controller number 11 in Kyma, was used to simultaneously control the amplitude and frequency parameters of the sound.



```
• Scale        !cc11 smoothed
• Frequency    (60 nn + (0.1 s random * 5) nn) + (!cc11 * 40) nn
```

Figure 7. Assigning one data stream to multiple destinations

Sometimes I used temporal context or position to make a controller active or inactive. As Figure 8 shows, the *Pitch* data, appearing as continuous data number 1 (!cc01) in Kyma, modulated the frequency of a sound only for the first 61 seconds. When the time reached 61 seconds, the frequency ceased being affected by *Pitch* data, and instead was randomly changed within a range of 0-120 (for MIDI note numbers) every 0.3 seconds (200 BPM).



```
▼ Frequency0
default * (((2 * !cc01) *
(!localTime lt: 61) + (1 bpm:
200) nextRandom + 1) *
(!localTime gt: 60))
```

Figure 8. Temporal context or position activating or deactivating control

Turning continuous data streams into button-like functions was used in later parts of the composition. Most of the sounds in *Frozen* were triggered through the four on-off buttons from the two Wiimote controllers. With the development of the composition, more and more dramatic and intensive sound changes were presented. Simply pressing buttons, which involved small performative actions, might have fulfilled the musical needs of the composition, but would not have met the expectations of an audience; an audience who had been conditioned through traditional acoustic instrumental performances to associate grand musical outcomes with large physical movements. Simply stated another way, if a drummer wants to produce a loud sound rather than a soft sound, the physical movement to achieve the louder sound is apt to be larger. In this case, I designed and used larger and more intense movements to cognitively align with the production of these more intense sounds. As Figure 9 shows, *Accelerometer* data, which was generated through large and dramatic physical movements, was used to trigger a sound and produce the indeterminate frequencies at the moment where intense sounds were involved.

Gate !CC04 gt: 0.8

AttackTime 1 samp

ReleaseTime 86400.0 s

Scale 1

Frequency

(50 -((!CC04 gt: 0.8) nextRandom abs * 20)) nn

Figure 9. Turning *Accelerometer* data into a button-like function

Sonic Motives

My concept of motive was deployed as a primary compositional feature in *Frozen* as two distinctive sonic motives. These two sonic motives occur over the course of

Frozen whose five-section form can be described as: A-B-C-D-A'. Both motives occur in all sections. My discussion below identifies each of the sonic motives and describes how it is developed over the five sections of the composition.

Rhythmic Motive

The primary motive used in this composition was derived from a recording of iced, tree branches being struck. The primary recording from which the motive arose contained three clear individual sonic events. Each of the three sonic events were at different amplitude levels and each generated a distinctive rhythmic pattern. Although the recording was not presented in its original form, the entire piece was developed from the recording's rhythmic traits and presented in variation. Below I discuss how the rhythmic patterns as well as other main sound materials were presented in the different sections of this composition.

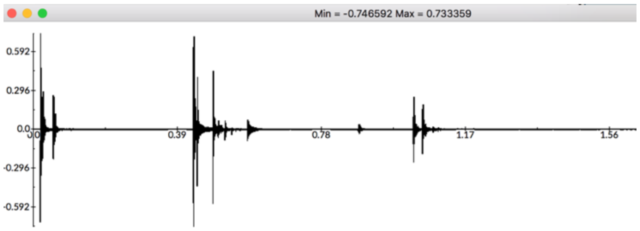


Figure 10. Recording of hitting iced tree branches

At the beginning of the piece, the rhythm pattern was created from a single striking of iced, tree branches as the input sound source and was triggered at a constant rate with a randomized amplitude (see Figure 11). This sound source was heard through a Kyma *CrossFilter* Sound object whose response was the original recording of hitting the iced, tree branches. As a result, the sound which followed the input and response created a constant rhythmic pattern with unpredictable rhythm accents (see Figure 12). In later parts of this section, I added an additional layer of rhythm that disrupted the constancy of the unchanging pulse. When I quickly moved the Wiimote upwards, downwards, or sideways, bell sounds of this new layer were triggered interrupting the previously constant rhythm. The breaking of the constant rhythmic pulse foreshadowed the next section which started with no perceptible rhythmic organization.

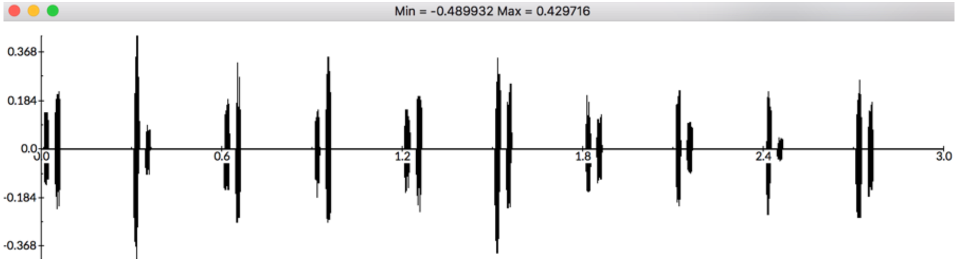


Figure 11. Input with constant rhythmic pulse

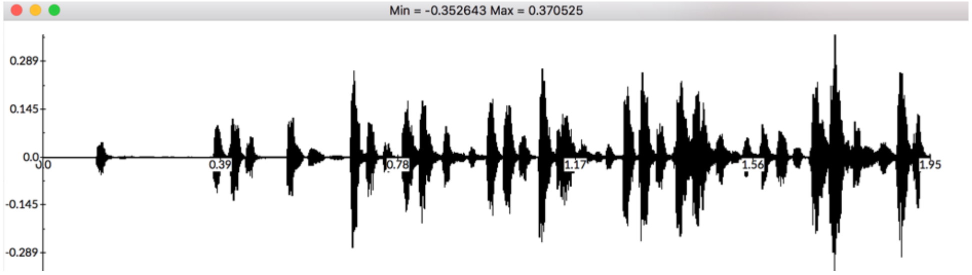


Figure 12. Rhythmic pulse with deconstructed through random amplitude modulation and sonic interjection

The B section (3:09-5:20 of the supplemental video) began without a clear rhythmic organization or pulse with only a sustained drone sound present. The rhythmic elements were introduced only in the second half of this section. Instead of having a constant rate used as in the previous section, the rhythmic elements were developed to incorporate rate changes which were realized through amplitude modulation. The accelerating and decelerating rate changes in this section were always followed by a crescendo. A decelerating speed of a bell sound was introduced when the crescendo was longer than five seconds. An accelerating rate of the bell sound was displayed concurrently with the crescendo over approximately two seconds. These two decelerating and accelerating rate changes were presented one right after another three times and were triggered by the right and left Wiimote, respectively. The antiphonal relationship between these two types of rate changes were similar to a conversation between the two Wiimote controllers.

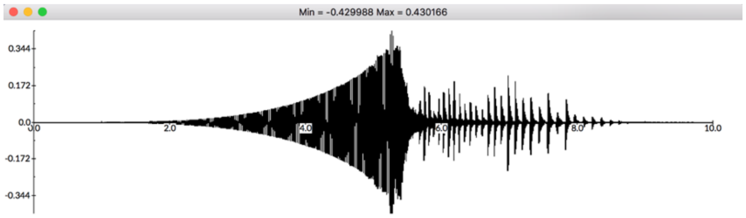


Figure 13. Longer crescendo with descending speed

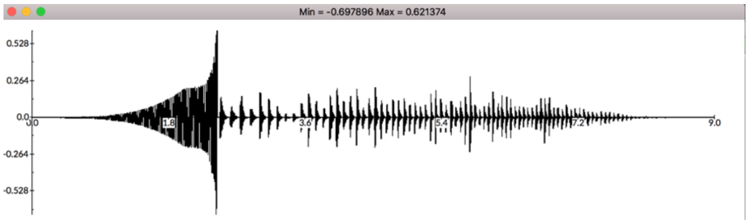


Figure 14. Shorter crescendo with ascending speed

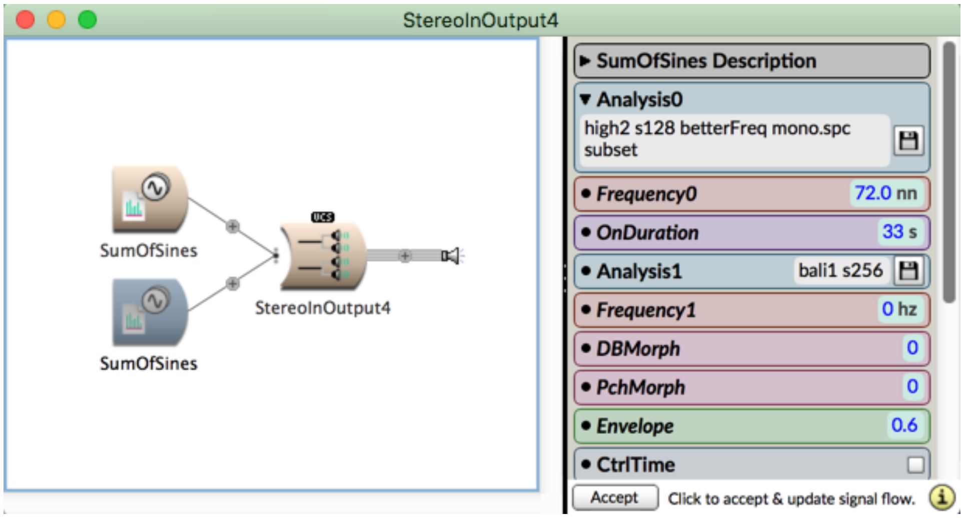
The musical feature of constant rhythmic articulation returned in section C (5:20-7:15 of the supplemental video). However, unlike the constant rhythmic rate articulated in the opening section where sounds were triggered at a single BPM rate, the sound in this section were triggered at 480 BPM and 960 BPM. I switched back and forth between these two rates by pressing the button on the backside of the Wiimote (Button B). Like the antiphonal conversation between the music controlled by the two Wiimotes in the previous section, section C also presented a conversation-like duet for the two controllers. The right Wiimote controlled the frequency and tempo changes of the high-frequency bell sounds. The left Wiimote controlled the same musical material and parameters, but the output frequencies were much lower.

The motive in section A’ (7:15-8:41 of the supplemental video) was essentially a restatement of the sonic features of the first two sections. The rhythmic motive derived from section B in which the speed of the motive was manipulated by the Wiimote*’s Roll* data. The accelerandos and ritardandos in the articulations of the sounds were realized using Kyma’s *Chopper* Sound object. Simultaneously, the bell sounds (occurring between 7:50-8:42 of the supplemental video), deriving from section A, were triggered and processed through the *Harmonic Resonator* Sound object which is a filter that has resonances at a specified frequency and at all of its harmonics. This specified frequency of the *Harmonic Resonator* was controlled in real time by the Wiimote’s X-dimension, *Pitch* data.

The A” section of the composition (8:41-9:27 of the supplemental video) recalled the previous sound materials in varied forms with sounds being presented primarily as decelerating rhythmic patterns and with the bell sonorities being presented with decreasing density. The decelerating rhythmic pattern was automatically produced through a Kyma Capytalk algorithm. The decreasing density of the bell sounds was manually realized through performative actions where the buttons were pressed less and less to gradually fade the composition to silence.

Sustained Bell Motive

In addition to the rhythmic motive based on the recording of iced, tree branches being struck, the sustained bell sounds were also essential sonic materials. In the first two sections the sustained sounds were realized through *SampleCloud* Sound objects where groups of grains form the continuous sonic tapestry. In these two sections frequency and timbre changes were gradually introduced. In section C, the sustained sounds were achieved through analysis and resynthesis techniques in which bell sounds were stretched from just a few seconds to as much as 243 seconds. These two ways of forming sustained sounds, granular synthesis and analysis and resynthesis, were used again during the last two sections and with subtle variation. The grains in section A' at first remained unchanged at a single frequency and over time indeterminately jitter both above and below the center frequency with greater and greater intensity and variation. The resynthesis sound in A" section was shortened to 33 seconds (see Figure 15).



Analysis0: high2 s128 betterFreq mono.spc subset
Frequency0: 72.0 nn
OnDuration: 33 s
Analysis1: bali1 s256
Frequency1: 0 hz
DBMorph: 0
PchMorph: 0
Envelope: 0.6
CtrlTime: ☐

Figure 15. Resynthesis stretched sound in section A"

Performance Techniques

The primary performance technique employed in *Frozen* can be conceptually simplified to pressing of buttons on both front and back side of the Wiimote to start musical events, to tilting of the Wiimote in *Pitch* and *Roll* dimensions to produce ongoing musical controls, and to breaching numerical values in order to trigger sonic events.

Pressing buttons can be seen at the beginning of the piece where I pressed the button on the back side of the left Wiimote controller to start the composition. Sound events were extensively triggered by pressing the two buttons, one on the front and one on the back side, of a Wiimote controller. This technique can be observed in the rhythmic duet, between 6:10-7:15 of the supplemental video, where both the front and back buttons of the two Wiimote controllers were used. Button A on the fronts of the two Wiimotes were mapped to start the duet and simultaneously control the frequency. The single buttons on the back of the two Wiimotes were mapped to switch in alternation between fast and slow rhythmic rates (occurring start at 6:44 of the supplemental video).

Changing the directions in which the Wiimote moved produced ongoing musical control. These performative actions can be found between 0:40-1:02 and 6:12-6:30 of the supplemental video. In the first example, I tilted the controllers upward and down to produce frequency changes, from high to low, respectively. In the second example, I twisted both Wiimotes to control the cutoff frequency of a highpass filter for the purpose of changing the timbre of the sound.

I also used *Accelerometer* data to breach numerical thresholds in order to trigger a number of sounds. This performance technique was apparent between 2:39-2:53 (bell sound), 4:45-5:19 (bass sound), and 8:26-8:42 (bell sound) in the supplemental video. In all of the examples, sounds were triggered when I speedily moved the two Wiimote controllers. The larger and more intense movements caused by *Accelerometer* data during these times cognitively aligned with the production of more intense sounds. In the last example, because the composition was developed, I also used the *Pitch* data to control the frequency of the sounds.

CHAPTER III.2

ORIGIN

Overview

Origin is a multichannel real time performance composition for Wacom Tablet and Kyma. The piece was based on audio recordings of pins dropping, the striking of glass, various percussion instruments, and the human voice.[19] By mapping the Wacom pen and touch controls to resident parameters in the sound-producing algorithms contained in the Kyma creation environment, I controlled the manipulation of sonic materials. The sonic materials developed and unfolded as musical journeys that are both dramatic and nuanced.

Design and Implementation of the Data-driven Instrument used in *Origin*

Since Kyma provided the capability to perform the data mapping, Kyma directly received data from the Wacom Tablet without using additional software. Thus, the data-driven instrument used in this composition was compressed into two parts: 1) the Wacom Tablet that output data when I operated it, and 2) Kyma that provided both mapping and sound-production services, which were parts two and three of the conceptual data-driven instrument.

---

[19] The type of pin I to which I refer is the type often used in sewing and is a short, sharp and thin piece of metal with a sharp point at one of its ends.

Figure 16. The overall structure of the data-driven instrument used in *Origin* (image credit: *Wacom*).[20]

When I operated the Wacom Tablet, two distinct types of data messages were output: messages that triggered events and messages that provided ongoing control. When I used the pen, the Wacom Tablet sent triggering messages including when I touched and the pen tip touched the tablet, and when I depressed the pen buttons. Additionally, the Wacom sent ongoing control messages that spanned continuous ranges of values, such as when the pen traversed the X and Y dimensions of the tablet or when the pen was tilted. When I used my fingers, the Wacom sent events that Kyma received and interpreted as data related to MIDI note numbers to start sounds and to specify pitches. Kyma also treated finger location on the tablet in the Y-dimension as data ranging from 0-1, the finger equivalent to the Y-dimension control of the pen. Data arising from pen and finger controls met the composition's performative needs.

Musical Challenges and Opportunities

The Wacom tablet is a device that was originally designed as a drawing tool for digital illustrators.[21] When the pen or my fingers touched the surface of the Wacom tablet

---

[20] "Wacom Intuos Pro- Large," Wacom, accessed February 6, 2020, https://estore.wacom.com/en-US/wacom-intuos-pro-l-us-pth860.html.

[21] Zbyszynski et al., "Ten Years."

the data output from the Wacom tablet was stable and precise. The stability and precision of the Wacom's data provided for me opportunities to create and manipulate subtle sonic changes. One of the most prominent challenges of using the Wacom tablet as a controller is the asymmetrical dimensions of its active performance area. The Z-axis, because it is based on pen pressure, has a very small physical range as compared to its X and Y dimensions. The large size of Intuos Pro Wacom is 311 mm long and 216 mm wide.[22] The X and Y axes of the Wacom offer 0 to 32480 output precision for each dimension.[23] However, the Z-axis has a 1024 discrete value compressed within a very small physical range, which is difficult to see the effort one applies when pressing the pen or the finger against the tablet. The small physical range makes the Z dimension much more difficult to use and observe during musical performance, so I chose to construct this piece using primarily the X and Y data streams.

Data Mapping Strategies

As in my other compositions contained in this portfolio, *Origin* also applied many data mapping strategies. Of these, basic data scaling and data offsetting were the most prominently used. In most of the cases, the Y-axis data was mapped to control amplitude with the bottom of the tablet being associated with softer musical volumes and the top of the tablet corresponding to louder musical volumes. Here, I used simple multiplication and addition to scale and offset this data's range from 0 to 1 to 0 to 0.5. Additionally, the X-axis data was mapped to control frequency—the left side of the tablet corresponded to

---

[22] "Wacom Intuos Pro- Large."

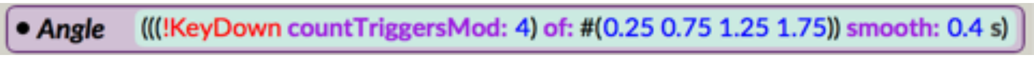
[23] Zbyszynski et al., "Ten Years."

low frequencies and the right side corresponded to high frequencies. Beyond this one-to-one mapping for frequency, in some cases frequency was indeterminate and depended on algorithms that randomly selected outcomes from an array of numbers. Within the composition, whenever my finger touched the surface of the tablet the action generated a KeyDown message that triggered the selection of one of five numbers. These numbers represented the MIDI note numbers 57, 60, 62, 64, and 67 that controlled the frequency of the specific sound. The actual Capytalk expression is shown below in Figure 17.

```
• Frequency   (((!KeyDown nextRandom abs * 5) rounded) of: #(57 60 62 64 67 ))nn
```

Figure 17. KeyDown message triggers the random selection of one of five MIDI notes

Algorithms that were not indeterminate were also used within *Origin*. Figure 18 shows how I triggered a sequence of numbers that were specifically ordered. In this example, KeyDown messages were counted to produce a sequence of integers that range from 0 to 3. These four numbers were mapped to the values 0.25, 0.75, 1.25, and 1.75 that were routed to control the angle parameter of a *MultichannelPan* Sound object. The musical outcome of this mapping was that the sound was moved clockwise in a circle of the multichannel listening environment. Data smoothing was also used in this parameter field, in which the sound took 0.4 seconds to gradually move from the current position to the next.



```
• Angle   (((!KeyDown countTriggersMod: 4) of: #(0.25 0.75 1.25 1.75)) smooth: 0.4 s)
```

Figure 18. KeyDown message in order triggers four pan positions

Some sonic events in this composition were triggered when multiple conditions were met. As shown in Figure 19, the three sounds, small cymbal, Chinese cymbal, and orchestra cymbal, were triggered when I moved my finger to three corners of the Wacom

Tablet. In each of these cases, the location of three corners were calculated based on the values of both X and Y-axes. For example, the “Small Cymbal” sound was triggered when both the values of the X and Y axes values went beyond 0.9; when I placed my fingers in the right, upper most corner.

