AMBISONICS SYMPOSIUM 2009
June 25-27, Graz

AMBIGRAINER - A HIGHER ORDER AMBISONIC GRANULATOR IN PD

Nicholas Mariette[1]

[1] LIMSI-CNRS, Orsay, France (Nicholas.Mariette@limsi.fr)

Abstract: *This paper presents AmbiGrainer - a higher order Ambisonic granulator developed in the Pure Data signal processing environment. AmbiGrainer is capable of spatializing individual grains in two dimensions, in spherical coordinates, using the IEM higher order Ambisonics encoding objects. Individual grain positions in the sound-field can be controlled directly or the meta-characteristics of the grain cloud could be controlled separately using an algorithm of choice. For example, the grain cloud could be directed to a single direction and distance, randomised according to a chosen probability distribution, or driven by a particle swarming or flocking model. This functionality enables the spatial shape of the generated grain cloud to be manipulated as desired to truly incorporate space as a musical parameter. AmbiGrainer outputs audio as Ambisonic channels that may then be decoded to a speaker array or binaural audio for headphones. Other features of AmbiGrainer are its capacity for real-time variable polyphony (numbers of simultaneous grains), synchronous or asynchronous sound-file granulation, and external control via OSC. In synchronous mode, playback rate is prioritised so that timbre may be altered by varying other parameters (such as grain size) while preserving tempo.*

Key words: Granular Synthesis, Sound Granulation, Higher-Order Ambisonics

1 INTRODUCTION

AmbiGrainer is a higher-order Ambisonic granulator developed in the Pure Data (Pd) signal processing environment, capable of spatializing individual grains in elevation and azimuth. This enables the spatial shape of the synthesised grain cloud to be manipulated, thus incorporating space as a musical parameter. Audio is output as Ambisonic channels that may then be decoded to a speaker array or binaural audio for headphones. The grain scheduler is also capable of synchronous or asynchronous sound-file granulation, variable numbers of simultaneous grain streams (polyphony), and external control via OSC. Grain parameter calculation from user interface controls has been designed so that perceptual characteristics can be controlled directly, in turn controlling several underlying per-grain parameters.

AmbiGrainer has been developed from earlier stereo and first-order Ambisonic versions that have both been used by the author and collaborators to produce live, improvised and studio electroacoustic music, and components of sound-art installations.

2 BACKGROUND

Granular synthesis is a method of producing sounds across the time and frequency axes, by overlapping playback of large numbers of short (commonly < 50 ms) sound "grains". Each grain is comprised of an arbitrary sound waveform with a short amplitude envelope, often in the shape of a Gaussian "bell curve" function or a trapezoid with parameterised attack and decay times. In *granular synthesis*, the underlying signal is a pure synthesised signal such as a sine tone at a given frequency. This method was first implemented for computer music composition by Roads in 1975 [1]. Alternatively, in sound *granulation*, the signal under the envelope is a sampled sound, as first implemented as a real-time process by Truax [2].

Prior to Roads' investigations, in 1971, Xenakis had developed a "temporary hypothesis" (to serve his realisations of stochastic music) that the nature of all sound could be represented as an "integration of grains, of elementary sonic particles" with "duration, frequency and intensity" [3, pp. 43, 373]. Xenakis' proposition was itself informed by Gabor's idea of "acoustical quanta" as elementary signals that mathematically represent sounds simultaneously in terms of time and frequency [4], see also [5]. Gabor presented this description of sound as a means to better express the elementary sensation of hearing in both frequency and time, which is not articulated by describing sound signals as a function of time, or its Fourier transform. He also notes that similarly, other descriptions of a phenomenon such as sound may be found by simultaneously investigating two aspects of the phenomenon that are "neither identical nor independent".

Space is another dimension in the description of real physical sound fields that exists simultaneously with time and frequency. The granulation and granular synthesis of sound presented by AmbiGrainer might be seen as a step towards empirical synthesis of arbitrary sound fields in space, frequency and time. Gabor's acoustical quanta enable description of sound shapes with changeable time and frequency structures, on an axis between delta functions and pure sinusoids. In the same way, spatialised granular synthesis opens an axis between acute directional sound sources and totally diffuse, enveloping soundfields. The effect is related to parallel developments in spatial sound-field generation such as Directional Audio Coding (DirAC) [6], which essentially quantises the spatialisation of particles of a sound-field.

