Third-Order Ambisonic Extensions for Max/MSP with Musical Applications

Graham Wakefield

Graduate Program in Media Arts and Technology, University California Santa Barbara
wakefield@mat.ucsb.edu

Abstract

This paper describes a package of extensions (externals) for Cycling '74's Max/MSP software to facilitate the exploration of Ambisonic techniques of up to third order. Areas of exploration well suited to the Max/MSP environment and techniques for composition within the Ambisonic domain using the presented externals are described.

1 Introduction

Ambisonics (Gerzon 1973) is a technique to re-create the impression of (or synthesize) a spatial sound-field; alternative techniques include Wave Field Synthesis (Berkhout 1988) and Vector Base Panning (Pulkki 1997). Ambisonics is a two-part process of encoding recorded or virtual spatial sources into an Ambisonic domain representation, and then decoding this representation onto an array of spatial outputs (typically loudspeakers). The Ambisonic domain is a multi-channel representation of spatial sound fields based upon cylindrical (2-D) or spherical (3-D) harmonics.

First-order Ambisonics, also known as the B-Format, encode sound-fields as an omni-directional signal (named W) plus three additional difference signals for each of the axes X, Y and Z. Higher Order Ambisonics (HOA) uses higher ordered spherical harmonics for additional encoded signals, increasing the detail of directional information and expanding the acceptable listening area of the decoded spatial sound field (Malham 2003).

This paper describes a package of externals developed by the author to add Ambisonic capabilities to Max/MSP up to third order with full access to the Ambisonic domain signals. Max/MSP (Puckette 1988, Zicarelli 1998) is a flexible visual data-flow programming environment with audio signal processing capabilities, including a C API for developers to extend the functionality of Max/MSP by building extensions (externals).

The rationale behind the development of the presented externals is twofold:

- To facilitate rapid experimentation and prototyping of Ambisonic-domain processing techniques (some examples follow in this paper)
- To develop tools for prototyping virtual environments within multi-user facilities such as the UCSB CNSI AlloSphere (Pope 2005, http://www.mat.ucsb.edu/allosphere/).

Ambisonic extensions for Max/MSP have until recently been limited to first-order representations (e.g. Courribet and Béguin 2005). Concurrently with the present set of externals, The Institute for Computer Music and Sound Technology ICST of the Zurich School of Music, Drama and Dance (Hochschule Musik und Theater Zürich HMT) announced a set of Ambisonics externals for Max/MSP, and both sets of externals share many capabilities. Beneficially, both the ICST externals and those described in this paper can be used together within the same application.

2 About the Ambisonic externals

The package described in this paper contains several different externals to encapsulate distinct elements of the Ambisonic process. Some aspects of these externals, however, are common to all. Instantiation arguments of the externals set the order of encoding, and whether they are two-dimensional (pantophonic) or three-dimensional (periphonic) formats, determining how many signal inlets and/or outlets the externals require. Ambisonic domain inlets and outlets are labeled by the channel naming conventions of Jérôme Daniel (2000) to assist the correct connection of signal graphs.

2.1 Encoding sources

A monophonic virtual or microphone source is encoded to the Ambisonic representation using the external *ambi.encode~*. The monophonic source signal is connected to the first inlet, and the azimuth and elevation control signals (in degrees or radians) are connected to the second and third inlets respectively (Figure 1). The outlets of the external are the encoded Ambisonic representation.

By default, the azimuth and elevation signal inputs are sampled at block-rate with linear interpolation, however a more CPU intensive sample accurate mode (causing the encoding matrix to be recalculated at every sample) is available if required by particular applications (e.g. fast moving sources).

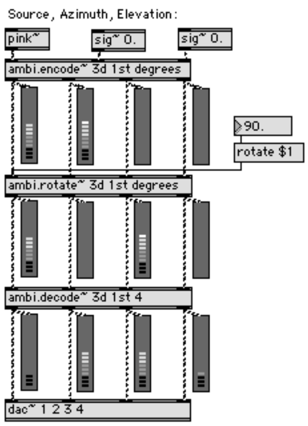

Figure 1. A simple Max/MSP patch, encoding a virtual source (white noise) at centre-front into a 3-D, first-order (B-Format) representation, rotated by π/2 radians, and decoded onto an array of four loudspeakers.

