Generative Music from Fuzzy Logic and

Probability:

A Portfolio of Electroacoustic Compositions

PhD Documentation

Richard Garrett

Student ID: 500254808

27 April 17

Abstract

This portfolio of thirteen recorded works was composed as an investigation into the application of generative processes to electroacoustic music, paying particular attention to the use of fuzzy logic and the rule-based constraint of chance events.

These works were developed by a rolling process of program design and musical composition, focusing on two areas: the generation and transformation of large groups of sounds within a multi-dimensional parameter space for acousmatic composition (using the author's software, Audio Spray Gun) and the real-time selection of sounds using audio descriptors, principally for live performance by instrument and electronics.

Later stages of the project attempted to unite these processes in two ways: by the agent-based generation of large sound-groups for multichannel audio from live instruments or pre-recorded audio datasets and by the software generation of such groups for fixed-media composition using trajectories and transformations in 'timbre space'.

An accompanying document charts the development of these works with a programme note, technical discussion and performance records for each, along with spectrograms and scores as appropriate. It also describes the programming methods used and discusses the implications and limitations of these approaches, particularly for object-based spatial music and timbre selection.

Contents

List of Illustrations

List of Tables

Sound Examples

File Structure

This portfolio and its supporting files are available in two formats: on a USB flash drive accompanying the hard copy of this document, and online. In each case, files are organised as shown in Table 1. Folders are shown in bold type (for example, **4_1 Once Below A Time**) and individual files are shown as regular text. All recorded works stored in the **Portfolio** folder are grouped in sub-folders corresponding to chapters in the text. Each work has its own sub-folder within the chapter folder, which in turn, contains a folder of multichannel files where appropriate and a stereo reduction for reference.

Each multichannel folder is labelled with a suffix denoting the speaker format employed and the number of mono files contained within (for example, **5_3 Penumbra 16_0** refers to 16.0 format and therefore contains 16 mono audio files). All audio files except the stereo sound examples are in 48 kHz, 24-bit format. Stereo sound examples are encoded as 16-bit, 44.1kHz audio for compatibility with those embedded in the .pdf version of this document. Loudspeaker maps for multichannel works are included both as an appendix (page 132) to this document and as separate .pdf files on the flash drive.

Folder hierarchy (descending) --->

Portfolio	-	1_The Constraint of Randomness	-	1_1 November	-	1_1 November 4_0 1_1 November stereo.wav
				1_2 In Flight	-	1_2 In Flight 8_0 1_2 In Flight stereo.wav
				1_3 Far Skies	-	1_3 Far Skies stereo.wav
		2_Agents Part 1_Exploration	-	2_1 Exploration	-	2_1 Exploration stereo.wav
				2_2 Exploration II	-	2_2 Exploration II stereo.wav
		3_Zones	-	3_1 Inside Track	-	3_1 Inside Track 8_0 3_1 Inside Track stereo.wav
		4_ASG Scan	-	4_1 Once Below A Time	-	4_1 Once Below A Time 8_0 4_1 Once Below A Time stereo.wav
		5_3D and Metadata	-	5_1 crunch	-	5_1 crunch 16_0 5_1 crunch stereo.wav
				5_2 Penumbra	-	5_2 Penumbra 16_0 5_3 Penumbra stereo.wav
		6_Agents Part 2_Harvesting	-	6_1 Harvest Improvisation	-	6_1 Harvest Improvisation 8_0 6_1 Harvest Improvisation stereo.wav
		7_Agents in Space	-	7_1 Guitar Piece	-	7_1 Guitar Piece 8_0 7_1 Guitar Piece stereo.wav
		8_Timbre Space	-	8_1 Morphology	-	8_1 Morphology 16_0 8_1 Morphology stereo.wav
				8_2 Objets Volants	-	8_2 Objets Volants 16_0 8_2 Objets Volants stereo.wav

Software	-	Audio Spray Gun 0.8	-	SC files param examples read me
		Exploration Patch	-	max project read me
		Harvesting	-	SC files read me

Sound Examples	-	Multichannel in folders Stereo examples (.wav)

Speaker Maps	-	3 x .pdf
Text	-	1 x .pdf

Appendix F		F Meanwhile 16_0 F Meanwhile stereo.wav

read me

Table 1: File Hierarchy

Preface

In the summer of 2004, it was my great good fortune to attend the second Workshop in Algorithmic Computer Music[1] (WACM), led by David Cope at the University of California, Santa Cruz. I had been interested in generative music, working with Koan Pro[2] since about 1996 and was keen to discover more about compositional techniques available in this field. Although all the teachers on the course inspired me, the person who most influenced my methods of composition was Peter Elsea.

In his lectures[3], Peter showed us how fuzzy logic could fashion precise systems to control diverse and apparently incompatible parameters. I left WACM convinced that this was what I needed to create my own generative music and some time later developed the method of combining it with probability upon which this work is based.

This technique became a central element of my compositional practice and is the heart of my work with *Weathersongs (2006)*[4] and *nwdlbots (2008 -11)*[5]. So, when I came to Bangor University in 2011 to study sound-art and electroacoustic composition for my Masters degree, it was no great surprise that I started applying this approach to those fields. These efforts have continued with the work documented here.

---

[1] WACM, <http://arts.ucsc.edu/programs/wacm/>, last accessed 5 May 2016.
[2] Koan Pro, <http://en.wikipedia.org/wiki/Koan_%28program%29>, last accessed 5 May 2016
[3] see Peter Elsea, 'Fuzzy Logic and Musical Decisions', 1995
<http://artsites.ucsc.edu/ems/Music/research/Fuzzy_Logic_And_Music.pdf >, last accessed 20 April 2016.
[4] Richard Garrett, 'The Weathersongs Project' < http://www.sundaydance.co.uk/the-weathersongs-project/> last accessed 20 April 2016.
[5] Richard Garrett, 'nwdlbots' <http://www.sundaydance.co.uk/nwdlbots/>, last accessed 20 April 2016.

Acknowledgements

Thanks to:

Andrew Lewis for supervision.

Ed Wright, Richard Nelmes, Alex Bailey.

All the colleagues who rigged the concerts at Bangor.

Richard Craig, Ellie Lighton and Hephzibah Leafe for performing *Exploration I* and *II*

Sean Goldthorpe for contributing flute sounds during their development.

All the software developers on whose work this music depends, notably Scott Wilson for his VBAP SuperCollider UGens, Tom Erbe for Autohack, Norbert Schnell et al. for Mubu, and the ever-helpful Max and SuperCollider communities online.

Members of the global electroacoustic community who've given me platforms on which to play and to talk about my work.

Patricia Alessandrini, who told me to do this.

The United Kingdom Arts and Humanities Research Council for funding the PhD

and Peter Elsea, without whom…

Finally, to Heather and Sean for putting up with it all the weird noises from the shed and supporting me throughout.

Declaration and Consent

Details of the Work

I hereby agree to deposit the following item in the digital repository maintained by Bangor University and/or in any other repository authorized for use by Bangor University.

Author Name: Richard Anthony Francis Nigel GARRETT

Title: Generative Music from Fuzzy Logic and Probability: A Portfolio of Electroacoustic Compositions

Supervisor/Department: Professor Andrew LEWIS/ School of Music

Funding body (if any):   Arts and Humanities Research Council

Qualification/Degree obtained: Doctor of Philosophy

This item is a product of my own research endeavours and is covered by the agreement below in which the item is referred to as "the Work". It is identical in content to that deposited in the Library, subject to point 4 below.

Non-exclusive Rights

Rights granted to the digital repository through this agreement are entirely non-exclusive. I am free to publish the Work in its present version or future versions elsewhere.

I agree that Bangor University may electronically store, copy or translate the Work to any approved medium or format for the purpose of future preservation and accessibility. Bangor University is not under any obligation to reproduce or display the Work in the same formats or resolutions in which it was originally deposited.

Bangor University Digital Repository

I understand that work deposited in the digital repository will be accessible to a wide variety of people and institutions, including automated agents and search engines via the World Wide Web.

I understand that once the Work is deposited, the item and its metadata may be incorporated into public access catalogues or services, national databases of electronic theses and dissertations such as the British Library's EThOS or any service provided by the National Library of Wales.

I understand that the Work may be made available via the National Library of Wales Online Electronic Theses Service under the declared terms and conditions of use (http://www.llgc.org.uk/index.php?id=4676). I agree that as part of this service the National Library of Wales may electronically store, copy or convert the Work to any approved medium or format for the purpose of future preservation and accessibility. The National Library of Wales is not under any

obligation to reproduce or display the Work in the same formats or resolutions in which it was originally deposited.

Statement 1:

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree unless as agreed by the University for approved dual awards.

Signed ………………………………………….. (candidate)
Date …………………………………………..

Statement 2:

This thesis is the result of my own investigations, except where otherwise stated. Where correction services have been used, the extent and nature of the correction is clearly marked in a footnote(s). All other sources are acknowledged by footnotes and/or a bibliography.

Signed …………………………………………. (candidate)
Date ………………………………………….

Statement 3:

I hereby give consent for my thesis, if accepted, to be available for photocopying, for inter-library loan and for electronic storage (subject to any constraints as defined in statement 4), and for the title and summary to be made available to outside organisations.

Signed …………………………………………. (candidate)
Date ………………………………………….

Statement 4:

I agree to deposit an electronic copy of my thesis (the Work) in the Bangor University (BU) Institutional Digital Repository, the British Library ETHOS system, and/or in any other repository authorized for use by Bangor University and where necessary have gained the required permissions for the use of third party material.

In addition to the above I also agree to the following:

1. That I am the author or have the authority of the author(s) to make this agreement and do hereby give Bangor University the right to make available the Work in the way described above.

2. That the electronic copy of the Work deposited in the digital repository and covered by this agreement, is identical in content to the paper copy of the Work deposited in the Bangor University Library, subject to point 4 below.

3. That I have exercised reasonable care to ensure that the Work is original and, to the best of my knowledge, does not breach any laws – including those relating to defamation, libel and copyright.

4. That I have, in instances where the intellectual property of other authors or copyright holders is included in the Work, and where appropriate, gained explicit permission for the inclusion of that material in the Work, and in the electronic form of the Work as accessed through the open access digital repository, *or* that I have identified and removed that material for which adequate and appropriate permission has not been obtained and which will be inaccessible via the digital repository.

5. That Bangor University does not hold any obligation to take legal action on behalf of the Depositor, or other rights holders, in the event of a breach of intellectual property rights, or any other right, in the material deposited.

6. That I will indemnify and keep indemnified Bangor University and the National Library of Wales from and against any loss, liability, claim or damage, including without limitation any related legal fees and court costs (on a full indemnity bases), related to any breach by myself of any term of this agreement.

Signature: …………………………………………………… Date : ………………………………………….

Introduction

This portfolio was composed as a practice-based research project into the application of generative processes to electroacoustic music, paying particular attention to the use of fuzzy logic and the rule-based constraint of chance events.

The thirteen works focus on two areas: the generation and transformation of large groups of sounds within a multi-dimensional parameter space for acousmatic composition and the real-time selection of sounds using audio descriptors, principally for live performance by instrument and electronics.

Later stages of the project attempted to unite these processes in two ways: by the agent-based generation of large sound-groups for multichannel audio from live instruments or pre-recorded audio datasets and by the software generation of such groups for fixed-media composition using trajectories and transformations in 'timbre space'.

This document charts the development of these works with a programme note, discussion and performance records for each, along with spectrograms and scores as appropriate. It also describes the generative methods used and discusses the implications and limitations of these approaches for object-based spatial music and timbre selection.

This introduction first defines some of the concepts around which the work is based, then looks briefly at some of the algorithmic compositional methods employed before discussing the methodology used and the structure of the portfolio.

Definitions

Generative Music

If the range of music that may be constructed algorithmically, with or without the use of a computer, can be thought to extend from wholly determined processes (for example, Steve Reich's *Clapping Music* (1972)[6]) to completely random ones (perhaps throwing dice to decide unconnected actions) then, for the purposes of this discussion, *generative music* lies between the two, employing algorithms that choose from one of several possible outcomes each time they are run. To quote Brian Eno, writing about instructions from Koan Pro (the program for which he coined the term 'generative music'):

> …rather than saying 'Do precisely this' (which is what a musical sequencer does) they say 'choose what you do from within this range of possibilities'.[7]

The extent to which generative processes are utilised in composition can vary enormously. Works range from the wholly algorithmic to those in which the composer generates and selects components before arranging them by hand. Two approaches are taken in this portfolio. In the instrumental pieces, generative agents are used to create an accompaniment in real time from the playing of a live performer. For the acousmatic works, software programs are used to generate multichannel audio files that are then arranged on the timeline of a digital audio workstation (DAW).

Fuzzy Logic

Fuzzy logic is a branch of mathematics and an engineering approach first developed by Lofti Zadah in 1965[8] (although the notion of vagueness can be traced back to Bertrand Russell[9]). It "is not logic that is fuzzy, it is a logic that deals with fuzzy issues."[10]

---

[6] Steve Reich, 'Clapping Music for two performers', < http://earreader.nl/wp-content/uploads/2016/01/SteveReich-ClappingMusic.pdf>, last accessed 22 March 2017.
[7] Brian Eno, *A Year with Swollen Appendices*, (London: Faber 1996) p. 331
[8] Lofti Zadeh 'Fuzzy Sets' *Information and Control* 8:3 (June 1965), 338 -353
[9] Betrand Russell, 'Vagueness', *The Australasian Journal of Psychology and Philosophy,* (June 1923), 84-92

In Aristotelian (bivalent or *crisp*) logic, every proposition is either *true* or *false*. While this is sufficient to answer some real world questions (a light switch, for example, is either on or off), others will elucidate responses whose shades of meaning cannot be captured merely by 'yes' or 'no'. Questions like 'are you tall?', 'is the room hot?' or 'do you like ice cream?', for example, may produce answers such as 'not very', 'a bit' or 'now and again'. Fuzzy logic seeks to quantify such responses, expressing for each one the degree to which it holds true. [11]

Consider the question 'are you tall?'. This cannot usefully be answered using bivalent logic. It seems reasonable to state that 5'0" is *not-tall*, and that 6'0" is *tall* but crisp logic can only set an arbitrary boundary (say 5'6") between the two states. In fuzzy logic, *membership values* between zero and one can be used to define the property *tall* for all possible heights. Thus, three people of 5'0", 6'0" and 5'8" might exhibit a degrees of tallness of 0.0, 1.0 and 0.7 respectively (see page 102). As with crisp logic, these statements can be combined by methods such as union (for example, *tall* OR *heavy)* and intersection (*tall* AND *heavy*) that also exhibit degrees of truth.

Although met initially with scepticism and even derision in the West (one professor describing it as "the cocaine of science"[12]), fuzzy logic was swiftly adopted in Japan for use in control systems and has been used globally for this purpose since the 1990s. Fuzzy control systems have been implemented for a large number of products including air conditioners, anti-lock brakes, dishwashers, microwave ovens and toasters. [13]

---

[10] Peter Elsea, 'A Fuzzy Logic Primer', 2007 <http://artsites.ucsc.edu/ems/Music/research/Fuzzy_primer07.pdf >, last accessed 20 April 2016.
[11] As often expressed by the Fuzzy Principle that "everything is a matter of degree". Anonymous, in Bart Kosko, *Fuzzy Thinking: The New Science of Fuzzy Logic,* (London: Harper Collins 1994).
[12] William Kahan quoted in *ibid* p.3
[13] *ibid* pp.184 -187

Generative methods from Fuzzy Logic and Probability

Fuzzy logic can be combined with probability in a number of ways to form generative processes. Here are the methods employed in this portfolio.

Method 1: The use of Fuzzy Rules to constrain probabilistic selection

Fuzzy control operates by the application of *rules*. A rule is equivalent to a conditional statement in conventional programming and defines the relationship between all the possible inputs and outputs of a process.
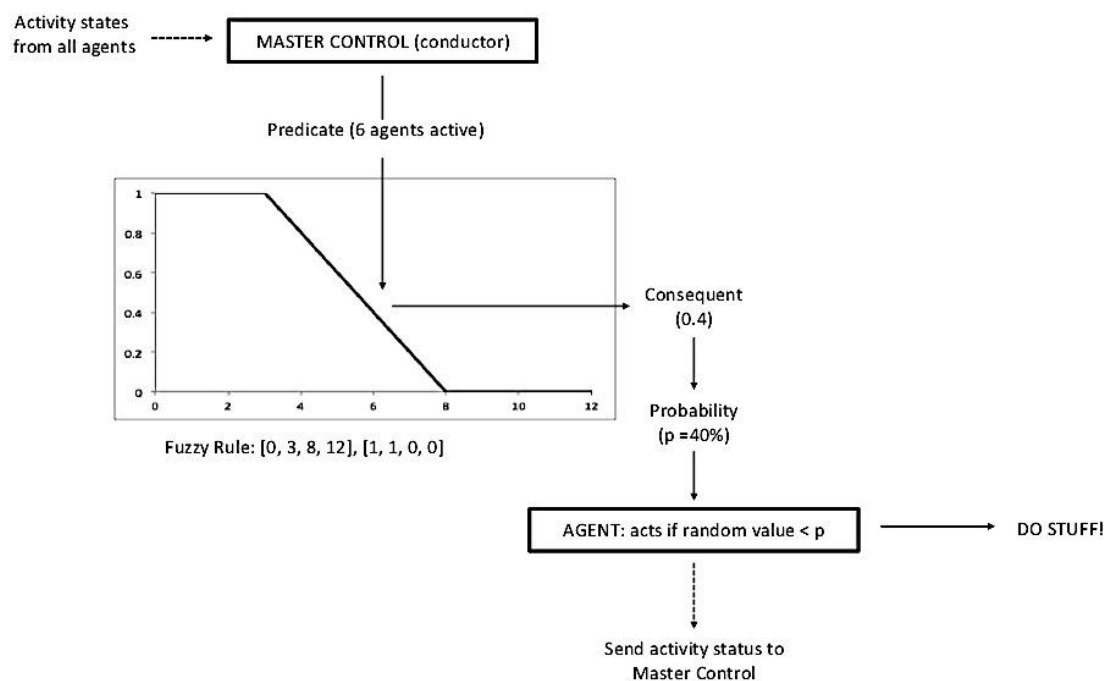
Figure 1: A fuzzy rule

Consider a system in which some agents are making sounds. To control the number of agents active at any one time, one might implement a rule that, stated in English, says 'if the music gets too busy, stop playing'. A crisp solution would be to simply prevent agents from acting when the density exceeded an arbitrary activity level (say, five agents sounding together). This would give a very consistent and probably unmusical result. A more subtle alternative would be to implement a fuzzy rule which returns different values according to the number of active agents, say, 1.0 if the number is

three or less, falling to zero as the number approaches eight. In Figure 1, the presence of six active agents returns a value of 0.4. This is then converted into a probability that determines the likelihood that another agent will sound. In this system, agent density will be constrained in a less predictable but more interesting fashion than results from the crisp approach.

Method 2: A Fuzzy-probabilistic Hybrid

A typical fuzzy control system consists of some combination of rules. As with a single rule, these define a set of behaviours (outputs) for all the possible inputs to a system. When such a system is used to control express trains or dishwashers and a specific input occurs, the best available output is selected from the set of all possibilities by a process known as *de-fuzzification*. This method, however, takes that set immediately before it would normally be de-fuzzified and converts into a weighted probability table from which values may be chosen at random. This results in a generative system that tends towards 'best-fit' solutions without always sticking to them.

If system rules are chosen so that at least one of them is dependant on the most recently selected value (for example, note interval) and one is independent of this (a scale) then a continuously varying but self-similar output can be obtained (see page 106).

This *fuzzy-probabilistic hybrid* was the basic building block for the code underlying *Weathersongs* (2006) and *nwdlbots* (2010) and was first adapted for electroacoustic music in my MA pieces *Only Now* (fixed media) and *Out of The Loop* (for hexaphonic guitar and electronics).

Method 3: Selection from a data array

This method, used in *Exploration Patch*, *Harvest* and *Guitar Piece*, demonstrates the ease with which fuzzy rules can be used to respond to disparate input parameters. This is achieved by applying method two to every element of a multi-dimensional data array.

Take, for example, a corpus of sound-objects in which each member has been analysed and described in terms of 'noisiness', duration, 'brightness' and so on. One could define a fuzzy rule for the condition *long*, apply it to the duration of each object in turn and then store the resulting membership values in an array. Longer duration events would then have the highest scores. If this process were repeated for a condition called *noisy*, an array of scores for the noisiness of objects could be obtained. The logical combination of these arrays would reveal those objects that are both *noisy* AND *long* and could be applied to a generative system, biasing its outputs towards long, noisy events.

Method 4: Rejection Zones

Method four uses of *rejection zones* to automatically distribute events or groups across a parameter range. For example, the automated spatial distribution of events used in *Harvest* and *Guitar Piece* (see page 110).

At the beginning of the process, an array is initialised that associates all the values available for a particular parameter with a set of probabilities. Typically, this would give each value an equal chance of selection. Once a value has been chosen, the array is modified to create a zone of reduced probability around it and its near neighbours giving them a lower chance of selection than more distant ones. This rejection zone persists for some limited time (for example, the duration of the associated event) and then is removed from the calculation. The result of this is that each time an event occurs, the current combination of zones forces the selection of a new value towards parts of the range where there is little or no activity.

Method 5: Attraction and Rejection

This method, first used in *The Inside Track,* is referred to here as the *Zones* algorithm and it works like this.

Whenever an event occurs, two fuzzy sets are created which control the value assigned to the next event. These are a *rejection rule,* for which all values are equally probable except those in the immediate vicinity of the current event, whose chances of selection decline with proximity and an *attraction rule* which excludes all values except those close to the current event value, whose chances increase with proximity.

When these two rules are combined to make a probability table, the value chosen for the next event is constrained within a fixed distance of the last (not too close, not too far away). These sets need not be symmetrical and may be 'skewed' so that event choice is biased in one direction. When this process is repeated a number of times, with each event 'stepping away' from its predecessor, apparent swirling motions can occur.

The full Zones algorithm, used in works after *The Inside Track,* develops this idea by making new event locations dependant on the positions of multiple events in the past. By steadily reducing the influence of previous attraction and rejection zones over time, selection is biased away from recent choices. By adjusting the sizes of the zones, their skew values and the rates at which their influence is reduced, a range of selection behaviours including random walks, bags and near-total randomness can be obtained (see page 112).

The Project

Methodology

The project was undertaken using a rolling process of tool development and work composition, somewhat similar to the *performer-developer*[14] cycle described by Benjamin Carey. Although this work did involve a certain amount of traditional design and implementation, Figure 2, from McLean and Wiggins[15] showing their bricolage programming cycle, serves to illustrate the approach.
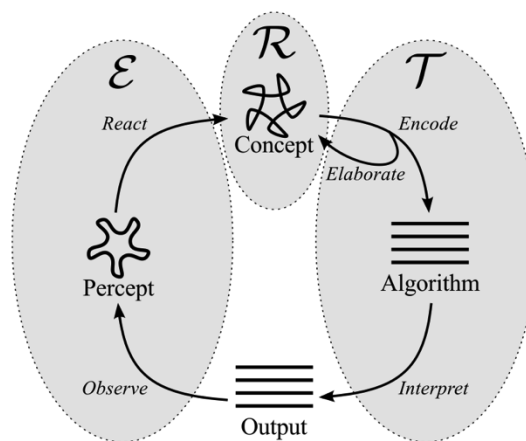
Figure 2: The Bricolage Programming Cycle of action and Reaction with components of the Creative Systems Framework

With one exception, each stage of the project ran like this: start with an idea; write some software to implement the idea; experiment with the software to make sounds; arrange some sounds to make a development piece; elaborate upon this to compose one or two larger pieces; reflect upon these results; decide on the next step and repeat the process.

---

[14] Benjamin Carey, '*_derivations* and the Performer-Developer: Co-Evolving Digital Artefacts and Human- Machine Performance Practices', PhD thesis 2016 p.3 <https://www.academia.edu/21746864/_derivations_and_the_Performer-Developer_Co-Evolving_Digital_Artefacts_and_Human-Machine_Performance_Practices>, last accessed 20 April 2016.
[15] Alex McLean and Geraint Wiggins, 'Bricolage Programming in the Creative Arts' *Proceedings of 22nd Psychology of Programming Interest Group 2010*, Madrid p.6

In the stage marked by chapter three, it became apparent that the zones algorithm needed further development and so no concert piece was produced. *The Inside Track,* although a development work, was the only piece composed at this stage and is therefore retained in the portfolio in order to demonstrate the progression of ideas.

The Works

In the text, each work is presented with a programme note, discussion and performance records. Some of the programme notes included are deliberately short. In some cases, this is due to restrictions imposed by the concert organisers but in others it reflects the composer's desire that the audience listen to the music rather than try to spot the 'tricks' employed. The discussion that follows each programme note focuses on technical and aesthetic aspects of the work.

Scores are included where appropriate and each acousmatic piece is accompanied by a spectrogram to illustrate its overall form and offer some insight into the *mesostructures*[16] employed. For consistency, all spectrograms were produced using *Acousmographe*[17] with each row on the page representing one minute of the piece on a 16 kHz vertical scale.

Durations

All the works in this portfolio are less than twelve minutes long and some readers may wonder why, in a composition PhD, there are none of longer duration. This is in no way due to any limitations inherent in the techniques employed but rather reflects considered choices made by the composer. The aim of this practice-based research project is to investigate a particular set of processes in sound and space. To this end, most of the works included use limited sets of initial sounds so that compositions may focus on the variety of available mesostructures. If works were composed using a more diverse set of sound sources (as might support longer durations), a tendency to replicate structures using different sources might emerge, diluting the listeners' insight into the structures

---

[16] Curtis Roads, *Composing Electronic Music: A New Aesthetic*, (New York: Oxford University Press 2015) p.305
[17] Ina GRM Acousmographe version 3.7.1, < http://www.inagrm.com/accueil/outils/acousmographe> last accessed 28 August, 2016

themselves. This risk becomes more apparent when these pieces are projected in a P-HDLA[18]. When listening to longer works in such an immersive environment, the audience's attention can become so diverted by individual spatial structures that they lose track of the overall form, reducing the work to a succession of grandiose cinema sound effects. To quote Richard Graham:

> The notion of sensory immersion may be an attractive prospect for many artists, performers, and composers, but it is all too easy to overstep into another technological "Oh, boy"-ism or technological management devoid of creative purpose.[19]

These works are neither stereo pieces artfully diffused across a number of speakers, nor arrangements of sound files distributed around a multichannel array as a subordinate process to their spectral composition. They represent a concrete attempt to make music that is conceived of and composed in spectral, temporal *and spatial* domains from the outset, giving equal weight to each. If one subscribes to Varèse's description of music as "organised sound", then this is music organised in space as much as it in time and, if size is considered a valid metric, these works should be judged as much by their spatial extent and sonic complexity as by their temporal duration.

On Listening

If at all possible, these works should be listened to in their multichannel formats. Stereo reduction (included here for reference) not only eliminates most perceived spatial movements but also causes the spectral components of independent events to become unpleasantly smeared together. For example, compare the opening figure of *Morphology* in sixteen channels with its stereo counterpart. In the full version, fifty events spiral out into the entire sound space but in stereo, they sound more like five.

---

[18] Permanent High-Density Loudspeaker Array, see Natasha Barrett, 'A Musical Journey towards Permanent High-Density Loudspeaker Arrays', *Computer Music Journal*, 40:4 (2016) 35-46.
[19] Richard Graham, forum contribution for *Computer Music Journal*, 40:4 (2016), 11-12.

Thesis Structure

Chapters 1 to 8 are roughly chronological, describing the works in the portfolio and the techniques employed to compose them (Figure 3). Earlier chapters cover two processes: the generation and transformation of large groups of sounds within a multi-dimensional parameter space for acousmatic composition (chapters 1, 3, 4 and 5); and the real-time selection of sound-events using audio descriptors (chapters 2 and 6). Chapters 7 and 8 attempt to unite these processes, and chapter 9 draws some conclusions.

Chapter 1 introduces locus composition and discusses three pieces (*November, In Flight* and *Far Skies*). These were all composed using early versions of Audio Spray Gun (page 117), an application for the pseudo-random generation of events within a moving locus in a multidimensional parameter space.

Chapter 2 covers two duets for live instrument and laptop, *Exploration* (2014) for Flute and *Exploration II* (2015) for Clarinet. In each piece, the sounds of the instrument following a score are recorded into a software patch (page 122) that segments and describes incoming events. From these segments, the laptop operator creates an accompaniment by controlling the behaviour of fuzzy-probabilistic software agents that select recorded events for playback.

Chapters 3 and 4 look at two works employing variants of Audio Spray Gun. *The Inside Track* uses a version of the Zones algorithm (see page 112) to create swirling movement within the parameter space locus, and *Once Below a Time* employs *ASG Scan* to extract audio segments from a larger source file and distribute them using parameter space loci as before.
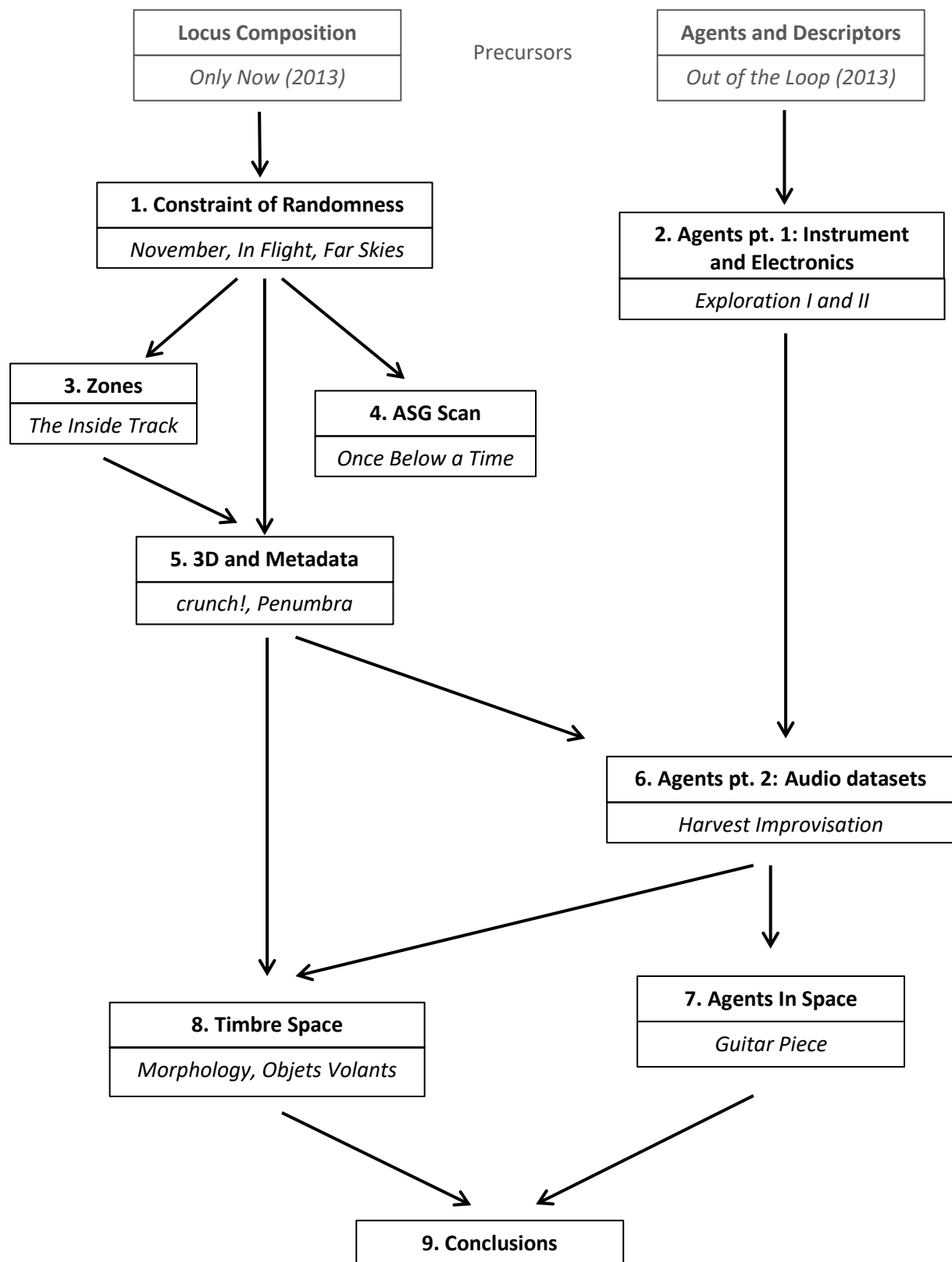
Figure 3: Chapter Map

In chapter 5, *crunch!* and *Penumbra* mark the addition of periphonic audio to Audio Spray Gun and Zones. These pieces also introduce metadata features of the software, which open up possibilities for post-production and spatial re-orchestration[20].

Chapter 6 returns to the theme of feature extraction and agents with *Harvest*, an agent-based system for live improvisation. This program creates sound groups by auto-selecting sources from a pre-analysed database of audio files and uses rejection zones (Page 110) to automate file choice, amplitude, spatial position and so on. This section includes an improvisation controlled by an iPhone to produce an eight-channel fixed-media piece in real time.

Chapters 7 and 8 show attempts to integrate the descriptor-based selection of sound sources with locus composition. In chapter 7, agents are used to produce sound-groups from a live performance (*Guitar Piece*). In chapter 8, 'timbre space' transformations are added to Audio Spray Gun to produce single features from multiple sources (*Morphology*, *Objets Volants*).

Chapter 9 draws some conclusions, reflecting on the work to date, looking at the aesthetic possibilities, practical advantages and shortcomings of working with loci and the possibilities for compositional tools raised by metadata.

Appendices follow these chapters that include a brief introduction to fuzzy logic and a more technical discussion of the methods used to create the pieces in this portfolio; descriptions of software packages (*Audio Spray Gun, Exploration patch*); lists of performances and presentations; and loudspeaker maps for various channel formats. The full-length version of *Meanwhile, In another part of the Forest*, a development work referenced in chapter 5 is also included as an appendix.

---

[20] Eric Lyon, "The Future of Spatial Computer Music." *Proceedings of the 11th Sound & Music Computing / 40th International Computer Music Conference,* Athens 2014, pp. 850-854.

Sound examples are used throughout the text to illustrate elements of the works and the techniques employed. The corresponding multichannel and stereo audio files can be found in the <sound examples> folder included with the project. The stereo versions of each example are also embedded in the .pdf and can be auditioned in Adobe Acrobat by clicking on the italicised captions beneath relevant paragraphs. (**Please note:** Preview for MacOS will not play embedded audio).

Chapter 1. The Constraint of Randomness

November, In Flight, Far Skies (2013 -14)

These pieces were composed using early versions of Audio Spray Gun[21], an application developed to formalise and explore some programming ideas developed during my Masters degree[22]. The application works by constraining random mutations of a single sound-object to a transformable locus within a four-dimensional parameter space. In these works, groups of mutated objects were rendered to multichannel audio files and then arranged in a Digital Audio Workstation (DAW).

This chapter discusses the concepts behind Audio Spray Gun, leaving technical aspects of the software to an appendix (see page 117).

1.1 Locus Composition

In this approach to spatial sound synthesis,[23] events are created as points constrained by a locus or *particle zone*[24] within a virtual parameter space. This locus is transformed over time to produce a sequence of events, which can be rendered to multichannel audio.