The extension of sound granulation or granular synthesis into the spatial axis has been considered since at least 1988, when Roads noted: "in a system capable of multichannel sound synthesis, each grain can be assigned to a particular spatial location" [7]. Since then, many granular synthesis implementations enable some manner of specifying individual grain spatial positions. Many implement stereo panning of grains, with random positioning, which enables pleasing diffuse stereo effects of granular textures (e.g. [8]). Several implementations note spatialisation of grains in terms of relatively arbitrary "multi-channel" panning (e.g. [9, 10, 11, 12]). First-order Ambisonic granulation has also been noted as a capability of a few implementations [13, 14, 15]. Wakefield implemented up to third-order Ambisonic granulation in the Max/MSP environment [14]. However, in comparison to the present Pd implementation, grain sequencing control was more rudimentary, without built-in parameter randomization. Also, the implentation was hard-coded as a closed-source Max/MSP object, which prevents further extension of the implementation. Control of Ambisonic order is also not a real-time variable parameter, whereas it is for the current implementation.

3 DESIGN

AmbiGrainer was developed in the Pure Data (Pd) signal processing environment [16] primarily for its strengths in rapid prototyping and modification of audio effects. The present patches are based on the author's stereo granular synthesis implementation (*patch*) released to the Pd list in 2002 [8]. This was developed further into a first-order Ambisonic version in 2007 for a live surround electronic music performance [17, 15].

The Pd patches created to implement AmbiGrainer form three functional layers. At the highest level is a graphical user interface (GUI) developed using Pd's graph on parent feature that enables the patch to be instantiated within another patch as a new Pd object (abstraction) with its own GUI design (Figure 1). At present, this abstraction has one input for control parameters, and 16 audio outputs for third-order Ambisonic channels that may then be connected to other Pd objects or abstractions.

The middle layer of AmbiGrainer connects the GUI through to the bottom, synthesis layer of the patch. In this section the raw sound file to be granulated is loaded into a single memory table, and the grain window function is calculated and stored in another table. A great deal of the character and range of the resulting sounds can be controlled by the loaded sound file. More traditional *granular synthesis* results from using a synthetic waveform such as a sine tone, while the broader possibilities of *granulation* result from using real sound recordings. A mix of these two paradigms could be effected by loading a waveform of a sine wave concatenated with a recording, then using the *playback position* parameter and randomisation to vary between the two.

The middle layer also translates user control parameters to grain synthesis and scheduling parameters via several formulae. These calculations also affect some of the user control parameters, which are then updated on the GUI at a limited rate (presently set to 100ms), for display only.

Any randomisation of parameters is applied at this stage, resulting in a fixed set of parameters for each grain to be synthesised. Firstly, the grain timing parameters are calculated and used to control a `metro` object, which triggers the calculation of other grain parameters at the necessary rate. Parameter sets are then fed through to the individual grain synthesis units via *not quite poly version 4* (`nqpoly4`) [18], a third-party polyphony construction set comprised of two patches.

Nqpoly4 provides a framework that programmatically generates a chosen number of copies of an individual synthesis unit patch (or *voice*) and connects their inputs and outputs to their surroundings. It also provides polyphonic voice management, which allocates incoming synthesis control parameters to free voices without any special organisation by the external patch. Within nqpoly4, incoming parameters are passed to the next available free voice, which then performs its synthesis process. While a voice is busy, it passes new incoming parameters onto the next voice, and so-on, until either a free voice is allocated, or no more voices are available. If all voices are busy, that set of parameters is dropped. Once a voice has finished its synthesis process, it once again accepts new parameters.

Finally, the deepest layer of AmbiGrainer is the grain synthesiser patch itself. This section is relatively simple, taking the per-grain synthesis parameters, playing back the requested section of the raw sound sample, multiplying it by the window function, and then panning it to a set of third-order Ambisonic outputs. The sample and window function playback is calculated using a 4-point polynomial interpolation (using the object `tabread4~`) driven by a high-precision ramp function (`vline~`). These functions provide high quality resampling (transposition) and time scheduling of the grain.

The grain's Ambisonic output channels are then sent back to the middle layer and mixed with the output of all other grains. This combined Ambisonic audio is finally amplified or attenuated by a global level parameter and made available at the outputs of the top-layer object for connection with other objects, or with the digital to analogue converter (`dac~`).

3.1. User control and per-grain parameters

As described already several formulae translate the user control parameters to scheduling and synthesis parameters per-grain. This transformation occurs because the user parameters have been designed to have greater perceptual relevance than the underlying grain parameters. In each section below, first the user control parameters are described, then any formula for calculating the low-level grain parameters is presented.

