Higher Order Ambisonics for SuperCollider

Florian Grond
Input Device and
Music Interaction Laboratory, McGill
555, Sherbrooke W
H3A 1E3 Montreal CANDA,
floriangrond@gmail.com

Pierre Lecomte
LMSSC,
2 Rue Conté
75003 Paris,
FRANCE,
pierre.lecomte@gadz.org

Abstract

In this paper we present a library for 3D Higher Order Ambisonics (HOA) for the SuperCollider (SC) sound programming environment. The library contains plugins for all standard operations in a typical Ambisonics signal-flow: encoding, transforming and decoding up to the 5th order. Carefully designed PseudoUgens are the interface to those plugins to aim for the best possible code flexibility and code reusability. As a key feature, the implementation is designed to handle the higher order B-format as a channel array and to obey the channel expansion paradigm in order to take advantage of the powerful scripting possibilities of SC. The design of the library and its components is described in details. Moreover, some examples are given for how to built flexible HOA processing chains with the use of node proxies.

Keywords

SuperCollider, Higher Order Ambisonics.

1 Introduction

Ambisonics, i.e. the description of sound pressure fields through spherical harmonics decomposition, has been around for quite a while since its invention by [Gerzon, 1973]. Back in the days, the harmonics decomposition was up to the first order (First Order Ambisonics, FOA), using the 4-channel B-format[1] The playback of FOA audio content depended on special hardware and did not make it into mainstream audio in the first decade of its existence for various reasons, one of which being that FOA offers only limited spatial resolution.

Ambisonics research made significant advances in the 2000's through the work of Bamford [Bamford, 1995], Malham [Malham, 1999] and Daniel [Daniel, 2000], who extended the sound pressure field decomposition to higher orders hence the term (HOA). HOA increases the spatial resolution and thereby reduces the limitation of low spatial definition when compared with other spatialization techniques.

For streamlining and standardising content production, one hurdle that HOA was facing in the past was the coexistence of various channel ordering and normalization conventions. In order to address this issue, the *Ambix* standard was proposed by [Nachbar et al., 2011] and is ever since increasingly adopted by recent HOA implementations.

Today, processors can handle with ease multiple instances of multi-channel sound processes. Further, the rise of video games and Virtual Reality (VR) applications has elicited new interest in Ambisonics amongst audio researchers and content creators. This is mostly due to its inherent property to yield easy-to-manipulate isotropic 360 degree sound pressure fields, which can be rendered either through multi loudspeaker arrays or headphones. In the case of VR applications head-tracking is already available and the listener is always in the sweet spot. For the capturing of HOA 3D sound pressure fields, various microphone array prototypes have been developed some of them being available as commercial products like mhacoustic's Eigenmike® for instance. As far as multi loudspeaker reproduction is concerned, the number of loudspeaker domes with semi spherical configurations is growing and electroacoustic composers have also shown increasing interest in HOA as a spatialisation technique, notably amongst them for composition and [Barrett, 2010] and sonification [Barrett, 2016].

1.1 Ambisonics in various platforms

Over the last few years HOA has seen various implementations in diverse sound software environments, mostly as plugins in DAWs. The *Ambisonics Studio* plugins by Daniel Courville, for instance, have been around for some time[2].

[1] The term B-format is often used for FOA signals, in this paper we use the term also for higher orders.

[2] `http://www.radio.uqam.ca/ambisonic`

Another recent and very comprehensive example its the *Ambix plugin suite* from [Kronlachner, 2013] also see [Kronlachner, 2014a]. For Pure Data and MaxMSP, HOA libraries have been made available by the *Centre de recherche Informatique et Création Musicale*[3] an early implementation can also be found with the ICST Ambisonic tools [Schacher, 2010]. An early version for Pure Data can be found in the collection of abstractions called *CubeMixer* by [Musil et al., 2003]. Recently, the HOA library*Ambitools*[4] developed mainly in Faust has been made available [Lecomte and Gauthier, 2015].

