

3DJ: A SUPERCOLLIDER FRAMEWORK FOR REAL-TIME SOUND SPATIALIZATION

Andres Perez-Lopez

contact@andresperezlopez.com

ABSTRACT

The field of real time sound spatIALIZATION is recently receiving much attention, as suggested by the large number of proposals appeared in last years - both from software spatIALIZATION frameworks and from hardware spatIALIZATION interfaces. However, most of the proposed works do not take into account the existing knowledge in Human Computer Interaction Design, which causes them to remain in a simplified approach. We propose a theoretical basis for real-time spatIALIZATION design from a holistic perspective, based on the Digital Musical Instruments theory, and use it to provide a comparative review of recent proposals. Furthermore, we develop our own state-of-the-art software spatIALIZATION system, 3Dj, which may help in the task of design and evaluation of new proposals for real-time sound spatIALIZATION in the fields of interactive performance, data sonification or virtual environments.

1. INTRODUCTION

Spatial position is one of the perceivable sound aspects. However, in opposition to other qualities such as frequency or amplitude, spatial position did never become a key element of western musical language and theory, due to different reasons.

Despite that fact, several historical proposals which used *space* as a compositional element can be found. *Polychoral Antiphony* is considered one of the oldest practices, which consisted of several choirs singing simultaneously in different church locations; Alessandro Striggio's *Missa sopra Ecco si beato giorno* for five choirs might be one of the most relevant pieces [1].

Spatial sound has not adopted a main role with the advent of recording and playback capabilities. The two-channel stereophony, which might be considered as the *de facto* standard speaker configuration, limits the sonic image to the line between speakers, thus reducing drastically the spatial dimension of sound. Therefore, spatial sound was mainly explored from the electroacoustic composition point of view - Edgar Varèse's *Poème Electronique* might be one of the most noticeable contributions.

Since the 70s, an increasing amount of spatial audio developments have appeared. Both scientific aspects, as for example the Ambisonics theory [2], and aesthetic ones, as can be the case of the *Acousmonium*, contributed to the growing interest in *sound difussion*. Furthermore, the constant increase in computational power also contributed to the widespread and adoption of spatial sound in different fields and contexts.

Spatial sound is a common element in the audiovisual industry nowadays. An example of this fact might be the importance of spatial sound in cinema productions, or the existing commercial *surround* 5.1, 7.1, 10.2 reproduction systems. Along with them, main Digital Audio Workstations allow spatial sound post-processing through the use of plugins. In such cases, sound spatIALIZATION is performed offline.

The usage of sound spatIALIZATION in a real-time (online) environment is though still not fully exploited. Cannon and Favila point that this fact might be due to the "*control complexity required to perform spatial motion*" [3]. Nevertheless, in a study carried by Peters et al., electroacoustic composers working with spatial sound rated "Spatial Rendering in Real-Time" as one of the most desirable features for compositional frameworks [4].

But the possibilities of live 3D sound are not limited to electroacoustic music performance. Indeed, as the *Special Theme* of ICAD 2015 suggests ("*ICAD in Space: interactive spatial sonification*"), real-time spatIALIZATION brings new perspectives and possibilities to a variety of fields such as exploratory data sonification, virtual/augmented reality environments or auditory display.

2. SPATIALIZATION SYSTEMS

2.1. Abstract Representation of Spatial Sound

As mentioned earlier, two-channel stereophony (*stereo*) might be the most common sound spatIALIZATION technique nowadays. In stereo panning, the level and the phase of each channel is adjusted, so that the perceived sound is positioned on an imaginary line between the speakers. The same technique is used in the modern *surround* systems.

However, such techniques present a great drawback: for a reliable reproduction of existing spatial sound material, the speaker layout must be exactly the one for which the material was produced. Additional speakers will not provide any extra information in the playback stage, and less speakers will lead to an information loss. Furthermore, a different speaker placement will cause a positional distortion. Geier et al. classify such technique as *channel-based* [5], and we propose in addition the related term *layout-dependent*.

On the other hand, *object-based* paradigm present a much more flexible scenario. In that way, each different sound source is represented by an object, which is located in a virtual room simulator or *Sound Scene* [5], and contains both audio information and position metadata (as in Figure 1). With such an abstraction layer, it is possible to adapt the corresponding audio output to the current speaker layout (since it is a *layout-independent* paradigm). Modern spatIALIZATION software usually presents an object-based approach due to its flexibility and conceptual easiness.



