You may print or download ONE copy of this document for the purpose of your own research or study.

The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site.

You are reminded of the following:

- Copyright owners are entitled to take legal action against persons who infringe their copyright.
- A reproduction of material that is protected by copyright may be a copyright infringement.
- A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.
Spectromorphology and Spatiomorphology: Wave Terrain Synthesis as a framework for controlling Timbre Spatialisation in the Frequency Domain

This thesis is presented for the degree of

Doctor of Philosophy

Stuart George James

Edith Cowan University
Faculty of Education and Arts
Western Australian Academy of Performing Arts
2015
Chapter 3: Mapping and Implementation

3.1 Introduction

In essence, Chapters 1 and 2 provide a conceptual, theoretical and performative framework for this research, and prepare for a subsequent evaluation of two mapping strategies, Model A and Model B, in the current chapter. Both of these mappings explore WTS as a means of controlling *timbre spatialisation* in the frequency domain. Previous discussions on spectral processing in Chapter 2 provide a context for the discussion of frequency domain processes explored in this chapter. Further, the contextualisation of spatial music practice in Chapter 1 provides a framework for the discussion of the panning algorithms and loudspeaker configurations used, in context, in this chapter.

Evaluation of a mapping strategy depends largely on a laptop performer’s ability to control the outcome of a system in a way that is meaningful and concise, yet also versatile enough for creative expression. Ideally, a control interface for *timbre spatialisation* should allow the laptop performer the ability to shape timbre and generate a variety of *sound shapes* and spatial gestures, and to control some relevant psychoacoustic parameters such as perceived source location, source width, spatial clarity and envelopment. An expressive control strategy should allow the laptop performer the freedom and scope to be able to explore outcomes that are predictable by navigating an intuitive visual software interface, and should also leave the opportunity open for exploring results that are more chaotic, unpredictable and esoteric. The research has not addressed the perceptual evaluation of these different implementations; rather two new possible mappings are presented with supporting theory and rationale. The discussion of expressive control, an issue that is dependent on an evolving approach to *sound shapes* is reliant to the morphology of data, and will be a focus of Chapter 4. However, the fundamental way in which WTS is used to spatialise spectra,
Insook Choi, Robin Bargar and Camille Goudeseune (1995) describe the process through which a performer learns to associate acoustic properties with the actions that produce them as feedback, allowing the performer to measure the consequences of a series of actions. They suggest that *intuition*, within this context of human–machine interaction, describes the experience of participating in feedback systems where the performer can draw compatible conclusions from a small number of trials, quickly learning to differentiate a large number of states in a complex system. The issue of mapping manifests in the choice of strategies for transference of physical input gestures to sonic results, which includes both the decision of what parameter associations to make as well as the behaviour of this transference itself (Van Nort & Wanderley, 2006a). There are many styles of digital music performance that do not have the communicative aspects of human gestures as their primary concern, yet real-time control and organisation of sound materials is often still of paramount importance (Van Nort & Wanderley, 2007).

Andy Hunt and Marcelo Wanderley (2002) define several mapping strategies that have developed over the years: one-to-one, one-to-many and many-to-one, and several of these approaches can be used in combination for a variety of many-to-many mappings. Doug Van Nort and Wanderley (2007) state that a mapping can be *explicit* or *implicit*. The former refers to a situation in which the mapping is known and can be expressed analytically, whereas the latter is based on internal adaptation of a system, and can be seen as a ‘black box’ model (Lee & Wessel, 1992). This latter approach is seen as promising in that it can allow a user to adapt a performance system to their unique gestures. However, the explicit approach is beneficial in that having knowledge
about the way that the mapping occurs allows one to tune, alter and expand it over time and for different musical contexts.

What follows in this chapter is a discussion on the generalised approach to mapping investigated, a discussion of previous control strategies for the spatialisation of spectral bands, a discussion on control resolution and FFT synchronicity, and the use of WTS as a control strategy for the intuitive control of large parameter sets. Finally, this chapter looks at two mapping strategies that explore WTS as a means of controlling

*timbre spatialisation: Model A and Model B.*

### 3.2 A Generalised Mapping Strategy

The biggest challenge for composers working in ... [the frequency] ... domain continues to be exploration of ways to meaningfully manipulate and interpret FFT data in a musical context. In a real-time interactive environment, expectation of some sort of correlation between a musician’s performance and the sounding output of a computer system is high, and successful strategies for manipulation and interpretation of FFT data via performer input are paramount. Two mapping problems are completely intertwined: mapping of performer data, and mapping of FFT data. (Lippe, 2007)

As Lippe elucidates here, fundamentally there are two mapping problems involved with FFT-based processes. In this research project, as outlined in Figures 38a and b, there is a three-stage mapping process that involves first the mapping of performer data to WTS and second, the mapping of WTS to the spatialisation algorithm responsible for calculating the respective weights of frequency bands for all speaker channels. Finally, this is mapped to the FFT process itself. This chapter largely ignores the first category, performer data, instead focusing on the way in which an audio signal created by WTS is mapped to the spatialisation process and the multichannel FFT process. The two mapping strategies evaluated in this chapter are illustrated in Figures 38a and b, showing several stages of mapping in both Models. Chapter 4 will discuss the first category in this generalised mapping, performer data, and how this influences
the kinds of sound shapes produced using Model A and B mappings. This chapter will discuss the other three stages: WTS, spatialisation, and frequency domain convolution.

There are two fundamental factors that need consideration in such a mapping, the first being the way in which the resulting audio signals generated by WTS are mapped to the spatialisation process; the second being the nature of the spatialisation used, which is what is ultimately responsible for deriving the SPF functions used for filtering the sound source over the specified number of loudspeakers. Simply put, the two different mappings explore alternatives for interpreting the audio signal generated by WTS: Model A interprets the contour as representing the azimuth for respective spectral bands; Model B interprets the contour as representative of frequency band.

It is important to note that all of the processes in Model A and B involve audio rate control signals, shown in Figures 38a and b with the abbreviation AR. The one
exception in both models is the haptic rate control signals (indicated with the abbreviation HR), which will be discussed in Chapter 4.

### 3.2.1 Previous Control Interfaces for Spatialising Spectra

Four control interfaces will be reviewed which span 21 years of development. Some of the first control interfaces for spatialising spectra were explored in 1994 by Settel and Lippe. They demonstrate the *multislider* object in *Max/FTS*, as can be seen in Figure 39a. At the time this was considered adequate as it allowed control over the relevant parameter space—that is, the FFT bin number and the associated left–right panning location.

Figure 39b illustrates Torchia and Lippe’s (2004) adaptation that permits controlling spectral panning for quadraphonic systems. Conclusions suggest that applying this technique “to more than two channels introduces a few obstacles, particularly with controllers, visualisation and the manipulation of large amounts of control data. Various interfaces are presented which address these issues” (Torchia & Lippe, 2004, p. 116).

*Figure 39a. The *multislider* object used for stereo diffusion of spectral material.*

*Figure 39b. A spectral spatialiser controller using separate *multislider* objects for left–right and front–back localisation. The *lcd* display indicates the location of FFT energy bins in two-dimensional space.*
Christopher Keyes (2004) implementation, illustrated in Figure 40, is designed for an octophonic system. The patch treats the loudspeakers in four groups of pairs: Channels 1&2, Channel 3&4, Channels 5&6 and Channels 7&8. This implementation is not a ‘connected’ system as whatever happens in, for example, Channel 1 will have no bearing on Channel 3. It also means that frequencies do not shift around the loudspeaker configuration according to panning rules, and may present the listener with the same frequency from multiple conflicting directions. Keyes describes the process that convolves the left audio input based on the equalisation curve indicated in dark blue on the graphic user interface (shown in Figure 40) whereas the right channel is convolved with the inverse of that curve. Keyes controlled the system either by drawing values in the multislider objects with a mouse, 2-dimensional MIDI (musical instrument digital interface) controller or via automation, parsing through the table bin by bin. Automation controls included stochastic ‘random walks’, sinusoidal contours and ramp contours (Keyes, 2004).

*Figure 40. Christopher Keyes’ (2004) GUI and control interface.*
David Kim-Boyle uses a different approach to control, where the spatial position of different FFT bins are controlled by generative and dynamical algorithms. Kim-Boyle has mapped not only Craig Reynolds’ Boids algorithm\(^{101}\) to the spatialisation of spectra (see Figure 41), but also simulations of clouds of smoke,\(^{102}\) and stochastic approaches for generating control data. The latter allows the control and the transformation of coordinates for several particles with just a few spatial controls. Kim-Boyle allows for the overall location of the bins within two-dimensional space to be controlled with the mouse or joystick, or alternatively assigned to circular trajectories. Kim-Boyle (2008) claims that the stochastic implementation demonstrates equivalent or better perceptual results than his implementation with Boids algorithm, with the advantage that it presents a more intuitive user interface. He also stresses the importance of developing an interface that not only is informed by perceptual considerations, but has the power to control and transform coordinates for hundreds of particles, which at the same time does not overwhelm the user with massive banks of control data.

Other methods of control used by these authors have included the use of sound sources to control spatialisation, creating a spatial cross-synthesis (Settel & Lippe, 1994; Torchia & Lippe, 2004), or envelope following allowing for random changes to occur in the processing parameters due to attack transients (Barreiro, 2010).

Kim-Boyle (2008) argues that while “frequency domain processing techniques have been used in many ways to transform the timbre qualities of a sound, their

\(^{101}\) Implemented in MaxMSP by Eric Singer (see www.ERicsinger.com). Kim-Boyle also incorporated spectral delay determined from the z-coordinate of a boid. When the squish and offset parameters are dynamically varied during performance in the x, y and z dimensions, interesting timbral effects can be achieved with certain spectral components of the sound lagging behind others or preceding them. With natural sound sources whose timbres depend on features such as attack transients, the transformations can be particularly arresting.

application for spectral spatialisation has been comparably limited” (p. 1). Torchia and Lippe (2003) express the same concerns in that control of the spatialisation of spectral bands has remained a problem due to the large numbers of parameters involved in such a process.

![Figure 41. Kim-Boyles (2008) implementation where values of the Boids algorithm are mapped to frequency bins.](image)

Perhaps this could be said for performative multichannel diffusion practice too. Normandeau (2009) observed that when working with the Zirkonium software used for designing spatial trajectories, “the only way to design trajectories in ... [software] ... is to write every movement line by line, which is not adequate for complex movements” (p. 284). Keyes (2004) describes a similar problem stating that the all-too-common problem is the “lack of an interface or controller to deal with this effectively. As is often the case, those responsible for the diffusion were sitting behind a mixer with all 10 fingers posed anxiously over faders, straining to maintain even panning curves as they worked in real-time” (p. 3):

From a technological perspective, the limitations of physical control interfaces for multichannel spatialisation have not been completely overcome at this point
in time and the studio mixing console in particular, as a relatively crude interface between composer and multichannel diffusion system, has become a serious barrier to the creative ambitions of live sound spatialisation. Certain types of spatiomorphology remain inaccessible to the composer without the use of customised software. (Stefani & Lauke, 2010, p. 253)

Approaches to timbre spatialisation in the frequency domain also require the control of large parameter sets. Therefore such approaches require the right balance of intentional control over the details of spatial gesture and sound shape, without overwhelming the laptop performer with hundreds or thousands of control parameters.