1.2 SuperCollider

The audio synthesis environment SuperCollider (SC) by [McCartney, 2002] is particularly well suited for the creation of dynamic audio scenes. SC is split into two parts: The server *scsynth* for efficient sound synthesis and *sclang*, an object oriented programming language for the flexible configuration and re-patching of DSP trees on the server. Similar to most sound programming environments, synthesis is based in SC on unit generators called *Ugens*. Third party Ugens are collected separately in *SC3plugins*. Extensions to sclang are mananged through *Quarks*. Ugens can be assembled to more complex arrangements through synthesis definitions, known as *SynthDefs*, which are executable binaries for synthesis in scsynth. In sclang, *PseudoUgens* can be created, which is another way of handling complex arrangements of Ugens in sclang, which are compiled for scsynth, when needed. For a detailed introduction to SC see [Valle, 2016] and the SuperCollider book[5] by [Wilson et al., 2011].

1.3 Ambisonics in SuperCollider

In 2005, Frauenberger et al. implemented HOA in SC as the AmbIEM Quark[6]. This implementation goes up to the 3rd order, and follows the old Furse Malham channel ordering and normalization. All unit generators (Ugens) like encoding, rotation, and simple decoding are implement in sclang as PseudoUgens. AmbIEM comes with an simulation of early reflections in a virtual room but lacks functionality such as beamforming. The Ambisonics Toolkit (Atk) for SC by [Anderson and Parmenter, 2012] is a more recent and very comprehensive set of tools. The Atk includes for instance various transformations to manipulate the directivity of the sound field such as pushing and zooming. It is however only a first order implementation of Ambisonics.

1.4 Library design in SuperCollider

In this context the paper presents a modern HOA implementation for SC, which is modular and adopts all established standards in terms of channel ordering and normalizations. Inspired by the approach found in the Atk and typical for the general design of SC, computationally intensive parts like Ugens are split from PseudoUgens convenience wrapper classes of sclang. The HOA library comes hence in three parts, SC3plugins, PseudoUgens and audiocontent plus HRTFs for binaural rendering in a support directory.

1.4.1 SC3plugins

The first part of the library is a collection of Ugens, which is part of the SC3plugins collection. Each Ugen is compiled from C++ code. It consists of a SC language side representation of the Ugen as a *.sc* class file and a *.scx*, *.so* or *.dll* compiled dynamic link library, for the platforms (OSX, Linux or Windows) respectively. For each Ambisonics order (so far up to order 5), there are individual Ugens for the encoding, transforming and decoding processes in a typical Ambisonics signal flow. The C++ code for these Ugens is generated from the HOA library *Ambitools* [Lecomte and Gauthier, 2015] with the compilation tool *faust2supercollider*. This approach was taken for two reasons:

First, to leverage the work already accomplished in Faust. Indeed, the Faust compiler generates very efficient DSP code and the Faust code base allows to efficiently combine existing functionality. The meta approach through Faust will lead to future additions of functionality, which can then be easily integrated in the HOA library for SC.

Second, each Ambisonics order comes with a defined multichannel B-format, this in turn defines the amount of input and output arguments for the Ugens. For instance, a Ugen rotating an Ambisonic signal of order 3 has 16 input arguments plus the rotation angles and 16 output channels. While it is of interest to expose the Ambisonics order as an argument for the flexibility and reusability of code on the side of *sclang*, it is an argument unlikely to be changed while

