COMPOSITIONAL CONTROL OF PERIPHONIC SOUND SPATIALIZATION

*Marlon Schumacher**

IDMIL DCS CIRMMT
Schulich School of Music of McGill University
Montreal, QC, Canada
marlon@music.mcgill.ca

Jean Bresson

IRCAM - CNRS UMR STMS
Paris, France
jean.bresson@ircam.fr

ABSTRACT

OMPrisma is a library for the control of sound source spatialization and spatial sound synthesis in the computer-aided composition environment OpenMusic. In this article we describe the basic concepts underlying this library and its integration into compositional workflows ranging from high-level symbolic control structures to spatial sound synthesis and reproduction. A generic framework is presented, consisting of authoring environment, rendering engines and real-time decoding application. We particularly focus on periphonic spatialization processes and the usability of ambisonics from a compositional perspective.

1. INTRODUCTION

The advent of sound spatialization technologies has made it possible to render three-dimensional audio scenes for large speaker arrays and a multiplicity of sound sources. Many research centres and concert venues are equipped with large-scale multichannel systems and the increased availability of spatialization technologies has made the presentation of multichannel works a common practice. From a compositional viewpoint, the ever-growing variety and complexity of spatial sound rendering techniques requires the development of appropriate control strategies and musically meaningful representations in order to facilitate the integration of three-dimensional sound into compositional processes and musical language.

Although a great number of electroacoustic compositions make use of spatialization technologies, current practices rarely consider the world of computer-aided composition (CAC) yet. In fact, spatialization is often regarded as a post-processing technique or effect which is applied to already-composed material at a later stage of work.

OMPrisma is a library for the control of sound spatialization created for the computer-aided composition environment OpenMusic (OM) [1]. OM is a visual programming language, allowing composers to develop musical processes and generate musical material via visual (or textual) programming. The programming environment includes a number of libraries and packages dedicated to various musical domains such as rhythmic structures, constraint satisfaction, sound analysis, and many more. One of these packages is OMChroma, an object-oriented framework for the control of sound synthesis [2]. OMPrisma extends this framework into the spatial domain, providing a library of spatial sound rendering classes and a collection of tools for flexible and ergonomic control of spatialization processes.

*This work has been supported by the Centre for Interdisciplinary Research in Music Media and Technology (CIRMMT)

We tackle the issue of musical control of sound spatialization from the perspective of computer-aided composition, by integrating symbolic, sound and spatialization data within a single high-level, programmable environment in order to facilitate the development of structural relationships between spatialization and other musical (or extra-musical) dimensions.

The Csound language [3] is invoked as underlying DSP engine, and a standalone application (entitled *MultiPlayer*) has been developed for real-time decoding and diffusion of the rendered multichannel audio formats. OMPrisma thus implements a generic framework constituted by three complementary components, each dedicated to a specific task-domain as shown in Figure 1.

2. THE OMPRISMA LIBRARY

The OMPrisma library provides an extensible set of OM classes (in terms of object-oriented programming) implementing a range of spatial sound rendering techniques. Each class is associated to a digital signal processing (DSP) patch (or "instrument") written in Csound, and embeds a matrix data structure, whose rows correspond to vectors of input parameters for this instrument. Columns ("components" in the terminology of OMChroma) correspond to different instances of the same instrument. These classes constitute the primary control interfaces which can be instantiated in the form of iconic representations within OM's visual programs.

2.1. Spatial Sound Rendering in Csound

The spatialization techniques currently implemented in OMPrisma include conventional stereo and quadraphonic panning, Vector Base Amplitude Panning (VBAP), Higher-Order Ambisonics, Reverberated VBAP and a mixed-order Ambisonics system with optional simulation of room-acoustics. Two classes for ambisonic spatialization allow up to third-order encoding, based on the Csound opcodes *bformenc1* and *spat3D*.[1]

The library of Csound instruments in OMPrisma is based on a modular DSP architecture. The ambisonics instruments include a pre-processing module to increase the impression of distance and motion of a sound source. The effect of air-absorption is accounted for with a 2nd order butterworth lowpass filter, doppler-effects are simulated using a moving write-head delay-line and the decrease in amplitude as a function of distance is accomplished with a simple gain-stage unit.

[1] Please refer to http://www.csounds.com/manual/html/index.html for documentation.