3.2.2 WTS as a Control Framework

The extensive and, shall we say, extendable nature of synthesis allows for ongoing chains of generators, controllers and processes; Wave Terrain Synthesis might be used as a controller for another methodology such as convolution in the frequency domain. (James, 2005, p. 149)

Table lookup procedures for sound synthesis were developed not long after the birth of the microcomputer in the mid-1970s, and it was in 1978 that Rich Gold first coined the term wave terrain synthesis to mean a table lookup procedure involving 2-dimensional arrays. There have been many advantages to the table lookup procedure. It is considered to be computationally efficient as processing requirements are consistently low, and the procedure allows for storing and reloading tables of data. These data sets might consist of arithmetically generated values, statistical information or measurement data, provided the data range is adequately sufficient. ‘Samples’ of real-world data may be used, or audio, video and other data formats. Extending the dimensionality of the table lookup procedure allows for describing the complex non-linear behaviour of electronic components and other complex applications such as the localised perception of sound in a binaural setting using HRTFs.

The idea of traversing data sets is well established in the realms of sound synthesis and computer music. Wavetable lookup traditionally refers to the repeated
scanning of a list of numbers that describe a single cycle of an audio waveform in memory (Roads, 1996); this was originally used for the generation of digital oscillators. Increases in memory capacity have since allowed for the use of larger wavetables, which has led to a flourish in other techniques in sound synthesis including audio sampling, waveshaping synthesis (Le Brun, 1979), audio analysis via FFT or wavelet methodologies, and the subsequent resynthesis of these data, graphical synthesis and convolution reverb. The extending of these wavetables by two or more dimensions has also seen other develops in sound synthesis including WTS (Gold, 1979), graphical synthesis and multi-impulse response\(^\text{103}\) techniques.

Some of the earliest investigations into WTS focused largely on mathematical functions, and later investigations involved the use of other sources of data. WTS is an open and extendable technique capable of reproducing parameter spaces that describe a variety of existing sound synthesis techniques as shown in Figures 42a and 2b (James, 2005). WTS could be used not only as a sound synthesis technique in its own right, but could also be employed as a means of controlling other processes (James, 2005). For example, techniques that typically require a large number of control parameters such as granular synthesis (Roads, 1996) and spectral synthesis (Wishart, 1994) may benefit from this application.

WTS methodology requires a trajectory signal to read values from a terrain surface, most usually at audio rate, in order to generate a resulting output signal. Research has seen a variety of curves that exhibit different kinds of behaviours from periodic and quasi-periodic, to more aperiodic and randomised structures. Many of the sounds produced have been characterised as being drone-like, pulsing and harmonically rich (Mikelson, 2000). Depending on the terrain and trajectory used, unpitched sounds have been suggestive of glitch and noise-based sample loops, as well as textures much

\(^{103}\) See the MIR-Project, Retrieved 10\(^{\text{th}}\) Jan 2015 from http://www.vsl.co.at/en/65/73/500/320.vsl.
like rain, cracking rocks, thunder, electrical intermittent noises and insects (Di Scipio, 2002; James, 2005).

![Figure 42a](image1.png)  
**Figure 42a.** A terrain surface describing a ring modulation synthesis parameter space. Source: James, 2005.

![Figure 42b](image2.png)  
**Figure 42b.** A terrain surface describing a frequency modulation synthesis parameter space. Source: James, 2005.

Just as one can spatially distort a terrain structure, geometric transformations can be applied to the trajectory resulting in further modulation possibilities. Some of these transformations include phase distortion or pulse-width modulation and affine transformation, which can involve the translation, scaling and rotation of a trajectory.

A number of researchers have explored the navigation of a parametric ‘surface’ or hyper-plane that represents all traversable regions of a parameter space. These are derived through the interpolation, extrapolation and regression of control/sound data (Van Nort & Wanderley, 2007). Several mapping toolkits, LoM (Library of Maps) (Van Nort & Wanderley, 2006) and MnM (Mapping is not Music) (Bevilaqua et al., 2005),¹⁰⁴ have also been developed to facilitate higher dimensional navigation across multi-parametric sets, and provide access to a toolbox of few-to-many mappings. Some other implementations using interpolation and regression techniques include the manifold

---

¹⁰⁴ Built on top of the FTM (Faster than Music) library distributed under the LGPL (lesser General Public License), Retrieved 10th Jan 2015 from http://ftm.ircam.fr/index.php/Main_Page
interface (Choi, Bargar & Goudeseune, 1995), Ali Momeni and David Wessel’s (2003) models and the MetaSurface (Bencina, 2005). WTS presents another case for navigating a multidimensional parameter space. The use of WTS as a control mechanism is a governing system that allows the performer to create a complex and coordinated change across an existing complex parametric system. Although the terrain surface and trajectory represent manifolds, the way in which these structures combine presents a wide range of morphologies—a parametric space that is itself malleable. The scientific concept of phase–space\(^{105}\) is an important notion here as it represents all parametric possibilities of a system (Choi et al., 1995). My previous research in WTS evaluated the flexibility of the system for synthesising and transforming a wide range of curves (James, 2005). Although WTS may involve some interpolation as part of the lookup process, it is a different process entirely to the mapping strategies used by Van Nort et al. (2004; 2006; 2007), Choi et al. (1995), Bencina (2005) and Momeni et al. (2003).

WTS may be used as a means of controlling *timbre spatialisation*, rendering such a complex processing system more manageable in live performance (James & Hope, 2012). In 2005, I described WTS as an open and adaptable framework for both sound synthesis and parameter control suggesting WTS could be mapped to parameters of a spatial process (James, 2005). In 2011, I devised an early implementation of WTS as a control mechanism for the diffusion of spectra over 1024 frequency bands, as seen in Figure 43 (James & Hope, 2011).

\(^{105}\) All the permissible combinations of parameter values of an algorithm.
In 2012, Ioannis Tsirikoglou published his master thesis on the use of multidimensional data sets for sculpting the spatial parameters of sound diffusion where applications of WTS as a mechanism for controlling the diffusion of spectra were discussed. Tsirikoglou focuses more on the philosophy and phenomenology of parameter space, without providing many examples of applications for controlling the spatialisation of spectral bands. Tsirikoglou describes some of the results:

We can also use any sound signal and modulate it into space. Nice results are achieved when we are speeding up from very low rates to higher rates. The sounds are moving slowly according to the trajectories’ coordinates, then they start moving faster as well as audible artefacts are appearing like LFO [low-frequency oscillation] modulations traveling in space and finally, as we increase further the rate of the trajectory functions, timbre transformations are occurring with unique spatial properties. Geometric and other transformations of the trajectories, like rotations and translations, act on the resulting modulations and timbres while the energy of sound is transferred to the position in space that the trajectory occupies. (Tsirikoglou, 2012, p. 58)

WTS as a control framework differs from other systems in a number of ways. One is that the system is mutually dependent on two independently generated structures – one haptic rate and the other audio rate. Another is that it is not necessarily autonomous, and can be easily controlled gesturally. The rationale for using WTS was

---

106 Although Ableton Live does not support by default audio tracks above two channels, and the Max4Live objects plugin~ and plugout~ only communicate with Ableton’s standard audio track architecture, there is a workaround with Max4Live live.remote~ object for sending an audio return track from a multi-channel process back to the Ableton Live mixer.
that it might be possible to decode spectral weighting curves from a multidimensional lookup table, and that certain geometrical distortions applied may in fact correlate with precise spatial movements or transformations, such as the scene rotation described by Wishart (1996) as illustrated in Figure 44.

![Figure 44](image)

*Figure 44. The rotation of a scene as described pictorially in Wishart’s On Sonic Art (1996).*

The current research engaged a different paradigm for describing spatial movement using the topographical structures of terrain maps, and the evolutionary temporal qualities of trajectory movement. The idea was originally spawned when reading Normandeau’s discussion of a vertical palindrome:

> The form is identical and symmetrical in both directions, and it is made from two mixes of the same 96 tracks whose ‘weight’, instead of the exact inversion, is the same from the beginning to the end and vice versa. The only difference between the two mixes, introduced for musical reasons, is that the first mix is a de-crescendo that begins with a tutti which is gradually filtered to the high-frequency register, while the second mix is a crescendo that begins with the low-frequency register gradually increased to a tutti. One can consider this as a vertical palindrome! (Normandeau, 2009, p. 280)

Normandeau’s description immediately spawned an image in my mind of a two-dimensional plane as a score or contour map for determining frequency weights distributed across a 2D horizontal speaker configuration. By tilting this flat surface about the centre, like a seesaw, frequencies could be biased at either extremity of the listener space. Rather than conceiving of a huge set of arbitrary points, this approach is governed by one connected system that choreographs the weights of frequency bins across any arbitrary speaker configuration.
3.2.3 Synchronisation v. Control Signal Protocol

Since open sound control (OSC) was developed, there appears to be an emerging trend in the development of multi-parametric generators for control. This control protocol claims to be superior to older systems like MIDI in terms of speed, bandwidth and resolution, as illustrated in Table 6.

According to an online resource, OSC implementations exist generally in three categories: hardware devices, software implementations and programming language libraries. There is a proliferation of OSC-based software that generates data with large quantities of parameters. These include Iannix, Geosonix, ISO-Flock, FaceOSC, Zirkonium, and the Blender 3D rendering engine. Some OSC applications are designed to work with specific hardware devices such as Synapse and OS Celeton for the Kinect interface. OSC has also seen the proliferation of physical interfaces such as the Jazzmutant Lemur, Apple iPad and iPhone, Monome and Smart controller, to name only a few. Perhaps the software implementations most suited to spatialisation are Iannix, Zirkonium and COSM.

107 Other research in the field has suggested that claims such as the speed of open sound control (OSC) being superior to MIDI is a myth (see Fraietta, 2008).
112 Retrieved 10th Jan 2015 from https://github.com/kylemcdonald/ofxFaceTracker/downloads
114 Retrieved 10th Jan 2015 from https://www.blender.org
116 Retrieved 10th Jan 2015 from https://github.com/Sensebloom/OSCeleton
120 Retrieved 10th Jan 2015 from http://opensoundcontrol.org/implementation/monome
121 Retrieved 10th Jan 2015 from http://opensoundcontrol.org/implementation/smart-controller
122 Developed by Wesley Smith and Graham Wakefield in the AlloSphere Research Group (Retrieved 10th Jan 2015 from http://www.allosphere.ucsb.edu/cosm/).
Table 6

*Speed and Bandwidth of Various Communication Protocols for Music Applications*

<table>
<thead>
<tr>
<th>Protocol</th>
<th>MIDI</th>
<th>OSC</th>
<th>Audio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fastest speed</td>
<td>2400Hz in frames per second, or approximately 1000Hz in <em>MaxMSP</em> scheduler</td>
<td>Dependent on network</td>
<td>Sample rate dependent (i.e. 44100Hz)</td>
</tr>
<tr>
<td>Resolution</td>
<td>Serial</td>
<td>Parallel (10-bit and 14-bit)</td>
<td>Serial</td>
</tr>
<tr>
<td>Time-synchronised</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>31.25 kilobits per second</td>
<td>10 megabits per second</td>
<td>640 kilobits per second</td>
</tr>
</tbody>
</table>

The closest implementation to WTS using the OSC protocol is Geosonix because it generates streams of data based on time domain trajectories that pass over an image. The trajectory is used to 'lookup' the pixel values from the image at a specific point in time. This generates a contour that correlates with the topography of the image pictured. As James (2003) discusses, the combinations of different possible streams of control data explodes exponentially when dealing with a multidimensional system as opposed to a one-dimensional system. In other words the phase–space of possible curves is largely infinite and limited only by the library of curves that are stored.