[3] `http://www.mshparisnord.fr/hoalibrary/en`

[4] `http://faust.grame.fr/news/2016/10/17/Faust-Awards-2016.html`

[5] `http://supercolliderbook.net/`

[6] `https://github.com/supercollider-quarks/AmbIEM`

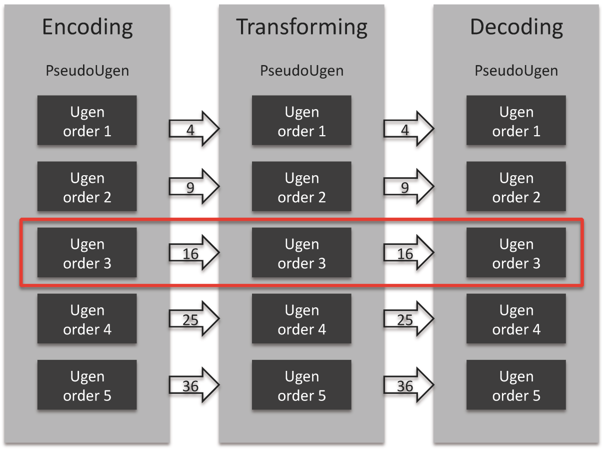

Figure 1: The Ambisonics processing chain in the HOA library for the selected order 3 with 16 channels.

an instance of the Ugen is running as a node in the DSP tree on *scsynth*. This is why for every order there is a unique Ugen for each function (encoding, transforming, decoding).

The Ugens follow these conventions, some of which are explained in subsequent sections):

- The Ambisonics channels are ordered according to the ACN convention.
- The default normalisation of the B-format is N3D.
- All azimuth and elevation arguments follow the spherical coordinates convention from SC.
- Operations of resource intensive Ugens can by bypassed.

Based on the implementation in Faust, the main functionalities of the HOA Ugens provided as the SC3plugins are so far:

- Encoding and decoding of planar waves and spherical waves using near field filters.
- Mirroring, Rotation (around azimuth and full 3D).
- Various Ugens for beamforming, returning mono as well as B-format signals.
- Various decoders in conjunction with Head-Related Impulse Responses (HRIRs) for binaural monitoring.

1.4.2 PseudoUgens

The second part of the library is available as the sclang extension HOA Quark. While the SC3plugins are designed for computational efficiency of the sound synthesis processes, The HOA Quark is conceived to unlock the flexibility of making sound in SC with respect to code reusability and the scaling of synthesis scripts. Each typical operation in Ambisonics (Encoding, Transforming, Decoding) is here provided as a PseudoUgen. Depending on the Ambisonics order provided as an argument, the PseudoUgen returns and instantiates the correct Ugen from the SC3plugins collection on the sound server. Since the Ambisonics order is an argument for the PseudoUgen, the number of channels in the B-format vary and so does the number of input arguments in the UGens. This is why the B-format is handled as a channel array. This makes the SC code flexible for experimentations with different orders depending on computational resources. All arguments of the PseudoUgen obey the channel expansion paradigm. This means that if any of the arguments is an array (or an array of arrays), the PseudoUgen returns an array (or an array of arrays) of Ugens.

Figure 1 shows the relation between the SC language side (PseudoUgens in light grey) and the SC3plugins (Ugens dark grey). If the Ambisonics order is set to 3 and passed as an argument to the PseudUgen, the corresponding Ugen with 16 channels is returned and a typical processing chain (Encoding Transforming Decoding, encircled in red) can be established. The main features of the design of the HOA library implementation on the language side are:

- B-format is handled as a channel array.
- All arguments obey the channel expansion paradigm.

This leads to the following advantage, when scripting HOA sound scenes in SC. Compared with graphical data flow programs like for instance Pure Data, changing the order means to reconnect all channels between objects interfacing with the B-format. In SC changing the Ambisonics order in a single global variable changes the order of the whole HOA processing chain.

1.4.3 Support directory

The third part of the library is a platform independent support directory for various HOA sound file recordings and convolutions kernels from HRIRs for the binaural rendering of HOA sound scenes. The support directory approach is similar to the Atk implementation from which we adapted the corresponding class. The reason to keep these resources separate from the Quark directory is mostly due to the size of the included sound files. We provide some 4th order HOA sound files that have been recorded with the Eigenmike® with the support of Romain Dumoulin from CIRMMT. The HRIRs provided are either measured from a KU-100 dummy head [Bernschuetz, 2013] or computed from a 3D mesh scan of several people's face. The directions of the HRIRs follow a 50-node Lebedev grid, allowing an Ambisonic binaural rendering up to order 5 [Lecomte et al., 2016b].

