Real-Time 3D Audio Spatialization Tools for Interactive Performance

Andrés Pérez-López

MASTER THESIS UPF / 2014

Master in Sound and Music Computing

Master Thesis Supervisor:
Daniel Arteaga, Davide Scaini
Department of Information and Communication Technologies
Universitat Pompeu Fabra, Barcelona

Ich würde nur an einen Gott glauben, der zu tanzen verstünde

F. W. Nietzsche
Also sprach Zarathustra

UNIVERSITAT POMPEU I FABRA

Abstract

Department of Information and Communication Technologies
Music Technology Group

Master in Sound and Music Computing

Real-Time 3D Audio Spatialization Tools for Interactive Performance

by Andrés Pérez-López

Spatial position is one of the main perceivable sound characteristics. Due to historical and technological reasons, however, it has been not fully exploited until last decades. Current technologies for compositional, offline spatialization are already consolidated; but the online domain remains still in a development stage, due to the inherent complexity added by the realtime control paradigm. We propose a theoretical basis for the analysis and design of realtime spatialization systems, borrowing the knowledge and tools from the Human Computer Interaction and Digital Musical Instrument design fields, and thus developing the concept of *spatialization instruments*. Based on that, we perform an analysis of both existing realtime sound spatialization software and recently developed interfaces for realtime sound spatialization control. Furthermore, on top of the design criteria provided by the analysis, we developed the *SCLiss: SuperCollider Live Spatialization System*, a new tool for helping in the *spatialization instruments* design process, and proposed an interface prototype as an example of the tool's possibilities.

Acknowledgements

I would like to specially thank my supervisors, Daniel Arteaga and Davide Scaini, for their support along the Thesis development.

Of course, thanks also to Xavier Serra and all MTG members, for all what I have learned here.

Thanks to the Marina d'Or team, the Carmelo Riders and the Canconyet crew. Special thanks to Nahuel for the logistic support.

Finally, many thanks to my classmates, with which we formed a great group.

Contents

List of Figures

List of Tables

To my cat

Chapter 1

Introduction

1.1 Motivation and Contextualization

Spatial position is one of the dimensions of perceived sound. In Western culture, historically, other sound dimensions such as pitch or loudness have conformed the basics of musical language and theory. Despite the objective of current work is not a musicological research on *space in music composition*, we will provide some examples that shows the usage of spatial information as a musical expression method along the music history.

The term *antiphonal music* refers to the liturgical style of singing, in which at least two choirs situated in different places sing alternately. Although there are evidences of this practice already in the early Christian liturgies [28], it was not until Renaissance that the *polychoral antiphony* became a common practice; Alessandro Striggio's *"Missa sopra Ecco sì beato giorno"* for five choirs is one of its main exponents [53].

Another example of spatial sound in classical music can be found in the performance depicted in Figure 1.1. It is a 1962 performance from the Leningrad Philarmonic Orchestra; they are interpreting Tchaikovsky's *1812* overture, to commemorate the 150th anniversary of the russian victory over Napoleon. For such a special performance, they located a fanfare in the balcony, next to the ceiling. The fanfare performs a chorus arrangement together with the church bells, in order to simulate the divine help for the russian victory.

Figure 1.1: Leningrad Philarmonic Orchestra performing Tchaikovsky's *1812*. Notice the fanfare located at the balcony

Technological developments in 19th Century provided the means to record sound and reproduce it in a different spatial and temporal location. This disembodiement caused a radical change of paradigm for music. Together with the new possibilities that microphones and speakers offered, they presented also drawbacks for some musical aspects. For instance, multitrack techniques encouraged individual, isolated performance for recordings, reducing to the minimum the interactive aspect, as pointed by Jordá in [44].

The *de facto* standardization of the stereo reproduction systems, which even arrives to our days, also limited the spatial sound dimension. Two-channel systems only allow to virtually position sound sources within the line between speakers, dramatically reducing the spatial dimension.

Despite sparse attempts to perform spatial music along the 20th Century (Langgaard's *Music of the Spheres* or Varese's *Poème Électronique*, to cite some of them), it was not until its last decades when the interest about *immersive sound* or *sound diffusion* received a new impulse. Multi-speaker systems such as the Acousmonium, and spatial sound theories such as Ambisonics were developed and investigated. Nowadays, spatial audio is topic with a growing interest from both technical and artistic disciplines, as the constant research and development can show. Following sections will highlight the main developments on spatial audio.