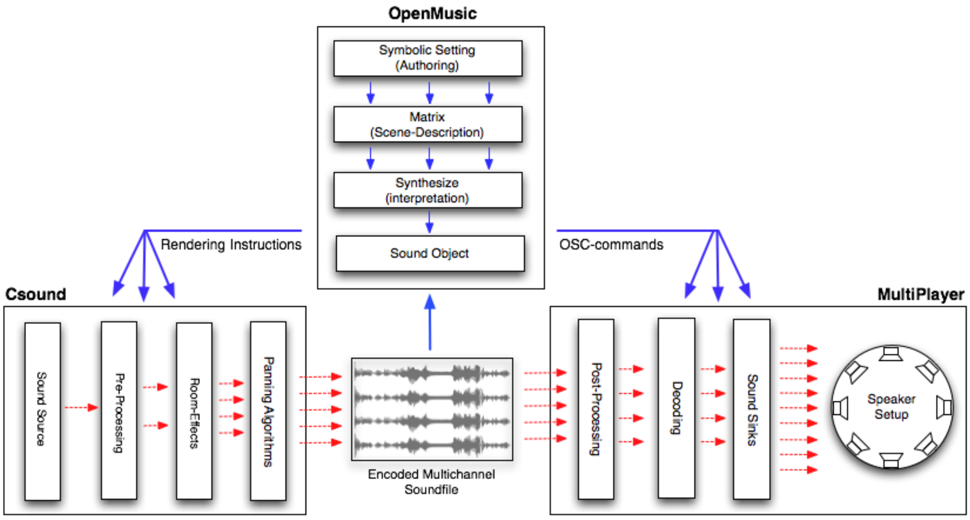

Figure 1: The OMPrisma framework. Authoring in OpenMusic, rendering in Csound and reproduction with the MultiPlayer.

2.2. Control Structures

OMPrisma separates the description of a spatial sound scene from its rendering and reproduction. All spatial sound rendering classes in OMPrisma share a common control interface in compliance with current SpatDIF specifications [4], allowing a user to describe spatialization processes from an abstract level and render it into a multichannel audio format using one of the available spatialization techniques. This offers the opportunity to rapidly interchange and compare spatial sound rendering techniques, such as Vector Base Amplitude Panning or Higher-Order Ambisonics, which is an advantage both from a practical point of view and in terms of developing a better understanding of the qualities of different techniques for periphonic spatialization.

The number of columns within a matrix, i.e. the number of synthesis components which are dynamically allocated during the sound synthesis process, is virtually unlimited. Thus, complex spatial sound scenes with arbitrary numbers of sound sources can be described and stored in a compact, yet flexible representation, and eventually rendered into a multichannel format in high-end audio quality via Csound. An identical copy of the DSP engine is allocated for each sound source in the synthesis process, allowing the individual control of the full range of synthesis parameters for each sound source (such as attenuation-laws or trajectories, but also individual room characteristics).

Dynamically changing parameters are specified in OM as break-point functions which are internally converted into wave tables read by high-precision oscillators. The representation of spatialization parameters as audio-signals provides sample-accurate control and the tightest possible coupling of sound synthesis and spatialization parameters.

In order to control the different source pre-processing units, a set of functions is available, implementing a number of common expressions for distance attenuation, air absorption and Doppler shift as a function of a sound source's distance. Rather than hard-wiring specific expressions into the spatialization engines directly, we implemented a table-lookup system, for greater flexibility and efficiency. This way a user may define her own equations (directly in Lisp or as graphical programs) to be used inside the spatial sound renderer. Moreover, tables can be manually edited using the graphical break-point function editors in OM. The tables can then either be connected to the corresponding slot of a class directly (to be applied to individual sound sources), or provided as global tables for the whole synthesis process.

2.3. Sound Spatialization in OpenMusic

One of the main interests of the OMPrisma framework is its integration into a general compositional environment (OM). This embedding leads to novel possibilities for integrating symbolic, sound and spatialization data into compositional frameworks and offers a degree of control difficult to achieve using conventional tools.

Several studies have revealed a great variety of compositional approaches and models for the musical use of spatialization, see for example [5]. Since the aesthetic aims of a composer can't be anticipated, a system for generalized control of spatial sound synthesis should be as open while at the same time as configurable as possible [2]. Hence, spatial parameters should be considered as musical materials independent of a concrete spatial sound scene, and the spatialization process conceptualized in terms of abstract musical objects, higher-level functional processes and relationships, which are eventually converted into low-level instructions at the synthesis stage.