2 Encoding

As the first step in an Ambisonics rendering chain, the library provides PseudoUgens for encoding into the B-format. One for the encoding of mono sound signals, one for microphone array prototypes and one for the commercially available Eigenmike® microphone array.

2.1 HOAEncoder

This PseudoUgen creates an HOA scene from mono inputs encoded as a (possibly moving) sound source in space. The source can be encoded 1) as a plane wave with azimuth and elevation (θ_p, δ_p) respectively 2) as a spherical wave with position $(r_s, \theta_s, \delta_s)$, where r_s is the distance to origin of the source. The spherical wave is encoded using near-field filters [Daniel, 2003]. In the current implementation, those filters are stabilized with near-field compensation filters. Thus, in this case, the radius of the loudspeaker layout r_{spk} used for decoding is needed. Note that if the spherical source radius is such as the source is focused inside the loudspeaker enclosure $(r_s \leq r_{spk})$, a "bass-boost" effect may occur with potential excessive loudspeaker gain. This effect increases as the source get closer to the origin [Daniel, 2003] [Lecomte and Gauthier, 2015].

```
HOAEncoder.ar(1,SinOsc.ar(f),a,e)
    ;
// returns
[OutputProxy,...,...,OutputProxy]

HOAEncoder.ar(1,
        SinOsc.ar([f1,f2]),
        a,e); // returns
[[OutputProxy,...,...,OutputProxy],
 [OutputProxy,...,...,OutputProxy]]
```

```
HOAEncoder.ar(1,
        SinOsc.ar([f1,f2]),
        a,e).sum;// returns
[OutputProxy,...,...,OutputProxy]
```

If an array of azimuth and elevation arguments, matching in size those of the source $SinOsc.ar([f1, f2])$, flexible and scalable code for multi source encoding can be created.

2.2 HOAEncLebedev06 / 26 / 50, HOAEncEigenmike

This collection of PseudoUgen offers at first the Discrete Spherical Fourier Transform (DSFT) for various spherical layout of rigid spherical microphone. In the current implementation the proposed geometries are 06- 26- or 50-node Lebedev grid [Lecomte et al., 2016b] and Eigen-Mike grid[Elko et al., 2009]. The components of the DSFT are then filtered to take into account the diffraction by the rigid sphere and retrieve the Ambisonic components [Moreau et al., 2006] [Lecomte et al., 2015] The filters are applied by setting the filter flag to 1 as shown in the next code listing:

```
// Encode the signals from the
// Lebedev26 grid microphone
HOAEncLebedev26.loadRadialFilters
   (s);
{HOAEncLebedev26.ar(4, SoundIn.ar
   (0!26), filters: 1)}.play
```

3 Converting

In order to correctly reconstruct a sound field from the channels of the B-format, it is important to know about standard normalization methods for the spherical harmonic components, as well as channel ordering conventions. Two main channel ordering conventions exist: The original *Furse-Malham* (FuMa) [Malham, 1999] higher-order format, an extension of traditional first order B-format up to third order (16 channels). FuMa channel ordering comes with maxN normalization, which guarantee maximum amplitude of 1. The FuMa format has been widely used and is still in use but is increasingly replaced by the *Ambisonic Channel Number* (ACN) ordering [Nachbar et al., 2011]. ACN typically comes with (the full three-D normalisation) where all signals are orthonormal. SN3D (Semi-Normalized 3D) spherical harmonics. This normalization has the advantage that none of the higher order signals exceeds the level of the first Ambisonic channel, W (ACN 0).