1.1.1 Online vs. Offline Spatialization

Since last decades, it is possible to observe a growing interest in the multi-channel reproduction systems. Cinema halls incorporate gradually bigger speaker layouts, which allow the listener a more immersive listening experience. Furthermore, the so-called *home cinema* systems, together with *surround sound* layout standards such as *5.1* or *10.2*, expand the sound spatialization possibilities in the consumer's domain.

Obviously, all these technologies need customized tools. All main Digital Audio Workstations, through the use of *plugins* (VST, AudioUnit, LADSPA), provide spatialization techniques to be used in music production. In the cinema domain, products such as *Dolby Atmos* [3] offer a object-based approach for post-production. Specific electroacoustic composition environments also provide their software tools, as can be the case for the WFSCollider software (Figure 1.2), developed to be used in the Game of Life Wave Field Synthesis system [4].

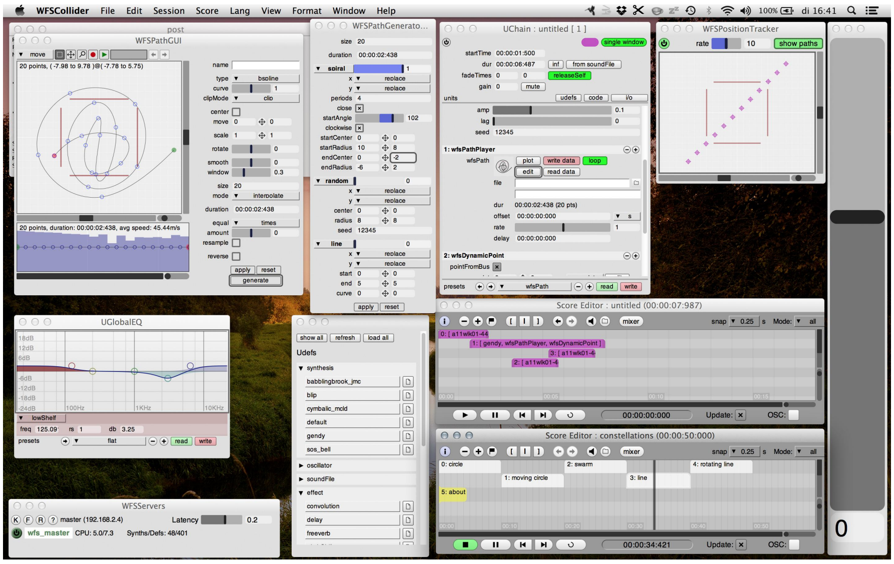


Figure 1.2: WFSCollider software screenshot

These spatialization systems are called *offline* systems, since the spatialization is used as a compositional or effect production tool. In other words, determining the spatial trajectory of a sound does not cause the sound to be spatialized instantaneously, in the same way that writing a score does not cause the music to be heard instantaneously.

However, the *online* or *real-time* sound spatialization with digital technology is still not fully exploited. Cannon and Favila explain clearly this situation in [22]:

> “[...] A large body of compositional audio work has exploited the computer system’s capacity to record and reproduce spatial sound expression. However, live performance work remains musically restricted by the control complexity required to perform spatial motion[...]”

What they are arguing is that performing sound spatialization in real-time needs a different approach from offline spatialization, due to the *control complexity*. But this control complexity is, in fact, the subject under study on the *Human Computer Interaction* field; and, specifically, its musical sub-field, whose center is thes *New Interfaces for Musical Expression* conference.

It is also noticeable that, in a research about the desired features of tools used by electroacoustic composers working with spatial sound, online characteristics were highly rated [56]. More precisely, *Spatial rendering in real-time* and *Controllability via external controller* were among the 3 most frequent answers in a 10-item questionnaire (Figure 1.3). The analysis also shows an empirical relation between these two features, based on a cluster analysis. This is not hard to envision, since these features describe what is needed for online spatialization.