This work is licensed under Creative Commons Attribution Non Commercial 4.0 International License. The full terms of the License are available at <http://creativecommons.org/licenses/by-nc/4.0>

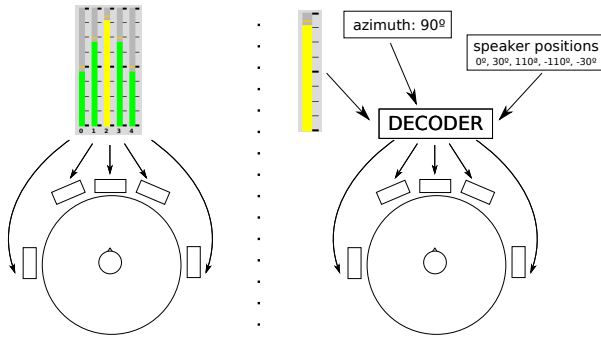


Figure 1: Channel-based, layout-dependent (left) vs. object-based, layout-independent (right) metaphors

2.2. Real-Time Spatialization Systems

In the object-based approach, the sound spatialization task can be divided into two individual domains: a *Scene Simulator*, which describes the spatial position of the objects, and a *Spatial Render*, which actually synthesizes the corresponding sound according to the used spatialization technique (VBAP, Ambisonics, WFS...).

We thus use the concept of *Spatialization System* as a system consisting of a scene simulator and a spatial render. The schema is represented in Figure 2.

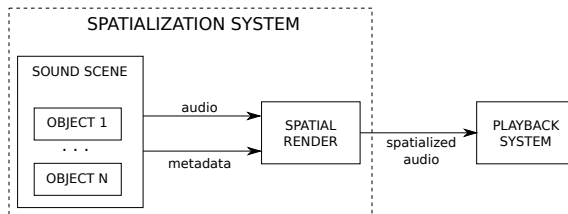


Figure 2: Spatialization System structure

As pointed by Marshall et al., sound spatialization is a multi-dimensional system, featuring 2D spatial position and volume parameters for each sound object in the simplest case [6]. He further proposes the following taxonomy of spatialization system control parameters:

- *Source position*: 2D / 3D spatial coordinates
- *Source characteristics*: Size, directivity, presence, brilliance
- *Environmental parameters*: Room size, reverberation, doppler effect, equalization, air absorption, distance decay

McGee and Wright [7] extend the taxonomy by including some desirable system features. Among them, we can highlight configurable speaker setup (*layout-independent* paradigm), arbitrary number of sources, and support for diverse spatialization techniques.

Schacher [8] presents a very interesting hierarchical perspective on real-time sound spatialization, which borrows concepts from 3D modelling. In his proposal, sound objects present their own *physical behavior*, which may include both trajectories and physical modelling. In fact, despite the lack of a *canonical set* of dynamic behaviors, many authors have developed a variety of proposals for predefined motions; Schmele offers a broad review of these dynamic approaches [9, chapter 2.3].

Schacher further defines two complementary interaction modes with the sound scene:

Top-down Users have a total direct control of the sound scene (the most common approach).

Bottom-up Users interact from *inside* the sound scene; their actions behave according to the scene's physical rules, thus losing an absolute control of the scene.

Since sound object parameters are described by metadata, it may be possible to think of a common, system-independent scene description format. The SpatDIF protocol [10] is a mature proposal towards the standardization of sound scene description, storage and transmission. Such protocol may overcome the historical lack of system interoperability [5, 10].

2.3. Comparative of Real-Time Spatialization Systems

Table 1 presents a review of the most relevant current systems for real-time sound spatialization, and compares them with respect to the parameters and features presented in Section 2.2. Spatialization systems are organized according to their implementation characteristics. Systems located at the left side are standalone applications: Zirkonium [11], Spat [12], Sound Scape Render [13] and Sound Element Spatializer [14]. Systems on the right side are based on existing sound processing environments: BEASTMulch [15] (SuperCollider), Spatium [16] (Max/MSP) and OMPrisma [17] (OpenMusic).