However, this normalization does not provide an orthonormal basis of spherical harmonics and this latter case is recommended for transformations which rely on the orthormality property of spherical harmonics. Therefore, the library uses internally the N3D (full 3D normalization) with ACN convention.

3.1 HOAConvert

The HOAConvert PseudoUgen accepts a B-format array as input and converts from and to ACN_N3D, ACN_SN3D, FuMa_MaxN. It is mostly meant to convert existing B-format recordings into ACN N3D for use within the library The other use case is to render B-format mixes to other conventions for other production contexts.

4 Transforming

In its current implementation, the HOA library provides 3 standard operations like rotation and mirroring to transform the B-format.

4.1 HOAAzimuthRotator

This PseudoUgen rotates the HOA scene around the z-axis, which is accomplished with a rotation matrix in x and y due to the symmetry in z of the spherical harmonics. For the matrix definition see [Kronlachner, 2014b]. In combination with horizontal head tracking, this transformation can stabilise horizontal auditory cues for left-right movements when the rendering is made over headphones in VR contexts.

4.2 HOAMirror

This PseudoUgen mirrors an HOA scene at the origin in the directions along the axes left-right (y), front-back (x), up-down(z). According to [Kronlachner, 2014b], this can be accomplished by changing a the sign of selected spherical harmonics.

4.3 HOARotatorXYZ

This PseudoUgen rotates a HOA scene around any given angle around x,y,z. The rotation matrix is computed in spherical harmonic domains using recurrence formulas [Ivanic and Ruedenberg, 1996].

5 Beamforming

5.1 HOAHCard2Mono

This PseudoUgen extract a mono signal from the HOA scene according to a beampatern. The channels from the B-format inputs are combined to produce a monophonic output as if a directional microphone was used to listen into a specific direction in the sound field. In the current implementation, the beampatern provided are regular hypercardioids up to order 5 see [Meyer and Elko, 2002]

5.2 HOAHCard2HOA

This PseudoUgen applies a hyper-cardioid beam-pattern to the HOA scene to enhance some directions and outputs a directional filtered HOA scene [Lecomte et al., 2016a]. The proposed beam-patterns are regular hypercardioids as described in [Meyer and Elko, 2002]. The selectivity of the directional filtering increases with the order of the beam-pattern. This transformation requires an order re-expansion such that the output HOA scene should be of the order of the input HOA scene plus the beam-pattern order [Lecomte et al., 2016a].

5.3 HOADirac2HOA

As in the previous section, this PseudoUgens performs a directional filtering on the HOA scene but this time the beam-pattern is a directional Dirac, that is to say a function which is zero everywhere except in the chosen direction. As a result the output HOA scene contains only the sound from the chosen direction. Thus, this tools helps to explore the HOA scene with a "laser beam". For more details see [Lecomte et al., 2016a].

6 Decoding

For the decoding of HOA signals two different ways of rendering the sound field are possible: First via headphones, or second through a setup of multiple loudspeakers.

For the headphone option the HOA signal is decoded to spherically distributed virtual speakers. For the best possible spatial resolution more speakers are needed then there are channels in the B-format. Each speaker signal is then convolved with HRTFs and the resulting left and right channels are summed respectively. For the distribution of the virtual speakers a regular distribution on the sphere is desirable, so that the decoding matrix is well behaved. This is why according to [Lecomte et al., 2015] and similar to the microphone array prototypes from above a Lebedev grid is chosen.