Playback rate

This controls the synchronous or asynchronous playback of the source audio sample. When set to zero, playback is asynchronous, with grain position determined directly by the user *grain position* and *randomness* parameters. At any other value, synchronous playback occurs with any positive or negative speed. A value of 1 plays the source audio at its natural rate, a negative value plays it backwards, values between -1 and 1 are slower than normal, and values outside this range are faster than normal.

Grain position in sample

This is the position of the start of the grain within the source audio sample. Grain position is only modified by the randomisation amount, so that the actual (low-level) grain position will vary by the chosen amount around the present selected position.

Grain position can be driven synchronously (to play back the source audio in sequence), or asynchronously (at the position chosen by the user). Any automatic movement of the grain position during synchronous playback is controlled using the *playback rate* parameter.

Grains per second (optional)

This straightforward parameter indicates how many grain parameter sets are presently being calculated every second. However, depending on available polyphony and other parameters, it is possible that not all grains will be synthesised. This parameter is designated *optional* because at any time, either it, or the *grain size* (but not both), may be selected for direct control. The un-controlled parameter is then determined, for display only, by Equation (1) – as a function of the *number of simultaneous grains* and *transpose* values, along with some constants (such as sample rate).

$$GPS \propto \frac{Num\,streams \times Transpose\,rate}{Grain\,size} \quad (1)$$

Only the *grain size* is directly used in grain scheduling – not the value of *grains per second* – because its perceptual effect is less intuitive. In fact, *grains per second* controls various perceptual effects from tempo (for low values) to timbre (for high values). When it is selected by the user directly, the chosen tempo or timbre is kept constant regardless of adjustments of *number of simultaneous grains* or *transpose* values. This is a novel feature of the AmbiGrainer implementation.

Grain size (optional)

This sets the grain duration in milliseconds. As noted above, *grain size* may be either controlled or displayed in mutual exclusion with the *grains per second* parameter. When displayed, its value is determined by the appropriately refactored version of Equation (1). When directly controlled, grain size may be modified by the randomisation amount.

Number of streams

This controls the number of voices (simultaneous grains) being used at present. The number of streams also affects either the *grain size* or *grains per second* parameters according to Equation (1). The upper limit is hard-coded into the patch, but it can be modified depending on the host machine's computation capacity. Note however that many parameters affect momentary computation load, particularly those that affect the number of grains per second.

Grain transpose

This sets the pitch transposition of individual grains in positive or negative, whole or partial semitones. The value can be randomised and also affects either the *grain size* or *grains per second* parameters according to Equation (1).

Grain amplitude (level)

This straightforward parameter affects the amplitude of individual grains, and may be randomised as desired. Increasing randomisation leads to increasing perceptual roughness of the total resulting sound, particularly for high grain densities.

Randomisation amounts

In all cases, where a parameter may be modified by a randomisation amount, this value is a uniquely calculated random number with uniform distribution, sampled from a white noise signal. It would be possible to parameterise the type of random distribution to explore the sound of various stochastic controls.

3.2. Ambisonic control

The addition of per-grain spatialisation features to this granular synthesis implementation opens many possibilites for utilisation to produce complex, two- or three-dimensional sound fields. While many spatialisation methods are possible (such as amplitude panning, or even wave-field synthesis), higher-order Ambisonics is particularly suitable because of its economical encoding process and output channel numbers. Ambisonics also separates the spatialisation from the output format, enabling decoding to almost arbitrary speaker arrays (of enough channels), or even binaural