From the table we can extract some conclusions:

- There is a general lack of dynamic behavioural features. When implemented, the systems only provide a small subset of predefined motions, and physical modelling is unsatisfactory in terms of appearance.
- None of the systems present a hierarchical structure of sound objects. Therefore, they only feature the top-down interaction mode.
- All systems are highly dependent on OSX operating systems. In fact, the only two systems which are also compatible with GNU/Linux presented major problems: SSR only allows 2D spatialization, among other limitations, and SES is not available.
- I must finally highlight the importance of Free Software implementations, which is partially covered by the systems. Free software licenses ensure the availability of code and the potential for improvement, as well as provide the basis for experimental replications; all these characteristics are considered as fundamental for a scientific research environment.

Table 1: Comparative of Real-Time Spatialization System Software

	Standalone systems				SPE based systems		
	Zirkonium	Spat	SSR	SES	BEAST Mulch	Spatium	OM Prisma
3D position	✓	✓	×	✓	✓	✓	✓
Source size	✓	?	×	✓	?	?	?
Source directivity	?	?	×	×	?	?	?
Room parameters	?	✓	×	×	?	✓	✓
Distance cue	×	✓	✓	✓	?	✓	✓
Configurable speaker setup	✓	✓	✓	✓	✓	✓	✓
Arbitrary number sound sources	?	?	?	✓	?	× (16)	?
Behavior support	?	?	×	×	?	✓	✓
Hierarchies support	×	×	×	×	×	×	×
Render type	VBAP	HRTF VBAP HOA	HRTF VBAP HOA* WFS	DBAP VBAP HOA WFS	?	VBAP HOA*	VBAP HOA DBAM ViMiC
OSC	✓	×	?	✓	?	✓	✓
Description format	?	SpatDIF	ASDF	SpatDIF	?	?	SpatDIF
Platform	OSX	OSX	Linux/OSX	Linux/OSX	OSX	OSX	OSX
License	BSD	proprietary	GPL	proprietary	GPL	CC	LGPL

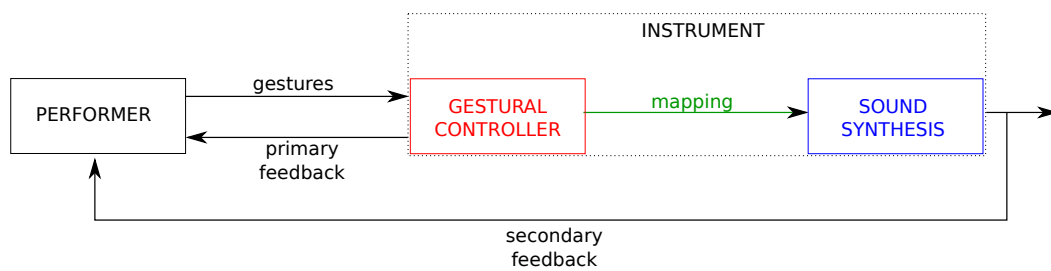


Figure 3: Digital Musical Instrument (by [18])

3. DIGITAL MUSICAL INSTRUMENTS

3.1. Concept

We commonly understand traditional *Musical Instruments* as devices capable to produce live music, which transform the performer gestures into sound events in real-time [19].

On the other hand, the *Digital Musical Instrument (DMI)* concept [18] defines a device in which the *gestural interface* is completely decoupled from the *sound synthesis*¹; their control parameters are related through *mapping* strategies. Furthermore, the produced sound may be potentially any imaginable one, since it is not restricted by the physical and acoustical constraints of conventional instruments [20].

Figure 3 schematizes the structure of a Digital Musical Instrument, as proposed by Wanderley [18].

In words of Jordà, “*performing music with ‘intelligent devices’ [Digital Musical Instruments] tends towards an interactive dialogue between instrument and instrumentalist*” [20].

Interactivity is, according to Winkler [21], a continuous quality, determined by both the amount of freedom given to the performer, and the computer’s ability to respond in an appropriate manner. That response is represented in Figure 3 via the *feedback* arrows:

- *Primary feedback*: Visual, tactile and/or haptic feedback provided by the input controller.
- *Secondary feedback*: Auditive feedback provided by the sound generator.

It is also possible to classify feedback according to its physical nature: *passive*, when it is a consequence of the physical characteristics of the instrument; or *active*, as if the response follows a predefined pattern.

3.2. Design Considerations

Several aspects might be taken into account for a proper interaction design.

Multithread - shared control When many control parameters are available, the performer may not have the capability of controlling every parameter simultaneously. In this scenario, *multithread* and *shared control* paradigms [22] provide the capacity of taking and leaving the control of processes at will, ensuring that their normal behaviour is continued without the user’s direct control.