1.1.1 Parameter Space

The versions of Audio Spray Gun used for these works operate in a four-dimensional space comprised of two spatial dimensions, inter-onset interval (or *delta time*) and resampling rate. The

---

[21] Richard Garrett, "In Flight and Audio Spray Gun: Generative composition of large sound-groups." *eContact! 17.3 Toronto International Electroacoustic Symposium 2014: 8th edition of TIES,* 2015, <http://econtact.ca/17_3/garrett_audiospraygun.html> last accessed 20 April 2016.

[22] notably in Richard Garrett 'Only Now". Soundcloud <https://soundcloud.com/sundaydance/only-now-2013> last accessed 24 May 2016.

[23] Marlon Schumacher and Jean Bresson, 'Spatial Sound Synthesis in Computer-Aided Composition.' *Organised Sound* 15:3 (2010) 271-289.

[24] Nuno Fonseca, '3D particle systems for audio applications.' *Proceedings of the 16th International Conference on Digital Audio Effects (DAFx-13)*, Maynooth, Ireland 2013, <http://dafx13.nuim.ie/papers/25.dafx2013_submission_50.pdf> last accessed 22 May 2016.

program generates groups of points (*sound-events*) within this space using simple rules that constrain otherwise random parameter choices to a specific transformable locus.

1.1.2 Sound-Events
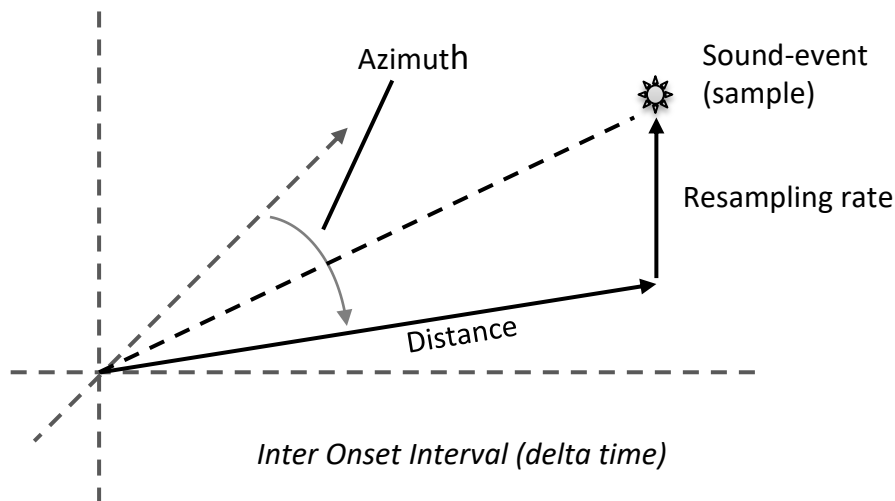
Figure 4: A Sound-Event

Each sound-event in the space is described by five parameters (Figure 4): the file name of the sample to be played by the event plus four co-ordinates describing its position. Each event is stationary within the parameter space and has no spatial or spectral trajectory (for example, panning or pitch bend) of its own.

1.1.3 The Locus

Consider a collection of events occurring within a fixed locus. In two dimensions, such a locus might consist of a circle on a horizontal plane. The system generates events at random points within the circle giving the impression, when rendered as audio, of a body of sound with specific width and depth, centred on a particular location. When this locus is extended into a third dimension (resampling rate), the events play back at random frequencies in a specific range, but still occupy the spatial locus described above (Figure 5). If a fourth, temporal, dimension is added, the time between events is constrained in a similar fashion.
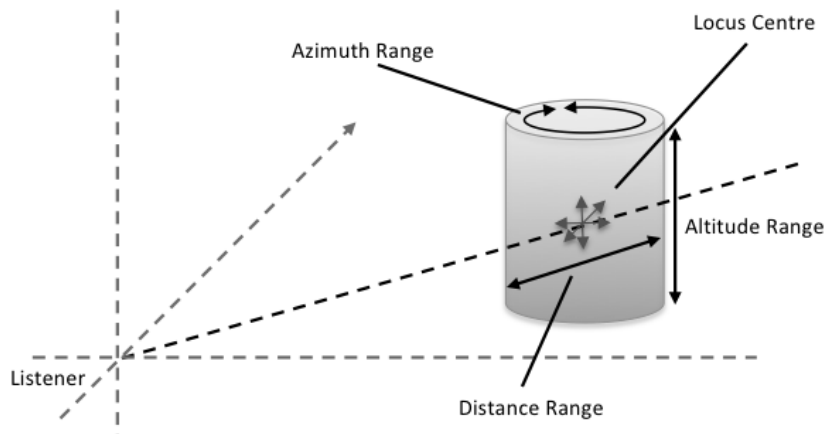
Figure 5: A 3-dimensional locus

1.1.4 Locus Transformation

Once such a locus has been defined, it may be modified over time by transforming its position and extent within the space. This is achieved by altering four pairs of values (one pair for each of the four parameter dimensions). The first of these values locates the focal point of the locus within a given dimension and the second defines its extent about that point. Changing these values over time allows the creation of various trajectories and transformations including: modification of spatial position and extent causing the locus to move through or expand and contract within the sound space; change of position and extent in the resampling dimension so that the spectral range of possible events moves, expands or contracts; modification of delta time to change event density from discernable individual events to thick textures; or any combination of the above.

1.1.5 Sound-groups

Each time the program runs, it creates an array of sound-events (*sound-group*) chosen at random from within the constraints of a locus undergoing some combination of transformations. For each instance of such a group, its constituent events may vary significantly but its gross features remain strictly defined by the program. Thus, when the same transformation is executed a number of times,

the results obtained exhibit strong similarities at the *meso-time*[25] scale but differ at the level of individual sound-events.

Sound-groups generated in this fashion can be made up of several hundred events running over periods of a few seconds to a few minutes. While each event is stationary and distinct, the overall effect of so many overlapping sounds can give a strong sense of spatial motion and the distribution of sounds around a finite spatial locus can suggest the presence of a single body as opposed to a mass of discrete points. At times, these impressions of motion and physical volume can be sustained even when traversing the acoustically hollow centre of a multichannel speaker array.

1.1.6 Identifying sound-groups

Some sound-groups can be discerned from their spectrograms.
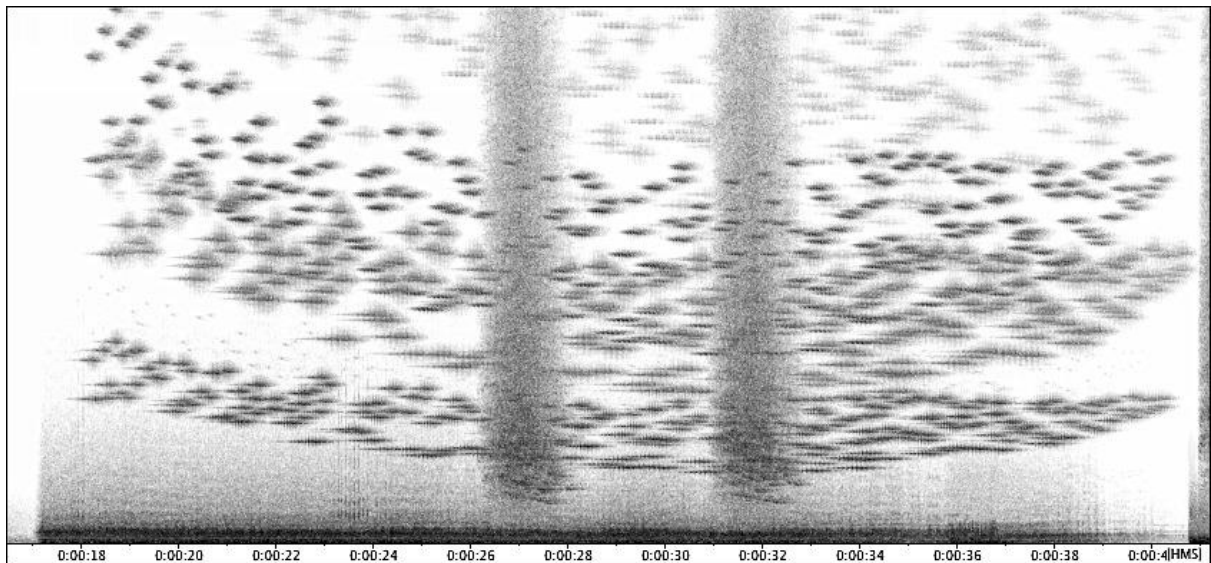
Figure 6: November Spectrogram (detail)

Consider, for example, the group of sounds in *November* between 00:17 and 00:39 (Figure 6). The component events combine to create a scintillating high-pitched sound, reminiscent of small birds twittering, that moves around the sound space. It may not be clear from listening to the

---

[25] Time periods measured in seconds that govern the local (as opposed to global) structure, see Curtis Roads, *Microsound*, (Cambridge USA: MIT Press 2001) p.14.

piece what process is responsible for this sound but, on the spectrogram, individual events can be clearly identified by their harmonic content, appearing as columns of discrete disc–shaped objects, varying in vertical position with resampling rate.

Sound Example 1: November Extract (13s)
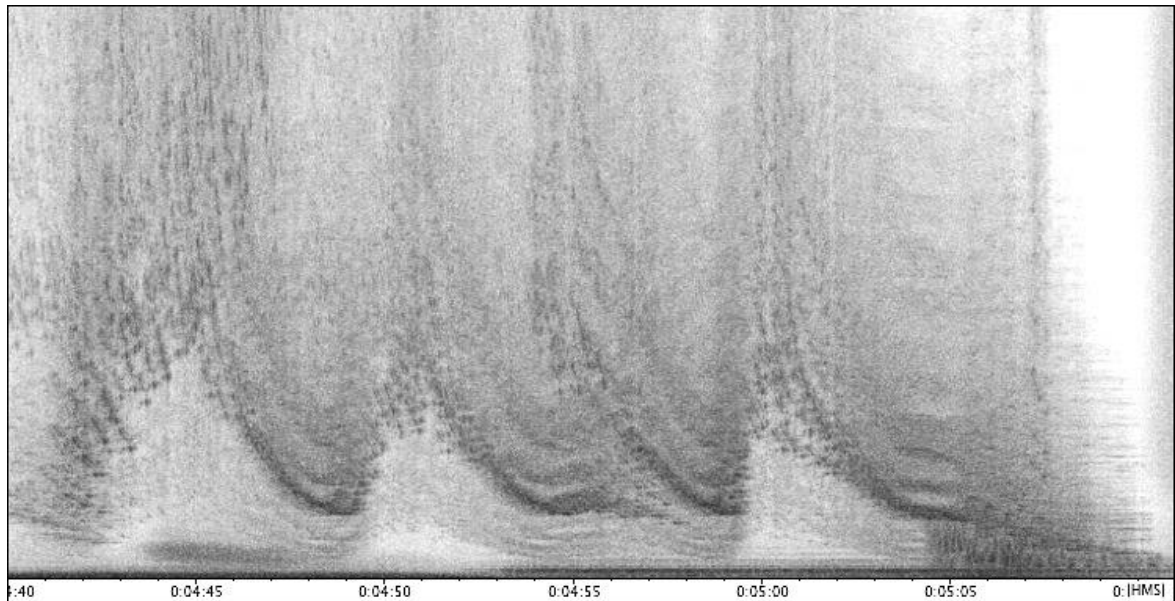
Figure 7: In Flight Spectrogram (detail)

Noisy groups can also be identified. In the piece, *In Flight,* between 04:45 and 05:10 (Figure 7), the four similar groups of ceramic sounds that swoop down from high to low frequencies then rapidly rise again can be seen as descending dark lines. To the ear, these might be confused with a single electronically filtered noise but the dotted patterns in the spectrogram reveal the particulate nature of the mesostructures.

Sound Example 2: In Flight Extract (25s)

1.2 Works

1.2.1 November (2013)

Programme Note

Many people are talking about the First World War these days.

Red poppies, white poppies, mud, death, sacrifice and scarred lives.

My grandparents' generation.

This did not start out as a narrative piece.

I made the sounds and the story arrived unannounced.

Discussion

November was composed as an entry for the 60x60 Surround competition curated by Hans Tammen,
then of Harvestworks in New York City, for Vox Novus. In a 60x60 concert, 60 works are presented
continuously over 60 minutes and synchronized with an analogue clock. Works are therefore
restricted to 60 seconds or less in duration and, in the case of this 'surround mix', limited to a 5.1
loudspeaker format or smaller (I elected to use 4.0). The programme note had similarly to be quite
short.

When the call for this event came out in late 2013, it was an ideal opportunity to develop and release
a small piece composed with the first working version of Audio Spray Gun. Also, at that time, there
was much talk in the media about all the coming centenaries associated with the First World War.

The piece was composed using eighteen (?) four-channel sound-groups created from sources in my
existing sound library and recorded directly into Ableton Live. At this early stage, it was unclear what
the resulting material would sound like but, to me, the groups produced were reminiscent of distant
explosions, bird flight, a shell falling and the adage that war is "long periods of boredom punctuated
by moments of terror". It was from this that the structure emerged.

To me, the piece sounds like early morning with bird flight and distant gunfire, a few seconds of utter

disaster, then birds again, fluttering away.

Performances

Date	Event	Location	Comment
16/05/2014	60x60 Surround	Harvestworks, NYC, USA	Premiere
29/10/2014	60x60 Surround	Jack Straw New Media Gallery, Seattle USA	
25/10/2015	Vox Novus Festival	Spectrum, NYC, USA	
06/01/2016	60x60 Surround	CUBA, Münster, Germany	
04/12/2016	60x60 Surround	CUBA, Münster, Germany	

Format

4.0 surround.

Duration

1 minute 0 seconds.
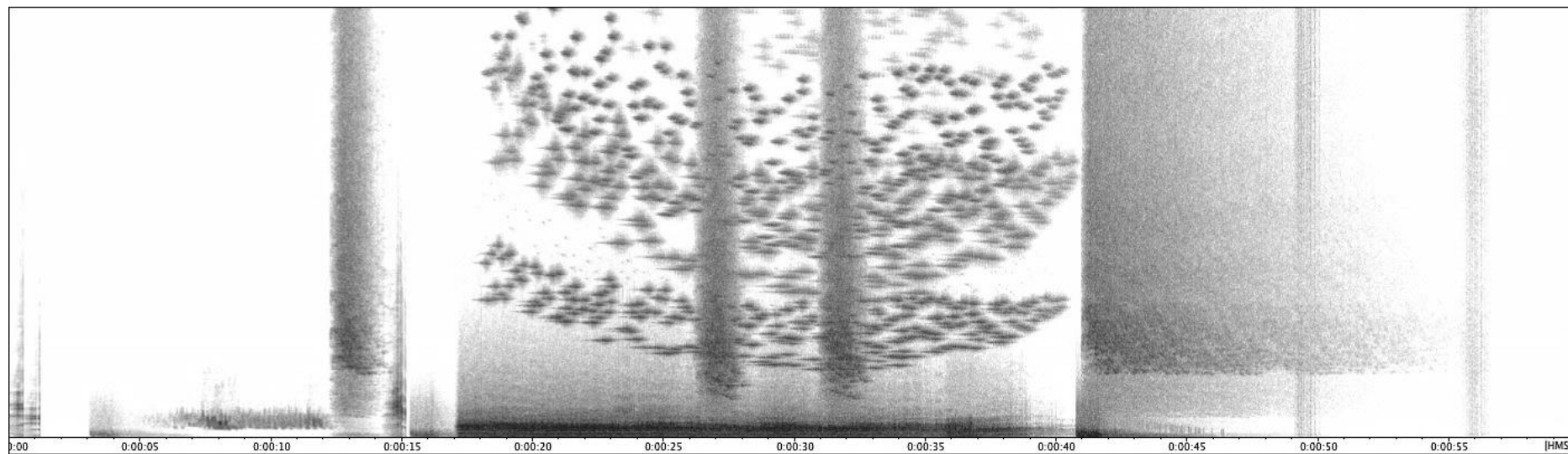
Figure 8: November (2013)

1.2.2 In Flight (2013)

Programme Note

The flapping sounds at the very start of this piece first appeared in an earlier, shorter work, called *November*. While it was not appropriate to develop them there, I was intrigued to see how they could be extended in another context.

As the piece develops, various "noisy" materials are introduced one at a time and then intermingled with earlier sound-ideas. Thus, the piece develops from sparse events, through larger event-groups of similar materials, to a storm of overlapping noise-based textures.

Discussion

This work commences with the two 'bird flight' sound-groups which feature at 00:47 and 00:55 in *November*. These very sparse sequences of sounds, distributed along straight-line trajectories gave the impression of objects moving from front to back across the two-dimensional space. The effect was reminiscent of ravens in slow flight – wings flapping only occasionally but with a flight path that can easily be discerned simply by listening.

Sound Example 3: In Flight - Birds (5s)

With these sounds as its starting point, the piece became an investigation into the effects of employing different delta time ranges across spatial trajectories. Noisy sounds were chosen as source material for their good directionality. The pieces uses numerous eight-channel sound-groups ranging from small, sparse units with high delta times (like the 'bird flight' groups) to large systems of events, which so overlap in time as to produce near-continuous streams of sound moving around the space. Some groups exhibit progressions from one extreme to other (Sound Example 4).

Sound Example 4: In Flight - Sparse to Thick (24s)

Another technical development introduced here was the option in Audio Spray Gun to delay events from sounding by a factor proportional to their distance from the listener in parameter space. This delay is approximately equivalent to the speed of sound in air ($340ms^{-1}$). *In Flight* employs spatial trajectories that exceed this speed to produce 'whip crack' effects as heard at 04:36 and elsewhere.

Sound Example 5: In Flight - Whip Crack (4s)

In addition to the eight-channel version included here, a twelve-channel version of *In Flight* was performed in Berlin as part of a programme[26] of pieces by over fifty composers from all over the world on a twelve channel solar-powered, wireless-networked audio system called Urban Solar Audio Plant (USAP).

Performances

Date	Event	Location	Comment
19/06/2014	USAP	Aufbau Haus am Moritzplatz, Berlin	Premiere (12 ch)
27 - 29/06/2014	USAP	Tempelhofer Feld, Berlin Tempelhof.	12 channels
04 – 07/07/2014	USAP	James-Simon-Park, Berlin Mitte	12 channels
16/08/2014	Toronto International Electroacoustic Symposium	Wychwood Theatre, Toronto, Canada	Lecture- Recital (8 channels)
08/01/2015	RMA postgraduate conference	Bristol University, UK	8 channels

Format

8.0 surround.

Duration

6 minutes 36 seconds.

---

[26] Program at <http://issuu.com/c60collaboratorium/docs/c60_collaboratorium_usap_berlin_201> last checked 18/6/14

In Flight
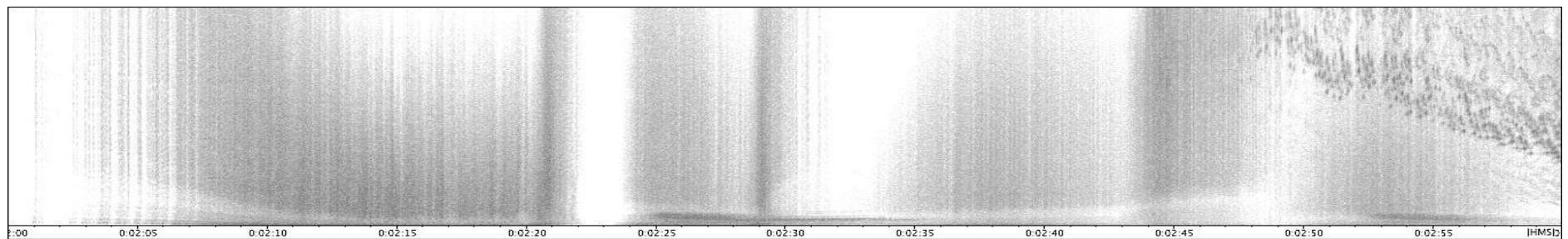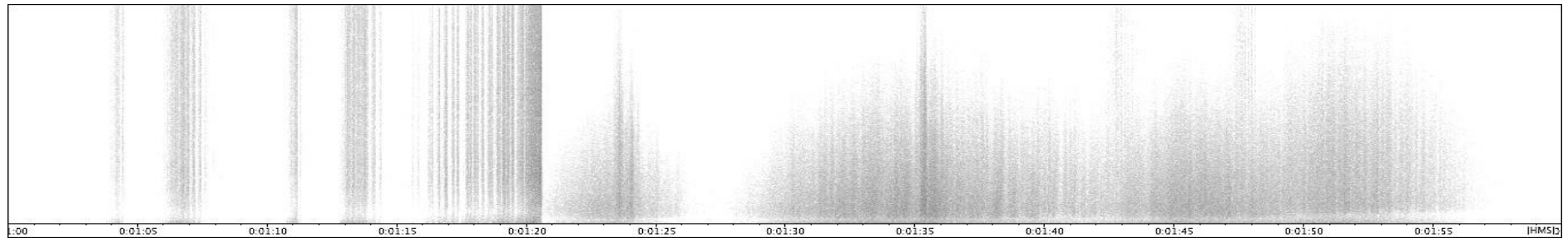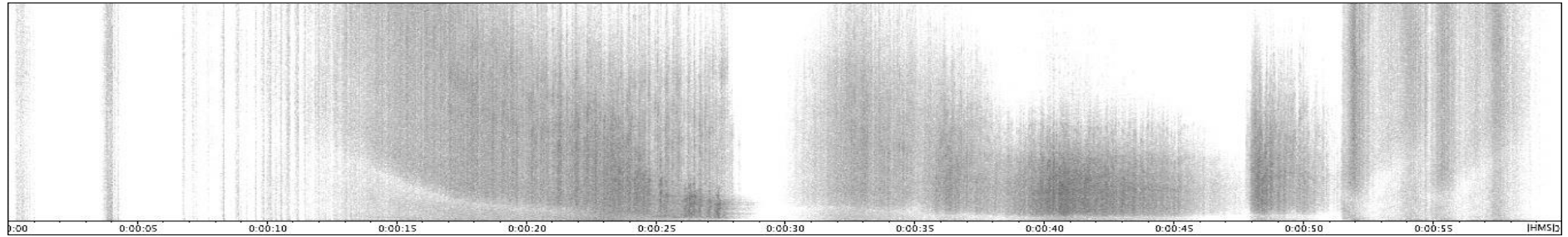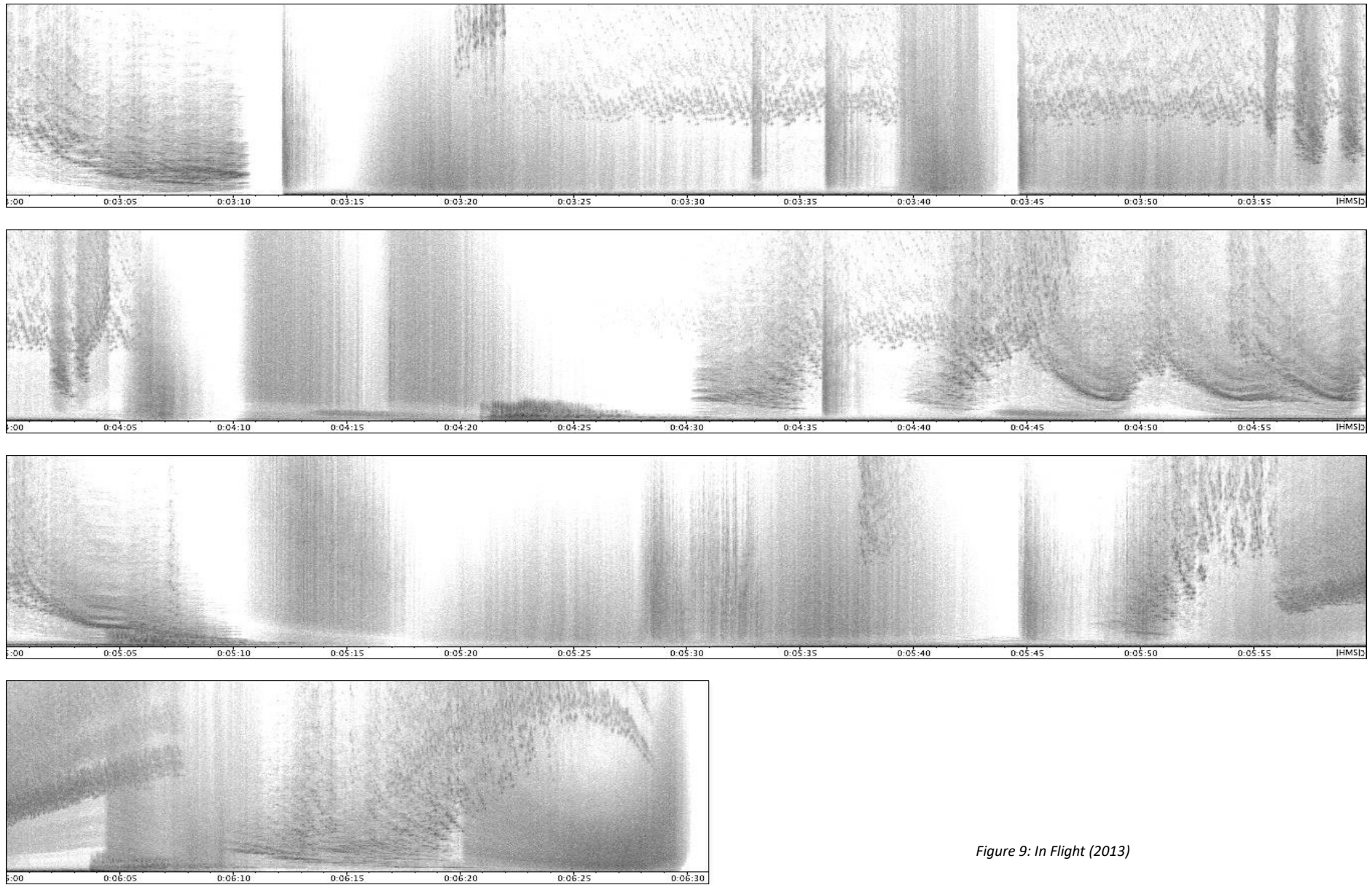
Figure 9: In Flight (2013)

26

1.2.3 Far Skies (2014)

Programme Note

On the Voyager space probe, launched by NASA in 1977 and now travelling beyond the edge of our

Solar System, there is a golden disc containing sounds and images of the Earth. On this disc, along

with music by Bach, Beethoven and Chuck Berry, there are sound recordings in fifty-five human

languages offering greetings to any alien being who should chance upon it. According to the NASA/

JPL website,[27] the Persian greeting translates as:

"Hello to residents of far skies"

This piece is composed of "Earth sounds": sonic representations (known as *sferics*, *tweeks*, *whistlers*,

chorus, and *hiss*) of very low frequency radio signals produced in the Earth's atmosphere, ionosphere

and magnetosphere. These emissions occur as the result of lightning, incoming meteorites and other

natural phenomena.

Perhaps there are other planets, with atmospheres similar to our own, in whose "far skies", similar

music can be heard – if anyone is there to listen.

Discussion

Far Skies was written as an entry to an open competition organised by the MAARBLE (Monitoring,

Analyzing and Assessing Radiation Belt Loss and Energization) project as part of their outreach

program.

The competition brief was that composers should use the natural sounds of the Earth's

magnetosphere in order to compose a work of electroacoustic music (of maximum 10 minutes

---

[27] <http://voyager.jpl.nasa.gov/spacecraft/greetings.html> last accessed June 6, 2014.

duration).[28] To this end, MAARBLE provided a large sample of magnetospheric sounds (as WAV files) compiled through the Canadian Array for Realtime Investigations of Magnetic Activity (CARISMA) at the University of Alberta.[29] Composers were required to use some of these sounds to form a prominent part of the composition.

A structure for the piece was obtained by time-stretching three bursts of sferics and placing markers in a DAW over the loudest clicks. The time-stretched audio was then discarded and the markers used to suggest event locations. The original three bursts can be heard between 0:17 and 0:27 in the recording.

While most of the sounds heard are largely untreated samples extracted from the supplied material, some, more dense, sound-groups were created using *Audio Spray Gun 0.4*. The samples provided were, by their nature, full of hisses and clicks that would be unacceptable in most acousmatic pieces but, here, it seemed appropriate to give them centre stage.

Performances

This work has yet to be performed in public.

Format

Stereo.

Duration

6 minutes 17 seconds.

---

[28]Full brief at <http://www.maarble.eu/outreach/index.php/multimedia/sounds-of-space> last accessed June 6, 2014.
[29] <http://bluebird.physics.ualberta.ca/carisma/semiconductor/FullRangeFiles/> and
<http://bluebird.physics.ualberta.ca/carisma/semiconductor/100Hz_FullRange/> last accessed June 6, 2014.
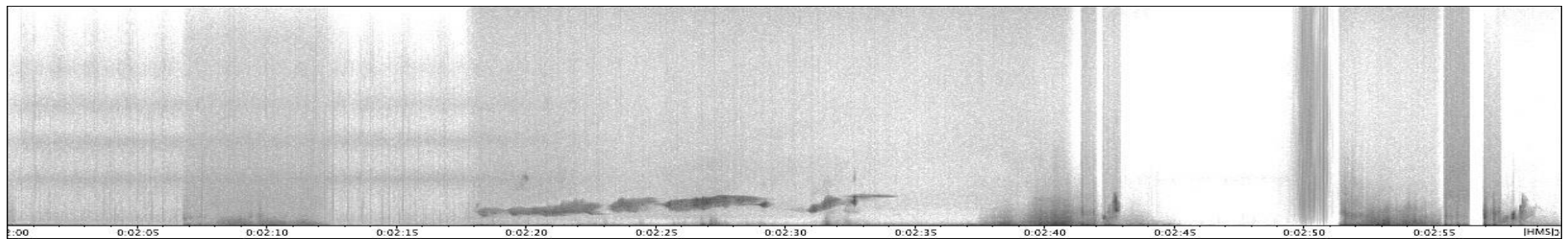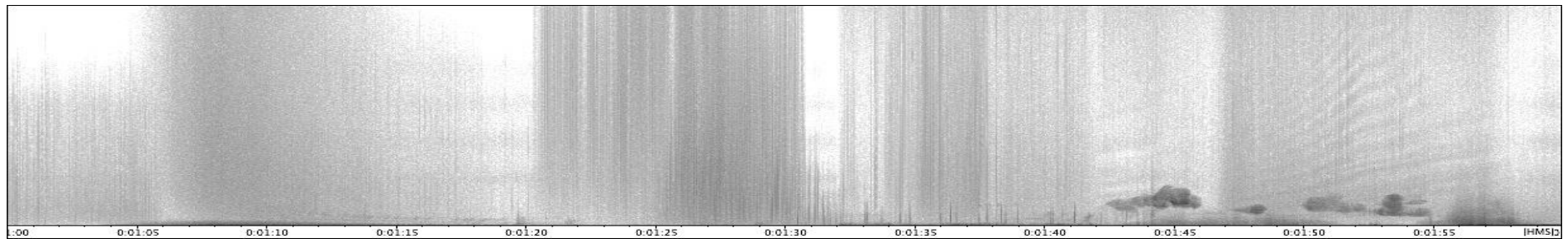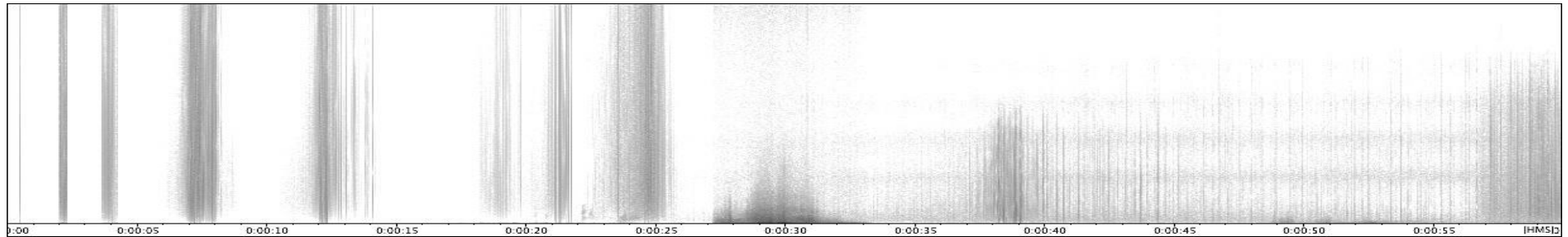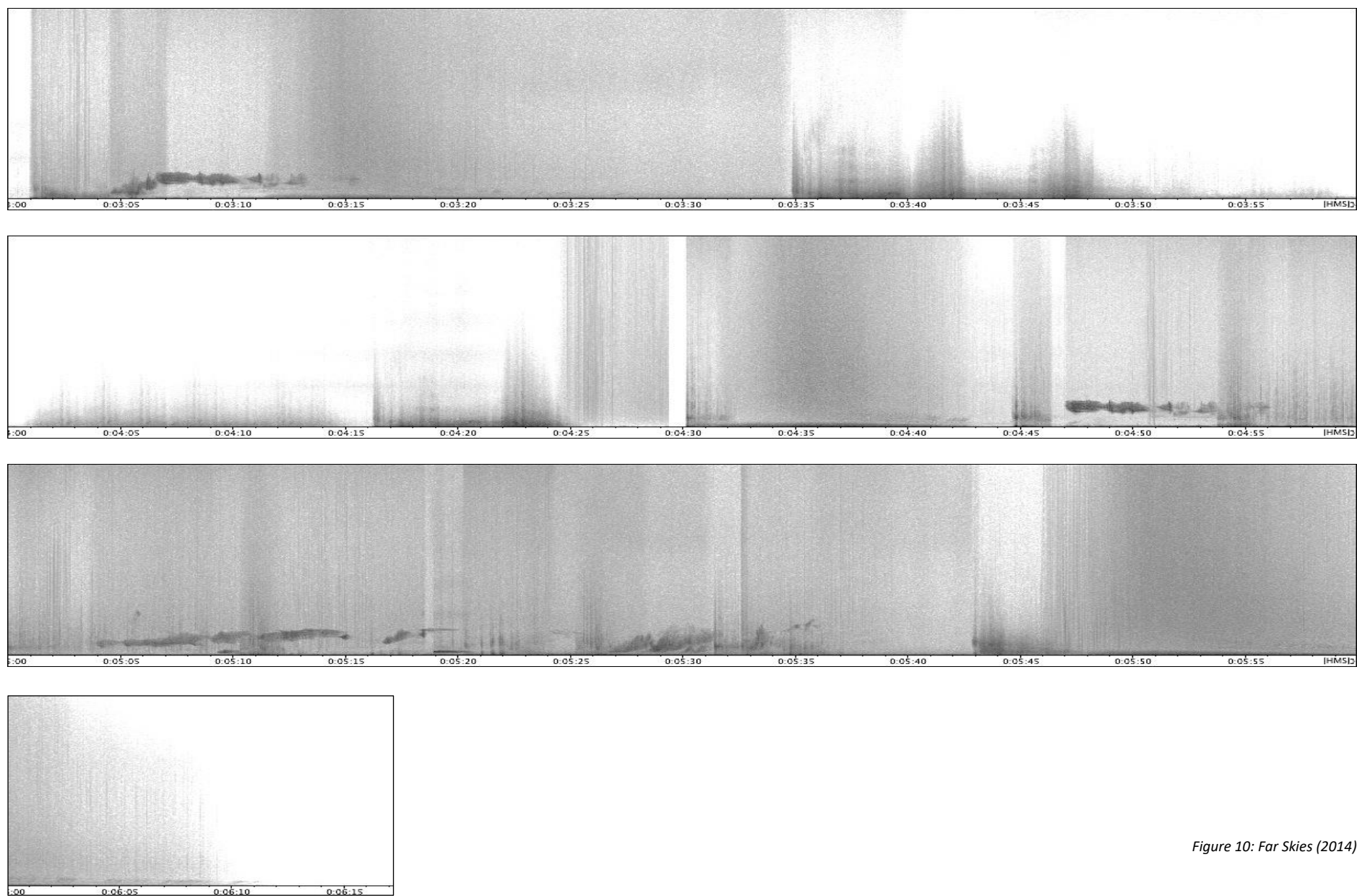
Figure 10: Far Skies (2014)

Chapter 2. Agents and Feature extraction pt. 1: Exploration

Exploration I (2014) and II (2015)

2.1 Exploration

2.1.1 Duets

The second strand of this PhD study looks at the use of fuzzy membership sets to categorise audio events in terms of descriptors,[30] and of fuzzy rules combined with probability to control software agents. The two works produced by this approach were both semi-improvised duets for an instrument and a laptop performer.