6.1 HOADecLebedev06 / 26 / 50

This collection of PseudoUgen decodes an Ambisonics signal up to 50 virtual speakers positioned as nodes on a Lebedev grid. The decoding on the 50-node Lebedev grid works up to order 5. This grids contains two several nested sub-grids which work up to lower order with less nodes[Lecomte et al., 2016b]. Therefore, the 6 first nodes are sufficient for first order and the 26 first nodes are sufficient up to the third order. If the HRTF filter flag is set to 1, the signals are convolved with the kernels and summed up to yield a left and right headphone speaker signal. Prior to this, the convolution kernels need to be loaded to the sound server as shown in the next code listing:

```
// load a HOA sound file
~file=Buffer.read(s,"hoa30.wav");
// prepare binaural filters
HOADecLebedev26.loadHrirFilters()
{HOADecLebedev26.ar(3,//order 3
   PlayBuf.ar(16,~file,1,loop:1),
   hrir_Filters:1)
}.play;
```

6.2 HOADec

For the case of decoding for speaker arrays [Heller et al., 2008] distinguish 3 cases:

1. regular polygons (square, octagon) and polyhedra (cube, octahedron)
2. semiregular arrays (non equidistant but opposing speakers, like in a shoebox)
3. general irregular arrays (e.g. ITU 5.1, 7.1 ... semispherical speaker domes)

For the cases 1 and 2, decoder matrices can be obtained by matrix inversion. If, depending on the positions of the speakers, the resulting decoder matrix has elements are of similar magnitudes, it is suitable for signal processing. For case 3, which are arguably the more realistic cases, a variety of state of the art techniques exists, see for instance [Zotter et al., 2012], [Zotter et al., 2010], and [Zotter and Frank, 2012]. An implememtation of these techniques exceeds the scope of this library. However, for the construction of decoders for specific irregular speaker arrays, we refer the user to the Ambisonic Decoder Toolbox by Aaron J. Heller[7]. This toolbox produces decoders as Faust files, which can be compiled online[8] as Ugens and in turn can then be integrated in the HOADec PseudoUgen class template.

7 The distance of sound sources

One novel aspect of the underlying Faust implmentation of Ambitools is the spherical encoding of sound sources using near field filters. For the correct reproduction of the HOA scene, the distance of the sound source and the radius of the reproducing (virtual) speaker array needs to be set. The correct near field filters are either applied by setting it in the encoding or in the decoding step.

```
// load the binaural filters
HOADecLebedev26.loadHrirFilters()
{ var src;
src=HOAEncoder.ar(
   3,//order
   PinkNoise.ar(0.1),//source
   az,//azimuth
   ele,//elevation
   plane_spherical:1,
   radius:2,
// set the speaker radius here
   speaker_radius:1)

HOADecLebedev26.ar(
   3,//order
   src,//source
// or set the speaker radius here
// speaker_radius:1,
   hrir_Filters:1)
}.play;
```

8 HOA and SynthDefs

The use of PseudoUgens leads to one important caveat when working with SynthDefs. The Ambisonics order is an argument pertaining to the PseudoUgen, it can hence not be an argument of a SynthDef. The reason is that at compile time the Ambisonics order would remain undefined and the PseudoUgen does hence not know which Ugen to return When working with SynthDefs code reusability can still be achieved as shown in the next code listing:

```
// set the max order:
~order = 5;
~order.do({|i| // iterate
 SynthDef( //create unique names
 "hoaSin"++(i+1).asString,
```

[7]https://bitbucket.org/ambidecodertoolbox/adt.git

[8]http://faust.grame.fr/onlinecompiler/

```
 {Out.ar(0,
        HOAEncoder.ar(
         i+1,//increase the order
         SinOsc.ar())}},
).add;
})
// play the Synths
Synth(\hoaSin1);
...
Synth(\hoaSin5);
```