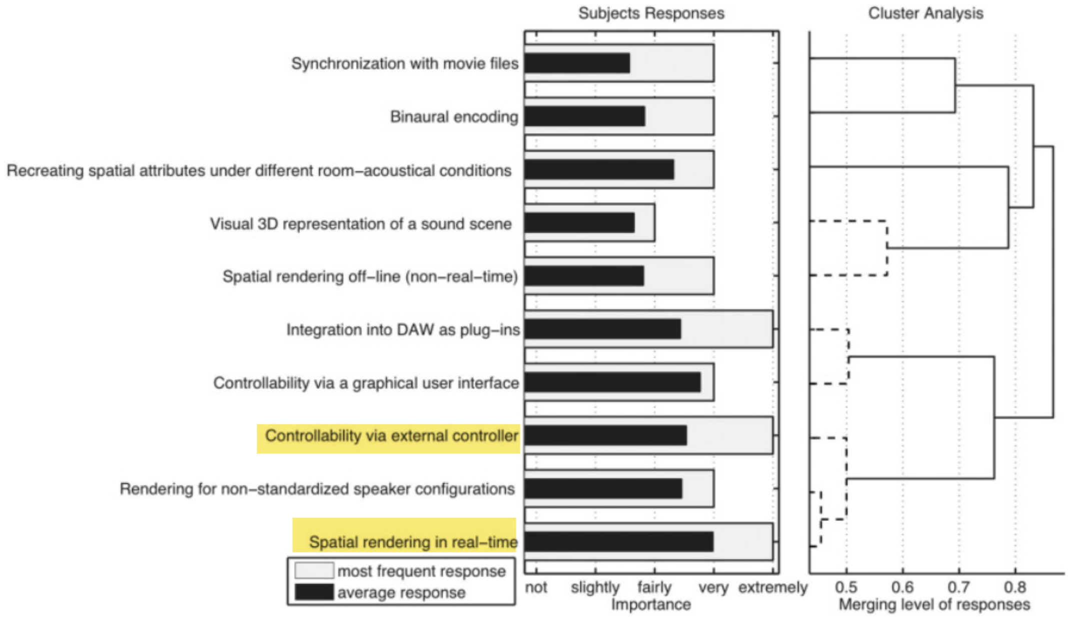


Figure 1.3: Importance of real-time spatialization features for skilled composers, from Peters *et al* [56]. The yellow highlight is made by the author

As a conclusion, we would like to remark the interdisciplinarity of the proposal. Figure 1.4 highlights the different knowledge areas that meet in the proposal: sound spatialization systems and Human Computer Interaction, from the sound/music performance point of view. They merge, at the center, in the concept of *Spatialization Instruments*, that we create and develop in Section 3.1. Regions with dark rim represent the topics discussed in Section 2.