In each of these pieces, the players follow a score suggesting actions to be performed. While the instrumentalist plays, the laptop player adjusts on-screen controls with either a mouse or an iPad running MIRA.[31]

Both works use the same Max[32] patch (see page 122) to record and segment audio signals from the live instrument, categorising each segment in terms of a number of descriptors as it does so. Four agents, presented as *channels*, then select events for the accompaniment using fuzzy rules by the method described on page 108. The laptop player changes the selection criteria used by each agent and the probability that it will sound in real time throughout the piece.

---

[30] Geoffroy Peeters, 'A large set of audio features for sound description (similarity and classification) in the CUIDADO project', <http://recherche.ircam.fr/equipes/analyse-synthese/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf> last accessed 9 May 2016.
[31] 'Mira: touch Controller iPad App for Max', Cycling 74, < https://cycling74.com/products/mira/#.V0r8Tktr7fN> last accessed 29 May 2016.
[32] Max, Cycling 74 <http://cycling74.com/> last accessed 22 May 2016

2.1.2 Instrumental Part

Each instrumental event in the piece is described by a number of symbols enclosed within a pair of brackets (Figure 11). The number outside the brackets shows how many times the event should be repeated. Repeated events should be non-identical and varied according to taste. If a range of numbers appears outside the brackets, the player can choose how many repetitions to perform within those limits.
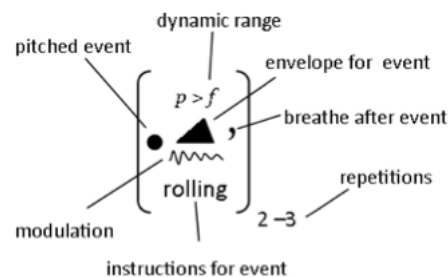
Figure 11: Exploration Score Event

The meanings of the symbols brackets are as follows:

- a *dot* or a *cross* represents a pitched or percussive event respectively;

- *black triangles* etc. describe a dynamic envelope for the event whose range is shown by the dynamics markings above it;

- *wavy lines* suggest modulation.

If a score event includes a comma, the player should leave a breath each event. A slur indicates places where many short events should be run together. Other symbols (staccato, trill and so on) should be interpreted as normal.

In the flute piece, wherever the word 'sing' appears, this should be interpreted as 'sing and play'. Symbols appearing below the brackets represent changes taking place over the total period of the repetitions.
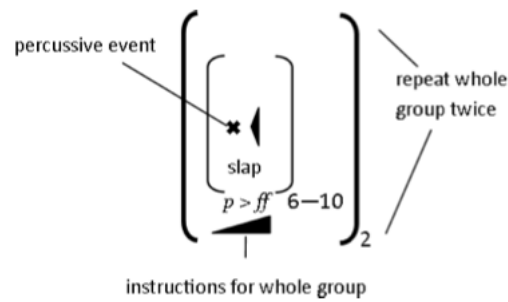
Figure 12: Exploration Score Group

When an event or group of events is nested in another pair of brackets (Figure 12), the contents should be repeated the number of times indicated.

Neither the sizes of the events on the page nor the length of the gaps between them indicate fixed time periods. Only the thick external brackets surrounding the large groups suggest a duration based on a period of 30 to 40 seconds between the vertical lines shown on the score. If these thick brackets are to be played more than once, the brackets suggest the total duration of all repetitions.

2.1.3 Laptop Part

The principal software controls for this piece are five on-screen faders. Four of these control the chance that a given voice will sound, while the fifth controls the density of events available to the accompaniment at any given time.

The top four rows in the laptop section of the score suggest the chance settings that should be employed at a given point in the piece. The darker the horizontal bar in each row, the higher that channel's chance fader should be set. The yellow chance buttons can turn channels on and off as required.

The symbols in the bottom row represent changes to the density rules (see page 104) governing all channels. As the density slider moves up, the number of simultaneous voices available to the accompaniment increases from $[0,1]$ (one at a time) to $[4,4]$ (everyone at once). For a state [x, y], x represents the number of active agents below which voices are free to sound and $y,$ the number above which voices remain silent. As the number of voices increases between x and $y,$ the chance of a voice sounding decreases. The actual likelihood of a given voice sounding is constrained by the combination of its chance control and the density slider.

In the score, a short arrow next to a state value means move the fader to this state immediately. A longer arrow between two states means move to this state over the time indicated.

2.2 Works

2.2.1 Exploration for flute and laptop (2014)

Programme Note

Exploration is a semi-improvised duet for flute and computer.

As the piece begins, the flautist, following a graphic score, plays short groups of discrete sound events, pausing between groups. These sounds are recorded by the computer and categorised using audio descriptors that may be thought to represent qualities like 'duration', 'loudness', 'brightness' and so on. During the pauses between events, four software agents select sounds from the flute sounds played so far to build an accompaniment.

Meanwhile, the computer operator, also following a score, adjusts the probability that each agent will sound and modifies the filtration, resampling rate and envelopes applied to sounds chosen by each one. He or she may also adjust the criteria by which the agents select sounds if required.

As the piece develops and the program adds more sounds to its *corpus* of events, the flautist and the computer sounds gradually overlap. At an agreed signal, there is a pause after which the computer operator takes time to adjust the agents' settings and further develop an accompaniment over which the flautist can improvise freely. During this improvisation, no new flute sounds are added to the corpus, though they are still recorded.

The piece finishes with both players improvising together with the computer using all the sounds recorded during the piece.

Discussion

To compose this piece, I first recorded a variety of sounds with flautist, Sean Goldthorpe. The more interesting samples were then edited together leaving sizeable gaps between each one to produce a test file that could be fed into the Max patch in emulation of a real player. A period of experimentation followed, testing out possible sounds for the accompaniment and re-editing the test file a number of times before the final structure evolved. The score was then written out from this version.

Exploration was played in the workshop at INTER/actions 2014 in Bangor University on the 15th March 2014. It was then selected for performance in the closing concert of the festival on the 16th March with Richard Craig playing flute and the composer operating the laptop (this performance is included in the portfolio). A lack of available rehearsal time (for the laptop player in particular) is reflected in some of more repetitious elements towards the end of the recording.

Performances

Date	Event	Location	Comment
15/03/2014	INTER/actions 2014	Bangor University, UK	Workshop, Richard Craig, (fl.)
16/03/2014	INTER/actions 2014	Bangor University, UK	Premiere, Richard Craig, (fl.)
27/08/2015	Flute Fest	Bangor University, UK	Ellie Lighton, (fl.)
20/11/2015	PhD recital	Bristol University, UK	Ellie Lighton, (fl.)

Format

Stereo.

Duration

8 to 10 minutes.

B

C

flute

sparse

Flute improvise *ad lib* over laptop material

ch.1

ch.2

laptop

ch.3

ch.4

density

[0,2]   [4,4]   [0,4]   change density *ad lib* to support flute

LIMIT on

D

E

flute

busy   *v. busy*   *stop*

Flute improvise *ad lib* over laptop material

ch.1

ch.2

laptop

ch.3

ch.4

density

ad lib   [4,4]   [0,1]

LIMIT off   PLAY off

Figure 13: Exploration, for Flute and Laptop (2014)

2.2.2 Exploration II for clarinet and laptop (2015)

Programme Note

Exploration II is a semi-improvised duet for clarinet and computer.

As the piece begins, the clarinettist, following a graphic score, plays short groups of discrete sound events, pausing between groups. These sounds are recorded by the computer and categorised using audio descriptors that may be thought to represent qualities like 'duration', 'loudness', 'brightness' and so on. During the pauses between events, four software agents select sounds from the clarinet sounds played so far to build an accompaniment.

Meanwhile, the computer operator, also following a score, uses an iPad to adjust the probability that each agent will sound and modifies the filtration, resampling rate and envelopes applied to sounds chosen by each one. He or she may also adjust the criteria by which the agents select sounds if required.

As the piece develops and the program adds more sounds to its *corpus* of events, the clarinettist and the computer sounds gradually overlap. At an agreed signal, there is a pause after which the computer operator takes time to adjust the agents' settings and further develop an accompaniment over which the player can improvise freely. During this improvisation, no new clarinet sounds are added to the corpus, though they are still recorded.

The piece finishes with both players improvising together with the computer using all the sounds recorded during the piece.

Discussion

This piece was composed in the same fashion as *Exploration*, starting with a recording session with Hephzibah Leafe, from which a number of clarinet sounds were extracted and arranged to create a test file. After much experimentation with the software and multiple modifications to the test file, a version was obtained which could be used to write the score.

The portfolio version of this piece is an unedited take recorded in Studio 1, Bangor University on 9th March 2015 with Hephzibah Leafe on clarinet. Unlike the flute recording, this piece was developed over a number of recording sessions allowing a more consistent interpretation to evolve.

Performances

Date	Event	Location	Comment
23/04/2015	Electroacoustic Wales	Bangor University, UK	Hephzibah Leafe, (cl.)
27/08/2015	Flute Fest	Bangor University, UK	Hephzibah Leafe, (cl.)
20/11/2015	MA recital	Bangor University, UK	Hephzibah Leafe, (cl.)

Format

Stereo.

Duration

8 to 10 minutes.

(Recorded version: 9 minutes 8 seconds)

A *gently*

clarinet			
"bell tone" short 4-6	"bell tone" long 1-2 1	breath long 1	

ch.1 rate: 1200. +/- 0. att: 0.7, rel: 0.3   short, high, pitched   resampling +/- 1200

ch.2 resample: 1200. +/- 1200. res. filter: 1024Hz +/- 50%, Q: 2 +/- 50% att: 0.5, rel: 0.3   noisy, sparse

laptop ch.3 resample: 700. +/- 700. res. filter: 256Hz +/- 50%, Q: 1 +/- 50% att: 0.5, rel: 0.3

ch.4 resample: −2400. +/- 1200. att: 0.5, rel: 0.3

density [0,1]   [0,2]

REC   PLAY   on

---

clarinet

$p > f$ 2 −3 low—middle	1	blowing on keys thing 1	short breath sounds 2-3 1

ch.1

ch.2

laptop ch.3

ch.4 long, low

density [0,3]   [0,4]

41

Figure 14: Exploration II for clarinet and laptop (2015)

Chapter 3. Zones: Fuzzy - probabilistic Loci

The Inside Track (2014)

This chapter deals with a development of the locus model called *Zones*, which adds internal spatial and spectral movement to sounds created by the system.

3.1 Zones Of Attraction And Rejection

In a sound-group created by Audio Spray Gun, each new event occurs at a random position, constrained only by the bounds of the locus and unaffected by the events preceding it. While this is adequate for producing fairly small and/or dense balls of sound, such random motion in larger and more diffuse loci can result in sizeable spectral and/or spatial jumps in the material produced. This effect is not always desirable and while such random behaviour can give a strong impression of 'solidity' to a sound-group, it invests it with almost no internal structure.

In Zones, each new event location is constrained by not only the boundaries of the locus but also by zones of attraction and rejection around the locations of preceding events. This means that, each time an event is created, its position is recorded and the program creates a *rejection zone* about that value which reduces the likelihood of the next point being chosen in close proximity to the present one. Simultaneously, an *attraction zone* stops it straying too far away (Figure 15). These zones, used in combination, create probability tables that determine the location of the next event.
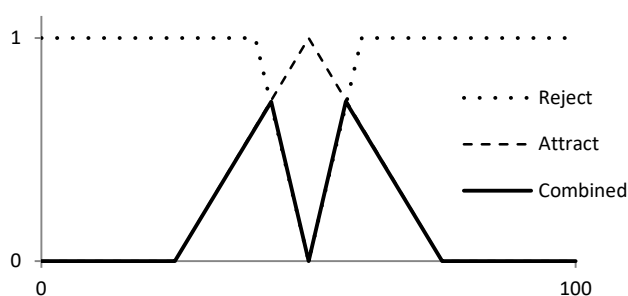
Figure 15: Attraction and Rejection

The effects of this algorithm can be heard in most of the acousmatic compositions in this portfolio from *The Inside Track* onwards, appearing as swirling spatial and spectral motion, sometimes reminiscent of the *flocking*[33] algorithm. This movement is discernably more organised than that produced by Audio Spray Gun.
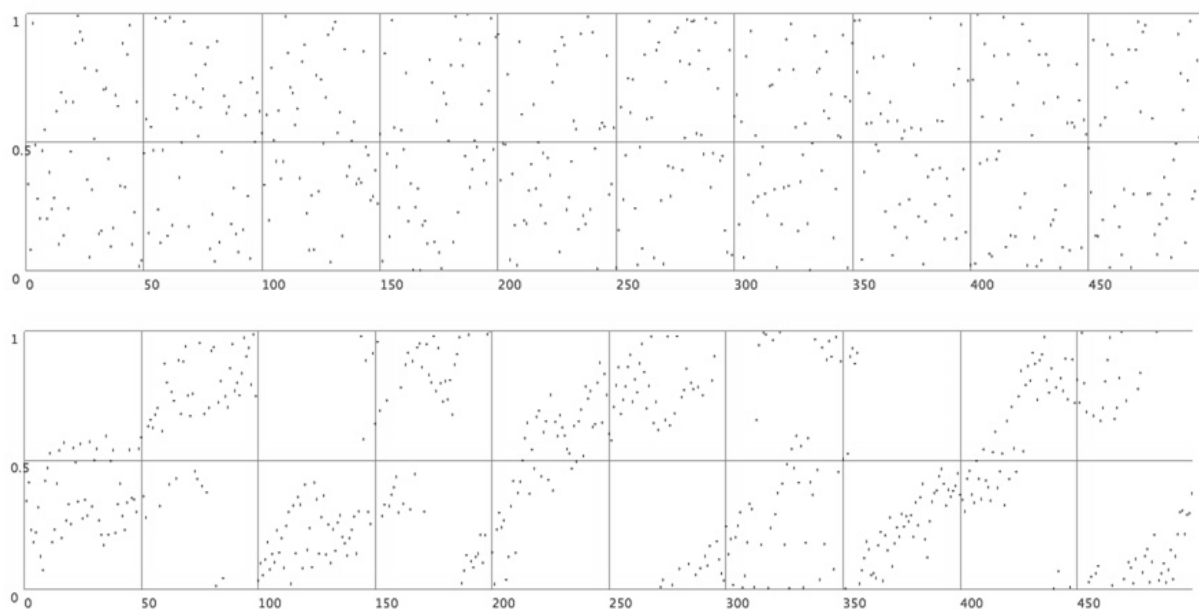
Figure 16: Random and Zones distributions

Figure 16 illustrates the two processes. The upper image shows 500 points in a random distribution as might occur in Audio Spray Gun and the lower (also 500 points) gives an example of the more directed progression typical of Zones.

3.2 Work

3.2.1 The Inside Track (2014)

Programme Note

Development piece.

---

[33] Craig Reynolds 'Flocks, Herds, and Schools: A Distributed Behavioral Model in Computer Graphics', 21(4) *SIGGRAPH '87 Conference Proceedings*, 1987.

Discussion

Unlike other stages in this project, this section did not produce a full concert work. This is because the Zones method used here has shortcomings in that it has no 'memory'. Thus, while the second event in a sequence is rejected from the immediate vicinity of the first, there is nothing to stop the third event returning to the first point. Because the algorithm was in need of further development, it was decided to suspend work on it without composing a more complete piece.

The Inside Track, although a development work, is the only piece in the portfolio composed exclusively with the zones algorithm and is therefore retained in order to demonstrate the progression of ideas. It also offers a useful comparison to the random material produced by Audio Spray Gun.

Here are two examples of Zones output.

In the opening (from 00:10 – 00:35) a sequence of events created from wind chimes can be heard making a meandering ascent in the resampling rate dimension while swirling around the listener in space.

Sound Example 6: Inside Track - Long Swirls (25s)

In the second example (03:41 -03:49), a large number of short noisy samples flicker out from front and centre and spread around the listener. At the same time, a zones effect is applied to the delta time axis causing the feature, which starts off as a very dense cloud of sounds, to become discretised and steadily dissipate.

Sound Example 7: Inside Track - Noisy Swirls (9s)

Performances

This work has yet to be performed.

Format

8.0 surround.

Duration

7 minutes 46 seconds.
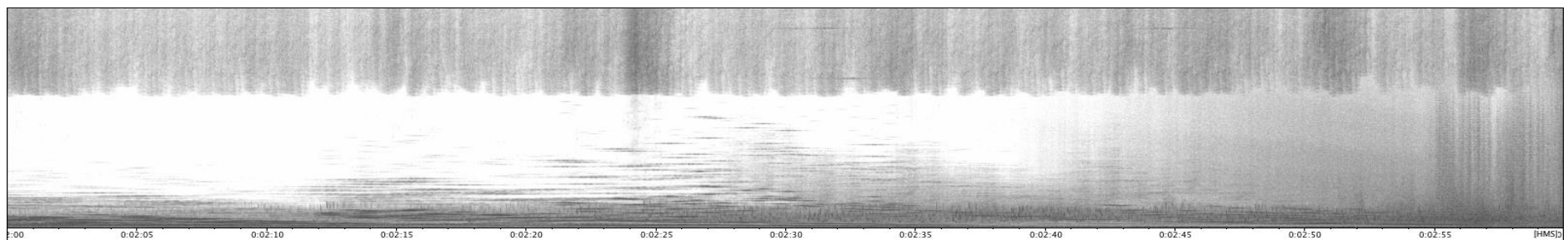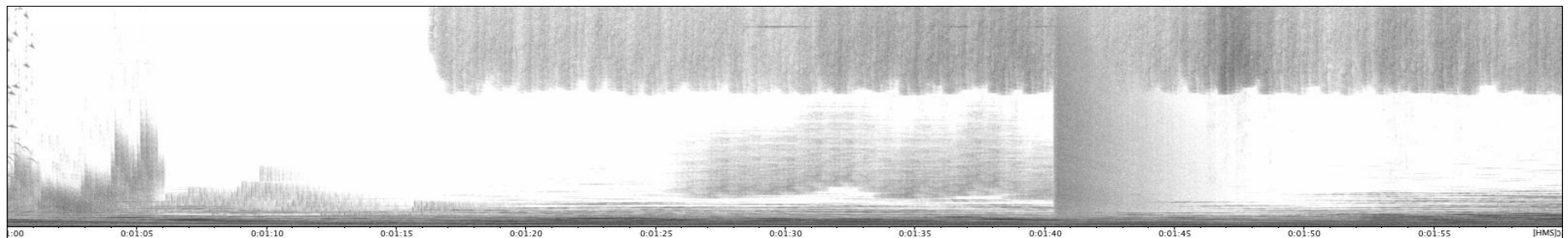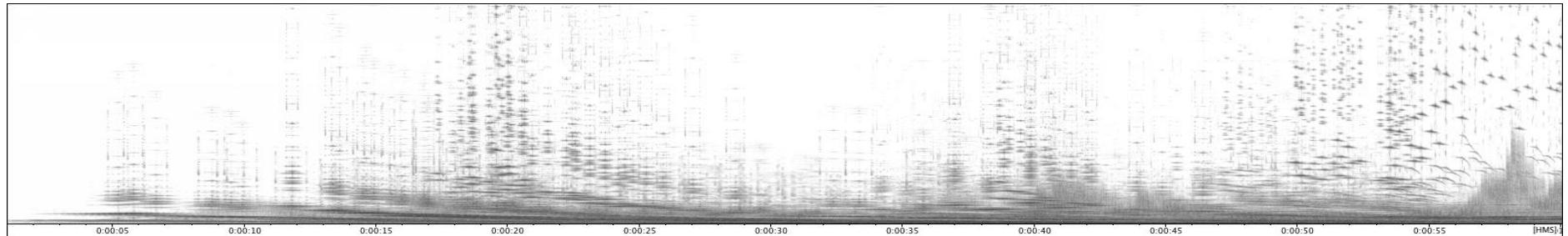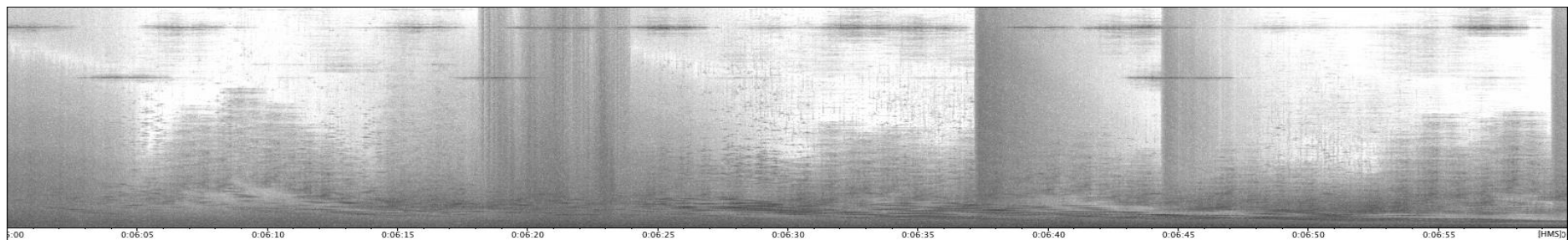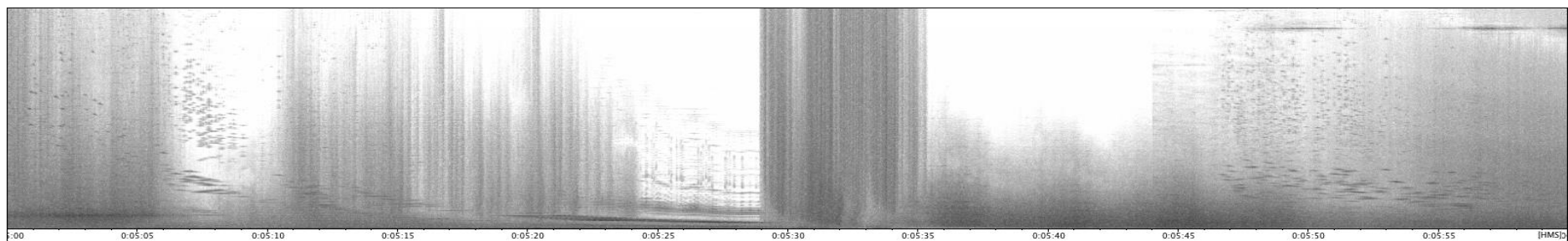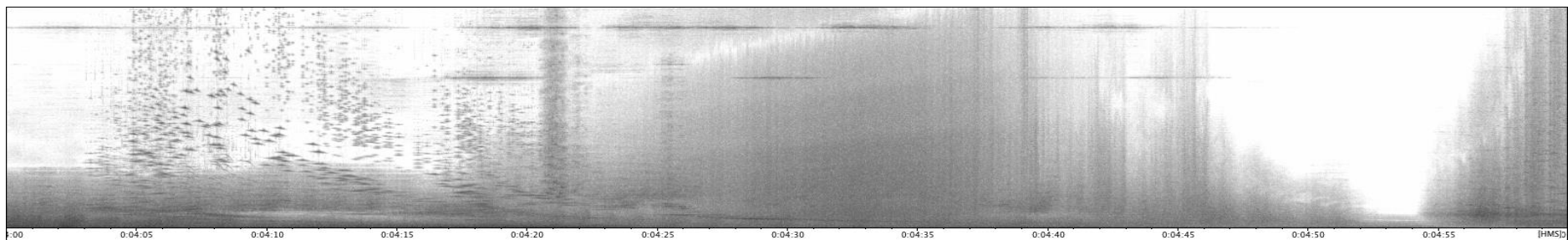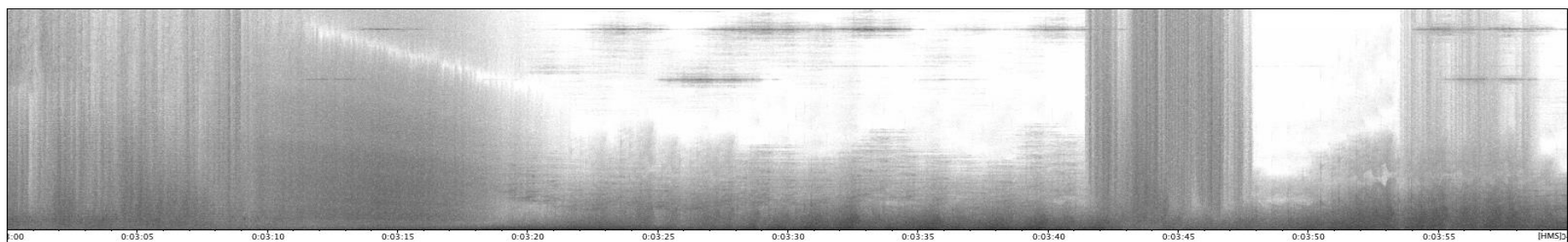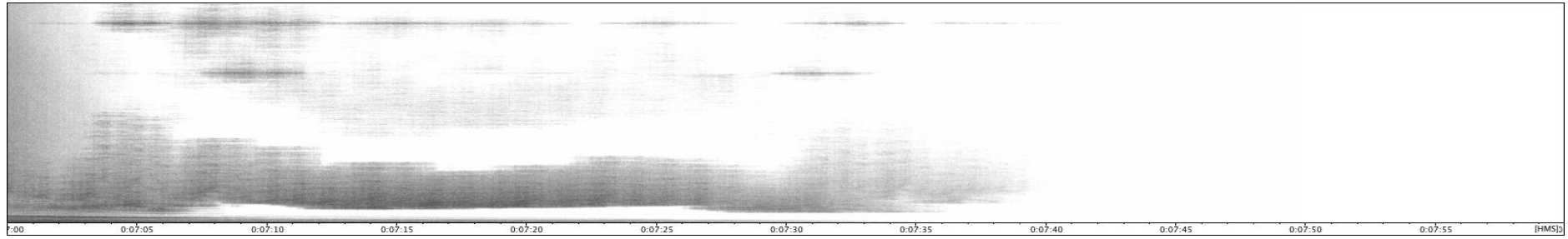
Figure 17: The Inside Track (2014)

49

Chapter 4. A Granular Variation

Once below a Time (2014)

4.1 ASGScan

It has been a recurring aspect of this PhD project to explore different ways of producing source material for sound-groups. In the works created with Audio Spray Gun, this material came from single sound files; in the Exploration pieces, segments were extracted from live audio input and so on. When I had the opportunity to work with recordings of Dylan Thomas reciting his own poetry for the Bangor Dylan Thomas Prize for electroacoustic composition, I wanted my piece to include almost choral sounds derived from the poet's own voice. To do this, I developed another variant of Audio Spray Gun, called *ASGScan,* which derived its source material from short segments extracted from a longer source file.

ASGScan, instead of playing numerous mutations of a single sound, extracts segments from a sample and applies sound-group processes to them by means of two further function generators of the type used in Audio Spray Gun. These control the position and extent of the segment within the source file over time. The function generator for position 'scrubs' back and forth through the source file at varying rates over the duration of the sound-group. The generator for extent varies the size of the segment between about 1% and 100% of the source-file length. To smooth the resulting sound, each segment has an amplitude envelope applied to it, similar to those used in granular synthesis[34] although rarely as short in duration. In some parts of the piece, the process results in a coarse granular time-stretching involving multiple overlapping segments played back at varying resampling rates to give an effect similar to electronic chorusing that suggests multiple voices.

---

[34] 'Three-stage line segment' envelope, see Curtis Roads, *Microsound* p.89

4.2 Work

4.2.1 Once Below A Time (2014)

Programme Note

The source material for this piece is the poem, *Fern Hill*, written by Dylan Thomas in 1945. In this, perhaps his best-known short poem, Thomas presents us with a flow of impressions from childhood visits to his aunt's farm in rural Carmarthenshire during the 1920s. In the poem, he ponders, some say with regret, the transition from the 'timeless' immediacy of youth to the time-bound nature of adult experience.

The structure for *Once Below A Time* was defined by first building a guide track from a 1952 recording of Thomas reading Fern Hill[35]. The poem was divided into individual phrases between which silence was inserted to stretch it to about ten minutes while maintaining their relative onset positions within the piece as a whole. This guide track was used as a substrate upon which were placed sounds derived from Thomas' voice and from environmental sounds related to images of the Welsh landscape that he presents. Initially, sounds inspired by particular phrases were placed at locations within the piece that corresponded to the locations of those phrases within the poem. As the piece became established, both the guide track and this rigid placement of sounds were abandoned so that the final arrangement could emerge.

In Wales, the summer of Dylan Thomas' centenary was one of the finest in many years and seemed, somehow, to resonate with the summers recalled in the poem. Such resonances also have their places within this piece.

---

[35] *Dylan Thomas Reading His Poetry* (audio CD) HarperCollins, London 2008 – used with permission.

Discussion

It is a commonplace to talk about *Fern Hill* as a work of nostalgia, the poet recalling and then lamenting the loss of childhood but, to my ears, it discusses a deeper relationship to Time. The poem seems to deal with the transition from early childhood (a life lived almost in the moment with the recent past long gone and the future miles ahead) to adult life in which years of remembered past and anticipated future bear down all too frequently on a fleeting present. To write this off as mere lament does injustice to a poet who repeatedly revised his work to produce precise written and spoken rhythmic structures and who could not have written Fern Hill without the adult time-sense he is assumed to decry. Yet still, he looks for that life in the moment, singing in his chains.

In this work, I wanted to reflect the multiple approaches in which time is addressed in the poem, among them the way that Thomas combines adult reverie with childhood immediacy and the ways in which exquisitely timed spoken rhythms ("the spellbound horses, walking warm"[36]) express timeless moments. I wanted to give the piece a subtle timeline that followed not only the verses of the written poem, but also the poet's reading on this specific recording. To this end, I first created the guide track described above as a framework upon which to construct the piece. Afterwards, I used the software to construct sound groups inspired by particular phrases and aligned them with those points in the 'expanded' voice track. For example, from 03:29 to 04:58, sounds derived from or reminiscent of water, a fire, owls and nightjars follow this sequence of images from verse three:

> …it was air
> And playing, lovely and watery
> And fire green as grass.
> And nightly under the simple stars
> As I rode to sleep the owls were bearing the farm away,
> All the moon long I heard, blessed among stables, the nightjars
> Flying with the ricks…[37]

---

[36] Dylan Thomas, 'Fern Hill' in Edward Leeson (ed.), *The New Golden Treasury of English Verse* (London: Pan 1980) pp. 475-6
[37] ibid.

Numerous such associations of text and sound exist throughout this work and while their precise locations with respect to the substrate have been allowed to move according to the needs of the piece, their sequence has largely remained intact.

Regarding the choice of sounds for this piece, it seemed in the spirit of the competition to make as much use of Dylan Thomas' voice as I could and, where possible, to make utilise whole words or phrases in ways that would give a sense of him speaking without offering a precise focus on their meaning.

Early experiments with multiple pitch-shifted mutations of untreated voice samples were disappointing, producing 'chipmunk' effects rather than the smooth choir I was looking for. To remedy this, I decided to convolve them, either with short elements of Thomas' speech (*grains*) or with environmental sounds (birdsong, water and so on) using Tom Erbe's *Soundhack*[38]. To produce sustained sounds, I put these modified, less obviously voice-like sources through ASGScan. Here is an example of this process (Figure 18).

First, the phrase "green and golden" from the 1952 recording was convolved with a spoken "ee" grain to produce a slightly electronic-sounding version of the phrase in which consonants were diminished and 'ee' sounds accentuated. Then this was used as source for ASGScan, running linearly from start to finish of the convolved sample, generating multiple sound grains distributed in eight-channel surround, over a period of 25 seconds.

---

[38] Soundhack 0.896 <http://www.soundhack.com/freeware/> last checked 9/0/9/2014

"Green and Golden", Dylan
Thomas 1952 (1.9 s)

"ee" grain, Dylan
Thomas 1952 (0.15 s)

Convolved version (1.9s)
(Soundhack)

Overlapping versions distributed over eight channels (25s)
(ASGScan imported into Live)
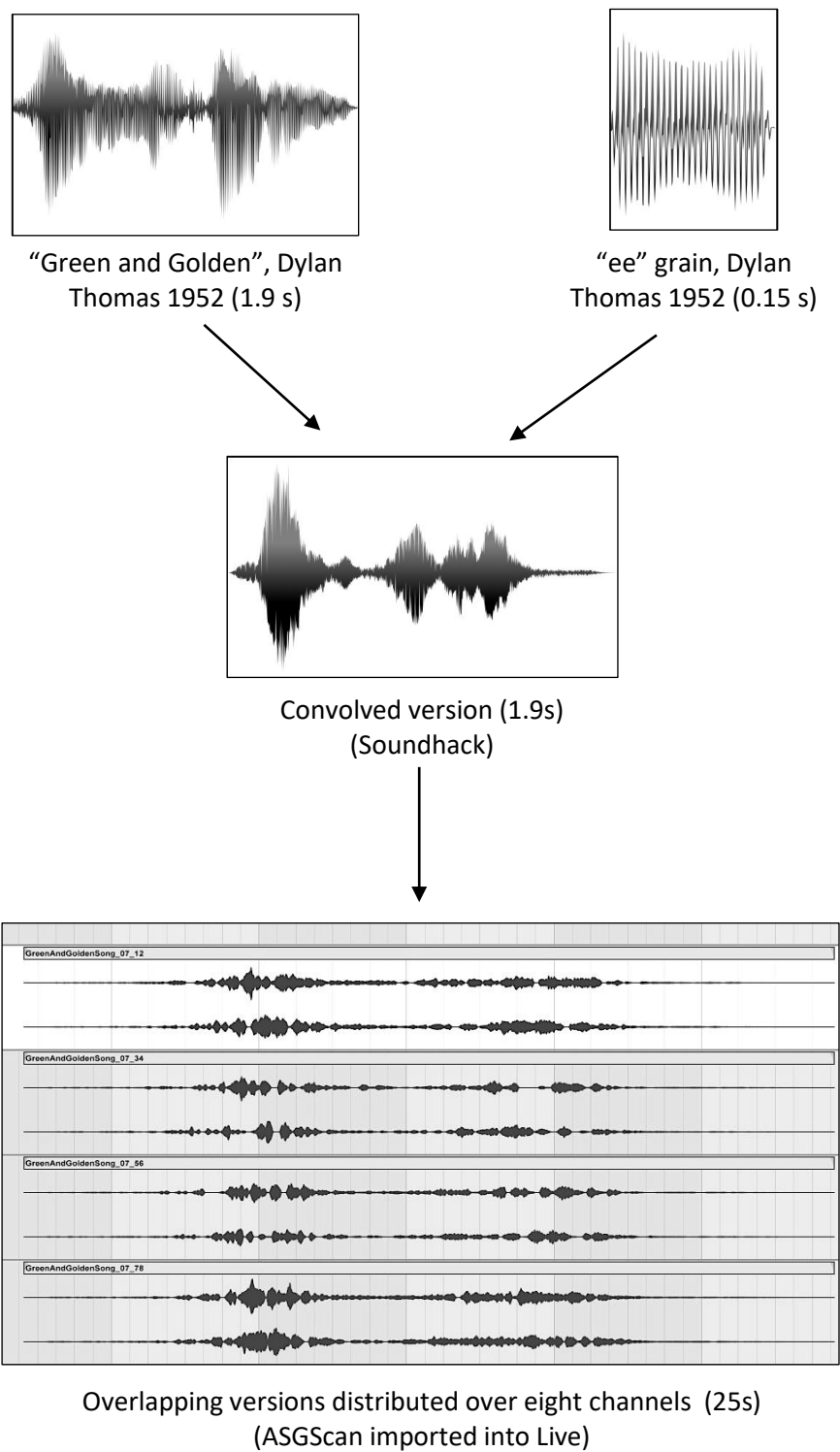
Figure 18: ASGScan and Convolution

Sound Example 8: OBAT - Green and Golden (1.9s)

Sound Example 9: OBAT - ee grain (0.15s)

Sound Example 10: OBAT - convolved (1.9s)

Sound Example 11: OBAT – ASGScan output (25s)

Each step of this process can be heard in sound examples eight to eleven and the finished element can be heard *in situ* at 06:15 of the finished piece.

