Tools and Applications for Interactive-Algorithmic Control of Sound Spatialization in OpenMusic

Jérémie Garcia, Jean Bresson, Marlon Schumacher, Thibaut Carpentier, Xavier Favory

TOOLS AND APPLICATIONS FOR INTERACTIVE-ALGORITHMIC CONTROL OF SOUND SPATIALIZATION IN OPENMUSIC

Jérémie Garcia[1], *Jean Bresson*[1], *Marlon Schumacher*[2], *Thibaut Carpentier*[1], *Xavier Favory*[1]

[1]UMR STMS: IRCAM-CNRS-UPMC
1, place Igor Stravinsky, 75004, Paris, France
{bresson,jgarcia,carpentier,favory}@ircam.fr
[2] CIRMMT / IDMIL, McGill University
555 Sherbrooke Street W. H3A 1E3, Montréal, Canada
marlon.schumacher@music.mcgill.ca

ABSTRACT

We present recent works carried out in the OpenMusic computer-aided composition environment for combining compositional processes with spatial audio rendering. We consider new modalities for manipulating sound spatialization data and processes following both object-based and channel-based approaches, and developed a framework linking algorithmic processing with interactive control.

1. INTRODUCTION: AUTHORING TOOLS FOR SPATIAL AUDIO

Authoring tools for spatial audio are naturally influenced by the orientation of their host environments or of the technological frameworks they fit in, be it from computational, representational or user interaction points of view. For instance, the Spat library [1] provides rich and multi-fold interfaces for monitoring real-time spatialization in Max, whereas spatialization tools in OpenMusic [2] integrate spatial audio control and rendering in offline computer-aided composition (CAC) processes, emphasizing the expressivity and computational power of algorithmic specifications [3]. Other examples include for instance the SSMN project (Spatialization Symbolic Music Notation [4]), which embeds spatial cues and trajectories in the music notation-oriented paradigm of a score editor, or ToscA [5], a new plugin providing a straightforward connection of the Spat real-time DSP and interfaces to Digital Audio Workstations. Some recent approaches focus on gestural controllers to manipulate complex sets of parameters with rich and expressive possibilities [6, 7, 8]. However, these tools are designed for performance applications and composers can rarely take advantage of such devices or interactions.

The diversity of orientations between the authoring and spatialization tools used during early compositional stages, and those actually used in production, either on stage or during recording sessions, provides a challenging research field for computer music: it is difficult to compose with spatial parameters and movements without structured and symbolic representations (as in scores or in CAC environments), and equally hindering to create spatial sound scenes without effective means for interactive audio-visual feedback. A common solution to combine expressive specification and interactive rendering is to separate spatial rendering engines from the authoring tools that produce control data [9, 4]. In such multi-layered environments, modularity is paramount for benefiting from the different representations and computation paradigms (for instance, combining explicit representations of time and real-time audio rendering). This modularity implies seamless protocols for data exchange between the respective components, which can be carried out either via file I/O or via networking (using OSC [10] or dedicated descriptions [11]).

Similar concerns have driven recent developments in the OpenMusic computer-aided composition environment [12]. Encoding of spatialization descriptors using the SDIF format [13] and OSC-based communication systems have been proposed to connect trajectories and other spatial specification data produced in OpenMusic to spatial rendering tools, either in an off-line mode (via SDIF and audio files produced and collected in the CAC environment) or in a real-time context (via dedicated players streaming the generated timed information to Max/*spat*$\sim$ for rendering). In this paper we present a number of related tools and new applications which combine algorithmic generation and transformation of spatialization controllers with interactive systems facilitating data input and exploration. We first describe the current technological framework for spatial sound available in OpenMusic (Section 2). Then we introduce two extensions of this framework (a) toward channel-based spatialization (Section 3) and (b) using OpenMusic reactive processes (Section 4).