Since one of the principal benefits of Ambisonics is the compact representation of a complex sound field, the author provided an external that can encode up to 16 individually positioned virtual sources to a mixed encoded output, named *ambi.encoden~*. The third instantiation argument defines how many monophonic sources the object will encode, and thus how many signal inlets it will have. Spatial orientations of azimuth and elevation can be specified (in degrees or radians) for individual sources using Max messages (Figure 2); *ambi.encoden~* is better suited to static or infrequently moving objects since encoding matrices are recalculated at message rate.

Ambi.granulate~ offers the ability to create even larger numbers of spatial events, based upon the granulation of a buffered delay line – creating thousands of discrete Ambisonic events per second would be otherwise difficult in Max/MSP. The encoding matrix of each grain is calculated when the grain is scheduled. Parameters for amplitude, azimuth, elevation, duration, inter-onset time, delay etc. are set (as centre value and deviation range) using Max messages, and the sizes of the delay buffer and maximum grain concurrency per block are specified as instantiation arguments (limited by available RAM).

2.3 Decoding the sound field

The Ambisonic domain signal representation is decoded into a multi-channel array of channels, typically for a loudspeaker array, using the *ambi.decode~* external. The number of output channels is determined by instantiation arguments, and the relative orientations for each speaker are specified using Max messages. If no speaker orientations are specified, a clockwise horizontal ring beginning from centre-left is assumed as by default (Figures 1,2).

It should be noted that Ambisonics works best with regular orthonormal speaker layouts (Daniel 2000). For the 2-D case, a regular (polygonal) layout has all speakers equally spaced on the radius of a circle around the listener. In the 3-D case, only five regular polyhedra exist (the platonic solids), and thus usually a semi- or quasi-regular layout is sought, and maintaining an equatorial ring is usually preferred for psychoacoustic reasons. Hollerweger (2006) provides extensive detail and MATLAB code for the choice of speaker layouts in Ambisonic decoding.

Ambi.decode~ uses the projection method to generate the decoding matrix, which is less sensitive to the regularity of the speaker layout than the pseudo-inverse method. Nevertheless, directional and energy balance distortions will increase as the layout becomes more irregular (Daniel 2000, Hollerweger 2006).

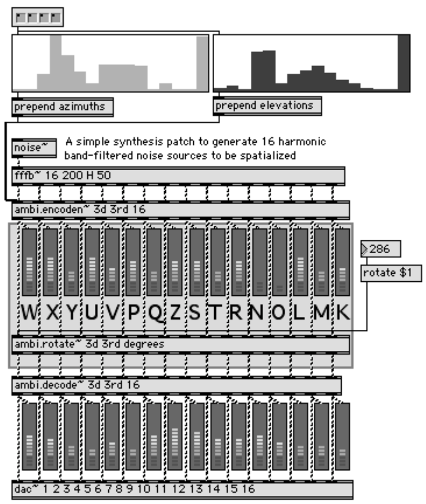

Figure 2. A Max/MSP patch encoding 16 individually positionable and tunable band-passed noise sources to a third-order Ambisonic representation, then decoded to a 16-speaker array.

Several different encoding conventions exist to scale Ambisonic signals at various orders. The original B-Format definition specified a scalar of √2 for the W channel, while the Furse-Malham set (Malham 1999) scales second-order signals to achieve normalized values in the Ambisonic domain, and Daniel (2000) specified up to third order without normalizing weights except for the original √2 W channel. In the presented Max/MSP externals the author opted to employ the same convention as presented by Daniel (2000) since the floating point signal operations of Max/MSP obviate the need for normalization, while retaining compatibility with existing B-Format recordings.

The *ambi.decode~* external includes sets of order coefficients from the literature (basic, in-phase, max-rE), which can be recalled using Max messages. In addition, Max messages can be used to directly set the balance between the zeroth- (omnidirectional), first-, second- and third-order encoded channels, to enhance or suppress the spatial separation, or to improve the spatial impression for combinations of distinct order representations and/or unusual speaker arrays.

2.3 Rotating the sound field

An Ambisonic sound-field can be rotated using a set of matrix operations (Malham 2003). The *ambi.rotate~* external can be inserted into the Ambisonic domain signal stream and used to rotate an entire sound field (of up to third order) around one or more of the principal axes (Figures 1,2). Max messages with float arguments of 'rotate', 'tilt' and 'tumble' will transform the field around the Z, Y and X axes respectively. When tilt and tumble are set to zero (the default), the rotation calculations are simplified for efficiency.

Rotating an entire sound field can be much more efficient than rotating each individual source when the number of sources is greater than the number of encoding channels. Sound field rotation is well suited to the sonification of a virtual environment, where navigation will shift apparent orientations of all sound sources in unison.