Here are two other examples of this process. At the beginning of the work (00:20 to 01:20), a drone composed of multiple versions of Thomas' voice convolved with the sound of waves starts in front of the audience and expands to fill the whole surround space, inviting the listener to enter into the poet's dreamy recollection ("Now as I was young and easy..."). [39] Later, between 03:40 and 03:53, there is a sparse cloud of variable length grains designed to sound like a crackling fire and sourced from a 't' sound in the word "night" in the original recording.

Sound Example 12: OBAT - time clicks fire (13s)

Performances

Date	Event	Location	Comment
28/10/2014	Electroacoustic Wales	Bangor University, UK	Premiere
18 - 20/4/2015	Sonorities	Goldsmiths, University of London UK	Rolling programme
22 -26/4/2015	Sonorities	SARC Belfast, UK	Rolling programme

Format

8.0 surround.

Duration

8 minutes 46 seconds.

---

[39] Dylan Thomas, 'Fern Hill' in Leeson, *Golden Treasury*
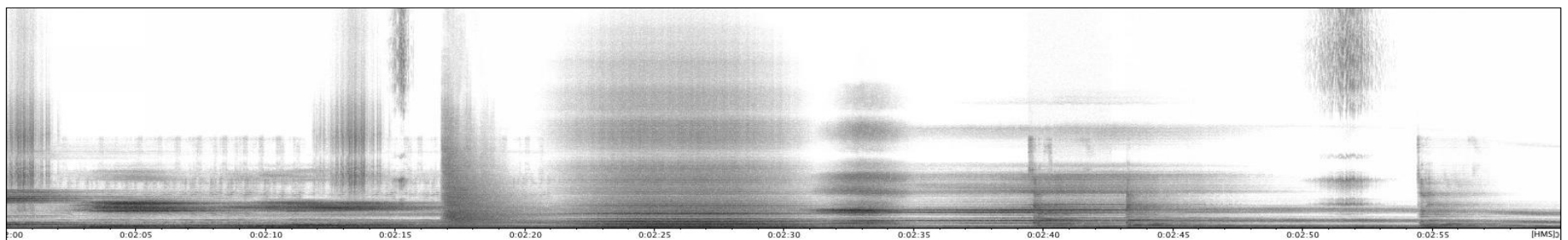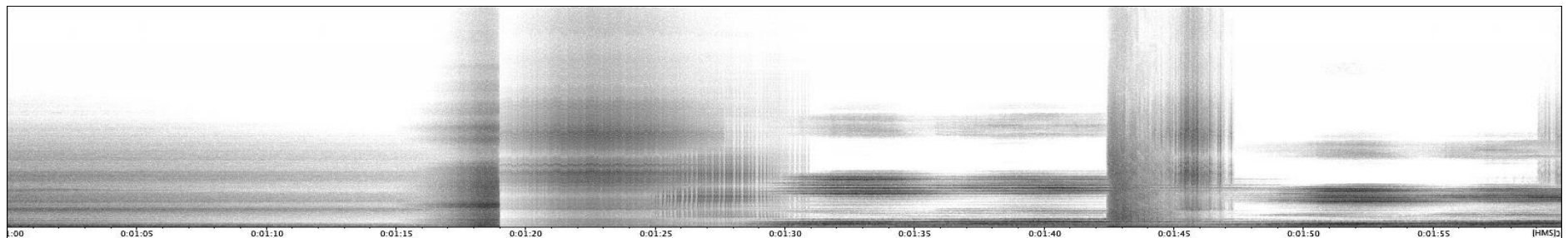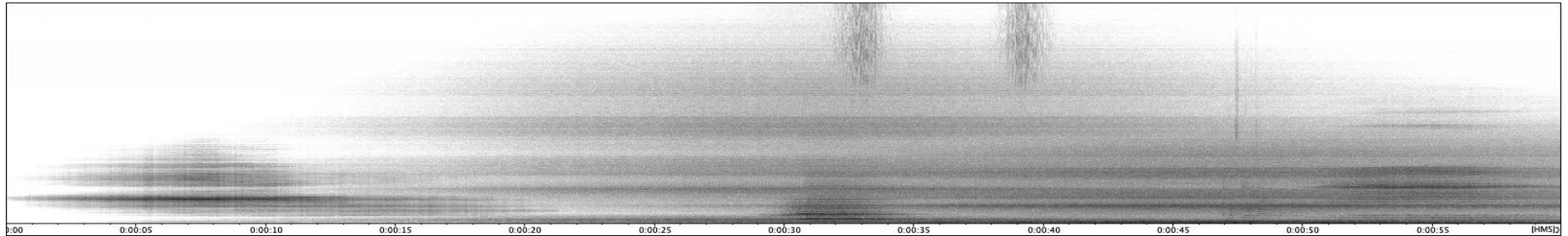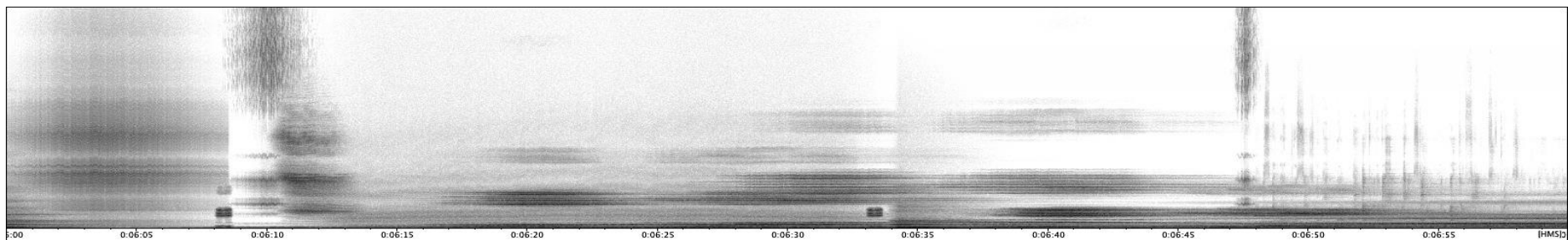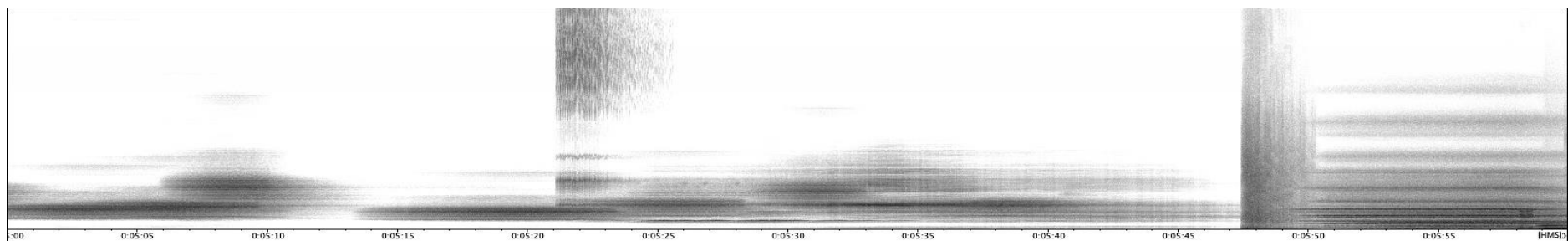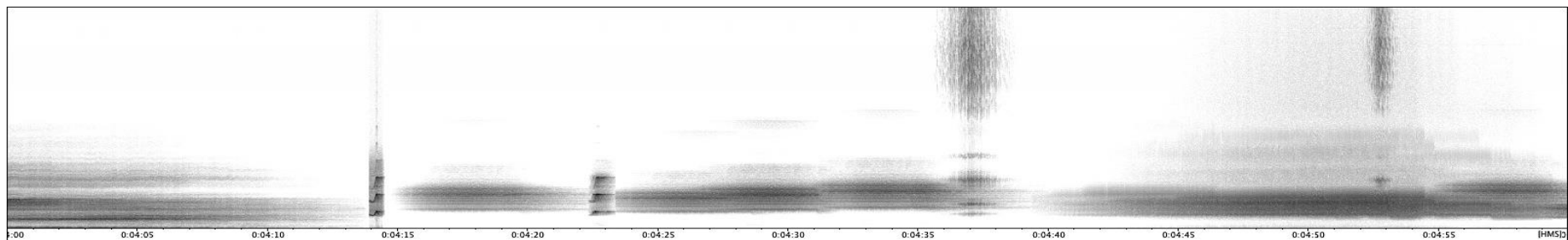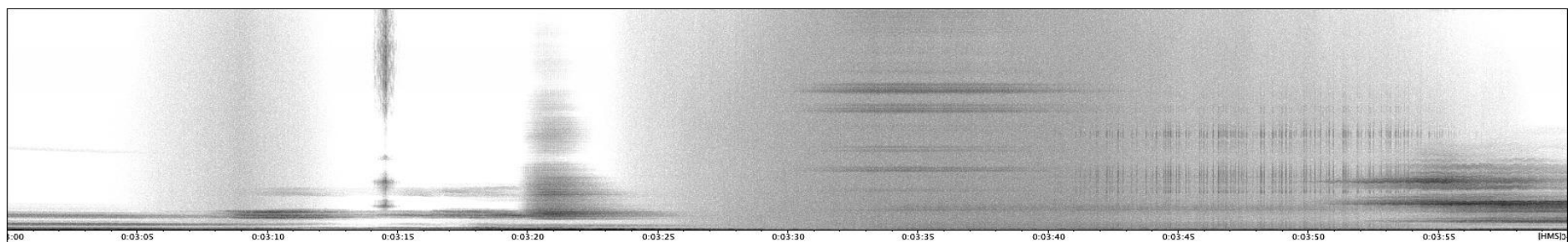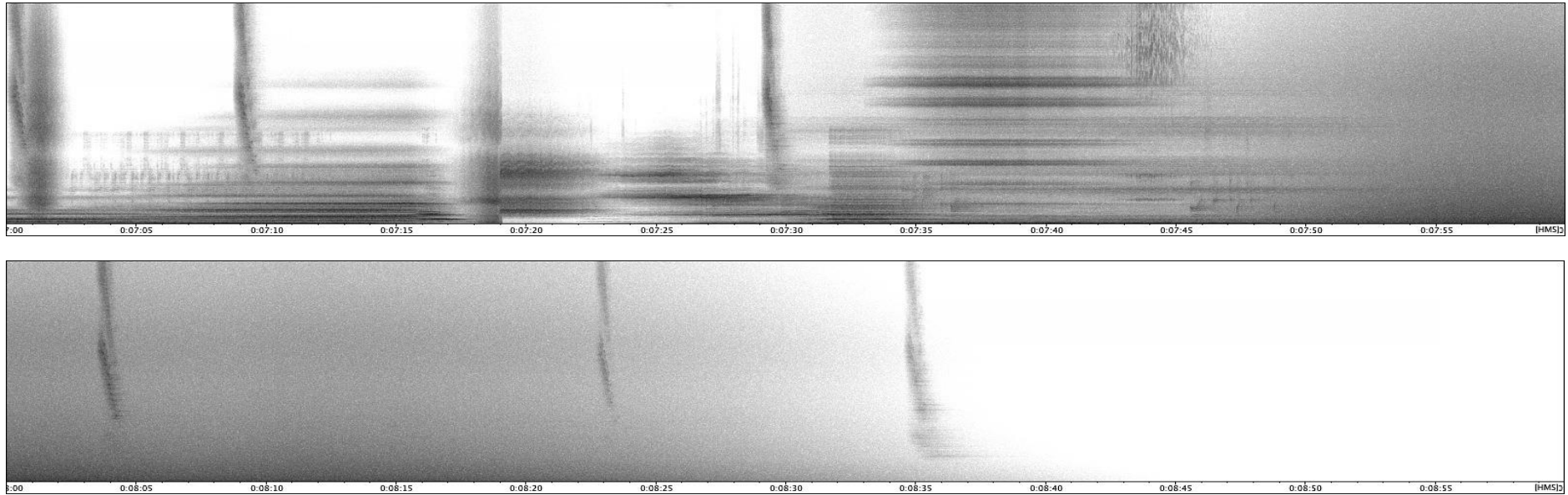
Figure 19: Once Below a Time (2014)

Chapter 5. 3D and Metadata

crunch! and *Penumbra (2015)*

This section of the PhD project saw two major developments. The first was the introduction of periphonic audio and the second, in part predicated by the first, was the addition of metadata storage.

5.1 Extending the Parameter Space

5.1.1 Vector Based Amplitude Panning

In Audio Spray Gun 0.8[40], the parameter space was extended from four to six dimensions using Scott Wilson's implementation of Vector Based Amplitude Panning[41] (or VBAP) for SuperCollider[42].
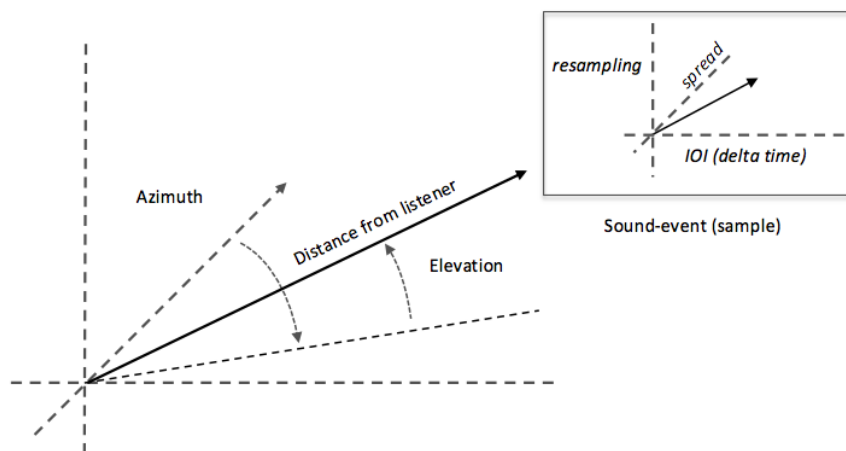
Figure 20: A three-dimensional sound-event

The addition of VBAP introduces vertical spatial components (expressed as either elevation or altitude) to both the extent and position of the locus in the space. This allows composition with loci

---

[40]Richard Garrett, 'Audio Spray Gun 0.8 – the Generation of Large Sound-Groups and Their Use in Three-Dimensional Spatialisation', *Proceedings of the 41st International Computer Music Conference,* Denton 2015, pp. 352-355.
[41] Ville Pulkki, 'Virtual sound source positioning using vector based amplitude panning.' *Journal of the Audio Engineering Society,* 45:6, (1997) 456-466
[42] BEASTMulch SuperCollider UGens <http://www.birmingham.ac.uk/facilities/ea-studios/research/mulch.aspx> last accessed 22 May 2016.

which, when rendered to audio for a dome-shaped array of loudspeakers, can be perceived to move and transform in three dimensions.

5.1.2 Image Spread

VBAP also includes a *spread*[43] function that, as the name suggests, spreads a mono image across a number of speakers around the dome. Experience has shown that trajectories which feature increasing spread as the locus approaches the central listening position or the lowest point in the resampling curve can increase the sense of a solid mass moving through the sound space.

The utility of random spread is still open to question but experiments suggest that mixing a few more widely spread events among near point sources can give the impression of spatially continuous textures without total loss of fine detail. For example, consider this sound-group from 04:40 in *crunch!* which has been isolated from the piece in the audio example. A very low sound-event is deployed at the centre of the listening space with 100% spread (that is, with equal gain in all speakers), followed by a sequence of forty-nine further events which expand around the surround space horizontally and vertically, following a rough spiral. As the sequence progresses, the highest available resample rate is increased and spread is decreased so that events become more spectrally and spatially distinct. This resulting sound is like a global implosion followed by other sounds bouncing off distant hard surfaces.

Sound Example 13: crunch! - woomph (24s)

As another example, in this figure from a development piece, *Meanwhile, in another Part of the Forest* (2016), a sequence of 25 events moves exponentially through the spread axis from zero to

---

[43] Ville Pulkki, 'Uniform spreading of amplitude panned virtual sources.' *Proceedings of the 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics,* New York 1999, <http://lib.tkk.fi/Diss/2001/isbn9512255324/article3.pdf>, last accessed 9 May 2016.

50% to create an effect whereby early events sound separate but the spread of the later events fills the multichannel space. [44]

Sound Example 14: Meanwhile (6s)

5.1.3 Metadata

Until this point, Audio Spray Gun stored data in two formats, either as the parameter set required to generate a sound-group or as a multichannel audio file for direct insertion into a piece. This approach was adequate for two-dimensional surround composition because the *de facto* standard (in academia at least) of eight-channel playback offers a high degree of portability between venues. However, such is the diversity of reproduction systems for periphony that composing fixed-media works for a specific arrangement of loudspeakers commits the composer to a very limited number of venues.

To work round this limitation, this version of Audio Spray Gun introduced a function whereby any given instance of a sound-group could be saved as a list of sound-events in parameter space. In this *object-based* format, each event is stored as a *metadata* string describing the sound source, its resampling rate, spatial position and so on. This introduces a number of possibilities for post-production, including:

- **Spatial Re-orchestration**. The easy re-rendering of works from metadata for concert performance using different reproduction systems;

- **Surgical Editing**. The ability to edit out inconvenient events in metadata;

- **Logical Editing**. Using 'search and replace' or other logical functions to modify sound-events at either the parameter or the audio level;

- **Duplication of sound-groups** with other source files.

---

[44] A spectrogram of *Meanwhile, in another Part of the Forest* can be found in Appendix F of this document and full 16-channel and stereo versions of the piece are included on the USB drive.

These possibilities are discussed further in the conclusions (page 97).

5.1.4 A change of DAW

The move to 3D also necessitated a change of digital audio workstation. Up until this point, I had been working in Ableton Live, dividing the output of ASG into four stereo tracks and editing each one separately. For these and subsequent pieces, I decide to switch to REAPER[45] which has the capacity to arrange interleaved .wav files of up to 32 audio channels as single tracks.

5.2 Works

5.2.1 crunch! (2015)

Programme Note

Old Batman FX, breakfast cereal, an exercise, something you do to numbers, a crisis, something it comes to, the final singularity or maybe a new beginning.

crunch! is one of a series of works created using software of the composer's own design called Audio Spray Gun. This program simultaneously generates and spatialises large groups of sound events from individual samples, extending naturally occurring sounds through space, time and frequency to form expanding clouds and swirling discs of sound that swoop across the sound space.

Discussion

This piece marks a transition from two- to three-dimensional performance. Early versions of the work were composed in octophonic surround as part of the continuing investigation of the Zones algorithm with its newly introduced memory functions. When it was accepted, along with a paper dealing with the three-dimensional possibilities of ASG,[46] for a lecture-recital at the International

---

[45] REAPER, Cockos Incorporated <http://www.reaper.fm>, last accessed 13 March 2017
[46]Richard Garrett, 'Audio Spray Gun 0.8' ICMC 2015 proceedings, pp. 352-355.

Computer Music Conference (ICMC) in Denton, Texas, I decided to re-render elements for the

sixteen-channel periphonic audio available there. To do this, it was necessary to strike a balance,

utilising the new technical developments without losing sight of the original composition.

One strategy for doing this was simply to duplicate existing eight-channel components to make

sixteen-channel versions while adjusting the balance between the two tiers of speakers. When

different vertical balances are applied to a number of sound-groups, a stratified effect can be

obtained. For example, in the section starting at 00:54, multiple scintillating sounds are introduced

above the audience followed by a loud interjection which fills the space (but with a bias towards the

lower tier) and then a further scintillation at ear level only.

Sound Example 15: crunch! strata (01:11 – 01:41, 30s)

Other elements were rebuilt by loading parameter files from the two-dimensional version of Zones

into the three-dimensional version and adding vertical and spread trajectories to them. Sound

Example 16 (00:40 – 00:45) demonstrates such a feature.

Sound Example 16: crunch! - zones 3D (5s)

Performances

Date	Event	Location	Comment
30/09/15	ICMC 2015	Voertman Hall, University of North Texas (UNT), Denton USA	Lecture – Recital (8 channels)
30/09/15	ICMC 2015	Merrill Ellis Intermedia Theater, UNT	Premiere (16 channels)
20/11/15	Electroacoustic Wales	Bangor University UK	

Format

16.0 surround

Duration

5 minutes 21 seconds

crunch!   Richard Garrett
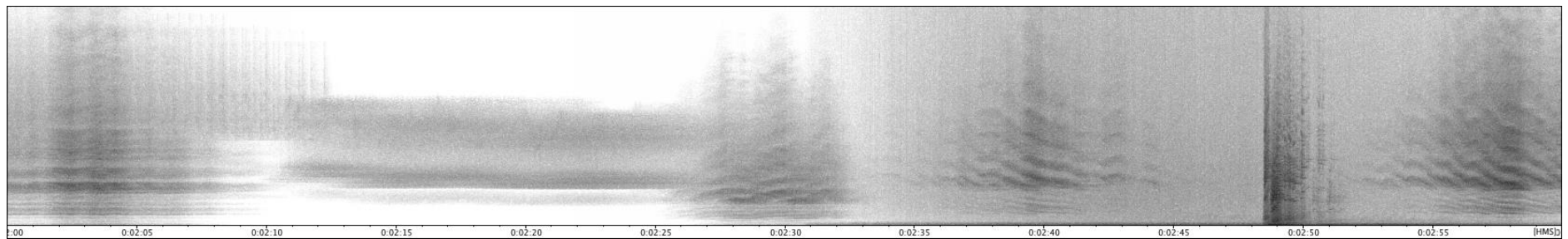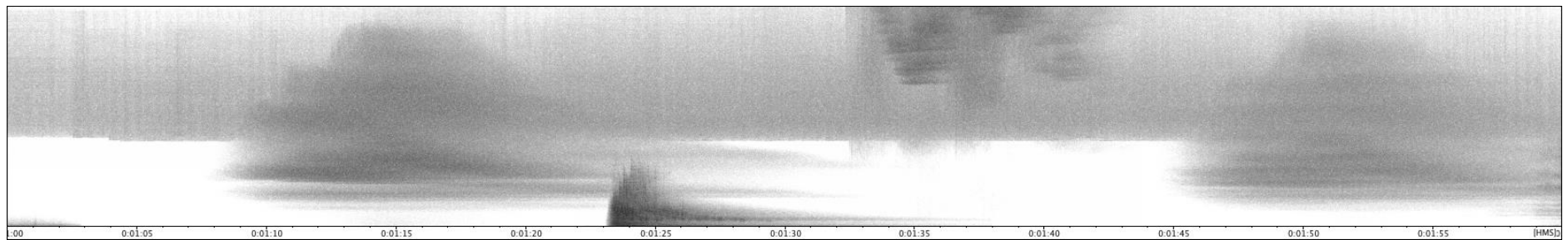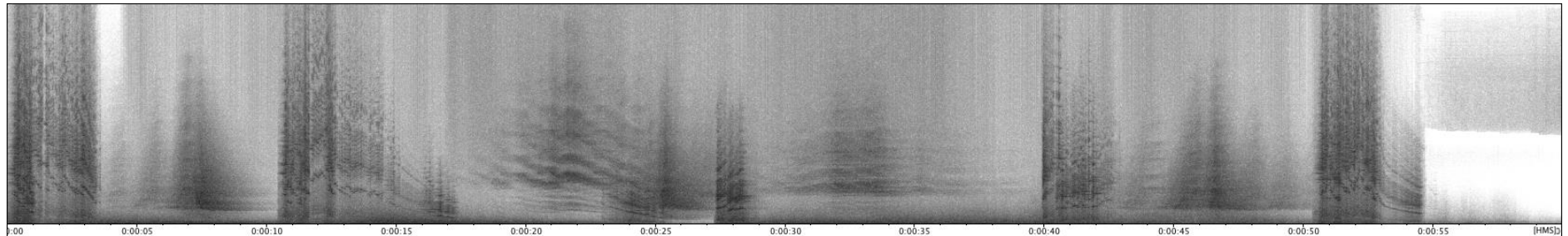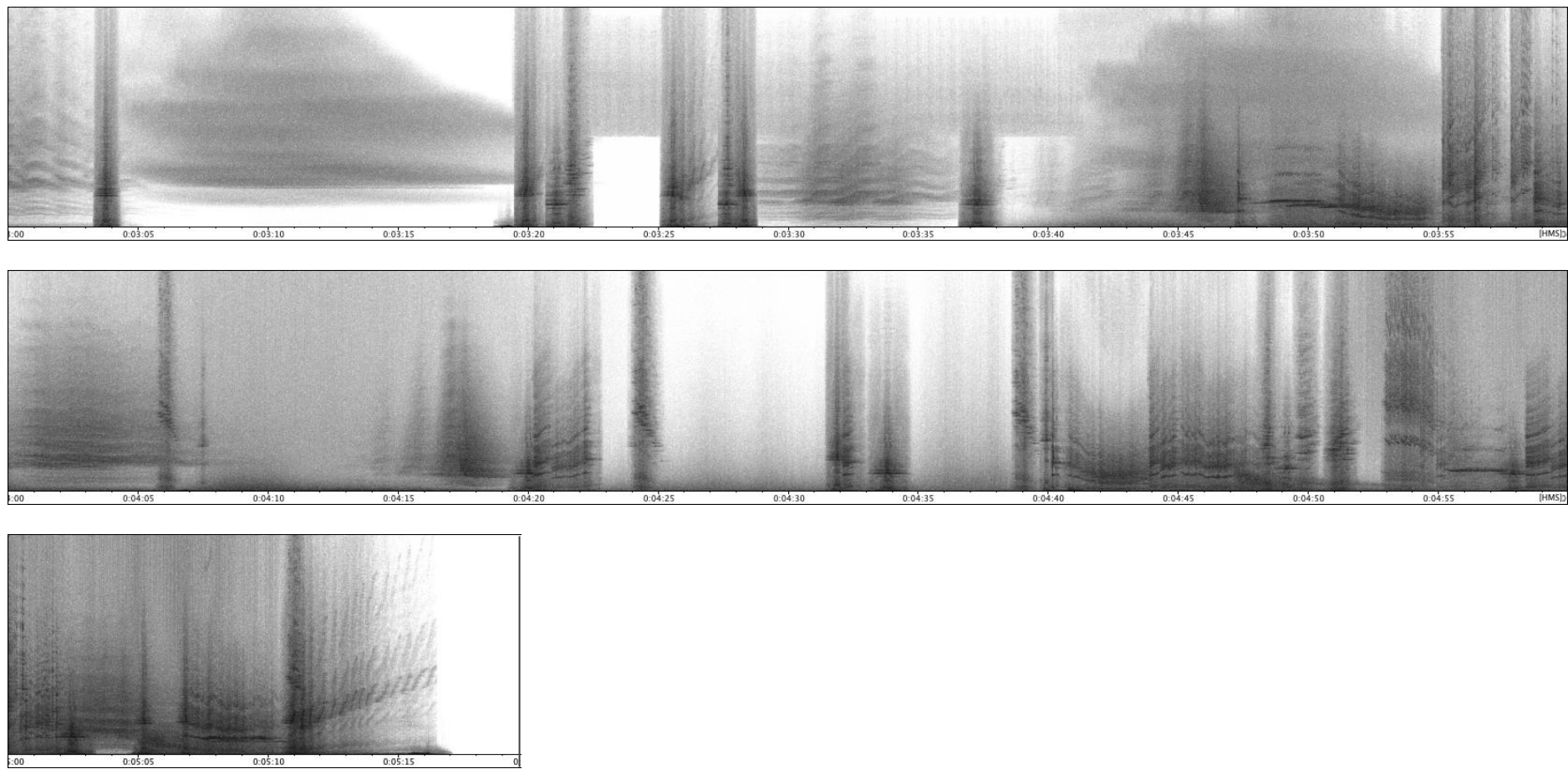
Figure 21: crunch! (2015)

5.2.2 Penumbra (2015)

Programme Note

On March 20th 2015, there was a near total eclipse of the Sun, visible from my home in North Wales. The sky dimmed and birds and animals in the surrounding fields fell silent. It was a very eerie experience.

I have seen three such eclipses so far in my life and, unless I move to another continent, it is unlikely that I shall see more than one other. The Earth and Moon, however, will continue in their orbits and the eclipse cycle will go on for some time to come.

This piece was inspired by the eclipse and by the intimations of mortality and eternity that accompanied it.

Discussion

Penumbra deals with the juxtaposition of pitched and noisy sound-groups. It has four discernable sections:

The first section (00:00 – 03:37) begins with a simple 27-second figure consisting of a group of sustained chime sounds suspended high above the audience followed by a noisy bass-rich 'thud' at ground level. Four extended versions of this figure follow, each starting with a similar chime-group but with each thud followed by other, often noise-based groups.

In the second section (03:37 – 06:01), the idea developed in section one is reversed such that noisy groups now precede more pitched material. Each figure in this section consists of a shrieking metallic group, followed by more noise-based material and ending with a sustained chime group. The figures

develop by increasing the amount of material between these two sounds. The ten episodes in this section are all shorter than those in section one.

In section three (06:01 – 08:22), the pitched material moves into the background forming a bed for sparse high-energy noise interjections. This leads into:

The final section (08:22 – 10:50), which is structurally similar to the second, with figures commencing with noisy groups and ending with sustained chimes but here elements similar to those used in all previous sections are employed. For example, the second figure at 08:37 is a restatement of the one at 05:16 but with an additional low-frequency element. After 'the stall' (see Sound Example 19), episodes get shorter culminating in a sequence of loud interjections (with a repetition of opening sound-group in the background) followed by a pause and big 'thud' at the end.

Penumbra is, in part, an investigation in the homogeneity of sound-groups. When a group of events is constructed using similar mutations of the source sound arranged quite closely together in time and evenly distributed in space, an impression can be given of a single, albeit complex, sound. However, if inter-onset intervals are increased and a less even spatial distribution is used, it becomes apparent that the group is composed of a number of distinct events each issuing from only one direction. Unless these events are reinforced from other directions (typically by adding reverberation) the effect can sound rather unnatural. Many of the sound-groups in this piece explore an intriguing middle ground between these two states producing sounds that exhibit a clear sequence of spatially distinct events without completely destroying any sense of immersion. Sound Example 17 presents three of these 'fractured' groups from 04:38 – 4:44 in the piece.

Sound Example 17: Penumbra - fractures (6s)

Elsewhere, the piece experiments with apparent movement in three-dimensional space. To give two examples: at 10:20, clouds of sound spiral and disappear into the air above the listener as they decay and between 09:00 and 09:21 (a feature referred to here as 'the stall'), the sound climbs toward the ceiling and then, just as one expects it to disappear into space, falls rapidly back to ear-level.

Sound Example 18: Penumbra - spiral (13s)

Sound Example 19: Penumbra - The Stall (21s)

Performances

All performances of Penumbra to date have been given using the sixteen-loudspeaker arrangement included here except for that at the Cube at Virginia Tech where the sixteen-channel file was rendered to third order Ambisonics to be played over all the 148 loudspeakers available there.

