



# Interactive-Compositional Authoring of Sound Spatialization

Jérémie Garcia, Thibaut Carpentier, Jean Bresson

## ► To cite this version:

Jérémie Garcia, Thibaut Carpentier, Jean Bresson. Interactive-Compositional Authoring of Sound Spatialization. *Journal of New Music Research*, 2017, 46 (1), pp.20. 10.1080/09298215.2016.1230632 . hal-01467080

**HAL Id: hal-01467080**

**<https://inria.hal.science/hal-01467080>**

Submitted on 14 Feb 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Interactive-Compositional Authoring of Sound Spatialization

JÉRÉMIE GARCIA

Université de Toulouse - ENAC  
7 av. E. Belin, 31055 Toulouse, France  
jeremie.garcia@enac.fr

THIBAUT CARPENTIER

UMR 9912 STMS  
IRCAM-CNRS-UPMC  
thibaut.carpentier@ircam.fr

JEAN BRESSON

UMR 9912 STMS  
IRCAM-CNRS-UPMC  
bresson@ircam.fr

## Abstract

*SPAT-SCENE is a graphical user interface and spatial sound processor embedded in computer-aided composition software. It combines the interface and the rendering engine of Ircam Spatialisateur with a compositional and temporal control of spatial scene descriptors. Previous studies of the composition process as well as interviews with the composers themselves emphasized the needs for tools integrated within compositional environments allowing for interactive input, visualization and manipulation of spatio-temporal data in spatial scene descriptions. The SPAT-SCENE interface addresses such requirements by providing a timeline view synchronized with the spatial view of the Spatialisateur, three-dimensional editors as well as several internal and external rendering possibilities, all in a unified framework. SPAT-SCENE is implemented in the OpenMusic visual programming language, and so can be articulated with other musical structures and compositional processes. Through the case study of sound spatialization, this work tackles the challenge of authoring real-time processes in a compositional context.*

**Keywords:** *spatialization, computer-aided composition, composition, interfaces*

## 1 Introduction

In the past decades, composers have been drawn to computers that allow them to use advanced signal-based processing tools for generating sounds or controlling their localization in auditory spaces. These processes are generally implemented in real-time media environments, enabling interactive and exploratory approaches to the production of music and sound. However, real-time environments provide limited possibilities for temporal authoring and the articulation of sound processing with musical structures [Puckette, 2004].

Sound spatialization techniques allow composers to define virtual acoustic spaces, arrange sources in spatial sound scenes or create motion of sound sources in such virtual spaces [Otondo, 2008, Baalman, 2010, Barrett, 2012]. They represent a typical example of the aforementioned problem: despite the efficient real-time spatialization tools available today, their musical control remains a challenging task for composers [Peters et al., 2011, Gottfried, 2015]. Spatialization requires the precise definition of a great number of features and parameters, such as the location of sound sources, their orientation, their directivity, or the reverberant characteristics of the virtual space. In order to produce motion or other dynamic effects, the temporal evolution of these parameters must also be specified, and often connected to external (musical) events or timing constraints.

So far, two main approaches have been proposed to tackle these issues. One involves extending real-time environments with time-oriented structures [Schnell et al., 2009, Agostini and Ghisi, 2013]. The other one consists of integrating audio processing engines within offline computer-aided composition frameworks [Laurson et al., 2005, Schumacher and Bresson, 2010, Agon et al., 2011]. In this paper we aim at blending these two solutions so that composers can explore and assess their ideas through real-time interaction while specifying the temporal aspects of digital sound processing. The objective of this work is to help composers structure real-time processing as part of a compositional approach, and to support their work with sound spatialization from early ideation to the production of sounds, control data or scores.

We built upon observations of composers’ practice to design SPAT-SCENE, a graphical user interface that combines the interface and rendering engine of the Ircam *Spatialisateur* with the compositional and temporal control of spatial scene descriptors. After a presentation of related work in Section 2 of this paper, we report and discuss these field observations in order to underline some motivations and objectives of this project in Section 3. The tool we developed supports the acquisition, visualization and editing of the spatialization parameters both in space and time. It also provides interactive mechanisms to connect real-time operations and rendering with deferred-time activities relating to the compositional processes. In Section 4 we detail several features designed to address specific challenges such as the visualization of three-dimensional trajectories, the acquisition of data using gestural controllers, and the interactive control of external rendering engines. SPAT-SCENE is integrated and developed in the OpenMusic visual programming framework; in Section 5 we show how this context allows to connect the spatialization control interface with programming and algorithmic generation, and to articulate spatialization processes with other musical structures and processes. Finally, we conclude with some technical considerations in Section 6 and more general discussions in Section 7.

## 2 Related work: Composing with spatial audio

### 2.1 Spatial composition techniques

Spatiality of music and sound has been a preoccupation of electroacoustic composers since the early experiments of *musique concrète* in the 1950s. From that point onward, and indeed throughout the history of electronic music, spatial composition techniques have taken many forms and explored a wide range of scenarios, from dispersion of orchestral groups, motion of sounds, performers and the audience, to the interaction of real and virtual sound entities [Harley, 1994].

Composers and performers today use spatial audio technologies to control auditory cues and spatio-musical attributes [Kendall, 2010]. As shown by various studies, spatial composition techniques used in this context are numerous and diverse [Otondo, 2008, Baalman, 2010, Peters et al., 2011]. These include (but are not limited to): manipulation of trajectories (motion and choreography of sound objects); control of spatial location and organization (distance, proximity, depth and extent of the spatial image); diffusion (envelopment and immersion, localized sounds vs. diffuse layers, sound mass vs. focal points, creation of subspaces, quiet zones, traveling waves [Harrison, 1998, Barrett, 2012]); artificial reverberation and spatial decorrelation [Kendall, 1995, Wilson and Harrison, 2010]; spatial granulation (where sonic grains are individually spatialized, typically by means of flocking or swarming simulations) [Kim-Boyle, 2006, McLeran et al., 2008]; frequency-domain diffusion (where spectral components are distributed in space) [Topper et al., 2002, Normandeau, 2009]; time-varying effects (Doppler shift, variations in loudness, dynamics, or spectral

filtering); etc.

The work discussed in the remainder of this paper will focus on a model of spatialization in which sound elements (typically point sources) are associated with time-varying data (such as position, orientation, etc.), and the manipulation of spatial trajectories is the main paradigm for the control of the spatio-temporal sound organization. From the user's perspective, data and metadata are easily understood regardless of the diffusion setup, and the rendering engine is responsible for delivering the sound scene according to the actual reproduction system (loudspeaker array or headphones). Such paradigms can be referred to as "source-centered", as opposed to "speaker-centered" paradigms, especially within the context of an acousmatic musical approach. As a consequence, it is acknowledged that the tools proposed in this paper will only provide solutions for a subset of compositional viewpoints. Nonetheless, the approach considered here already offers a broad range of possibilities, and appears to be a popular one in contemporary compositional practice [Peters et al., 2011].

## 2.2 Spatialization engines

Over the last decades, the processing power of computers has grown exponentially, multichannel audio equipment has become affordable, multichannel communication protocols (such as MADI<sup>1</sup> or Dante<sup>2</sup>) have been standardized and largely adopted, and ambitious 3-D diffusion systems of dozens or even hundreds of loudspeakers have been installed in several venues. All these factors have led to a renewed interest in sound spatialization techniques, and numerous digital rendering tools have been developed.