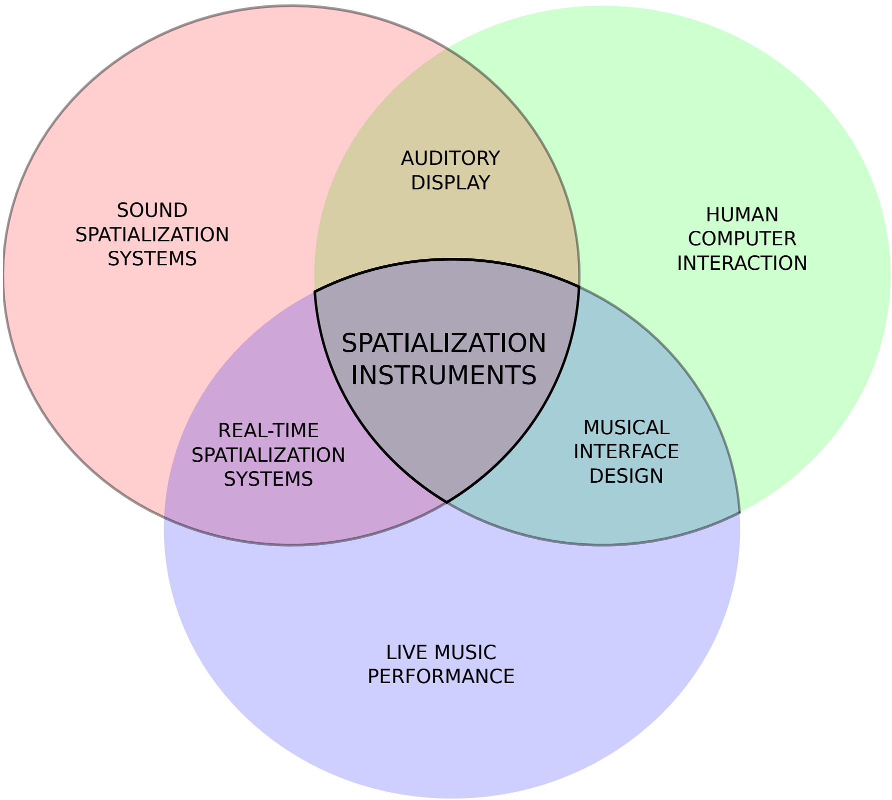


FIGURE 1.4: Background topics in the thesis

1.2 Applications

We can enumerate a set of potential application fields for real-time spatialization systems:

- **Live Performances** With the gradual adoption of multichannel speaker layouts in music venues and concert halls, interactive spatialization might become a new, distinguishing element for live music performances.
- **Interactive installations** Real-time spatialization systems can provide the technical basis for interactive installations, which could include a spatial element in their artistic proposal.
- **Data Sonification** In recent years we witness the rise of data sonification for exploratory analysis. Spatialization systems take advantage of sound perception mechanisms (such as spatial sensitivity or simultaneous streams) to provide a new dimension for the sonified data. Furthermore, the real-time paradigm allows to automatize the spatialization process, allowing the users to focus on the data analysis itself.
- **Virtual Reality** Within virtual and augmented reality, it is desirable to provide a consistent soundfield recreation. Real-time spatialization tools are therefore a potential option for such interactive environments.

1.3 Goals

As we already presented in this Section, most of the work on sound spatialization has focused on the offline domain. But real-time spatialization needs specific approaches, due to its interactive paradigm.

This leads to our main research questions: *Which are the specificities of a real-time spatialization system?* and, consequently, *Do the existing software provide the required tools?*.

The answer to these questions lead us to the following expected outcomes:

- Propose a holistic approach to the sound spatialization considering the existing knowledge on the Human Computer Interaction field and, specifically, the Digital Music Instruments design.
- Provide a comparison of existing real-time spatialization software, as well as proposed interfaces for sound spatialization, and analyse them with the proposed criteria.
- Implement our own proposal of spatialization system, which encourages its usage within the spatialization interfaces development.
- Extend the capabilities of current spatialization techniques, by allowing innovative features such as arbitrary sound source shapes.
- Develop a spatialization interface prototype which can serve as an exemplification and case-study for our implementation.

Chapter 2

State Of The Art Review

In this section, we will provide the necessary background knowledge for the development of the Thesis. The review is done with an informative purpose, but nevertheless with a critical intention.

The State of the Art Review is structured in the following way:

1. Section 2.1 reviews the main sound spatialization techniques, which play a fundamental role in *sound spatialization systems*
2. Section 2.2 presents the *Sound Spatialization System* concept, discussing proposed features and characteristics.
3. Section 2.3 gives an insight into the *Musical Interface Design*, taking into account its characteristics, and specifically the Interface Design applied to sound spatialization.

2.1 Sound Spatialization Techniques

2.1.1 Towards abstract representations of Spatial Sound

Stereo sound is the oldest, most used and most known sound spatialization technique [31]. In stereophony, speaker levels are adjusted in order to (virtually) position the sound source in a point between the speakers. The location perception is accomplished by the several psychoacoustic mechanisms of spatial sound perception [32].

The concept of sound level balancing or *panning* has been further extrapolated to multichannel stereophony, commercially known as *surround sound* formats, such as 5.1, 10.2, etc. In this case, since the speakers surround the listener, sound sources can be placed in the virtual arc among speakers.

This paradigm presents a major drawback: a mix produced for a specific speaker layout can only be successfully reproduced (according to the mixing intended purpose) by the *exact* same speaker setup. Less speakers will unfailingly provide an incomplete sonic representation, while more speakers will not add any extra information.
We say that such a spatialization technique is:

- *Channel-based*, because the mix is done by working (panning) directly in a specific number of channels, and
- *Layout-dependent*, because any change in the speaker setup will provide an incomplete representation.