Date	Event	Location	Comment
6/1/2016	Electroacoustic Wales/ RMA Postgraduate Conference.	Bangor University, UK	Premiere
13/6/2016	New York City Electroacoustic Music Festival	Abrons Art Center, New York City NY, USA	
7/8/2016	CUBE Fest	Moss Arts Center, Virginia Tech, Blacksburg VA, USA	148 channels
25/11/2016	Sonorities	SARC, Queens University Belfast UK	

Format

16.0 surround

Duration

10 minutes 40 seconds.
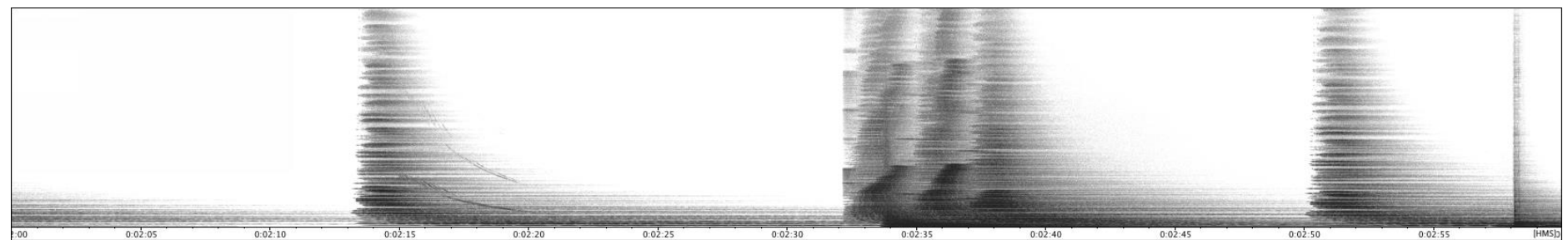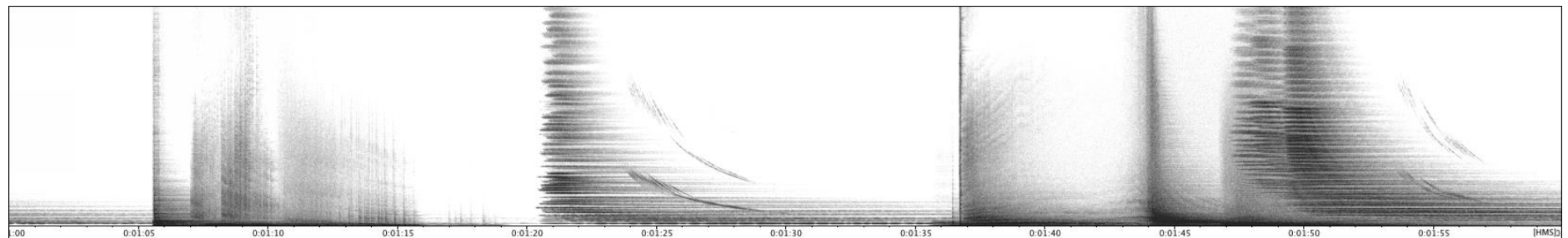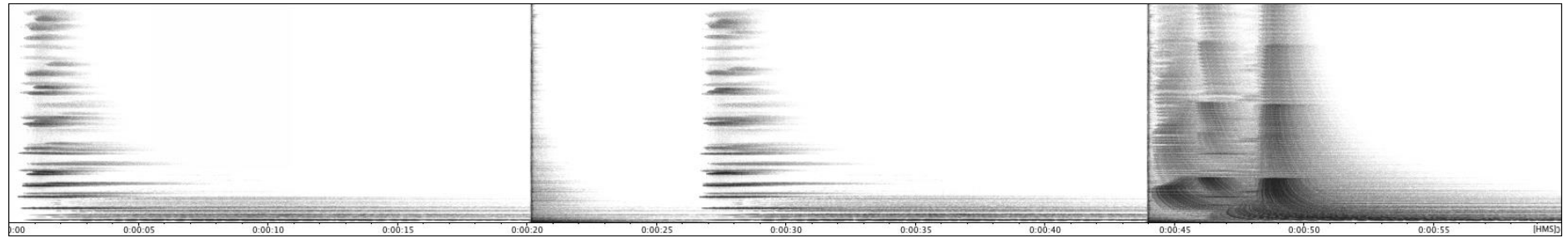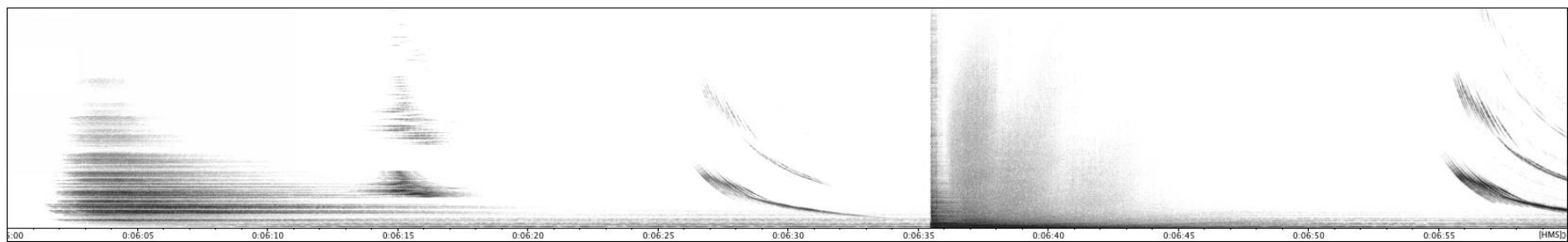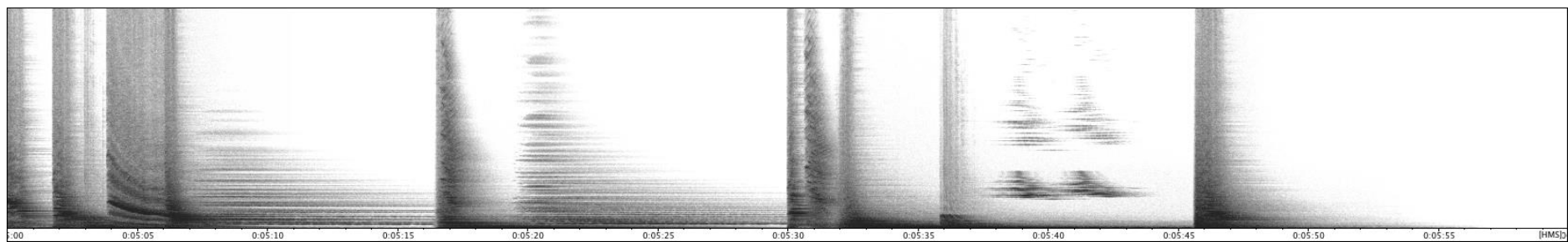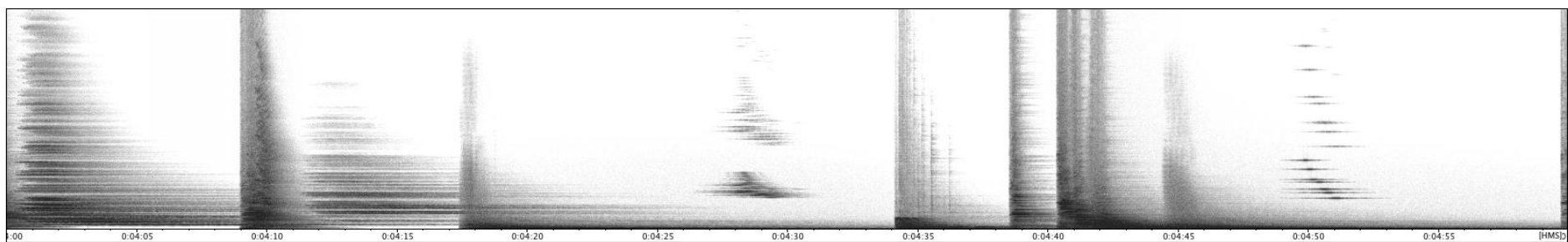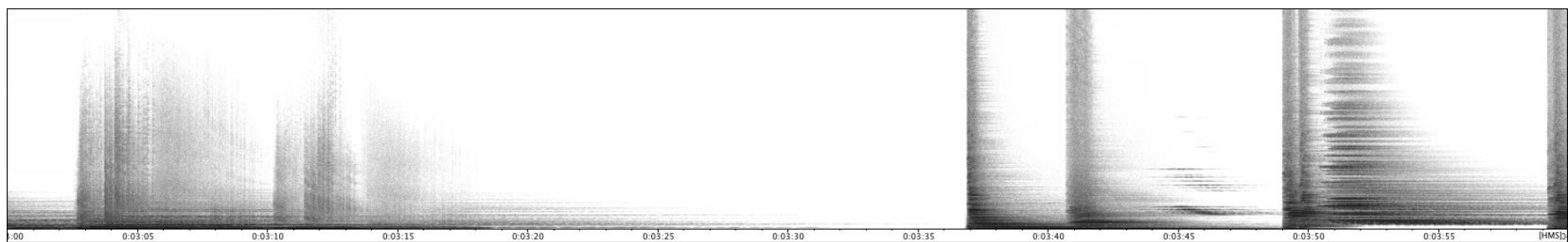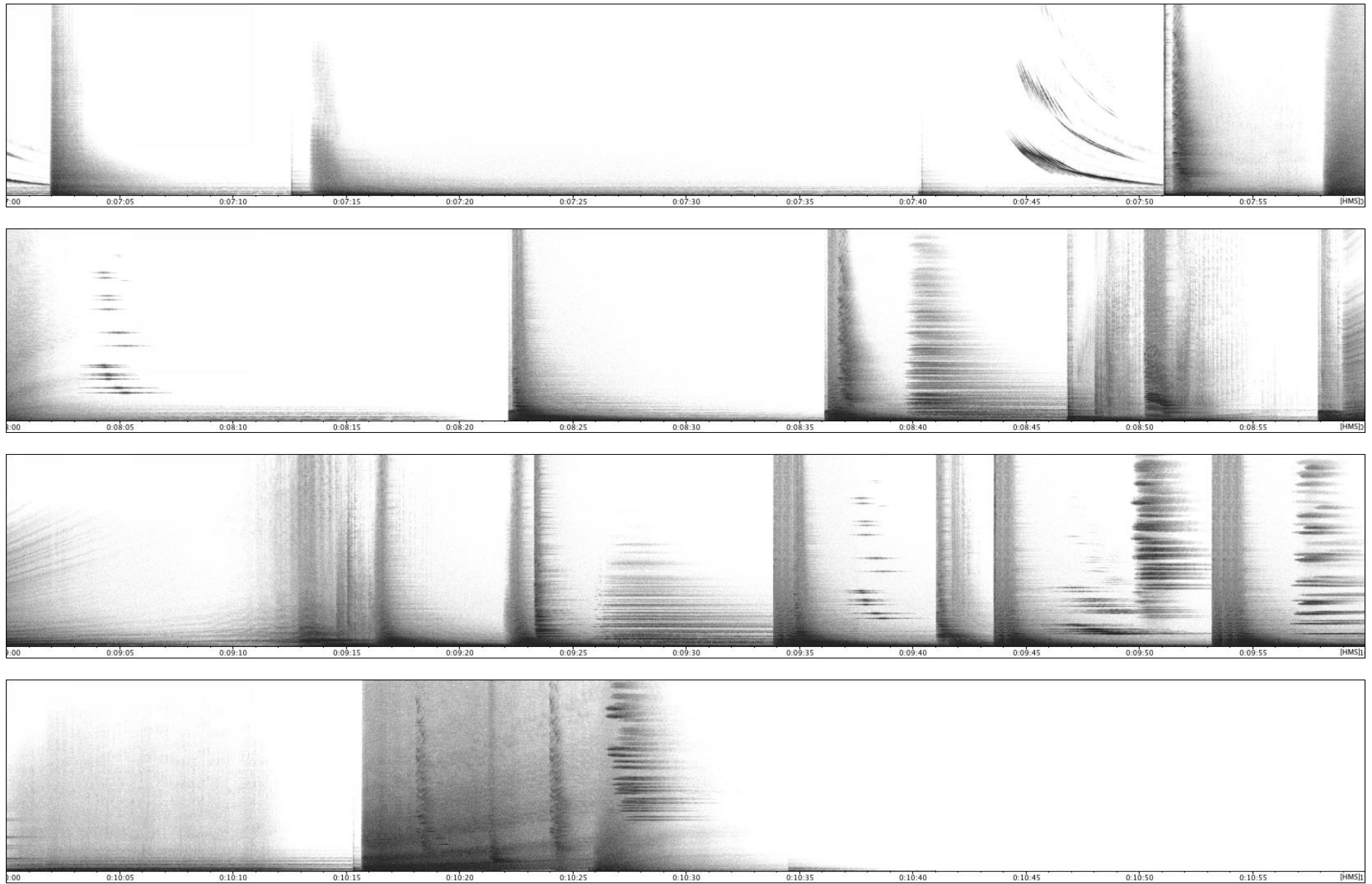
Figure 22: Penumbra (2015)

71

Chapter 6. Agents and Feature extraction, part 2: datasets

This chapter documents the first attempt to unify the sound-group and feature extraction streams of the project. The aims at this stage were first to develop software that could build a descriptor corpus from any collection of audio files and then to apply rejection zone methods to it that would automatically select spectrally contrasting sources from which to generate sound-groups and spatially distribute those groups around the sound field.

Once these techniques for analysis, selection and distribution of sounds were established, it would then become possible to:

a) extract sound events from a live audio signal in order to create complex multichannel accompaniment for an instrumental performer in real-time (chapter 7) and

b) extend the capacities of Audio Spray Gun to include trajectories through timbre-space using a database of previously recorded sound-objects (chapter 8).

Two small programs were written to achieve these aims: *Harvest*, to analyse the files and *Harvester* to create improvisations from the resulting data using the Apple iPhone as a controller. This allowed the use of the analysed data before delving into the programming intricacies involved in steps a) and b).

6.1 Harvest

The function of the Harvest program is to assess a hierarchical array of audio files on a computer hard disk and store the resulting audio descriptors as a file that can be interrogated by other programs.

At the start of the process, Harvest is given a folder location on an Apple Mac hard drive. When the program runs, it systematically catalogues every sound file (.wav or .aif) below an arbitrary maximum length (15 seconds) in the target folder and all the folders beneath it in the disk hierarchy.

Amplitude	Tracks the peak amplitude of a signal.
SpecFlatness	Uses an FFT chain to calculate the geometric mean of a power spectrum divided by its arithmetic mean to give an indication of signal "noisiness".
SpecCentroid	Uses an FFT chain to measure the spectral centroid, that is, the weighted mean frequency, or the "centre of mass" of the spectrum. This can be used to indicate of the perceptual brightness of a signal.
Loudness	Perceptual loudness measured in sones.
Pitch	Outputs two values: The first estimates the fundamental frequency of a signal, and the second, its *clarity*, which indicates to what extent the signal can be said to have pitch.

Table 2: Harvest SuperCollider UGens

When a sound file is processed, it is divided into 20ms segments, each of which is analysed by the SuperCollider UGens[47] shown in Table 2. Once all the segments have been analysed, the resultant data are used to calculate various audio descriptors for the entire file, the most useful of which are shown in Table 3. A list of these descriptor values, along with the name of the file, is then added to a data array. At the end of the process this array (or *corpus*) is written to a SuperCollider archive file for interrogation by other programs.

---

[47] SuperCollider help files < http://doc.sccode.org> last accessed 16 May 2016

Duration	Time elapsed between amplitude onset and reset detection.
Amplitude	Maximum
Temporal Centroid	Amplitude 'centre of mass'
Pitch	Loudness weighted arithmetic mean[48] of **Pitch** frequencies
Clarity	Loudness weighted arithmetic mean of **Pitch** clarity values
Spectral Centroid	Loudness weighted arithmetic mean of **SpecCentroid** values
Spectral Flatness	Loudness weighted arithmetic mean of **SpecFlatness** values

Table 3: Harvest Audio Descriptors

6.1.1 Harvester and the iPhone

The companion program to Harvest, called *Harvester*, employs twelve identical software agents to build and deploy sound-groups around an eight-channel space in real time, controlled by an improviser using an iPhone.

To do this, the program first takes a file built by Harvest and uses it to construct arrays based on the descriptor values assigned to each sound. It also builds similar arrays describing the ranges of spatial position, resampling rate and delta time available to the agents during the piece. When play begins, one of the agents selects a point at random from each descriptor array and chooses the sound file that most nearly fits the descriptor values as the source for a sound-group of between one and twenty events. Spatial co-ordinates, resampling rate and so on are also chosen at random at this time. As soon as the sound-group plays, rejection zones (page 110) are created around the chosen points in all the arrays. These zones, which stay in place for the duration of the group, reduce the likelihood of similar choices will be made by subsequent agents to those made by the first. This process repeats for all active agents throughout the duration of an improvised performance. In this piece, composed for an eight-channel ring of loudspeakers, azimuth selection is further biased towards the front and centre of the sound space.

---

[48] Rather than take the simple arithmetic mean of the parameter values over the duration of the sample, each value is multiplied by the Loudness value at the time and then divided by the sum of the Loudness values. See < https://en.wikipedia.org/wiki/Weighted_arithmetic_mean> last accessed 28 August 2016

Consider this sound example of rejection zone behaviour taken from another version of the improvisation. In it, one can hear a succession of sound-groups being distributed around the azimuth axis. Although there is a bias towards the front and centre location, when each group sounds it is placed in a location away from those currently sounding. This effect is clearer on the eight-channel version of the example than on the stereo reduction.

Sound Example 20: Harvest Extract (27s)

Agent Activity Indicators

2D slider: Time between activations
vs. Agent Activity Limit

Stop/ Start (Pause) Button

Time elapsed – display only.
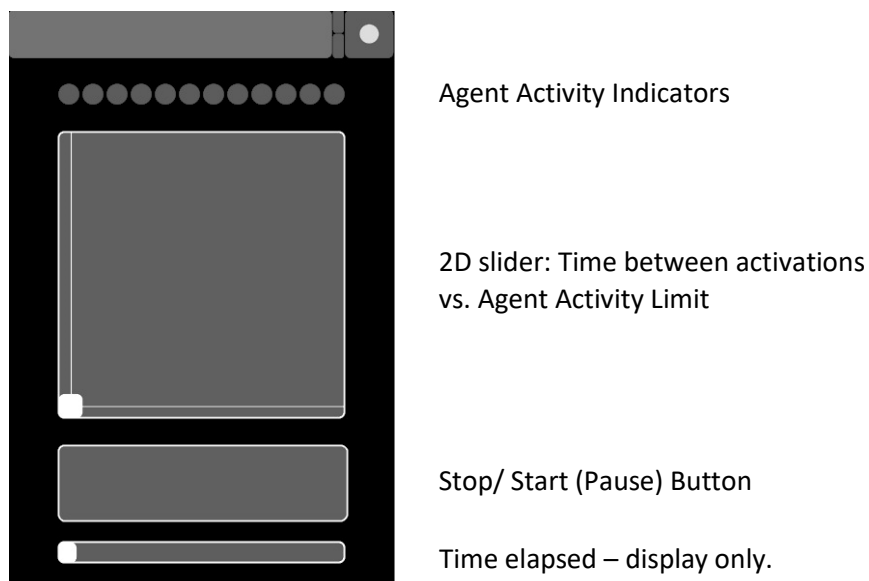
Figure 23: Harvester iPhone Interface

Improvisation with Harvester is performed on an iPhone running TouchOSC.[49] This app allows the user to create a touch-sensitive interface on the phone that sends Open Sound Control (OSC)[50] instructions to SuperCollider via Wi-Fi. The interface for this program features a two-dimensional slider that controls the number of agents playing and the time between the start of sound-groups. A button is provided to pause the loop at will (Figure 23).

---

[49] TouchOSC <http://hexler.net/software/touchosc> last accessed 6 May 2016.
[50] Open Sound Control, <http://opensoundcontrol.org/introduction-osc> last accessed 29 September 2016.

6.2 Work

6.2.1 Harvest Improvisation (2016)

Programme Note

This is a live improvisation made with a computer program called *Harvester* controlled by a performer using an iPhone. The program uses twelve agents to generate groups of sounds and distribute them around a circle of eight loudspeakers. The sounds used to construct these groups are selected from a large collection of audio files that have already been analysed in terms of *audio descriptors*. When an agent selects a sound with which build a group, it uses these descriptors to select one whose timbral qualities are unlike any groups currently playing. From the iPhone, the player can control the density (the number of agents playing) and the time between the start of sound-groups. S/he can also pause the loop at will.

Discussion

This piece is a wholly open improvisation designed to construct a performance around any available sound library. The performer is given a deliberately limited interface controlling only the timing and density of events, while the computer generates and distributes sound-groups. Thus the chief responsibility of the performer is to listen to the sounds, respond to them as they occur, and decide how the sequence will unfold.

An interesting option would be to perform this work with a large but previously unheard selection of files and see what emerged.

In addition to use in performance, Harvester can function a useful for the experimenting composer. It lets one audition numerous juxtapositions of sound sources in a short time and often reveals combinations of materials that might not have been considered otherwise.

Performances

No Performances to date

Format

8.0 surround

Duration

Five to ten minutes

(Recorded example: 7 minutes 3 seconds)

Chapter 7. Agents in Space

Guitar Piece (2016)

7.1 Using a live instrument to generate Sound-Groups

In this section, source sounds are extracted from a live performance, selected using fuzzy rules and then used to create accompanying sound-groups in the style of Audio Spray Gun.

In *Guitar Piece (2016)*, audio from a live instrument is recorded and segmented (as in the *Exploration* patch) and then each segment catalogued using the same descriptors as in *Harvest*. The accompaniment is then generated, in much the same way as it was in *Harvest Improvisation*, by a number of agents (ten in this case), each of which builds rejection zones (see page 110) in both spatial and audio descriptor dimensions around the sound it is playing. When an agent selects a segment, it takes this as the source from which to generate a sound-group. As the piece progresses, further selection biases are applied to different parameter arrays by a system of *scenes* and *actions*.

7.2 Work

7.2.1 Guitar Piece (2016)

Programme Note

This is a semi-improvised work for electric guitar and electronics, where the guitarist follows a score giving loose instructions on what sounds to play and the computer generates an accompaniment derived from the performance so far. Ten software agents choose sounds for the accompaniment based upon what the guitar and any other agents are currently playing. At the same time, the agents distribute the sounds they create around eight loudspeakers surrounding the audience, to build an enveloping sonic environment from a single instrument. As the piece progresses, the guitarist

switches the computer through a succession of scenes, placing limits on the density and character of the accompaniment.

Because of the generative nature of the program employed, a unique, if structurally similar, accompaniment will be created each time this work is performed.

Discussion

The guitar score for this piece has no tempo markings. At each bar, the guitarist should play one or two examples of the sound shown and then, if a circled number is shown, s/he should trigger the next scene and wait for a response from the computer before proceeding to the next bar.

The player should use e-bow 'hot spots' (that is placing the e-bow directly over the poles of the pick-up to produce an overdriven signal) only where allowed in the score.

```
// ============== SCENE  3 ==============

(
        recState: true, density:  4, duck: 0.2,
        actions:
        [
                (weight: 0.15, action: \highScrape),
                (weight: 0.15, action: \lowScrape),
                (weight: 0.2, action: \singleBlip),
                (weight: 0.5, action: \noAction)
        ],
),
```

Figure 24: A Scene in Guitar Piece

In this piece, the guitarist triggers a succession of scenes using a footswitch (or the mouse). Each scene consists of a set of possible actions and a number of *global* values (see the example in Figure 24). Global values apply to the scene as a whole or to all the agents active during it and can include values for chance, density and duck rules, recording status and so on. Each item in the action list consists of a weight value and an action to be performed. Together, these weights form a probability

table from which newly available agent select further actions. The scene in Figure 24 occurs between 00:32 and 00:48 on the recording.

Sound Example 21: Guitar Piece Scene 3 (16s)

Figure 25 shows an action in which the agent is required to create between one and five short (duration), noisy (clarity) events, to be played at a long distance from the listener (rho), with a medium time interval between each (delta).

---

highScrape:

(**events:** {rrand(1,5);} , **duration:** ~bias[\low], **clarity:** ~bias[\lowWide],   **rho:** ~bias[\high], **delta:** ~bias[\mid])

Figure 25: An Action in Guitar Piece

As the work progresses, subsequent scenes impose a different balance of choices on the agents producing a changing accompaniment.

From the perspective of the guitarist, this is not a virtuoso work. Because the onset detection method used by the program is dependent on silences between events, any legato phrase or chord will be stored as a single segment. Such segments produce sound-groups that are too easily source-identified (often producing 'chipmunk' effects when resampled) and bear little repetition. Because of this, the score consists of a number of discrete events that, by virtue of their simplicity, can generate a good variety of sound-groups for the accompaniment. The principal requirement for the player, therefore, is that s/he should consider what they play and listen to results, choosing each sound not only for the way it fits with the accompaniment at the time of playing but for the contributions it will make later on the piece.

This work requires the following equipment: an electric guitar, ideally fitted with a whammy bar or tremolo arm; an e-bow (plus or standard model); a nylon plectrum and a steel or glass slide.

Performances

No Performances to date

Format

Electric guitar and laptop (8.0 surround)

Duration

9 to 15 minutes

(Recorded example: 9 minutes 51 seconds)

Guitar Piece

Richard Garrett

Figure 26: Guitar Piece

Chapter 8. Timbre space

Morphology, Objets Volants (2016)

The final stage of this PhD project uses a variant of Audio Spray Gun called *MorphGun* which, rather than create sound-events from a single source file, uses an analysis file generated by *Harvest* to create trajectories through 'timbre space'. *MorphGun* selects the sound source for each event from a corpus of sounds using the array scoring method described on page 108 either to find a 'best fit' result or to choose from a filtered probability table. This technique exhibits a number of limitations due to the discontinuous nature of the descriptor space derived from finite datasets.

8.1 Multiple Sources and their limitations

Whereas the external parameter dimensions applied to a single source in Audio Spray Gun (distance, azimuth, resampling and so on) are all continuous, audio descriptor dimensions derived from individual samples in a finite corpus are discontinuous, being made up of discrete values with irregular and sometimes significant steps between them. Selection from a dataset containing disparate sources can produce clusters of events where trajectories jump from one set of similar sounds to another different set with nothing in between. The more parameter dimensions the system attempts to traverse at once, the more likely it is to encounter these discontinuities. Conversely, in more cohesive datasets, samples may be so similar that trajectories are hard to distinguish from purely random selection.

This results in a kind of uncertainty principle operating between the regularity of value distribution along a parameter axis and the width of the search rule applied. If you apply a narrow search rule to a regular distribution, its output will progress smoothly, morphing between adjacent values but will return null results when it hits a gap. If the rule is wide, it will cross the gaps more easily but is more

likely to return a large number of similarly weighted candidates when traversing more densely populated sections of the graph. In the extremes, a narrow rule will jump between single values and a wide one will produce randomness.

Furthermore, timbral descriptor dimensions can exhibit perceptual co-dependencies. For example, while the spectral centroid of a sound containing a number of clear harmonics can be said to describe its 'brightness', the same parameter can also represent the median frequency of normally distributed noise. Therefore, two adjacent spectral centroid values may refer to two radically different sounds and, unless some constraint is applied in a noise dimension, simple trajectories along the spectral centroid axis can be quite unsmooth.

8.2 Works

8.2.1 Morphology (2016)

Programme Note

1. The study of the forms of things.

2. A particular form, shape, or structure.

Oxford English Dictionary

Morphology is a fixed media work for three-dimensional surround sound. The figures that make up the piece were created using a computer program that builds sequences of events using sound sources that change over time according to their timbral features. As these 'timbre space' trajectories unfold, they are accompanied by other trajectories in spatial and frequency domains. As the work progresses, the sound-groups used move from those developed from small collections of similar samples to ones built from larger collections of more diverse sources. With this growing variety of source content, homogeneous clouds of sound are steadily replaced by more

heterogeneous structures whose components display individual timbral identities and spatial locations. Morphology studies these forms, investigating their fragility and their capacity to disrupt an immersive sound space.

Discussion

This work, composed almost entirely using material generated by *MorphGun*, and contrasts sound-groups derived from small spectrally consistent datasets with those produced from large disparate ones. Sound-groups at the beginning of the piece are all derived from small datasets with groups from larger sets appearing over time and all but taking over by the end. In a parallel development, the piece moves from strongly articulated sequences at the beginning to multiple overlapping groups at the end.

This piece can be thought of as having seven short sections followed by one long one. The first two sections (00:00 to 00:31 and 00:31 to 00:59) are composed wholly of sound-groups generated from restricted datasets, typically of around a dozen samples. These groups sound quite coherent.

Take for example, the metallic, scraping sounds starting at 00:00 are a sequence of 50 events produced using a set of 11 related recordings of saucepan lids. These events spiral out three times horizontally from the listening centre, their elevation rising linearly through the group duration. Spectral content also rises but more exponentially, with delta times ranging between 10 and 20ms. Timbre selection tracks from high centroid and flatness values (approximately 'bright and noisy ') to lower in middle of group and back again. Because low scores are unfiltered, sample selection shows quite high randomness.

Sound Example 22: Morphology - restricted source group (7s)

85

The third section sees the introduction of material created using the larger corpus of ~180 diverse sources. At 01:06, the first such group, a sequence of eight events starting with a sliding ceramic sound and ending with two strikes of a bell, can be heard. This forms the third element of this section, two other diverse-source elements occurring at 01:19 and 01:23. As the work develops, more diverse-source elements are heard, occurring earlier in the sections and, with a few exceptions (notably between 02:51 to 03:12), steadily replacing the restricted-source groups.

A clear example of a diverse-source group can be heard at 01:46. This feature comprises 300 events chosen from the large corpus. Like the previous example, it forms a complex spiral but in this case, four arms move from out the centre while the feature makes three complete rotations in the horizontal plane. Resampling rate and elevation move from low to high twice during the 300 events with spread following distance from the centre (25% at centre, zero at edge). Random delta times range from 10 to 110 ms. Over the length of this feature, descriptor choices run from lowest to highest measured fundamental frequency with a slight emphasis on higher centroid and flatness values towards peak frequency.

Sound Example 23: Morphology – diverse source group (13s)

As can be heard in Sound Example 23, diverse-source groups form heterogeneous sequences of events that are sometimes reminiscent of conventional acousmatic composition. The difference here is one of method. Rather than painstakingly handcraft a phrase by placing, balancing, and panning a number of individual sound files in a DAW, many possible phrases are created and auditioned in real time before one is selected for the piece.

From 03:12 until the end, *Morphology* is made up almost entirely from diverse-source groups, overlaid on one another.

Morphology revisits the idea of fractured sound-groups discussed on page 67, made more fragile here by the spectral variation inherent in multiple sound sources. For the most part, as in *Penumbra*, the aim is to produce clear sequences of events that are spatially distinct yet maintain an immersive experience for the listener. Occasionally, however, sound-groups are chosen that deliberately expose a unidirectional monophonic event to create an effect reminiscent of 'breaking the fourth wall' in cinema, that reveals the artifice for a second and then jumps back into the sound world as if nothing had happened.

Sometimes such breaches in spatial homogeneity can cause a particular sound-event (perhaps a sustained bell sound issuing from a single loudspeaker) to stand out too far from its surroundings. In *Morphology* this was remedied by locating the offending event in metadata and either changing its image spread or using Audio Spray Gun to create a distribution of similar events around it.

Sadly, these effects can only be truly appreciated in at least eight-channel and preferably periphonic surround sound. The spectral smearing together of spatially independent events brought about by stereo reduction eliminates much important detail from the piece.

Performances

No Performances to date

Format

16.0 surround

Duration

5 minutes 53 seconds

Morphology (2016)   Richard Garrett
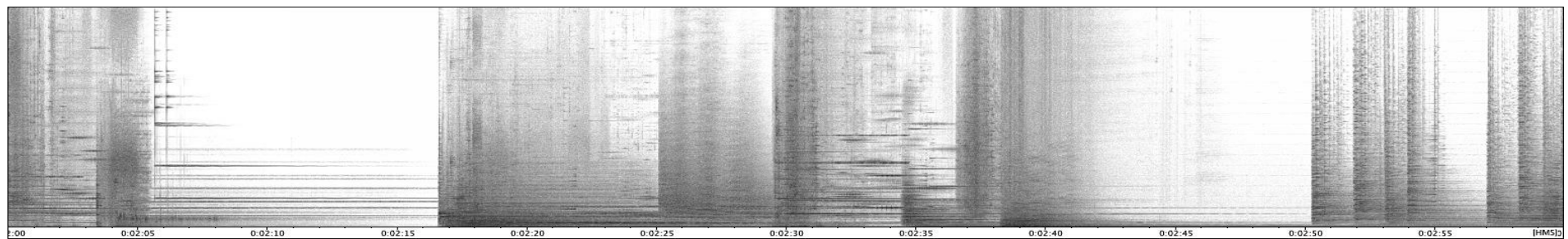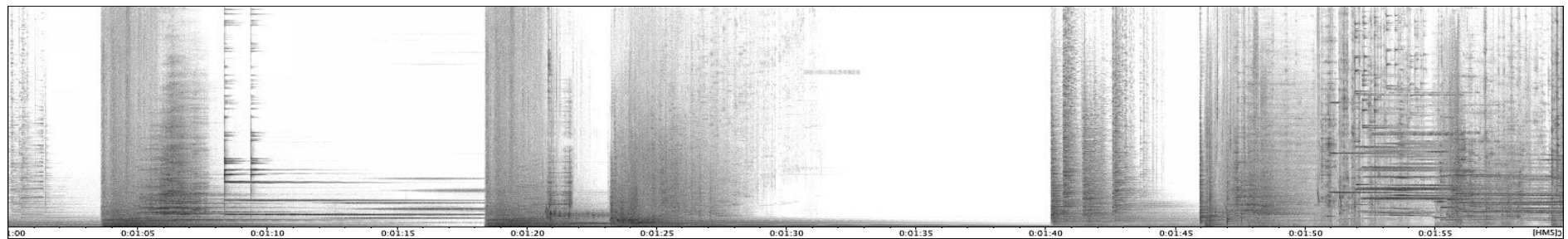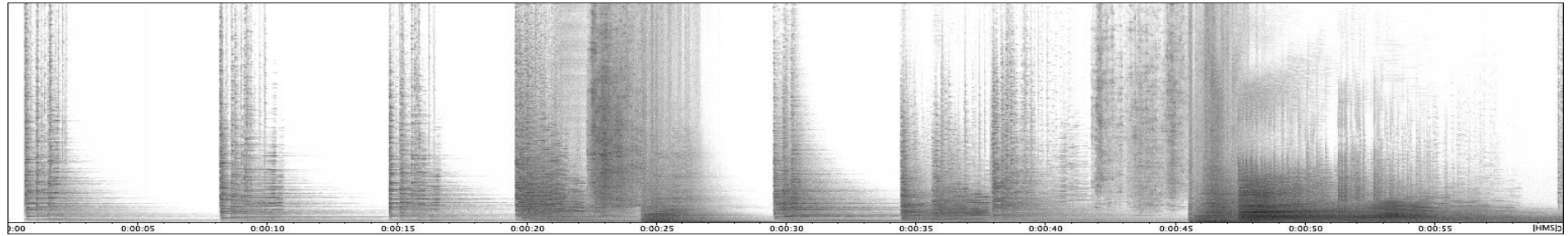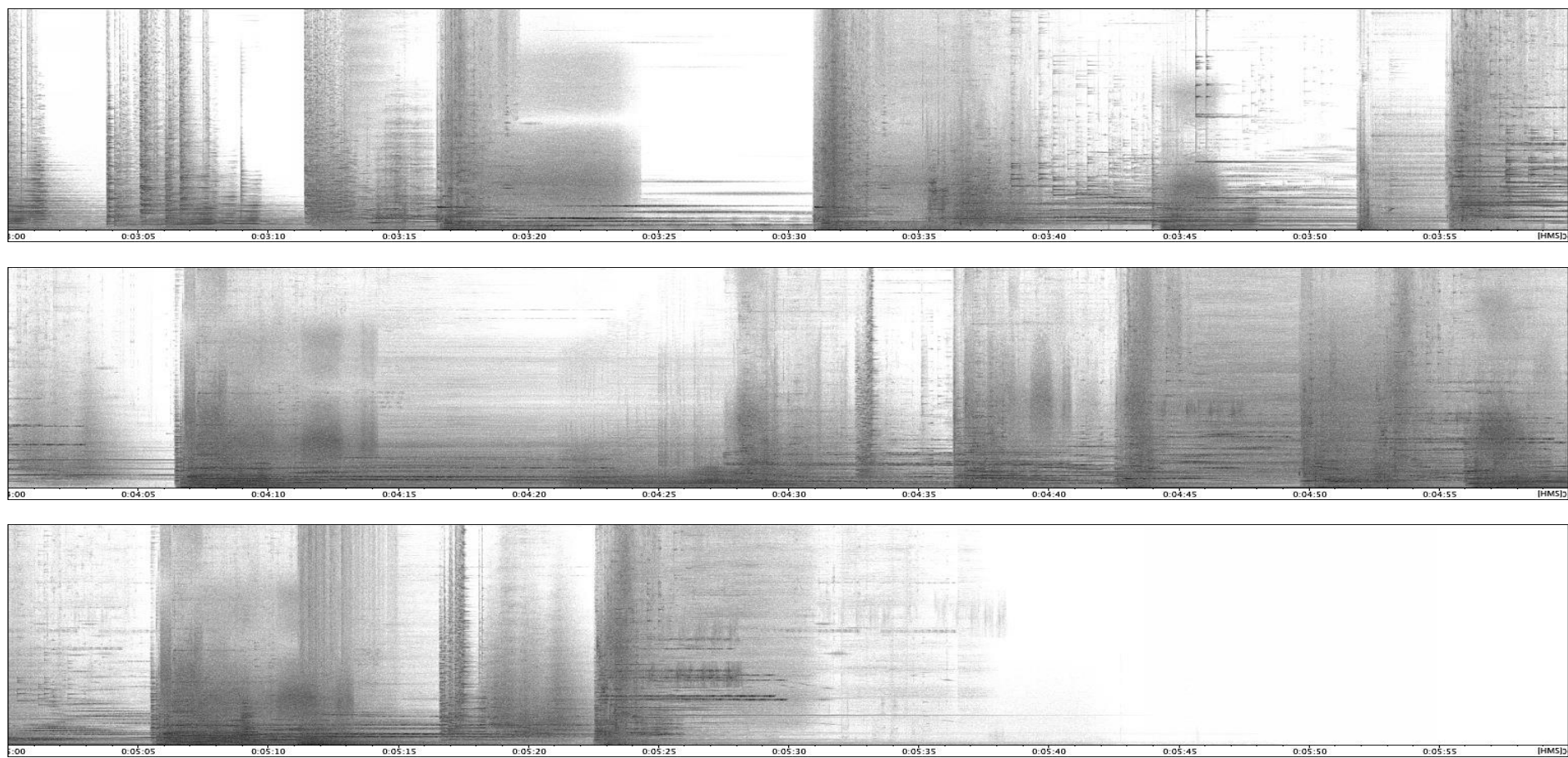
Figure 27: Morphology (2016)

8.2.2 Objets Volants (2016)

Programme Note

The French have a word for it.