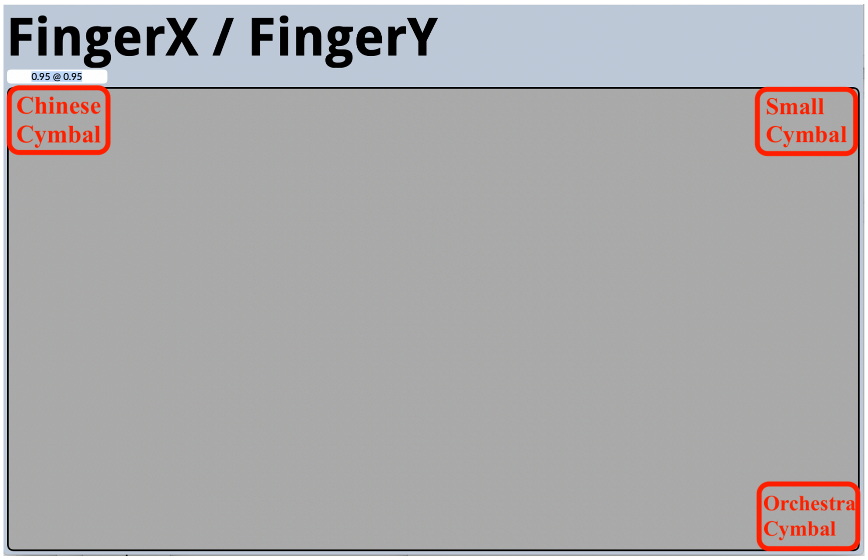


Figure 19. Three sounds are triggered based on meeting multiple conditions

Sonic Motives

My concept of motive was deployed as a fundamental compositional feature in *Origin* as four distinctive sonic motives. These four sonic motives occur over the course of *Origin* whose five-section form can be described as: A-B-C-D-A’. Of the four sonic motives some are more prominently featured than others. The struck glass motive occurs in sections A, B, and C; the percussion motive occurs in in sections D and A’; the human voice motive occurs in section C and D; while the primary sonic motive that I identify as pin dropping occurs in all sections. The pin dropping recording is about one second in duration and creates a percussive-like amplitude envelope. To construct compositional unity within the work, other sound materials used in this composition have similar envelopes. This amplitude envelope shape served as a motive for this composition with sounds of glass being struck and the percussion instruments all possessing this amplitude in a sharp attack and gentle decay envelope. Below I identify each sonic motive and describe how it is developed over the five sections of the composition.

Pin Dropping Motive

The composition started with the pin dropping. This pin-dropping sound was played twice which formed a two-note pattern. In a later part of section A, the length of pin-dropping sound was doubled which included a sequence of four pin-droppings. While each of the pin-dropping sounds maintained the envelope of a sharp attack and a gentle decay, the overall amplitude shape of the four sound events in series formed a decrescendo, realized through a fade-out in volume for the entire sound (see Figure 20). In the beginning part of section A, the pin dropping motive was developed by extending its duration.

Figure 20. Decrescendo of four pin-dropping sound

Section B (2:47-3:39 of the supplemental video) kept the original two-pin-dropping motive as the initial sound. However, the four-pin dropping was developed in this section. By applying a granular reverb processing algorithm to the four-pin-dropping sound, a granular trail of tiny sonic particles was created. Since the sound material of each one of the grains came from the pin-dropping recording, the motive of the four-pin-dropping sound in this section was further extended by a cluster of grains.

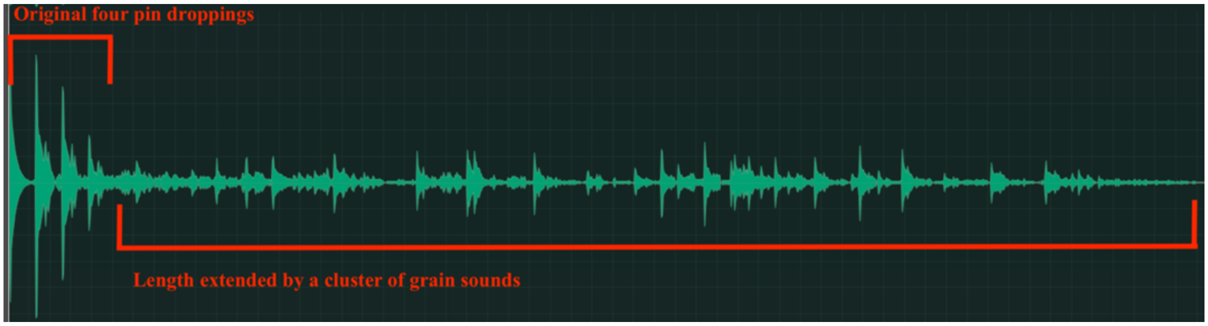


Figure 21. Expanded four pin-dropping sound

In section C (3:39-5:17 of the supplemental video), the pin-dropping motive was presented fully transformed, unrecognizable, and disconnected from its original form. There are two variations derived from the pin-dropping. The first variation in section C included two parts. The first part served as a bass drum, realized by transposing the sound to a low frequency, and the second part created a moving noise effect, produced by emphasizing the harsh part of the sound. The different sonic characteristics of the two parts of the two sounds were heard clearly when played simultaneously. The second variation also had two parts. The first part was the reverse of one pin-dropping sound in which expectations and tensions were built through small crescendos, and the second part played the four-pin-dropping sound forward but at variable playback speeds in which a variety of rhythmic combinations was created to avoid repetitions of an unaltered looped audio file.

The pin-dropping motive used in section D (5:17-6:39 of the supplemental video) related to its initial presentation in the beginning section of the composition. As in section A, the two-note pattern of the pin-dropping motive articulated the beginning of section D. The four-pin-dropping sound kept the decrescendo feature as in section A but expanded its frequency range of MIDI note numbers from a perfect fourth (55-60 nn) to two and half octaves (15-45 nn).

In the last section (6:39-9:28 of the supplemental video), the pin-dropping sounds summarized the previous motives' sonic features and continued their development. At the beginning of the section, the sound material combined both the reversed and original pin-dropping motives derived from section C (see Figure 22). This sound's duration was also extended by using the same granular synthesis technique used in section B. However, instead of presenting both the original motive and extended sound, only the extended grain sounds occurred in this section. The next sound—based on the pin-dropping motive—borrowed the rhythmic feature from the previous sections. The composition ended by recalling the beginning two-note pin-dropping motive.

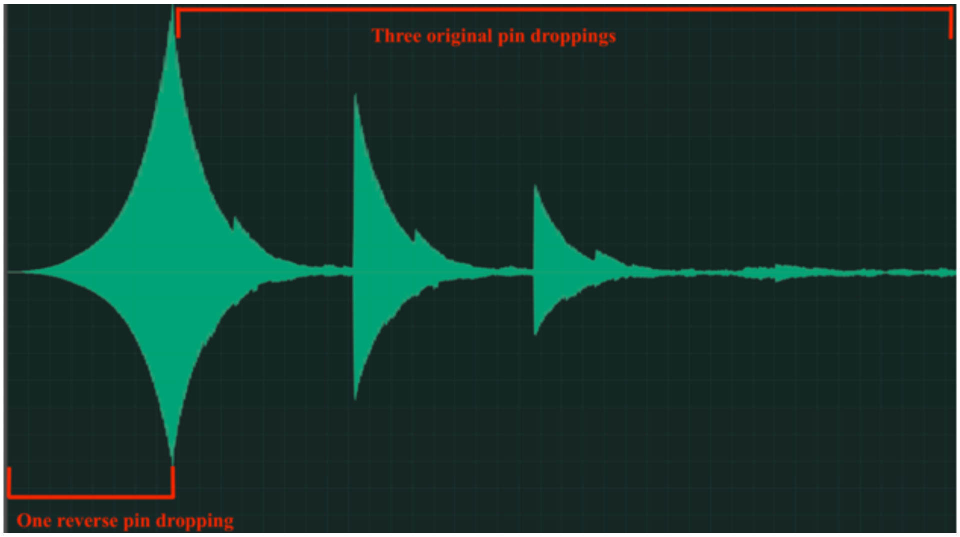


Figure 22. Waveform of the beginning pin-dropping sound used in section D

Struck Glass Motive

The struck glass motive included only one sound event of two glasses hitting each other. Taking advantage of the obvious pitch feature the sound material had, the struck glass motive was mostly handled as a melodic line in this composition. The pitches that the struck glass motive used were based on the traditional Chinese pentatonic scale in which the sound was transposed into one of the five pitches when triggered.

In the beginning section, the struck glass motive occurred in a pentatonic scale of A C D E G. One of the five pitches was randomly triggered each time my finger touched the surface of the Wacom tablet. Up to two occurrences of this motive could simultaneously occur.

In section B, the struck glass motive, realized through the use of the same pentatonic scale, occurred with an enlarged frequency range. The pentatonic scale was transposed upward by one whole step each time the struck glass motive was triggered.

In section C, the presentation of the struck glass motive was based on a pre-recorded melody comprised of the pentatonic scale discussed above. In this section, I gradually transposed the entire recording up bit by bit over the course of forty seconds by applying a Capytalk expression to modulate the frequency upward by a minor seventh.

Percussion Instruments Motive

Percussion sounds formed another sonic motive that I used during the later parts of the composition. In section D, the percussion motive contained three cymbal recordings: a small cymbal, an orchestra cymbal, and a Chinese cymbal.[24] These three recordings, drawn from the same basic instrumental family, had slight timbral differences. The small cymbal contained more pitch elements. The orchestra cymbal had many noisy partials during the attack phase. The timbre of the Chinese cymbal rested between the small cymbal and orchestra cymbal, containing both pitch and noisy features.

---

[24] The Chinese cymbal audio file was taken from the *Legend of China* sample library, Discovery Sound, 1999.

Each one of these recordings was triggered using different conditional mapping strategies. The motive was presented in various rhythmic patterns depending on when I triggered to meet the conditions of my mapping operations. This percussion motive began sparsely, where a single cymbal sound as triggered approximately every five to ten seconds. When additional sonic materials were introduced later in the section the percussion motive gradually increased its density of occurrences. By increasing the density of its occurrences, all three cymbals, originally sounding every five seconds, were triggered approximately twice a second. The composition reached its moment of climax concurrently with the point of maximum cymbal activity.

In the final section, the percussion motive used only the small cymbal recording and was realized as fourteen unique sonic manifestations through the use of the *Replicator* Sound object. These fourteen cymbal sounds possessed distinct trigger times, frequencies, and spatial positioning, and each began their playback at different starting points of the recording. The small cymbal sound with its obvious pitch feature created a fourteen-note melody that recalled the out of tune struck glass motive in section C. The composition ended with this sequence of cymbal sounds.

Voice Motive

The voice motive first occurred in section C and consisted of a swarm of grains based on a recording of a male voice speaking from Chinese opera.[25] The grains of the human voice motive were generated through a granular synthesis technique where each grain was detuned a little bit from other grains. By adding lowpass filters in the signal

---

[25] The male voice audio file was taken from the *Legend of China* sample library, Discovery Sound, 1999.

flow, the voice motive was gradually presented in a wider frequency range as the cutoff frequency of the lowpass filter was changed from 300 to 5000 Hz over a one-minute time span.

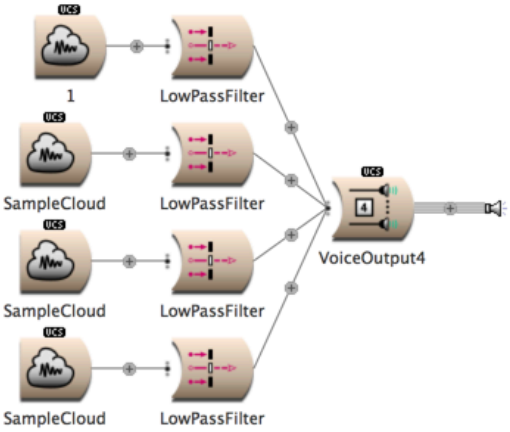


Figure 23. Sound structure of human voice motive

The male voice speaking recording was revealed more extensively in section D. Instead of using grains that only present the voice in very short durations, the voice motive used the *TAU* Sound object that allowed me to present the entire length of the male speaking recording and to reveal the complete spoken text for the first time. When mapping the PenX to the *SampleCloud*'s TimeIndex parameter, I was able to arbitrarily access any part of the voice recording to present the male speaking voice in great sonic detail. Just like the previous voice motive, the four *TAU* Sound objects also were tuned to frequencies that differ slightly from each other.

Performance Techniques

The essential performance techniques of *Origin* included 1) applying the pen to the tablet and moving it in the X and Y dimensions, 2) placing my finger directly to the tablet to start musical events, 3) moving my finger or fingers across the tablet in the X and Y dimensions to produce ongoing control of timbre and frequency, and 4) tilting the pen in the Y-axis to modulate timbre. Playing the instrument by applying the pen to the tablet and moving it in the X and Y dimensions can be seen at 1:46-2:39 and 5:25-6:38 of the supplemental video. In these instances, the pen was used to trigger events and to control the frequency of those events. In one case, at 2:53-3:24, touching the pen to the tablet caused a stream of indeterminate numbers to be produced. Touching the tablet directly with my fingers played a prominent role. I used the finger-to-tablet technique to start musical events. The position on the tablet where my finger touched was not significant because the pitches of these events were indeterminate and were generated algorithmically within a range that was compositionally specified beforehand.

Having a good intuitive sense of the temporal flow was essential in performing this composition. This fact was perhaps most apparent when I addressed the Wacom Tablet like a keyboard. In sections A and B, it was not where I touched the tablet that is important, but rather when I touched the tablet. Because the pitches for each of these triggered events were algorithmically produced and were indeterminate, my work as a performer involved timing each event musically within the overall flow.

Three motives were manipulated by the two sets of data controls that the Wacom Tablet provided individually. The pen controls were mostly dedicated to the pin-dropping motive. The struck glass and percussion motives were mostly manipulated by finger controls. This separation of different controls to different motives offered better and clearer observation opportunities for the audience. The performance technique took advantage of the Wacom's precision and stability. For example, PenX was mapped to control the playback location of the human voice recording. A small change in X-axis resulted in dramatic musical changes. The precise data values provided by the Wacom Tablet granted the ability for me to explore the moment to moment timbre changes of the human voice. This precision also challenged me to be very accurate to get expected musical results. I needed to place the pen or my fingers at precise locations on the tablet in order to get the desired X and Y values.

CHAPTER III.3

PULSING

Overview

Pulsing, for Roli Lightpad Block, custom Max software, and Kyma, is a stereo interactive composition based on the sound of a female voice. The primary compositional material was a recording of a woman singing Inner Mongolia Chinese folk music. By applying real time processing through Kyma's *Chopper* and *Delay* Sound objects, as well as using sequencing and granular synthesis techniques, the original structure of the folk song was rearranged, but the singer's elegant voice and superb singing skills remained intact and were apparent even through considerable sonic alterations.

Design and Implementation of the Data-driven Instrument used in *Pulsing*

The data-driven instrument used in this composition was comprised of three parts: 1) the Lightpad Block controller that output data when I operated it, 2) Max, which analyzed the initial data that came from the Lightpad Block and created replacement data based on its analysis, and 3) Kyma, which received the new data as MIDI messages and responded to it by producing sonic events. The helper application KymaConnect served as a go between facilitating the passing of data from Max to Kyma. The basic instrumental structure is shown in Figure 24.

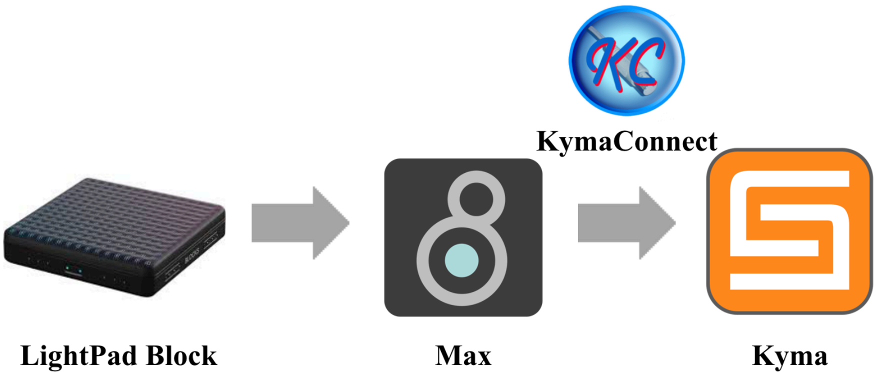


Figure 24. The basic structure of the data-driven instrument used in *Pulsing*

The Roli Lightpad Block is a small device whose width, height, and depth are only 3.7 x 0.86 x 3.7 inches.[26] The size of this device impacted how I engaged it and ultimately how I used it in the performance. Based on where and how the device was touched control messages were sent to the computer via a USB connection. The Lightpad Block reported the on-off status of the toggles contained in Lightpad Block layouts and the number of fingers touching its surface. Additionally, the Lightpad Block reported the location of each point of physical contact on its two-dimensional surface. As a means of enunciation when a finger touched it, the Lightpad Block responded to a touching action by illuminating itself at the location it was touched.

To get data from the interface to the software layer, I used three external Max objects to receive data from the interface: *blocks.pad*, *blocks.multitouch*, and *blocks.toggle*. The *blocks.pad* object was used for creating user interface layouts. The concept of a layout for the Roli Lightpad Block was comprehensive and involved virtually all aspects of its operation. For example, a layout saved information about the number of toggles, where the toggles were located on the surface of the interface, the

---

[26] 26. “Lightpad Block,” Roli, accessed April 28, 2020, https://roli.com/products/blocks/lightpad-m.

numbers of points of physical contact, among other settings. The *blocks.pad* object allowed up to sixteen layouts to be created for the Lightpad Block, and I created four layouts for the Lightpad Block controller for *Pulsing* (see Figure 25). The *blocks.multitouch* object reported the location of fingers on the surface based on the X and Y axes. The *blocks.toggle* object provided the ability to toggle between on and off states.

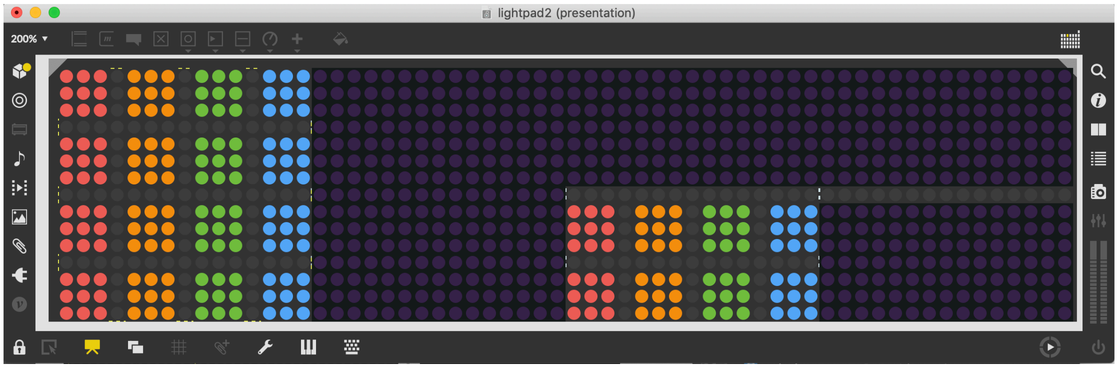


Figure 25. Four layouts used in *Pulsing*

To switch between the four layouts, I pressed a button that was positioned on the left side of the Lightpad Block. By switching between layouts, I was able to change what data got sent and the performative actions used to send that data.

The first layout contained sixteen toggles that were distributed over the performance surface in a 4 X 4 matrix, in which each one of the toggles only sent on and off messages (1 and 0). Since Kyma received on-off MIDI messages as 0 and 127, data scaling was applied to the original data in Max to prepare data for mapping.

The second layout used only one multitouch object that was able to report multiple data points as finger location, pressure, and velocity. To transmit my musical intentions, I used data values sent based on my fingers’ location in the X and Y dimensions, pressure, velocity, and the number of fingers used.

Data scaling was employed to all these three data streams, but each one of them was scaled in a slightly different way. The data stream of fingers index reported all integer numbers of how many fingers touched on the surface of the Lightpad Block. I decided to narrow the range down to five which meant up to five fingers could touch the controller at the same time, and then scale this original data range 0-5 to an appropriate MIDI data range of 0-127. Since both X and Y axes output data from 0 to 1, a simple mathematical operation of multiplying by 127 scaled the data appropriately. Because the Y-axis by default placed higher numbers on the lower portion of the interface and lower numbers on the upper part of the interface, I inverted this arrangement using the *scale* object to make the data produced seem more logical to me (see Figure 26).