An exhaustive review of sound spatialization software is beyond the scope of this paper. As shown by [Peters et al., 2011], a majority of these tools are integrated into real-time media programming environments such as Max, PureData or SuperCollider. Others operate as standalone applications or can be inserted as audio plugins into digital audio workstations (DAWs). Some of them feature a visualization/control interface which most often consists of a schematic representation of the sound scene, either in 2-D or 3-D perspective, at a given instant in time. As a matter of fact, most of this software is biased towards live/interactive performances and real-time audio workflow.

## 2.3 Spatial authoring

The generalization of real-time spatial audio processing led to interesting works on spatialization control using interactive systems. Gestural interfaces and input devices such as game controllers, video input [Bozzolan and Cospito, 2008], or sensors and mapping technology [Marshall et al., 2009], have been used for this purpose. Such interactive approaches, emphasizing instantaneous representation and processing of spatial features, are important and meaningful in real-time monitoring scenarios. However, they appear ill-suited for compositional purposes where temporal extent and organization (both short-term and long-term) have to be considered [Gottfried, 2015].

Early attempts at addressing the issue of temporal authoring for spatial audio date back to the 1970s and were notably documented in seminal works by John Chowning [Chowning, 1971]. Since then a number of tools and projects have tackled this issue following varied perspectives and approaches.

---

<sup>1</sup>Multichannel Audio Digital Interface

<sup>2</sup><http://www.audinate.com>

Trajectory editors such as Holo-Edit [Pottier, 1998] or Zirkonium [Wagner et al., 2014] allow the manipulation of spatial data through the use of breakpoint curves or sequences of time-tagged points in a 3-D space. Motion can be graphically edited through lines, Bézier or spline curves, and multiple viewpoints of the scene are usually offered. WFS Path Editor [Baalman, 2007] also allows the speed, the acceleration or the quantization of the trajectories to be modified. The Holo-Edit software further provides generative (Lissajou, Brownian, random algorithms, etc.) and transformative (rotation, translation, proportion) functionalities, while Octogris [Normandeau, 2015] comes with a set of predefined parametric trajectories (ellipse, spiral, etc.). Freehand drawing (actually controlled with the mouse) can also be used to record trajectories, e.g. in Ambicontrol [Schacher, 2010] or SpaceJam [Madden, 2014], fostering a simple and intuitive way of describing motions that can later be used as compositional patterns. A common characteristic of these different tools is the explicit separation of the authoring environment in which composers draw, edit or generate trajectories from the rendering engine that actually performs spatial audio processing. While this is meant to emphasize modularity, this approach requires the user to regularly switch working environments and contexts.

Automation in DAWs is also sometimes used to store, edit and lay out spatial controllers in compositional sessions. This type of workflow was adopted by the ViMic [Braasch et al., 2008] and Spatium [Penha and Oliveira, 2013] systems, among others, and made possible through dedicated control plug-ins that communicate with the auditory rendering engines. The AscoGraph application [Coffy et al., 2014] also offers visual interfaces for handling time structures that control electronic actions and processes, and has been used by some composers as an authoring tool for sound spatialization. In these editors, one automation track is typically allocated to each unidimensional parameter (e.g. a spatial coordinate); as such, it is difficult to graphically apprehend the spatial composition, which intrinsically requires a multidimensional standpoint.

The Spatialization Symbolic Music Notation framework (SSMN) [Ellberger et al., 2015] defines a taxonomy of spatial features (sound source descriptors such as patterns, figures, motives, etc.) represented by graphical symbols, which can be embedded into a (traditional) score in order to include the spatialization of sounds in musical notation. A dedicated rendering engine can interpret the spatial symbols and produce the spatialized audio signals accordingly. Following a similar path, [Gottfried, 2015] encourages symbolic graphic notation for composing sound spatialization, and introduces an interpretive layer between the visual representation and the rendering system.

Trajectories can also be generated and edited in other graphical/media sequencers. For instance, Ian-niX [Jacquemin et al., 2012] or i-score [Celerier et al., 2016] provide spatial metaphors (e.g. polytemporal cursors evolving along 2- or 3-D curves) and scripting possibilities to produce interactive scores or scenarios operating as remote controllers for audio renderers. Authoring of auditory scenes is also possible with software programs made for gaming or virtual reality, such as Blender, Unreal Engine or Unity. These environments, used as 3-D scene modelers, editors and animators, can be linked with audio rendering APIs such as OpenAL or FMOD, and serve for spatialization or auralization purposes, see e.g. [Olaiz et al., 2009]. To our knowledge however, such tools have rarely been adopted by composers.

Various approaches have been proposed to reduce authoring complexity and/or foster computational expressivity as the number of sources and parameters increases. These approaches include the use of audio features [Todoroff et al., 1997], physical models [Baalman, 2004, Penha and Oliveira, 2013], constraints engines [Delerue, 2004] or flocking and swarming algorithms [Kim-Boyle, 2006]. In the OpenMusic computer-aided composition environment, a number of tools and libraries exist that allow the integration of the

algorithmic generation of trajectories to compositional processes and offline spatial audio rendering [Bresson, 2012]. In particular, the OMPrisma library provides a compositional toolbox featuring algorithmically driven manipulations of spatial parameters [Schumacher and Bresson, 2010]. These tools lack real-time feedback and monitoring of dynamic aspects, since the composers have to convert the graphical scores into control data or sounds before hearing and assessing any audio results.

### **3 A study of composers' practice**

The design of the tools presented in this paper was informed by a field study conducted with professional contemporary music composers in order to better understand their actual work and issues with sound spatialization. The compositional process of eight composers, all experienced in computer-assisted composition tools and in spatial audio, was examined during the elaboration of sound spatialization, from the initial sketch to the layout of the final score. We collected photographs or copies of the sketches and scores, and we conducted interviews (lasting approximately one hour and a half) with each participant. Critical Incident-style questions [Flanagan, 1954] were asked to provide concrete examples from their recent work of how they addressed problems, followed by more open-ended discussions.<sup>3</sup>

As we mentioned previously, existing compositional approaches are rich and diverse: through the collection of this field-data we did not seek to narrow a generic practice, but rather to explore a design space and gain insights or ideas grounded in real-world composition activities. Most composers interviewed worked with real-time audio software, and indeed had to tweak their favorite tools (Max, Antescofo, ProTools...) in order to turn them into sound spatialization controllers. In the remainder of this section, we describe and discuss three aspects that have retained our attention during this study: (1) graphical/gestural input, (2) time management for the control of spatial audio processing, and (3) scoring and notation.

#### **3.1 Data input**

The composers participating in our study generally worked with graphical representations to design spatial effects and spatial sound characteristics in the early creative stages, often sketched on paper as illustrated in Figure 1. Some of the sketches are very rough and meant to illustrate an idea (Figure 1.a) whereas others can be very detailed and used for more formal explorations (Figure 1.b).