2. TECHNOLOGICAL FRAMEWORK FOR SPATIAL SOUND IN OPENMUSIC

Currently the main tools available for the control sound spatialization in OpenMusic are:

OM-Spat. This library provides a set of tools built around the *spat-matrix* object: an array of parameters specifying the position, orientation, aperture, reverberation of an arbitrary number of sound sources. Parameters can be either static or time-varying data. The matrix is converted to streams of data frames written into an SDIF file. The *spat* command line tool shipped with the library is invoked for offline rendering of the spatial description: it processes input audio files according to the SDIF control file and produces a multi-channel bounce.

OMPrisma. This library implements a stratified approach to sound spatialization, separating authoring from rendering and reproduction stages using dedicated environments [14]. It also uses matrix structures to represent parameter vectors (rows) of individual sound sources (represented as rows). Sound synthesis and spatialization patches can be merged into hybrid structures for the spatialization of individual sound synthesis components (additive, granular, etc.) – an approach referred to as "spatial sound synthesis" [15].

An essential part of the definition and programming of complex spatial scenes and trajectories in compositional processes is the consideration of time as a musical parameter in the specification of spatial data. This specificity is one of the main distinctive aspects of computer-aided composition systems as compared with authoring tools integrated within real-time systems. Curves and trajectories are represented as standard objects in the OpenMusic visual programming environment and can be used for instance to parameterize the control matrices in OM-Spat/OMPrisma. Specific editors have been developed to visualize data in 3D and help editing it using projection planes. These object can be set and edited manually, or programmed and generated by arbitrary complex algorithms related to the compositional processes (see Figure 1). Temporal parameters can be either explicitly specified or derived from the context, e.g. corresponding to the duration or to the markers in a sound file, or expanded via interpolations between reference points.

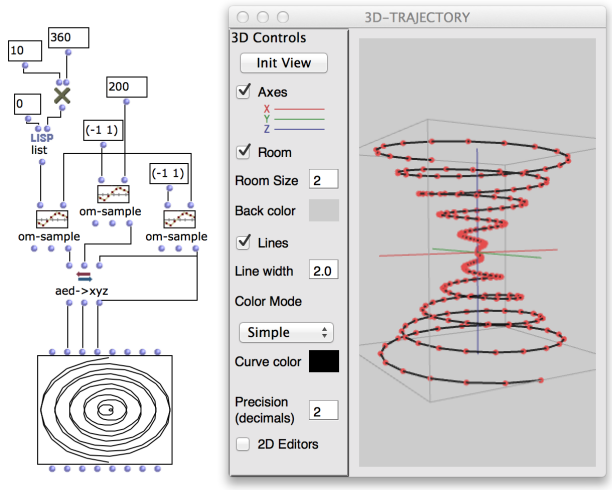

Figure 1: Generation of 3D trajectories in OpenMusic.

As previously noted, the communication with external devices and applications provides a means to bridge the gap between formal/algorithmic compositional tools and interactive/real time spatial audio renderers. Such communication can take place at various stages of the composition and processing of spatial parameters: during the authoring phase, where trajectories and other parameters are defined, during the rendering when this data is transferred to DSP systems and integrated with real-time interactions, or during the reproduction phase for interactive auralization. Accordingly, several external tools are currently available and connected to the computer-aided composition framework:

Spat-SDIF-Player. This application is a Max standalone communicating with OM to load SDIF files containing spatialization control data. It provides functionalities for streaming their contents (typically, the positions of an array of sound sources) as OSC messages encoded according to the SpatDIF description [11]. The OM-generated data can then be rendered by the Spat~ or any system able to receive an interpret these messages.

Trajectoires. This web application runs either on desktop computers, smart-phones or tablets [16]. It allows to draw and edit trajectories with finger-based interaction and communicates bi-directionally with OM (or other applications) through the OSC protocol. It also acts as a mobile remote control to stream the data in real-time (see Figure 2).