Accordingly, OMPrisma employs generic OM objects, such as break point functions and break point curves as abstract data which are interpreted and converted according to a given synthesis context. Trajectories, for example are represented as autonomous objects in the form of geometric configurations (e.g. 3-dimensional break-point functions), which can be algorithmically processed or manually edited, and finally 'applied' to a concrete synthesis process. For temporal control of trajectories, frequency-envelopes can be provided (for each Cartesian dimension separately), allowing to dynamically change the speed-of-travel of a sound source along its trajectory (including stopping or reversing the travel direction), create spatial patterns (e.g. lissajous figures), or work with audio-rate oscillations and frequency modulations at the border between sound synthesis and spatialization.

Figure 2 shows an example of a sound spatialization process in OM using OMPrisma. The class *ambi.3D.continuous* is used to spatialize a monaural sound file and render a third-order B-format file. The spatial trajectory of the sound source is computed from a *3DC* (3D Curve—top of the figure), an OM object for representation and manipulation of three-dimensional breakpoint functions. The function *gen-trajectory* converts the 3DC into envelopes for the respective Cartesian dimensions (*x-position*, *y-position*, *z-position*).

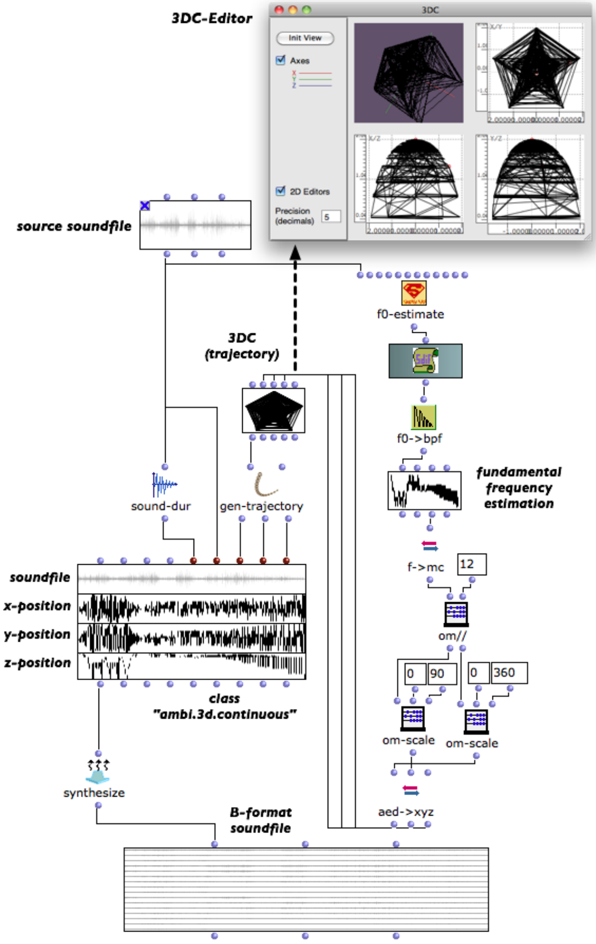

Figure 2: A 3-dimensional spatialization process carried out in OMPrisma. Control parameters are derived from an analysis (fundamental frequency estimation) of the source sound file itself. The extracted pitch profile is mapped to azimuth and elevation angles.

The 3DC in Figure 2 is calculated from external data derived from the fundamental frequency estimation of the source sound file (see the right-hand part of the figure). This data is then mapped to spatial positions in a hemispheric volume via geometrical operations. This is, therefore, an illustrative example of how spatialization parameters can be considered as symbolic (compositional) objects related to external processes developed in the computer-aided composition environment. (This technique could also be considered an *auto-adaptive digital audio effect* [6], as the control data is derived from sound features using specific mapping functions.)

2.4. Post-Processing and Rule-Based Transformations

The synthesis procedures inherited from the OMchroma system allow to eventually process and modify the matrix contents before parsing the data into a Csound score and starting the synthesis process. This feature has proven to be a powerful tool allowing a user to specify and apply arbitrary rule-based manipulations of entire spatial sound scenes.

Possible applications include rotations/translation of spatial sound scenes, the implementation of control extensions, such as "W-panning" [7] (to control the apparent size of a sound source), or the generation of composite sound sources from point sources, as described for example in [8]. Such post-processing rules, implemented as functions or visual programs (patches) in OM, can be attached to matrices for processing of spatial sound scenes without the need to modify the original control structure nor the underlying rendering engine.