In opposition to these categories, we can think of systems that overcome their limitations. For instance, it would be desirable to have a *layout-independent* technique, which might be extrapolated to any desired speaker configuration.
In order to get layout-independence, one possible strategy is to think about *sound objects*: objects situated in space, with a set of properties (at least, audio and spatial position). This is called an *object-based* approach, in contrast to the channel-based paradigm. Figure 2.1 depicts the paradigm change.

Notice that this dichotomy (channel-based and layout-dependent *vs.* object-based and layout-independent) between spatialization techniques is not absolute. For instance, as we will see in Section 2.1.2.1, Higher Order Ambisonics techniques provides an intermediate channel-based (but still layout-independent) representation.

When we switch towards an object-based representation, we are adding an extra abstraction layer. This layer is usually called *sound scene* or *audio scene* [30]. A sound scene is a representation of all sound objects, with their associated audio and relevant spatiotemporal parameters (which are usually also called *metadata*). An overview of these parameters is discussed in Section 2.2.

In the offline spatialization domain, some systems have implemented concepts from the object-based paradigm: fully, as in the case of WFSCollider for electroacustic composition [13] or partially, such as Dolby Atmos for cinema postproduction [3]. In the live domain, which is our specific focus, object-based and layout-independent spatialization techniques offer obvious advantages.

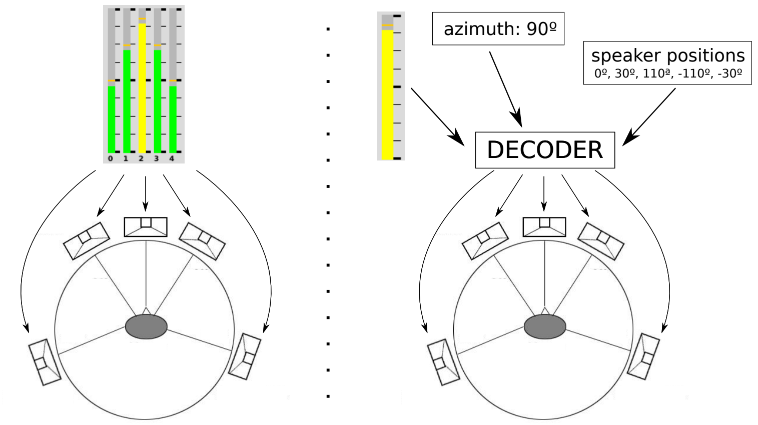


FIGURE 2.1: Example schema showing the paradigm difference between channel-based/layout-dependent (left) and object-based/layout-independent (right) paradigms

2.1.2 Review of Sound Spatialization Techniques

In this section, we will review some of the common spatialization techniques that are used nowadays, and benefit from the sound scene abstraction approaches described in Section 2.1.1.

2.1.2.1 Higher Order Ambisonics

Ambisonics is a complete sonic theory (including both audio recording and reproduction) developed by M. Gerzon and P. Fellgett in the 1970s. It is based on the sound wave decomposition into a truncated series of spherical harmonics. The *order* of the ambisonic representation is given by the number of terms used in the spherical harmonic expansion. Accordingly, there is the usual distinction between *First Order Ambisonics (FOA)* and *Higher Order Ambisonics (HOA)*.

Ambisonics divides the sound spatialization into two stages. The schematic overview is shown in figure 2.2:

1. **Encoding** The sound waves produced (or simulated, in the case of virtual environments) by the sound objects are projected into the spherical harmonics basis. Ambisonics encoding produces a *channel-based*, *spatial encoding* intermediate representation of the sound scene, whose number of channels is given by the expansion order. Figure 2.3 depicts the encoding of a punctual sound source for a 3rd order expansion.