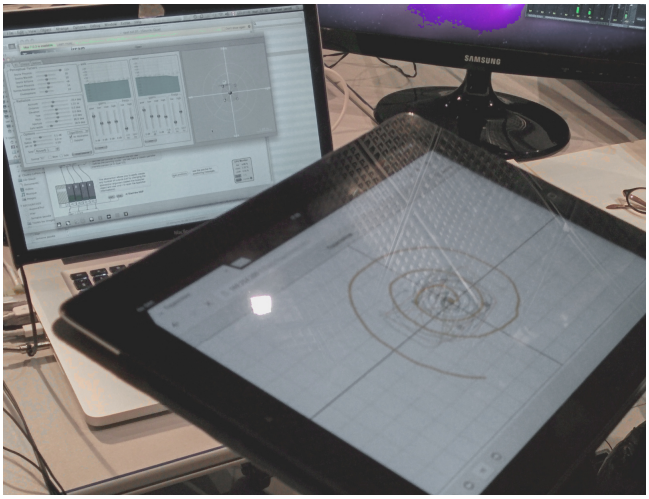

Figure 2: Using the *Trajectoires* mobile application to control spatialization in Max/*Spat*.

MultiPlayer. This Max-based application allows for real-time decoding, diffusion and auralization of multichannel audio formats which have been produced by the sound rendering tools in OM. It provides an OSC interface and can be controlled by OM or any other external source of OSC messages, e.g. for interactive control of soundfield manipulations [14].

3. CHANNEL-BASED SPATIALIZATION: AN APPLICATION WITH OM-SOX

Techniques for sound spatialization can be categorized on the basis of two common representations: *object-based* and *channel-based* [17]. In the former, a parametric description of the spatial scene is specified, which is rendered into audio using a spatialization algorithm at a later stage (this is the case for most of the tools discussed in the previous sections). In the latter, multichannel audio is directly manipulated using DSP techniques in order to process audio content in respective channels. While object-based control is an efficient paradigm for simulating virtual sound scenes, channel-based control allows composers to create arbitrary spatial patterns (e.g. using irregular loudspeaker configurations) and develop creative approaches which do not adhere to simulating real-world situations.

From an aesthetic perspective, these two models are characterized by two respective viewpoints. On the one hand the notion of spatialization systems as a means for the creation of virtual, illusory sources and spaces within the actual physical listening space. In this case, loudspeakers are merely technological artifacts of a rendering system, rather than musical elements (the fact that in systems such as wave-field synthesis, loudspeakers are often hidden behind panels or embedded into walls, emphasizes this idea). On the other hand, the notion of loudspeakers as an extension of acoustic instruments, i.e. as distinct sound-producing sources which are "assigned" musical materials. In channel-based techniques, the mapping between higher-level parameters and lower-level DSP functions can become part of compositional specifications, e.g. in terms of patterns or functions for distributing audio to channels.[1]

Figure 3 shows an example for channel-based spatialization realized in OpenMusic using functionalities of the library **OM-SoX** [19]. This library provides a system of classes (representing audio data) and functions (representing audio processes) which can be connected together in order to create a multichannel audio graph. OM-SoX processes can be nested into each other, which is well-aligned with the notion of functional programming. An interesting feature regarding composer-program interaction, is that these processes can be carried out in memory, which provides a hybrid offline/inline paradigm for symbolic audio manipulation: it is possible to pre-audit processes at different positions in the audio graph in real-time, e.g. for iterative tweaking of parameters. Once a satisfying result is obtained, the audio can be rendered into a file on disk.