Intended users The interactive experience may vary depending on the user approach: casual users expect only *enjoying a positive experience*, but trained ones might search some kind of *expressivity* [19].

Number of performers Traditionally, digital musical instruments have been designed for a single trained performer. However, in special cases, multiple performers are considered, as in the context of *interactive installations*.

¹The existence of digital musical instruments whose control and synthesis modules might be coupled is possible. However, in that case, such instrument could not be considered a Digital Musical Instrument.

3.3. DMI for Sound Spatialization

Marshall et al. explore theoretically the problem of interactive sound spatialization [6]. They propose a performer role classification, based on the required *cognitive load*.

- Performer controls only sound spatialization.
- Performer controls both sound synthesis and spatialization.
- Performer *directly* controls sound synthesis, and spatialization is performed *indirectly* or *unconsciously*.

3.4. Review of Spatialization DMI

Table 2 performs a comparative analysis of DMI for sound spatialization, according to the parameters commented in this section. Selected DMI proposals are taken from recent NIME and ICMC conferences: Bokowiec [23], Bredies [24], Cannon [3], Caramiaux [25], Carlson [26], Fohl [27], Johnson [28], Marentakis [29], Marshall [30, 6], Nixdorf [31] and Park [32].

We can extract some common trends by analysing Table 2. Most of the comments may be explained by the lack of background in interaction and DMI design fields.

- Most of the proposals do not take into account the multi-thread/shared control paradigms; in Bokowiec [23], they are described as *unusual* features. This fact is related with the overall simple 2D position control, avoiding more complex input modalities and parameter mappings.
- There is a general trend on relating individual performances with potential trained users, and multi-user performances with casual, non-expert performers.
- Visual active feedback is preferred for all the implementations. Only Carlson [26] offers haptic active feedback, in his non-standard proposal.
- Most of the proposals did not present or mention any evaluation. This is due to various reasons, among them, the lack of a standard evaluation methodology, and the prototype character of most of proposals may be pointed out.

Table 3 provides a review from the same spatialization DMI proposals, in this case compared according to the parameters of the spatialization system used.

The following comments can be extracted from Table 3:

- First of all, there is a general lack of full periphonic (3D) support, in contrast with the capabilities of the spatialization systems. This is probably due to various reasons, which can be the lack of 3D interfaces [3], the technical limitations (Wave Field Synthesis), or simply the control complexity.
- The historical trend towards direct position control is followed. Even when most of the proposals also allow trajectory control, the use of high-level physical abstractions might lead to innovative proposals, as in the case of the Caramiaux’s particle system [25].
- Only the *Sound Flinger* by Carlson et al. [26] proposes a *bottom-up* interaction mode, in which the user is located *inside* the sound scene. All other proposals follow the traditional *top-down* approach.

Table 2: Comparison of Spatialization Instruments according to DMI design

	Multithread Shared control	Intended User	Number of performers	Rol of performer	Interface	Active feedback	Evaluation
Bokowiec	✓	trained	1	spatialization & synthesis	gesture tracker	×	×
Bredies	×	casual	1/many	spatialization	tabletop	visual	×
Cannon	×	trained	1	spatialization & synthesis	extended	visual	×
Caramiaux	×	casual	1	spatialization & synthesis	gesture tracker	visual	×
Carlson	×	both	1	spatialization	slider	visual/haptic	×
Fohl	✓	?	1	spatialization	gloves	×	×
Johnson	×	casual	1/many	spatialization	tabletop	visual	✓
Marentakis	×	trained	1	spatialization & synthesis	extended	×	×
Marshall1	×	trained	1	spatialization	gesture tracker	×	×
Marshall2	×	trained	1	spatialization	gesture tracker	visual	×
Nixdorf	×	trained	1	spatialization	?	visual	✓
Park	×	both	1/many	spatialization	smartphone	visual	✓

Table 3: Comparison of Spatialization Instruments according to spatialization system parameters

	Periphonic	Control parameters	Trajectories	Hierarchies	Interaction mode	Spatialization technique	Spatialization system
Bokowiec	×	position & trajectories	✓	×	top-down	?	×
Bredies	×	position	✓	✓ (groups)	top-down	(SSR)	SSR
Cannon	×	position & trajectories	✓	×	top-down	FOA	×
Caramiaux	×	physical model	✓	×	top-down	WFS / VBAP	×
Carlson	×	physical model	×	×	bottom-up	VBAP	×
Fohl	×	position & trajectories	✓	×	top-down	WFS	×
Johnson	×	position	×	×	top-down	VBAP	×
Marentakis	×	position	×	×	top-down	VBAP	×
Marshall1	✓	position & environmental	✓	×	top-down	ViMiC	×
Marshall2	✓	position & environmental	×	×	top-down	?	×
Nixdorf	×	position	×	×	top-down	?	×
Park	×	position / casual	×	×	top-down	custom	×