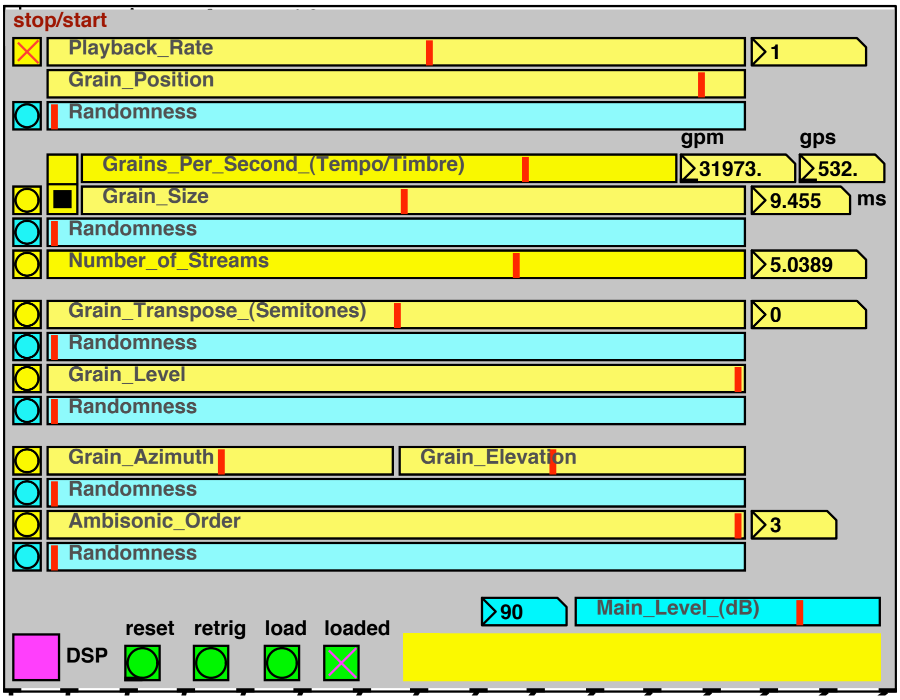

Figure 1: AmbiGrainer user interface

for headphones.

Grain azimuth and elevation

These two parameters control the spatial location of each grain in spherical coordinates of azimuth and elevation, expressed in degrees. Azimuth spans from 0° to 360° (with 0° at front, 180° at rear, and positive clockwise). Elevation spans from -90° (below) to 90° (above) with 0° in front. Both parameters may also be randomized by a chosen amount.

Grain Ambisonic order

The Ambisonic order may be set per grain from 0-3 (zeroth to third order) with a chosen amount of randomization. Lower Ambisonic orders produce perceptually wider sound sources (with greater localization blur), and conversely, higher orders enable more precise angular localization.

3.3. Grain scheduling

Grain scheduling is not a directly controllable parameter, but a process that determines the timing with which each grain is triggered to start. At the simplest level, two types of grain scheduling are available in the present patch: synchronous and asynchronous scheduling. The type of scheduling being used at any time is determined by the setting for *playback rate*.

$$IOT \propto \frac{Grain\,size}{Transpose\,rate \times Num\,streams \times Playback\,rate} \qquad (2)$$

3.4. Control methods and possibilities

In prior granular synthesis systems, particularly those developed and used by composers, musical capabilities are significantly expanded by the development of meta-control methods of the underlying parameters. For instance, Roads developed the concept of *events* [1] that control the shapes of granular synthesis musical objects in amplitude, frequency and time. Events use randomization parameters to specify two-dimensional areas for parameters such as frequency, instead of more conventional one-dimensional lines possible in most sequencer software (e.g. in the frequency axis). In this manner, granular synthesis is used in a more sculptural way than many other processes such as additive or subtractive synthesis.

4 SPATIAL EFFECTS

The provision of per-grain Ambisonic spatialisation opens up a wide range of possible spatial effects. For instance, direct panning of the individual grains behaves slightly different to panning of the complete result, with the effect being one of *dragging* the sound around, leaving behind a fading tail that depends on current grain lengths. This occurs because each grain is only panned to a single static position, so for longer grains, and faster motions, the leading edge of the grain cloud is in a different position to grains that were