3 The Ambisonic externals in use

The Ambisonic externals described in this paper give the designer of a signal graph a great deal of flexibility to define spatial sound fields and explore spatial manipulation techniques, due to the direct access they provide to individual Ambisonic channels, and the comprehensive range of signal and event processing tools that the Max/MSP environment itself provides. In this section a number of possible areas of exploration will be described.

A particular benefit of the Ambisonic encoding is that summing the encoded channels will correctly superpose spatial representations encoded at the same order. A multitude of sources may therefore be individually manipulated before mixing down to a single Ambisonic representation to be decoded over many loudspeakers. Mixing Ambisonic signals of different orders is also possible, at the cost of spatial balance.

3.1 Using existing spatial recordings

A spatial sound field representation can be captured to disk as a multi-channel sound file using standard Max/MSP objects such as *sfrecord~, record~*, though it may be necessary to apply normalization to the signals when recording to an integer based file format. Similarly, an Ambisonic sound file can be played back over a speaker array using the Max/MSP built-in objects *sfplay~, play~* etc. connected to an *ambi.decode~* object. Placing the sound field manipulations between the file player objects and the decoder allow the real-time transformation of the stored Ambisonic sound field at playback.

Recordings for particular speaker arrangements can be somewhat generalized to larger speaker arrays using Ambisonics. For example, a typical 5.1 surround layout can be simulated in Ambisonics by a set of 5 virtual sources placed around the listener. A sound source encoded for 5.1 could be spatialized using ambi.encoden~ with five inputs, and the LFE channel submixed into the omnidirectional W Ambisonic channel, and the Ambisonic signals could then be decoded onto a larger array of loudspeakers.

3.2 Simulating distance cues

Spatial hearing based upon directional and distance based cues. Ambisonics is a convenient approach to tackle directional information since it is based on spherical coordinates. Distance is not represented in the standard Ambisonic encoding (for mathematical reasons all sources are assumed to lie on the surface of a unit sphere), however distance is a vital component in virtual environment simulation. Distance cues can be used in combination with Ambisonics however (Malham 1998), and existing Max/MSP signal processing objects can be used alongside the Ambisonics externals accordingly (Figure 3).

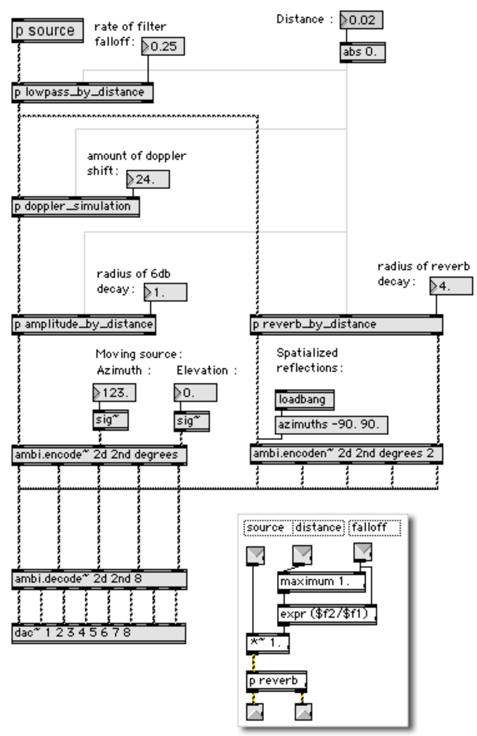

Figure 3: Simulating distance cues in combination with Ambisonics. Inset shows the *reverb_by_distance* sub-patcher.

Distance cues might include processing a sound source before encoding (including decreasing amplitude with increasing distance, applying IIR filters to simulate air absorption and near-field compensation (Daniel 2003), and using variable delays to mimic Doppler shift), as well as mixing in 'room effect' first reflection and tail reverberations to the encoded sound field according to distance and/or position.

The Jitter extensions to Max/MSP include bindings to OpenGL for three-dimensional rendering. Ambisonic encoding alongside distance cues can facilitate the auralization of rendered virtual worlds within a single application.

3.3 Processing in the Ambisonic domain

Access to a sample-accurate Ambisonic encoder suggests experimentation with high-frequency variations in spatial orientation. Furthermore, access to the Ambisonic domain signals within a flexible DSP environment invites experimentation with new manipulations of the sound field, and the possibility of directly synthesizing the directional difference signals (rather than or in addition to encoding virtual sources). For example, a relatively simple FM synthesis technique when used to generate differential signals of the Ambisonic domain itself may result in quite complex and new spatial sound events. Certainly it must be valid to attempt the extension of existing synthesis techniques with spatial parametrics, if possible! The author has for example explored passing comb filters in a feedback matrix across the Ambisonic channels, producing subtle, complex and chaotic spatial images warranting further investigation.