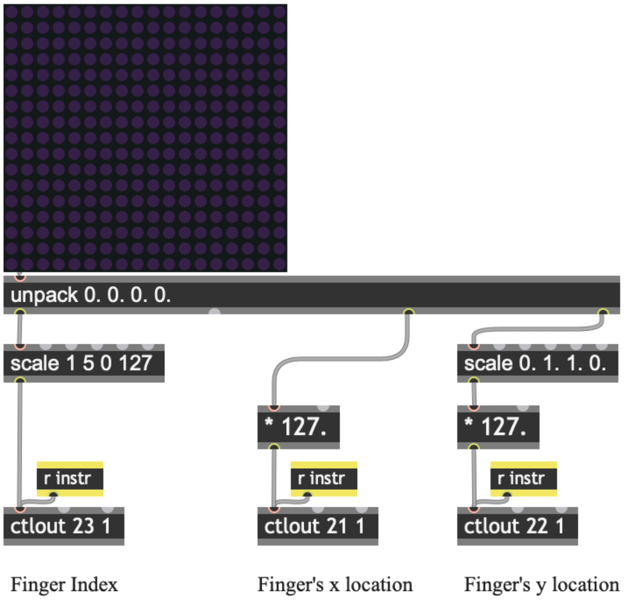


Figure 26. Three data scaling applied in the software layer

The third layout combined the toggle and multitouch objects that together provided both on-off (button) and ongoing control (fader) functionality. The last layout, which was the same as the second layout, used only the multitouch objects.

The transformed data generated from the software layer were sent as MIDI messages to Kyma via KymaConnect. These MIDI messages controlled musical parameters where Kyma generated sonic events based on its sound-producing algorithms.

Musical Challenges and Opportunities

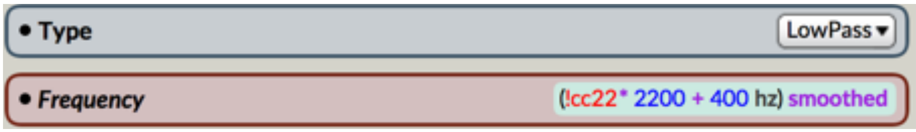
Because the location of my fingers could be seen by a built-in LED illumination, the visual aid assisted the audience in tracking how I am interacting with the controller. The LED illumination tightly connected the relation among me (performer), the controller, and the sonic results.

The controller had a soft and deep tactile surface that made my interactions with the controller more intuitive. The haptic feedback of the controller was very helpful in training my muscle memory. The softness and malleability of the Lightpad Block's performative surface helped develop my kinesiological understanding about how I should touch and musically operate the device. However, from the performance observability standpoint, how hard I pressed on the interface was not very obvious. To compensate for this inadequacy in Z-axis expressivity, the built-in LED illumination displayed different pressure levels by changing the size of LED circles on the surface.

At the same time, the Lightpad Block's pocket size limits the performance options. To expand the capabilities of this small device I created four layouts that represented reconfigurations of the operational mode of the Lightpad Block, and which I could recall at my convenience. There were various ways of operating the controller such as depressing and lifting multiple fingers and moving multiple fingers left and right, or up and down.

Data Mapping Strategies

In *Pulsing*, data scaling and offsetting were the two most common data-mapping procedures applied. For example, when continuous controller data streams were created by moving my fingers along the X-axis, the data received by Kyma was assigned to control the cutoff frequency of a lowpass filter. To make continuous 0-1 data usable to control the cutoff frequency of a filter the data needed to be scaled to a much larger range. I accomplished this scaling by multiplying all numbers in that data stream by 2200. This wider cutoff frequency range made the timbre changes produced by the lowpass filter easier to perceive. After that, the whole frequency range was offset by adding 400 Hz. The original recording used in this signal flow was based on a voice of a female singing, which did not contain many low-frequency components. Raising the lowest cutoff frequency to 400 Hz assured the lowpass filter made perceivable changes in the relatively high-frequency range of the female singing recording. The final cutoff frequency range was from 400 to 2600 Hz (see Figure 27).

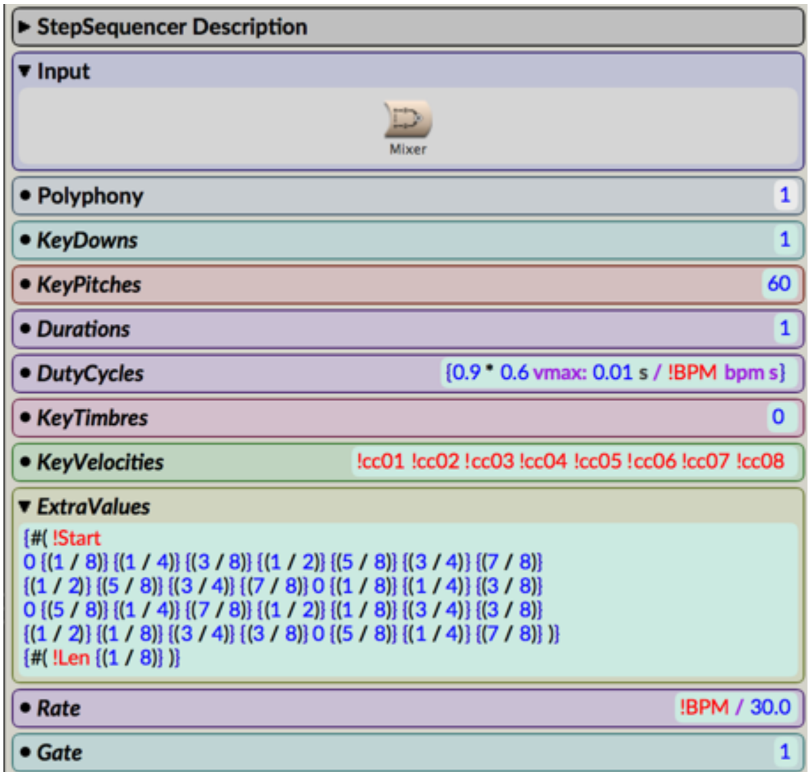


• Type: LowPass

• Frequency: (!cc22* 2200 + 400 hz) smoothed

Figure 27. Data scaling and offsetting of lowpass filter cutoff frequency

The *StepSequencer* was one of the main Kyma Sound objects employed in this composition. This Sound object was used to create one-to-one mappings between the interface and the sound-producing algorithm. As Figure 28 shows, the ExtraValues parameter field indicated the starting point and length for each step. The total thirty-two steps shown in the ExtraValues parameter field corresponded to the eight MIDI messages in the KeyVelocities, from cc01 to cc08. In other words, each of the continuous control messages (cc) controlled four steps from the thirty-two-step sequence. For example, when cc01, a toggle, was active or on, the velocities of 1st, 9th, 17th, and 25th steps were activated. When cc01 was off, the four steps were de-activated.



```
▶ StepSequencer Description
▼ Input
• Polyphony        1
• KeyDowns         1
• KeyPitches       60
• Durations        1
• DutyCycles       {0.9 * 0.6 vmax: 0.01 s / !BPM bpm s}
• KeyTimbres       0
• KeyVelocities    !cc01 !cc02 !cc03 !cc04 !cc05 !cc06 !cc07 !cc08
▼ ExtraValues
{#( !Start
0 {(1 / 8)} {(1 / 4)} {(3 / 8)} {(1 / 2)} {(5 / 8)} {(3 / 4)} {(7 / 8)}
{(1 / 2)} {(5 / 8)} {(3 / 4)} {(7 / 8)} 0 {(1 / 8)} {(1 / 4)} {(3 / 8)}
0 {(5 / 8)} {(1 / 4)} {(7 / 8)} {(1 / 2)} {(1 / 8)} {(3 / 4)} {(3 / 8)}
{(1 / 2)} {(1 / 8)} {(3 / 4)} {(3 / 8)} 0 {(5 / 8)} {(1 / 4)} {(7 / 8)} )}
{#( !Len {(1 / 8)} )}
• Rate             !BPM / 30.0
• Gate             1
```

Figure 28. Kyma step sequencer

Sonic Motives

My concept of motive was employed in *Pulsing* as three distinctive sonic motives. These three sonic motives occur over the course of *Pulsing* whose five-section structure can be described as: A-B-C-D-A'-B'. The final section is recapitulatory of both the initial A and B sections. The motive I identify as voice sequencer occurred in all sections of the composition while the motive I labelled as voice grain occurs in the final four sections and the xiao motive based on the sound of the ancient Chinese bamboo flute occurred in the first four sections. Below I discuss each motive and describe how it is developed over the five sections of the piece.

The primary sound material used in this composition was based on the recognizable quality of a female singing Mongolia Chinese folk music.[27] By using the *StepSequencer* Sound object in Kyma, the recording of female singing was divided into many small segments as a motive for this composition. These small segments form distinct rhythmic patterns through various step combinations. Another motive, voice grains, derived from the voice-sequencer motive. However, unlike the rhythmic voice-sequencer motive that the short singing sounds were played in a constant tempo, the voice-grains motive did not involve any rhythmic elements. By grouping these short singing sounds, the voice-grain motive was able to form a sustained sound effect. As complementary material, a recording of a xiao was employed and subjected to sound processing techniques completely different from the primary sound source.[28]

Voice-Sequencer Motive

Pulsing began with a thirty-two-step sequence of the voice sound. The original nine-second long singing recording was divided into eight sections of equal duration for the beginning voice-sequencer motive (see Figure 29). Each of the sections was about 1.125 seconds. Table 1 shows how the thirty-two-step sequence arranged the eight equally divided sound segments into four groups and was triggered by eight toggles. This thirty-two-step sequence of the voice sound was triggered and performed based on the

---

[27] The female singing audio file was taken from the *Legend of China* sample library, Discovery Sound, 1999.

[28] The xiao audio file was taken from the *Legend of China* sample library, Discovery Sound, 1999.

first Lightpad Block layout which contained only buttons. By interacting and manipulating the toggles of the first layout, I was able to create varied sonic results based on the differing on/off combinations of toggles. This voice-sequencer motive also occurred with slight delay effects.

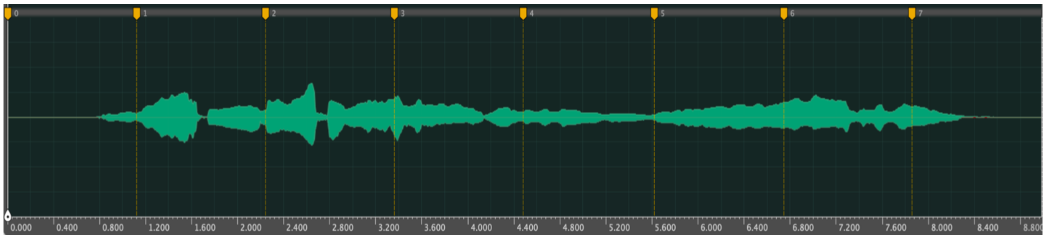


Figure 29. Eight equal-length sections of the singing recording

Table 1. Thirty-two-step sequencer[29]

Toggle / Group	1	2	3	4	5	6	7	8
1	0	1 / 8	2 / 8	3 / 8	4 / 8	5 / 8	6 / 8	7 / 8
2	4 / 8	5 / 8	6 / 8	7 / 8	0	1 / 8	2 / 8	3 / 8
3	0	5 / 8	2 / 8	7 / 8	4 / 8	1 / 8	6 / 8	3 / 8
4	4 / 8	1 / 8	6 / 8	3 / 8	0	5 / 8	2 / 8	7 / 8

In section B (2:06-3:47 of the supplemental video), the voice-sequencer motive was modified in three respects: the narrowing of the sound source, the reduction in the number of steps triggered, and the narrowing of the frequency spectra of the sound (through filtering). Since the second section incorporated more sound materials, and the voice-sequencer motive became the secondary element, the input of the voice sequence was not sounded, and only its delayed sound in four steps was involved in order to yield more musical opportunities to other sounds. The subtractive synthesis technique was

---

[29] The fractions indicate the starting point of the recording. For example, 4/8 at the beginning of the second row means the sound for this step plays from 4.5 seconds, 4/8 multiply 9 seconds, of the original singing recording.

utilized after the sequencer. By moving one finger up and down on the second layout of the interface, I was able to control a cutoff frequency of a lowpass filter which manipulated the timbre changes of the voice motive between a muffled and bright sound.

In section C (3:47-5:23 of the supplemental video), the climax of the piece, the voice-sequencer motive became much more compact and intense. Different from the previous two step-sequencer motives, the motive in this section was changed in three respects that included the number of layers, the number of sequencer steps, and the length of each step. The voice-sequencer motive contained three layers in section C. The first layer was a 7-step sequence that divided the original singing recording into five segments, and each of which was approximately 1.8 seconds in duration. The five segments were triggered in the order of 0, 1, 2, 3, 4, 3, 4. The second layer was a 7-step sequence in which the singing recording was divided into twelve segments with each segment being about 0.75 seconds in duration. These segments were triggered in an order of 1, 7, 5, 4, 11, 4, 11. The third layer was a thirty-two-step sequence based on a bass drum sound recording that only triggered the first step of the sequence. Although these three layers were in distinct step-sequences with different triggering orders, this unalignment created complex rhythmic combinations.

After the climax of the composition, I reduced the layers in which the voice-sequence motive occurred to one and kept the duration for each sequencer step at 0.75 seconds (5:23-8:13 of the supplemental video). Instead of following previous sequence patterns, the voice-sequence in this section contained thirty-two steps in a completely re-ordered sequence.

The composition concluded with a restatement of the main characteristics of the beginning and second voice-sequencer motives. The voice-sequence in last section, 8:14-end of the supplemental video, included thirty-two steps and applied the same triggering order as the beginning motive. Borrowing the subtractive synthesis idea from section B, the voice motive gradually revealed itself in a wider frequency range, compared to the section B, where the frequency changed from 161 to 3961 Hz.

Voice Grains Motive

Similar to the voice-sequencer motive, the voice-grains motive also divided the singing recording into short sonic segments, but the arrangement of these segments were arrhythmic. The voice-grains motive sometimes presented the individual grains of the original sound, separate from other sounds, and at the times presented the sonic-grains grouped into clouds to create a sustained sound.

Through controlling sonic parameters of granular synthesis algorithm such as density, duration, frequency, and the maximum number of grains that could occur simultaneously, the voice-grains motive was developed in section B. Individual grain sounds were first presented at the beginning of section B, while sustained grain sounds were presented in the second half of section B. Both the individual grain and sustained grain sounds were articulated with the Gaussian wave shape envelope (see Figure 30, left side). By increasing the number of grains and extending their duration, the sustained version of the voice-grain motive produced choir-like sonic results.

Section C mainly focused on sustained manifestation of the grain-motive. Instead of demonstrating a gentle and choir-like timbre, the voice-grains in this section were presented in a manner that created noisy and richer timbres. Each one of the grains was formed by four articulations of a reverse exponential envelope that contained multiple sharp attacks and gentle decays (see Figure 30, middle image). The grains with multiple sharp attack envelopes produced noisy sonic results with the characteristics of voice being barely perceived.

Section D primarily explored the grain motive as individual grain sounds. Unlike the noisy timbre contained in the previous section, the grains in this section retained the sonic characteristics of the singing voice. The grain envelope used for the grains in this section contained only a single iteration of the reverse exponential shape generating a less noisy timbre (see Figure 30, right side).

Like the voice-sequencer motive, the voice-grains motive also restated the beginning materials in the final section. The high-pitch individual grains and choir-like grains, first presented in section B, were help articulate the end of the composition.

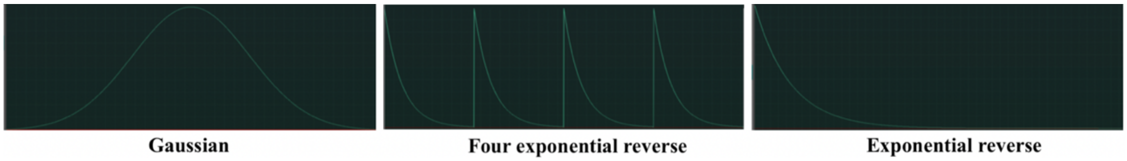


Figure 30. Three grain envelope shapes used in the voice-grains motive

Xiao Motive

A third sonic motive used in *Pulsing* was based on a recording of a xiao playing four notes. The xiao's sound was modified through analysis and resynthesis techniques in two ways. In the A, B, and recapitulatory sections of the piece, the analysis and resynthesis technique was realized through Kyma *SumOfSines* Sound objects. The *SumOfSines* Sound object resynthesizes a spectral analysis using sine waves. Because the analysis and resynthesis technique provided independent control of playback speed and pitch level, the original two-second xiao recording was extended up to 500 seconds and transposed to lower frequency ranges. In Section C and D, the xiao sound was resynthesized through Kyma's *CloudBank-Resynthesis* Sound object. Similar to the *SumOfSines* Sound object, the *CloudBank-Resynthesis* Sound object also made the frequency and playback speed independent. However, the *CloudBank-Resynthesis* represented the xiao sound by sustained groups of grains. In *CloudBank-Resynthesis*, the waveform inside each grain and the envelope shape on each grain could be specified. By assigning the exponential reverse envelope to the buzz waveform for each grain, a crispy and clicky sound effect was produced based on the xiao recording in section C and D (3:47-6:38 of the supplemental video).

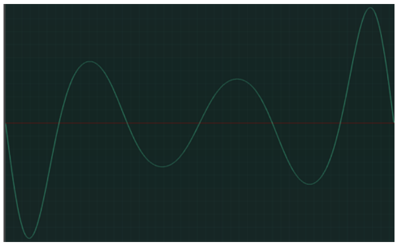


Figure 31. Buzz waveform

Performance Techniques

The performance technique employed in this composition included 1) pressing the toggles on the Lightpad Block layouts by fingers, which generated on-off data messages to turn on and off various sound samples, 2) moving fingers in the X and Y spatial dimension across the Pad, which generated continuous data messages to control ongoing musical parameters such as frequency, timbre, and amplitude in real time, and 3) touching fingers to toggles to start sound events and moving fingers in two spatial dimensions to manipulate musical parameters in real time. The first section of the piece highlights playing the instrument by touching the surface. The second section requires moving fingers across the Pad. The initial sequence section only contained toggles which challenged my sensibilities in the creation of various rhythms and my response to these rhythms. The continuous data produced in the second section allowed me to control the cutoff frequency of a filter and change the timbre of the human voice in real time by moving my fingers in the Y dimension. The third section, where the climax of the composition was reached, combined the use of toggles and ongoing controls as the regulatory mechanisms. This combination restated the primary performance techniques presented earlier in the composition. Following the climax in this section, the final two sections returned to having separations between the two different types of controls—ongoing controls occurring in the penultimate section and toggles dominating the final section.

CHAPTER III.4

HUADAN'S WHISPER

Overview

Huadan's Whisper is an interactive composition for Leap Motion, custom software created in Max, and Kyma. The main sound materials were based on audio recordings of bells, the Chinese instrument pipa, and a singing phrase from Huadan, a female role in Chinese opera.[30] By physically engaging the invisible three-dimensional performance area provided by the Leap Motion, position data of the two hands were sent to Kyma to control its sound-producing algorithms, and I was able to control musical parameters in real time. Through this interactive composition the audience was encouraged to travel through multiple sonic worlds.

Design and Implementation of the Data-driven Instrument used in *Huadan's Whisper*

The data-driven instrument used in this composition was comprised of three parts: 1) the Leap Motion controller that output data when I operated it, 2) Max that analyzed the initial data coming from the Leap Motion and remapped or created replacement data based on the analysis, and 3) Kyma that received the modified data and responded to it by producing the desired sonic outcomes.

---

[30] The pipa and Huadan audio files were taken from the *Legend of China* sample library, Discovery Sound, 1999.