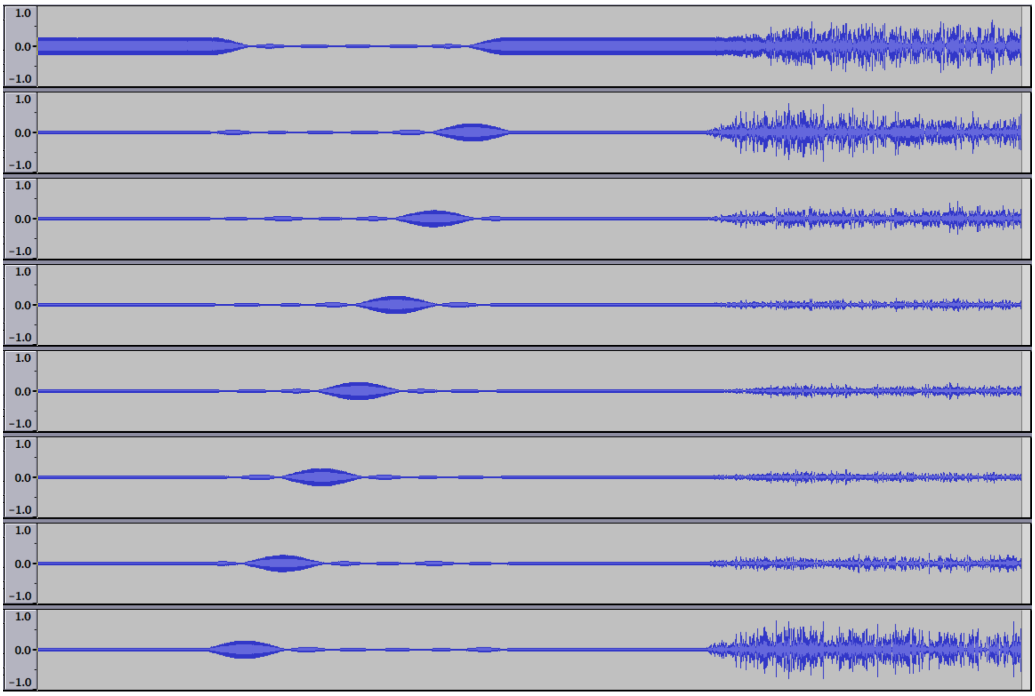

Figure 2: Multi-channel output decoded to a 2D circular speaker array from a third-order Ambisonic signal. Grains are panned around the circle in Azimuth, then direction is randomised to simulate a highly diffuse source.

triggered a short time before. Thus, panned sounds become like a comet with a tail.

Randomization of grain spatial direction perceptually simulates either a broader sound source or a fragmented cloud of grains (see Figure 2), depending on the momentary grain density and amount of randomisation. Temporally proximate grains are more likely to fuse into a perceptually single sound source (apparently wider than a point source) for smaller amounts of randomization, or larger grain density values (number of grains per second).

This granular, time-variant method of synthesising source area extents can be contrasted to more common methods of multichannel decorrelation, which are usually time invariant [19, 20]. Perceptually, the result of this granular source spread effect is likely to be quite similar to time-variant decorrelation methods where the randomized phase, all-pass decorrelation filter coefficients are changed at regular block intervals.

4.1. Physical acoustics analogue

Purely as a matter of curiosity, it is possible to imagine a physically possible, though very complex and unlikely, architectural construction that could produce acoustically, the same sound processing as this and many other sound granulation software implementations. At the core of such a construction would be the raw sound source (producing the sound stored as a sample in the software), and a multichannel, periphonic receiver (such as a higher-order Ambisonic microphone). To begin with the simplest building block of the granulator is the delay line, which can be represented by a simple, hard reflective surface at a given distance from a sound source and a receiver. Thus each grain could be produced by the appearance of a reflective surface at the appropriate distance (delay time) and angular bearing in relation to the receiver. One can imagine the appearance of the appropriate reflector as the rapid rotation of a suitably placed thin panel from a sideways orientation (with respect to the receiver) to a frontal orientation. Grain transpositions could be produced as doppler effects created by the rapid motion of reflecting panels towards or away from the receiver for the duration of the grain playback. In such a manner, Ambisonic sound granulation can be imagined as the acoustic product of a complex architecture-in-motion.

5 CONCLUSIONS AND FURTHER WORK

This paper presented AmbiGrainer, a higher-order Ambisonic sound file granulator (or granular synthesiser). AmbiGrainer is capable of spatialising each grain individually in azimuth and elevation, including per-grain Ambisonic order settings (presently from zeroth to third order). Grain spatialisation may then be controlled by meta parameters that perform at a stochastic level, such as the present implementation of a control for “randomisation amount”, using a uniform distribution. The addition of sophisticated, per-grain spatialisation, granular synthesis offers great flexibility for generating complex spatial environmental or musical sound fields.

Many further features have been conceived but not yet implemented, and the open nature of granular synthesis opens the doors to potentially infinite ways of specifying the synthesis and scheduling of grains. The range of possible non-spatialisation effects is vast, and future parameters may include per-grain glissando, or filtering. A live input buffer could also be implemented to enable real-time spatial granulation performance.

Spatial effects could be controlled by complex algorithms such as flocking, swarming or cellular automata simulations, simply by adding external patches to generate a stream of grain spatial parameters. Further per-grain parameters that could be added include a distance effect, using near-field Ambisonic rendering, or one or several reverberation channel(s) to control per-grain direct-to-reverberant ratio. Outside the patch, reverberation channel(s) could be processed with an Ambisonic convolution reverberator.