9 HOA and Node Proxies

For the flexible creation of typical Ambisonics render chains, Node Proxies [Rohrhuber and deCampo, 2011] provide an excellent tool in SC. Node Proxies autonomously handle audio busses and conveniently allow to crossfade between audio processes of a selected node, freeing silent process when the crossfade is completed. This allows to dynamically change sources in the encoding, transforming and decoding step in the rendering chain. The following code example shows a flexible scenario with changing seamlessly from an XYZ rotation to beamforming.

```
~o=3;
~chn=(~order+1).pow(2);
// load hoa sound file:
~bf=Buffer.read(s,"file.wav");

// b-format file player:
~player=NodeProxy(s,\audio,~chn);
~player.source=
          {PlayBuf.ar(~chn,~bf)};

// Node for xyz rotation:
~trans=NodeProxy(s,\audio,~chn);
~trans.source=
       {var in;in=\in.ar(0!16)
        HOATransRotateXYZ.ar(
               ~o,in,
               yaw,pitch,roll)};

// rotate the scene
~trans.set(\yaw, angle);

// decoding,
~dec=NodeProxy(s,\audio,~chn);
~dec.source=
       {var in;in=\in.ar(0!16)
       HOADec.ar(~o,in,)};

// chain the proxies together
~player <>> ~trans <>> ~dec;
```

```
// change rotation to beamforming

~trans.source=
       {var in;in=\in.ar(0!16)
        HOABeamDirac2Hoa.ar(
               ~o,in,
               az,ele)};

// direct the beam
~trans.set(\az, angle);
```

10 Conclusions

We have presented a HOA library for SC. The design of which resulted in great flexibility and makes it a valuable addition to experiment with HOA in various contexts. Due to the meta approach through Faust, future additions to the library are feasible and we look forward to experiment with it in the context of VR and video gaming platforms but also for the creation of sound material for electro acoustic compositions. We believe that the flexibility and live coding capacity of SC is particularly useful in the context of HOA, where repeated listening is essential to asses the perceptually complex mutual interdependence of temporal and spatial sound characteristics.

11 Acknowledgements

This work has been supported through the research-creation funding program of the Fonds de Recherche du Québec - Société et Culture (FRQSC).

References

J. Anderson and J. Parmenter. 2012. 3D sound with the Ambisonic Toolkit. In *Presented at the Audio Engineering Society 25th UK Conference / 4th International Symposium on Ambisonics and Spherical Acoustics*, York.

Jeffrey S. Bamford. 1995. *An Analysis of Ambisonic Sound Systems of First and Second Order*. Ph.D. thesis, University of Waterloo, Waterloo.

Natasha Barrett. 2010. Ambisonics and acousmatic space: a composer's framework for investigating spatial ontology. In *Proceedings of the Sixth Electroacoustic Music Studies Network Conference Shanghai*, 21-24 June.

Natasha Barrett. 2016. Interactive Spatial Sonification of Multidimensional Data for Composition and Auditory Display. *Computer Music Journal*, 40(2):47–69.

B. Bernschuetz. 2013. A spherical far field hrir/hrtf compilation of the neumann ku 100. In *in Proceedings of the 40th Italian (AIA) Annual Conference on Acoustics and the 39th German Annual Conference on Acoustics (DAGA)*, page 29.

J. Daniel. 2000. *Représentation de Champs Acoustiques, Application à la Transmission et à la Reproduction de Scènes Sonores Complexes dans un Contexte Multimédia, Ph.D. Thesis*. Ph.D. thesis, University of Paris 6, Paris, France.

Jerome Daniel. 2003. Spatial sound encoding including near field effect: Introducing distance coding filters and a viable, new ambisonic format. In *In Audio Engineering Society Conference: 23rd International Conference: Signal Processing in Audio Recording and Reproduction*, pages 1–15, Helsingor. AES.

G. Elko, R. A. Kubli, and J. Meyer. 2009. Audio system based on at least second-order eigenbeams.

Michael A. Gerzon. 1973. Periphony: With-height sound reproduction. *Journal of the Audio Engineering Society*, 21(1):2–10.

Aaron J. Heller, Richard Lee, and Eric M. Benjamin. 2008. Is My Decoder Ambisonic? In *the 125th Convention of the Audio Engineering Society*, San Francisco, oct. 1-5.

J. Ivanic and K. Ruedenberg. 1996. Rotation matrices for real spherical harmonics. Direct determination by recursion. *J. Phys. Chem.*, 100(15):6342–6347.