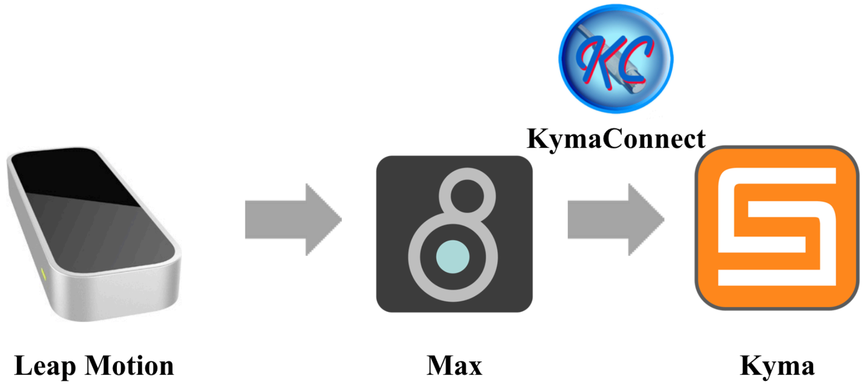


Figure 32. The overall structure of the data-driven instrument used in *Huadan's Whisper*

The Leap Motion is a computer hardware sensor device that supports the sensing hand and finger position in open physical space as its method to input data into a computer.[31] This touchless, intangible aspect of the Leap Motion makes it very different from many other controllers. By detecting the distance between the controller and the hands above it, the optical sensors and infrared light inside of Leap Motion can output many simultaneous data streams such as the X, Y, and Z spatial position of both hands, to name only several of the data streams that flow from the Leap Motion.[32]

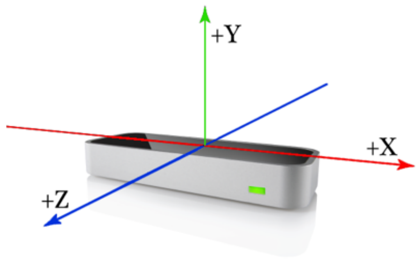


Figure 33. Three-dimensional control of the Leap Motion Controller

---

[31] Daniel Bachmann, Frank Weichert, and Gerhard Rinkenauer, "Evaluation of the Leap Motion Controller as a New Contact-Free Pointing Device," *Sensors* 15, no. 1 (2014): 214-233, https://doi.org/10.3390/s150100214.

[32] "API Overview," Leap Motion, accessed February 16, 2020, https://developer-archive.leapmotion.com/documentation/v2/csharp/devguide/Leap_Overview.html.

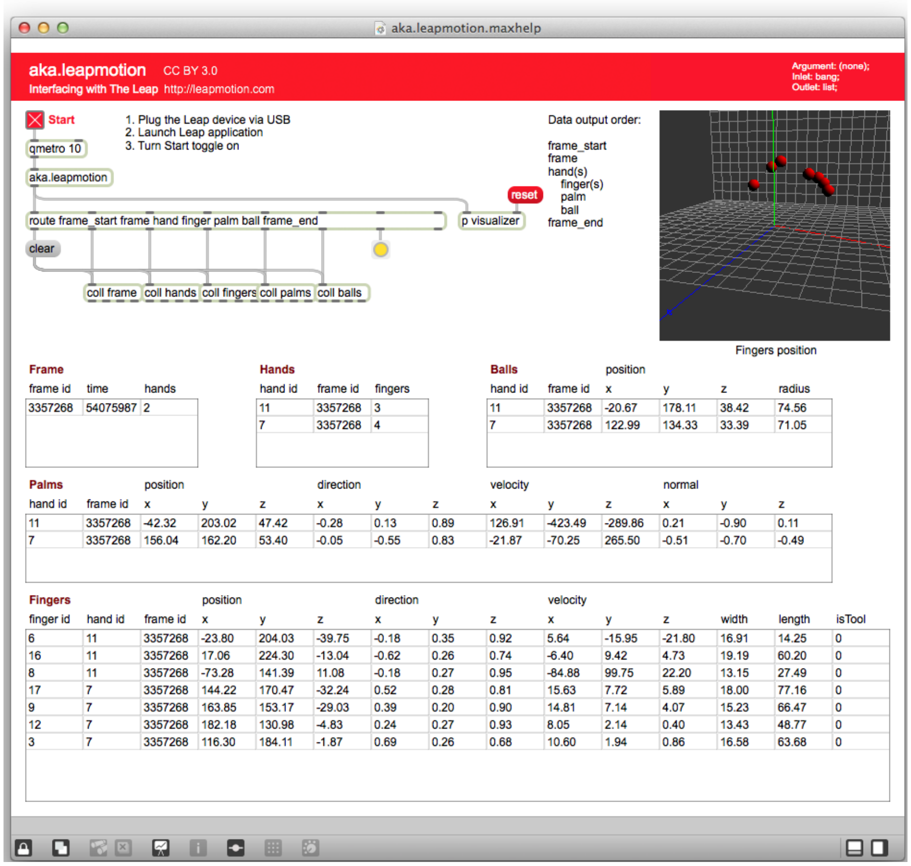
Through an external object *aka.leapmotion* developed by Masayuki Akamatsu,[33] Max acquired a number of data streams from the Leap Motion. This group of data streams included information about the number of hands in the performance area, the number of fingers extended on each hand, and the X, Y, and Z positions of each hand (see Figure 33). The Max patch used in this composition is derived from Dr. Iris Wang's work found in her composition *Magic Fingers.*[34] According to Wang, when "two hands are both continuously controlling data inside of the interaction area," Max sometimes confuses which hand is which.[35] To resolve this problem, Wang used a re-indexing procedure in Max to align data with the hand that is generating it. To represent the substantial amount of data provided by Leap Motion, Wang also converted the original data into high-resolution MIDI before transferring the data to the next layer of the instrument. Through KymaConnect, Kyma received these high-resolution MIDI messages and realized its sound-production based on its additional data mappings. I followed a similar procedure.

---

33. Masayuki Akamatsu, "Aka.objects," Akalogue, accessed February 16, 2020, http://akamatsu.org/aka/max/objects/.

34. Chi Wang, "*Magic Finger*," https://vimeo.com/95141260.

35. Wang, "Narrative to Action," 38.



1. Plug the Leap device via USB
2. Launch Leap application
3. Turn Start toggle on

Data output order:

frame_start
frame
hand(s)
  finger(s)
  palm
  ball
frame_end

Frame

frame id	time	hands
3357268	54075987	2

Hands

hand id	frame id	fingers
11	3357268	3
7	3357268	4

Balls

		position			
hand id	frame id	x	y	z	radius
11	3357268	-20.67	178.11	38.42	74.56
7	3357268	122.99	134.33	33.39	71.05

Palms

		position			direction			velocity			normal		
hand id	frame id	x	y	z	x	y	z	x	y	z	x	y	z
11	3357268	-42.32	203.02	47.42	-0.28	0.13	0.89	126.91	-423.49	-289.86	0.21	-0.90	0.11
7	3357268	156.04	162.20	53.40	-0.05	-0.55	0.83	-21.87	-70.25	265.50	-0.51	-0.70	-0.49

Fingers

			position			direction			velocity					
finger id	hand id	frame id	x	y	z	x	y	z	x	y	z	width	length	isTool
6	11	3357268	-23.80	204.03	-39.75	-0.18	0.35	0.92	5.64	-15.95	-21.80	16.91	14.25	0
16	11	3357268	17.06	224.30	-13.04	-0.62	0.26	0.74	-6.40	9.42	4.73	19.19	60.20	0
8	11	3357268	-73.28	141.39	11.08	-0.18	0.27	0.95	-84.88	99.75	22.20	13.15	27.49	0
17	7	3357268	144.22	170.47	-32.24	0.52	0.28	0.81	15.63	7.72	5.89	18.00	77.16	0
9	7	3357268	163.85	153.17	-29.03	0.39	0.20	0.90	14.81	7.14	4.07	15.23	66.47	0
12	7	3357268	182.18	130.98	-4.83	0.24	0.27	0.93	8.05	2.14	0.40	13.43	48.77	0
3	7	3357268	116.30	184.11	-1.87	0.69	0.26	0.68	10.60	1.94	0.86	16.58	63.68	0

Figure 34. External *aka.leapmotion* object developed by Masayuki Akamatsu

Musical Challenges and Opportunities

Like the Theremin, a performer can control the Leap Motion without touching it. Since the primary features of the Leap Motion are two monochromatic IR cameras and three infrared LEDs, the Leap Motion could not recognize my hands and fingers when my hands and fingers were too close or too far away from it. The effective range of the Leap Motion controller extended from approximately 25 to 600 millimeters above the device.[36] When performing, I needed to be aware of the area the Leap Motion "saw". This area is depicted in Figure 35.

---

[36] "API Overview."

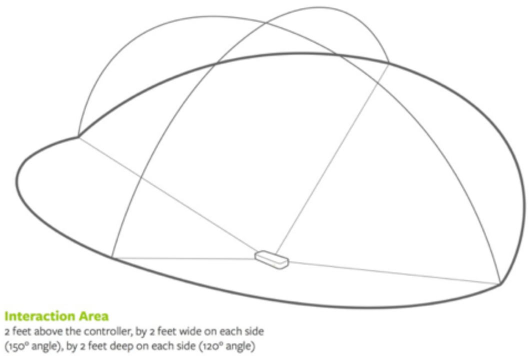


Figure 35. Three-dimensional performance area offered by the Leap Motion controller (image credit: *Leap Motion*).[37]

Unlike some of the controllers that output only a few data streams, the Leap Motion was able to output more than fifty simultaneous data streams including detection of each finger's presence and position, each hand's presence and position, and their respective movement direction and velocity in three spatial dimensions. These various data streams were very useful and performable for controlling musical features such as timbre, pitch, and loudness. However, there was no true button-like function to turn on and turn off sonic events. To solve this problem, thresholds needed to be set to trigger sound events whenever particular numerical values were breached. By setting up different thresholding mechanisms, the continuous data of my hands position also served in button-like functions. A simple example of this occurred when the data from the X-axis contains a value greater than a predetermined numerical threshold, a button function was triggered. When the data contains values less than this threshold the button was turned off.

Another challenge of this intangible interface was that the data controlled by the hand position was difficult to accurately maintain and hold at a specific numerical value.

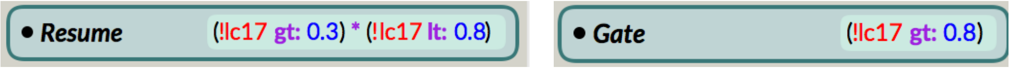
---

37. Colgan, "Leap Motion Controller Work."

There were several reasons for this. First of all, the range that the Leap Motion could detect was relatively small, compared to controllers such as the Gametrak whose performative space could be extended up to ten feet. Even a subtle moving or jiggling of hands impacted the data that Leap Motion output. Second, the optical sensors and infrared light used in the Leap Motion were very sensitive to light. Different light situations in performance settings impacted how the sensor and infrared light responded and challenged my muscle memory. I needed to adjust the performative actions to cooperate with the complicated light situation. Third, because the X, Y, and Z axes data were interconnected, when controlling one data stream, I also changed the other two at the same time. To avoid data streams affecting each other, most data mapping in the piece only used one stream at a time. If two data streams were desired to control musical parameters, one data stream from each hand solved this problem since the two data streams were isolated from each other. These two methods work very well at the beginning of the piece since sonic materials of the composition were not yet fully developed. With the development of the composition, for each hand, more and more data streams became involved to control various musical parameters. With multiple data streams, data smoothing helped to control data jitter that naturally occurred.

Data Mapping Strategies

There were many data mapping strategies employed in *Huadan's Whisper* including data scaling, data offsetting, and data smoothing. I used data scaling of MIDI continuous controller 17 when I communicated to Kyma if one, two, or zero hands are currently present in the Leap's performance area. Initially, the Leap simply counted hands yielding the possible values 0, 1, or 2. Those values were scaled to a range of 0 to 16383 to accommodate the full resolution of 14-bit MIDI. Kyma automatically scaled these values to 0, 0.5, and 1 to indicate whether no hands (0), one hand (0.5), or two hands (1.0) were present in the performance area. To accommodate the basic need of articulating how many hands were in the performance field, I created conditions in my mappings. In Figure 36, the first Capytalk expression determined if one hand was present in the performance area; the second Capytalk expression assessed if two hands were present in the performance area.



● **Resume** `(!lc17 gt: 0.3) * (!lc17 lt: 0.8)`

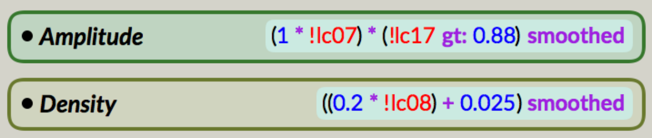
● **Gate** `(!lc17 gt: 0.8)`

Figure 36. Two ways I triggered events by mapping data in Kyma

Data smoothing was used to resolve the challenge of data jitter. Data jitter is random, unintended variation of data output from an interface. I would consider the Leap Motion to be a “jittery” device because it was impossible for a person to hold adequately still in one physical position so that the numbers output do not change. I used the smoothing operation within Kyma to solve this problem. For example, in Figure 37, I controlled the jitter of a two-dimensional fader (the X-axis and the other one Y-axis respectively) represented by the two data streams continuous controllers 7 and 8. These two controllers’ values were based on my left hand’s position in the performance area. I applied 0.1 seconds of smoothing to each new value to reduce the impact of jitter. The term *smoothed* in Kyma means that when a new value is received “a continuous linear interpolation from the previous value to the current value” of 0.1 seconds that is generated. The Capytalk for this procedure is shown in Figure 37.[38]

---

[38] Kyma, Online documentation inside of Kyma 7+, version 7.3.

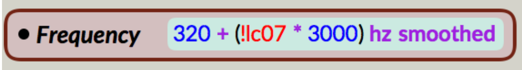


● Amplitude (1 * !lc07) * (!lc17 gt: 0.88) smoothed

● Density ((0.2 * !lc08) + 0.025) smoothed

Figure 37. Data smoothing

I also used data offsetting and scaling to better control musical parameters and produce desired sonic results. For example, when mapping the X-axis to the cutoff frequency of a lowpass filter, the data stream was first scaled by multiplying 3000 to affect a wider frequency range, and then offset by adding 320. As a result, I was able to control the cutoff frequency of a lowpass filter between the frequencies 320 and 3320 Hz.



● Frequency 320 + (!lc07 * 3000) hz smoothed

Figure 38. Data scaling and offsetting

Sonic Motives

My concept of motive was deployed as a primary compositional feature in *Huadan's Whisper* as three distinctive sonic motives, based on recordings of Huadan's voice, bell, and pipa. In Chinese opera, Dan describes general female characters. Among the other types of Dan such as Wudan which refers to martial women, Qingyi which refers to elite women, and Laodan which refers to older women, Huadan usually describes young female characters who are lively, unmarried, and vivacious.[39] Pipa, also called the Chinese lute sometimes, is a commonly used instrument in Chinese opera and has been played for almost two thousand years in China. These three sonic motives occur over the course of *Huadan's Whisper* whose four-section form can be described as: A-B-

---

[39] *Beijing Tourism*, "Chinese 'Dan,' A Form of Chinese Opera," (accessed February 16, 2020), http://english.visitbeijing.com.cn/a1/a-X9Y2KW46700B483C0FF417.

C-A’. Of the three, the bell and voice motives occur in all sections whereas the pipa motive occurs only in the B and C sections.

The bell and voice sonic referents were presented using shorter grain durations near the beginning of the composition. Because sounds did not show their distinct sonic features when presented with the extremely short grain lengths, these two motives gradually evolved to longer durations and revealed their own sonic characters with the development of the composition. Below I analyze each of the three motives and describe how they are developed over the composition.

Bell Motive

The bell motive consisted of three layers of bell sounds, each of which was distinctly spatialized with its own frequency range and shaped by distinct grain durations. The grain durations of the three layers depended on the amplitude modulation rate of the *Chopper* Sound object in which the faster rates produced shorter grains. The three layers utilized different chopped speed controls. The speed of the first layer bell sound was changed via algorithm based on a Capytalk expression, (((((1 repeatingTriangle: 5 s) * 2000) + 100) smoothed) * 4) bpm, which produced grain lengths ranging from 6 milliseconds to 138 milliseconds. The other two layers were controlled by one continuous data from the Leap Motion. The chopped rates changed upwards and downwards when I put my right hand above the Leap Motion and moved it horizontally. The last two layers were controlled inversely of each other. In other words, when my right hand moved from the left to the right, one layer of the bell sounds was produced at a descending rate and a second bell sound was produced at an ascending rate. The grains generated from these two bell layers were range up to 27 milliseconds in duration. The bell sound in these extremely short lengths lost its representative sharp attack and gentle decay envelope. At 1:45 of the supplemental video, the bell motive had a longer duration. By applying analysis and resynthesis techniques, the duration of the bell sound recording could be specified without affecting the pitch of the sound. Instead of extremely short durations, two sets of bell sounds, generated by two *Replicator* Sound objects, were played at a duration of two seconds. Although all bells were specified to be a two-second playback duration, the randomized delay time for triggering each one produced complex, unpredictable rhythmic patterns. The frequencies of the two sets of bell sounds were also randomized. Every time when bell sounds were triggered, they produced different pitches and rhythms due to the multiple triggerings of sounds. Associated with these triggering the two sets of bell sounds were given oppositional spatial placements. One set of bell sound was triggered when I moved my hand from the top to the bottom, and another set of bell sound was triggered when I moved my hand from the bottom to the top. These two sets of bell sounds were also in different spatialization placements. One of them was spatially presented in front, and another one was placed towards the rear.

At 2:53 of the supplemental video, the bell sound returned to the form in which it was initially presented with its duration being extremely short. However, rather than alternating between accelerandos and ritardandos, the bell motive only accelerated its rate of occurrence. This accelerating rate change produced bell sounds with grain durations that range from 3 to 75 milliseconds.

At 5:55 of the supplemental video, during the climax of the composition, the bell sounds played in its original recording format for six seconds. Borrowing the sound structure of the previous motive at B section, a group of bell sounds with randomized frequencies and triggering time represented unpredictable rhythmic patterns but the sound materials were longer in duration. In its original length, the bell sound fully demonstrated its sharp attack and gentle decay envelope.

Voice Motive

Similar to the bell motive, the voice motive gradually revealed itself, progressing from shorter to longer durations. However, different from the bell sound which was always in a percussive-like envelope, the human voice had varied envelopes, which changed from syllable to syllable, and richer timbre, which changed from moment to moment. The rich timbre of the human voice provided more opportunities for me to discover musical results based on only one sound recording.

The first voice motive consisted of two sparsely dense *SampleCloud* Sound objects with two *Chopper* Sound objects in a constant speed based upon two words, Zao and Bian, from a singing phrase from Huadan (see Figure 39). The two words were selected based on their timbral and melody qualities, literally the way that each part of the individual words sounded. The grains generated through *SampleCloud* and *Chopper* Sound objects were 80 milliseconds long. This short length made the two words hard to recognize since the grains only contained a portion of each syllable. Additionally, only part of the frequency spectra of Huadan’s voice were present through the subtractive synthesis technique. A lowpass filter at the end of the signal flow of this first voice motive only allowed frequencies below 1200 Hz to pass through. Although Huadan’s voice was unintelligible at the beginning of the composition because of the short duration of the audio file playback, short grain lengths, and partial frequency spectra, the beauty of Huadan’s voice is revealed later in the composition.

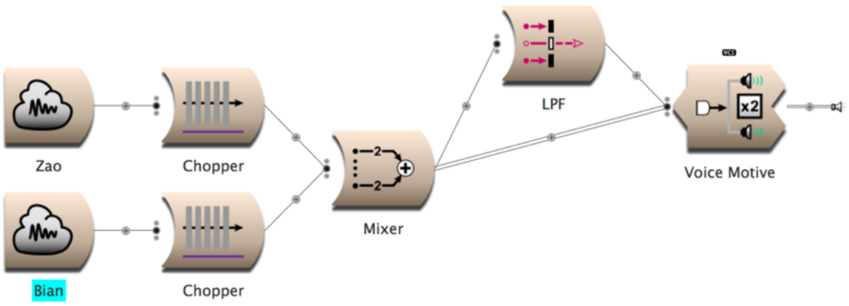


Figure 39. Signal flow of first voice motive

At 2:07 of the supplemental video, the voice motive, controlled by my left hand, used a sound structure similar to that of the voice motive, but explored a more extended segment of the recorded material (see Figure 40). Instead of only two words, a long recording of Huadan’s singing was articulated with this motive. The playing segment of Huadan’s singing was randomly determined every 110 milliseconds. Second, the grain duration generated by the *SampleCloud* Sound object was increased to 500 milliseconds. Finally, this voice motive had greater density and wider frequency ranges. The density of the *SampleCloud* ranged from 0.05 to 0.55. The cutoff frequency of the highpass filter ranged from 400 to 2000 Hz.