Controlling *timbre spatialisation* requires a very particular kind of control signal. Although audio signals are always guaranteed to pass a certain number of samples per second, *MaxMSP* messages (control rate) and video in *Jitter* cannot always guarantee exact timing synchronicity. OSC suffers from the same kind of clocking issues, as the timing of events is not tightly synchronised, and using certain network protocols such as UDP (user datagram protocol) may not be guaranteed to arrive at all. Due to the precise timing of audio rate, FFT processes—that is, synchronicity to FFT frames—must be performed at audio rates. This is necessary in order to create tightly
synchronised and smooth spectromorphological and spatiomorphological transitions from frame to frame. Audio signals as control signals have the benefit of being time synchronised and can transmit high-precision numbers of 16-, 24- or 32-bit floating point resolution.

*Lua*~ is a *MaxMSP* external, embedding an extension to the Lua programming language for computer music composition (the Vessel library). It supports sample-accurate interleaving of synthesis and functional control, as well as low-dimensional control rates. This is helpful in circumventing some of the limitations of the *MaxMSP* environment for such work by supporting highly dynamic signal processing graphs in parallel processes according to timing specifications below block rate. This is similar to the time-synchronised features found in software such as the *kr* control rate *v. ar* audio rate *Csound*, and the lower clocked rate for control signals found in *Reaktor*.

---

**Figure 45.** A block diagram showing how WTS is synchronised to the FFT frame.

---

Temporal synchronicity between WTS and *timbre spatialisation* requires that both systems be determined by the same clocking structure. As the FFT frame rate is dictated by the FFT window size, I used this as a control signal for driving WTS. In this way the index numbers of each FFT frequency bin form a ramp function, which can be used to drive the audio rate trajectory signal used for WTS, much in the same way as a phase-driven oscillator. The consequence of this is that the trajectory signal is completely synchronised with each FFT frame; therefore any transformation applied to this signal will correlate consistently with the same respective spectral bands. As we are 'windowing’ this control signal within each FFT frame, we can align each sample with a corresponding frequency bin, assigning each spectral band spatial properties such as its azimuth. In Section 3.3 these properties are extended to account for further spatial attributes such as distance, elevation and spatial width, etc. These are dependent on the spatialisation algorithms used. Figure 45 illustrates this synchronisation process in a block diagram.

### 3.2.4 Real-time Control of Large Parameter Sets

Traditionally, processes of sound diffusion approach spatialisation using an explicit one-to-one mapping via a circular panner interface. Interaction with a point on this interface determines the perceived localised position of the diffused sound within a sound scene or listening space. Most implementations use a compact mono or stereo sound source, and diffuse this across a multichannel speaker system. Besides some of the commercial DAW solutions offering an included surround panner, such as Apple *Logic Pro* (as pictured in Figure 46a) and Avid *ProTools*, there are several more-specialised multichannel diffuser plugins for DAW systems including Flux IRCAM...
Spat\textsuperscript{126} (as pictured in Figure 46b), Longcat Audiostage\textsuperscript{127}, Waves S360\textsuperscript{128} surround panner and imager and ViMiC\textsuperscript{129} audiounit plugin by Nils Peters. All of these diffusion tools allow the user to move the source across a circular interface representing the perceived position of the sound within the listening area. Most of these tools additionally allow the user to adjust an assortment of other parameters such as the width of the sound source, source presence, envelopment, air absorption and Doppler shift in addition to the relative distance and azimuth cues.

![Figure 46a. A standard surround panner window used for ITU 5.1 standard.](image)

![Figure 46b. A more sophisticated panner interface used in Flux IRCAM Spat, which is more adaptable to various loudspeaker configurations.](image)

Panning techniques available in DAW-based solutions are generally\textsuperscript{130} limited to sounds in a 2-dimensional plane and are not flexible in accommodating loudspeaker configurations beyond consumer formats such as stereo, quadraphonic, 5.1, 7.1, 10.2 and so on (Peters et al., 2009). Third-party plugins for commercial DAW software also lack the associated range of available panning algorithms, making techniques such as

---

\textsuperscript{126} Retrieved 10\textsuperscript{th} Jan 2015 from http://www.fluxhome.com/products/plug_ins/ircam_spat
\textsuperscript{127} Retrieved 5\textsuperscript{th} Jun 2012 from http://www.longcat.fr/
\textsuperscript{128} Retrieved 10\textsuperscript{th} Jan 2015 from http://www.waves.com/plugins/s360-surround-imager-panner
\textsuperscript{129} Retrieved 10\textsuperscript{th} Jan 2015 from http://redmine.jamoma.org/projects/vimic
\textsuperscript{130} One exception is the DAW Reaper. The ReaSurround tool allows for 3D spatialisation. Retrieved 6\textsuperscript{th} Jun 2015 from http://www.reaper.fm
WFS and *swarm-based spatialisation* inaccessible in commercial DAWs (McGee, 2010). *Timbre spatialisation*, as approached in the frequency domain, is another technique that is inaccessible with these tools.

The most common approach in a DAW host is to have a separate panner on every audio track, meaning it is not possible to work with the movement of several trajectories within the one panner interface. However, there are many instances in which it may be beneficial to visualise the movement of multiple sources within the one interface such as *ReaSurround* inside *Reaper*. Software environments like *MaxMSP* and *Pure Data* have a more flexible architecture for routing audio. Such environments feature some more versatile libraries and frameworks for spatialisation, which allow the user to choose the spatialisation method, and customise the number of both input sources and speaker outputs, as well as the orientation of the loudspeakers used. Such tools also allow the user to visualise the positions of sound sources all in the one view window. Some of these tools include *ambipanning*~ and *ambimonitor* by Jan Schacher at Institut for Computer Music and Sound Technology (ICST) in Zurich (see Figure 47a), Jamoma *mod.sur.dbap*~ (see Figure 47b) and *mod.sur.vbap*~, *hoa.decode*~ made available by Centre de recherche Informatique et Création Musicale (CICM) as part of their HOA library and *Orbit 2D* by Peter Batchelor. Other tools without a graphical user interface include *Spatium* by Andrew Horsburgh and *ambi-decode*~ by Graham Wakefield.

131 Flux IRCAM *Spat* does allow up to eight inputs via auxiliary inputs, and additionally provides flexible routing for up to eight audio outputs. *Longcat Audiostage* is a standalone application that allows for multiple instances of audio input sources via a bridging plugin creating a client–server interaction with an existing host digital audio workstation (DAW).


133 Retrieved 10th Jan 2015 from http://www.jamoma.org


137 Retrieved 10th Jan 2015 from http://www.grahamwakefield.net/soft/ambi~/
A number of these tools allow the design of customised speaker configuration by sending a series of simple messages to the relevant object or module. Applications such as ReSound further extend this functionality with a framework for spatialisation practice that is particularly well suited to real-time interaction and improvisation (Mooney & Moore, 2008). The ability to specify and control mappings in groups, proportionally, additively, subtractively and semi-automatically via behaviours allows laptop performers to realise dynamic, flexible and easy-to-control spatialisation routines that would be difficult or impossible to achieve in real time with analogue mixing hardware, particularly with multichannel input sources.

Figure 47a. An Ambisonic panner interface by the Institut for Computer Music and Sound Technology for MaxMSP that illustrates three input sources panned with different azimuth, elevation and distance cues.

Figure 47b. A distance-based amplitude panning interface in MaxMSP as part of the Jamoma library. This tool allows adaptable configuration for the number of inputs, outputs and the speaker configuration used.

Although panner interfaces tend to be suited for live performance, this does not account for situations where the diffusionist is responsible for panning higher numbers of point sources that would otherwise be infeasible in real-time performance. There is a
range of software for sound diffusion aimed at ‘composing’ complex multi-layered spatialisation. These kinds of tools tend to fall into two categories: those that require the composer to create, edit and transform sound trajectories with respect to time, and those concerned with the semantics of notating spatial gesture in musical scores. These include standalone applications such as HoloEdit,\(^{138}\) (see Figure 48a) ReSound,\(^ {139}\) Zirkonium, BeastMULCH,\(^ {140}\) Sound Element Spatialiser (McGee, 2010) and OpenMusic Prisma (see Figure 48b).\(^ {141}\) HoloEdit features different ways to create, transform and manipulate sound trajectories: the Room Editor offers a top view of the space and the trajectories, so that it is possible to handle trajectories and loudspeakers from a spatial point of view, and the Time Editor permits the precise temporal manipulation of trajectories.

---

139 Retrieved 10\(^{th}\) Jan 2015 from http://resound.sourceforge.net/about.php
The *OMChroma* and *OMPrisma* software present a modular environment in which spatial sound synthesis processes can be carried out in an ergonomic and efficient way, independently of the number of sound sources, audio channels or the chosen sound synthesis/spatialisation technique. *OMPrisma* provides composers with logistical simplicity for real-time applications. A multichannel playback (decoder) application is provided that is tightly integrated with a compositional environment and can be used either in compositional contexts (using OSC) with *OpenMusic*, or for on-site reproduction (Schumacher & Bresson, 2010). In this environment each synthesis component (e.g. sinusoid, formantic waveform and grain) can be associated to a set of spatialisation parameters and can be individually controlled.

*Timbre spatialisation* in the frequency domain is a technique that requires the control of potentially thousands of separate spectral bins, and for larger speaker configurations the number of relevant parameters increases. The logistical problem of applying dynamic pan curves to thousands of independent spectral bins is not only a management issue, but a cognitive and real-time temporal issue. This is further complicated for panning techniques that incorporate not only direction cues, but also distance, elevation and spatial width for separate spectral bands. Even in more traditional point source spatialisation techniques, there are often problems in controlling multichannel sources. Harrison states:

> If you’ve got an eight-channel source, and every channel of the eight has a fader, how do you do crossfades? You haven’t got enough hands! (Mooney (Ed.), 2005, Appendix 2)

The prospect of taking a sound source that (itself) generates as many sources as spectral bins becomes a logistical problem. Torchia and Lippe (2003) and Kim-Boyle (2006, 2008) have all expressed the need for researching other mapping strategies for spatialising spectra. Most existing methods utilise control rate approaches, most notably
Kim-Boyle’s use of the Boids algorithm. However, what is being sacrificed for the sake of logistical feasibility? Semi-autonomous systems do not always allow for the precise level of control desired by the performer. Nor does the laptop performer usually have the time to compose a large number of break-point function curves:

Several such [explicit mapping] strategies have been implemented for computer-based musical control, and yet there are not many readily accessible tools for constructing mappings between large parameter sets. (Van Nort and Wanderley, 2006, p. 1)

Unlike the other existing applications for designing trajectories for spatial audio such as Resound, HoloEdit, Zirkonium and OpenMusic (using OMPrisma), there is strong impetus for finding real-time solutions for large parameter sets. Sound spatialisation, particularly in real-time settings, is not only a process we apply ‘in the moment’, but is also a process that requires precision and intentionality with respect to the kinds of spatial gesture one applies to a given sound source. This requires a control methodology that is both intuitive and where audible movement of sound shapes across the sound scene results in a clear relationship between the actions performed in software and the resulting diffusion. A laptop performer cannot be intentionally responsible for the diffusion of every independent frequency bin, but rather is responsible for the global distribution of all frequency bins. An analogy in audio engineering is the notion of a group fader, where the performer or engineer is not responsible for riding the level of multiple individual faders on the mixer, but rather a single fader that controls the proportional level of multiple audio channels. In this way, parameter management reduces the necessary burden to one parameter change, making the system easier to manage and simple enough cognitively for the performer or engineer to focus on the sounds themselves, and the musicality of the spatial gestures applied.