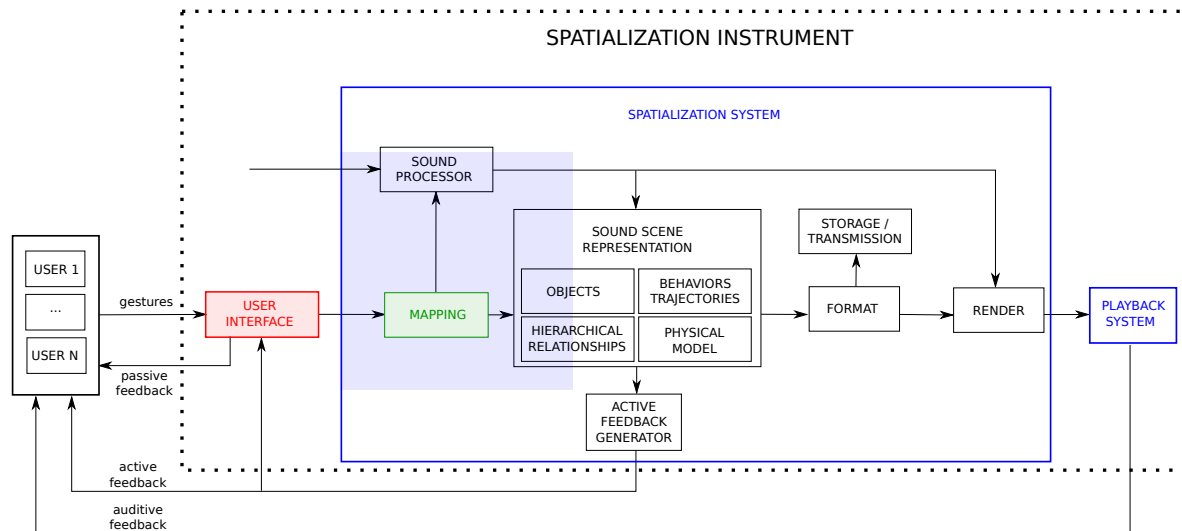


Figure 4: Proposed Spatialization Instrument schema

- Among the spatial sound techniques used, there is only one proposal using First Order Ambisonics; none of them uses Higher Order Ambisonics.
- It is very interesting to observe the general lack of usage of spatialization system frameworks, with the exception of the Bredies [24], in which a spatialization system is used (SSR, which is in fact developed by the same authors). That situation causes the interfaces not to reuse all the capabilities provided by the spatialization systems.
- Finally, we must remark the difficulty of replicating those Spatialization Instruments. In the case of control interfaces, it is clear that such replicability may be difficult or expensive, specially when using custom hardware. However, a common spatialization system might contribute to the Spatialization Instrument reproducibility. That fact might help to ease the creation and usage of standard evaluation methodologies for Spatialization Instruments.

4. SPATIALIZATION INSTRUMENTS

As commented in Section 3.4, instrument designs for sound spatialization usually lack the knowledge of the state of the art of interaction design, adopting a narrow perspective into the problem. In our opinion, a *holistic* approach is preferred and, consequently, we present the concept of *Spatialization Instruments*:

A *Spatialization Instrument* is a Digital Musical Instrument, which has the ability of manipulating the spatial dimension of the produced sound, independently of its capability of producing or manipulating other sound dimensions^{2,3}.

Figure 4 depicts a conceptual structure of the parts composing a Spatialization Instrument. We can appreciate all separate components described in Sections 2 and 3, and also how are they linked to each other.

We must point out that the idea of Spatialization Instruments is a conceptual approach for the design of DMI for sound spatialization. Therefore, instruments presented in Section 3.4 may be considered Spatialization Instruments.

5. SOFTWARE IMPLEMENTATION

Previous sections highlighted the fact that existing spatialization systems do not provide all desired control and interactivity. Furthermore, their availability in terms of cost or multiplatform compatibility is not always ensured. This is the motivation to implement our own spatialization system, with the objective of serving as a tool for Spatialization Instruments development. Moreover, its adoption might help in the replication of the design and evaluation.