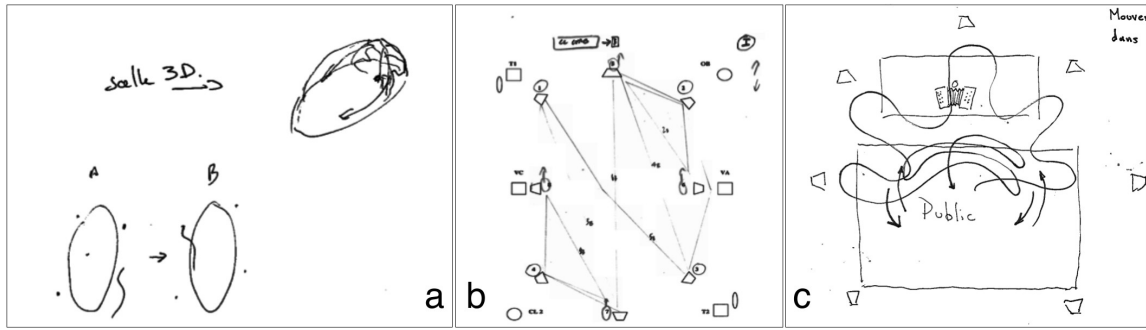
After the ideation phase, the participants used various kinds of software such as Max, Protools, Antescofo, OpenMusic or Zirkonium to implement their ideas and capture preliminary sketches of sound motion. When asked about it, they all expressed interest in easier and more intuitive means to input spatial data in musical software, and the need for better visual cues, for instance to facilitate the identification of trajectories in a set of spatial data, or the assessment of speed and overall temporal dynamics of a given source's motion.

#### **3.2 Temporal control**

The different tools used to implement spatialization processes promote a wide variety of approaches to their temporal organization and possible connection to the score or performance data. The majority of composers interviewed found it difficult to define the temporal aspects of the drawn trajectories, and mentioned

---

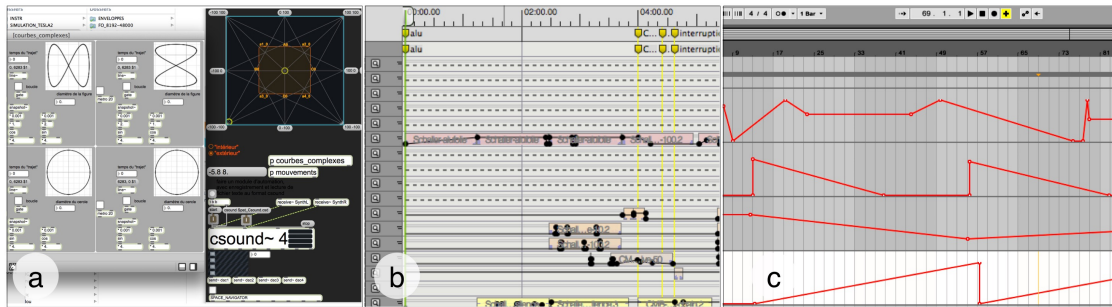
<sup>3</sup>Parts of these interviews have been published and discussed in [Favory et al., 2015].



**Figure 1:** Examples sketches of spatial trajectories collected from interviewed composers.

the temporal specification in general as being a major compositional problem when working with sound spatialization. This is consistent with the results of [Peters et al., 2011] which also highlighted such lack of tools for musical specification.

The composers often used pre-recorded data or parameters stored in series of cued Max message events, triggered during the performance by real-time interactions or using score following software (e.g. Antescofo [Cont, 2008]). As we mentioned already, spatial parameters are also sometimes controlled by automations defined in standard Digital Audio Workstations. Figure 2 shows examples of composers' workspaces involving these different kinds of approaches and controllers. In the case of Figure 2-a (Max) the trajectories are displayed but their timing and relative placement/occurrence in the piece is implicitly defined by algorithmic processes and external events (therefore, not visible or accessible in the compositional realm). In Figures 2-b and -c (DAWs) a timing is visible and editable, but the individual parameters automation fails in expressing consistent (musical) spatial cues and representations.

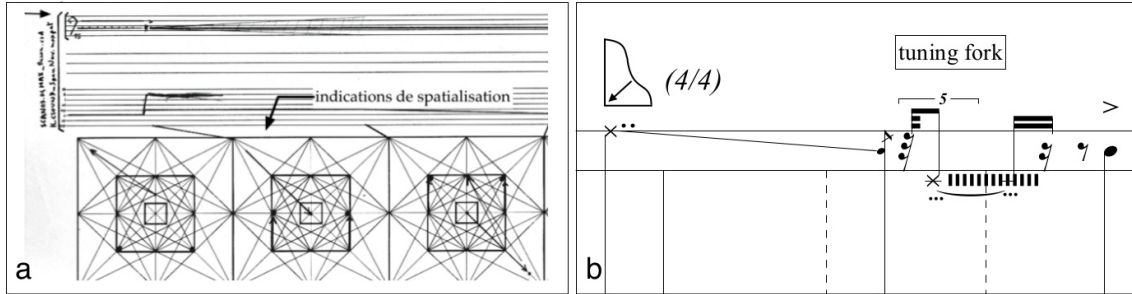


**Figure 2:** Workspaces. a) Max environment: predefined parametric curves are triggered using algorithmic timers. b) Electronic sketch in Digital Performer. c) Automation curves for spatialization in Ableton Live.

### 3.3 Score and notation

Another important issue brought forward in this study is the articulation of spatial descriptions with the other musical elements of a music score. In order to enhance score readability and to facilitate the performance of their music, the composers we interviewed tended to create their own idiosyncratic kinds of visual

representations for sound motion and trajectories to be inserted in the score. Figure 3 shows two extracts from the scores of two interviewed composers including such spatial indications as annotations. These graphical cues can be derived and simplified from the original sketches. They are generally added at a late stage of the score editing phase, as compared to pitches, rhythms or even sound synthesis elements. One possible explanation is that composers need to have sound elements (virtual sources or performers) available before to start experimenting with sound spatialization, which is generally not the case at the beginning of the score editing process.



**Figure 3:** Details from scores including spatialization indications. a) Miniatures of trajectories using arrows at different time positions in the score. b) Symbolic indications of a motion using an arrow.

### 3.4 Findings of the observational study

Although this study focuses on some specific issues and compositional standpoints, it has nonetheless confirmed some conclusions of previous surveys [Peters et al., 2011] and has put forward a number of critical needs and features which could be inferred to more general cases, such as:

- The easy and intuitive input of spatio-temporal data;
- Graphical cues to process this data visually;
- The temporal control over spatial descriptions;
- The connection of spatial descriptions to the other components of a compositional process (score, other sound controllers and effects, other spatial information...)

As we will show in the following sections, the tools presented in this paper address these different points through an original combination of computer-aided composition and real-time spatialization software.

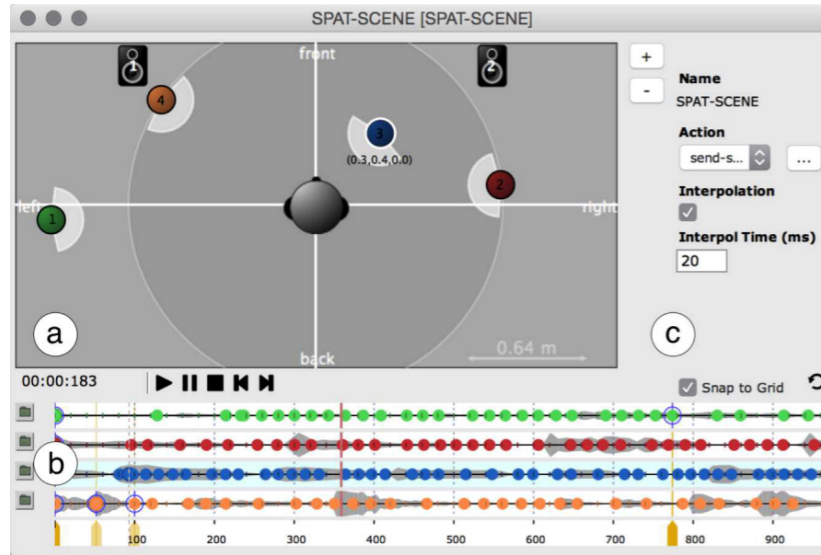
## 4 SPAT-SCENE: Interacting with sound scenes in space and time

SPAT-SCENE is a graphical user interface (GUI) based on the *Spatialisateur* library [Jot, 1999, Carpentier et al., 2015] and implemented in the OpenMusic visual programming, computer-aided composition environment [Bresson et al., 2011]. It takes advantage of the powerful real-time audio engine of the *Spatialisateur* and of its graphical controls (that are familiar to many computer music composers), and integrates them in a compositional framework. SPAT-SCENE provides orthogonal views of the space vs. time dimensions of spatial



sound scenes: it allows the spatial layouts of sound sources to be specified at specific points in time, and the timed trajectories for the individual sound sources to be edited or synchronized.