Approaching the control of thousands of simultaneous parameter using WTS works out to be ergonomically efficient, and allows for a certain degree of
‘connectiveness’ across parameter sets in a way that they are grouped and interact, according to the system used to govern them.\(^\text{142}\) It enables a viscosity and fluidity that is evident in the \textit{sound shapes} generated, and how they evolve. This sense of interaction between speaker channels is important if the results are to generate a morphology of \textit{sound shapes}—for example, a scene rotation would require the parameters of one speaker to be transferable from speaker to speaker, and so forth. More complex \textit{sound shapes} involve some considerably more non-linear transitions.

The smooth morphology of sound shapes relies on WTS being synchronised with the FFT process. Section 3.1.3 described this process of synchronisation, but what does this mean for both frequency domain and time domain processes in \textit{MaxMSP}?

Time domain signals in \textit{MaxMSP} consist of a single stream of values. Frequency domain signals, on the other hand, are computed as a vector of samples in \textit{MaxMSP}. What this means is that for an FFT frame size of 2048 samples, one audio signal will consist of a sequence of values that count sequentially through 1024\(^\text{143}\) successive spectral bins, and then the sequence starts again (as illustrated in Figure 49).

Synchronising the WTS process to the FFT frame influences the way in which these signals are derived and interpreted such that WTS is interpreted as a frequency domain process when it is synchronised, and a time domain process when it is asynchronous (see Figure 50).

\(^{142}\) In order to move forward into the research questions proposed in Chapter 1, it is necessary to present my own research findings, as published previously, and drawn in the proof of concept and early implementation stage. This statement was an observation I made as a performer during the iterative creative process. Whilst this kind of observation cannot have been known to me before I began the entire research project, it emerged fairly early in the research phase, and ultimately allowed me to question many more useful questions in terms of how WTS may more specifically be applied musically and performatively.

\(^{143}\) This is half the fast Fourier transform (FFT) frame size because of complex conjugate symmetry of the FFT.
In the time domain, WTS of itself can be understood as being an explicit many-to-one mapping, as the process requires at the input stage both a haptic rate terrain signal and an audio rate trajectory signal— a multidimensional index often consisting of two or three audio signals describing trajectory coordinates. The output stage of WTS consists simply of a single one-dimensional audio rate signal. These values are retrieved from the terrain by the trajectory index specified.

However, when we consider that WTS is synchronised to the FFT frame, the resulting audio signal created is in fact a vector of samples. This makes it very clear that the mapping of WTS to the process of timbre spatialisation is an explicit many-to-many mapping, or more specifically in the case of this research an explicit 1024-to-1024 parameter mapping. For asynchronous approaches the same principle applies, except that the signal derived by WTS is largely driven by time domain processes (as illustrated in Figure 50b). This approach is not dissimilar to what Lippe and Settel (1999) describe as low-dimensional audio rate control of FFT processing.
This vector of 1024 values is then mapped to a spatialisation function responsible for decoding the various frequency bands into a series of SPF functions used for convolving a live input source to a series of loudspeakers. This is again an explicit many-to-many mapping that depends largely on the computations involved in the spatialisation panning algorithm used. For a four-channel speaker configuration the mapping is specifically an explicit 1024-to-4096 mapping, whereas for 32 speakers it would be an explicit 1024-to-32768 mapping.

Figure 50a. In an FFT synchronous approach to WTS, the process of lookup will derive vectors of values according to the size of the FFT frame. This means WTS is computed as a frequency domain signal.

Figure 50b. In an FFT asynchronous approach to WTS, the process of lookup will derive values that do not align with the FFT frame. This means WTS is computed as a time domain signal that is arbitrarily mapped to a frequency domain process.

At the final stage, this mapping involves the process of resynthesising the SPF functions. This results in a conversion of data from the frequency domain back to the

144 The spectral process of convolution involves a multiplication process performed in the complex number domain or the frequency domain. In this way, one signal becomes a time-varying filter for another signal.
time domain involving the inverse FFT. This is essentially an explicit many-to-one mapping. A four-channel speaker configuration demonstrates an explicit 4096-to-4 mapping, and a 32-channel speaker configuration demonstrates an explicit 32768-to-32 mapping.

In conclusion, this first section covers a general mapping strategy for using the framework of WTS to control *timbre spatialisation*. The section also involved a review of previous control strategies for the spatialisation of spectral bins, and an evaluation of other software solutions used for controlling the spatialisation of sound within the broader framework of spatial music practice. This has elucidated the need for new interfaces for controlling spatialisation, particularly when dealing with complex processes such as *timbre spatialisation*. WTS has been discussed as an effective strategy for controlling such a system as it can be synchronised to the FFT frame. Finally this section discussed the explicit mappings throughout the suggested general mapping strategy proposed. What follows in Sections 3.2 and 3.3 is a more detailed discussion on two mapping strategies explored as part of this research project: *Model A* and *B*.

### 3.3 Model A: Mapping Terrain Height to Azimuth

This mapping strategy was the first model implemented and is the simpler of the two mapping strategies investigated. This model was implemented within the first few months of the research project as it was seen that, even at this early stage of research, it was necessary to evaluate whether WTS would provide a successful framework for controlling *timbre spatialisation* in the frequency domain.

*Model A* directly maps the resulting audio signal generated by WTS to the azimuth parameter of a *spectral spatialisation* patch similar to that developed by Barriero (2010). The contour \( z \) generated by WTS is used to determine the azimuth of different spectral bands (Figure 51), and basic amplitude panning is used to calculate
the relative weights of spectra across a quadraphonic speaker configuration. The four SPFs derived through this process are then used to filter the sound source, resulting in a *spatiomorphology* of this sound.

The most significant advantage of this model is that it is more computationally efficient than *Model B*. For this reason, it is effective in creating smooth and responsive\(^\text{145}\) changes in *sound shapes*. As is shown in Figure 52, this mapping is generally only effective for exploring *spatiomorphology* over equidistant circular speaker configurations, but there are a few exceptions where *spectromorphology* is also evident, as shown in Section 4.2.3.1 and Appendix D.

\(^{145}\) In terms of latency.
It is important to note here that spectromorphology is referred to with regard to the reproduction of sound, rather than the spectromorphology that results from the comb filtering of the pinnae and other non-linearities of the human auditory system. The results from this mapping are spectral distributions that range from highly localised, to wide and circumspectral sound scenes as illustrated in Figure 53.

3.3.1 Basic Amplitude Panning

Basic amplitude panning is a spatialisation technique that renders spatial movement using azimuth cues, but does not account for distance or elevation cues. In this model, the technique is implemented using a window function (shown in Figure
that is stored in memory and serves as a lookup table. By changing the width of the
window function used, this model can control the spatial width of rendered spatial
movements (to be discussed in Section 3.2.3). There are a number of different kinds of
curves that could be used here, but the model tested generally used the following
piecewise function for quadraphonic speaker configurations:

\[
A = \begin{cases} 
0, & \text{for } \theta < 0.25 \\
4\theta - 1, & \text{for } 0.25 \leq \theta < 0.5 \\
2 - 4\theta, & \text{for } 0.5 \leq \theta < 0.75 \\
0, & \text{for } \theta \geq 0.75 
\end{cases} \quad f(x) = \begin{cases} 
0, & x < 0.25 \\
\sqrt{4x - 1}, & x \geq 0.25 \text{ and } x < 0.5 \\
\sqrt{2x - 1}, & x \geq 0.5 \text{ and } x < 0.75 \\
0, & x \geq 0.75 
\end{cases} \quad (9)
\]

where \( A \) is the amplitude ratio and \( \theta \) is the phase of the lookup pointer. Phase, when
applied to the lookup process is generally considered a floating point value of 0–1. By
shifting the phase of this lookup table we can specify the necessary weight applied to a
point of azimuth for a given speaker. In order to apply this for a number of speakers we
could either have four separate lookup tables, or use the more efficient method of phase-
shifting the lookup table for each respective speaker. Figure 54b shows four curves that
are colour coded according to speaker: Speaker 1, Blue; Speaker 2, Red; Speaker 3,
Yellow; and Speaker 4, Green.

Figure 54a. A standard window function used for
basic amplitude panning over a quadraphonic speaker
configuration.

Figure 54b. A phase shift of the standard
window function at 0, 0.25, 0.5 and 0.75 phase.
In Figure 54b, we can see that a phase of 0.5 corresponds to an amplitude ratio of one in Speaker 1, and zero in all other speakers. Similarly at a phase of zero we have an amplitude ratio of one in Speaker 3, and zero in all other speakers. In other words by shifting the phase we can effectively pan the source around four channels. As this is performed in the frequency domain, the convolution\textsuperscript{146} is applied to consecutive spectral bins within each FFT frame, determining the weights for all individual spectral bands. The result is the construction of a unique filter curve for all speaker channels that consists of the weights of all frequency bands with a frame of the FFT. Each lookup stage is phase-shifted according to the values 0, 0.25, 0.5 and 0.75, as demonstrated in Figure 55.

\textbf{Figure 55.} Inside the pfft\textasciitilde patcher showing the spectral process applied.

Although \textit{Model A} was originally implemented in \textit{MaxMSP} version 5, a minor change in the release of \textit{MaxMSP} version 6 saw that audio send\textasciitilde and receive\textasciitilde ceased to communicate between a main patcher window and a pfft\textasciitilde subpatch. This required a necessary change to include [fftin\textasciitilde # nofft] and [fftout\textasciitilde # nofft] objects instead,

\textsuperscript{146} Convolution is achieved through multiplication in the frequency domain.
ensuring that data could be sent in and out of the \texttt{pfft}~ subpatch without being converted from the time domain to the frequency domain and vice versa.

It is worth noting that for circular trajectories, another approach of implementing the phase shift would be with application of block delays dependent on the relative to the size of the FFT frame. Figure 56 shows block sample delays used for the same effect over a four-channel equidistant speaker configuration with an FFT size of 2048 samples (James & Hope, 2012).

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure56.png}
\caption{An adaptation of Figure 60 that utilises \texttt{[fft~ 1 nofft]} objects instead of \texttt{[send~]} and \texttt{[receive~]} as well as the use of \texttt{[delay~]} instead of \texttt{[+-]}.
}\end{figure}

It should be noted here that the use of a slight curve in the window function, as opposed to using a linear ramp or triangular function compensates for the ‘hole in the middle’ scenario that is often referred to in reproducing the phantom image (Kendall & Cabrera, 2011; Roads, 1996). This function creates a consistent level of amplitude as the sound passes from one speaker to the other in a smooth continuous pan. This
window function generally ensures we have unity gain, or an amplitude ratio of one across the system or 0 dB.\textsuperscript{147}

### 3.3.2 Supported Speaker Configurations

Because basic amplitude panning is used, Model A relies on a circular and equidistant configuration of speakers, although the number of speakers used is variable. Testing of this model was performed over four- and eight-speaker configurations, but in theory could be extended to the common powers-of-two (16 or 32), or speakers of an arbitrary number, provided they are arranged in a circular and equidistant fashion. Figures 57a and b show two equidistant speaker configurations with the associated angles of displacement.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{speaker_configurations.png}
\caption{An equidistant quadraphonic speaker configuration.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{speaker_configurations.png}
\caption{An equidistant octophonic speaker configuration.}
\end{figure}

\textsuperscript{147} Without accounting for the non-linear implications of speaker placement, speaker frequency response, audience position, acoustical properties of the space, physiology and subjective interpretation. These factors are variable in all performance scenarios.
The panning loudness curves in basic amplitude panning do not support non-equidistant configurations, which would ultimately introduce a distortion of the phantom image between different pairs of speakers. This would be evident when panning across a Dolby 5.1 system (see Figure 57c), and would be most apparent with moving sound sources across the surround left and right channels. The more speaker channels the more precise the spatial rendering.