This is spatial music constructed of flying (and floating) sound-objects, all unidentified as befits acousmatic music, with a French title to honour the tradition.

Also, OVNI (for *objet volant non identifié*) rolls off the tongue so much better than UFO.

Discussion

Objets Volants focuses on groups generated by less-disparate datasets interrogated by fairly narrow rules and employed in combination with single-source (Audio Spray Gun) forms. The work seeks to contrast stable forms based on single pitched samples with more abrupt groups using multiple noisy sounds. At the same time it aims to spatially orchestrate numerous groups expanding, contracting and moving around the space simultaneously yet each maintaining their independence from one another.

Because of the careful choice of datasets, the MorphGun groups may not always be easily discerned from single-source sounds. Some that do stand out, however, are those figures reminiscent of radio static (for example from 04:25 to 04:36) and quick multi-sample interjections like that at 06:42.

Sound Example 24: Objets Volants - static (11s)

Sound Example 25: Objet Volants - interjection (2s)

Performances

A performance of this work in 148 channels is scheduled for CUBE Fest at Virginia Tech, Blacksburg

USA in August 2017.

Format

16.0 surround

Duration

10 minutes 17 seconds
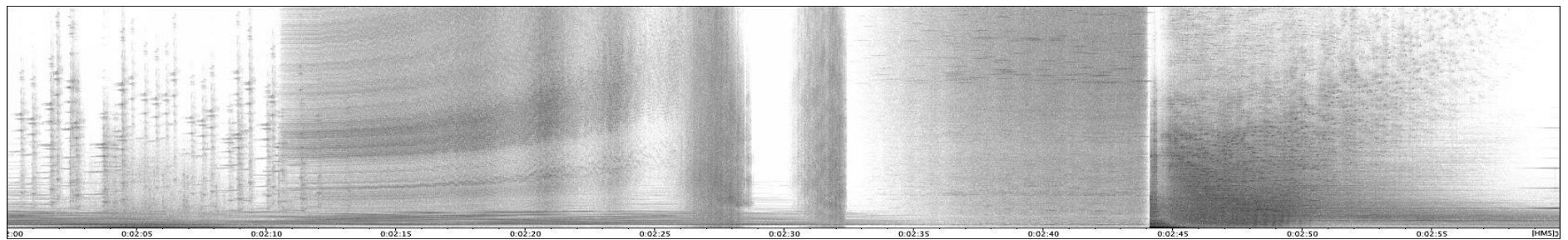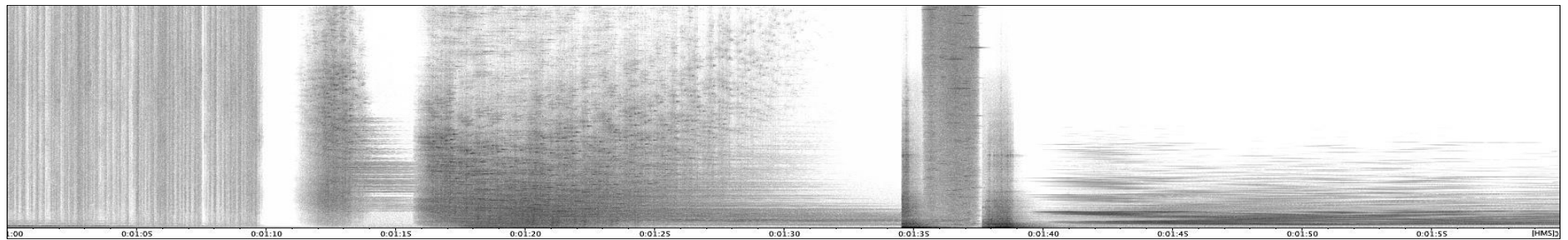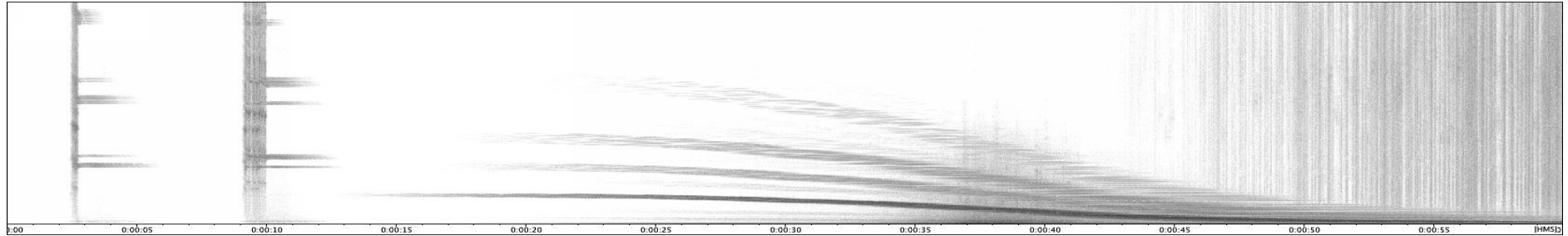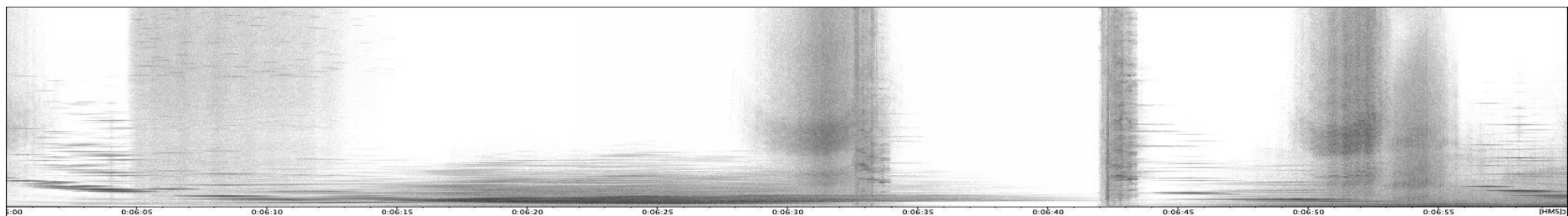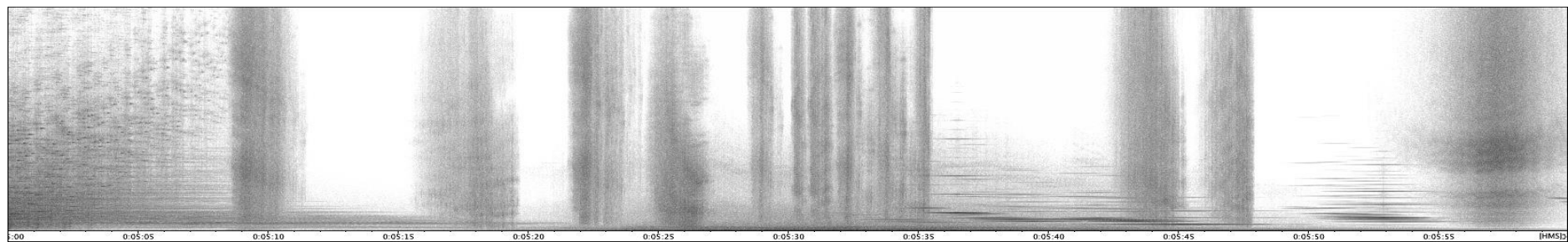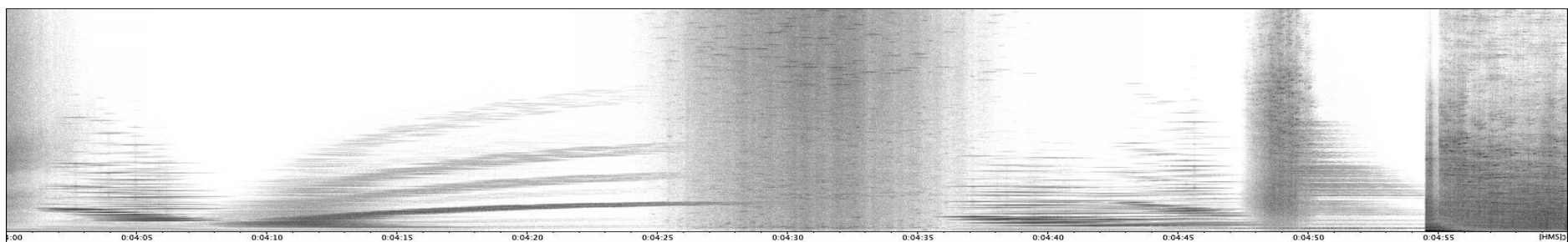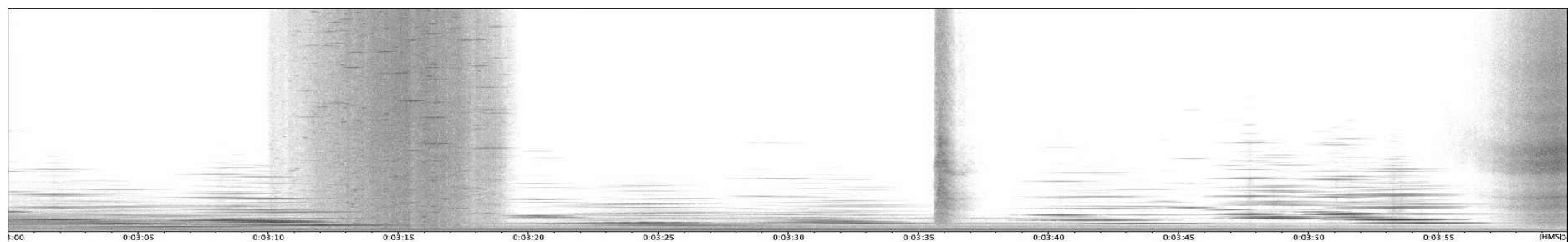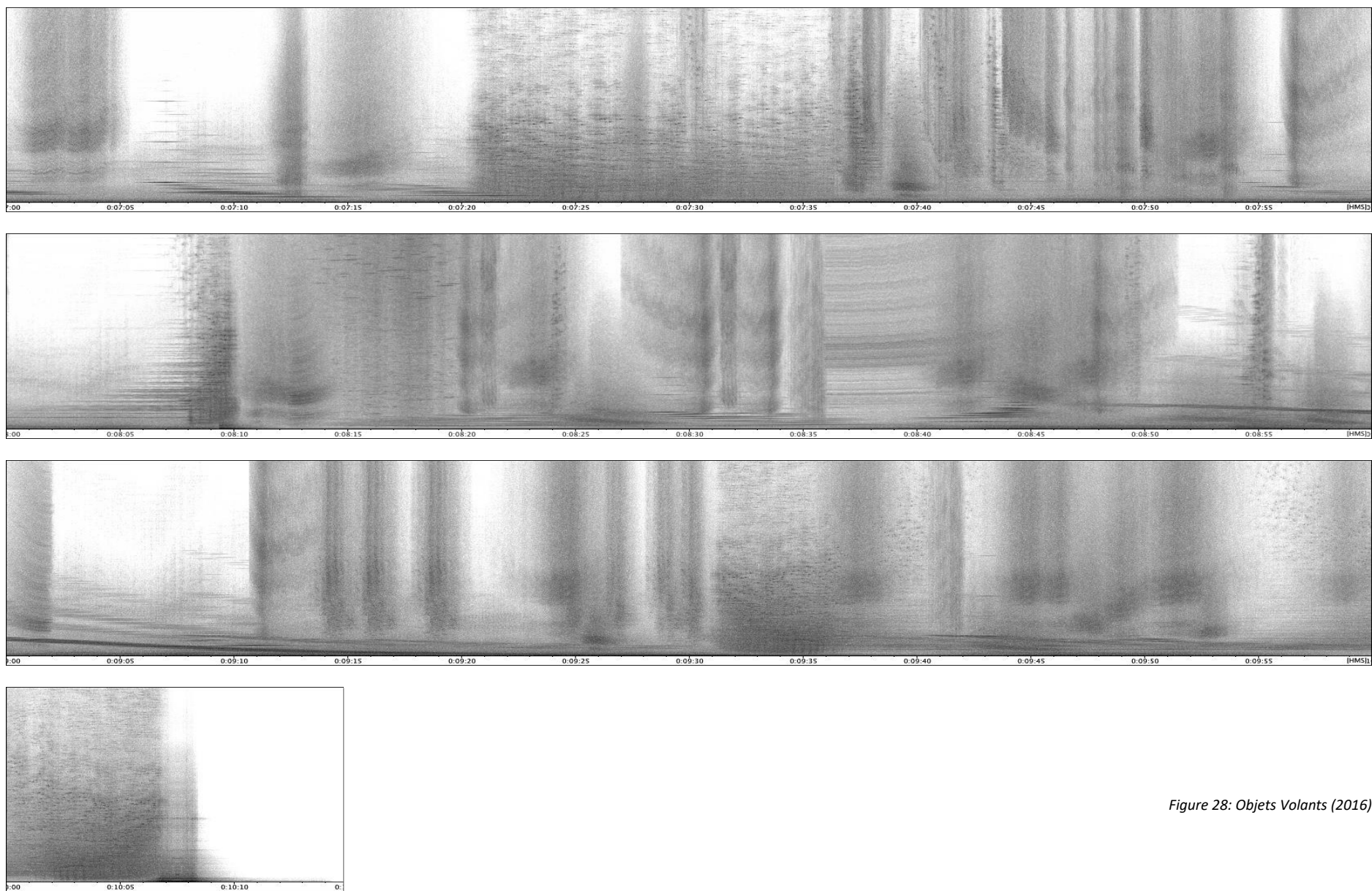
Figure 28: Objets Volants (2016)

Chapter 9. Conclusions

9.1 Reflections on the work

"From a compositional point of view, music is an *n*-dimensional design space."[51]

While the initial focus of this project was on the use of fuzzy logic for sound-group generation and sound source selection, its development has thrown up a number of other issues. These surround the use of the locus as a compositional tool, particularly for object-based spatial music as applied to High Density Loudspeaker Arrays. Here are some thoughts.

9.1.1 Locus Aesthetics

The locus metaphor used throughout this study offers some interesting insights. When discussing these, many parallels can be drawn with granular synthesis and it is useful to adopt some of the language used in that field.

Curtis Roads, when discussing granular synthesis, talks about "new mesostructures", and a range of textures that "include sound masses, dense clusters, flowing streams and billowing clouds... characterized by qualities like their density, opacity and transparency..."[52]. Although Audio Spray Gun operates at sound-object rather than granular durations, such forms are still observable.

If an idealised version of the locus used in Audio Spray Gun were contracted to a zero extent in all available dimensions, a *singularity* would occur. That is to say, a source would be produced consisting of an infinite number of versions of the same sound, playing at the same resampling rate,

---

[51] Curtis Roads, *Composing Electronic Music* p. 289
[52] *ibid*, pp. 308 - 313

at the same location in space and time. Infinitely loud, it would form the ultimate, totally fused

sound mass. Fortunately, this is unobtainable in the real world but we can imagine that an achievable

minimum locus would approximate to a monophonic 'point source' for which:

- there is a single sound source;

- the mean time between process events would be small but non-zero;

- the source characteristics are homogenous (e.g. a sin wave or narrow band noise); and

- the source maintains a constant duration, independent of frequency shifts.

When this locus expands in any direction, its volume increases and the density of events falls with a

variety of results (Table 4). In each case, this process of "evaporation"[53] causes individual events to

become more distinct, transforming the fused sound mass into a finer cloud.

Parameter	Contracted	Expanded
ρ	All events same amplitude	Amplitude varies, near sound may mask distant ones.
θ	All events same azimuth	Events come from different horizontal directions. May cause 'undulating cloud' around listener.
ϕ	All events same elevation	Events come from different vertical directions. May cause 'undulating cloud' above/below listener.
f	All events same resampling rate	Events become more discernable as spectral range increases
Δt	All onsets very close together (sound mass)	Mass => Cloud => overlapping events => discrete events (sequence)

Table 4: Contraction and expansion in different parameter dimensions.

Consider, for example, this evaporation as it occurs in delta time. When the mean time between

events (Δt_{mean}) is small, the sounds form a mass. Next, as Δt_{mean} approaches the duration of the

source sound, clouds become perceivable as groups of overlapping individual events. Then, as

expansion passes this point, discrete events become separated by silence. Eventually as delta time

approaches infinity, silence ensues.

---

[53] *ibid*, p. 312

Audio Spray Gun departs from this idealised model in two ways. Firstly, the sound sources employed can vary significantly in timbre over their duration allowing one to discern different instances of the same sound at different points in their evolution, even when playing at similar resample rates. Secondly, event duration changes in inverse proportion to resampling rate (f), such that clouds become discretised at around Δt_{mean} = source duration / f_{mean}. This evaporation becomes even more acute when multiple sound sources are employed as the result of locus transformations in descriptor space (see discussion on page 83).

However, the locus remains a useful analogy for thinking about the spectral and spatial aspects of a mesostructure as a unified whole. Rather than design the spectral characteristics of a sound first and then later decide where to place it in the sound space, one can experiment with all the spatial, spectral and temporal characteristics of a sound-group simultaneously. In this way, the organisation of sounds in space can be placed on an equal footing with their organisation in time, timbre and any other available dimensions, perhaps offering opportunities to create truly *spatial*, as opposed to merely *multichannel* music.

9.1.2 Metadata and Spatial Music

Audio Spray Gun version 0.8 introduced a function whereby an instance of a sound-group could be saved as a list of sound-events in parameter space. In this *object-based* format, each event is stored as a *metadata* string describing the sound source, its resampling rate, spatial position and so on (see page 60). Access to this metadata opens up a number of possibilities:

Spatial Re-Orchestration

With the storage of sound-groups as events in an abstract space, it becomes possible to take a stratified approach to spatial composition.[54] Sound-groups may first be rendered to audio in the studio and arranged there and later, should a concert opportunity arise, be re-rendered to the same plan, for performance by whatever method (for example VBAP, Ambisonics[55] or Wave Field Synthesis[56]) and for however many speakers are available at the target site. Thus it may become possible to compose works that can be spatially re-orchestrated for a variety of performance systems with relative ease.

Surgical editing

In broad-brush techniques such as this, it is often difficult to perform surgical edits on near perfect results. While it may generally be quicker to run the process a few more times until one obtains a better version, it is now possible to edit inconvenient events within the metadata and then re-render the modified material.

Logical editing

Once a sound-group has been written to metadata, it becomes possible to change it within SuperCollider by searching for events whose parameters meet certain conditions and modifying them either by changing parameter values or by treating the sound file used by those particular events.

For example, because its only spectral dimension is resampling, Audio Spray Gun lends itself to the generation of long sounds whose low frequency components are of extended duration and tend to

[54] Nils Peters, Trond Lossius, Jan Schacher, Pascal Baltazar, Charles Bascou and Timothy Place, 'A Stratified approach for sound spatialization.' *Proceedings of the 6th Sound and Music Computing Conference*, Porto, 2009.

[55] David G Malham and Anthony Myatt, '3-D Sound Spatialization Using Ambisonic Techniques.' *Computer Music Journal* 19.4 (1995), 58-70.

[56] Berkhout, A. J., de Vries, D. and Vogel, P. (1993). Acoustic control by wave field synthesis. *The Journal of the Acoustical Society of America*, 93:5.

rumble on after higher frequency elements have died out. It is possible to use metadata to locate lower frequency events and apply gain envelopes to them, constraining their length so as to compose more abrupt features.

Substitution of different sound files

A further use of metadata is to substitute different sound samples for the one used when the group was first generated. This can be done globally, for example by rendering a second copy of a sound-group with a different sample and perhaps layering it over the first, or locally, using probability to skew the choice of sound sample over the duration of the group.

No formal tools for manipulating metadata from Audio Spray Gun have yet been implemented but a number of experiments in re-rendering and logical editing have been tried with some success.

9.2 Future Possibilities

9.2.1 Applications to Concatenative Synthesis

As discussed above, the fuzzy-probabilistic hybrid used in this study seems to function well when applied to the generative selection of events from continuous source parameters but does less well when applied to small and/or disparate discontinuous datasets. This is particularly noticeable in Audio Spray Gun because of the internal timbral variations and the range of durations that exist at the level of sound-objects. It would be interesting however to apply this method to the large corpi of short, uniform duration, granular sources of the kind generated by concatenative synthesis systems like CataRT,[57] and to compare it, in terms of choices made and computational resources required, to the geometrical *nearest neighbour* approach currently employed.

---

[57] CataRT – IMTR <http://imtr.ircam.fr/imtr/CataRT> last accessed 2 September 2016

Another possible use of fuzzy rules in this environment would be to address multiple descriptor sets simultaneously combining, for example, 'brightness' and 'noisiness' into a single, more perceptually useful parameter.

9.2.2 More Ideas for Spatio-spectral Tool Design

While the Audio Spray Gun design is fast reaching maturity, I think there may be other possibilities for tools that simultaneously generate and spatialise multiple sound-objects.

Perhaps some kind of 'modular synth' approach could be employed that allowed different parameters to be patched together. Possibilities might include correlating image spread to event resampling rate; using other live or synthetic sound sources; patching different algorithms into different axes and so on.

Another possibility might be to develop the Audio Spray Gun idea into a geometry-based reverberation tool[58] in which copies of a source could be delayed and attenuated according to their spatial position with respect to a number of virtual surfaces distributed around the sound space.

9.2.3 Integration with other object-based methods

Perhaps the biggest issue that has arisen from this metadata approach has been that of portability between three-dimensional sound systems. With the ever-decreasing cost of processing power, systems for the real-time rendering of audio for variable numbers of loudspeakers are becoming affordable. As a result, object-based audio applications are starting to become available in cinema,

---

[58] Nicolas Tsingos. 'Pre-Computing Geometry-Based Reverberation Effects For Games'. *AES 35th International Conference*, London 2009. <http://www-sop.inria.fr/reves/Nicolas.Tsingos/publis/ntsin_aes35_reverb.pdf> last accessed 2 September 2016.

home video[59] and other fields.[60] I think it would be desirable therefore to develop some system for automatically assembling works from their component metadata and storing them in formats appropriate to these systems. Perhaps developing an object-based substitute for the conventional track-based DAW.

9.3 End Note

While computers have made many advances in past years in the fields of audio synthesis and processing, systems for the organisation of sound in time (and, I would argue, in space) still exhibit little more sophistication than word processors do for text. This is to say that they merely automate familiar tasks by imitating paradigms embodied in long-established real world technologies. At best, a high quality multi-track studio in one's laptop significantly reduces the amount of time required to compose but, at worst, it reduces all creativity to the mind-numbing tyranny of cut-copy-paste.

It's been a while coming but relatively inexpensive computers are now capable of creating and utilising sound sources of a complexity that was, until recently, unimaginable. The technologies required to arrange and spatialise vast numbers of sound-events and to have those events respond to human beings in real time already exist. What we lack are the necessary tools to investigate their artistic potential. This portfolio is an excursion into this new world of musical possibilities.

---

[59] e.g. Dolby Atmos, <http://www.dolby.com/us/en/brands/dolby-atmos.html>, last accessed 29 September 2016
[60] see ORPHEUS audio project, < http://orpheus-audio.eu/ > last accessed 29 September 2016

Appendix A: Programming Methods

A.1 A short introduction to Fuzzy Logic[61]

A.1.1 Crisp vs. Fuzzy

As was stated in the introduction, many real world questions are too vague to answered by crisp (bivalent) logic. The question 'are you tall?' cannot usefully be answered by the statements *true* or *false*. Fuzzy logic uses values between 0.0 (complete falsehood) and 1.0 (absolute truth) to show the degree to which answers to such vague questions can be true.

A.1.2 Membership Sets

Consider again the question 'are you tall?' If we assume that 5'0" is *not-tall*, and 6'0" is *tall* then, using fuzzy logic, we can define membership values for all the heights between. Thus, the fuzzy membership set for *tall* might look like Table 5.

Height	≤5'0"	5'1"	5'2"	5'3"	5'4"	5'5"	5'6"	5'7"	5'8"	5'9"	5'10"	5'11"	≥6'0"
Tall	0	0.02	0.08	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.92	0.97	1

Table 5: A Fuzzy Membership set for tall

In this set, an individual who is 5'0" has a membership value of zero, one who is 6'0 would score 1.0 and another who is 5'8" would be have a degree of tallness of 0.7. Membership values do not have to be linearly distributed and it is possible to be a member of more than one set at a time. For example, an individual of height 6' 1" could have differing membership scores in the sets *tall* and *very tall* simultaneously.

---

[61] This section draws extensively on examples from Peter Elsea's papers: 'A Fuzzy Logic Primer' and 'Fuzzy Logic and Musical Decisions'.

A.1.3 Fuzzy Pitch Class Sets

To take a more musical example, consider the idea of a *fuzzy pitch class set*.

Table 6 shows a crisp logic representation of the pitch class set[62] for C major:

C	C$^\sharp$/D$^\flat$	D	D$^\sharp$/E$^\flat$	E	F	F$^\sharp$/G$^\flat$	G	G$^\sharp$/A$^\flat$	A	A$^\sharp$/B$^\flat$	B
1	0	1	0	1	1	0	1	0	1	0	1

Table 6: Crisp Pitch Class Set for C Major

Here, pitch classes are either in the scale or they are not but if one is creating an installation or an agent whose behaviour hints at a particular style, one can bias the pitches it favours using a fuzzy pitch class set as in Table 7.

C	C$^\sharp$/D$^\flat$	D	D$^\sharp$/E$^\flat$	E	F	F$^\sharp$/G$^\flat$	G	G$^\sharp$/A$^\flat$	A	A$^\sharp$/B$^\flat$	B
1.0	0	0.75	0	0.80	0.20	0	0.60	0	0.75	0	0.80

Table 7: A Fuzzy Pitch Class Set for C Major

Once again, non-members of the C major scale score zero but the member pitches now have a variety of values. For an improvising musician, this set might represent an answer to the question "What notes do you like playing over a C major chord?"

A.1.4 Union and Intersection

In binary logic, one can test the truth of a combination of statements using methods of *intersection* (AND) and *union* (OR) such that:

if both A AND B are *true* then the result is *true;*

if either A OR B is *true* then the result is *true;*

---

[62] see David Cope, *Techniques of the Contemporary Composer,* (New York: Schirmer 1997)

Fuzzy logic has this too but here the degree to which two statements are true effects the combination. Thus in fuzzy logic, given membership values for two sets *A* and *B*, membership of the set *A* AND *B* is defined as the minimum of the two

A AND *B* = minimum(*A*, *B*)

and membership of *A* OR *B* is the maximum,

A OR *B*   = maximum(*A*, *B*).

So, if an individual has membership values of 0.7 for the set *tall* and 0.3 for the set *heavy*, then they will have a membership of 0.3 for the set *tall* AND *heavy* and 0.7 for the set *tall* OR *heavy*.

We can apply this approach to combine selection mechanisms for generative processes.

A.1.5 Fuzzy Decision Making

In crisp logic computer systems, decisions are made by conditional statements, which often take the form, **IF** {a condition is *true*} **THEN** {perform some action}. In fuzzy systems, however, the condition to be tested will reflect a degree of truth somewhere between zero and one and the resulting action will vary according to that degree.

The fuzzy equivalent of a conditional statement (called a *fuzzy rule*) therefore describes the relationship between a set of possible input values (*predicates*) and a set of outputs (*consequents*). Fuzzy rules can be represented as a table (Table 8), a graph (Figure 29) or, in programming by a mapping between a pair of lists (for example, [0, 3, 8, 12], [1, 1, 0, 0]). All three representations could

illustrate a rule for controlling the activity of a number of agents.[63] Here, the probability of an agent

acting at a given moment is 100% if the number of agents in action is less than or equal to three and

falls to zero as the number of active agents approaches eight, remaining at zero for any higher

values.
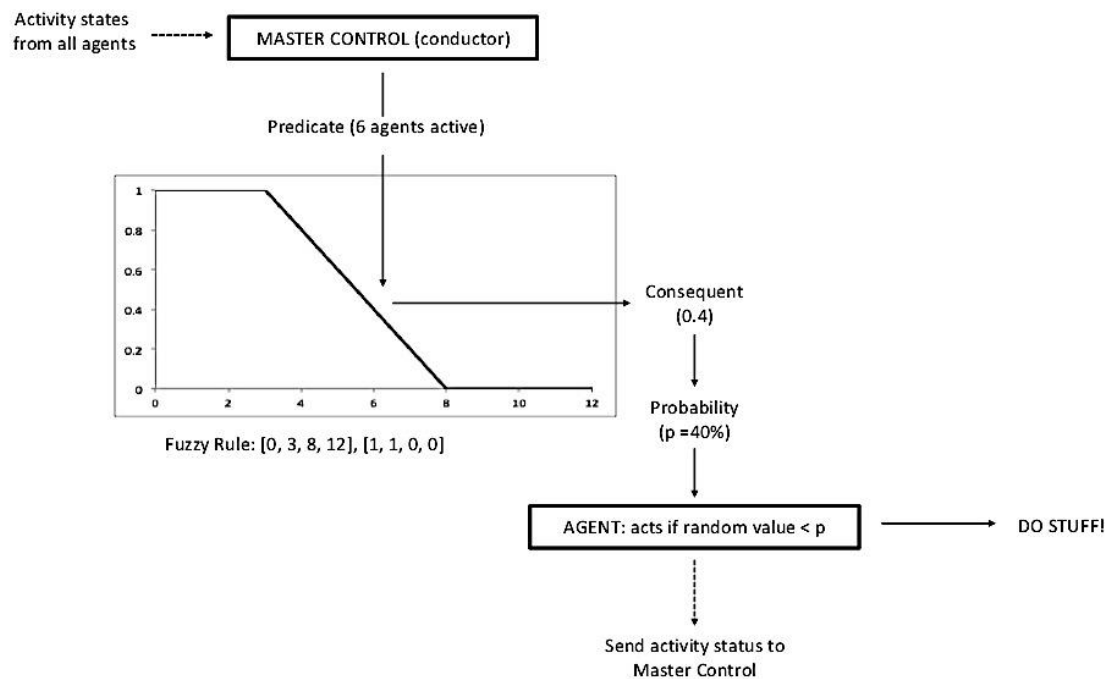
Figure 29: A fuzzy density rule

n	0	1	2	3	4	5	6	7	8	9	10	11	12
p	1	1	1	1	0.8	0.6	0.4	0.2	0	0	0	0	0

Table 8: Fuzzy Density Rule as a Table

Individual fuzzy rules are usually fairly simple in form and most can be represented as triangles or

rhomboids.

---

[63] as used in nwdlbots, harvest, exploration patch

A.2 Methods

A.2.1 Fuzzy-probabilistic parameter selection

The most pervasive generative technique used in this portfolio involves the logical combination of fuzzy sets to produce a weighted probability table. To show how this works, here is an example using fuzzy sets to constrain pitch choices made by a software agent (Figure 30).

The first step is to establish a *universe of discourse*, that is the fixed range of pitches available to the agent. In this case, it is the octave and a fifth starting at middle C. Next, the set of available pitches is constrained by combining the universal set with a set derived from the fuzzy pitch class set used in Table 7, by means of a fuzzy AND statement. This reduces the choice of notes available to from 17 equal values to 12 varying ones.

Now, the set of choices can be constrained further by adding an interval set. This particular agent has a rather limited taste in interval choices, disliking unisons, favouring seconds and, to a lesser extent, thirds. Whereas the scale set is defined in terms of absolute pitch, the interval set is transposed each time a note is played. This illustration assumes that the last pitch played was G3, so the interval set is centred on that value.

At this point, the process employs a second AND operation to combine the interval set with the earlier result producing a membership set containing five possible pitches. In control applications, the final step would be to de-fuzzify this set by selecting the item with the highest membership value as the 'best fit'. However, for the purposes of generative composition, it's more appropriate to normalise the membership values so that they add up to 1, thus producing a weighted probability table from which the agent can make its next choice.
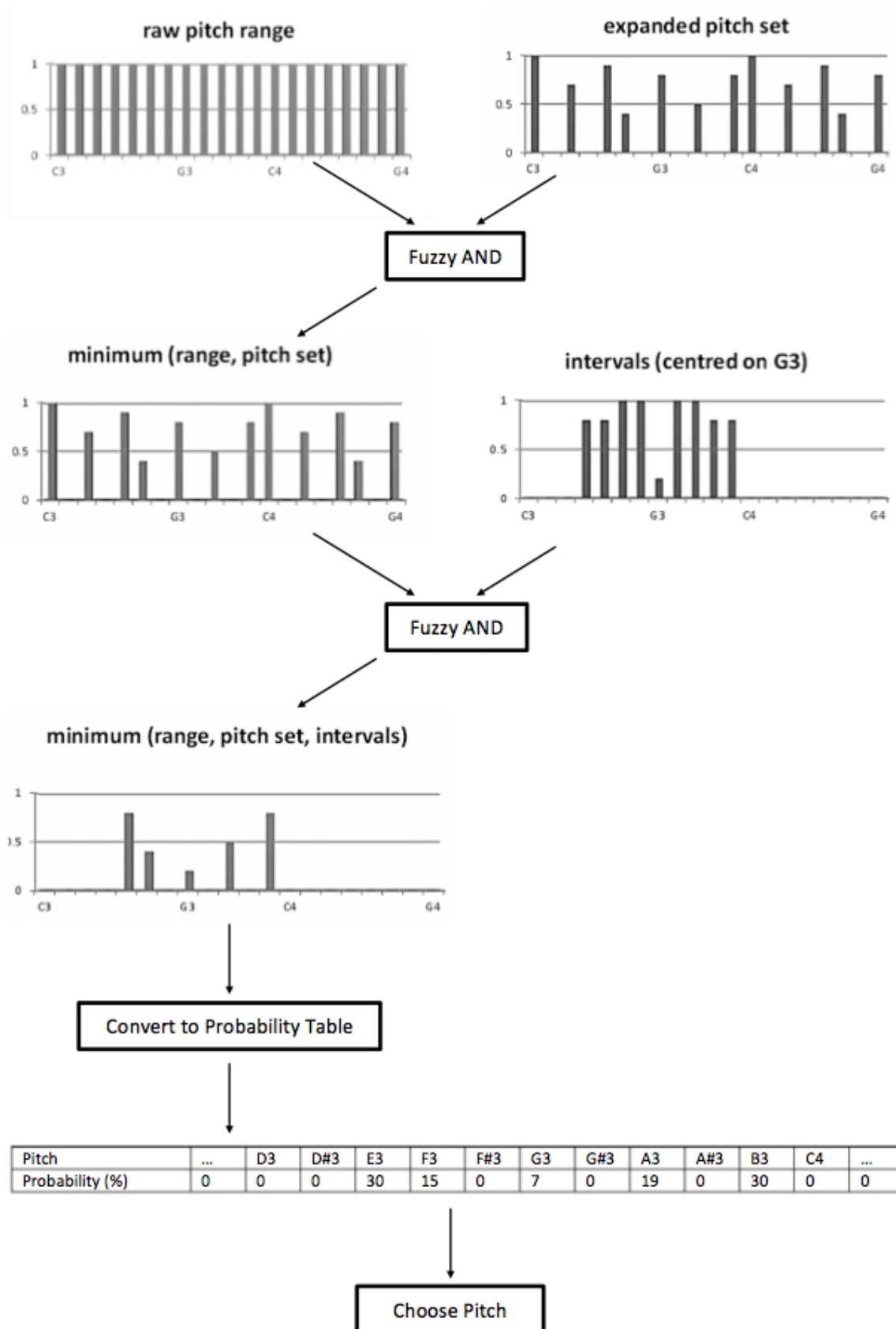
Figure 30: A Fuzzy- Probabilistic Pitch Selector

Once a pitch has been selected, the interval routine will be transposed so that the next set of pitch choices can be calculated relative to the new value.