Figure 4 presents a SPAT-SCENE interface containing four sources associated with four sounds and four trajectories, respectively. The interface is divided into three main areas: (a) the *sound scene* view, (b) the



**Figure 4:** SPAT-SCENE graphical user interface: (a) scene view including four sources, two speakers and a representation of the listener’s virtual position; (b) timeline; and (c) settings panel.

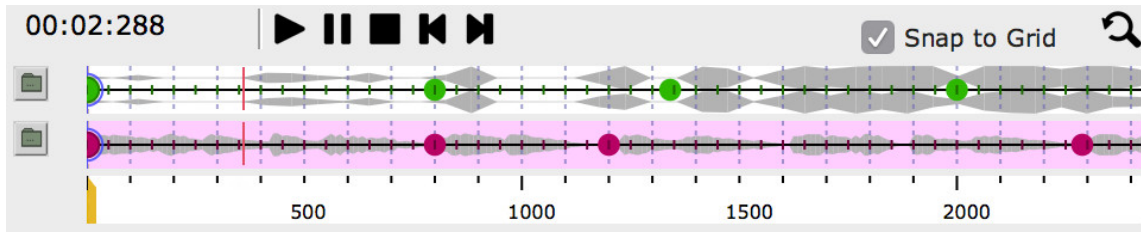
timeline section, and (c) the *settings* panel. The scene view is handled by the Ircam *Spatialisateur* library (see Section 6) and depicts the sources and speakers spatial characteristics in a traditional 2-D representation (showing the listener at the center of the auditory space, and numbered sound sources around it). The sound sources can be manually displaced on the graph, setting their position at a given moment in time (this “current” time is represented by a vertical bar in the timeline underneath).

The timeline section contains, for each source, an individual timeline displaying the waveform of the associated sound and the timing of all position changes (similarly to “key frames” in video editing tools). A toolbar above this panel offers buttons to play, pause, stop and navigate through the time positions. Finally the settings panel provides controls to add or remove sources, to specify the interpolation time between keyframes and to select a rendering method (see Section 4.4).

#### 4.1 Timelines and spatio-temporal specification

The main data structure behind SPAT-SCENE is a set of 3-D trajectories, each trajectory being a sequence of timed 3-D points. The timeline interface provides interactive controls to manipulate the temporal dimension of each of these trajectories. Figure 5 depicts a timeline view with two sound sources. Each point of the trajectory is represented as a circle, indicating its timestamp with the same color as the trajectory. The time ruler at the bottom displays the visible time range (in milliseconds) and can be used to navigate or zoom in and out in the timeline.

When a sound source is moved in the sound scene view, its position at the current time is updated in the



**Figure 5:** Timeline with audio waveforms in the background and ticks for interpolation times.

trajectory. If no point was defined for that source at this specific moment, a new one is automatically inserted in the trajectory. Users can also insert new points directly in the timeline. In this case the new point is created with interpolated spatial coordinates ensuring speed continuity in the trajectory. Points can be moved over the timeline as well, so as to apply a temporal shift. The actual timing of the dragged points can either be free or snapped to a temporal grid or to points of other trajectories. Finally, the timeline lets users set specific points called “master” which define specific portions of the trajectories where timing can be stretched or compressed. By default, the start and the end points are defined as master points to enable the scaling of the overall duration of a trajectory.

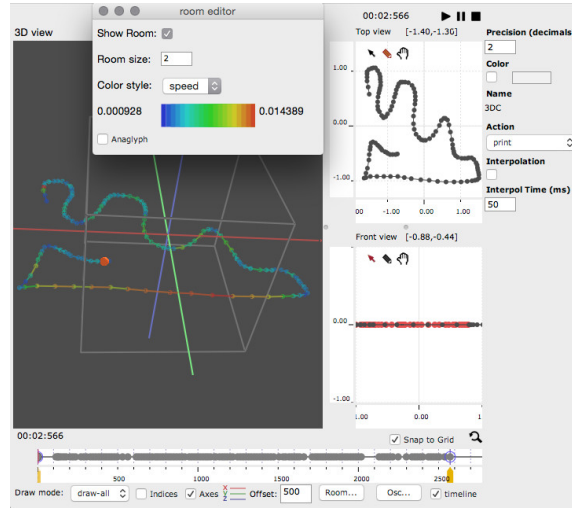
## 4.2 Editing trajectories

Trajectories in SPAT-SCENE can also be displayed and edited as 3-D curves. The 3-D editor embedded in SPAT-SCENE is based on the previously available trajectory interfaces in OpenMusic (3DC object, see [Bresson, 2012]) but features several improvements for interacting with the temporal data, visualizing motion and supporting gestural input (see Section 4.3). Figure 6 shows the graphical user interface of the trajectory editor, which includes a 3-D representation and synchronized 2-D projections displaying top and front perspectives.

Composers can use the 2-D views to draw points and edit their positions. While drawing a curve within one of these views, the motion timing is recorded and defines the time values of the new points. An individual timeline at the bottom of the interface supports the same time manipulations as the one in SPAT-SCENE, such as edition of individual points or global/local time scaling. A setup panel, also visible in Figure 6, provides access to the different visualization options (e.g. mapping a color scale to its duration or instantaneous speed).

## 4.3 Gestural Input

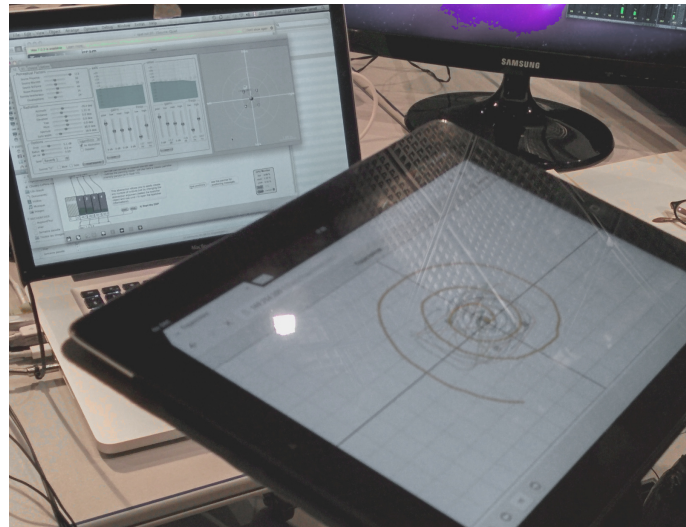
The interviews with composers (Section 3.1) highlighted a potential interest in gestural approach for the input and prototyping of trajectories and spatial patterns. In order to let composers use external gestural controllers and devices to define trajectories, we developed an OSC server [Wright, 2005] integrated into the 3DC interface that can receive and process data streams. The protocol is simple and consists of two main messages: the *move* message that simply translates the sound source in the 3-D space, and the *add* message that inserts points in the corresponding trajectory. The graphical user interface displays a green or red sphere in the 3-D view when the user moves or draws a point in the trajectory, respectively. The primary goal of this protocol is to facilitate the sketching of trajectories, without considering complex mapping strategies (as proposed for instance in previous works on gestural control — see e.g. [Marshall et al., 2009]). Two different kinds of devices have been tested and connected to our software framework in order to input and process