For larger speaker configurations, the number of frequency bins that require control will increase respectively. Equation 10 describes the relationship between the number of frequency bins, $b$, of all speakers, up to $N$, and the FFT frame size $W$.

$$\sum_{i=1}^{N} b = \frac{W}{2} \times N$$  \hspace{1cm} (10)

For example, for an FFT size of 1024 samples, and eight speakers, we would have 4096 frequency bins to control simultaneously for every FFT frame. In the case of

---


149 The 3-to-1 rule when applied to a speaker monitoring setup suggests that in order for there to be equal loudness when cross-fading from one speaker to the other, both signals need to output at an amplitude ratio of $1/\sqrt{2}$ at the centre position. If the speakers are more narrowly or widely spaced than 90° then a different weighting should ideally be applied. This is also necessary in order to maintain equal perceived loudness across the entire system.
spatialising different frequency bins within an FFT, we are in fact dealing with an input source of $N$ frequency bins; this may commonly be 256, 512 or 1024. Table 7 tabulates this explicitly showing the number of control parameters based on the number of speaker channels and the FFT frame size.

Table 7

*The Associated Number of Spectral Bands that Require Control According to Multichannel Loudspeaker Configuration and FFT Frame Size*

<table>
<thead>
<tr>
<th>Speaker Channels</th>
<th>256-sample frame size</th>
<th>512-sample frame size</th>
<th>1024-sample frame size</th>
<th>2048-sample frame size</th>
<th>4096-sample frame size</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>512</td>
<td>1024</td>
<td>2048</td>
<td>4096</td>
<td>8192</td>
</tr>
<tr>
<td>8</td>
<td>1024</td>
<td>2048</td>
<td>4096</td>
<td>8192</td>
<td>16384</td>
</tr>
<tr>
<td>16</td>
<td>2048</td>
<td>4096</td>
<td>8192</td>
<td>16384</td>
<td>32768</td>
</tr>
<tr>
<td>32</td>
<td>4096</td>
<td>8192</td>
<td>16384</td>
<td>32768</td>
<td>65536</td>
</tr>
</tbody>
</table>

3.3.3 Calibration

The terrain map, when stored in a Jitter matrix, is often stored using a char data range, an 8-bit format used primarily for video, and is within the range of 0–255, or $2^8$ bits in precision. However for the purposes of simplification, terrain maps are normalised and plotted with ranges of 0–1. In many cases, due to the size of the terrain map or 2D lookup table, a lack of precision can occur for values that are read between whole number index pairs $(x, y)$. To avoid this quantisation, linear interpolation is recommended during the WTS lookup process.

The polar terrain curve, illustrated in Figure 58a, is an ideal terrain surface for spatialisation with Model A as the angle of azimuth around the centre of this curve is described linearly. The use of the hue colour map to describe differences in azimuth is
appropriate given that angles of 360° wraparound to 0°. The colour grading is exactly the same here as in the colour wheel pictured in Figure 58b, so we can immediately recognise that a specific colour, such as the aqua–blue, will coincide with a centre pan. In Figure 58a we can see that this colour corresponds with a phase of 0.5, and in Figure 58b we can see that this colour corresponds with Loudspeaker 3.

The purpose of calibration is to ensure that the results of the mapping process match what we are expecting to hear. This research treats front centre as a reference marker, associating this with a terrain height \( z \) that is half way within its own domain range. We have already observed that the aqua–blue in Figures 58a and b correspond with a front centre pan and a phase of 0.5. This results in the majority of the energy reproduced in Speaker 3. Instead, if we were working with a quadraphonic system, as shown in 59b, Front Centre is in fact between Speakers 1 and 2, and so for a front centre pan we need an additional phase offset of +0.125. Therefore, for a quadraphonic configuration, Speakers 1–4 that correspond with a circular configuration L, R, RR and RL should have phase offsets of +0.125, +0.875, +0.625 and +0.375 respectively, as shown in Figure 59a.
Figure 58a. An ideal terrain surface for spatialisation using Model A mapping where differences in colour conform to differences in angle of azimuth.

Figure 58b. The corresponding hues in Figure 58a shown against the speaker placements allowing for an intuitive relationship of colour to angle of azimuth.

Figure 59a. Speaker 1 blue (left), Speaker 2 red (right), Speaker 3 yellow (rear right), Speaker 4 green (rear left).

Figure 59b. Assignment to a four-speaker configuration.

The same principles apply to an eight-channel speaker distribution with its window function:

\[
f(x) = \begin{cases} 
0, & x < 0.375 \\
\sqrt{8x - 3}, & 0.375 \leq x < 0.5 \\
\sqrt{2x - 1}, & 0.5 \leq x < 0.625 \\
0, & x \geq 0.625 
\end{cases} \quad (11)
\]
Speakers 1–8 will have phase offsets of +0.125, +0, +0.875, +0.75, +0.625, +0.5, +0.375 and +0.25 accordingly, as shown in Figure 60a.

The same principles can be applied to other speaker configurations. For a 16-channel circular distribution we subdivide further with phase offsets +0.125, +0.0625, +0, +0.9375, +0.875, +0.8125, +0.75, +0.6875, +0.625, +0.5625, +0.5, +0.4375, +0.375, 0.3125, +0.25 and +0.1875.

Figure 61 illustrates three different panning curves used for different speaker configurations. The speaker configurations associated with each curve make an assumption that the sound when passing from one speaker to another will only sound in a maximum of two speakers at a time. This can be observed in Figures 59a and 60a, as the crossovers only involve a maximum of two speakers at a time. The window function provides an indicator of the spatial width of the rendered spatial movements. By widening the size of the window, it is possible to extend the window so that up to four speakers may be active at any time, and consequently spatial cues will be perceived as being generally wider. Alternatively, narrower window functions will highlight spatial trajectories and allow them to be perceived as being more directional. In both cases there are a range of outcomes from doing this.
If we apply the window function we used earlier for a quadraphonic speaker configuration (Figure 59a) and apply this same window function to eight loudspeakers, we have up to four loudspeakers sounding at a time, as shown in Figure 62a with their respective overlaps. What results is an overall amplitude ratio of two, resulting in a 6dB increase over the system. With larger speaker configurations we have even more overlap, so in a 16-speaker configuration, as shown in 62b, up to eight speakers sound at a time, resulting in an amplitude ratio of four across the system correlating with an increase of 12dB.

In order to account for these increases in level for larger window sizes, window functions may need to be scaled accordingly to account for these changes in amplitude. The amplitude ratio is equal to the speaker overlap divided by two, so for eight speakers we have an amplitude ratio of two with an overlap of four, and for 16 speakers we have an amplitude ratio of four and an overlap of eight. Of course reducing the width again brings us back to unity gain.
Although this particularly mapping strategy theoretically only applies changes in azimuth, adjusting the window size ‘between’ the window functions, shown in Figure 61, results in spectromorphology—that is, a non-linear distribution of energy throughout the spectrum. Appendix D shows some of these spectromorphologies, and this is also briefly discussed in Section 4.2.3.1.

In conclusion, although Model A presented an efficient approach to controlling the spatialisation of spectral bands, due to non-linearities of the system this meant that the relationship between the terrain and trajectory, and the resulting sound shape were not always intuitive. This is discussed further in Chapter 4. Some other observations of Model A were that it presented limitations in that the direction of the trajectory path always determined the spectral distribution, and as the terrain and trajectory structures work together topographically, it meant that some outcomes were frequently unavoidable, such as the energy of frequency bins wrapping around from highest to lowest and vice versa.\(^{150}\) I was eager to find a solution that counteracted these. In short, I was after a model that translated the topography of the terrain structure from its visual representation to a more literal audible analogy.

\(^{150}\) Although this was a very interesting effect, much like the Risset tones—a forever ascending spatial sound shape associated with a rotated sound scene discussed in Chapter 4.
3.4 Model B: Mapping Terrain Height to Spectral Bin

Eric Lyon (2012) describes a method where digital images provide the computer with detailed control information for diffusion. Although the article describes a very different mapping strategy, it was this process as well as my own dissatisfaction with Model A that drove me to consider alternative approaches to mapping. Fundamentally the point of exploring a different mapping strategy was motivated by the need to establish a stronger correlation between the terrain and trajectory structures and the resulting sound shape. This led to a two-year project on the development of Model B, a mapping that interprets the height of the terrain \( z \) as difference in frequency. Differences in \( z \) are rendered as differences in colour in the software’s visual interface, and so in this way colours displayed are mapped to frequency. These frequencies are then spatialised independently based on their spatial position on the terrain. Model B spatialises spectra independently with both azimuth and distance cues across 2D horizontal speaker configurations, as well as elevation cues for 3D volumetric speaker configurations.

Smalley’s concept of perspectival space, connecting timbre spatialisation with the visual arts, enables a conceptual alignment with the ‘painting’ of colour onto a canvas—this canvas being analogous to the spatial listening area. I wanted to express a painting of timbre, or harmonics across space, where the room is the canvas. Graphical sound synthesis methods such as Oramics, Percy Grainger’s Free Music machines, and Iannis Xenakis’s Upic (Marino et al., 1993) system involve the drawing of sound in time. This concept has been extended into the digital domain with systems including Christopher Penrose’s HyperUpic, UI Software Metasyth, Izotope Iris, and the

---

Virtual ANS\textsuperscript{155} iOS application for iPhone and iPad. Lyon (2012) discusses the automated mapping of video to space including the work of Shawn Greenlee and Augustus Leudar, a PhD student at SARC,\textsuperscript{156} who has worked with video to space in a quadraphonic context, using Ambisonic tools for spatialisation. Kim-Boyle (2008) has also been involved in building a perceptually informed interface for control, opting for a visually intuitive method of amplitude scaling spectra by the brightness of pixels in a Jitter matrix.

The process adopted in Model B is shown more explicitly in Figure 63 where the WTS lookup process is used to define the frequency or spectral bin to be spatialised, and the spatialisation of the spectra is determined by the real-time spatial coordinates of the trajectory. This process is essentially three-fold: first values are read from the terrain map using coordinates determined by the trajectory, the coordinates of the trajectory calculate an $n$-channel weighted distribution dependent on $n$ loudspeakers, and finally these weighted values are stored according to their spectral bin in $n$ histograms that calculate the SPF for each speaker.

The result is an eloquent system: the topography of the terrain describes a timbre distribution or spectromorphology, and the trajectory defines a sound shape or spatiomorphology. In order to achieve this, additional software was developed by the author. This includes audio rate and FFT frame-synchronous histograms, audio rate and FFT frame-synchronous SPF smoothing algorithms, an FFT frame-synchronous spectral centroid width modulator, an FFT frame-synchronous spectral spline interpolator, and audio rate implementations of panning techniques including AEP and DBAP. These were all developed in MaxMSP either in Java code embedded in the $mxj~$ object, Javascript code embedded in the $js$ object or using $codebox$ inside $gen~$.

\textsuperscript{155} Retrieved 10\textsuperscript{th} Jan 2015 from https://itunes.apple.com/au/app/virtual-ans/id711384847?mt=8
\textsuperscript{156} Sonic Arts Research Centre, Queens University, Belfast.
Model B allowed for a more flexible model in the end, giving rise to an intuitive relationship between WTS and the resulting timbre spatialisation, and allowing the performer separated control over the nature of spectromorphology and spatiomorphology generated as shown in Figure 64. In this model, the terrain surface describes the spectromorphology and the trajectory curve describes a series of points from this terrain that are used for describing the frequency–space distribution. Where the terrain and trajectory intersect is what determines the resulting sound shape. Figure 64b shows a sound shape that emerges using a noisy and asynchronous trajectory v. Figure 64c which shows a sound shape that emerges using a synchronous and periodic trajectory. Figure 64d shows Figure 64b with a listener and four speakers superimposed to show in context that these diagrams represent a frequency–space distribution of different spectra that are ultimately spatially panned according to their spatial position.
on the diagram. Each point in Figures 64b, c and d are colour coded using the colour scale introduced in Section 1.2.4, where red corresponds with low frequency and blue corresponds with high frequency through the audible frequency range.