Figure 3 shows an OM patch implementing W-panning. An additional control parameter is provided to the matrix ("width", at the very right), which is used in the W-panning function (box on the left, whose content is not visible on the figure). The data contained in the matrix is internally processed by the W-panning function, using spatial positions (already contained in the matrix data) and the "width" value list, in order to set the ambisonic encoding order as a function of distance and width.

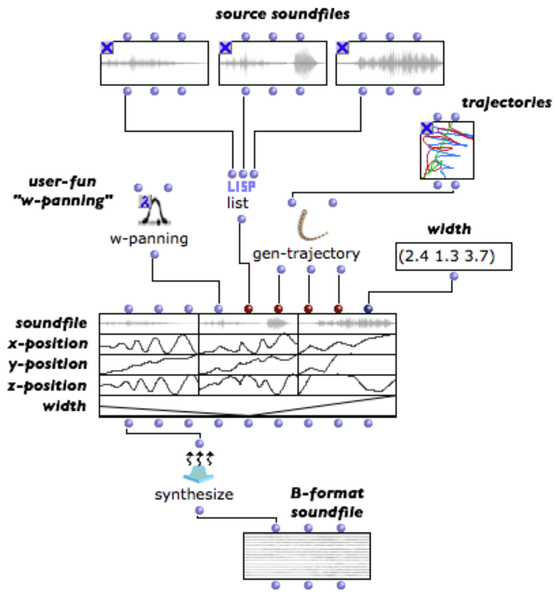

Figure 3: W-panning realized in OMPrisma

3. SPATIAL SOUND SYNTHESIS

The common underlying structure of OMPrisma and OMChroma classes permits the definition of hybrid objects by combining sound synthesis units with spatial sound renderers into spatial sound synthesizers. The same concepts and spatialization processes used for complex sound sources can then be applied to arbitrary numbers of micro-level sound components, allowing to synthesize sounds with complex spatial morphologies.

An original class-merging procedure has been implemented which dynamically combines OMChroma and OM-

Prisma classes into compound classes for spatial sound synthesis. A parsing-algorithm inspects the corresponding Csound instruments and generates a new instrument in which the output signals from the synthesis instrument are connected to the input of the spatial sound renderer. The resulting DSP unit's control parameters are inherited from the original sound synthesis and spatialization units and a new class is internally defined and associated to the generated Csound instrument.

From a user's perspective, this is accomplished by simply connecting any OMChroma synthesis class to any OMPrisma spatial sound rendering class using the function *chroma-prisma* as shown in Figure 4 below.

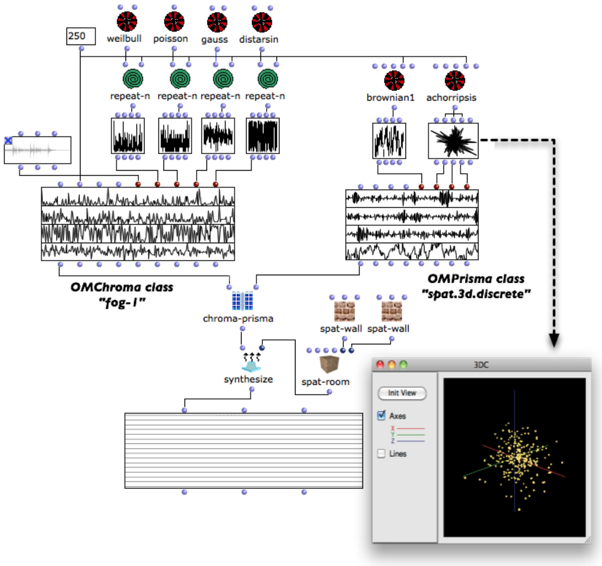

Figure 4: A spatial sound synthesis process carried out by merging the OMChroma class *fog-1* (for granular synthesis) with the OMPrisma class *spat.3d.discrete* for ambisonic spatialization. Both sound synthesis and spatialization parameters are controlled via stochastic processes.

Each synthesis component (e.g. sinusoid, formantic waveform, grain, etc.) is therefore associated to a set of spatialization parameters and can be individually controlled. Hence, OMChroma and OMPrisma constitute 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/spatialization technique.