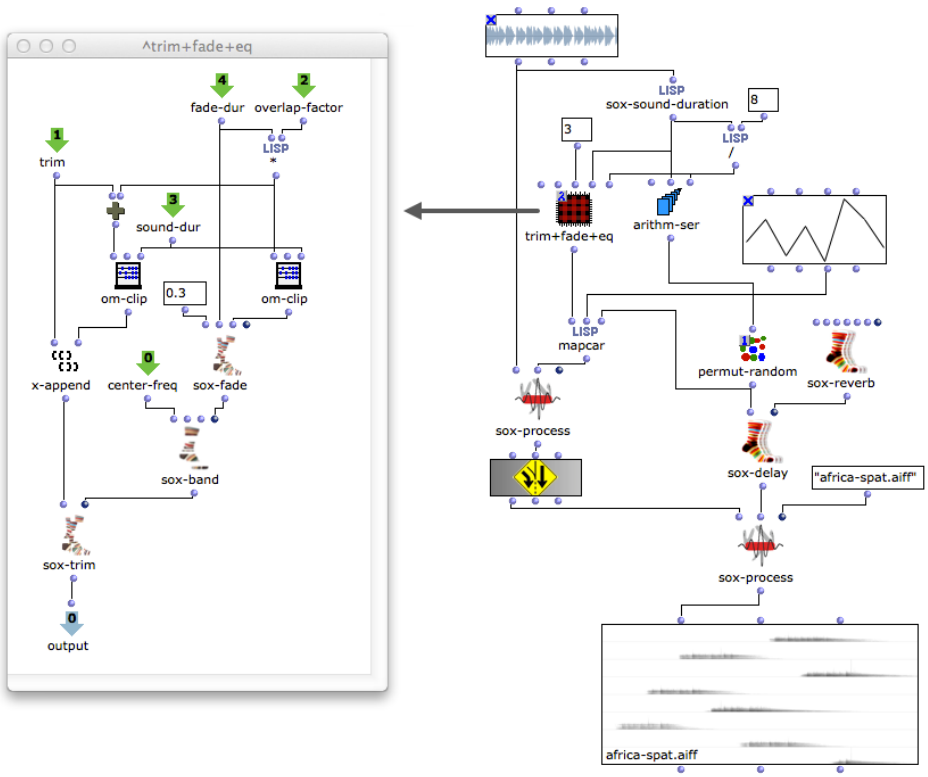

Figure 3: Channel-based spatialization with OM-SoX.

The spatialization process in Figure 3 is realized as a visual program in OM through explicit manipulation of audio channels, rather than a spatialization algorithm. In this process, a source sound (at the top of the figure) is segmented, filtered, and panned between 8 loudspeakers (possibly in an irregular setup). This is achieved by nesting two sox-processes. In the first, the source sound is segmented (*sox-trim*) into eight equally long sections, which are individually filtered through a band filter (*sox-band*) with a centre frequency specified in a graphical BPF editor (breakpoints determine the centre frequency of the filter for the respective segments). Each segment is faded in over 0.3 seconds and faded out in 3 seconds (*sox-fade*). This produces asymmetric panning functions, which

[1] In another taxonomy, these two approaches have been termed "simulation-oriented" vs. "pattern-oriented" [18].

have been described as preferable for irregular loudspeaker setups (on-stage, installations, etc.) and for fast sound transitions between loudspeakers, unlike conventional symmetrical panning functions used by most spatialization algorithms [18]. The individual segments resulting from the first sox-process (a) are then merged into a multi-channel representation. This multi-channel representation is used as input to the second sox-process (b), which applies individual time delays (*sox-delay*) and reverberation (*sox-reverb*) to each of the channels. Lastly, their order in the multichannel representation is shuffled: a spatialization pattern in which sound segments are panned between non-adjacent channels.

Since each of the sound segments has been processed with an individual band filter and is assigned to a dedicated channel, the spatialized sound changes its timbre while travelling between spatial locations. This application can be compared to the notion of different "spatial zones" [20], however, not determined as a function of direction, but via distinct spectral characteristics for individual loudspeakers, similar to an *Acousmonium* as described in [21]. The different spectral characteristics and asymmetrical panning functions between non-adjacent speakers can be seen in the multichannel sonogram at the bottom of the figure.