![Figure 64a](image1.png)

**Figure 64a.** A greyscale contour plot of a non-linear terrain surface. Differences in colour correspond with frequency.

![Figure 64b](image2.png)

**Figure 64b.** A visualisation of a sound shape generated over 1 second (44100 audio rate lookups) using an asynchronous random trajectory reading points off the terrain pictured in Figure 64a.

![Figure 64c](image3.png)

**Figure 64c.** A visualisation of a sound shape generated over 1 second (44100 audio rate lookups) using a synchronous and periodic trajectory reading points off the terrain pictured in Figure 64a.

![Figure 64d](image4.png)

**Figure 64d.** A visualisation of the sound shape in Figure 64b with a listener and loudspeakers superimposed to show an ideal visual representation of the perceived sound shape.

As illustrated in Figure 65, *Model B* offers the laptop performer not only more flexibility in determining the nature of *spectromorphology* and *spatiomorphology*, but it
also allows for other kinds of speaker configurations that *Model A* did not account for. The two panning models, AEP and DBAP, allow the laptop performer to configure the *timbre spatialisation* for other non-equidistant and arbitrary loudspeaker placements.

![Venn diagram showing the adaptability of Model B to spectromorphology and spatiomorphology over various speaker configurations.](image)

**Figure 65.** A Venn diagram showing the adaptability of Model B to *spectromorphology* and *spatiomorphology* over various speaker configurations.

### 3.4.1 The First Implementation

Early implementations of *Model B* provided a range of challenges. Most of these were largely computational and related to the speed and synchronicity of the necessary processes involved.

The computation of the FFT in *MaxMSP* is managed within a vector of samples that matches the size of the FFT window. Frequency domain signals within FFT processes involve a sequential counting-through of spectral bins according to an index as outlined in Section 2.3.3. This means that control signals for FFT processes need to conform to this same vector format. The WTS lookup process reads terrain $z$ values as corresponding with frequency; however this will usually occur in a highly non-linear way. It was necessary to structure these data in a particular way using a histogram.$^{157}$

---

$^{157}$ A histogram is a data process often used in statistical analysis that accumulates the history of a parameter over a given time interval.
that allows the frequency events to be accumulated in a table before sending them to the frequency domain process for convolution synthesis.

The early model utilised some of the processing functions in the Jitter library for MaxMSP, such as jit.histogram and jit.normalise. Although the read and write processes to and from the Jitter matrix are audio rate, the entire SPF is reliant specifically on the speed at which jit.histogram can compute the data. The results consequently had significant audible segmentation artefacts as Jitter computation rates are much slower than the window size of the FFT. Further, the histogram had to be zeroed with a clear message that was not synchronised, leading to further undesirable artefacts introduced.

In order to calculate the processing latency, the various computational processes were benchmarked\(^{158}\) and converted to their equivalent proportion with respect to FFT frame lengths, given the audio sampling rate, and the FFT size as shown in Table 8. Ideally 16 channels of 256-point histograms could be computed within a single 256-sample FFT frame at 44100 Hz audio sampling rate. However, these benchmarks prove unreliable in practice as further latency is incurred when additional parallel processes take precedence, and hence Jitter processes are either cued or dropped to prioritise other real-time computations. The slack scheduler timing of computing the histogram resulted in a ‘smearing’ over of the detailed movements often evident in the trajectory path of lookup. These details had been evident in Model A, and in this early implementation, it was apparent this detail had been lost.

---

\(^{158}\) Benchmarking in computing refers to an efficiency and speed test of computer hardware or software algorithms.
Table 8

_Benchmark Timings of Histogram Processes_

<table>
<thead>
<tr>
<th></th>
<th>1 256-point</th>
<th>8 256-point</th>
<th>16 256-point</th>
<th>1 256-point + jit.normalize</th>
<th>8 256-point + jit.normalize</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>jit.histogram</em></td>
<td>0.278462</td>
<td>2.549427</td>
<td>5.305182</td>
<td>0.299299</td>
<td>3.089392</td>
</tr>
<tr>
<td><em>jit.histograms</em></td>
<td>0.04796</td>
<td>0.43916</td>
<td>0.91390</td>
<td>0.05156</td>
<td>0.53219</td>
</tr>
</tbody>
</table>

However, the histogram solved an issue in _Model A_ whereby the information generated from the terrain was inherently bound by the order in which these events were read from the _terrain_—this being determined by the path of the _trajectory_. By resorting to a histogram analysis of the _terrain_ topography it means that no assumptions need to be made about the order of events within the time scale of an FFT frame, meaning the trajectory could traverse in the opposite direction and the results computed would be exactly the same.

A second potential problem arose where continuous terrain curves become discontinuous curves after computation of the histogram, as illustrated in Figure 66.\(^\text{159}\) The results tend to be clumpy and discontinuous histograms, as illustrated in Figure 66b, and using this curve as an SPF function creates results that sound severely comb filtered. Although this can be used to interesting effect, it was also necessary to interpolate across these values in order to create smooth and continuous SPFs. It was not until later in this research project that tools were developed to counteract this

\(^\text{159}\) This can be exacerbated when a data source is of a lower resolution than the histogram table. For example, a terrain contour defined by an 8-bit data type _char_ is inadequate for describing all of the frequency bins within a 2048-sample size FFT that is 10-bit. Values are interpolated from the terrain to account for more accuracy considering the 256 steps of an 8-bit _char_ matrix is hardly enough to span the scope of a large FFT window, the values of the terrain contour either must be interpolated or the terrain must be generated with the appropriate range—that is, a matrix of 32-bit signed integer _long_ data type, or 32-bit _float32_ floating point data type.
problem. At this stage a novel approach that used `jit.bsort` instead of `jit.histogram` proved to be less CPU demanding, even though the outcome was not technically faithful to the original intention of *Model B*. *Jit.bsort* allowed for ‘biasing’ the spectral contours by a prevalence of low or high spectral bands.

Although `jit.histogram` or `jit.bsort` will determine the necessary SPFs, the spatialisation of these spectral bands also requires discussion. Several possibilities were investigated, such as storing the spatial cues in a second matrix for lookup, but the immediate solution was to segment the resulting audio signal after WTS before a series of histograms generate the SPFs for each loudspeaker. As this early implementation was still limited to an equidistant and circular speaker configuration, a circular trajectory is used, and this trajectory within the time of an FFT frame is segmented $n$ times, where $n$ is the number of speakers. These segments of the circle are analogous to the slices of a pizza, whereby the trajectory that traverses its circumference will determine frequencies that correspond to each loudspeaker. Each series of points is separated discreetly to
individual channels, meaning there is no timbral ‘spill’ between speakers, so the model is not performing panning as such. This process is illustrated in Figure 67.

Figure 67. The sub-patcher responsible for ‘splitting’ the trajectory into segments.

The way in which the signal is managed has some similarities with the process of time division multiplexing. After accounting for the respective window sizes within each FFT, calculating the multiplexing window sizes for different speaker configurations is shown in Table 9.

Table 9

| Number of Audio Samples Assigned to Each Speaker for Different FFT Frame Sizes |
|----------------------------------|------------------|------------------|------------------|------------------|------------------|
|                                  | 128 samples      | 256 samples      | 512 samples      | 1024 samples     | 2048 samples     |
| 2 speakers                       | 64               | 128              | 256              | 512              | 1024             |
| 4 speakers                       | 32               | 64               | 128              | 256              | 512              |
| 8 speakers                       | 16               | 32               | 64               | 128              | 256              |
| 16 speakers                      | 8                | 16               | 32               | 64               | 128              |
| 32 speakers                      | 4                | 8                | 16               | 32               | 64               |
| 64 speakers                      | 2                | 4                | 8                | 16               | 32               |

Although I devised a means of controlling the amount of spill between adjacent speakers across the system.
A by-product of this approach is that the frequency-space resolution is completely dependent on the number of loudspeakers used. By essentially windowing the SPF function by the equivalent number of speakers used, the inherent spatial topography is lost to a kind of quantised abstraction of the terrain surface realised in sonic form. Even the spatial resolution of panning algorithms is dependent on loudspeakers, as the number of loudspeakers affects the quality of the reconstruction of a sound scene. However as there is no panning of sound spectra here, perception of localisation is quantised by the number of loudspeakers.

Figure 68. The process of signal segmentation, histogram computation and smoothing explored in the early implementation of Model B.

The histogram computed in the Jitter library does cause some asynchronous behaviour with the frequency domain convolution process, so I decided to combine Model A with this early implementation by essentially convolving the two systems
together. This resulted in the best of both worlds: responsiveness of audio rate morphologies, along with the more intuitive and accurate representations of colour across the multichannel system. This meant the quantisation issue was not as apparent as spatial panning cues would be rendered by Model A and simply just filtered by the early implementation of Model B. In this way Model A was responsible for the *spatiomorphology*, and this early approach to Model B was responsible for the *spectromorphology*.

Flexibility in speaker configurations, most particularly the option of working in the elevated dimension also became desirable:

Composers of electroacoustic music have engaged with elevated loudspeaker configurations since the first performances of these works in the 1950s. Currently, the majority of electroacoustic compositions continue to be presented with a horizontal loudspeaker configuration. Although human auditory perception is three dimensional, music composition has not adequately exploited the creative possibilities of the elevated dimension. (Sazdov et al., 2003, p. 1)

I was concerned at how I would accommodate other panning algorithms with the existing mapping strategy given azimuth cues in Model A are computed at audio rate. On the other hand, none of the existing third-party software that supports panning over 3D loudspeaker configurations supported audio rate control, probably as it is not normally required! Instead, I explored a method that had two instances of WTS running concurrently—the first determining the horizontal distribution and the second determining the vertical distribution, and finally these are convolved in the frequency domain in a matrix configuration.

### 3.4.2 Ambisonic and DBAP Spatialisation

The early prototyping and testing phase for Model B was worthwhile in that it made it clear that the concept and mapping could work provided fast timing resolution is maintained, and that the processing is tightly synchronised to the FFT frame. As was
found in *Model A*, if these conditions are met and the model adheres to an audio rate model, this allows for a highly responsive model.\(^{161}\)

Functioning of this model relies on it being able to calculate spatial cues at the audio sampling rate. I considered Kim-Boyle’s approach where he integrated Ville Pulkki’s *vbap*\(^{162}\) software, and used *vectral~* to smooth out transitions across a series of FFT frames. Kim-Boyle’s implementation is controlled using event-based control of spatialisation. It was not out of the question to do this too, but again this would have compromised the responsiveness of the model and for reasons to be explained in Chapter 4, it will have compromised some of the more detailed and evolving *sound shapes* generated, particularly those exhibiting *spatial texture*.

Upon reflecting on the model, and in terms of iterative creative methodology, it became apparent that I needed to introduce more extensive spatial cues for spatialising spectral bands, and retain a design that is based on a more intuitive interpretation of colour. This necessitated the development of audio-rate spatial algorithms, and several other further developments that will be discussed in the following sections.

Currently the only implementation of an audio rate spatialiser is the basic amplitude panning mechanism used in *Model A*, but I researched two further panning models—AEP and DBAP spatialisation—with the intention of coding the algorithms from the ground up. The interest in both of these spatial algorithms is two-fold. First, both are able to render sound in a virtual 2D or 3D speaker configuration using distance, azimuth and elevation cues. Second, they are adaptable to various speaker configurations, meaning that the *timbre spatialisation* instrument would be transferable from system to system proving compatibility with a wide range of in-house installations.