Finally, vital further work includes the exploration of sound compositions using the tool. Just as the compositional constructions of stochastically-specified *musical events* by Xenakis, Roads, Truax and others lead to vital new musical ideas, the spatial possibilities of Ambisonic granulation open the way to new ways of working with space in music.

6 ACKNOWLEDGMENTS

AmbiGrainer is built in the Pure Data (Pd) signal processing environment by Miller Puckette [21], based on a stereo sound file granulator (nm-grainer) by the author, released to the Pd list in 2002 [8]. AmbiGrainer uses `nqpoly4`, a polyphonic patch constructor developed by Stephen (pix) Pickles, modified by Frank Barknecht [18]. The raised-cosine window generator patch and the basic grain player are based on patches available in the Pd help system. Ambisonic encoding is provided by Pd objects and patch examples by Thomas Musil at IEM, Graz, Austria (2002-2006).

REFERENCES

[1] C. Roads, "Automated granular synthesis of sound," *Computer Music Journal*, pp. 61–62, 1978.

[2] B. Truax, "Real-time granular synthesis with a digital signal processor," *Computer Music Journal*, pp. 14–26, 1988.

[3] I. Xenakis, *Formalized Music: Thought and mathematics in composition*. Pendragon Press, 1992.

[4] D. Gabor, "Acoustical quanta and the theory of hearing," *Nature*, vol. 159, no. 4044, pp. 591–594, 1947.

[5] D. Gabor *et al.*, "Theory of communication," *SPIE MILESTONE SERIES MS*, vol. 181, p. 120, 1946.

[6] V. Pulkki, "Spatial sound reproduction with directional audio coding," *JOURNAL-AUDIO ENGINEERING SOCIETY*, vol. 55, no. 6, p. 503, 2007.

[7] C. Roads, "Introduction to granular synthesis," *Computer Music Journal*, pp. 11–13, 1988.

[8] N. Mariette, "nm-grainer granular synthesis patch," Email to Pd list, 2002. [Online]. Available: http://markmail.org/message/xsyf5in7ik3w7siv

[9] N. Wolek, "Granular Toolkit v1. 0 for Cycling74's Max/MSP," *Journal SEAMUS, XVI (2)*, pp. 34–46, 2001.

[10] D. Thall, "Experiments in sound granulation and spatialization for immersive environments," Interactive Digital Multimedia IGERT Annual Research Review, 2004.

[11] J. Stelkens, "crusherx-live!" 1999-2007. [Online]. Available: http://www.crusher-x.de

[12] A. Fletcher, "Crusher-x live! granular synthesis software," *Computer Music Journal*, vol. 32, no. 4, pp. 91–93, 2008.

[13] M. Leahy, "SCREAM–SuperCollider Resource for Electro-Acoustic Music," in *Proc. of the 2004 Int. Computer Music Conf*, 2004, pp. 79–81.

[14] G. Wakefield. (2006) Ambisonic externals for max/msp. [Online]. Available: http://www.grahamwakefield.net/soft/ambi~/index.htm

[15] N. Mariette, "Ambisonic improv," 2007. [Online]. Available: http://www.ambisonia.com/Members/nickmariette/ambisonicfile.2007-04-24.4578066237

[16] M. Puckette, "Pure data: another integrated computer music environment," *Proceedings of the Second Intercollege Computer Music Concerts*, pp. 37–41, 1996.

[17] N. Mariette, "the NOW now series 2007 #6: Surround sound at Lan Franchis," Live electronic music performance, 13 April 2007. [Online]. Available: http://thenownow.net/2007/04/06/170/

[18] S. Pickles (pix) and F. Barknecht, "nqpoly4," Email to Pd list: http://markmail.org/message/r6bieyzttpblcby4, 2003. [Online]. Available: http://pix.test.at/pd/nqpoly/nqpoly4.html

[19] G. Kendall, "The decorrelation of audio signals and its impact on spatial imagery," *Computer Music Journal*, pp. 71–87, 1995.

[20] G. Potard and I. Burnett, "Decorrelation techniques for the rendering of apparent sound source width in 3D audio displays," in *Proc. of the 7th Int. Conf. on Digital Audio Effects*, 2004, pp. 5–8.

[21] M. Puckette, "Max at seventeen," *Computer Music Journal*, vol. 26, no. 4, pp. 31–43, 2002.