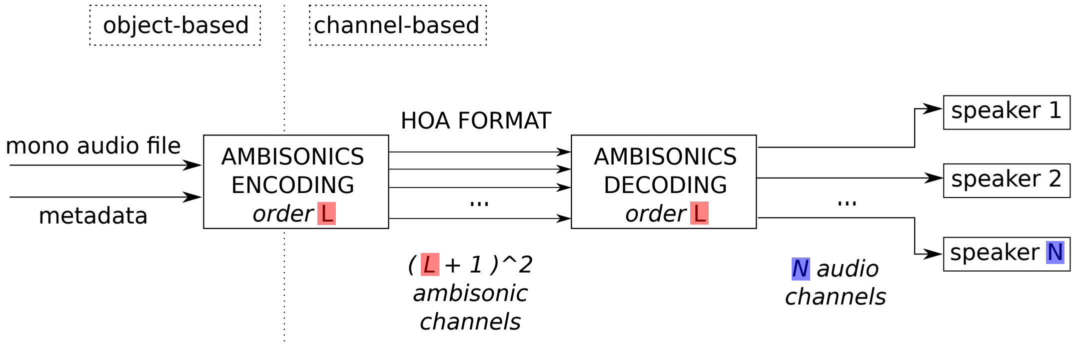


FIGURE 2.2: Ambisonics encoding and decoding stages

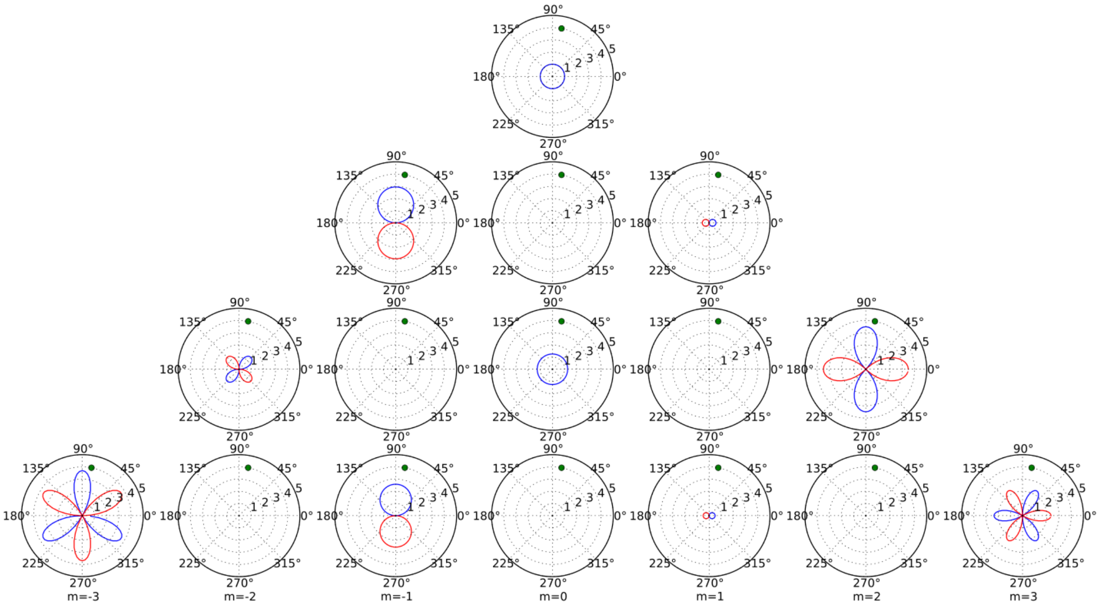


FIGURE 2.3: 3rd order ambisonics encoding of a punctual sound source (green) located around 85° azimuth. Positive values are in blue, and negative in red

2. **Decoding** The ambisonics *decoder* takes the intermediate representation given by the *encoder*, and reconstructs the original sound scene for the current speaker system (it is therefore a *layout-independent* technique). It is a complex task, which might require non-linear search techniques.

The ambisonic order defines the reproduction accuracy. Since exact soundfield reconstruction would need the infinite spherical harmonic series, an expansion truncation must be performed. As an example, Figure 2.4 represents the encoded directivity of a punctual sound source, for different orders. Notice that, with the infinite expansion, the curve will tend to a delta in the origin.