**Figure 6:** Velocity visualization of a trajectory using color coding in the 3-D editor. The color scheme ranges from slow/blue to fast/red.

spatial data:

- **Tablets and tactile mobile devices**, using *Trajectoires* [Garcia et al., 2016], a mobile application featuring curve drawing, storage and playback functionalities (see Figure 7).
- **Game controllers**, using *GameTrak*, a gestural controller consisting of two wires anchored and articulated to a base unit, delivering 3DOF positions derived from the length and angle of these wires.



**Figure 7:** Input of trajectories using a mobile device.

Experiments with these gestural controllers to draw 2-D and 3-D trajectories have put forward a number of interesting characteristics. Mobile devices are convenient as they allow for fast and easy on-site experimentation with sound positioning and motion. Composers can interact with the positioning of sound sources,

input new motions and patterns, while moving in the performance space and assessing the perceptive effect and results of the spatialization processes at different spots in the room. The game controller experience, on the other hand, allowed to record the gesture performing mid-air hand movements to move sources and “draw” trajectories, and to explore sound source motion in 3-D [Garcia et al., 2015a]. Other types of devices can of course be used with the server, so that composers can plug their own tools to SPAT-SCENE and explore other forms of gestural interaction.

## 4.4 Playback and Rendering

The SPAT-SCENE object can be rendered in different ways, providing auditory feedback and/or visualization of the evolution of the sound scene with time. In the context of compositional vs. real-time interactions, it can act as a controller or monitoring interface for the real-time spatialization processes.

While playing, the active time positions of each trajectory in the timeline are progressively highlighted, giving a clue of the current time and state of the overall scene. The spatial display (scene view) is also updated periodically to show the positions of the sources.

A combo box in the setting panel allows to choose an action to be executed during playing-back of the trajectories: depending on compositional situations, the SPAT-SCENE can either render a spatialized audio, or control external rendering engines via OSC messages.

**Audio preview in OpenMusic:** The audio rendering action uses the *Spatialisateur* engine to synthesize multichannel audio from the SPAT-SCENE. The OpenMusic scheduling and audio rendering system slices short buffers from the sound sources that are processed by the *Spatialisateur* and then played-back on the audio interface. Users can choose among the different rendering techniques available in the *Spatialisateur* such as binaural or amplitude panning, depending on their reproduction system (headphones or loudspeakers).

**External Rendering via OSC:** This other option streams the time-tagged positions of all sound sources through OSC messages, which can be interpreted by external rendering engines. This scenario is more likely to occur when composers need to use a specific audio renderer (possibly dedicated to a specific performance venue) or to apply other audio transformations in addition to the spatialization process. Figure 8 presents an example in which a Max version of *Spatialisateur* is being controlled by a SPAT-SCENE containing 8 sources.

Besides audio preview and OSC streaming, user-defined functions can also be programmed in OpenMusic and assigned to SPAT-SCENE as playback actions.

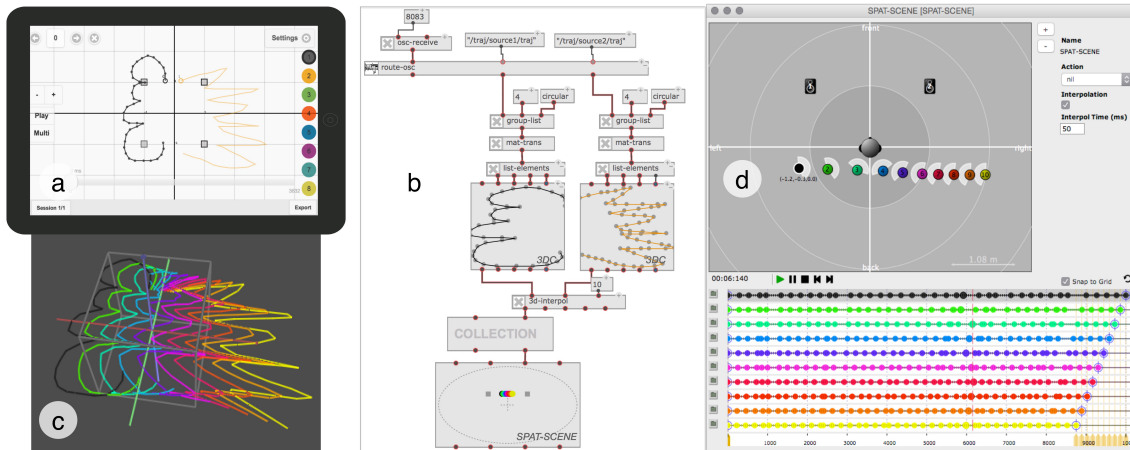
## 5 Integration in the computer-aided composition workflow

As an element of the OpenMusic framework, SPAT-SCENE can be integrated with larger-scale compositional processes. In this section we describe how composers can use visual programs in OpenMusic to feed a SPAT-SCENE, and how the SPAT-SCENE can be embedded and articulated with other musical structures and processes.



(rightmost) has been generated algorithmically. The design of arbitrary processes generating or transforming trajectories in OpenMusic is straightforward and has been already demonstrated in various contexts, e.g. [Schumacher and Bresson, 2010, Garcia et al., 2015b].

Since OpenMusic allows both data from files in different formats to be read, and incoming OSC messages to be received or processed [Bresson, 2014], existing trajectories can also be imported from external tools and applications in order to build a SPAT-SCENE object. Recent experiments carried out with mobile interfaces [Garcia et al., 2015b] have used OpenMusic to produce and render complex spatial sound scenes using rotations, translations or interpolations from initially hand-drawn curves (see Figure 10).



**Figure 10:** Algorithmic processing of hand-drawn input feeding a SPAT-SCENE. a) The Trajectoires mobile application, sending two initial trajectories. b) OpenMusic patch receiving the OSC messages, generating 10 interpolated curves. c) Visualization of all the generated trajectories in a 3-D editor. d) Resulting SPAT-SCENE.