\(^{161}\) Responsive in a real-time sense to performer action.
This new version of Model B combines the best of both the responsiveness of Model A without the path of the trajectory affecting the spectral evolution: rather the two structures, terrain and trajectory, determine spectromorphology and spatiomorphology independently.

Both implementations are coded in gen-. The gen- codebox operates on a synchronous scheduler, so input from inlets gets updated once per sample for signals.\textsuperscript{163}

What follows is some contextualisation of the processes involved in Model B, including some background on Ambisonic techniques v. DBAP. A discussion of audio rate histograms and interpolation and smoothing methods follows. Finally in this section, supported speaker configurations for Model B are discussed, along with their calibration.

3.4.2.1 Ambisonic Equivalent Panning

Ambisonics is a full-sphere surround sound technique: in addition to the horizontal plane, it covers sound sources above and below the listener (Gerzon, 1973). Ambisonics was developed in the UK in the 1970s under the auspices of the British National Research Development Corporation. In contrast to other panning techniques, Ambisonic panning functions normally produce signals for all speakers at the same time. The functions are defined on the whole horizontal circle or the whole sphere. All speaker gains sum to 1. Unlike other multichannel surround formats, its transmission channels do not carry speaker signals. Instead, they contain a speaker-independent representation of a soundfield called B-format, which is then decoded to the listener’s speaker setup.

Ambisonics originated as a recording and reproduction method for reconstructing a periphonic 3D soundfield. Ambisonics can be understood as a 3D extension of M/S (mid/side) stereo, adding additional difference channels for height and

\textsuperscript{163} See McCulloch (2013).
depth. It is this resulting signal set that is called **B-format**. Its component channels are labelled $W$ for the sound pressure (the M in M/S), $X$ for the front-minus-back sound pressure gradient, $Y$ for left-minus-right (the S in M/S) and $Z$ for up-minus-down. This is referred to as first-order Ambisonics. The coordinate system here follows the right-hand rule\(^{164}\) convention with positive $X$ pointing forwards, positive $Y$ pointing to the left and positive $Z$ pointing upwards. Horizontal angles run anticlockwise from due front, and vertical angles are positive above the horizontal, negative below.

The spatial resolution of first-order Ambisonics as described above is quite low. In practice, that translates to slightly blurry sources, but also to a comparably small usable listening area or sweet spot. In Ambisonics, the auditor should be ideally placed in the very centre of the circle or the soundfield image will be distorted. The resolution can be increased and the sweet spot enlarged by adding groups of more selective directional components to the **B-format**. The resulting signal set is then called second-, third-, or collectively, **HOA**. The polar patterns of these different orders take the form of a series of spherical harmonics showing increasing detail in the resolution of the construction of the soundfield, as illustrated in Figure 69.

The option of panning sound sources using Ambisonics is also possible. It became apparent that taking spherical coordinates—azimuth, elevation and distance (AED)—representing the virtual position for a sound source, this could be used to encode $W$, $X$, $Y$ and $Z$ **B-format** data, which could then be decoded as in-phase and out-of-phase weights for $n$ number of speakers in a circular or spherical configuration. There are many different decoders used in this field and basic decoders, such as the regular polygon decoder only support equidistant circular speaker configurations, the square 4.0 decoder being one of the most commonly used for quadraphonic speaker

---

\(^{164}\) In mathematics and physics, the right-hand rule is a common mnemonic for understanding notation conventions for vectors in three dimensions. There are several right-hand rules that make it easy to understand the invisible matters or substances (University of Delaware Department of Physics and Astronomy, n.d.).
configurations. Some more sophisticated decoders will support conversion to Dolby 5.1 surround, and to some more arbitrary speaker configurations and binaural.\footnote{For a current list of supported software, Retrieved 10\textsuperscript{th} Jan 2015 from http://en.wikipedia.org/wiki/List_of_Ambisonic_Software}

![Figure 69. Visual representation of the Ambisonic B-format components up to third order. Dark portions represent regions where the polarity is inverted. Note how the first two rows correspond to omnidirectional and figure-of-eight microphone polar patterns. Source: Wikipedia, 2015.](image)

One of the arguments of Martin Neukom and Jan Schacher is that when using traditional Ambisonics for panning sound sources—as the method is based on harmonic decomposition, and in order to increase spatial resolution and directivity—it is possible in some instances to produce side effects such as signals on speakers far away from the original sound position and inverted phases (Neukom & Schacher, 2008): while weighting the Ambisonic channels according to their order—that is, for \( n \) speakers and order \( m \), \( n \) is equal to \( 2m + 1 \) in 2D and \( (m + 1)^2 \) in 3D (Neukom, 2007)—the side effects can be reduced but at the cost of the precision of the directivity. Figure 70 shows a third-order Ambisonic function used to determine speaker weights (with the bars indicating 13 symmetrically positioned speakers) for Ambisonics using basic decoding
and an alternative method called in-phase decoding. The first has both positive and negative speaker weights whereas in the second, all of the speaker weights are positive. In-phase decoding\(^{166}\) methods can be used to overcome the constraint at the expense of spatial resolution and offer the possibility of an audience spread across the circle (Barrett, 2012).\(^{167}\) This second approach, AEP, is investigated as part of Model B.

\[ \text{Figure 70. Two-level functions for a speaker at position (sound at } \theta = 0, \text{ order } m = 3), \text{ the first without correction (basic decoding) } f_{\text{bas}}(\theta) \text{ and the second for so-called in-phase decoding } f_{\text{inph}}(\theta). \text{ The bars indicate the levels of 13 symmetrically positioned speakers. Source: Neukom and Schacher, 2008.} \]

One particular advantage of AEP over other panning methods is that the diffusion performer can determine the extent of directivity by adjusting the order, where order of zero is omnidirectional, one is cardioid, two is hypercardioid et cetera. As the order increases, the resulting spatialisation becomes increasingly localised (see Figure 71). However, the main point of interest here is the spatial width of these different orders as can be seen in the crossover that occurs with lower order values. A change in

---

\(^{166}\) And other decoding methods like MaxRe (as alternatives to Ambisonic basic decoding).

the order can also result in an increase of the perceived loudness of the source. The converse applies when narrowing the function with higher orders. A potential problem occurs for orders \( m \) larger than the number of speakers \( n \). If \( m > n \), problems arise in the reconstruction of the phantom image between speakers, but this does not apply when \( m < n \).

![Figure 71a. Ambisonic equivalent panning (AEP) Order 1.](image)

![Figure 71b. AEP Order 4.](image)

![Figure 71c. AEP Order 16.](image)

Although Ambisonics does not support the rendering of distance cues, the documentation by Neukom and Schacher and its implementation in the ICST\(^{168}\) Ambisonics library for MaxMSP extend the algorithm to account for distance encoding. This is controlled with a separate function that weights all of the speakers globally.

As the AEP algorithm supports not only give localisation cues but also a control for spatial width, this can be adapted in various ways in live performance for a wide variety of intentional diffusion. It should be stated here that the ICST implementation does bind the order of directivity to the distance, so as sources move further away they become narrower, and when they move closer they have more spatial width, until when they are centre they are omnipresent. The result of this process is that we have many

---

\(^{168}\) The Institute for Computer Music and Sound Technology in Zürich, Switzerland.
different spectra with different orders of directivity forming a very complex spatial soundfield.

Another advantage of AEP is for single source panning: the encoding and decoding processes outplay Ambisonic panning using basic decoding methods (Neukom & Schacher, 2008).

The implementation of audio rate AEP by the author takes either two or three audio signals that each describes a stream of coordinates (XYZ) or (AED). These are computed in two models shown in Figure 72 for 2D and 3D speaker configurations. Care was taken to ensure the algorithm worked seamlessly with existing GUI objects designed in both the HOA and ICST Ambisonics libraries. The panning algorithms are documented in Appendix B and the gen~ code is documented in Appendix C.

3.4.2.2 Distance-based Amplitude Panning

DBAP is a matrix-based spatialisation technique that takes the actual positions of the speakers in space as the point of departure, while making no assumptions as to
where the listeners are situated (Lossius et al., 2009). This makes DBAP useful for a number of real-world situations such as concerts, stage productions, installations and museum sound design where predefined geometric speaker configurations may not apply (Lossius, 2008). The method uses vector-based calculations based on the distance of the virtual source to each subsequent speaker to determine a series of speaker weights where the total gain of the system remains 1.

One of the parameters this technique allows for is spatial blur. In some ways this could be aligned with the notion of anti-directivity, because by increasing the amount of blur, when sources move over speakers, they will begin to incorporate more of the speakers grouped locally; whereas with a low level of blur, DBAP ensures that when a sound source approaches a speaker, other speakers will strongly attenuate that source. In this way spatial blur serves to increase a sense of spatial width and therefore encourages a sense of immersion.

The algorithm also has a loudness rolloff setting for determining how quickly sounds are attenuated the further they move away from a designated speaker. A rolloff of $R = 6$ dB equals the inverse distance law for sound propagating in a free field. For closed or semi-closed environments $R$ will generally be lower, in the range 3–5 dB, and depend on reflections and reverberation (Everest, 2000).

The *gen~* implementations of DBAP in *MaxMSP*, like the AEP implementations, allow for two or three audio signals that determine a stream of ($XYZ$) coordinates for spatialisation that are computed at audio rate for both 2D and 3D speaker configurations as shown in Figure 73. Like the AEP implementation, this model allows for giving each spectral bin a different spatial blur or rolloff also, depending on the context—although in most instances these parameters would tend to be used as optimisations to the algorithm in order to account for the idiosyncrasies of the listening
environment. The algorithms are documented in Appendix B and the gen~ code is documented in Appendix C.

Figure 73a. A 2D implementation of distance-based amplitude panning.

Figure 73b. A 3D implementation of distance-based amplitude panning.

3.4.3 Audio Rate Histograms in Gen~

Now with two audio rate methods for spatialising spectral bins at audio rates, building of software capable of computing a histogram at audio rates within the length of an FFT frame was required. Depending on the number of speakers used (n), Model B would need an equivalent n number of histograms to accumulate the various speaker weights computed by either AEP or DBAP for the relevant spectral bands. Computation of the histogram was relatively simple in gen~. The poke object, which is responsible for writing values into a memory buffer, has an accumulator function suitable for this process.

The resolution of the data read from the terrain will influence how ‘rough’ the histogram appears. Data are rarely smooth in statistical analysis; rather it clumps in certain regions of the data range. Smoothing algorithms need to be applied to ensure the
creation of smooth SPFs. These smoothing algorithms, like the histogram, also needed to be audio rate and FFT synchronised.

The degree of interpolation and smoothing applied to the histogram can dramatically influence the nature of the SPF functions generated, as can be seen in Figure 74 with smoothing over 3, 10, 30 and 80 samples respectively.

Eugene Fink, Ankur Sarin and Jaime Carbonell (2009) describe two solutions to smoothing the underlying probability distribution calculated by a histogram: the greedy algorithm and the faster algorithm. In a C++ implementation of their algorithm, it can smooth 500,000 points in 1.4 seconds—that is, 256 points in 0.7168 milliseconds.

Sixteen instances of these smoothing functions for multichannel application may be
optimally capable of processing within 11.5 milliseconds, which does exceed the length of smaller FFT sizes. This also makes an assumption that there are no other computational processes running concurrently, so in actuality this process in context could be significantly slower.