²A Spatialization Instrument is a DMI in the sense that it shows the DMI characteristics: interface/sound synthesis decoupling, and potential to produce *any kind of sound*. In that case, *any kind of sound* refers to the fact that the sound position might be any desired, not restricted to physical constraints, and controlled by the user with arbitrary mappings.

³According to the physical laws, any changes in a sound's position will imply temporal and spectral changes as well. We refer here only to the capability to modify non-spatial sound dimensions, in the sense that these other sound dimensions are *control parameters* of the sound synthesis unit.

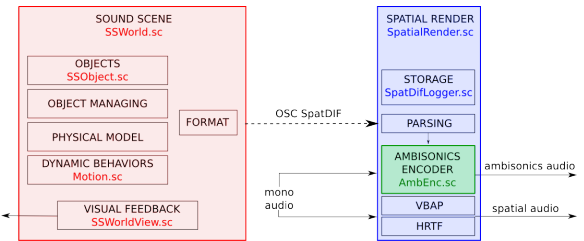


Figure 5: 3Dj internal structure

5.1. Design Specifications

Device independence To provide compatibility with user interface protocols, such as HID, MIDI and OSC.

Flexible mapping To allow arbitrarily complex relations between input gestures and parameters.

Control parameters To provide a variety of potentially relevant control parameters, as reviewed in Section 2.2.

Feedback To integrate visual feedback, and to allow other kinds of active feedback by the user interface protocols.

Spatial render To provide the tools for using different sound spatialization techniques, such as VBAP, HOA, WFS or Binaural.

Exchange format To be compatible with scene description formats, such as SpatDIF.

Modularity The use of standard formats and protocols provide the basis for software modularity, which might be useful for distance or high computing load scenarios.

Free Software The free spreading, modification and understanding of the software is a compulsory requisite for research tools. Furthermore, multiplatform compatibility may be useful for a wide software adoption.

5.2. Implementation: 3Dj

In order to implement the desired features, we opted for the SuperCollider environment [33]. SuperCollider is a real-time audio processor and an object-oriented programming language. It is Free Software and multiplatform, and embraced by a big community of technicians and artists, both users and developers.

Therefore, our spatialization system takes the form of a SuperCollider *quark* (i.e. external library), called 3Dj. The code is available through the internal *quark* installation system, and in its code repository [34].

3Dj follows a modular structure, as can be seen in Figure 5. The main building blocks are the Sound Scene and the Spatial Render, which are connected through a custom implementation of SpatDIF through OSC.

The **Sound Scene** provides the abstraction layer for managing the sound objects. It provides a simple acoustic room model, and a physical model, with parameters such as gravity, medium viscosity or wall friction; it is also possible to interact with the objects by exerting forces to them, thus supporting the *bottom-up* interaction mode.

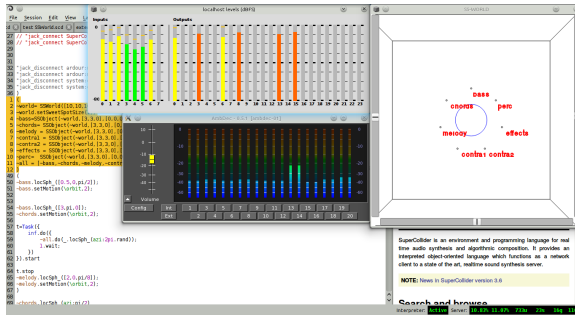


Figure 6: 3Dj environment screenshot

Object grouping and joint management are provided by the SuperCollider language features. Furthermore, there is a set of predefined motions that an object can adopt: for instance, linear, brownian, simple harmonic or orbital. A simple graphical representation of the sound scene is generated, in order to provide direct active feedback (Fig. 6). The sound scene state is saved at a configurable rate, formatted into SpatDIF format, and sent through OSC to the destination Spatial Render.

The **Spatial Render** receives the SpatDIF information, parses it and synthesizes the sound accordingly. Three sound spatialization techniques are available: VBAP, HRTF Binaural, and HOA.

In the case of the latter which is provided by the *AmbEnc* module, the generated audio is in *intermediate Ambisonics format*, which must be further processed by an Ambisonics Decoder; in our case, we provided the wrapper for AmbDec [35] for GNU/Linux systems. Moreover, HOA provides the use of a variety of experimental sound source shapes, different from punctual ones, such as parallel, meridian, or custom spherical surfaces.