A number of extensions for the theory have been presented. Among them:

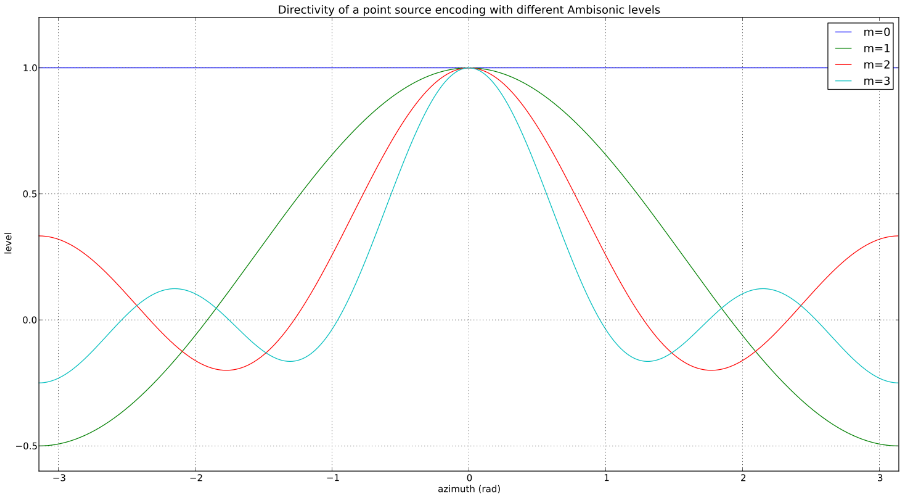


Figure 2.4: Punctual source directivity encoding for different ambisonics orders

1. *Distance Coding* Ambisonics theory does not provide directly a way to encode distance in synthesized sources. This effect can be recreated in different ways: artificial reverbs, wave curvature simulation, direct/reverberant sound ratio, Doppler effect, air attenuation and absorption simulation, etc.

2. *Sound Source Characteristics* It is possible to simulate sound objects with arbitrary shapes by mean of spherical harmonics. In this way, we can model other than punctual sound sources.

One of the major drawbacks of Ambisonics is the need of the listener to be placed near to the *sweet spot*. Furthermore, sweet spot area increases with the Ambisonics order, at the expense of computational power and required speakers and channels.

The reader is referred to the PhD thesis of Daniel, which is the most complete source of information about Higher Order Ambisonics [26].

2.1.2.2 Vector Based Amplitude Panning

Vector Based Amplitude Panning, also know as *VBAP*, is a spatial reproduction technique developed by Pulkki [57]. It consists on a vectorial reformulation and extrapolation of the stereophonic techniques, towards a three-dimensional and layout-independent representation.

In order to place virtual punctual sources, the VBAP algorithm first determines which are the three nearest speakers (two in the case of two-dimensional reproductions). Then, the source gain is computed by linear combination of the speakers location. Figure 2.5

depicts the spatialization process.

Since VBAP is partially based on psychoacoustic cues, one of its major drawbacks is the need of the listener to stand in the so called *sweet spot*, as in the case of Ambisonics. However, the sweet spot is generally much bigger in VBAP than in Ambisonics (at least for First Order).

VBAP does not incorporate distance coding into its theory. Furthermore, in the case of custom source shapes, VBAP is mathematically limited.

Another characteristic of VBAP is the dependency of the source's perceived area with spatial position. Usually the source is reproduced in 2 or 3 different speakers (depending on the dimensions used). But when the source's location coincide with the speaker's location, only this speaker will reproduce sound, reducing the perceived source width. This fact contrasts with the constant perceived source width that Ambisonics provides.

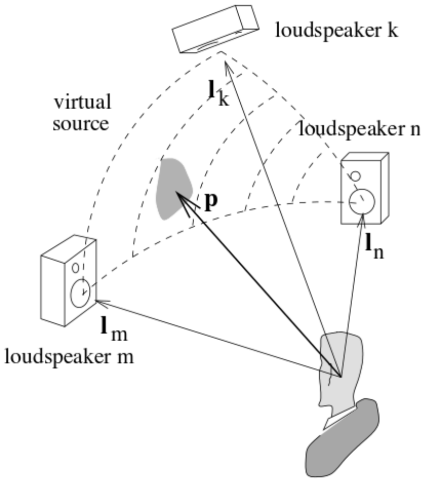


Figure 2.5: Representation of VBAP algorithm (from [57])

2.1.2.3 Wave Field Synthesis

Wave Field Synthesis (WFS) is a spatial reproduction technique developed by Berkhout, de Vries and Vogel [15]. Taking the Huygens principle as a basis, WFS intends the complete reconstruction of the soundfield, considering the speakers as wavefront points. A graphical representation of this principle is depicted in Figure 2.6.