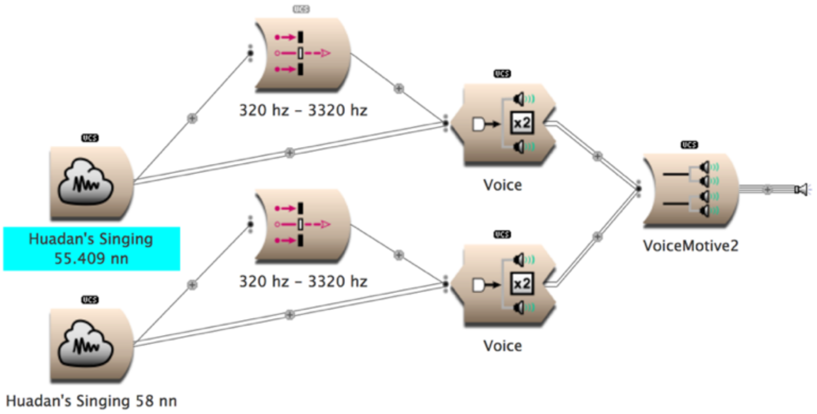


Figure 40. Signal flow of second voice motive

At 5:55 of the supplemental video, the voice motive was presented not as a sparse cloud of grains, but rather as a continuous sounding resynthesis that used the *SumOfSines* Sound object. Using the *SumOfSines* Sound object, I explored the details of Huadan's singing by scrubbing through the TimeIndex parameter—the playback location—of the resynthesized sound with my left hand across the X-axis. When I placed my hand in the center position of Leap Motion's active performance area, the beginning part of Huadan's singing was emphasized. When my left hand moved from the center position to the left position, later portions of Huadan's singing were emphasized. This back and forth action of my left hand not only controlled the playback location but also the envelope of the sound. This parametric support made the sound more vivid and realistic. Parametric support occurs as a natural outcome in much acoustic music performance and composition where timbral intensity, musical dynamics, and the instrument's pitch, together support a particular musical expression.

Pipa Motive

The pipa motive, a melody played on pipa, was developed in a different way from the previous two motives. Instead of dealing with grain lengths, the pipa sound was gradually revealed through the use of subtractive synthesis. The pipa motive appeared at 4:04 of the supplemental video and consisted only of lower frequencies of its spectrum. A lowpass filter removed all frequencies above 2100 Hz. However, in the next section, the pipa motive played its full spectrum without any filters.

Performance Techniques

The essential performance techniques in this composition included placing my hands in the active area of the performance space to initiate musical events, moving my hands to certain positions in order to breach thresholds and trigger sonic events, and moving my hands in the X, Y, and Z dimensions to control frequency, timbre, and other parameters. Triggering sound events by placing my hands in the active area and certain spatial locations can be seen at 1:45-1:54 and 6:08-6:54 of the supplemental video, respectively. In the first case, I moved my right hand upward and downward along the Y-axis to trigger two sets of bell sounds. In the second case, the Leap Motion counted the number of hands that I placed in the interactive area, and based on that number, one or two, different sounds were triggered by my hands.

Continuously controlling musical parameters such as filter cutoff frequencies, playback speed, amplitude, and pan positions in a continuous manner were the primary performative concerns in this piece. By moving my two hands within the Leap's three-dimensional hemispheric performance area, I was able to shape the moment-to-moment transformations of all sonic characteristics including timbre, pitch, loudness, spatial location, and the duration of sonic events.

The performance technique of moving my hands in the X, Y, and Z dimensions employed also corresponded to the unfolding process of the primary motives of the composition. When motives were restricted to narrow frequency spectra and used short grain durations, my hands were placed primarily in the middle area of the performance space, directly above the Leap Motion controller. As additional musical characteristics were introduced, my hands more actively moved in the extended of performance space, both vertically and horizontally. For example, at 2:07 of the supplemental video, my left hand controlled the voice motive which occupied only a narrow portion of the total frequency spectrum. The performance actions of my left hand were primarily located near to the Leap Motion controller at these moments (see Figure 41, left side). After thirty-five seconds, additional frequency components of the voice motive were introduced and the performance space of my left hand was also extended (see Figure 41, right side).

Figure 41. Performative actions of voice motive

CHAPTER III.5

DOTS ILLUSION

Overview

Dots Illusion is a multichannel real time interactive audio-visual composition for iPad, custom Max/MSP/Jitter software, and Kyma. I simultaneously controlled the sound and videographic domains. The basic motives in the audio and video dimensions were the sounds of water droplets and the image of dots (points). The image of the dot was the visual metaphor of the sound recording of a water droplet. The triggering, modification, and development of the sonic motive – the recording of a water droplet – was controlled in real time.

Design and Implementation of the Data-driven Instrument used in *Dots Illusion*

The performance interface I used in this composition was the Kyma Control application running on an Apple iPad. Kyma Control is a wireless multi-touch application specifically designed for the Kyma sound algorithm system and iPad.[40] The Kyma Control provided a number of powerful features including "a multi-touch pen/tablet controller, an auto-mapped and bi-directional mirrored controller/display of Kyma's Virtual Control Surface, a standard piano-style keyboard, and a Tonnetz pitch-space keyboard for experimenting with different pitch layouts."[41] In addition to the four visual controllers, accelerometer and compass-heading controls are also embedded in the application.

---

[40] "Kyma Control for iPad," Kyma, accessed April 30, 2019, https://kyma.symbolicsound.com/kyma-control-for-ipad/.

[41] Kyma, "Kyma Control for iPad."

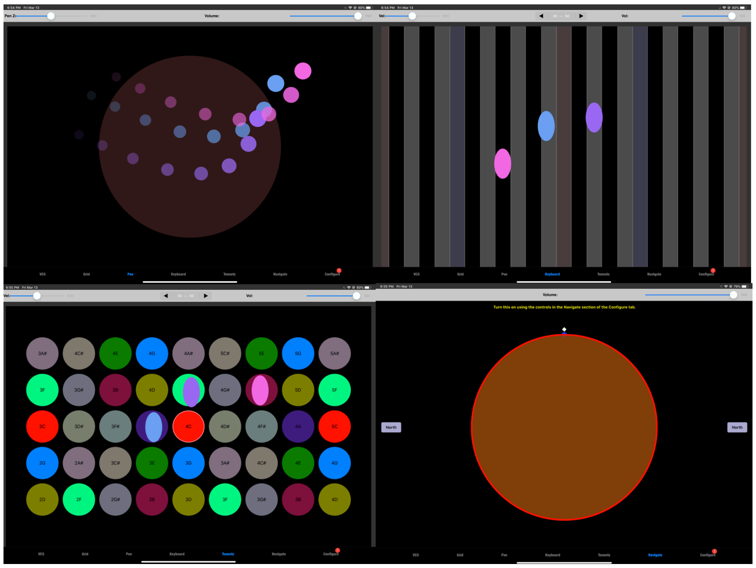


Figure 42. Four types of controllers in Kyma Control

The data-driven instrument used in *Dots Illusion* was comprised of four parts: 1) an iPad, which provided an operative interface for me to, 2) the Kyma Control application, which ran on the iPad, that passed on the data it received from me, 3) Kyma, which received and responded to Kyma Control by producing sonic results based on its sound-producing algorithms, and 4) Max/MSP/Jitter, which received MIDI data from Kyma and responded it by producing videographic content. This overall structure is depicted in Figure 43 below.

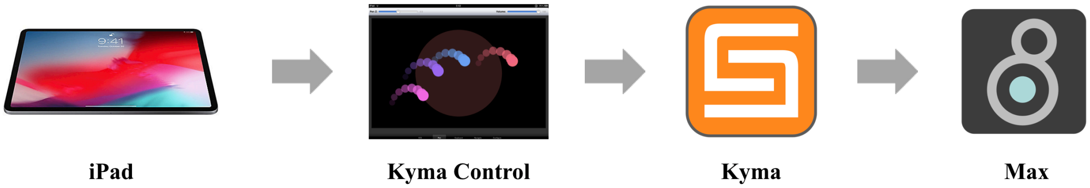


Figure 43. The overall structure of the data-driven instrument used in *Dots Illusion*

Because the fundamental motives in the audio and video dimensions were the sounds of water droplets and the image of dots shown on a black canvas, I decided to only use the multi-touch Pen tab from the Kyma Control application which has a similar visual motive, a single dot positioned in the center of its two-dimensional space. The multi-touch Pen tab was able to output data based on the position of up to sixteen fingers simultaneously placed on the iPad as continuous controller numbers from 5 to 14 on MIDI channel 1. Instead of employing MIDI messages from all fingers in their X, Y, and Z dimensions, I chose to use only the continuous data created by the touch of two fingers in the X and Y dimensions while still permitting touch events by additional fingers. The data streams generated from Kyma Control were passed through Kyma as MIDI messages to Max. The portion of the Max patch that received these data is shown in Figure 44. These data messages were mapped to control various videographic parameters, such as the size, location, and dimension of dots, to control the visual domain in real time.

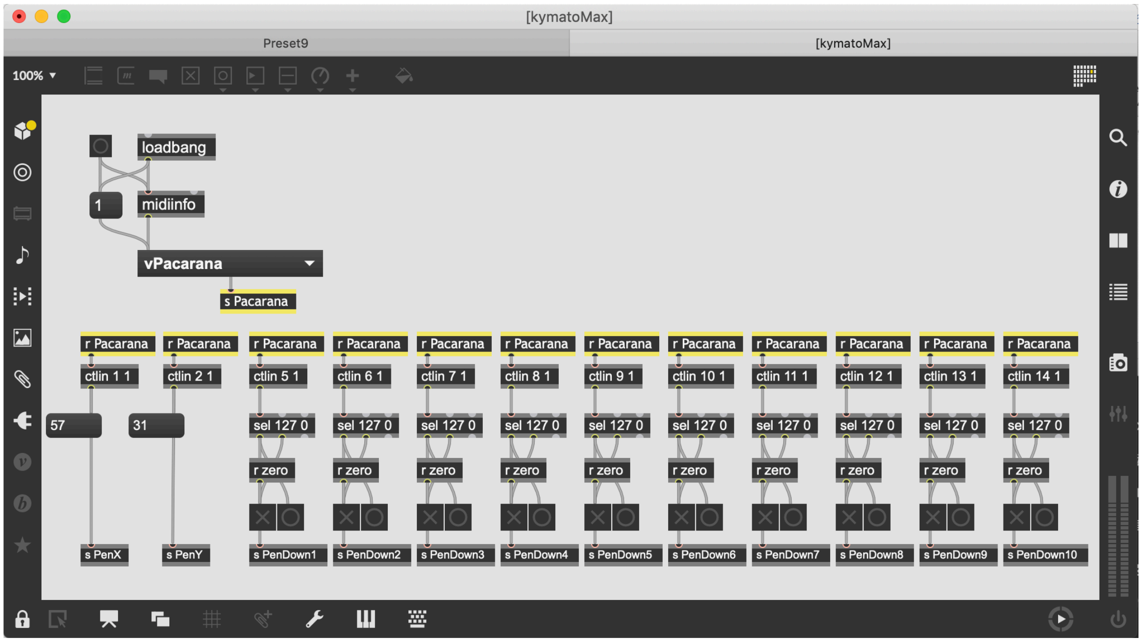


Figure 44. Data received from Kyma in Max

Musical Challenges and Opportunities

The Kyma Control application provided me with a rich array of data streams that I could use to control the sonic and videographic domains. Kyma Control's standard piano-style and Tonnetz pitch-space keyboards allowed me to explore possible pitch control with the different interfacial layouts. The accelerometer and compass-heading controls offered the opportunity to expand the performance space so that I could lift the iPad off a table surface and move it freely in space. The multi-touch Pen tab provided precise and stable data streams. Since the visual similarity between the interface layout and the videography generated through Jitter, I decided to confine my control by only using the multi-touch Pen without other types of controls (Figure 45).

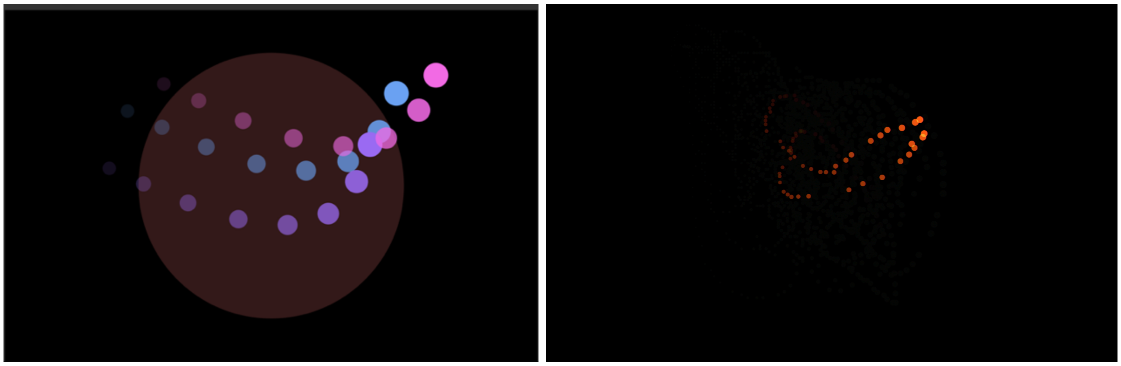


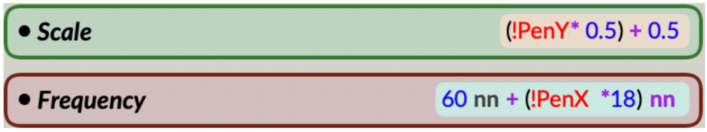
Figure 45. Multi-touch pen/tablet control layout (left) and videography generated through Jitter

Like the Wacom tablet, Kyma Control's multi-touch Pen tab provided both on-off and continuous control capabilities. These capabilities gave me the opportunity to start musical events and then control musical parameters of the events. The precision, stability, and separation of the continuous data streams that derived from the position of my fingers in the X and Y dimensions allowed me to obtain consistent musical results.

However, the iPad's flat interface presented a challenge with respect to making my performative actions observable. It was difficult for the audience to observe the details of my real-time performative actions given that the interface was flat and small. To address this problem, the video generated in Jitter helped me to better depict my performance where finger down messages were mapped to the appearance of dots, and the X and Y dimensional data streams were mapped to the dots' locations. However, it must be acknowledged, that any composition, of course, where a performer is creating and controlling videographic material has the potential conflict between splitting an audience's attention between action of the live performance and the videographic output. This piece is no exception.

Data Mapping Strategies

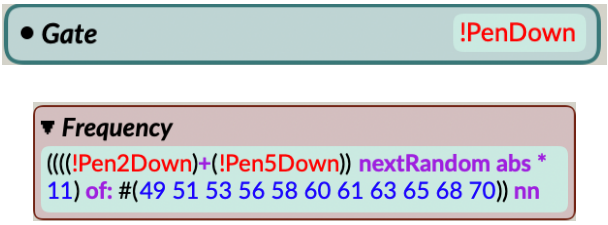
Similar to other compositions in my portfolio, data offsetting and scaling were the two primary data mapping operations employed in *Dots Illusion*. Figure 46 shows how the two PenX and PenY data streams were mapped to desired parametric targets. The PenY data stream, which reported my finger's position on the Y-axis, was first scaled to a range of 0.5 to 1 to control the amplitude of the sound. In the same figure, the PenX data stream, which reported my finger's position based on the X-axis, was scaled and offset to the range of MIDI note numbers 60 to 78 in order to control the sound's frequency.



- **Scale** (!PenY* 0.5) + 0.5
- **Frequency** 60 nn + (!PenX *18) nn

Figure 46. Data scaling and offsetting

The on-off data, generated when my fingers touched the tablet, were used to trigger sonic events (see Figure 47, left side). On-off data were also used to actuate an algorithmic procedure that selected a value from a pitch array. Figure 47 shows an example of this procedure. Here, each time the second or fifth finger touched the tablet a number from an eleven-value array was randomly chosen as the MIDI note number that specified the frequency of the sound.



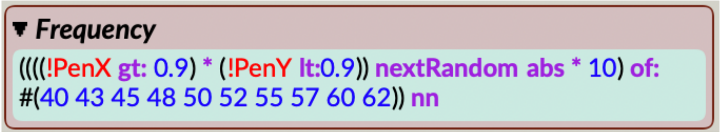
● Gate !PenDown

▼ Frequency

```
((((!Pen2Down)+(!Pen5Down)) nextRandom abs *
11) of: #(49 51 53 56 58 60 61 63 65 68 70)) nn
```

Figure 47. On-off data used in *Dots Illusion*

Data analysis and replacement was another mapping operation used in this composition. Figure 48 shows an example of this type of procedure as a Capytalk expression with multiple conditions. Only when both PenX data was greater than 0.9 and PenY data was less than 0.9, that a number from a ten-value array was randomly chosen as the MIDI note number that specified the frequency of the sound.

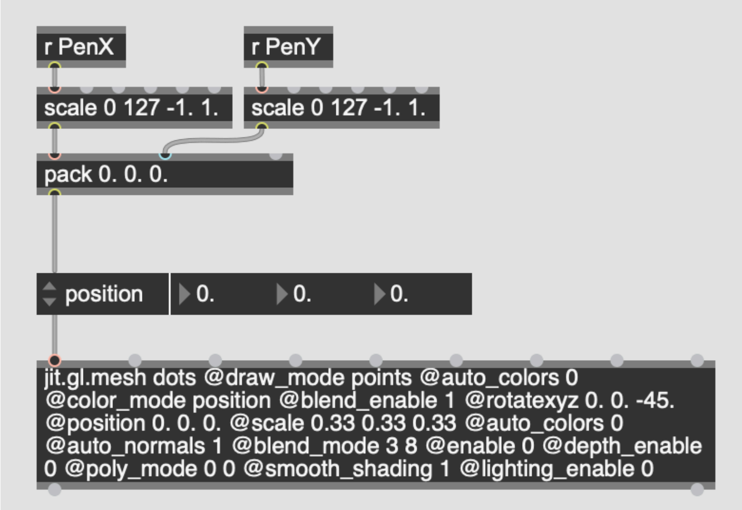


▼ Frequency

```
((((!PenX gt: 0.9) * (!PenY lt:0.9)) nextRandom abs * 10) of:
#(40 43 45 48 50 52 55 57 60 62)) nn
```

Figure 48. Triggering sound events based on continuous data streams

In addition to the data mapping strategies mentioned above, some of the same mapping operations were used in Max/MSP/Jitter to manipulate visual parameters. As Figure 49 shows below, data scaling was applied to both the PenX and PenY data streams to better control the position of the dots generated by the *jit.gl.mesh* object.



```
r PenX
r PenY
scale 0 127 -1. 1.
scale 0 127 -1. 1.
pack 0. 0. 0.
position 0. 0. 0.
jit.gl.mesh dots @draw_mode points @auto_colors 0
@color_mode position @blend_enable 1 @rotatexyz 0. 0. -45.
@position 0. 0. 0. @scale 0.33 0.33 0.33 @auto_colors 0
@auto_normals 1 @blend_mode 3 8 @enable 0 @depth_enable
0 @poly_mode 0 0 @smooth_shading 1 @lighting_enable 0
```

Figure 49. Data scaling in Max/MSP/Jitter

Figure 50 shows data smoothing applied in Max/MSP/Jitter to the number of dimensions for the *jit.gl.gridshape* object through the *line* object. Whenever my fourth finger touched the tablet the dimension parameter in Jitter changed gradually from 2 to 5 over 3000 milliseconds.

```
r PenDown4
2, 5 3000
line 1. 20
pak 0 0
dim 5 5
jit.gl.gridshape dots
@matrixoutput 1 @dim
128 128 @shape sphere
@color 0.8 0.4 0.2 1.
```

Figure 50. Data smoothing in Max/MSP/Jitter

Sonic Motives

My concept of motive in *Dots Illusion* concentrates on the use of a single sonic motive that occurs in all sections of the piece. This singular motive occurs over the course of *Dots Illusion* whose five-section structure can be described as: A-B-C-D-E. Below I describe how I developed the motive over the course of the composition.