This process was used as illustrated in *nwdlbots*,[64] a suite of generative devices for Ableton Live.

A.2.2 Selection from a data array

The method shown in the previous section can be expanded to analyse multi-parameter data arrays as shown in the following example.

Imagine a corpus of sound-objects in which each member has been analysed and described in terms of 'noisiness', duration, 'brightness' and so on. To extract long duration objects from the corpus, one could define a fuzzy rule for the condition *long*, apply it to the duration of each object in turn and then store the resulting membership values in an array. The *long* events would then have the highest scores.

Figure 31 applies this method to two descriptors in order to extract events that best fit a pair of conditions. Section (a), shows corpus of 17 objects as they are described by two parameters, the first with a range of values between 0 and 100, and the second in range 0 to 23. Section (b) shows the rules defining the events we wish to extract (events with values of 'around 50' in parameter 1 and 'around 18' in parameter 2). Section (c) shows the membership sets for the whole corpus derived from each rule in which every object attains a 'score' between zero and one describing how well it complies. Then, to find the objects that most nearly fit both criteria, we perform a fuzzy AND operation on the two sets (d).

Figure 31: Application of rules to Parameter sets (overleaf)

---

[64] nwdlbots <http://www.sundaydance.co.uk/nwdlbots/>

(a

(b

(c

(d

(e

(f

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Probability (%)	0	0	37	0	0	0	0	0	0	0	21	0	0	0	0	42	0

109
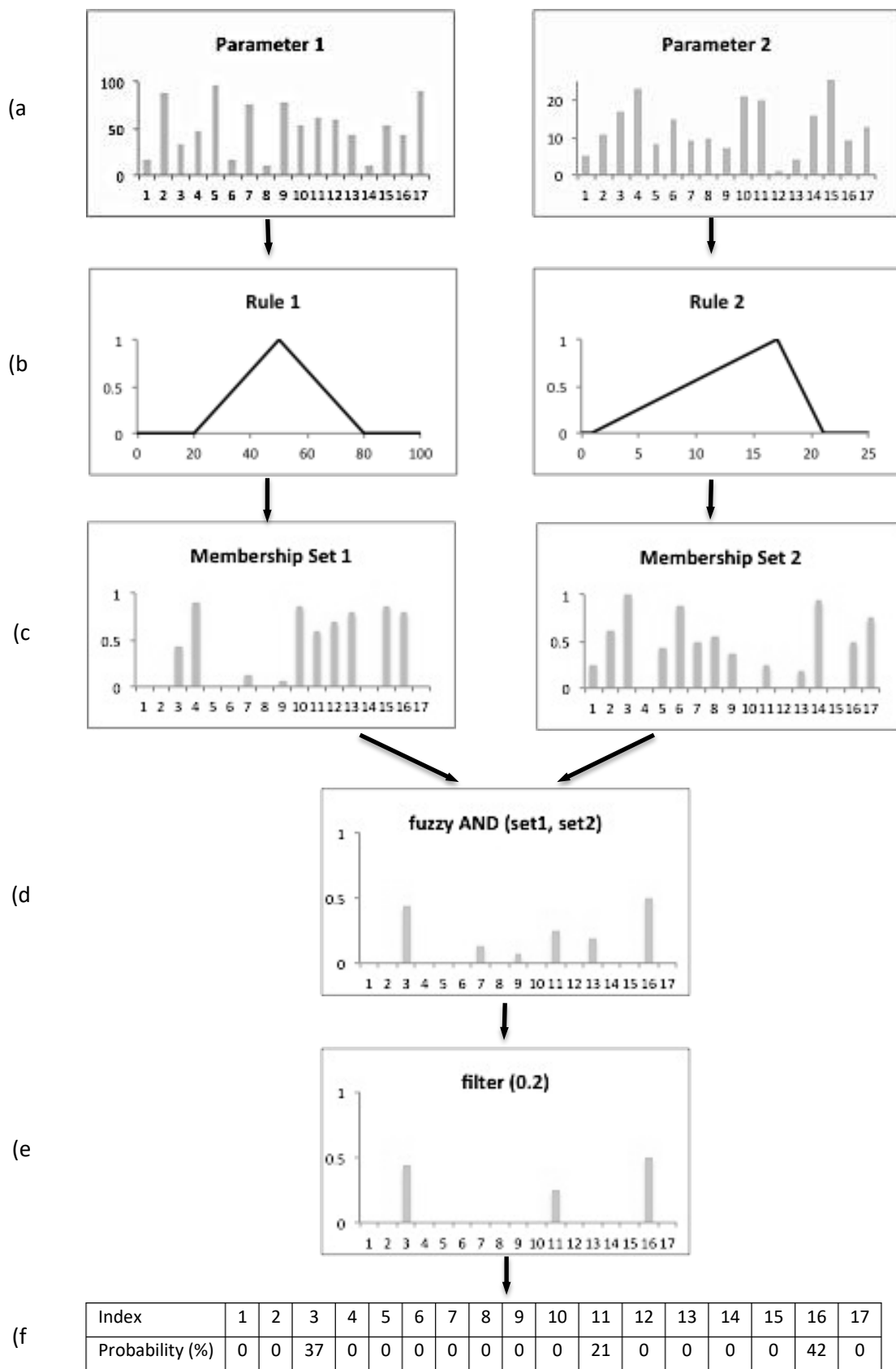
At this point, the result could be de-fuzzified by taking the maximum value in the combined set and using the object at that index. Alternatively, the set could be converted into a probability table as it stands but this method can generate disparate weak solutions with highly uncertain results so a third possible approach is to filter out all solutions that score less than a certain value (e) and generate the probability table from what remains (f).

A.2.3 Rejection Zones

This example deals with the use of fuzzy intersection to distribute events around a multichannel speaker system by the insertion of *rejection zones* into a probability table for azimuth (as used in *Harvest* and *Guitar Piece*).

Consider a set of 100 values between zero and one as shown in the first row of Figure 32. For many applications, the initial state of this process would offer an equal chance of selection to all events in the set but, as this example deals with azimuth, the chance of selection has been biased towards the centre of the array. When no event is sounding, the system chooses any point from the constrained set at which first event will occur. When the first event sounds, an inverted triangle (rejection zone) is incised into the probability table that reduces the chance of subsequent events occurring in close proximity to the first. This zone remains present for the duration of the first event and, as each new event occurs, another rejection zone is cut into the probability table, remaining in force for the duration of that event. If the values chosen are mapped between $-\pi$ and π radians, this sequence of events will be distributed around the sound space, avoiding currently sounding events but constrained towards the front-centre position whenever possible. Figure 32 shows this process for the automated spatialisation of nine events each with a 'lifetime' of five iterations. Listen again to the Sound Example 20 to hear a spatial distribution of sound-groups using this method.

Figure 32: Rejection Zones (overleaf)
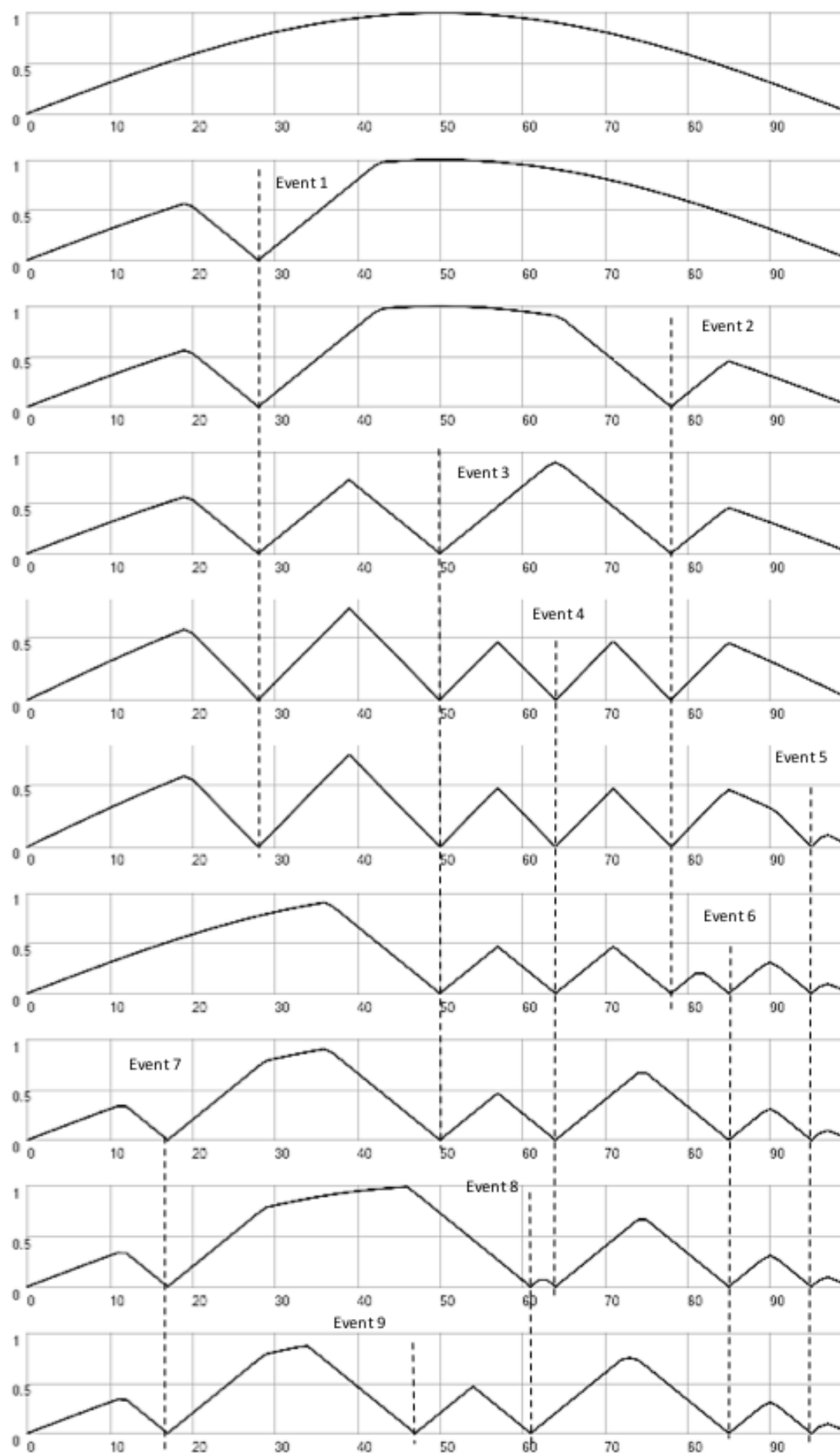
A.2.4 The Zones Algorithm: Attraction and Rejection

Here is a one-dimensional example of the *Zones* algorithm, first used in *The Inside Track*.

Whenever an event occurs, two sets are created which control the value assigned to the next event. The first of these is a *rejection rule,* for which all values are equally probable except those in the immediate vicinity of the last event, whose chances of selection decline with proximity. The second generally wider set, called the *attraction rule*, excludes all values except those close to the previous event value, whose chances increase with proximity.

When these two arrays are combined by a fuzzy intersection (AND), the resultant probability table constrains value chosen for the next event within a fixed distance of the most recent one (not too close, not too far away). These sets need not be symmetrical about the current event location and may be 'skewed' so that event selection is biased in one direction (Figure 33).
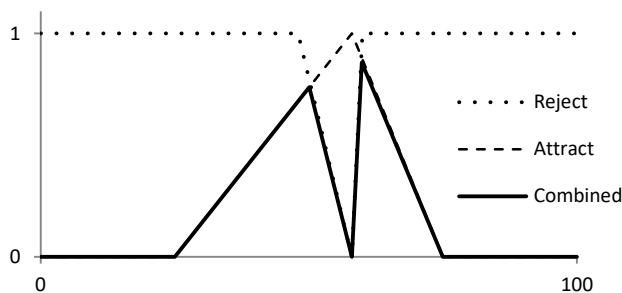
Figure 33: Skewed Attraction/ Rejection zones

This method, as used in *The Inside Track* to producing swirling spatial and spectral motion, has shortcomings in that it has no 'memory'. Thus, while the second event in a sequence is rejected from the immediate vicinity of the first, there is nothing to stop the third event returning to the first point. The full algorithm, as used in later works, improves upon this by making new event locations dependant on the positions of *multiple* events in the past. This process is illustrated in Figure 34, sections a to g:

a) Initialise three arrays of the same size: the *master array* describing the range of parameter values from which values can be chosen; the *reject array* which accumulates the reject zones generated by the sequence of points; and the *attract array* which accumulates attraction zones. To start the process, a value is selected at random from the master array. A rejection zone and an attraction zone are generated around it.

b) These two zones update the reject and attract arrays respectively using fuzzy intersection (AND) for rejection and union (OR) for attraction.

c) The intersection of the modified reject and attract arrays is written to the master array which acts as a probability table from which the second value in the sequence may be chosen.

d) The second point is chosen from the probability table along with its rejection and attraction zones.

e) All the values in the existing reject array are now incremented by some small value and these values 'clipped' to one. This slightly reduces the influence of the first event and introduces a small but non-zero possibility that the next event could occur where the first one did. The existing attract array is decremented in a similar fashion and clipped to zero.

f) These two arrays are then combined with the zones generated by point 2 using the same method as in section (b).

g) A new master Array is generated using the same method as (c) and the next point is chosen.

The process continues for the duration of the sound-group with the influence of each event steadily diminishing as new events occur.

Figure 34: The Zones Algorithm (overleaf)

a)

Master Array 0

Reject Array 0

Attract Array 0

b)

Point Reject 1

Point Attract 1

AND

OR

c)

Reject Array 1

AND

Master Array 1

Attract Array 1

d)

Increment

Decrement

Point Reject 2

Point Attract 2

e)

AND

OR

f)

Reject Array 2

AND

Attract Array 2

g)

Master Array 2
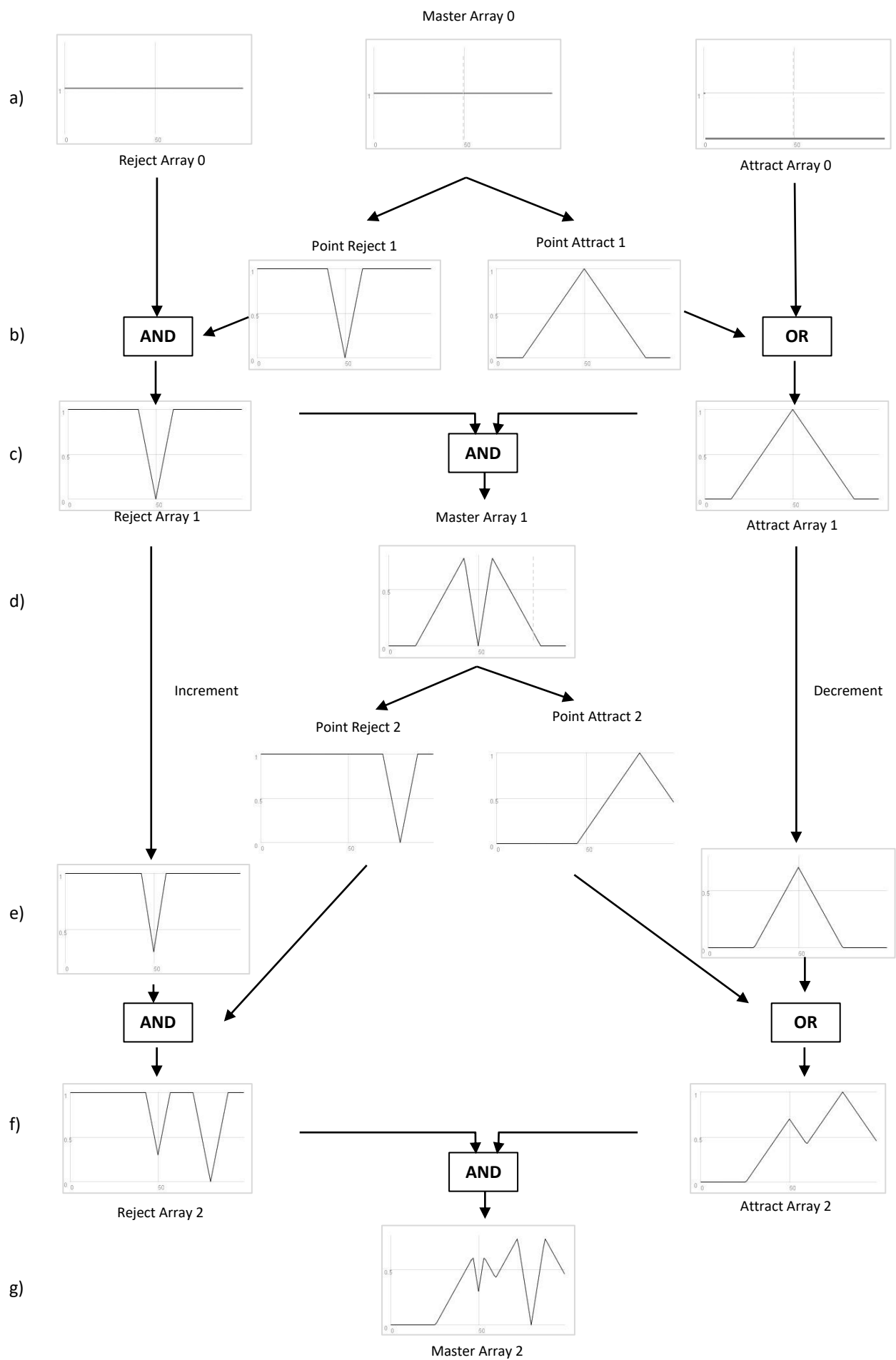
The effects generated by the Zones algorithms are controlled by five factors, three governing the shape of the attraction/ rejection zones and two their recovery rate.

The width of the rejection zone will control the spacing between events, low widths producing lots of close values, high widths spreading them out. The attraction zone width restricts the maximum step between values. If zones are skewed, new values will be allowed further from the centre to one side of the attraction zone than to the other, causing a net movement of values to that side.

The process is also dependent on the recovery rates used to increment and decrement the reject and attract arrays at each step of the process. By varying these independently, a number of effects can be achieved. If n_R is the number of iterations required to restore the rejection zone associated with a particular event to 1.0, and n_A is the number of iterations required to restore the corresponding attract zone to 0.0, then

Reject rate (R) = 1/ n_R, and

Attract rate (A) = 1/ n_A.

R	A	Narrow	Wide
$\Rightarrow \infty$	$\Rightarrow 0$	A bag empties after narrow range of values, starts again at different spot. Mixture of random walk and bag.	A bag. Items are selected once and will not be re-selected until all have been chosen.
$\Rightarrow \infty$	$\Rightarrow \infty$	Events move away from initial value in both directions as attract and reject arrays expand. Two groups slowly decay into random behaviour.	Events move away from initial value in both directions (according to skew), quickly becoming random.
$\Rightarrow 0$	$\Rightarrow 0$	Random walk.	Randomness
$\Rightarrow 0$	$\Rightarrow \infty$	Random within an expanding zone.	Zone expands quickly then decays into randomness

Table 9: Reject and Attract Rates

Table 9 summarises some of the extremes of behaviour when different reject and attract rates are applied to narrow and wide zones.

This technique can be combined with the process of scoring a data array (page 108) to produce varieties of random selection in non-continuous data sets.

Appendix B: Audio Spray Gun

B.1 Introduction

Audio Spray Gun[65, 66] is an experimental tool for fixed-media composition that simultaneously generates and spatialises large groups of events, all derived from a single sound-object (typically between 250ms and a few seconds in duration). In this approach to spatial sound synthesis[67], events are created as points constrained by a locus or *particle zone*[68] within a virtual parameter space. By transforming this locus over time, the program produces a sequence of events, which can then be rendered to multichannel audio.

At first, the program was designed to render audio direct to eight-channel surround, but more recent versions include three-dimensional spatialisation. Audio Spray Gun also has the capacity to store sound-groups as metadata for later editing and spatial re-orchestration.

Audio Spray Gun is so named for its similarity to the spray gun function common to many computer graphics programs, in which dots are distributed at random within a moving locus on a canvas. However, because the program operates in a time-based medium, its desired output is a collection of events constrained by the trajectory and transformation of the locus rather than a final static image.

---

[65]Richard Garrett, "In Flight and Audio Spray Gun" *eContact! 17.3* 2015.

[66]Richard Garrett, 'Audio Spray Gun 0.8 – the Generation of Large Sound-Groups and Their Use in Three-Dimensional Spatialisation', *Proceedings of the 41st International Computer Music Conference,* Denton 2015, pp. 352-355.

[67] Nuno Fonseca, '3D particle systems for audio applications.' *Proceedings of the 16th International Conference on Digital Audio Effects (DAFx-13)*, Maynooth, Ireland 2013, <http://dafx13.nuim.ie/papers/25.dafx2013_submission_50.pdf> last accessed 22 May 2016.

[68] Marlon Schumacher and Jean Bresson, 'Spatial Sound Synthesis in Computer-Aided Composition.' *Organised Sound* 15:3 (2010) 271-289.

B.2 Theory

In Audio Spray Gun, the trajectory and transformation of a sound-group is defined using twelve

function generators, one for each of the twelve values described above. Two modes of operation are

available so that these functions are dependent either upon time elapsed (as a fraction of a

predefined total duration) or on the number of events played so far (as a fraction of a predefined

total number).

The general format of these functions is

value(*x*) = *add* + *f* (*mult* × *env*(*x*))

where *add* and *mult* are constants and *env*(*x*) is the height of a curve or envelope (0 – 1) at point *x*,

defined as either

x = (*time now*)/(*total duration*)

or

x = (*index of current event*)/(*total number of events*)

according to operating mode.

The function *f* can be selected from a number of options, for example:

linear   *add* + (*mult* × *env*(*x*))

rand   *add* + *random*(*mult* × *env(x)*)

rand2   *add* ± *random*(*mult* × *env(x)*)

Typically, the trajectory of the locus will be described by linear functions and its extent will be

controlled by random functions.

To generate an event, the program calculates the position and extent of the locus by evaluating these twelve functions for the current value of *x*. It then selects a random point from within that locus, which is converted to an event location with respect to the listener.

This process repeats in real time until the end-point of the functions is reached. As each event occurs, it is played back as audio and added to a data array so that the sound-group may be replayed later or stored as metadata.

B.3 User Interface

The program interface consists of three windows (Figure 36):

The *parameter window* contains twelve graphical function generators that define the trajectory and transformation of the sound-group. Each function generator (Figure 35) uses a multislider to display a function curve, the shape of which can be edited using the mouse or selected from a menu of pre-defined shapes. The curve can also be either inverted or reversed using a single mouse click and the 'mode' menu allows the user to choose which type of function will be output. Spatial curves may be defined in spherical, cylindrical or Cartesian coordinates.
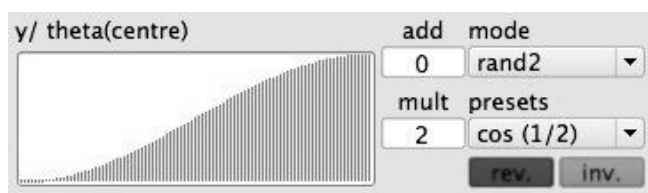
Figure 35: A function generator

Display Window

display (radius: 1948 metres)

Parameter Window

GrillTrayRattle+reverb+fadeIn_n_210_spiralBoom_params

x/ rho(centre)   add   mode
0   linear
mult   presets
2000   zero line
rev.   inv.

y/ theta(centre)   add   mode
0   linear
mult   presets
-2   zero line
rev.   inv.

z/ phi(centre)   add   mode
0   linear
mult   presets
0.6   zero line
rev.   inv.

spectral(centre)   add   mode
0.5   linear
mult   presets
4   zero line
rev.   inv.

spread(centre)   add   mode
0   linear
mult   presets
25   zero line
rev.   inv.

delta (centre)   add   mode
0.01   linear
mult   presets
0.07   zero line
rev.   inv.

x/ rho(event)   add   mode
0   rand
mult   presets
30   zero line
rev.   inv.

y/ theta(event)   add   mode
0   rand2
mult   presets
1   zero line
rev.   inv.

z/ phi(event)   add   mode
0   rand2
mult   presets
0.1   zero line
rev.   inv.

spectral(event)   add   mode
0   rand
mult   presets
0.5   zero line
rev.   inv.

spread(event)   add   mode
0   rand
mult   presets
10   zero line
rev.   inv.

delta (event)   add   mode
0   rand
mult   presets
0.01   zero line
rev.   inv.

Audio SprayGun 0.8.1

/Users/richard/Documents/SC workspace/sources/kitchenDoors/GrillTrayRattle+reverb+fadeIn.wav   source

New   Open   Save   co-ordinates   Spherical   dur (s)   events   volume
30   210   0.80
Events   Delay Off   Show   Go   Play   Write

events: 210   duration (onsets): 9.56 seconds

Launch Window

Figure 36: Audio Spray Gun Interface (version 0.8.1)

120

The *launch window* holds controls for program execution and data storage. From this window, the user can trigger and replay sound-groups; select the sound file from which events will be created; load and save parameter sets; select between time and event mode of operation or save sound-groups as metadata. The user can also define the length of the group in either events or seconds; set the maximum gain for individual events; and apply a delay to each event proportional to its distance from the listener. This delay is approximately equivalent to the speed of sound in air (340 ms^{-1}).

The *display window* shows an animation of the spatial locations of events as they occur.

Audio Spray Gun is written entirely in SuperCollider 3.6.3.[69]

---

[69] James McCartney, 'Rethinking the computer music language: SuperCollider', *Computer Music Journal*, 26:4 (2002) 61–68

Appendix B: Exploration Patch

C.1 Introduction

This is a short technical discussion of the software used in *Exploration I* and *II*.

The software consists of a patch written in Max 6.1[70] using IRCAM Mubu[71] 1.6.7 multi-buffer externals.

	Channel 1	Channel 2	Channel 3	Channel 4
Duration	short 0 1 2 3 4 long	short 0 1 2 3 4 long	short 0 1 2 3 4 long	short 0 1 2 3 4 long
Amplitude	soft 0 1 2 3 4 loud	soft 0 1 2 3 4 loud	soft 0 1 2 3 4 loud	soft 0 1 2 3 4 loud
Frequency	low 0 1 2 3 4 high	low 0 1 2 3 4 high	low 0 1 2 3 4 high	low 0 1 2 3 4 high
Freq. Deviation	pitchy 0 1 2 3 4 noisy	pitchy 0 1 2 3 4 noisy	pitchy 0 1 2 3 4 noisy	pitchy 0 1 2 3 4 noisy
Totals	28 28 28 19 20	19 28 28 19 11	28 28 28 27 28	10 28 23 27 5
Resampling – Resample	-1200.	1200.	700.	-2400.
Resampling – ResVar	1200.	1200.	700.	1200.
Envelope – Attack	0.7	0.5	0.5	0.5
Envelope – Release	0.3	0.3	0.3	0.3
Filter – Mode	off	resonant	resonant	off
Filter – Freq	0.	1024.	256.	0.
Filter – FreqVar	0.	0.5	0.5	0.
Filter – Q	0.	2.	1.	1.
Filter – Qvar	0.	0.5	0.	0.
Filter – Gain	0.	1.	1.	1.
Ducking Rule – Duck	0.	0.	0.	5.
Voices Rule	Voices 0 3 -ve	Voices 0 3 -ve	Voices 0 3 -ve	Voices 0 3 -ve
Audio Output – level	0.4	0.4	0.4	0.4
Audio Output – pan	-0.1	0.2	-0.2	0.0
Density – chance	17.	24.	24.	10.

inlet — MUTED — 1.
threshold — onset 0.005 — taboo dur (ms) 20. — endpoint 0.003 — taboo peak 0.005
level 0.6 — pan 0.0 — LIMIT — p initChannels — 0

EXPLORATION
Richard Garrett 2014

STOP / RESET 02:29

Density — ADC — REC — PLAY — chance

master level 1. — Out 1 2

Playback

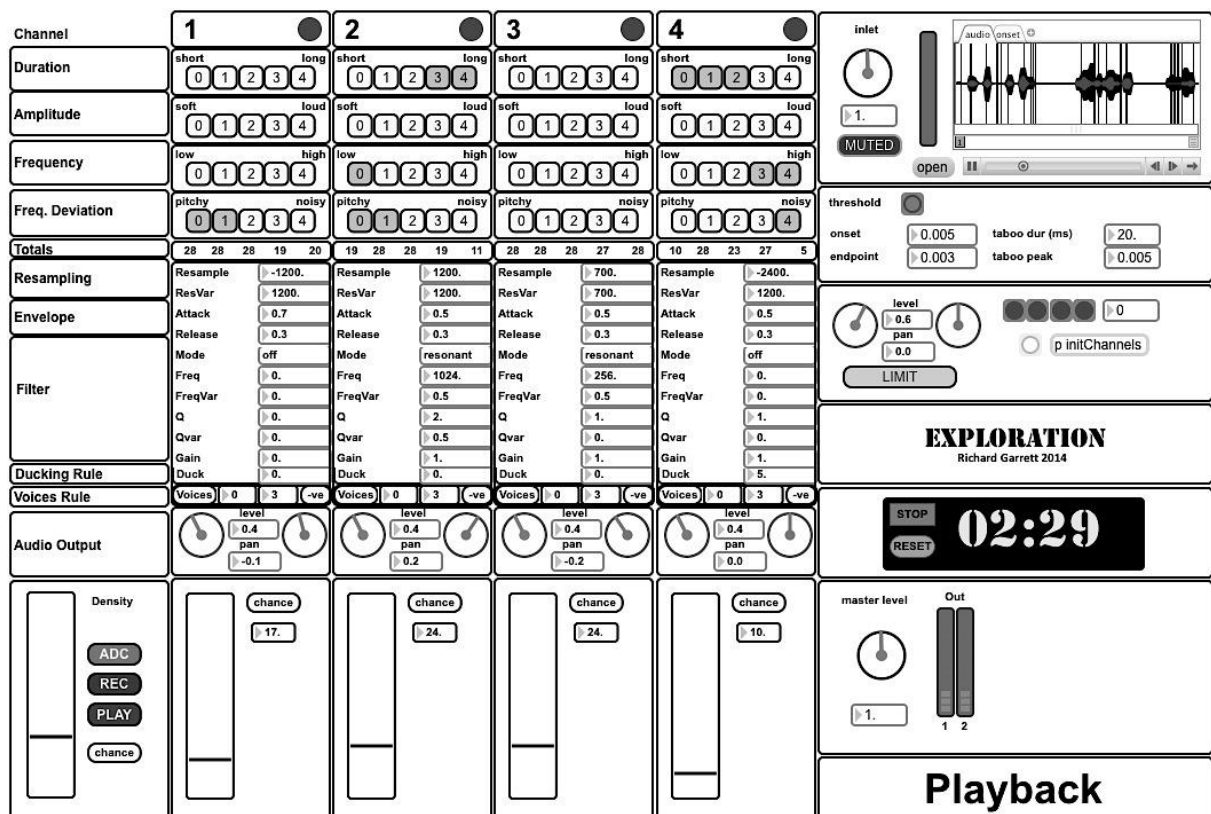*Figure 37: The Exploration Patch*

The functions of the patch are as follows:

- To take audio signals from the live instrument and record them into a buffer;

- To identify discrete audio events and categorise them with reference to a number of descriptors;

---

[70] Max, Cycling 74 <http://cycling74.com/>
[71] Mubu, < http://forumnet.ircam.fr/product/mubu-en/> last accessed 22 May 2016.

- To test each event recorded for membership of a number of fuzzy sets used by each of four fuzzy-probabilistic agents presented as 'channels';

- To combine these sets and convert them into probability tables governing the choice of events played back by each channel; and then

- To select events using these probability tables and play them back, generating a stereo accompaniment from the events recorded so far.

During playback, the laptop player can adjust the membership set definitions for each channel; set the probability that a particular channel will sound; and control the overall density of the accompaniment. He/ she can also alter resampling rate, amplitude envelope, filtration, panning and gain for each channel.

Some elements of the Exploration Patch are derived from the patch for *Out of the Loop (2013)* for hexaphonic guitar written during my MA.[72]

C.2 Event Detection

Sound from instrument is recorded into the microphone and stored in a Mubu track using the **mubu.record~** object. This audio signal is also passed to the event detection system and the stereo output bus.

Event detection consists of three processes:

Figure 38: Exploration Channel

[72] Richard Garrett. 'Out of the Loop'. SoundCloud. < https://soundcloud.com/sundaydance/out-of-the-loop> last accessed 22 May 2016.

Onset Detection: Audio input is passed through a two level onset detection routine that returns an index plus onset time and duration for the incoming event. The onset time is then passed to a Mubu data track (using **mubu.record**). This routine also sends a flag to the Event Triggering routines to indicate that the instrument is playing.

Analysis: Once an event has been detected, it is analysed to find its peak amplitude; its mean fundamental frequency (using the Max **fzero** object); and a measure of its 'noisiness' (the average deviation of the measured fundamental frequency from its mean, sampled at intervals over the duration of the event).

Taboo Filtering: If the duration and/or the peak amplitude of the event are found to be less than predefined 'taboo values', the event is rejected as too short or too quiet to be of use. Events that pass through this filter are sent as lists (of format *index, duration, peak amplitude, frequency, noisiness*) to all four control channels.

C.3 Membership Testing

Once a new event is passed to a channel, its four component parameter values (duration, peak and so on) are tested for membership of a user-defined fuzzy set. This test set is defined by the status of five on/off buttons representing membership sets for different value ranges (for example, *very short*, *short*, *medium*, *long*, and *very long* duration). The membership set specific to that parameter on that channel is defined as the union of the sets with active buttons. Once the incoming parameter value has been tested against this set, its membership value is appended to the current list of values (one for each event so far). If membership criteria are changed by the laptop player, the whole set is recalculated.

The five component membership sets are not defined absolutely but are continuously re-calculated from incoming events in real time. This is to say that *very long* events are the longest events detected so far and are not necessarily very long in absolute time.

Updated membership sets are output to the next stage.