Certain signal processes are quite expensive and thus often cannot be duplicated per channel for large speaker arrays. Because the number of Ambisonic domain signals is almost always less than the number of loudspeakers, it can be more efficient to perform expensive operations in the Ambisonic domain.

The author has employed a similar approach in order to spatialize individual bands of a spectral resynthesis process. The spectral resynthesis produces a summed monophonic output from a large number of individual bands, but provides independent amplitude control for each voice. To spatialize the individual bands, four spectral resynthesizers were Ambisonic encoded as four virtual speaker locations (in a regular tetrahedral pyramid layout), and then decoded onto an array of 16 real loudspeakers. The amplitude of a particular frequency band can thus be modulated spatially quite efficiently using four rather than sixteen banks, and although there is a tradeoff of spatial accuracy, the sonic effect is quite effective. A similar approach might be taken to simulate frequency variant radiation patterns, as described by Daniel (2003).

4 Conclusions

Ambisonics is a powerful and efficient means to work with spatial digital audio. Second- and third-order Ambisonics involve complex algorithms that are more efficiently and conveniently managed using compiled C externals than existing Max/MSP objects. The open-ended nature of the Max/MSP signal graph architecture, and the access to the Ambisonic domain signals that the presented externals deliver, provide a flexible environment for exploring new kinds of signal manipulations and techniques for composition in the Ambisonic domain.

The extensions described in this paper constitute a minimal set to begin exploring Ambisonics in electro-acoustic composition, however a few key areas are currently under consideration for future development. Speaker arrays of 30 or more may demand the development of fourth, fifth or higher order encoding and decoding externals to expand the sweet area. A HRTF decoder for headphone listening would be a welcome addition.

At the time of writing, the externals are available as binaries from the author's website (http://www.grahamwakefield.net).

5 Acknowledgments

The presented Ambisonic extensions to Max/MSP were inspired by work on the CSL signal-processing library (Pope, Ramakrishnan, 2003) alongside Florian Hollerweger (IEM, Graz, Austria), and Jorge Castellanos (MAT, Santa Barbara, USA).

References

Berkhout A. J., 1988. A Holographic Approach to Acoustic Control, *AES Journal*, 36(12), pp 977-995.

Courribet B., Béguin R. 2005. *Outils de panoramisation pour l'environnement , 5.1 dans MaxMSP*. Journées d'informatique musicale, June 2005.

Daniel, J. 2000 *Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia* PhD thesis, Université Paris 6

Daniel, J. 2003. Spatial Sound Encoding Including Near Field Effect: Introducing Distance Coding Filters and a Viable, New Ambisonic Format. *Proceedings of the AES 23rd International Conference,* Copenhagen, Denmark, 2003.

Gerzon, M. A. 1973 *Periphony: With-height Sound Reproduction* Journal of the Audio Engineering Society, Vol. 21 No. 1 Jan/Feb pp.2-10

Hollerweger, F. 2006. *Periphonic Sound Spatialization in Multi-User Virtual Environments*. MSc. thesis, Institute of Electronic Music and Acoustics (IEM), March 2006.

Malham, D.G. 1999. Higher order Ambisonic systems for the spatialisation of sound. *Proceedings, ICMC99,* Beijing, October 1999.

Malham, D. G. 2003. *Space in Music - Music in Space*. MPhil thesis, University of York, April 2003

Pope, S. T., Ramakrishnan, C. 2003. The CREATE Signal Library ("Sizzle"): Design, Issues, and Applications. *In Proceedings of the 2003 International Computer Music Conference* (ICMC '03).

Pope, S. T. 2005. Audio in the UCSB CNSI AlloSphere. Online: http://www.create.ucsb.edu/~stp/AlloSphereAudio_01.pdf. Retrieved July 2006.

Puckette, M. 1988. The Patcher. *Proceedings of the 14th International Computer Music Conference,* Koln.

Pulkki, V. 1997. Virtual sound source positioning using Vector Base Amplitude Panning. *AES Journal*, 45(6), pp456 – 466.

Zicarelli, D. 1998. An Extensible Real-Time Signal Processing Environment for Max. *Proceedings of the International Computer Music Conference*, Ann Arbor, Michigan.