M. Kronlachner. 2013. Ambisonics plug-in suite for production and performance usage. In *LAC*, Graz, Austria, May.

M. Kronlachner. 2014a. Plug-in Suite for Mastering the Production and Playback in Surround Sound and Ambisonics. In *AES Student Design Competition (Gold Award)*, Berlin, April.

M Kronlachner. 2014b. Spatial transformations for the alteration of ambisonic recordings. *Graz University Of Technology, Austria*.

P. Lecomte and P.-A. Gauthier. 2015. Real-Time 3D Ambisonics using Faust, Processing, Pure Data, And OSC. In *In 15th International Conference on Digital Audio Effects (DAFx-15)*, Trondheim, Norway.

P. Lecomte, P.-A. Gauthier, C. Langrenne, A. Garcia, and A. Berry. 2015. On the use of a Lebedev grid for Ambisonics. In *in Audio Engineering Society Convention 139*.

P. Lecomte, P.-A. Gauthier, C. Langrenne, A. Berry, and A. Garcia. 2016a. Filtrage directionnel dans un scène sonore 3D par une utilisation conjointe de Beamforming et d'Ambisonie d'ordre elevés. *in CFA / VISHNO 2016*, pages 169–175.

Pierre Lecomte, Philippe-Aubert Gauthier, Christophe Langrenne, Alain Berry, and Alexandre Garcia. 2016b. A fifty-node lebedev grid and its applications to ambisonics. *Journal of the Audio Engineering Society*, 64(11):868–881.

D.G. Malham. 1999. Higher order ambisonic systems for the spatialisation of sound. In *ICMC99*, pages 1–4, Beijing. ICMC.

J. McCartney. 2002. Rethinking the Computer Music Language: SuperCollider. *Computer Music Journal*, 26(4):61–68.

J. Meyer and G. Elko. 2002. A highly scalable spherical microphone array based on an orthonormal decomposition of the soundfield. *in IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2:1781–1784.

S. Moreau, J. Daniel, and S. Bertet. 2006. 3d sound field recording with higher order ambisonics-objective measurements and validation of spherical microphone. In *in Audio Engineering Society Convention 120*, pages 1 – 24.

T. Musil, J. Zmoelnig, M. Noisternig, A. Sontacchi, and R. Hoeldrich. 2003. AMBISONIC 3D Beschallungssystem 5.Ordnung fuer PD. Report 15, Institute for Elektronic Music and Acoustics.

Christian Nachbar, Franz Zotter, Etienne Deleflie, and Alois Sontacchi. 2011. AMBIX - a suggested ambisonics format. In *Ambisonics Symposium*, Lexington, KY, June 2-3.

Julian Rohrhuber and Alberto deCampo, 2011. *The SuperCollider Book*, chapter Just in Time Programming (ch 7) . MIT Press, MA: Cambridge.

Jan C. Schacher. 2010. Seven Years of ICST Ambisonics Tools for MAXMsp - a Brief Report. In *Proc. of the 2nd International Symposium on Ambisonics and Spherical Acoustics*, Paris, France, May 6 - 7.

Andrea Valle. 2016. *Introduction to SuperCollider*. Logos Publishing House.

S. Wilson, D. Cottle, and N. Collins, editors. 2011. *The SuperCollider Book*. MIT Press, MA: Cambridge.

F. Zotter and M. Frank. 2012. All-Round Ambisonic Panning and Decoding. *J. Audio Eng Soc*, 60(10):807–820, Nov.

F. Zotter, M. Frank, and A. Sontacchi. 2010. The virtual t-design ambisonics-rig using vbap. In *presented at the 1st EAA-EuoRegio 2010 Congress on Sound and Vibration*, pages 1 – 4, Ljubljana, Slovenia.

F. Zotter, H. Pomberger, and M. Noisternig. 2012. Energy-Preserving Ambisonic Decoding. *Acta Acoustica united with Acoustica*, 98:37 – 47.