At a point in each piece, the laptop player activates the LIMIT button. While events recorded after this point in time are still stored and described, they are temporarily excluded from the pool of events available for playback. This is achieved by creating a crisp set in which all events recorded before the button was pressed have values of one and those recorded after this time are set to zero. When the limit is turned off, all values are reset to one, allowing all events to be included in the selection process.

Membership sets for all events up to the present moment are combined with the LIMIT set by a fuzzy AND operation. This 'master' set is then converted into a probability table used for event selection by this channel (see page 108).

C.4 Event Triggering

After the laptop player presses the PLAY button, each channel starts to trigger events for the accompaniment. The probability that an event will sound on a given channel is constrained by the combination of three criteria:

Chance: A simple probability (0.0 to 1.0) controlled by the chance slider for that channel.

Density: A density rule (see page 104), acting on the number of channels playing, controls the likelihood that another voice will sound. The master density fader defines the threshold values for this rule.

Duck Rule: When the instrument is playing, the probability that an event will occur is 'capped' at a particular value (0.0 to 1.0).

Each time these criteria change, the system calculates a *launch probability* by taking the minimum of the three values (equivalent to a fuzzy AND). This value (between 0.0 to 1.0) becomes the weight for a simple *coin toss* procedure that generates a random value between 0 and 1.0. If this is less than or equal to the launch probability ('heads'), an event will be selected and played back, otherwise ('tails') there is a fixed time delay before the coin is thrown again.

C.5 Event Selection

If the coin toss returns heads, an event is selected from the probability table and its index and duration sent to the player (**mubu.concat~**) for that channel. No further events on that channel are selected until the current one is complete.

The process continues until probability sliders are set to zero and the user presses the STOP button.

C.6 Playback

Playback is modified by user selection of values for resampling, enveloping, filters and so on.
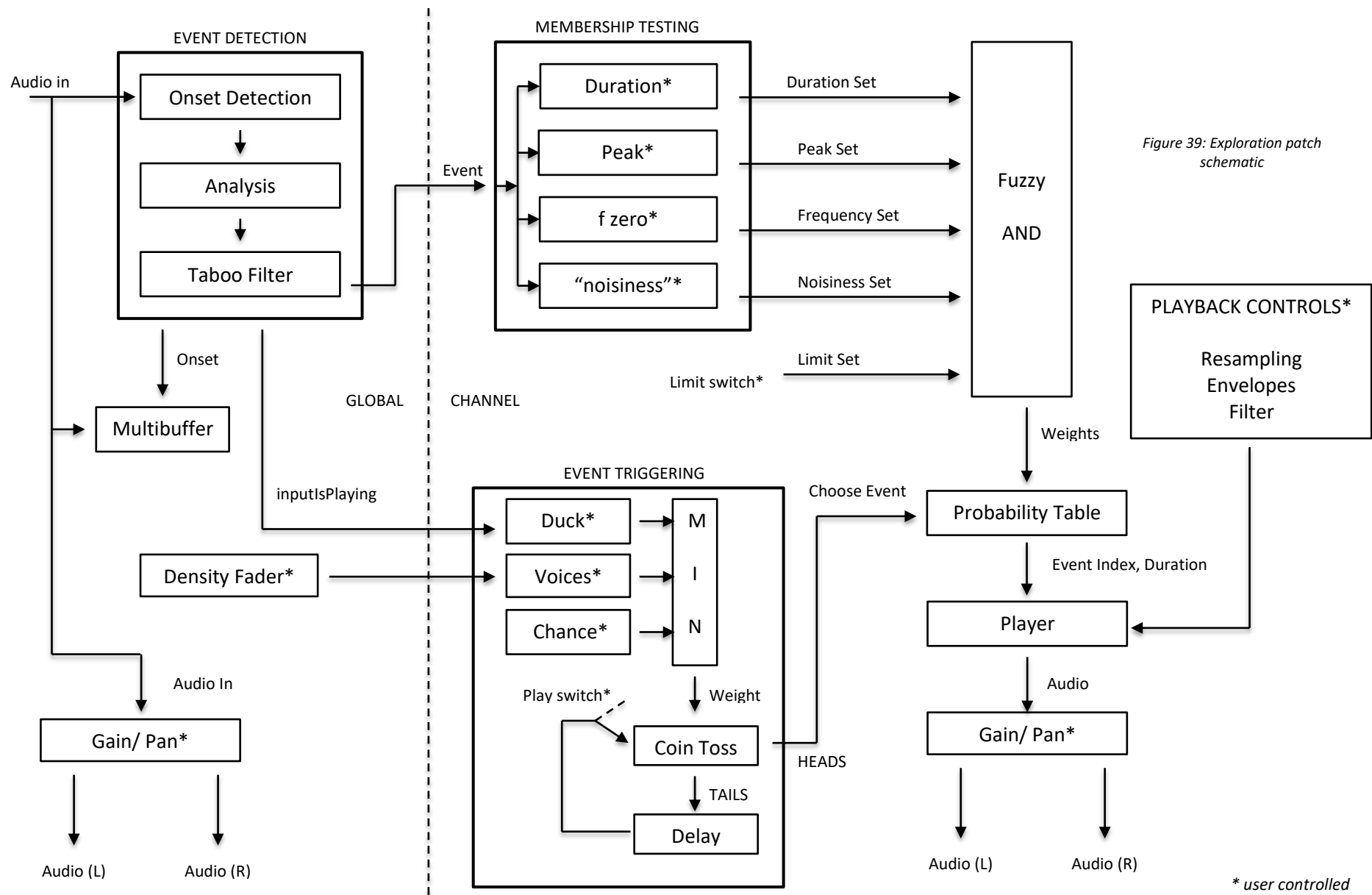Audio then output through gain/ pan controls in stereo.

Figure 39: Exploration patch schematic

Appendix D: Performances and Presentations

D.1 Performances

Date	Work	Event	Location	Performers	Comments
07/11/2013	Only Now*	Electroacoustic Wales	Bangor University, Wales, UK		Premiere
16/03/2014	Exploration (fl, laptop)	INTER/actions 2014	Bangor University, Wales, UK	Richard Craig (fl.)	Workshop
16/03/2014	Exploration (fl, laptop)	INTER/actions 2014	Bangor University, Wales, UK	Richard Craig (fl.)	Premiere
24/04/2014	Only Now	Sweet Thunder Music Festival	Fort Mason Center, San Francisco USA		
16/05/2014	November	60x60 Surround	Harvestworks New York City, USA		Premiere
19/06/2014	In Flight	USAP	Aufbau Haus am Moritzplatz, Berlin		Premiere (12.0)
27/06/2014	In Flight	USAP	Tempelhofer Feld, Berlin		12.0
04/07/2014	In Flight	USAP	James-Simon-Park, Berlin		12.0
14/08/2014	Only Now	TIES 2014	Wychwood Theatre, Toronto, Canada		
16/08/2014	In Flight	TIES 2014	Wychwood Theatre, Toronto, Canada		Lecture-recital (8.0)
17/09/2014	Only Now	ICMC/ SMC 2014	Athens, Greece		
28/10/2014	Once Below a Time	Electroacoustic Wales	Bangor University, Wales, UK		Premiere
29/10/2014	November	60x60 Surround	Jack Straw New Media Gallery, Seattle USA		

08/01/2015	In Flight	RMA postgraduate conference	Bristol University, UK		8 .0
18/04/2015	Once Below a Time	Sonorities	Goldsmiths, University of London, UK		Rolling playback (18 -20 /04)
22/04/2015	Once Below a Time	Sonorities	SARC, Belfast, UK		Rolling playback (22 -26/04)
23/04/2015	Exploration II (cl, laptop)	Electroacoustic Wales	Bangor University, Wales, UK	Hephzibah Leafe (cl.)	Premiere
12/08/2015	Only Now	ISSTC	Limerick, Ireland		
27/08/2015	Exploration (fl, laptop)	Flute Fest	Bangor University, Wales, UK	Ellie Lighton (fl.)	
27/08/2015	Exploration II (cl, laptop)	Flute Fest	Bangor University, Wales, UK	Hephzibah Leafe (cl.)	
30/09/2015	crunch!	ICMC 2015	Voertman Hall, UNT Denton, USA		Lecture-recital (8.0)
30/09/2015	crunch!	ICMC 2015	MEIT, UNT Denton, USA		Premiere (16.0)
08/10/2015	Exploration II (cl, laptop)	MA degree recital	Bangor University, Wales, UK	Hephzibah Leafe (cl.)	
25/10/2015	November	Vox Novus Festival (60x60)	Spectrum, NYC, USA		
20/11/2015	Exploration I (fl, laptop)	Postgraduate recital	Bangor University, Wales, UK	Ellie Lighton (fl.)	
27/11/2015	crunch!	Electroacoustic Wales	Bangor University, Wales, UK		
06/01/2016	November	60x60 Surround	CUBA, Münster, Germany		
06/01/2016	Penumbra	RMA postgraduate conference	Bangor University, Wales, UK		Premiere (16.0)
13/06/2016	Penumbra	NYCEMF	New York City USA		
07/08/2016	Penumbra	CUBE Fest	Virginia Tech, Blacksburg USA		148 channels
25/11/2016	Penumbra	Sonorities	SARC, Belfast UK		24 channels

| 04/12/16 | November | 60x60 Surround | CUBA, Münster, Germany | | |
| August 2017 | Objets Volants | CUBE Fest | Virginia Tech, Blacksburg USA | | 148 channels |

*Although "Only Now" (2013) predates this portfolio, it contains prototypes of many ideas developed during the PhD and is therefore included in this list for completeness.

D.2 Presentations

Date	Event	Location	Title	Comments
16/08/2014	TIES 2014	Wychwood Theatre, Toronto, Canada	"In Flight and Audio Spray Gun: Generative composition of large sound-groups."	Lecture-recital (8.0)
1/05/2015	BEAST FEaST 2015	Birmingham University, UK	"Audio Spray Gun – generation and spatialisation of large sound groups"	Lecture with 24 channel demonstration
30/09/2015	ICMC 2015	University of North Texas, USA	'Audio Spray Gun 0.8 – the Generation of Large Sound-Groups and Their Use in Three-Dimensional Spatialisation'	Lecture-recital with 8-channel reduction of *crunch!*

D.3 Publications

'In Flight and Audio Spray Gun: Generative composition of large sound-groups.', a written version of the Toronto lecture-recital above published in

eContact!, the online journal of the Canadian Electroacoustic Community edition 17.3.[73]

'Audio Spray Gun 0.8 – the Generation of Large Sound-Groups and Their Use in Three-Dimensional Spatialisation', *Proceedings of the 41st International*

Computer Music Conference, Denton 2015, pp. 352-355.

Forum contribution for Computer Music Journal (CMJ) 40:4, a special edition on High Density Loudspeaker Arrays (HDLAs) [74]

---

[73] <http://econtact.ca/17_3/garrett_audiospraygun.html>, last accessed 20 April 2016.

D.4 Radio Broadcasts/ Podcasts

A stereo reduction of *Only Now* was played on the show 'Beethoven Was Wrong', Resonance FM August 6th 2015[75]

Short stereo extract from *Penumbra* on WVTF Public Radio in Blacksburg, Virginia USA 1st August 2016[76]

'Glitching the Riff, episode 3', podcast interview with Jordan Cutler.[77]

D.5. Other Outcomes

Two anonymous peer reviews for CMJ 40:4

---

[74] Richard Garrett, forum contribution for *Computer Music Journal,* 40:4 (2016), 12-13.

[75] 'Beethoven was Wrong' 1:6 archived at < https://bwwradio.wordpress.com/> last accessed 1 September 2016

[76] 'CubeFest to Crank the Sound Way Beyond 11'<http://wvtf.org/post/cubefest-crank-sound-way-beyond-11#stream/0 > last accessed 1 September 2016

[77] 'Glitching The Riff, episode 3'. YouTube video. Posted by 'GlitchingtheRiff', 25 November 2015. <https://www.youtube.com/watch?v=cGA-Bf5ewEE>, last accessed 30 April 2016.

Appendix E: Loudspeaker Maps

The audio files for works in this submission are stored on the USB drive in the *Portfolio* folder. Each work is contained in a sub-folder labelled with the piece name and track format (16.0, 8.0, 4.0 or stereo) for example *In Flight 8_0*.

E.1 Stereo

Stereo pieces are stored as single interleaved sound files (.wav)**.**

E.2 Four Channel

Four Channel pieces are stored as four mono 24bit, 48000 Hz sound files (.wav)**,** indexed as follows:
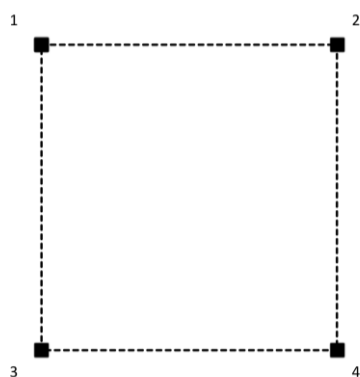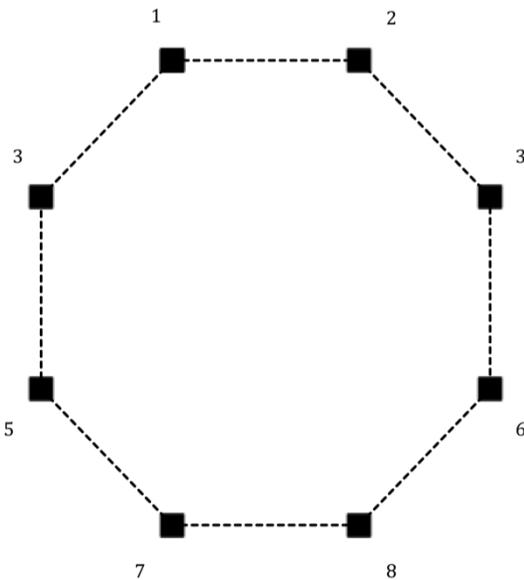

Figure 40: Four Channel Loudspeaker Map

Track Name	Location
Title_4_0_1.wav	Front Left
Title_4_0_2.wav	Front Right
Title_4_0_3.wav	Back Left
Title_4_0_4.wav	Back Right

E.3 Eight Channel

Eight Channel pieces are stored as eight mono 24bit, 48000 Hz sound files (.wav). These are numbered from 1 to 8 to represent the following positions.

Figure 41: Eight Channel Loudspeaker Map

Track Name	Speaker	Azimuth	Location
Title_8_0_1.wav	1	-22.5 °	Front Narrow Left
Title_8_0_2.wav	2	22.5 °	Front Narrow Right
Title_8_0_3.wav	3	-67.5 °	Front Wide Left
Title_8_0_4.wav	4	67.5 °	Front Wide Right
Title_8_0_5.wav	5	-112.5 °	Side Left
Title_8_0_6.wav	6	112.5 °	Side Right
Title_8_0_7.wav	7	-167.5 °	Back Left
Title_8_0_8.wav	8	167.5 °	Back Right

E.4 Sixteen Channel

Sixteen Channel pieces are stored as sixteen mono 24bit, 48000 Hz sound files (.wav). These are numbered from 1 to 16 to represent the following positions.
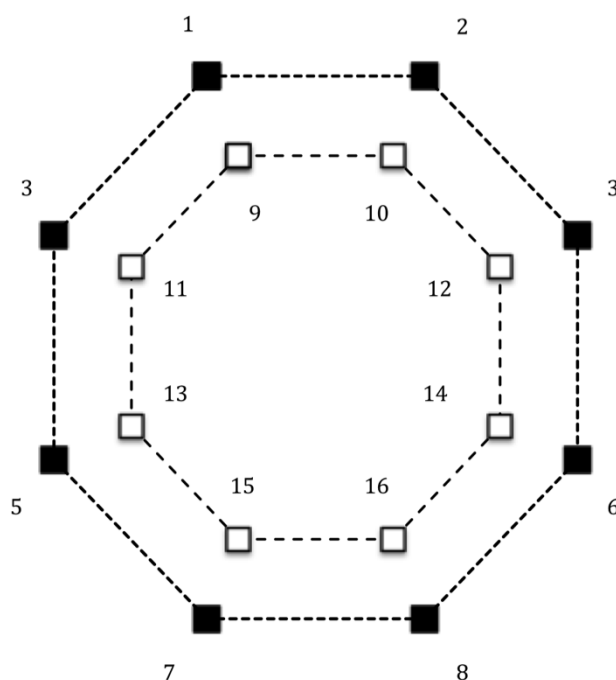

Figure 42: Sixteen Channel Loudspeaker Map

Track Name	Speaker	Azimuth	Elevation	Location
Title_16_0_01.wav	1	-22.5 °	0 °	Front Narrow Left
Title_16_0_02.wav	2	22.5 °	0 °	Front Narrow Right
Title_16_0_03.wav	3	-67.5 °	0 °	Front Wide Left
Title_16_0_04.wav	4	67.5 °	0 °	Front Wide Right
Title_16_0_05.wav	5	-112.5 °	0 °	Side Left
Title_16_0_06.wav	6	112.5 °	0 °	Side Right
Title_16_0_07.wav	7	-167.5 °	0 °	Back Left
Title_16_0_08.wav	8	167.5 °	0 °	Back Right
Title_16_0_09.wav	9	-22.5 °	30 °	Upper Front Narrow Left
Title_16_0_10.wav	10	22.5 °	30 °	Upper Front Narrow Right
Title_16_0_11.wav	11	-67.5 °	30 °	Upper Front Wide Left
Title_16_0_12.wav	12	67.5 °	30 °	Upper Front Wide Right
Title_16_0_13.wav	13	-112.5 °	30 °	Upper Side Left
Title_16_0_14.wav	14	112.5 °	30 °	Upper Side Right
Title_16_0_15.wav	15	-167.5 °	30 °	Upper Back Left
Title_16_0_16.wav	16	167.5 °	30 °	Upper Back Right

Appendix F: Meanwhile, in another part of the forest (2015)

Programme Note

Development piece.

Performances
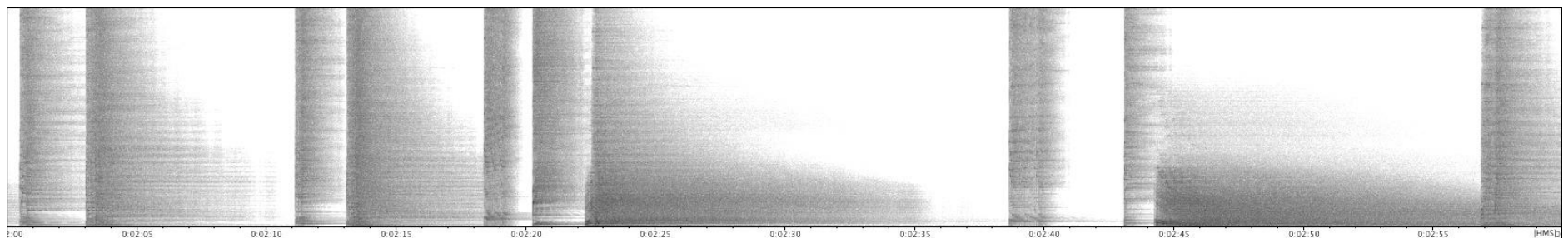
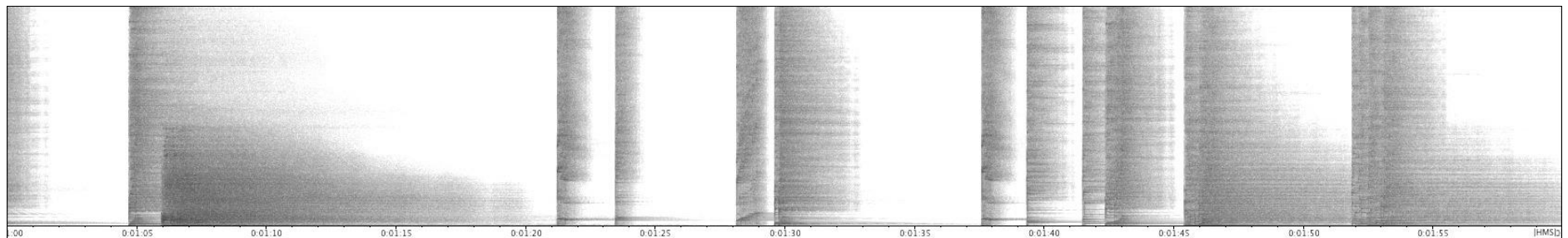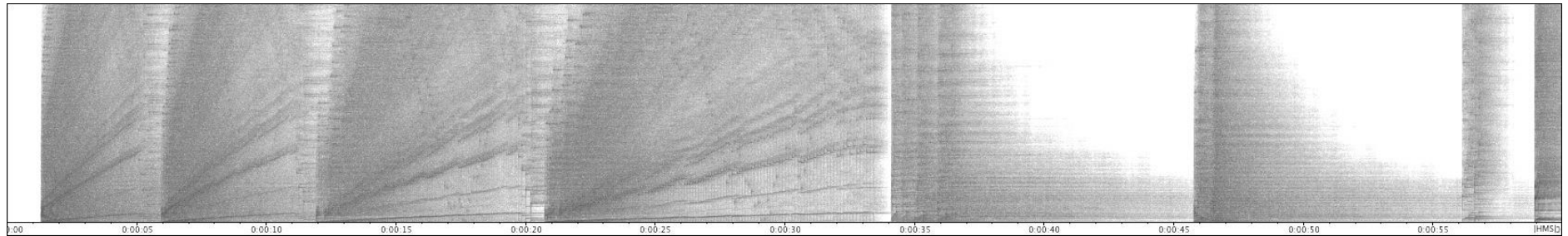This work has yet to be performed.

Format

16.0 surround.

Duration

4 minutes 31 seconds.

Meanwhile, in Another Part of the Forest
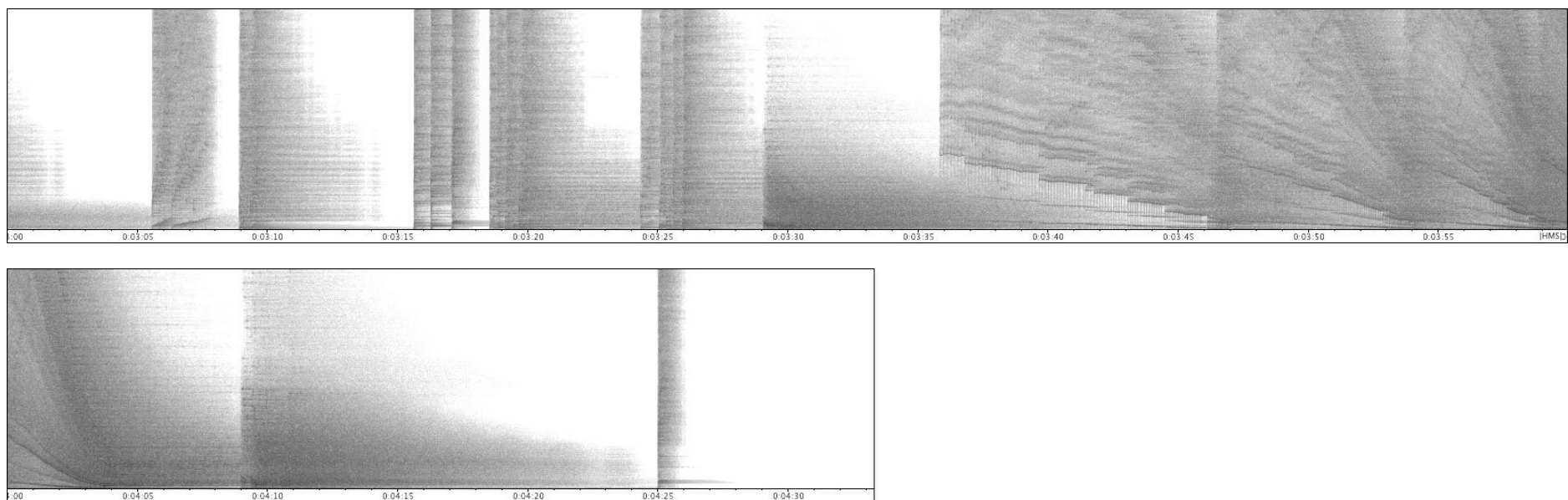
Richard Garrett

Figure 43: Meanwhile, in Another Part of the Forest (2015)

Bibliography

Barrett, Natasha, 'A Musical Journey towards Permanent High-Density Loudspeaker Arrays', *Computer Music Journal*, 40:4 (2016) 35-46.

Bates, Enda, 'The Composition & Performance of Spatial Music' PhD thesis 2009, <http://endabates.net/Enda%20Bates%20-%20The%20Composition%20and%20Performance%20of%20Spatial%20Music.pdf>, last accessed 20 April 2016.

BEASTMulch SuperCollider UGens <http://www.birmingham.ac.uk/facilities/ea-studios/research/mulch.aspx> last accessed 22 May 2016.

Cádiz, Rodrigo, *Computer Music Control by Fuzzy Logic: Theory and Applications*, (Saarbrücken: VDM Verlag Dr. Müller 2009)

Carey, Benjamin, '_derivations and the Performer-Developer: Co-Evolving Digital Artefacts and Human- Machine Performance Practices', PhD thesis 2016 <https://www.academia.edu/21746864/_derivations_and_the_Performer-Developer_Co-Evolving_Digital_Artefacts_and_Human-Machine_Performance_Practices>, last accessed 20 April 2016.

Collins, Nick, 'Towards Autonomous Agents for Live Computer Music: Real time Machine Listening and Interactive Music Systems', PhD thesis 2006 <http://composerprogrammer.com/thesis.html>, last accessed 20 April 2016.

Collins, Nick, 'Automatic Composition of Electroacoustic Art Music utilizing Machine Listening', *Computer Music Journal*, 36:3 (2012) 8-23.

Cope, David, *Techniques of the Contemporary Composer,* (New York: Schirmer 1997)

Dolby Atmos, <http://www.dolby.com/us/en/brands/dolby-atmos.html>, last accessed 29 September 2016

Elsea, Peter, 'Fuzzy Logic and Musical Decisions', 1995 <http://artsites.ucsc.edu/ems/Music/research/Fuzzy_Logic_And_Music.pdf >, last accessed 20 April 2016.

Elsea, Peter, 'A Fuzzy Logic Primer', 2007 <http://artsites.ucsc.edu/ems/Music/research/Fuzzy_primer07.pdf >, last accessed 20 April 2016.

Elsea, Peter, 'Musical Applications of Fuzzy Logic', 2011 <http://artsites.ucsc.edu/ems/Music/research/Music_App_Fuzzy.pdf >, last accessed 20 April 2016.

Eno, Brian, *A Year with Swollen Appendices*, (London: Faber 1996).

Fieldsteel, Eli, 'SuperCollider Tutorials' YouTube video. Posted by Eli Fieldsteel, 2013 – 2016. <https://www.youtube.com/playlist?list=PLPYzvS8A_rTaNDweXe6PX4CXSGq4iEWYC>, last accessed 27 May 2016.

Flanagan, David, *Javascript*, *The Definitive Guide*, (Sebastapol USA: O'Reilly 2011)

Fonseca, Nuno, '3D particle systems for audio applications.' *Proceedings of the 16th International Conference on Digital Audio Effects (DAFx-13)*, Maynooth, Ireland 2013, <http://dafx13.nuim.ie/papers/25.dafx2013_submission_50.pdf> last accessed 22 May 2016.

Gamma, Erich, Richard Helm, Ralph Johnson, John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software* (Boston: Addison-Wesley 1995)

Garrett, Richard, 'The Weathersongs Project' < http://www.sundaydance.co.uk/the-weathersongs-project/> last accessed 20 April 2016.

Garrett, Richard, 'nwdlbots' <http://www.sundaydance.co.uk/nwdlbots/>, last accessed 20 April 2016.

Garrett, Richard. 'Out of the Loop'. SoundCloud. < https://soundcloud.com/sundaydance/out-of-the-loop> last accessed 22 May 2016.

Garrett, Richard. 'Only Now'. SoundCloud <https://soundcloud.com/sundaydance/only-now-2013> last accessed 24 May 2016.

Garrett, Richard, "In Flight and Audio Spray Gun: Generative composition of large sound-groups." *eContact! 17.3 Toronto International Electroacoustic Symposium 2014: 8th edition of TIES,* 2015, <http://econtact.ca/17_3/garrett_audiospraygun.html> last accessed 20 April 2016.

Garrett, Richard, 'Audio Spray Gun 0.8 – the Generation of Large Sound-Groups and Their Use in Three-Dimensional Spatialisation', *Proceedings of the 41st International Computer Music Conference,* Denton 2015, pp. 352-355.

Garrett, Richard, 'Glitching The Riff, episode 3', podcast interview with Jordan Cutler. YouTube video. Posted by GlitchingtheRiff, 25 November 2015. <https://www.youtube.com/watch?v=cGA-Bf5ewEE>, last accessed 30 April 2016.

Garrett, Richard, forum contribution for *Computer Music Journal*, 40:4 (2016), 12-13.

Graham, Richard, forum contribution for *Computer Music Journal*, 40:4 (2016), 11-12.

Kim-Boyle, David, 'Sound Spatialization with Particle Systems.' *Proceedings of the 8th International Conference on Digital Audio Effects (DAFx-05)*, Madrid 2005, <http://jim.afim-asso.org/jim2005/download/19.%20Kim-Boyle.pdf > last accessed 6 May 2016.

Koan Pro, <http://en.wikipedia.org/wiki/Koan_%28program%29>>, last accessed 17 January 2013.

Kosko, Bart, *Fuzzy Thinking: The New Science of Fuzzy Logic,* (London: Harper Collins 1994).

Koutsomichailis, Marinos, *Mapping and Visualization with SuperCollider,* (Birmingham: Packt, 2013)

Lyon, Eric, 'The Future of Spatial Computer Music.' *Proceedings of the 11th Sound & Music Computing / 40th International Computer Music Conference,* Athens 2014, pp. 850-854.

Lyon, Eric, 'Eric Lyon - Music Composition for HDLA's (High Density Loudspeaker Arrays)**,** CCRMA Colloquium 18 May 2016. YouTube video. Posted by ccrmalite1, 18 July 2016. <https://www.youtube.com/watch?v=9xujQrLO0gk>, last accessed 1 September 2016

Malham, David G and Anthony Myatt, '3-D Sound Spatialization Using Ambisonic Techniques.' *Computer Music Journal* 19.4 (1995), 58-70.

Max, Cycling 74 <http://cycling74.com/> last accessed 22 May 2016

McCartney, James, 'Rethinking the computer music language: SuperCollider', *Computer Music Journal*, 26:4 (2002) 61–68

McLean, Alex and Geraint Wiggins, 'Bricolage Programming in the Creative Arts' *Proceedings of 22nd Psychology of Programming Interest Group 2010*, Madrid

Mubu, <http://forumnet.ircam.fr/product/mubu-en/> last accessed 22 May 2016.

Open Sound Control, <http://opensoundcontrol.org/introduction-osc> last accessed 29 September 2016.

ORPHEUS audio project, <http://orpheus-audio.eu/ > last accessed 29 September 2016

Peeters, Geoffroy,'A large set of audio features for sound description (similarity and classification) in the CUIDADO project', <http://recherche.ircam.fr/equipes/analyse-synthese/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf> last accessed 9 May 2016.

Peters, Nils, Trond Lossius, Jan Schacher, Pascal Baltazar, Charles Bascou and Timothy Place, 'A Stratified approach for sound spatialization.' *Proceedings of the 6th Sound and Music Computing Conference*, Porto, 2009.

Pulkki, Ville, 'Virtual sound source positioning using vector based amplitude panning.' *Journal of the Audio Engineering Society,* 45:6, (1997) 456-466

Pulkki, Ville, 'Uniform spreading of amplitude panned virtual sources.' *Proceedings of the 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics,* New York 1999, <http://lib.tkk.fi/Diss/2001/isbn9512255324/article3.pdf>, last accessed 9 May 2016.

REAPER, Cockos Incorporated <http://www.reaper.fm>, last accessed 13 March 2017.

Reich, Steve, 'Clapping Music for two performers', <http://earreader.nl/wp-content/uploads/2016/01/SteveReich-ClappingMusic.pdf>, last accessed 22 March 2017.

Reynolds, Craig, 'Flocks, Herds, and Schools: A Distributed Behavioral Model, in Computer Graphics', *SIGGRAPH '87 Conference Proceedings*, Anaheim, USA 1987 pp. 25-34.

Roads, Curtis, *Microsound*, (Cambridge USA: MIT Press 2001).

Roads, Curtis, *Composing Electronic Music: A New Aesthetic*, (New York USA: Oxford University Press 2015).

Russell, Bertrand, 'Vagueness', *The Australasian Journal of Psychology and Philosophy,* (June 1923), 84-92

Sauer, Arthur and Wouter Snoei, 'Working with WFSCollider (v.2.2)' <https://sourceforge.net/projects/wfscollider/files/WFSCollider%20Manual/> last accessed 4 September 2016.

Schacher, Jan, Chikashi Miyama and Trond Lossius, 'The SpatDIF library – Concepts and Practical Applications in Audio Software' *Proceedings of the 11th Sound & Music Computing / 40th International Computer Music Conference,* Athens 2014, pp. 861-868.

Schnell, Norbert, Axel Röbel, Diemo Schwarz, Geoffroy Peeters, Riccardo Borghesi, 'MuBu & Friends - Assembling Tools for Content Based Real-Time Interactive Audio Processing in Max/MSP', *Proceedings of the International Computer Music Conference (ICMC 2009),* Montreal, Canada,

<http://quod.lib.umich.edu/cgi/p/pod/dod-idx/mubu-and-friends-assembling-tools-for-content-based-real.pdf?c=icmc;idno=bbp2372.2009.096>, last accessed 17 May 2016.

Schumacher, Marlon and Jean Bresson, 'Spatial Sound Synthesis in Computer-Aided Composition.' *Organised Sound* 15:3 (2010) 271-289.

Schwarz, Diemo, Roland Cahen and Sam Britton. 'Principles and Applications of Interactive Corpus-Based Concatenative Synthesis'. Journées d'Informatique Musicale (JIM), Mar 2008, Albi, France. 2008. < https://hal.archives-ouvertes.fr/hal-01161401/document > last accessed 2 September 2016.

Smalley, Denis, 'Space-form and the acousmatic image' *Organised Sound* 12:1 (2007) 35-58.

Soundhack 0.896 <http://www.soundhack.com/freeware/> last accessed 9 September 2014

SuperCollider help files < http://doc.sccode.org> last accessed 16 May 2016

Taube, Heinrich, *Notes from the Metalevel: Introduction to Algorithmic Music Composition*, (London: Taylor and Francis 2004).

Thomas, Dylan, 'Fern Hill' in Edward Leeson (ed.), *The New Golden Treasury of English Verse* (London: Pan 1980) pp. 475-6

TouchOSC <http://hexler.net/software/touchosc> last accessed 6 May 2016.

Tsingos, Nicolas. 'Pre-Computing Geometry-Based Reverberation Effects For Games'. *AES 35th International Conference*, London 2009. <http://www-sop.inria.fr/reves/Nicolas.Tsingos/publis/ntsin_aes35_reverb.pdf> last accessed 2 September 2016.

WACM, <http://arts.ucsc.edu/programs/wacm/>, last accessed 17 January 2013.

Wilson, Scott, David Cottle and Nick Collins (editors), *The Supercollider Book*, (Cambridge USA: MIT Press 2011).

Wishart, Trevor, *On Sonic Art* (Abingdon: Routledge 1996).

Zadeh, Lofti, 'Fuzzy Sets' *Information and Control* 8:3 (1965) 338 -353.