4. CONSIDERATIONS ON THE USE OF AMBISONICS IN COMPUTER-AIDED COMPOSITION

Ambisonics separates encoding and decoding of a spatial sound scene into different stages, which makes it particularly interesting in the context of CAC:

A composer can focus on the creative process and render both sounds together with directional information into an intermediate representation (the B-format), and dedicate herself to issues of reproduction at a later stage. In addition, since ambisonics encodes a soundfield and not the sound sources themselves, the information required to reproduce a spatial sound scene (i.e. the size of a B-format file) does not grow as a function of its content, but its spatial resolution, which lends itself particularly well to complex spatialization processes invoking large numbers of sound sources, such as spatial sound synthesis (see Section 3).

Another interesting property of ambisonics is the ability of processing an encoded soundfield in the spherical-harmonics domain in computationally efficient ways, sometimes referred to as 'soundfield effects'. Typical manipulations include rotation and dominance (also known as 'zoom' or 'acoustical lense'), but also synthesis of discrete reflections (FDNs) and reverberation (see for example [9, 10]).

Since the decoding of the spatial sound scene can be performed in he final performance venue, soundfield effects can be applied in order to adapt the encoded material to the technical setup and acoustic properties. It is also possible to tweak the reproduction of the soundfield by changing the weightings of the spherical components of the loudspeaker signals, which allows for example adjusting decoder-settings after having observed the distribution of an audience in a performance space (e.g. trading off spatial accuracy vs. increased size of sweetspot).

However, this flexibility also introduces usability issues: Since there is no standard format for encoding ambisonic files yet, the 'Metadata' (such as the ordering of spherical components in a B-format file, encoding flavours, etc.) must be managed and maintained by the user. Moreover, the reproduction is not as straightforward as with channel-based systems, since information about the decoding system is required (such as speaker positions, decoder settings, channel ordering, etc.), which can hinder exchange and portability. In addition, composers are often required to work in several environments simultaneously, for example for sound synthesis/editing, temporal/structural arrangement and spatialization (i.e. B-format-files need to be created in one environment, then exported into another one and finally decoded in yet another one), which can significantly slow down the workflow. In fact, composers often prefer using the spatialization tools available in their environment of choice in order to avoid this logistic overhead. Finally, historical problems remain, such as the limited access to loudspeaker arrays, suboptimal reproduction setups and acoustic properties in concert venues. In order to address these usability issues, we propose a tight coupling of encoding (in a high-level composition environment) and decoding process (in an interactive real-time application) to assure compatibility and efficient workflow.

5. DYNAMIC SOUND FILE DECODING AND DIFFUSION: THE MULTIPLAYER

The MultiPlayer is a stand-alone application for real-time processing, decoding and diffusion of multichannel files in various formats. It is implemented as a set of high-level modules complying with the Jamoma framework for Max/MSP [11], as shown in Figure 4. Developed to complement the rendering and authoring part in the OMPrisma framework, it seamlessly integrates into the workflow in OM via bidirectional communication using Open Sound Control [12]. In addition, all parameters can be controlled in real-time through the graphical user interface and remotely via UDP, e.g. using external hardware controllers or tracking devices.

Three-dimensional loudspeaker setups can be configured using numerical and graphical interfaces (Fig.5-c). Optionally, sound-pressure level and time-differences in non-equidistant loudspeaker setups can be manually or automatically compen-

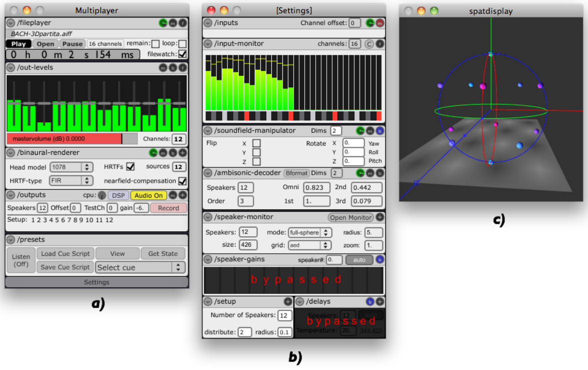

Figure 5: The MultiPlayer: Real-time processing, decoding and diffusion of multichannel audio files.

sated for. The MultiPlayer features modules for manipulations of ambisonic soundfields (rotations and mirroring along the principal axes) and ambisonic decoding for up to 32 speakers (using ICST's ambisonics tools [13]) (Fig.5-b).

To facilitate the work in the absence of loudspeaker-arrays, we implemented a module for virtual loudspeaker binauralization based on Ircam's SPAT [14]. This permits composers to audition 3D-audio scenes from off-center listening positions and for arbitrary loudspeaker setups. Depending on the available computing resources the binaural rendering can be gracefully degraded, as a tradeoff between quantity of sound sources (i.e. virtual loudspeakers) and quality of rendering (Fig.5-a).