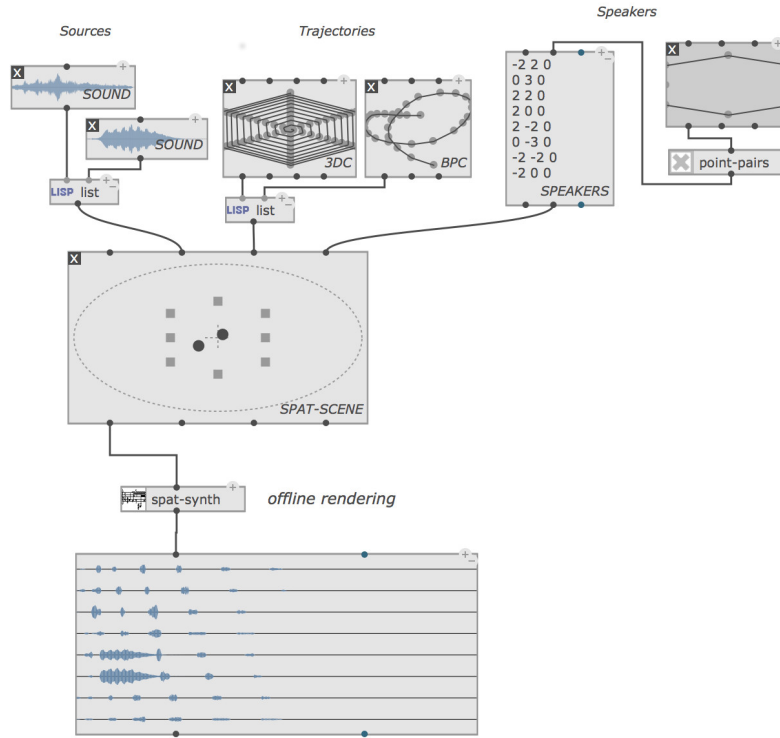
## 5.2 Offline rendering

As a comprehensive representation of a spatial sound scene, SPAT-SCENE can also be unfolded and converted by the *Spatialisateur* audio processor to a multi-channel audio file where each channel corresponds to a speaker in the specified configuration. The function *spat-synth* allows to generate such audio file in an OpenMusic visual program. This function corresponds to the “standard” offline spatialization process as performed for instance by existing OM-Spat or OMPrisma libraries [Bresson, 2012]. Figure 11 shows an example of SPAT-SCENE offline sound rendering.

## 5.3 Articulation with musical and temporal structures

SPAT-SCENES are also likely to be connected with scores or other OpenMusic objects, and arranged in larger-scale time structures. Figure 12 shows a sequencing interface derived from the OpenMusic *maquette*, containing a sound file, a break point function and a SPAT-SCENE with four moving sound sources. Depending on its duration, the SPAT-SCENE display gets divided into one or more keyframes, offering an overview of the temporal evolution of the scene. During playback, as the cursor moves over the display, the successive keyframes get animated to reflect the current state of the scene at all time.





**Figure 11:** Synthesizing a multichannel audio file from SPAT-SCENE (two input sources/trajectories, eight speakers/output audio channels).

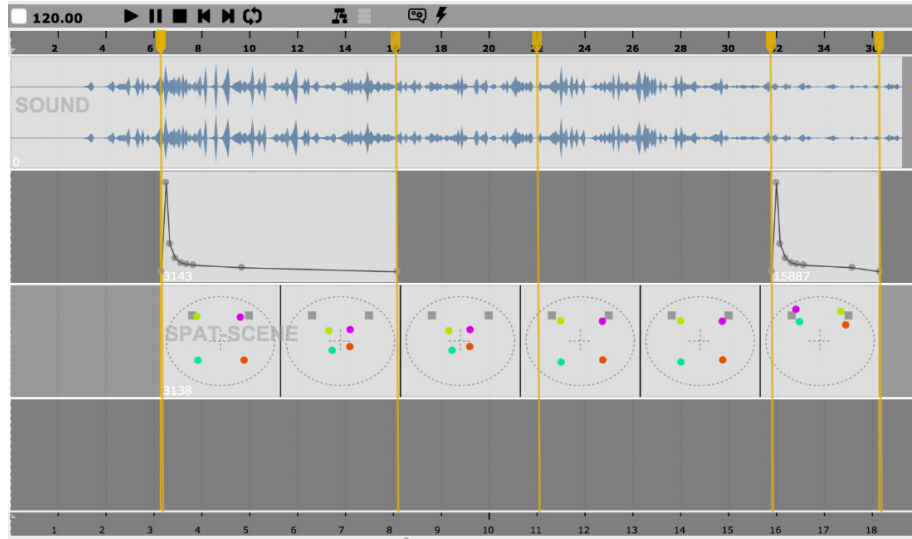
The “master” points in the SPAT-SCENE timelines are lifted to this upper hierarchical level and can therefore be manipulated and synchronized in relation to the other objects and time markers embedded in the time structure.

## 5.4 Exporting to Notation Software

Our observations have revealed a common strategy among composers, in that they manually add visual annotations to the final scores as indications for sound spatialization (see Figure 3 for example). To facilitate this task, the SPAT-SCENE can also be exported as vector graphics representing symbolic descriptions of the scene at a given time or in a given time interval. The graphics are exported as SVG files and imported into music notation software such as Finale or Sibelius. Figure 13 shows an excerpt of a score featuring such graphics, produced from a SPAT-SCENE at two consecutive time intervals.

## 6 Embedding *Spat* components – Technical considerations

SPAT-SCENE constitutes an original and strong coupling between the OpenMusic/Common Lisp environment and a real-time audio processing kernel. The Ircam *Spatialisateur* is a C library, here linked dynamically and connected to the Lisp system through a foreign function interface (FFI). Its graphical interfaces are developed with the Juce C++ library, which provides an efficient and cross-platform framework for the design and



**Figure 12:** SPAT-SCENE embedded in a maquette, displaying several keyframes. The “master points” of the SPAT-SCENE are represented as yellow vertical markers in the maquette to provide visual cues and help stretching and synchronizing temporal elements.



**Figure 13:** Example score containing spatialization indications exported from a SPAT-SCENE. The cross at the center represents the listener, and the black squares represent the speakers. The colored circles represent the sound sources positions at the beginning (white filling) and at the end (colored filling) of the specified interval. The colored lines represent the actual trajectory between these positions.

deployment of GUIs. Graphic components of the library (e.g. the scene view of the SPAT-SCENE editor) were therefore re-used at no cost by sharing OS graphical resources and objects, which minimized the development workload, but also allowed a similar version of the *Spatialisateur*, which many users have previously worked with in Max and other platforms, to be incorporated into OpenMusic.

Each environment handles different kinds of operations and resources; the *Spatialisateur* is dedicated to instantaneous display and rendering, while OpenMusic/SPAT-SCENE deals with the scene structure and temporal dimension but does not interfere with real-time audio streaming. A callback system allows the propagation of actions from the *Spatialisateur* interface to the underlying structure of SPAT-SCENE, and



the synchronization of the corresponding representations. Conversely, SPAT-SCENE and the OpenMusic scheduling system [Bouche and Bresson, 2015] provide timed callbacks to the *Spatialisateur* library while updating displays or while rendering, in order to compute the spatialized sound buffers transferred to the audio output.

An interesting aspect of this cooperation, both for GUI interactions and audio processing, is the generalized use of OSC bundles (containers of OSC-formatted messages) for the communication between the main application and the linked library. Each function call between the two environments passes an OSC bundle as unique parameter. The messages contained in the bundles, although not transmitted via network, are treated as such in order to set the current state of a spatial scene, and perform graphical updates and audio rendering actions on demand. This strategy also minimizes the development costs, since entry points of the library are drastically limited, and no much specific programming has to be done in the library to make it interpreting new external function calls. It also bases the semantics of these external calls on the structure and format of the OSC bundles, and thereby makes it independent from the OpenMusic/*Spatialisateur* software interface design.