Both the music and visual parts of *Dots Illusion* started with simple ideas that developed into complex structures. Musically, the sounds that developed throughout this composition were inspired by a recording of a water droplet. As the composition progressed, a recording of a kalimba sound was employed because its envelope was similar to that of the water droplet (see Figure 51). Visually, the dotted images generated by the *jit.gl.gridshape* and *jit.gl.mesh* objects acted as visual metaphors for the drops of water.

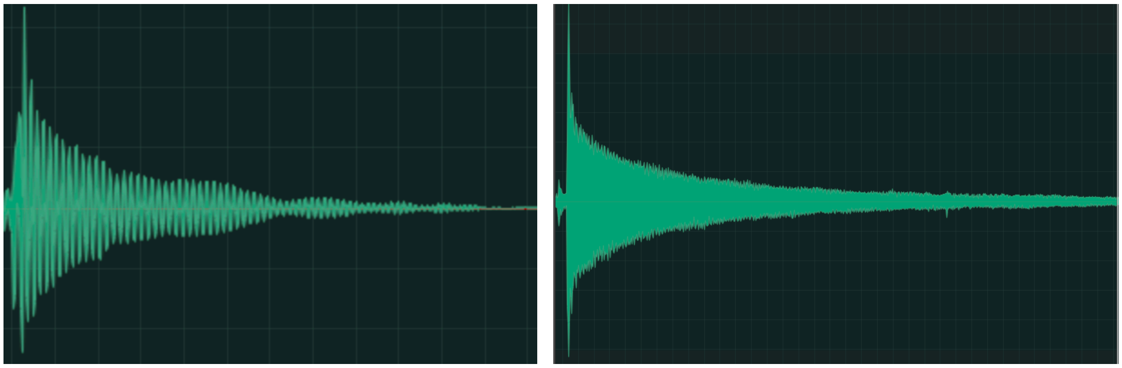


Figure 51. Waveform of one water droplet (left) and one note of kalimba (right)

The composition began with the sound of a water droplet followed by a cloud of sonic particles based on the same water droplet recording. When I placed my finger on the tablet, I triggered the sound of a water droplet and its grains. Simultaneously, FingerDown data was sent to Jitter, which produced several dots, initially presented in a straight-line segment that paralleled the drop of water sound and the cloud of sonic particles that followed it.

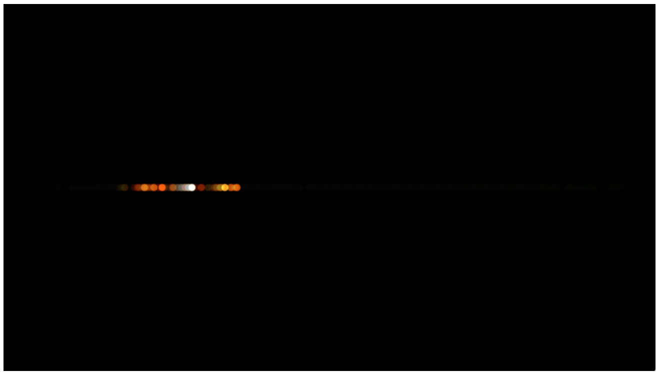


Figure 52. Videography in section A

In section B (2:53-4:06 of the supplemental video), the primary motive was realized through granular synthesis where many grains were realized one after another to form grain clouds. The frequency parameter of the grains was controlled in real time based on the Y-axis location data. The dots in the visual domain were extended to parallel the sustained granular sound. The dots gradually became more prominent while the lower frequency (not subaudio) sine-wave oscillator crescendoed (see Figure 53).

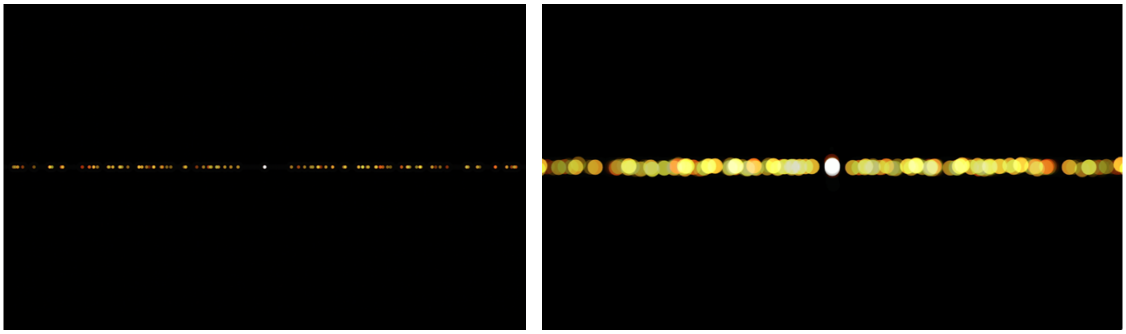


Figure 53. Videography in section B

In section C (4:06-6:12 of the supplemental video), the motive was realized as a sustained sound that was gated into small parts using amplitude modulation. The rate of this modulation was controlled by the Y-axis data stream. Corresponding to this sonic development of the motive, the dots of the visual domain began to disperse from their straight-line positions, and the visual line was disrupted (see Figure 54).

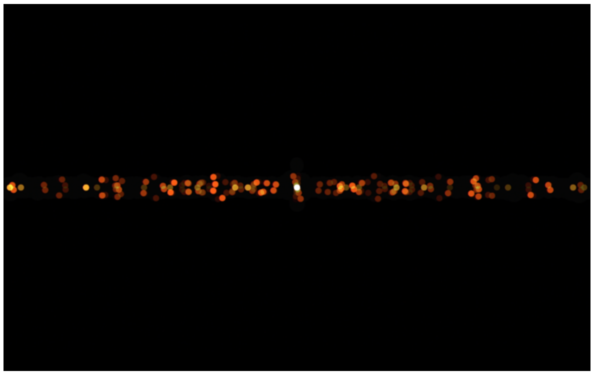


Figure 54. Videography in section C

The sound and image departed even further from their original states in section D (6:12-8:20 of the supplemental video). Instead of using the water droplet sound, a kalimba sound was used as the main sonic material in this section. This new timbre was transposed both upwards and downwards. In the visual domain, the dots were also transformed to correspond to the sonic changes. Instead of forming a single straight-line of dots, the dots were distributed in two-dimensional space in a variety of shapes that paralleled the movements of my finger moving on the control surface (see Figure 55).

Figure 55. Videography in section D

The granular sounds returned in the last section. The occurrence of the grains reached their highest density and widest frequency range at the beginning of this section (at 8:21 of the supplemental video). As the density declined and the frequency range of the grains narrowed, the piece gradually faded into silence. The dots in the visual domain paralleled the changes in the sonic domain by changing their size and density as well. The dots were presented with intensive density at the beginning of the section and then gradually became less dense and smaller.

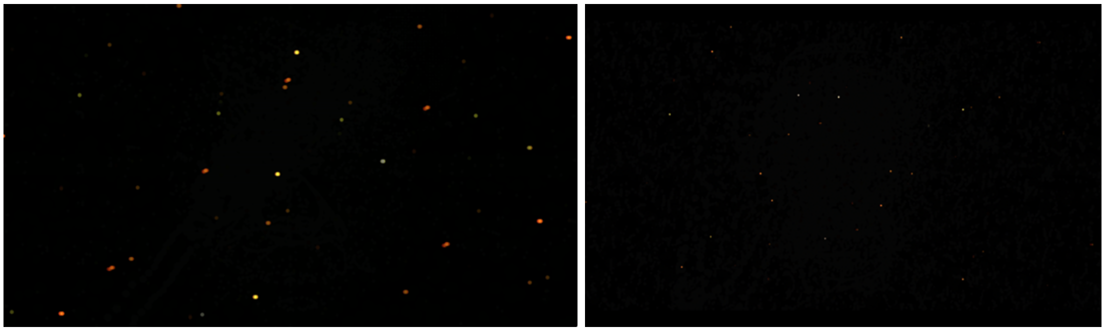


Figure 56. Videography in section E

Performance Techniques

The essential performance techniques in *Dots Illusion* were simple and straightforward. These techniques included touching my fingers directly to the tablet to start musical events and moving my finger or fingers across the iPad in the X and Y dimensions to produce ongoing control of frequency and playback speed. These two performance techniques were usually used one after another in which a sound event was first triggered and then modified in real time. Examples for these two basic techniques can be seen and heard throughout the composition. For example, at the beginning of the supplemental video I directly applied my fingers to the surface of the iPad to actuate the sound of a water droplet and its cloud of grains. Here I also controlled the frequency of the water droplet sound by positioning one of my fingers along the X-axis to produce the desired value. Another example of triggering sound events can be found in the supplemental video from 6:14-7:29, where I applied my index finger on the tablet to trigger the kalimba sound. Beginning at 6:48, two kalimba sounds were triggered when I applied both my index and middle fingers to the tablet.

CHAPTER III.6

OVERLAPPING STRINGS

Overview

Overlapping Strings is a composition for Gametrak controller, Symbolic Sound Kyma, and custom software created in Max. By performatively operating the Gametrak as an interface, I triggered and controlled various musical parameters in real time. The sound materials upon this composition was based include recordings of double bass, human voice, and bells. Because the strings of the Gametrak could be extended up to ten feet in length, a sizeable performance area was provided that encouraged full-body movements in the performance of the music.

Design and Implementation of the Data-driven Instrument used in *Overlapping Strings*

The data-driven instrument used in this composition was comprised of three parts: 1) a Gametrak controller that output data when I operated it, 2) Max, which analyzed the Gametrak data and created replacement data based on the analysis, and through KymaConnect, 3) Kyma, which received the modified data and responded to it by producing desired sonic results.

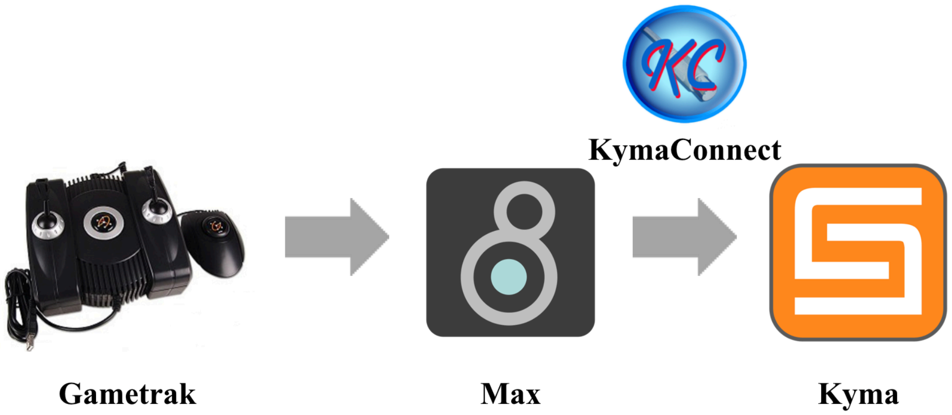


Figure 57. The overall structure of the data-driven instrument used in *Overlapping Strings*

The Gametrak is a three-dimensional game control system based on position tracking that was originally designed for a virtual golf video game.[42] Comprised of two thin retractable cables, Gametrak produced six data streams: X, Y, and Z axes data streams for each cable.

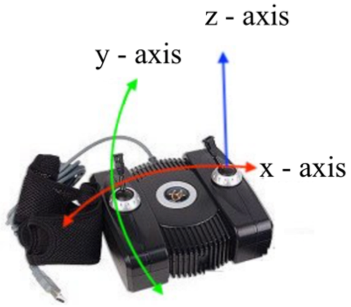


Figure 58. Three-dimensional control of the Gametrak Controller

When the Gametrak was connected to a computer through a USB cable, Max received data streams via the *hi* object. The initial data arriving at the *human interface* object included six data streams from 0 to 4095. Because Kyma normally receives 7-bit

---

[42] Adrian Freed et al., "Musical Applications and Design Techniques for the Gametrak Tethered Spatial Position Controller." In *Proceedings of the SMC 2009 – 6th Sound and Music Computing Conference*. Porto, Portugal, 2009.

MIDI values (0-127), the low-resolution MIDI cannot adequately represent Gametrak's 12-bit data. To make better use of the data, Max converted the values and sent them as 14-bit high-resolution MIDI to Kyma. By multiplying four to the Gametrak data in Max, the new data range of 0-16383 was created. A *line* object was used to reduce the differences between contiguous values of data that was generated by the Gametrak. The data scaling and smoothing achieved in Max are shown in Figure 59. The Max patch used in this composition was derived from Dr. Jeffrey Stolet's Max patch used in his composition *Lariat Rituals.*[43]

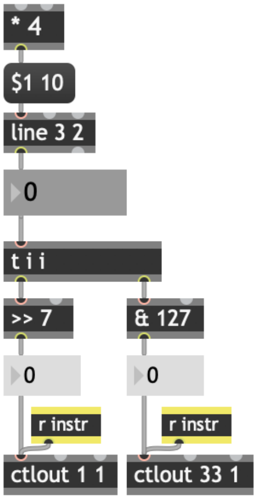


Figure 59. High-resolution MIDI in Max

Musical Challenges and Opportunities

Because the two cables of the Gametrak could be extended to about ten feet, the operational movements of the controller could be either big or small, gentle or dramatic, depending on the musical needs. In fact, there was almost no limitation in terms of the performative actions that could be used to control sounds of various sonic characteristics.

---

[43] Jeffrey Stolet, "*Lariat Rituals*," https://www.youtube.com/watch?v=quA3WxWCdzI.

As one example, I unfolded timbral changes over time by scrubbing through the Kyma *SampleCloud* Sound object's TimeIndex parameter by moving one of its cables from the left to the right very slowly in the Gametrak's X-dimension. As another example, I mapped data values created using the Gametrak's Y-axis to shift gradually through a range of frequencies.

However, the Gametrak strings only output continuous control data, and this shortcoming presented a challenge in starting sonic events. To resolve this problem, I created thresholds, which I could breach to generate directives to start musical events. Taking advantage of the large performance space the Gametrak provided, I employed large dramatic performative actions that were used to create numerical values that would breach thresholds and trigger sounds. An example of this can be observed at 1:42 of the supplemental video.

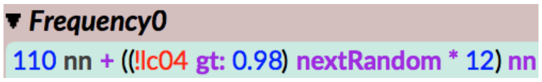
Data Mapping Strategies

One of the frequently used data mapping strategies in this composition was the transformation of continuous data streams into button-like functions. These button-like functions were set to trigger sounds or generate numbers for particular musical parameters. For example, when a continuous data stream was configured to meet certain conditions, sonic events were triggered in Kyma (see Figure 60).

● **Gate** (!lc03 gt: 0.3)

Figure 60. Using continuous data as button

Another example of using continuous data streams to create button-like functions is shown in Figure 61. In the frequency parameter field, when the X-axis of the right string was placed at the most right spatial position, my action triggered an algorithmic expression. The expression resulted in a randomized value between -12 and 12 that offset the frequency of the sound. The sound's final pitch ranged between MIDI note number 98 and 122.



▼ Frequency0

```
110 nn + ((!lc04 gt: 0.98) nextRandom * 12) nn
```

Figure 61. Data scaling and offsetting

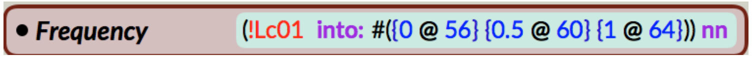
I achieved real-time control of musical parameters such as amplitude and frequency using a variety of techniques. The data scaling example in Figure 62 shows that the X-axis of the right string was assigned to control the amplitude of a sound, and the desired range of the amplitude was only 70% of the full range. The word "smoothed" at the end of the Capytalk expression was used to achieve a natural and smooth sounding sculpting of the amplitude-related data.

● Amplitude

```
( !Lc04 * 0.7) smoothed
```

Figure 62. Data scaling in Kyma

Another example of mapping the incoming data to a specific range is shown in Figure 63. In this Capytalk expression, (!Lc01 into: #({0@56}{0.5@60}{1@64})) nn, the range of values between 0 to 1 got linearly mapped to a range of values between 56 and 64. These new values (56-64) controlled the frequency parameter and represented values for MIDI note numbers.



● Frequency

```
(!Lc01  into: #({0 @ 56} {0.5 @ 60} {1 @ 64})) nn
```

Figure 63. Data scaling and offsetting in Kyma

Sonic Motives

My use of motive in *Overlapping Strings* is apparent in the use of two distinctive sonic motives. These two sonic motives occur in *Overlapping Strings* whose five-section form can be described as rather rondo-like: A-B-A’-B’-A”. In *Overlapping Strings*, the contrasting motives are essential in articulating the form where the grain motive is used in the three A sections and the rhythmic motive is used in the two B sections. Below I examine both sonic motives and describe how they are developed during the course of the composition.

Grain Motive

The grain motive consisted of a moderately dense *SampleCloud* built on the recording of a female speaking voice whose density gradually disintegrated to almost nothing over six seconds. The grain motive was presented at the beginning of the composition. This grain motive was sustained for five seconds and gradually faded out. The shape of the envelope on each grain was controlled by a Gaussian wave shape (see Figure 64). This grain shape was then motivically developed in the sections that followed.

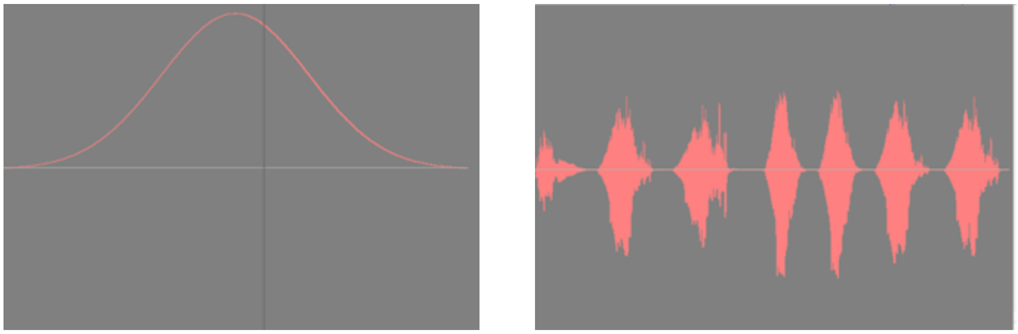


Figure 64. Gaussian wave (left) and grains with Gaussian wave envelope (right)

The first development of the grain motive appeared in section A and consisted of two layers (at 1:21 of the supplemental video). The first layer applied the same structure and sound material as the original grain motive except for the shape of the envelope. Instead of the Gaussian wave with a more gradual attack and release shape, grains in this layer used an exponential reverse wave shape, which possessed a sharp attack (see Figure 65). This change gave the grains a clearer transient that punctuated the mix, which made the sound easier to perceive. Instead of using the long phrase of female speaking, the second layer only used one syllable, Ti, from the recording. Because the syllable Ti had a noisy timbre at its beginning, the grains produced a unique sonic character. Therefore, when these two layers of grains with distinct sonic features were combined, both layers could be heard clearly.

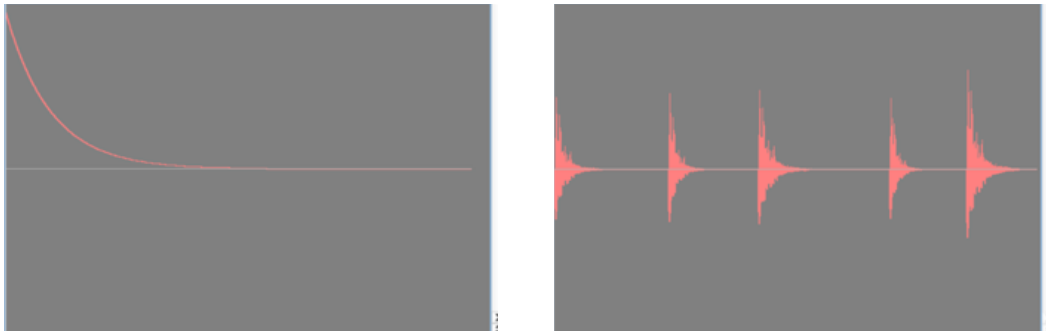


Figure 65. Reverse Exponential (left) and
grains with exponential reverse wave envelope (right)

The second development of the grain motive appeared immediately after the previous variation and was changed in two aspects: sound material and technique. A bell recording was used as the sound material. The bell sound had a similar envelope with the previous grain envelope, which also produced a sound with a sharp attack (see Figure 66). As for the technique, grains were not generated through the *SampleCloud* but with a *Chopper* Sound object. The bell sound was modified through the amplitude modulation at a very high rate and generated a series of grains extremely short in duration. The shape of the envelope on each grain generated by *Chopper* was a glottal pulse wave, a shape between the Gaussian and exponential reverse waves (see Figure 67). This in-between grain shape generated a musical outcome between the original and the first development of grain motive. This in-between shaped grain was sharper than the grains with the Gaussian wave envelope, yet gentler than the grains molded by the reverse exponential wave envelope.