Existing smoothing functions in MaxMSP might be applied as a more efficient way of reducing the extent of noise in the resulting SPF function. Unconventional low-pass filters ramps\textit{smooth}\textasciitilde and slide\textasciitilde or low-pass filters lores\textasciitilde and svf\textasciitilde are some ways of reducing the noise in a signal, or alternatively average\textasciitilde will create a moving average function through the system. The most effective of these is the low-pass filter. The most effective filters were found to be frequency domain and linear phase as they did not introduce phase shift of harmonic components, and did not blur values from frame to frame in the FFT process.

For the purposes of creating continuous SPF functions, spline interpolation curves also present a solution. The gen\textasciitilde library for MaxMSP offers some interpolation functions for audio signals. In Figure 75 we see the contents of a histogram smoothed with the use of spline curves. The gen\textasciitilde patch is responsible for computing interpolation between non-zero values in a buffer dynamically. Figure 75 uses one of the more intensive and accurate interpolations used: Catmull–Rom spline interpolation.\footnote{gen\textasciitilde offers linear, cosine, cubic and Catmull–Rom spline interpolation as part of the buffer related objects including nearest, sample, peek, lookup and wave.}
As solutions had to be computed within a small number of FFT frames, it was decided to build two implementations in the `gen~` environment of *MaxMSP*: spectral centroid width and spectral Catmull–Rom spline interpolation. Manipulation of spectral centroid is the most efficient and allows the user to control the centroid width of spectral energy surrounding ‘bumps’ in the histogram (shown in Figure 76).

Spectral wraparound was another user parameter added so as to control whether high frequencies fold over into low-frequency bands and vice versa, represented in Figure 77.
This spectral width control allows for a very musical way of shifting from noisy sounds to pitched sounds (see Figure 78). The process is achieved by processing four tasks within the time of four FFT frames:

1. write data for a frame
2. simultaneously count through data in reverse with temporal write and count through data forwards with temporal write
3. count through data forwards in all buffers and make combined signal while detecting the largest known value in the frame
4. delay signal in Step 3 by a frame and normalise it according to the largest known value in the frame, and erase data in main audio buffer.

The *Catmull–Rom spline interpolator* allows the user to control the interpolation width. Both of these processes were implemented in *gen~* using the following steps:

1. write data for a frame
2. count through data in reverse with temporal write and count through data forwards with temporal write
3. reconstruct SPF based on average position of points and store these in a new buffer
4. interpolate through the points defined in the buffer, and erase data in main audio buffer.

Figure 79 illustrates Catmull-Rom spline interpolation applied to white noise stored within an FFT frame.
The methods of audio rate AEP and DBAP spatialisation present solutions to the spatialisation of spectra at FFT frame rates over non-equidistant and arbitrary speaker configurations, including those that involve the elevated dimension. What is more, both AEP and DBAP account for distance and directivity/blur in the synthesis of spatial movements. The WTS process also generates an audio rate signal representing a series of frequencies, which when accumulated by a histogram, provide an SPF. Due to the nature of the curves generated through the process of computing the histogram, these distributions required smoothing and interpolation in order to avoid comb filtering effects. Two implementations have been discussed that present solutions to smoothing and interpolation that are frame synchronised to the FFT: SPF smoothing using *spectral centroid width* and SPF interpolation using *spectral Catmull–Rom spline interpolation*. What follows is an overview of supported speaker configurations for *Model B*, and their subsequent calibration.

### 3.4.4 Supported Speaker Configurations

The subsequent implementation of *Model B* explores two methods of panning that make this model adaptable to a wide range of speaker configurations, ranging from circular and spherical, to very arbitrary configurations. The implementation of AEP allows for some variations in the angles of displacement with a circular or spherical speaker configuration. The additional implementation of DBAP does not have these circular and spherical restrictions, and can work with very abstract kinds of groupings of speakers in both 2D planar configurations, and also in 3D volumetric configurations.

In the case of Ambisonics, the minimum number of loudspeakers is three for an order of decomposition of 1. Additionally, in this context the loudspeakers must be placed on a circle equidistant from each other. However, the angles of speakers distributed around this configuration can be customised for non-equidistant configurations, such as Dolby 5.1.
In the case of more arbitrary speaker placements in both 2D and 3D speaker configurations, DBAP ultimately proved to be the more appropriate panning method. The advantage here is that timbre spatialisation may be ‘tuned’ for in-house speaker system and installation works, taking into consideration the height and distance of the various speakers within the room.

Both AEP and DBAP offer some further parameters such as the level of directivity or the extent of spatial blur, both of these correlating with (to varying degrees) the psychoacoustic notion of spatial width. In this way the performer may choose to modulate the extent of directivity of each and every spectral band—allowing the individual sounds to emerge as points in space—or to determine whether they are treated more immersively across the entire space.

3.4.5 Calibration

Calibration of Model B by the author has maintained the same messaging methods ICST and Jamoma have adopted for their AEP and DBAP models in MaxMSP.

AEP uses a method of specifying the coordinates of speakers in the form ‘aed [speaker number] [speaker azimuth] [speaker elevation] [speaker distance]’. For 16 speakers in a circular and equidistant configuration we may calibrate the system in the following way:

```
aed 1 315. 0. 1.;
aed 2 -22.5 0. 1.;
aed 3 0. 0. 1.;
aed 4 22.5 0. 1.;
aed 5 45. 0. 1.;
aed 6 67.5 0. 1.;
aed 7 90. 0. 1.;
aed 8 112.5 0. 1.;
aed 9 135. 0. 1.;
aed 10 157.5 0. 1.;
aed 11 180. 0. 1.;
aed 12 202.5 0. 1.;
aed 13 225. 0. 1.;
aed 14 247.5 0. 1.;
aed 15 270. 0. 1.;
aed 16 292.5 0. 1.
```
For a Dolby 5.1 system there is an industry standard for matching speaker positions to speaker channel: Channel 1 is left, Channel 2 right, Channel 3 rear left, Channel 4 rear right, Channel 5 centre, and Channel 6 is the subwoofer. In this way we can calibrate the channels as follows:

\[
\text{a}ed 1 -30. 0. 1.; \\
\text{a}ed 2 30. 0. 1.; \\
\text{a}ed 3 -110. 0. 1.; \\
\text{a}ed 4 110. 0. 1.; \\
\text{a}ed 5 0. 0. 1.
\]

The subwoofer is given a feed of all channels and reduced in level by almost 14 dB to account for the potential increase in gain of five external sources, as well as a low-pass filter to ensure that the subwoofer is reproducing frequencies largely in its own frequency range, just as the other speakers should have a high-pass filter to ensure they are not trying to reproduce low frequency.

The SpADE facility at DMARC, University of Limerick, shown in Figure 80, presents a 3D speaker configuration of two concentric circles of speakers.

\[\text{Figure 80a. A speaker configuration at the Digital Media and Arts Research Centre.}\]

\[\text{Figure 80b. The displacement of the upper speaker group as opposed to the lower group.}\]
The elevation angle can be calculated given the distance measurements shown in Figure 80b.

\[ \theta = \sin^{-1} \left( \frac{171}{242} \right) = 44.96^\circ \]  

(12)

The distance of the lower speaker group proportionally to the upper speaker group is:

\[ d = \frac{171}{242} = 0.707 \]  

(13)

We can calibrate Model B using AEP in the following way:

aed 1 0. 0. 0.707;
aed 2 30. 0. 0.707;
aed 3 45. 0. 0.707;
aed 4 60. 0. 0.707;
aed 5 90. 0. 0.707;
aed 6 120. 0. 0.707;
aed 7 135. 0. 0.707;
aed 8 180. 0. 0.707;
aed 9 -135. 0. 0.707;
aed 10 -120. 0. 0.707;
aed 11 -90. 0. 0.707;
aed 12 -60. 0. 0.707;
aed 13 -45. 0. 0.707;
aed 14 -30. 0. 0.707;
aed 15 0. 44.96 1.;
aed 16 30. 44.96 1.;
aed 17 45. 44.96 1.;
aed 18 60. 44.96 1.;
aed 19 90. 44.96 1.;
aed 20 120. 44.96 1.;
aed 21 135. 44.96 1.;
aed 22 180. 44.96 1.;
aed 23 -135. 44.96 1.;
aed 24 -120. 44.96 1.;
aed 25 -90. 44.96 1.;
aed 26 -60. 44.96 1.;
aed 27 -45. 44.96 1.;
aed 28 -30. 44.96 1.

DBAP has a similar method of calibration. However, rather than spherical coordinates used in AEP, DBAP uses Cartesian coordinates. Depending on the size of the space in metres, the coordinates should be entered relative to metres, as the dB per metre rolloff used in the algorithm is also specified this way. The Jamoma DBAP
implementation has adopted a message in the format ‘dst_position [speaker channel number] [speaker x coordinate] [speaker y coordinate] [speaker z coordinate]’:

```plaintext
dst_position 1 9.95625 10.8373 0;
dst_position 2 12.66075 10.8373 0;
dst_position 3 14.58675 9.37135 0;
dst_position 4 14.58675 7.17315 0;
dst_position 5 12.66075 5.6782 0;
dst_position 6 9.95625 5.6782 0;
dst_position 7 8.1135 7.17315 0;
dst_position 8 8.1135 9.37135 0;
dst_position 9 11.27925 11.75225 1;
dst_position 10 15.048 10.7764 1;
dst_position 11 16.7175 8.27225 1;
dst_position 12 14.904 5.7391 1;
dst_position 13 11.27925 4.8836 1;
dst_position 14 7.6815 5.64775 1;
dst_position 15 5.95575 8.2418 1;
dst_position 16 7.62525 10.86775 1;
dst_position 17 9.495 9.84135 1;
dst_position 18 13.0635 9.84135 1;
dst_position 19 9.495 7.0209 1;
dst_position 20 13.0635 7.0209 1;
dst_position 21 3.5685 12.1191 1;
dst_position 22 19.3635 12.1191 1;
dst_position 23 3.5955 4.42685 1;
dst_position 24 14.70375 3.48 1
```

### 3.5 Chapter Summary

In this chapter, two models for using WTS as a means of spatialising spectral bands have been discussed. These two models each focus on a different mapping strategy. *Model A* maps *terrain* height values read by a *trajectory* signal as spatial azimuth cues for successive spectral bins. This model creates a diverse range of *sound shapes* that explore predominantly *spatiomorphology* over equidistant speaker configurations. *Model B* maps *terrain* height values read by a *trajectory* signal as frequency, and required a more sophisticated computational model involving histograms and smoothing algorithms. The additional need for exploring the spatialisation of spectral bands over a wide range of speaker configurations, including 3D arrangements, led to the development of audio rate implementations of AEP and
DBAP spatialisation techniques. Model B consequently creates a diverse range of sound shapes that explore both spectromorphology and spatiomorphology.

The AEP and DBAP modules I developed were implemented using gen~ in MaxMSP. However there were some limitations in MaxMSP version 6 with respect to the number of audio inputs and outputs supported. A maximum of 16 audio outputs is supported by gen~, therefore placing constraints on the DBAP implementation due to its adaptive normalisation stage in the algorithm. On the other hand, the AEP modules are mutually exclusive and allow for many 16-channel gen~ modules to be used in tandem, thus facilitating 32-, 48- and 96-channel speaker configurations. The possibility of higher channel counts in this case ultimately boils down to a CPU issue, as 96 different audio rate stages of histograms, smoothing functions, table store and lookup, convolution and inverse FFT processes is hugely computationally intensive. The need for efficiency is clear and the entire model would benefit from being implemented at lower level. Alternatively, processes can be sacrificed, but this may also compromise the detail of sound shapes and their morphological evolution.