4. ONLINE ALGORITHMIC TRAJECTORIES MANIPULATIONS WITH REACTIVE VISUAL PROGRAMS

"Reactive processes" in OpenMusic [22] combine the traditional approach of computer-aided composition systems, based on off-line, functional programming, with the notion of reactivity inspired by real-time musical systems. This hybrid paradigm allows changes or *events* occurring in visual programs (or in the data they contain) to produce series of updates and evaluations. In this context, the *events* can originate from user actions (e.g. from graphical user interfaces) or from external sources (e.g. MIDI or UDP/OSC messages), which allows for bidirectional communications between compositional programs and remote applications or devices. Reactive CAC processes can therefore participate in a structured interaction with their environment, taking place either in a compositional perspective (in the processes leading to the generation of musical material) or in a context of performance. For instance, an OpenMusic visual program can collect data from a music performance, compute representations on the fly or process this data to generate new musical material (as in an automatic accompaniment or improvisation system where sequences played by a musician are analyzed and recombined to produce other parts [23]).

The following examples show potential applications involving spatialization tools (and in particular the *Trajectoires* mobile application) in reactive visual programs, and how interesting workflows can emerge at the crossways between formal composition and interaction. We will consider a work session where a composer uses OpenMusic (OM) for composing spatial structures (we can imagine that other aspects of his/her compositional process are also carried out in this environment and related to these spatial structures), *spat*$\sim$ within Max for rendering and monitoring the spatialization, and the *Trajectoires* application running on a mobile device (iPad or equivalent).

It is straightforward to design arbitrary processes generating trajectories in OM (see for instance Figure 1) and to send them to the *Trajectoires* application in order to constitute a dictionary of spatial data, which can be selected, composed and manually transformed to constitute some of the spatial scene components. Conversely, the trajectories produced in the mobile application can be sent over the network and received both in the spatial rendering environment and in OM.

In Figure 4 an OM patch receives and filters incoming trajectories (lists of points), and processes them through transformation algorithms (in this case, a simple rotation, but we can imagine any transformation related to a compositional process). As the patch is reactive, this transformation operates automatically as soon as a trajectory is received through OSC, and the internal data containers and editors are updated accordingly. Sending back the data to the mobile application can also be part of the reactive process (as it is the case in Figure 4) and the mobile application will immediately receive and store a new transformed trajectory.

Similar processes can be applied iteratively to generate sets of trajectories (e.g. in this case rotated with different angles) from a single input coming from the drawing application. In Figure 5 the rotation process is applied iteratively to produce eight trajectories by successive transformations of a single input curve. The generated curves sent back to the mobile application can be assigned to different sources and "played", that is, streamed together as OSC messages to control a real-time spatial audio renderer.

Figure 6 shows an example involving slightly more complex interactions: the *route-osc* box in the OM patch routes the downstream reactive computations depending on an identifier assigned to the received trajectory. A first trajectory (`"/traj/source1/traj"`) is just stored, while upon reception of a second trajectory (`"/traj/source2/traj"`), an interpolation process is executed to compute ten intermediate trajectories. The interpolated curves are also sent back to the *Trajectoires* interface for monitoring and control of the 10-sources spatialization.

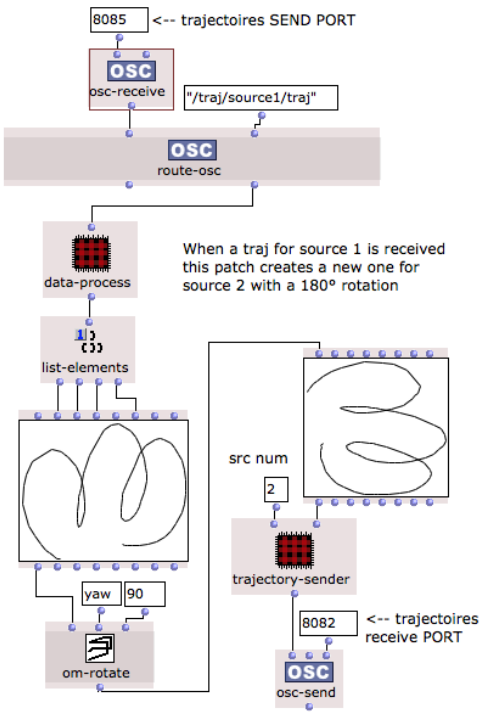

Figure 4: Reactive processing of trajectories in OpenMusic. The dark-framed boxes are *reactive* to upstream changes. The *trajectory-sender* sub-patch formats OSC messages to be sent via UDP by the *osc-send* box.