## 7 Discussion and Conclusion

Sound spatialization processes often run in the context of real-time frameworks and therefore rarely benefit from larger-scale compositional scope and consideration. The approach we propose aims at producing and timing sound spatialization processes with real-time audio-visual feedback. The seamless collaboration between deferred-time computer-aided composition processes and a real-time rendering and audio processing library allows composers to deal interactively with both space and time domains, bringing together accurate spatial control with musically relevant structures. We consider this work a specific case in a more general reflection about the integration of real-time processing in compositional frameworks, and believe this approach is likely to enhance composers' creative processes by fostering both the expression and evaluation of musical intentions.

The design of SPAT-SCENE was informed by observations of composers' works, which highlighted a number of critical needs, among which we would like to emphasize (1) the easy and intuitive input of spatio-temporal data; (2) graphical cues allowing the visualization of the different characteristics of this data; (3) the temporal control over trajectories and compound spatial scenes; and (4) the connection with compositional processes. Respectively, composers can use SPAT-SCENE to define motion of sound sources with different input methods such as gestural controllers, algorithmic processes or by manually entering and interpolating between keyframes. We also improved existing tools of the OpenMusic environment to provide better visualization features for the motion parameters of individual trajectories. However, our current system does not support the input and control of additional important parameters such as the orientation or the aperture of the sound sources: we are planning to include these parameters following a similar approach. The temporal control issue is mostly addressed via the multiple-timeline interface provided by the SPAT-SCENE editor. Finally, the integration of SPAT-SCENE into OpenMusic visual programs and time structures allows composers to use these tools in a computer-aided composition context.

Even if the composers we interacted with all used source-based spatialization techniques, our interviews also highlighted that the spatial environment of the performance is often taken into account during the composition process: sketches often started with positioning some elements of the venue, the speaker system

or other important environmental aspects prior to drawing the actual trajectories. Figure 1 clearly highlights this point: the composer’s sketches in this figure are designed in relation to a given venue or to a configuration of loudspeakers. Currently, SPAT-SCENE provides basic spatial referential indications and axes, as well as a simple shoebox-shaped room model, but future developments may also allow users to load external 3-D CAD models of the performance venue, or to graphically sketch ad-hoc additional environmental information.

Future work will also investigate different spatialization approaches such as the speaker-centered techniques, building upon other graphical interfaces available within the *Spatialisateur* such as the “matrix” representation, as well as existing interactive applications such as MIAM Spat [Le Vaillant et al., 2015]. We will also pursue collaborations with composers in order to refine and assess these tools in real composition and production contexts.

## References

- [Agon et al., 2011] Agon, C., Bresson, J., and Stroppa, M. (2011). OMChroma: Compositional control of sound synthesis. *Computer Music Journal*.
- [Agostini and Ghisi, 2013] Agostini, A. and Ghisi, D. (2013). Real-Time Computer-Aided Composition with bach. *Contemporary Music Review*, 32(1):41–48.
- [Baalman, 2004] Baalman, M. A. (2004). Application of Wave Field Synthesis in electronic music and sound installations. In *Proc. of the Linux Audio Conference*, Karlsruhe, Germany.
- [Baalman, 2007] Baalman, M. A. (2007). On wave field synthesis and electro-acoustic music - State of the art 2007. In *Proc. of the International Computer Music Conference*, Copenhagen, Denmark.
- [Baalman, 2010] Baalman, M. A. (2010). Spatial Composition Techniques and Sound Spatialisation Technologies. *Organised Sound*, 15(3):209 – 218.
- [Barrett, 2012] Barrett, N. (2012). The Perception, Evaluation and Creative Application of High Order Ambisonics in Contemporary Music Practice. *Ircam Musical Research Residency report*.
- [Bouche and Bresson, 2015] Bouche, D. and Bresson, J. (2015). Planning and Scheduling Actions in a Computer-Aided Music Composition System. In *Proc. of Scheduling and Planning Applications Workshop (SPARK)*, Jerusalem, Israel.
- [Bozzolan and Cospito, 2008] Bozzolan, M. and Cospito, G. (2008). Smusim: a prototype of multichannel spatialization system with multimodal interaction interface. In *Proc. of the 8<sup>th</sup> International Conference on New Interfaces for Musical Expression (NIME08)*, page 24.
- [Braasch et al., 2008] Braasch, J., Peters, N., and Valente, D. L. (2008). A Loudspeaker-Based Projection Technique for Spatial Music Applications Using Virtual Microphone Control. *Computer Music Journal*, 32(3):55 – 71.
- [Bresson, 2012] Bresson, J. (2012). Spatial Structures Programming for Music. In *Proc. of Spatial Computing Workshop (colocated w. AAMAS’12)*, Valencia, Spain.
- [Bresson, 2014] Bresson, J. (2014). Reactive Visual Programs for Computer-Aided Music Composition. In *Proc. of the IEEE Symposium on Visual Languages and Human-Centric Computing*, pages 141 – 144, Melbourne, Australia.

- [Bresson et al., 2011] Bresson, J., Agon, C., and Assayag, G. (2011). OpenMusic: Visual Programming Environment for Music Composition, Analysis and Research. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 743–746. ACM.
- [Carpentier et al., 2015] Carpentier, T., Noisternig, M., and Warusfel, O. (2015). Twenty Years of Ircam Spat: Looking Back, Looking Forward. In *Proc. of the International Computer Music Conference*, pages 270 – 277, Denton, TX, USA.
- [Celerier et al., 2016] Celerier, J.-M., Desainte-Catherine, M., and Couturier, J.-M. (2016). Outils d’écriture spatiale pour les partitions interactives. In *Proc. of Journées d’Informatique Musicale*, pages 82 – 92, Albi, France.
- [Chowning, 1971] Chowning, J. M. (1971). The Simulation of Moving Sound Sources. *Journal of the Audio Engineering Society*, 19(1):2 – 6.
- [Coffy et al., 2014] Coffy, T., Giavitto, J.-L., and Cont, A. (2014). AscoGraph: A User Interface for Sequencing and Score Following for Interactive Music. In *Proc. of the 40<sup>th</sup> International Computer Music Conference (ICMC)*, Athens, Greece.
- [Cont, 2008] Cont, A. (2008). ANTESCOFO: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music. In *Proc. of the International Computer Music Conference (ICMC)*, pages 33–40, Belfast, Ireland.
- [Delerue, 2004] Delerue, O. (2004). *Spatialisation du son et programmation par contraintes : le système MusicSpace*. PhD thesis, Université de Paris VI.
- [Ellberger et al., 2015] Ellberger, E., Toro-Perez, G., Schuett, J., Cavaliero, L., and Zoia, G. (2015). A Paradigm for Scoring Spatialization Notation. In *Proc. 1<sup>st</sup> International Conference on Technologies for Music Notation and Representation (TENOR)*, Paris, France.
- [Favory et al., 2015] Favory, X., Garcia, J., and Bresson, J. (2015). Trajectoires: a mobile application for controlling and composing sound spatialization. In *Proc. of the 27<sup>ème</sup> conférence francophone sur l’Interaction Homme-Machine.*, page a5, Toulouse, France. ACM.
- [Flanagan, 1954] Flanagan, J. C. (1954). The critical incident technique. *Psychological Bulletin*, 51(4).
- [Garcia et al., 2015a] Garcia, J., Bresson, J., and Carpentier, T. (2015a). Towards Interactive Authoring Tools for Composing Spatialization. In *Proc. of the IEEE 10<sup>th</sup> Symposium on 3D User Interfaces*, Arles, France.
- [Garcia et al., 2015b] Garcia, J., Bresson, J., Schumacher, M., Carpentier, T., and Favory, X. (2015b). Tools and Applications for Interactive-Algorithmic Control of Sound Spatialization in OpenMusic. In *Proc. of inSONIC2015, Aesthetics of Spatial Audio in Sound, Music and Sound Art*, Karlsruhe, Germany.
- [Garcia et al., 2016] Garcia, J., Favory, X., and Bresson, J. (2016). Trajectoires: a Mobile Application for Controlling Sound Spatialization. In *Proc. of CHI EA ’16: ACM Extended Abstracts on Human Factors in Computing Systems.*, page 4, San Jose, United States.
- [Gottfried, 2015] Gottfried, R. (2015). SVG to OSC Transcoding as a Platform for Notational Praxis and Electronic Performance. In *Proc. of the 1<sup>st</sup> International Conference on Technologies for Music Notation and Representation (TENOR)*, pages 154 – 161, Paris, France.
- [Harley, 1994] Harley, M. A. (1994). *Space and Spatialization in Contemporary Music : History and Analysis, Ideas and Implementations*. PhD thesis, McGill University, School of Music.
- [Harrison, 1998] Harrison, J. (1998). Sound, space, sculpture: some thoughts on the ‘what’, ‘how’ and ‘why’ of sound diffusion. *Organised Sound*, 3(2):117 – 127.

