Universal Ambisonic v0.9

A specification for representing ambisonic data across environments.

"**Universal Ambisonic**" is a specification for representing ambisonic data that can be applied to both file formats and audio processing software (to facilitate interoperability between software packages).

It trades flexibility (in mixed orders combinations) for consistency and useability. No metadata is required to describe the contents of a multi-channel "Universal Ambisonic" signal, and therefore all processing environments (from VSTs, to PD patches, to mono files on disk) can know exactly what each channel contains. It is easily encapsulated in file formats.

It caters for horizontal speaker arrays and full-periphony speaker arrays.

Revision

2009-03-27 Initial release.
2009-04-01 Added several sections.
2009-04-07 Released v0.9. Changed valid component combinations to include only full-periphonic or full-horiozntal. Removed section on mixed orders. Explicitely stated that software implementations must be either horizontal or full-peripony.

The current version of this document can be accessed at http://docs.google.com/Doc?id=df4dtw69_81dsgmgqc3&hl=en

Authors

This document has been crafted with input from many people on several lists. Core contributors discussed the issues on the Ambisonic Google Group. Etienne Deleflie ... (Ambi Googlers, email me to be included here (if you wish to support the contents)).

Characteristics of the scheme

- It requires no metadata (for software implementations), and only one metadata (for file format implementations).
- It scales up to 4th order for full periphony, and 5th order for horizontal only.
- It is compatible with B format.
- It is unambiguously understandable by any processing environment (VSTs, AUs, PD patches, MAxMSP, CSound, etc.)
- The artist/producer does not require knowledge of ambisonics (specifically, the orders of spherical harmonics).

Normalisation Scheme & Component sequence

The normalisation scheme is *N3DxW*. This is N3D with the zero-degree component raised to be compatible with B format's W (which makes it compatible with all existing B format software and hardware).

The component sequence is the *Gerzon Ambisonic* sequence. For each degree, we have the *cos* and *sine* components of decreasing |m|.

Note: Please see Analysis of normalisation schemes for an analysis of the strengths/weaknesses of different normalisation schemes, and the reasoning for choosing N3DxW.

Note: Please see Four Candidate Component sequences V09.pdf (by Chris Travis) for an analysis of different component sequences, and a reasoning for choosing the Gerzon Ambisonic sequence.

Distance Compensation

All distance encoding must be fixed at 1m.

Supported component combinations

The following table illustrates the only valid component combinations/sequences. The combinations in dark green are horizontal only, the combinations in light green are full periphony.

Chs	HVP	**W0**	X1	Y1	**Z1**	X20	Y20	X21	Y21	**Z22**	X30	Y30	X31	Y31	X32	Y32	**Z33**	X40	Y40	X41	Y41	X42	Y42	X43	Y43	**Z44**	X50	Y50
3	1H	**W0**	X1	Y1																								
4	1P	**W0**	X1	Y1	**Z1**																							
6	2H1P	**W0**	X1	Y1	**Z1**	X20	Y20																					
8	3H1P	**W0**	X1	Y1	**Z1**	X20	Y20				X30	Y30																
9	2P	**W0**	X1	Y1	**Z1**	X20	Y20	X21	Y21	**Z22**																		
10	4H1P	**W0**	X1	Y1	**Z1**	X20	Y20				X30	Y30						X40	Y40									
12	5H1P	**W0**	X1	Y1	**Z1**	X20	Y20				X30	Y30						X40	Y40								X50	Y50
16	3P	**W0**	X1	Y1	**Z1**	X20	Y20	X21	Y21	**Z22**	X30	Y30	X31	Y31	X32	Y32	**Z33**											
25	4P	**W0**	X1	Y1	**Z1**	X20	Y20	X21	Y21	**Z22**	X30	Y30	X31	Y31	X32	Y32	**Z33**	X40	Y40	X41	Y41	X42	Y42	X43	Y43	**Z44**		

Core Compromise

It is the *channel count that communicates exactly what order/components are being used*. This comes at the loss of arbitrary combinations of mixed orders. This is the core compromise. The main gain is that there is no need for any metadata to describe which channel is which component, and the format therefore remains unambiguous in software processing environments.

Mixed orders bring a lot of complexity to ambisonics (from an end-user's useability perspective, to file format representations, to decoder design). By restricting the use of mixed orders, much is gained without losing the principle advantage of mixed orders (being smaller channel counts for horizontal speaker arrays).

Software Implementation

Implementation of "Universal Ambisonic" software / file formats *must implement all of the following points:*

Horizontal only plugins

1. All the above (horizontal only) channel combinations must be implemented (i.e. up to 5th order). This represents the following channel counts: 3, 6, 8, 10, 12.
2. All software must always preserve the order of the incoming ambisonic signal (unless it is a generator, in which case the user must choose the 'order' (see section 'Useage' below)).

Full-periphony plugins

1. The above horizontal only channel combinations must be implemented (i.e. up to 4th order). This represents the following channel counts: 4, 9, 16, 25.
2. All software must always preserve the order of the incoming ambisonic signal (unless it is a generator, in which case the user must choose the 'order' (see section 'Useage' below)).

Exception: When a plugin architecture does not have the capacity to dynamically 'switch' between one channel count and an other (eg. 3 to 6 output channels) then each order can be implemented in separate plugins (this is the case for VST2 hosts, but not VST3 hosts (ed. please confirm)).

Note: the term 'plugins' refers to audio digital signal processing plugin architectures such as VSTs/AUs, PD or MaxMSP patches, Bidule and Audiomulch abstractions, CSound opcodes, SuperCollider objects, Jack clients, etc.

File Implementation

File implementations of "Universal Ambisonic" *must implement the following points:*

1. File header must define that the contents are "Universal Ambisonic". This should be done with "UniversalAmbisonic = 1".
2. The file must not contain more than 25 channels.

Useage

Universal Ambisonic is aimed at users/producers who do not necessarily have intimate knowledge of Ambisonics. As such, the choice of horizontal or full-periphony plugins and the order to be used must be established using simple criteria.

To establish the useage of horizontal or full-periphony plugins, the user might be asked:

> Would you like to include height information?

To establish the order, the user could be asked:

> How many speakers is your main target array?

If the user is 'creating' an ambisonic piece, they only need to choose the 'resolution' (read: order) at the beginning of a processing chain (chain of VSTs or whichever plugin architecture is being used). From that point on, every processor on the chain (eg, every VST) knows which order they are operating on by the count of channels coming in, and it is their duty to then process at that level (i.e. maintain the order).

Labelling/Identification

Software implementations of "Universal Ambisonic" should be clearly labelled as such.