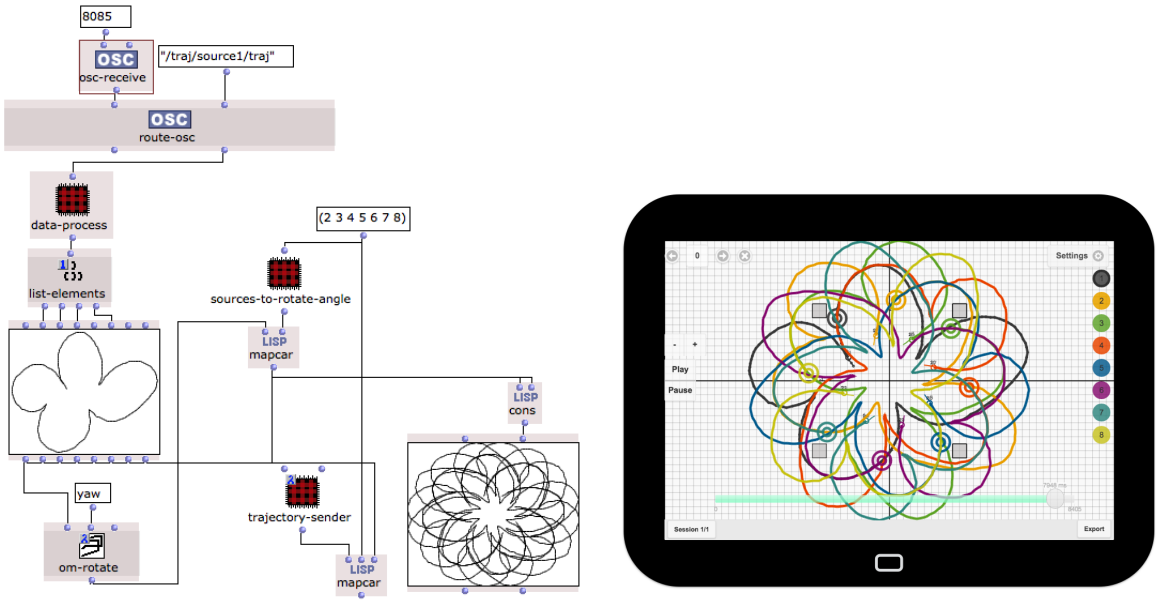

Figure 5: Generating series of trajectories. Left: Iterative processing and generation of trajectories in OM. The reactive *mapcar* boxes iterate the rotation and OSC-send operations. Right: Reception and monitoring in the *Trajectoires* mobile application.