6. CONCLUSION

We presented the OMPrisma library and a number of examples involving the control of periphonic sound source spatialization and spatial sound synthesis within a computer-aided composition framework. The paradigm of dynamic instantiation of synthesis patches in Csound combined with high-level programmable control structures in OM allows overcoming common limitations of conventional systems, imposed by the complexity of managing large amounts of individual spatialization parameters and/or the limited computing resources available for rendering in real-time contexts. Spatialization processes can be carried out on different scalar levels making use of the same tools and interfaces as for the composition of scores or the control of sound synthesis processes.

We tried to demonstrate that ambisonic technology is particularly well-suited for our approach. However, usability issues remain and often composers prefer working with less flexible spatialization techniques due to their inherent logistic simplicity. We address these issues with a real-time decoding application (MultiPlayer) which is tightly integrated with the compositional environment and can be used either in compositional contexts (using OSC communication with OM, auralization, etc.) or for in-site reproduction.

Interesting directions for future work include the implementation of an auxilliary audio-bus for external processing in the spherical harmonics domain and a vision-based head tracking system which uses a webcam only (as included in many Laptops nowadays) to control the binaural rendering without the need of dedicated hardware.

Additional information, sound examples and downloads of OMPrisma and the MultiPlayer are available at `http://www.music.mcgill.ca/~marlon/OMPrisma`

7. ACKNOWLEDGMENTS

M. Schumacher would like to thank Sean Ferguson and Ircam's Music Representation Research Team for supporting this project. Many thanks to Marco Stroppa for sharing his knowledge on sound synthesis control and Nils Peters for valuable discussions on the MultiPlayer application.

8. REFERENCES

[1] G. Assayag, C. Rueda, M. Laurson, C. Agon, and O. Delerue, "Computer Assisted Composition at IRCAM: From PatchWork to OpenMusic", *Computer Music Journal*, 23(3), 1999.

[2] C. Agon, M. Stroppa, and G. Assayag, "High Level Musical Control of Sound Synthesis in OpenMusic", in *Proc. International Computer Music Conference*, Berlin, Germany, 2000.

[3] R. Boulanger (Ed.), *The Csound Book. Perspectives in Software Synthesis, Sound Design, Signal Processing and Programming*, MIT Press, 2000.

[4] N. Peters, S. Ferguson, and S. McAdams, "Towards a Spatial Sound Description Interchange Format (SpatDIF)", *Canadian Acoustics*, 35 (3), 2007.

[5] F. Otondo, "Contemporary trends in the use of space in electroacoustic music", *Organised Sound.*, 13(1), 2008.

[6] V. Verfaille, U. Zolzer and D. Arfib, "Adaptive digital audio effects (A-DAFx): A new class of sound transformations", in *IEEE Transactions on Audio Speech, and Language Processing*, 14(5), 2006.

[7] D. Menzies, "W-panning and O-format, Tools for Object Spatialization", in *AES 22nd International Conference: Virtual, Synthetic, and Entertainment Audio*, Espoo, Finland, 2002.

[8] C. Ramakrishnan, J. Goßmann, and L. Brümmer, "The ZKM Klagdom", in *Proc. New Interfaces for Musical Expression*, Paris, France, 2006.

[9] D. Menzies, *New Electronic Performance Instruments For Electroacoustic Music*, PhD Thesis, University of York, UK, 1999.

[10] J. Anderson and S. Costello, "Adapting artificial reverberation architectures for B-format signal processing", in *Proc. of the Int. Ambisonics Symposium*, Graz, Austria, 2009.

[11] T. Place and T. Lossius, "Jamoma: A modular standard for structuring patches in Max", in *Proc. International Computer Music Conference*, New Orleans, USA, 2006.

[12] M. Wright, "Open Sound Control: an enabling technology for musical networking", *Organised Sound.*, 10(3), 2005.

[13] J.C. Schacher, P. Kocher, "Ambisonics spatialisation Tools for Max/MSP", in *Proc. International Computer Music Conference*, New Orleans, USA, 2006.

[14] J.-M. Jot and O. Warusfel, "A Real-Time Spatial Sound Processor for Music and Virtual Reality Applications", in *Proc. International Computer Music Conference*, Banff, Canada, 2005.