- [Jacquemin et al., 2012] Jacquemin, G., Coduys, T., and Ranc, M. (2012). Iannix 0.8. In *Actes des Journées d’Informatique Musicale (JIM 2012)*, pages 107 – 115, Mons, Belgium.
- [Jot, 1999] Jot, J.-M. (1999). Real-time spatial processing of sounds for music, multimedia and interactive human-computer interfaces. *Multimedia Systems*, 7(1):55–69.
- [Kendall, 1995] Kendall, G. S. (1995). The decorrelation of audio signals and its impact on spatial imagery. *Computer Music Journal*, 19(4):71 – 87.
- [Kendall, 2010] Kendall, G. S. (2010). Spatial perception and cognition in multichannel audio for electroacoustic music. *Organised Sound*, 15(3):228 – 238.
- [Kim-Boyle, 2006] Kim-Boyle, D. (2006). Spectral and Granular Spatialization with Boids. In *Proceedings of the International Computer Music Conference*, pages 139 – 142, New Orleans, LA, USA.
- [Laurson et al., 2005] Laurson, M., Norilo, V., and Kuuskankare, M. (2005). PWGLSynth: A Visual Synthesis Language for Virtual Instrument Design and Control. *Computer Music Journal*, 29(3).
- [Le Vaillant et al., 2015] Le Vaillant, G., Giot, R., and Muñoz Gomez, R. (2015). Interface Multimodale pour la Spatialisation de Musique Acousmatique. In *Actes des Journées d’Informatique Musicale*, Montréal, Canada.
- [Madden, 2014] Madden, A. (2014). Developing spaceJam: The New Sound Spatialization Tool for an Artist and Novice. Master’s thesis, Steinhardt School of Culture, Education, and Human Development New York University.
- [Marshall et al., 2009] Marshall, M. T., Malloch, J., and Wanderley, M. M. (2009). Gesture control of sound spatialization for live musical performance. In *Gesture-Based Human-Computer Interaction and Simulation*, pages 227–238. Springer.
- [McLeran et al., 2008] McLeran, A., Roads, C., Sturm, B. L., and Shynk, J. J. (2008). Granular Sound Spatialization Using Dictionary-Based Methods. In *Proc. of the 5<sup>th</sup> Sound and Music Computing Conference*, Berlin, Germany.
- [Normandeau, 2009] Normandeau, R. (2009). Timbre spatialisation: The medium is the space. *Organised Sound*, 14(3):277 – 285.
- [Normandeau, 2015] Normandeau, R. (2015). Octogris2 et ZirkOSC2 : outils logiciels pour une spatialisation sonore intégrée au travail de composition. In *Proc. of Journées d’Informatique Musicale (JIM)*, Montréal, Canada.
- [Olaiz et al., 2009] Olaiz, N., Arumi, P., Mateos, T., and Garcia, D. (2009). 3D-Audio with CLAM and Blender’s Game Engine. In *Proc. of the Linux Audio Conference*, Parma, Italy.
- [Otondo, 2008] Otondo, F. (2008). Contemporary trends in the use of space in electroacoustic music. *Organised Sound*, 13(01):77–81.
- [Penha and Oliveira, 2013] Penha, R. and Oliveira, J. P. (2013). Spatium, Tools for sound spatialization. In *Proc. of the Sound and Music Computing Conference*, Stockholm, Sweden.
- [Peters et al., 2011] Peters, N., Marentakis, G., and McAdams, S. (2011). Current technologies and compositional practices for spatialization: A qualitative and quantitative analysis. *Computer Music Journal*, 35(1):10–27.
- [Pottier, 1998] Pottier, L. (1998). Dynamical spatialization of sound. HoloPhon : a graphic and algorithmic editor for Sigma1. In *Proc. of the International Conference on Digital Audio Effects (DAFx-98)*, Barcelona, Spain.

- [Puckette, 2004] Puckette, M. (2004). A divide between ‘compositional’ and ‘performative’ aspects of Pd. In *Proc. of the 1<sup>st</sup> International Pd Convention*, Graz, Austria.
- [Schacher, 2010] Schacher, J. C. (2010). Seven years of ICST ambisonics tools for Max/MSP - a brief report. In *Proc. of the 2<sup>nd</sup> International Symposium on Ambisonics and Spherical Acoustics*, Paris, France.
- [Schnell et al., 2009] Schnell, N., Röbel, A., Schwarz, D., Peeters, G., and Borghesi, R. (2009). MUBU & friends - Assembling tools for content based real-time interactive audio processing in MAX/MSP. In *Proceedings of International Computer Music Conference*, pages 423–426.
- [Schumacher and Bresson, 2010] Schumacher, M. and Bresson, J. (2010). Spatial Sound Synthesis in Computer-Aided Composition. *Organised Sound*, 15:271–289.
- [Todoroff et al., 1997] Todoroff, T., Traube, C., and Ledent, J.-M. (1997). NeXT-STEP Graphical Interfaces to Control Sound Processing and Spatialization Instruments. In *Proc. of the International Computer Music Conference (ICMC)*, Thessaloniki, Greece.
- [Topper et al., 2002] Topper, D., Burtner, M., and Serafin, S. (2002). Spatio-Operational Spectral (S.O.S) Synthesis. In *Proc. of the International Conference on Digital Audio Effects (DAFx-02)*, Hamburg, Germany.
- [Wagner et al., 2014] Wagner, D., Brümmer, L., Dipper, G., and Otto, J. A. (2014). Introducing the Zirkonium MK2 System for Spatial Composition. In *Proc. of the International Computer Music Conference (ICMC) / Sound and Music Computing (SMC)*, pages 823 – 829, Athens, Greece.
- [Wilson and Harrison, 2010] Wilson, S. and Harrison, J. (2010). Rethinking the BEAST: Recent developments in multichannel composition at Birmingham ElectroAcoustic Sound Theatre. *Organised Sound*, 15(3).
- [Wright, 2005] Wright, M. (2005). Open Sound Control: An Enabling Technology for Musical Networking. *Organised Sound*, 10(3):193–200.