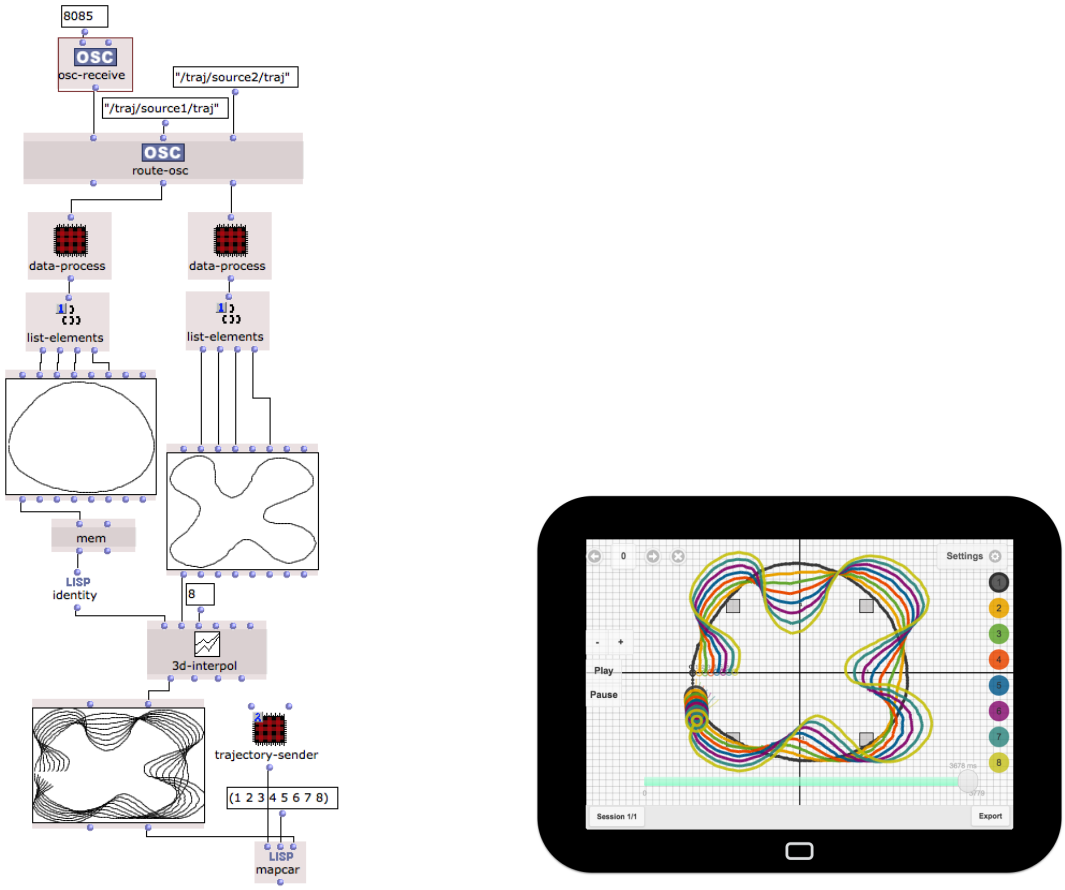

Figure 6: Left: Interpolation between trajectories in OM. The first received trajectory is memorized (*mem*) and the interpolation is triggered upon reception of the second trajectory. Right: Reception and monitoring in the *Trajectoires* mobile application.

5. CONCLUSION

In this paper we presented a number of interactive-algorithmic tools for the control of sound spatialization in OpenMusic. We discussed possible representations and techniques and showed an example for controlling spatialization via direct manipulation of multi-channel audio signals defined as a visual program, as well as the application of the reactive visual programming framework of OpenMusic to spatialization control.

In scenarios where sound spatialization is carried out in-site within a given venue, the control is commonly performed in real time, with little long-term temporal or musical control. Computer-aided composition tools and representations allow to integrate the dimension of time into these scenarios, through off-line programming, compositional specification and high-level musical structures. The examples presented in this paper illustrate the potential of computer-aided composition programs considered as components in larger-scale frameworks comprised of varied interfaces, tools and rendering engines, emphasizing the links between compositional/off-line processing and audio/real-time systems and interactions, which we believe, are promising perspectives for computer music systems.

6. ACKNOWLEDGEMENTS

This work is supported by the French National Research Agency project EFFICACe ANR-13-JS02-0004.

7. REFERENCES

[1] T. Carpentier, M. Noisternig, and O. Warusfel, "Twenty years of Ircam Spat: looking back, looking forward," in *Proceedings of the International Computer Music Conference*, Denton, United States, 2015.

[2] J. Bresson, C. Agon, and G. Assayag, "OpenMusic. Visual Programming Environment for Music Composition, Analysis and Research," in *ACM MultiMedia (MM'11) OpenSource Software Competition*, Scottsdale, United States, 2011.

[3] J. Bresson, "Spatial Structures Programming for Music," in *Proceedings of the Spatial Computing Workshop (SCW) – Co-located w. Autonomous Agents and MultiAgent Systems (AAMAS)*, Valencia, Spain, 2012.

[4] E. Ellberger, G. Toro-Perez, J. Schuett, L. Cavaliero, and G. Zoia, "A Paradigm for Scoring Spatialization Notation," in *TENOR: First International Conference on Technologies for Music Notation and Representation*, Paris, France, 2015.