Furthermore, the Spatial Render provides SpatDIF *logger* and *playback* capabilities, in order to record the incoming messages, as well as the generated ones, and replay them; this feature might specially be of interest in the case of non-live spatialization.

5.3. Discussion

In this Section we presented our framework for real-time sound spatialization. 3Dj is built upon the desirable features for spatialization systems, as described in Section 2. Furthermore, there is a special focus on the application of that systems for the interactive performance, based on the contents developed in Sections 3 and 4.

Therefore, the tool is primarily addressed to those developers working in Human-Computer Interaction Design and Spatial Sound. In that way, they might benefit from a very flexible, customizable and open-source framework with unique features, and center their efforts in the design tasks. The same concept might also apply to composers, performers and artistic designers.

The Free Software distribution model of 3Dj (through the *General Public License*) facilitates design and experimental replicability, as already commented. The usage of the SpatDIF protocol, together with its storage and playback capabilities, contributes as well to that goal.

Nonetheless, the Auditory Display and Sonification communities may also greatly benefit from the framework. 3Dj, through the SuperCollider scripting language, provides the means to easily perform any arbitrarily complex real-time spatialization task. This feature might be particularly useful for virtual environments, augmented reality or live data sonification.

In addition to that, we must take into account that SuperCollider itself is a highly optimized environment for sound processing and synthesis. Accordingly, any desired sound aspect might be controlled by 3Dj's SuperCollider instance itself, reducing the overall system complexity and interdependency.

It is also true that 3Dj (in fact, the SuperCollider language) may have a longer learning curve than other spatialization frameworks, specially those ones in which user interaction is based on Graphical User Interfaces or Graphical Programming Languages. A manual is conveniently provided for end users [34].

But the initial difficulty is greatly compensated by the arbitrary complexity of sound spatialization mappings and behaviors that might be reached, which surpass features of other spatialization frameworks. Furthermore, the SuperCollider characteristics provide many appealing features: native capabilities of real-time networked remote sound synthesis, native OSC integration, or (as already mentioned) integration with state-of-the-art sound processing, synthesis and classification capabilities.

6. CONCLUSIONS AND FUTURE WORK

This study contributes to the state of the art of real-time sound spatialization in the following manners:

- Proposing the Spatialization Instrument concept as a way to integrate the existing knowledge in interaction design into real-time spatialization domain, thus analysing the problem from a holistic approach.
- Creating a critical review of most recent spatialization systems and Spatialization Instruments.
- Developing a State-of-Art framework for live spatialization.

Regarding the future work, a variety of proposals can be carried in the context of our research:

- To develop different Spatialization Instruments, which may be adequate for diverse use scenarios.
- To research and to develop an evaluation methodology for spatialization instruments; use our spatialization system to perform a case-study evaluation.

7. ACKNOWLEDGMENTS

Thanks to Barcelona Media for allowing the usage of facilities, which was fundamental for the software development. Thanks also to Davide Scaini (Department of Information and Communication Technologies, Universitat Pompeu Fabra) and Daniel Arteaga (Dolby Iberia) for their help in the development process