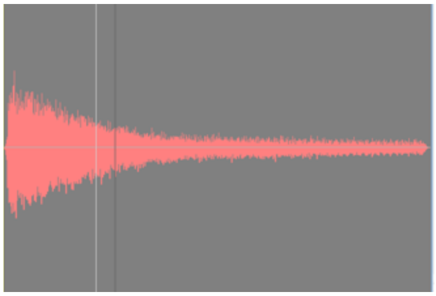


Figure 66. Bell sound

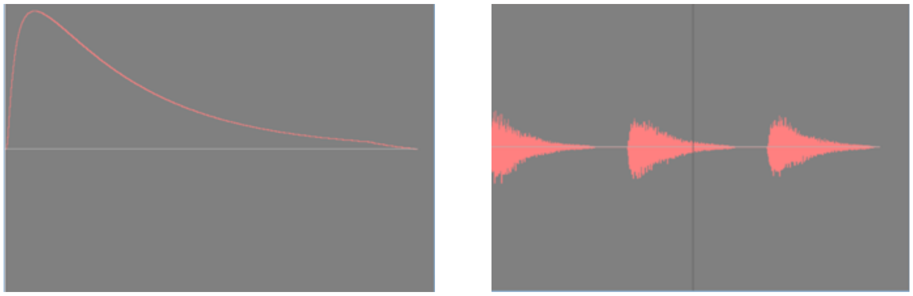


Figure 67. Glottal pulse wave (left) and grains with glottal pulse wave (right)

A third example of the development of the grain motive can be found in A' section (at 4:30 of the supplemental video). In this instance, two previously used sound materials, the voice and bell, were combined. Both sounds were panned randomly and controlled in real time. The left string of the Gametrak controller triggered and controlled the voice grain sound. The right string of the Gametrak controller triggered and controlled the bell grain sound. The frequency of the bell sound was controlled when the string was moved left and right. The performative action of moving the string from left to right generated a rising pitch of the bell grain sound. The opposite generated a descending pitch. The grains of the two sounds were generated by both granular synthesis and amplitude modulation with the *Chopper*. The grains were presented simultaneously through the *SampleCloud* and also unfolded over time with the *Chopper*.

At 7:47 of the supplemental video, the ending section of the piece, a third variation of the grain motive returned, but only the descending pitch feature of it remained. The chopped grain sounds were tightly triggered and layered over each other. By overlapping the unfolding chopped grains, the grains occurred simultaneously. The sound source for the restatement of this third motivic variation was changed to a double bass timbre, one of the main timbral sources used from the beginning of the piece. Here, the sound of the double bass was placed in the background. A few seconds later, the second variation of the bell grains returned in a lower frequency range that functioned as the ending material for the piece.

Rhythmic Motive

The first rhythmic motive in this composition appeared in section B which was comprised of a series of short sounds of equal length, based on the speaking voices and generated through amplitude modulation with Kyma's *Chopper* Sound object (at 2:03 of the supplemental video). Instead of gating the sound at a very fast rate to produce the grain motive, I utilized a slower rate to generate an elegant rhythmic pattern. Two syllables, Ma and Bi, sounded the pattern at a constant tempo. By pulling the Gametrak's strings upward and downward, I was able to control the playback frequency in real time.

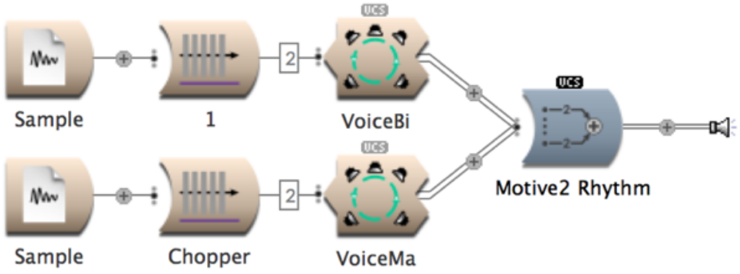


Figure 68. Signal flow graph of rhythm motive

The first development of the rhythmic motive occurred immediately following the presentation of the original motive in the same section. Besides Ma and Bi, one more syllable Ti, which previously occurred in the grain motive, was introduced to create a three-layer rhythmic pattern with the voice in this development. Bell sound material was also rhythmized using this same technique.

A second instance of the motivic development of the rhythmic motive was changed in terms of the number of layers used, the rate of the amplitude modulation, and the spectrum of the sound in section B (at 3:34 of the supplemental video). Only two rhythmic layers with the syllables Ma and Bi were used; the constant rhythm of the amplitude modulation was gradually slowed until the sound stopped; and subtractive synthesis was applied to the entire sound to shape the timbre by removing high-frequency components from the spoken voice. This developmental simplification of the rhythmic motive functioned as a transition to the next section that was much quieter than previous parts of the composition.

A third example of the development of the rhythmic motive can be found in section B’ where the rate of rhythmic articulation was half that of the original (at 5:42 of the supplemental video). The same rhythmic structure was applied in this stage but with other layers. The two primary rhythm layers, based on syllables Gei and Ong, were initiated through automation. As a basis of the third motivic development, these two syllables were not subjected to frequency change or pitch variation. Two additional rhythmic layers with syllables Tai and Tu were added in this development. The Tai and Tu syllables needed to be triggered through my own performative actions. The frequency of the sound on these two layers were varied and changed when I moved the strings horizontally in the X-axis. Besides the four short syllables, longer phrases were used as another two layers of rhythm. Since the lengths of these two sound materials were different and longer than other four rhythmic layers, an out of sync effect was generated. By diverging from a constant rhythmic articulation, the current sonic event smoothly transitioned to the next section, which contained no rhythmic elements. I chose to use the four syllables, Gei, Ong, Tai, and Tu, because of the sonic traits each offered, particularly the syllables Tai and Tu, which begin with notably noisy and percussive accents. After working with the sound materials, I found that these particular syllables created satisfying and desirable sonic results.

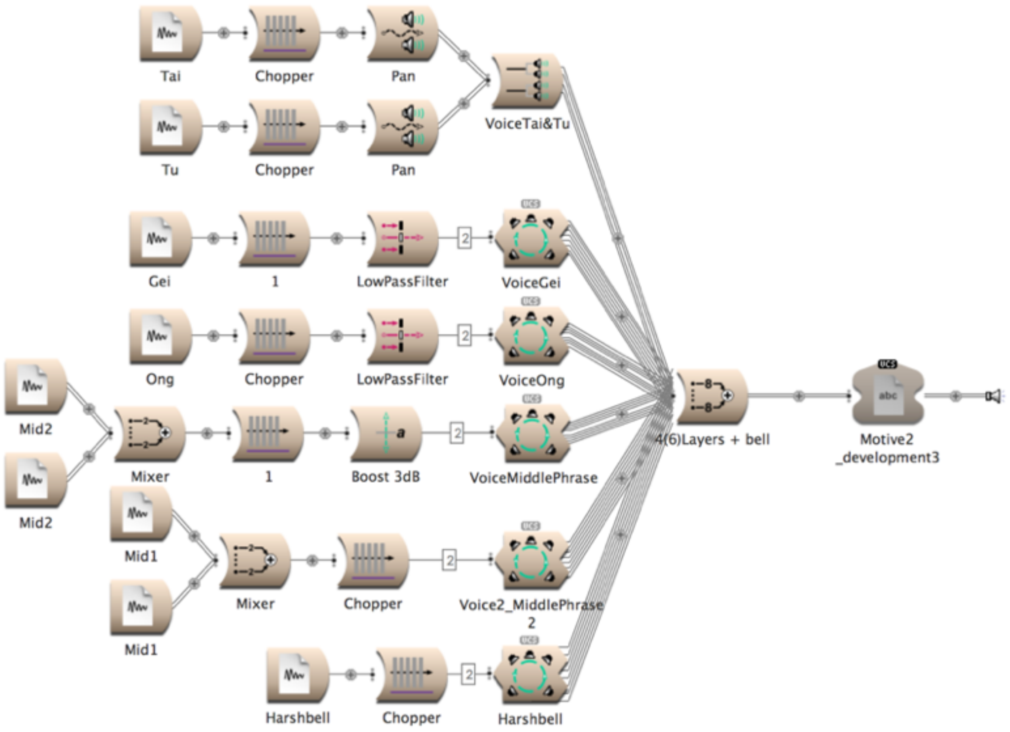


Figure 69. The signal flow graph of the third development of rhythm motive

Performance Techniques

The primary performance techniques employed in *Overlapping Strings* involved moving both strings of the interface in the X, Y, and Z dimensions to produce ongoing musical control, and positioning the ends of the Gametrak's two strings in specific spatial positions to trigger musical events. Playing the instrument by moving strings in all three dimensions can be observed between 0:37-2:02 of the supplemental video, where I moved the right string in the horizontal dimension, the X-axis of the string. This movement controlled the timbre of a bell sound. At 2:03 of the supplemental video, I pulled both strings both upward and downward to modulate the frequency of the human voice. Breaching thresholds to trigger musical events was extensively employed to play the rhythmic motive. Since the rhythmic motives occurred at a fast tempo, triggering sounds at precise moments was one of the harder performance challenges. To perform these sections I had to prepare myself both mentally and physically to position the ends of the strings at precise spatial locations. Moving in psychological sympathy with the music, such as head nodding or foot tapping, helped me to better prepare for specific moments in the performance. In order to not distract myself from the rapid changes of performative actions, I intentionally did not look at the computer monitor positioned in front of me and fully focused on listening to the sound changes when I was performing.

CHAPTER III.7

DOUBLE SHADOWS

Overview

Double Shadows, for two Blue Air controllers, Max custom software, and Kyma, is a stereo interactive composition based on the idea of imitation. The sounds controlled by the infrared sensors contained in the two Blue Air Controllers acted like shadows of each other. In their performative actions, my two hands alternated in leading and following the sounds to create an interactive contrapuntal musical experience.

Design and Implementation of the Data-driven Instrument used in *Double Shadows*

The Blue Air controller was based on the technology of infrared sensors. Conceptually designed by Jeffrey Stolet, the Blue Air is a MIDI controller with an infrared sensor inside of the controller, which detects and reports how close the nearest object is. The operating range is approximately 10 to 150 centimeters.[44]

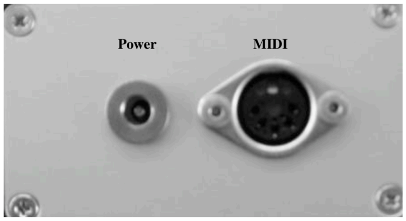


Figure 70. The power and MIDI outlet from Blue Air controller[45]

---

[44] Stolet and Lane, *Blue Air*, 23.

[45] Stolet and Lane, *Blue Air*, 5.

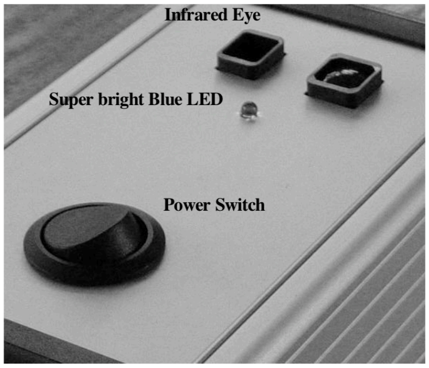


Figure 71. Top of Blue Air controller[46]

Based on the distance my hands were placed from the Blue Air's infrared sensing eye, the Blue Air output MIDI continuous controller data. The Blue Air controllers connected via a MIDI cable to a MIDI interface and the MIDI interface transferred continuous control messages through a USB cable connection to the computer. Running on the computer, Max software received and modified the MIDI data. To communicate between the software layer and sound-producing algorithm environment, the assistive software KymaConnect helped deliver MIDI messages from Max to Kyma. The data-driven instrument used in *Double Shadows* was comprised of three parts: 1) two Blue Air controllers, which output data when I operated them, 2) Max, which analyzed the initial MIDI data output from Blue Air controllers and modified data based on the analysis of that data, and 3) Kyma, which received the transformed data (through KymaConnect) to produce sonic results.

---

[46] Stolet and Lane, *Blue Air*, 5.

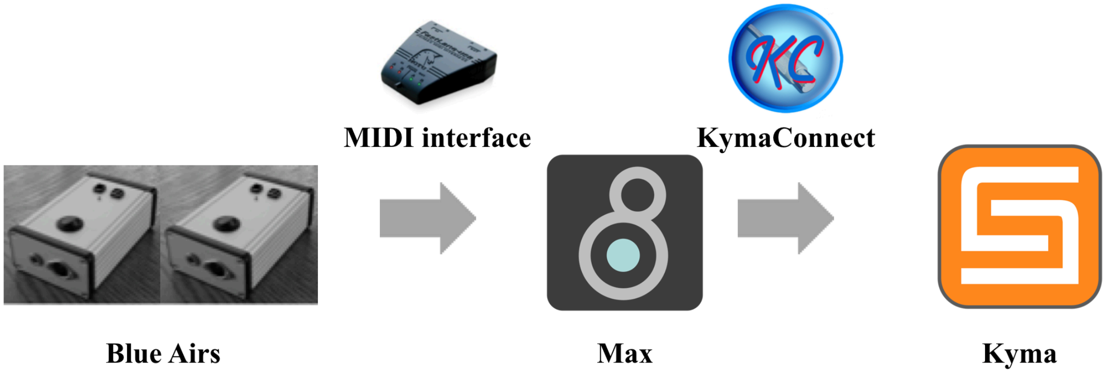


Figure 72. The overall structure of the data-driven instrument used in *Double Shadows*

The Max patch that received and transferred MIDI messages from the Blue Air controllers to Kyma is shown in Figure 73. These controller messages were received by the *ctlin* object within Max. These original MIDI data were then thinned by using two *speedlim* objects. Data thinning is a mapping strategy which results in a fewer number of data points being output than input. Because Blue Air controllers output data every 2 milliseconds, I used data thinning to reduce the number of data points in order to save computational resources. The two modified data streams were connected to two *ctlout* objects which sent the MIDI messages through KymaConnect to Kyma.

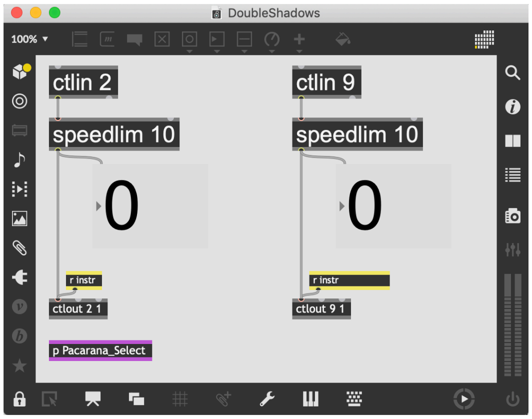


Figure 73. Data received into and sent out of Max

Through the assistive software KymaConnect, Kyma received the modified MIDI data and responded to the data by triggering audio files of percussion instruments (e.g., rainstick, cabasa, and bells) and by controlling musical parameters (e.g., frequency and amplitude) of the audio files in real time. These two sounds comprised the two sonic motives for the piece.

Musical Challenges and Opportunities

The rapid speed that data was output provided the opportunity for me to generate much data in a very short period of time. Because of the rate at which data was output from the Blue Airs, a wide variety of musical parameters including frequency, amplitude, and timbre could be changed at nearly instantaneous speeds.

The Blue Air is a simple device that outputs only one data stream of continuous controller values. This could be seen as a challenge since the single data stream did not provide many flexible possibilities to control sound parameters. To solve this particular challenge, two Blue Air controllers were used in this composition to provide two individual streams of MIDI continuous controller values.

Another challenge for the Blue Air controller was, from a performer's perspective, the lack of physical touch. Because I interacted with the controllers without physical contact, performative actions were all based on a measurement of distance. Even small deviations in the performative actions produced variations in data values and sonic results. To help mitigate this problem, I placed the two Blue Air controllers at the same height when rehearsing and performing. To conveniently place the two devices at the same height, I decided to place the Blue Air controllers on a piano bench. This accomplished two things. First, piano benches are common and could be easily found in concert halls. When performing at different festivals, conferences, or concerts, I could request a piano bench for my performance without difficulty. Second, piano benches are dependably the same height and many permit their bench level to be adjusted.

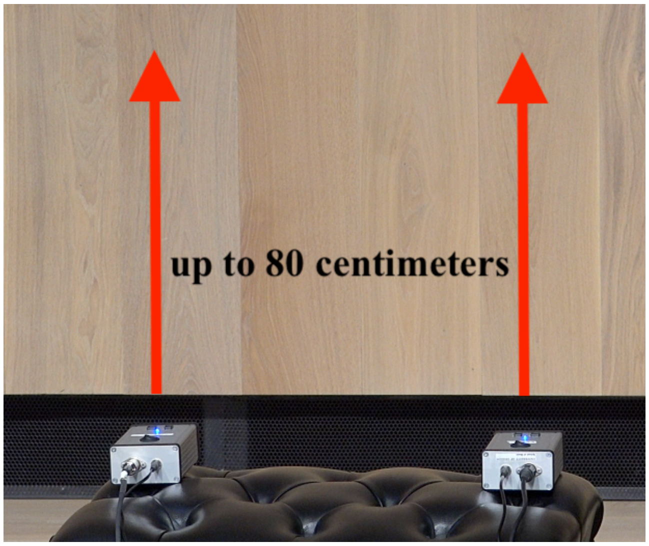
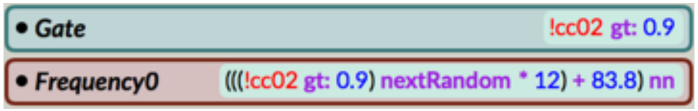


Figure 74. The placement of Blue Air controllers

Data Mapping Strategies

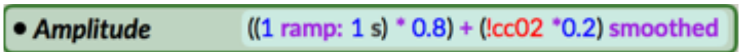
In addition to data thinning applied in the Max layer to reduce the data transmission rate, I used data scaling, offsetting, and smoothing in Kyma. Additionally, I constructed thresholds that data streams could breach in order to convert fader-like functions into button-like functions. Since the Blue Air controllers produced only one continuous data stream at a time, transforming these streams to produce button-like functionality was one of the essential data mapping strategies I used. When a sound needed to be turned on and off, a button or toggle was required. Figure 75 shows how this technique was executed in two ways. When values from continuous controller number 2 were greater than 0.9, I triggered the sounds and simultaneously produced the sound's frequency—random MIDI note values between 71.8 and 95.8.



```
• Gate                          !cc02 gt: 0.9
• Frequency0     (((!cc02 gt: 0.9) nextRandom * 12) + 83.8) nn
```

Figure 75. Turning continuous controller data stream into a button

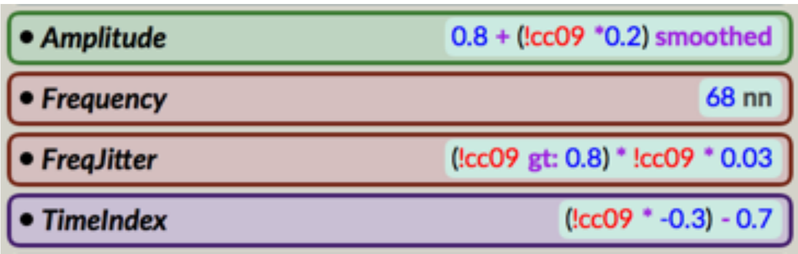
To control the amplitude of my sounds I used data smoothing. Smoothing played a vital role in the composition’s essential musicality. For example, the smoothing of the amplitude values helped remove jagged data streams (and their resultant envelope shapes) that might have resulted with my rapid physical movements. Figure 76 shows continuous controller 2 values smoothed within Kyma to produce rather supple and nuanced musical results.



```
• Amplitude     ((1 ramp: 1 s) * 0.8) + (!cc02 *0.2) smoothed
```

Figure 76. Data smoothing

Since I had only two data streams to apply to all musical parameters, the concept of parametric support was an important guiding principal in the mapping and routing strategies I used in this composition. Due to this circumstance, the two data streams often had to be directed to control multiple sounds and multiple musical parameters. As Figure 77 shows, the data stream of continuous controller 9 concurrently controlled the Amplitude, Frequency, and TimeIndex parameters of a *SampleCloud* Sound object. When my hand was positioned close to the controller (outputting lesser values), this action equated to a softer musical dynamic, a lower frequency, and a less intense timbre. When I moved my hand farther away from the controller, I was able to produce a louder musical dynamic, a higher frequency, and a more intense timbre.