[5] T. Carpentier, "ToscA: an OSC communication plugin for object-oriented spatialization authoring," in *Proceedings of the International Computer Music Conference*, Denton, United States, 2015.

[6] J. J. Nixdorf and D. Gerhard, "Real-time sound source spatialization as used in challenging bodies: implementation and performance," in *Proceedings of the conference on New Interfaces for Musical Expression*, Paris, France, 2006.

[7] M. T. Marshall, J. Malloch, and M. M. Wanderley, "Gesture control of sound spatialization for live musical performance," in *Gesture-Based Human-Computer Interaction and Simulation*. Springer, 2009.

[8] G. Le Vaillant and R. Giot, "Multi-touch Interface for Acousmatic Music Spatialization," in *Joint International Computer Music, Sound and Music Computing Conference*, Athens, Greece, 2014.

[9] C. Bascou, "Adaptive Spatialization and Scripting Capabilities in the Spatial Trajectory Editor Holo-Edit," in *Proceedings of the Sound and Music Computing Conference*, Barcelona, Spain, 2010.

[10] M. Wright, "Open Sound Control: An Enabling Technology for Musical Networking," *Organised Sound*, vol. 10, no. 3, 2005.

[11] N. Peters, T. Lossius, J. Schacher, P. Baltazar, C. Bascou, and T. Place, "A Stratified Approach for Sound Spatialization," in *Proceedings of the Sound and Music Computing Conference*, Porto, Portugal, 2009.

[12] J. Bresson and M. Schumacher, "Representation and Interchange of Sound Spatialization Data for Compositional Applications," in *Proceedings of the International Computer Music Conference*, Huddersfield, United Kingdom, 2011.

[13] D. Schwartz and M. Wright, "Extensions and Applications of the SDIF Sound Description Interchange Format," in *Proceedings of the International Computer Music Conference*, Berlin, Germany, 2000.

[14] M. Schumacher and J. Bresson, "Compositional Control of Periphonic Sound Spatialization," in *2nd International Symposium on Ambisonics and Spherical Acoustics*. Paris, France: Citeseer, 2010.

[15] ——, "Spatial Sound Synthesis in Computer-Aided Composition," *Organised Sound*, vol. 15, no. 3, 2010.

[16] X. Favory, J. Garcia, and J. Bresson, "Trajectoires : une application mobile pour le contrôle et l'écriture de la spatialisation sonore," in *Conférence Francophone sur l'Interaction Homme-Machine*, Toulouse, France, 2015.

[17] M. Geier, J. Ahrens, and S. Spors, "Object-based Audio Reproduction and the Audio Scene Description Format," *Organised Sound*, vol. 15, no. 03, pp. 219–227, Oct. 2010.

[18] F. Canavese, F. Giomi, D. Meacci, and K. Schwoon, "Asymmetrical Envelope Shapes in Sound Spatialization," in *Proceedings of the Sound and Music Computing Conference*, Berlin, Germany, 2009.

[19] M. Schumacher. OM-SoX: Multichannel Audio Manipulation and Functional Batch Processing in Computer-Aided Composition. [Online]. Available: http://www.idmil.org/software/OM-SoX

[20] R. Penha, "Distance Encoding in Ambisonics Using Three Angular Coordinates," in *Proceedings of the Sound and Music Computing Conference*, Berlin, Germany, 2008.

[21] F. Bayle, "A propos de l'Acousmonium," *Recherche Musicale au GRM*, vol. 397, 1989.

[22] J. Bresson, "Reactive Visual Programs for Computer-Aided Music Composition," in *IEEE Symposium on Visual Languages and Human-Centric Computing*, Melbourne, Australia, 2014.

[23] J. Nika, D. Bouche, J. Bresson, M. Chemillier, and G. Assayag, "Guided Improvisation as Dynamic Calls to an Offline Proccess," in *Proceedinsg of the Sound and Music Computing conference*, Maynooth, Ireland, 2015.