8. REFERENCES

- [1] D. Moroney, “Alessandro striggio’s mass in forty and sixty parts,” *Journal of the American Musicological Society*, vol. 110, no. 3, pp. 1598–1608, March 2007.
- [2] M. A. Gerzon, “Periphery: With-height sound reproduction,” *Journal of the Audio Engineering Society*, vol. 21, no. 1, pp. 2–10, 1973.
- [3] J. Cannon and S. Favilla, “Expression and spatial motion: Playable ambisonics,” in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2010, pp. 120–124.
- [4] N. Peters, G. Marentakis, and S. McAdams, “Current technologies and compositional practices for spatialization: A qualitative and quantitative analysis,” *Computer Music Journal*, vol. 35, no. 1, pp. 10–27, 2011.
- [5] 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, 2010.
- [6] 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, pp. 227–238.
- [7] R. McGee and M. Wright, *Sound element spatializer*. Ann Arbor, MI: MPublishing, University of Michigan Library, 2011.
- [8] J. C. Schacher, “Gesture control of sounds in 3d space,” in *Proceedings of the 7th international conference on New interfaces for musical expression*. ACM, 2007, pp. 358–362.
- [9] T. Schmele, “Exploring 3d audio as a new musical language,” Ph.D. dissertation, Masters thesis, Universitat Pompeu Fabra, 2011.
- [10] N. Peters, T. Lossius, and J. C. Schacher, “The spatial sound description interchange format: Principles, specification, and examples,” *Computer Music Journal*, vol. 37, no. 1, pp. 11–22, 2013.
- [11] Zirkonium. <http://www.zkm.de/zirkonium>.
- [12] Spat. <http://forumnet.ircam.fr/product/spat/>.
- [13] M. Geier and S. Spors, “Spatial audio with the soundscape renderer,” in *In VDT Convention*, 2012.
- [14] R. M. McGee and M. Wright, “Sound element spatializer,” in *n Proceedings of the International Computer Music Conference 2011*, 2011.
- [15] Beastmulch. <http://www.birmingham.ac.uk/facilities/BEAST/research/mulch.aspx>.
- [16] Spatium. <http://spatium.ruipenha.pt/>.
- [17] Omprisma. <http://www.idmil.org/software/omprisma>.
- [18] M. M. Wanderley, “Gestural control of music,” in *International Workshop Human Supervision and Control in Engineering and Music*, 2001, pp. 632–644.
- [19] S. Jordà, “Digital instruments and players: Part ii—diversity, freedom and control,” in *Proceedings of the International Computer Music Conference*, 2004, pp. 706–710.
- [20] N. Collins and J. d’Escrivan, “Interactivity and live computer music,” in *The Cambridge Companion to Electronic Music*. Cambridge University Press, p. 312.
- [21] T. Winkler, “Making motion musical: Gesture mapping strategies for interactive computer music,” in *ICMC Proceedings*, 1995, pp. 261–264.
- [22] J. Chadabe, “Interactive composing: An overview,” *Computer Music Journal*, pp. 22–27, 1984.
- [23] M. A. Bokowiec, “Voct (ritual): An interactive vocal work for bodycoder system and 8 channel spatialization,” in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2011.
- [24] K. Bredies, N. A. Mann, J. Ahrens, M. Geier, S. Spors, and M. Nischt, “The multi-touch soundscape renderer,” in *Proceedings of the working conference on Advanced visual interfaces*. ACM, 2008, pp. 466–469.
- [25] B. Caramiaux, S. F. Alaoui, T. Bouchara, G. Parsehian, M. Rébillat, et al., “Gestural auditory and visual interactive platform,” in *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11)*, 2011.
- [26] C. Carlson, E. Marschner, and H. McCurry, “The sound flinger: A haptic spatializer,” in *Proceedings of International Conference on New Interfaces for Musical Expression*, 2011, pp. 138–139.
- [27] W. Fohl and M. Nogalski, “A gesture control interface for a wave field synthesis system,” in *International Conference on New Interfaces for Musical Expression*, 2013.
- [28] B. Johnson and A. Kapur, “Multi-touch interfaces for phantom source positioning in live sound diffusion,” in *Proceedings of International Conference on New Interfaces for Musical Expression*, 2013.
- [29] G. Marentakis, N. Peters, and S. McAdams, “Dj spat: Spatialized interactions for djs,” in *Proceedings of the 2007 International Computer Music Conference (ICMC)*, 2007, pp. 360–363.
- [30] M. Marshall, N. Peters, A. R. Jensenius, J. Boissinot, M. M. Wanderley, and J. Braasch, “On the development of a system for gesture control of spatialization,” 2006.
- [31] J. J. Nixdorf and D. Gerhard, “Real-time sound source spatialization as used in challenging bodies: implementation and performance,” in *Proceedings of the 2006 conference on New interfaces for musical expression*. IRCAMCentre Pompidou, 2006, pp. 318–321.
- [32] S. Park, S. Ban, D. Hong, and W. Yeo, “Sound surfing network (ssn): Mobile phone-based sound spatialization with audience collaboration,” in *New Interfaces for Musical Expression*, vol. 2013, 2013, pp. 111–114.
- [33] J. McCartney, “Rethinking the computer music language: Supercollider,” *Computer Music Journal*, vol. 26, no. 4, pp. 61–68, 2002.
- [34] 3Dj quark. <https://github.com/andresperezlopez/3Dj>.
- [35] F. Adriaensen, “Ambdec 0.4. 2 user manual,” *kokkinizita. linuxaudio.org*, 2011.