Amplitude 0.8 + (!cc09 *0.2) smoothed

Frequency 68 nn

FreqJitter (!cc09 gt: 0.8) * !cc09 * 0.03

TimeIndex (!cc09 * -0.3) - 0.7

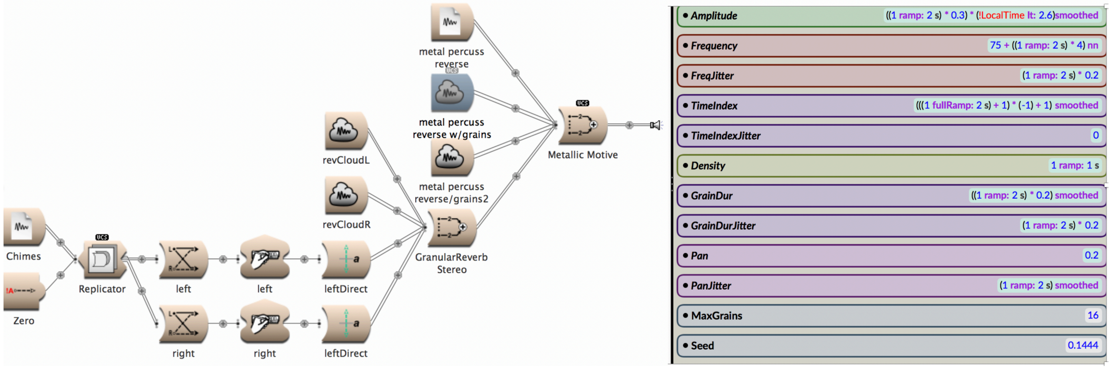
Figure 77. Parametric support using continuous controller 9

Sonic Motives

My concept of motive appears in *Double Shadows* as two distinctive sonic motives, a metallic-sonic motive based on audio recordings of various pitched, metal percussions, and a rustling-sonic motive that was based on the audio recordings of unpitched shaker percussion instruments like the rainstick and cabasa. These two sonic motives occur over the course of *Double Shadows* whose four-section form can be described as: Introduction-A-B-C. Both motives occur in all sections. Below I discuss each motive and describe how it is developed over the four sections of the composition. Because the two motives appear in all sections of the piece and in relationship to each other the organizational of my discussion of motive is related to the composition’s form.

Introduction

The two sonic motives were presented during the introductory section of the piece and each articulated their distinct sonic features. The metallic-sonic motive was comprised of two parts. The first part was based on the striking of a metallic percussion instrument and was played backwards, and the second part was a sonic cloud of granulated chimes with the grains tuned to multiple frequencies. By increasing the amplitude of this sound over 0.8 seconds, the metallic-sonic motive generated a rapidly intensifying crescendo that worked concurrently with other parameters that were also changed to intensify this moment. For example, as Figure 78 shows, the grains changed over the course of two seconds with grain frequency rising four equal-tempered half-steps and the grain density increasing. The parametric support applied in these sounds produced more consistently dramatic and powerful sonic results. The cloud of the grain sound after the crescendo gradually faded the motive to silence at 0:27 of the supplemental video.



Amplitude	((1 ramp: 2 s) * 0.3) * (!LocalTime lt: 2.6)smoothed
Frequency	75 + ((1 ramp: 2 s) * 4) nn
FreqJitter	(1 ramp: 2 s) * 0.2
TimeIndex	(((1 fullRamp: 2 s) + 1) * (-1) + 1) smoothed
TimeIndexJitter	0
Density	1 ramp: 1 s
GrainDur	((1 ramp: 2 s) * 0.2) smoothed
GrainDurJitter	(1 ramp: 2 s) * 0.2
Pan	0.2
PanJitter	(1 ramp: 2 s) smoothed
MaxGrains	16
Seed	0.1444

Figure 78. Signal flow and parametric support related to the metallic-sonic motive

The rustling-sonic motive was presented immediately after the metallic-sonic motive in the introductory section. The rustling-sonic motive included a cabasa recording that was transposed to a high-frequency range and realized within an accelerating rhythmic pattern. The accelerando triggering was realized by using an accelerating *PulseTrain* control signal. Each time when the gate of the high-frequency cabasa recording was actuated, the accelerando triggering pattern followed the speeding-up *PulseTrain* control signal shown in Figure 79.

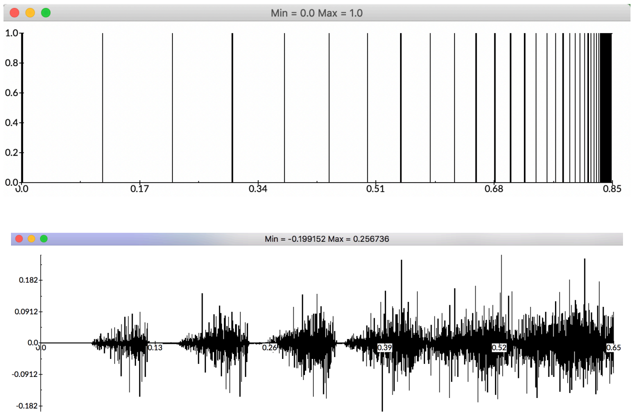


Figure 79. *PulseTrain* control signal (left) and cabasa recording being controlled by *PulseTrain* (right)

Section A

In section A (0:59-3:40 of the supplemental video), I first extended the duration of the metallic-sonic motive and then transformed it into short, rhythmic patterns. Later the metallic-sonic motive was transformed into a sustained sound by using the *SampleCloud* Sound object. Like the introductory section, parametric support was also applied to this sustained sound. The data streams output from the two Blue Air controllers were mapped to control multiple musical parameters including amplitude, frequency jitter, and TimeIndex. The three parameters of the sustained sound changed simultaneously and produced natural and consistent sound results. During the second half of this section, the sustained metallic-sonic motive employed amplitude modulation at a subaudio rate that was controllable in real time.

The rustling-sonic motive in section A included twelve rainstick-based sounds in which the frequency and timbre were constantly modulated. These sounds were detuned from one another with two clear variations: one was transposed down an octave lower than the original recording and one was transposed up an octave. The frequencies of these twelve rainstick-based sounds smoothly alternated back and forth between these two variations throughout the entirety of section A (see Figure 80). By applying a highpass filter to the rustling sound, only higher frequency components occurred. The cutoff frequency of this highpass filter was modulated between a range from 1000 to 2800 Hz and was controlled by data streams produced by the Blue Air controllers in real time.

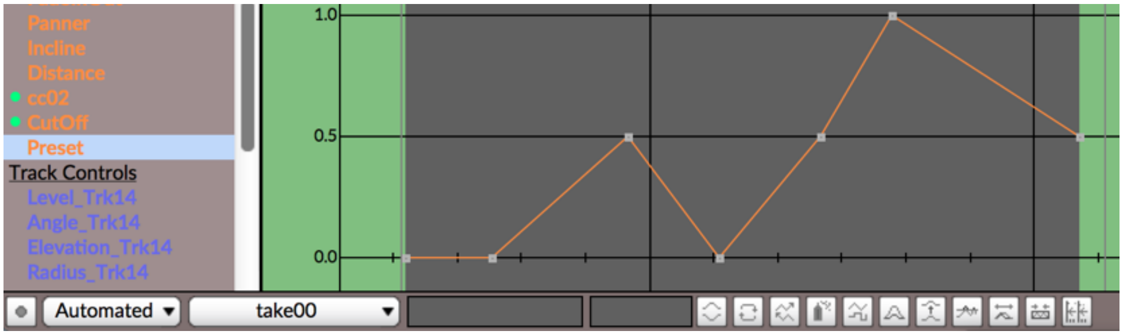


Figure 80. Frequency change between preset 1 and 2 of the rustling-sonic motive

Section B

The rustling-sonic motive in section B (3:40-6:11 of the supplemental video) was presented through sustained sounds with changing frequencies. In contrast, the metallic-sonic motive emphasized individual sonic events triggered at great speeds with rapid changes in pitch. The new motivic structure formed in this section had three parts that occurred in succession: a rustling sound, a metallic sound, and the rustling and metallic sounds occurring together. This motivic tripartite comprised of the two motives parametrically changed over a short period of time. The rustling sound played and descended two octaves over five seconds. The metallic sound that occurred immediately afterwards accelerated in its rate of occurrence from 60 BPM up to 1060 BPM over five seconds (see Figure 81, left side). During the last part of this new motivic pattern, the rustling sound’s frequency ascended over a more extensive time span, and the bell sound was triggered only once. A few seconds later, the same motivic structure was triggered again, but in contrast the bell sound was articulated at a descending rate, from 1060 BPM to 60 BPM (see Figure 81, right side).

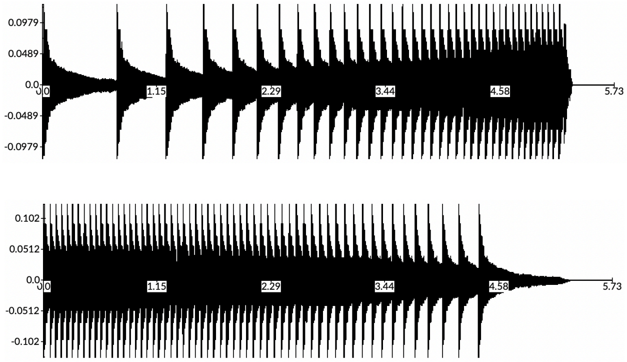
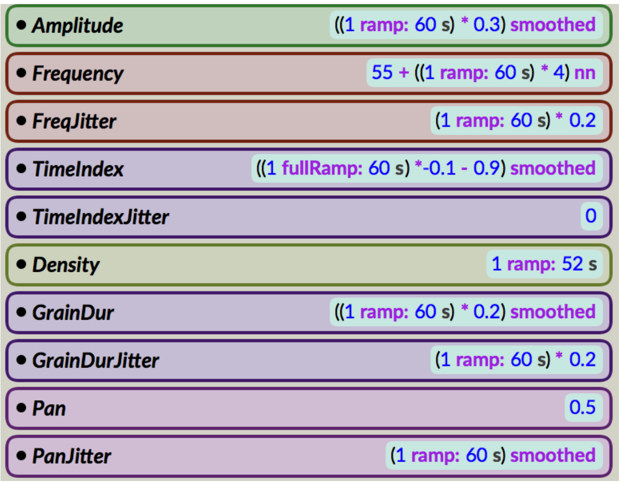


Figure 81. Metallic motive in ascending (left) and in speed descending speed (right)

Section C

In the last section, I developed the metallic-sonic motive further. In the first half of this section, the metallic-sonic motive recalled the sustained sound with parametric support appeared in section A but in a longer duration (see Figure 82). In the second half of this section, I changed the timbre of the bell sound by employing a lowpass filter that attenuated frequencies of the bell sounds above 3834 Hz which darkened the tone.



- **Amplitude** ((1 ramp: 60 s) * 0.3) smoothed
- **Frequency** 55 + ((1 ramp: 60 s) * 4) nn
- **FreqJitter** (1 ramp: 60 s) * 0.2
- **TimeIndex** ((1 fullRamp: 60 s) *-0.1 - 0.9) smoothed
- **TimeIndexJitter** 0
- **Density** 1 ramp: 52 s
- **GrainDur** ((1 ramp: 60 s) * 0.2) smoothed
- **GrainDurJitter** (1 ramp: 60 s) * 0.2
- **Pan** 0.5
- **PanJitter** (1 ramp: 60 s) smoothed

Figure 82. Gentle change in parametric support

The rustling-sonic motive provided a final example of transformational development of sonic motives in my composition. At 6:15 of the supplemental video, the motive appeared in a granular form and was realized through a Kyma-specific analysis and resynthesis technique. By employing the *CloudBank-Resynthesis* Sound object, the rainstick recording, after being analyzed to derive frequency and amplitude information, was resynthesized using a large cluster of grain clouds. Since this synthesis technique is exceptionally distinctive, my sonic motive was effectively transformed in the last section.

Performance Techniques

The primary performance techniques of *Double Shadows* included 1) placing my hands at spatial positions in order to breach numerical thresholds in the data streams, and 2) moving my hands vertically above the mechanism of the infrared sensors to control musical parameters in real time.

Breaching thresholds to trigger sonic events could be seen from 5:38-5:58 of the supplemental video. In this section of the composition, bell sounds were triggered when I placed my hand low enough to breach a numerical threshold. To trigger a sound more than once, I needed to move my hand to a position where it did not breach the threshold, and only then would I be able to breach the threshold again.

The ongoing control created by moving my hands upward and downward above the infrared sensors could be observed from 1:35-1:55 of the supplemental video. In this instance, my right hand controlled the timbre of the sound; I produced a notably harsh timbre when I positioned my hand lower.

The imitation and interaction between my two hands played an essential role in this composition. Such musical and physical counterpoint could be observed at many places in the supplemental video. For example, from 1:19-2:30, my left hand moved downwards to produce harsher and louder sounds. Starting at 1:57 of the video, my right hand made a similar movement and produced the same timbral change but at a different pitch. The interaction between my two hands could be most overtly observed from 4:24-5:12. Here, quick physical movements were required to produce the two short motives one after another. I first raised my right hand high enough to trigger a rustling sound with a descending frequency. Then, I lowered my right hand low enough to trigger the metallic sound that occurred with an accelerating speed. Finally, I quickly placed the left hand into performance area to trigger a combination of the two motives. A few seconds later, the same motivic structures were restated but the two hands recurred an inversional relationship with one hand imitating the other in a canon-like manner.

Chapter IV

SUMMARY

This text document discusses how I composed and performed the seven electroacoustic musical works for data-driven instruments that were centrally structured on the concept of motive. My concept of motive extended the traditional notion of musical motive—primarily predicated on pitch contour and rhythm—to also include concepts of musical timbre, texture density, and spatial location, to name only a few of my additional musical concerns. Because my sonic motives involve issues of timbre, density and spatial locations, matters of sound synthesis played an important role.

Some of the compositions in this dissertation were developed from a single sound recording. For example, *Origin* developed from a one-second pin-drop audio recording. *Dots Illusion* was based thoroughly, both sonically and visually, on the sonic development of a 67-millisecond water droplet recording. More complex synthesis techniques were the primary vehicle for the development process of some pieces of the dissertation. In *Huadan's Whisper* I used two synthesis techniques, granular and subtractive synthesis. In this case, the bell and voice frequency spectra were developed and varied using filters. In *Overlapping Strings,* I employed granular synthesis and sampling techniques to produce the rhythmic developments of the motive. In *Pulsing* I used a sequencing technique that divided sounds into small particles and a resynthesis technique which stretched sounds into longer durations. Instead of a specific sound recording and synthesis technique, *Double Shadows* focused on building a composition dependent on distinct timbres. The metallic and rustling sounds generated from different recorded materials were developed into different variations using sampling, granular, and resynthesis techniques. *Frozen* arose from an unusual ice storm event that occurred in Eugene, Oregon in 2016 and was based entirely on the timbral development of recordings made immediately following the ice storm.

In contrast, the Lightpad Block and Wiimote controllers did not allow or facilitate the simultaneous outputting of X- and Y-dimension data without difficulty. The small size of the Lightpad Block controller limited the ability to employ easily performatively produce X- and Y-dimension data. I used multiple layouts of the Lightpad Block to circumvent this problem. With the Wiimote, the data performatively produced was notably intertwined, also influencing which performative actions were available to me. Here, my solution was to embrace the characteristics of the entanglements and to map the data onto musical parameters that produced unique sonic outcomes. For controllers like Gametrak, Leap Motion, and Blue Air, which only provided continuous control, musical changes were expressed through delicate movements in three-dimensions and musical events were triggered using data thresholds and quick movements. The totality of the musical influences created by the substantial variation in my performance interfaces coupled with the structural unity provided by my concept of sonic motive helped me to produce a satisfying body of compositions where my personal artistic voice can be seen and heard.

BIBLIOGRAPHY

Akamatsu, Masayuki. “aka.leapmotion,” akalogue. Accessed February 16, 2020. http://akamatsu.org/aka/max/objects/.

Bachmann, Daniel, Frank Weichert, and Gerhard Rinkenauer. “Evaluation of the Leap Motion Controller as a New Contact-Free Pointing Device.” *Sensors* 15, no. 1 (2014): 214–33. https://doi.org/10.3390/s150100214.

Beijing Tourism. “Chinese ‘Dan,’ A Form of Chinese Opera.” Accessed February 16, 2020, http://english.visitbeijing.com.cn/a1/a-X9Y2KW46700B483C0FF417.

Colgan, Alex. “How Does the Leap Motion Controller Work?” Leap Motion, August 9, 2014. http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/.

Drabkin, William. “Motif.” *Grove Music Online*. 2001; Accessed 28 Apr. 2020. https://www.oxfordmusiconline.com/grovemusic/view/10.1093/gmo/9781561592630.001.0001/omo-9781561592630-e-0000019221.

Freed, Adrian, Devin McCutchen, Andy Schmeder, Anne-Marie Skriver, Dan Overholt Hansen, Winslow Burleson, Camilla Nørgaard Jensen, and Alex Mesker. “Musical Applications and Design Techniques for the Gametrak Tethered Spatial Position Controller.” In *Proceedings of the SMC 2009 – 6th Sound and Music Computing Conference*, 189-194. Porto, Portugal, 2009. https://doi.org/10.5281/zenodo.849645.

Kyma. “Kyma Control for iPad.” Accessed April 30, 2019. https://kyma.symbolicsound.com/kyma-control-for-ipad/.

Landy, Leigh. *Making Music with Sounds.* New York: Routledge, 2012.

Leap Motion. “API Overview.” Accessed February 16, 2020. https://developer-archive.leapmotion.com/documentation/v2/csharp/devguide/Leap_Overview.html.

OSCulator. “What are Pitch, Yaw, and Roll?” Wiimote FAQ. Accessed March 9, 2020. https://osculator.net/doc/faq:wiimote.

Roli. “Lightpad Block,” Accessed April 28, 2020, https://roli.com/products/blocks/lightpad-m.

Scaletti, Carla. “Computer Music Languages, Kyma, and the Future.” *Computer Music Journal* 26, no. 4 (2002): 69–82. https://doi.org/10.1162/014892602320991392.

Schaeffer, Pierre, Christine North, and John Dack. *Treatise on Musical Objects: An Essay across Disciplines*. Oakland, California: University of California Press, 2017.

Stolet, Jeffrey. *Kyma and the SumOfSines Disco Club.* Morrisville: Lulu Press, 2012.

———. "Twenty-three and a Half Things about Musical Interfaces." *Keynote Address.* Brussels, Belgium: Kyma International Sound Symposium, September 14, 2013.

Stolet, Jeffrey, and Andrew Lane. *Blue Air Manual and Documentation*. Eugene, University of Oregon, 2004.

Wang, Chi. "Narrative to Action in the Creation and Performance of Music with Data-driven Instruments," DMA diss., University of Oregon, 2018.

Wacom. "Wacom Intuos Pro- Large." Accessed February 6, 2020. https://estore.wacom.com/en-US/wacom-intuos-pro-l-us-pth860.html.

Zbyszynski, Michael, Matthew Wright, Ali Momeni, and Daniel Cullen. "Ten Years of Tablet Musical Interfaces at CNMAT," In *Proceedings of the 7th International Conference on New Interfaces for Musical Expression*, 100–105. New York, USA, 2007. https://doi.org/10.1145/1279740.1279758.

Discography

Legend of China sample library, WAV audio. Discovery Sound DFSD630, 1999.