As a consequence, WFS is capable of reconstructing whole soundfields, even when the sound source is inside the speaker area, and also integrates Doppler effect. The major drawback is that the number of speakers needed for an *acceptable* soundfield representation is very high (usually in the order of hundreds).

WFS algorithm requires a considerable amount of computational power. Together with the required number of speakers, three-dimensional WFS systems are still not practical nor widespread, although mathematical formulations are already available [62].

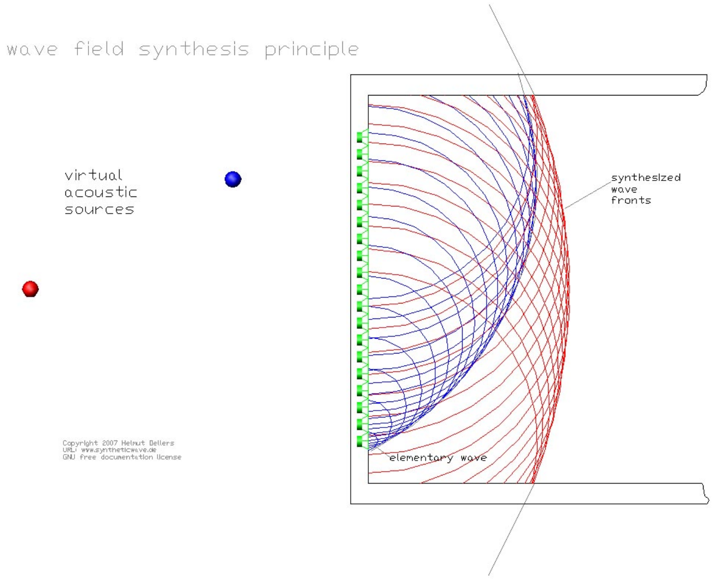


FIGURE 2.6: Huygens principle applied in Wave Field Synthesis (from [12])

2.1.3 Comparative of Sound Spatialization Techniques

Once the different spatialization techniques are reviewed, we present a schematic comparative. Our hypothesis states that HOA presents a series of advantages with respect to the other spatialization systems.

Advantages of HOA respect to VBAP:

1. HOA allows sound sources with arbitrary shapes. VBAP allows only specific size and shape configurations
2. HOA allows introducing reverbs in a more *natural* way
3. Sound source movements in HOA are smoother than in VBAP

Advantages of HOA respect to WFS:

1. HOA requires many less channels, speakers and infrastructure than WFS
2. WFS is *de facto* limited to 2 dimensions

Advantages of HOA respect to both VBAP and WFS:

1. Ambisonics incorporates both object-based (before the encoder) and channel-based (encoded channels) approaches. Intermediate representation has a fixed number of channels (given by the order), which can be an advantage when encoding a big number of sound sources
2. Ambisonics encoding and decoding are separate processes, and can be treated in specific ways (for example, split both processes in different computers, if computational power is critical)
3. It is possible to record sound directly in HOA, since Ambisonics is a complete soundfield theory

Disadvantages of HOA respect to VBAP:

1. HOA requires more computational power
2. Sound source directivity in HOA is worst than in VBAP
3. The required sweet spot area in HOA is smaller that the required in VBAP

2.2 Spatialization Systems

2.2.1 Definition

In Section 2.1, we have seen an overview of sound spatialization techniques. In object-based contexts, we use the term *spatial render* to refer to a system that implements the spatialization. As explained in Section 2.1.1, such a system receives both audio and metadata from the objects as inputs, and outputs the spatialized audio streams to the playback system.

However, as pointed beforehand, the *sound scene* representation of object-based models requires an additional abstraction layer before the render. The possibilities and features of this layer will be developed and discussed in subsequent chapters.

We thus define a *Spatialization System* as the set consisting of a spatial render and the aforementioned intermediate abstraction layer, required for the sound scene handling.

Figure 2.7 illustrates schematically the concepts of *spatialization system* and *spatial render*:

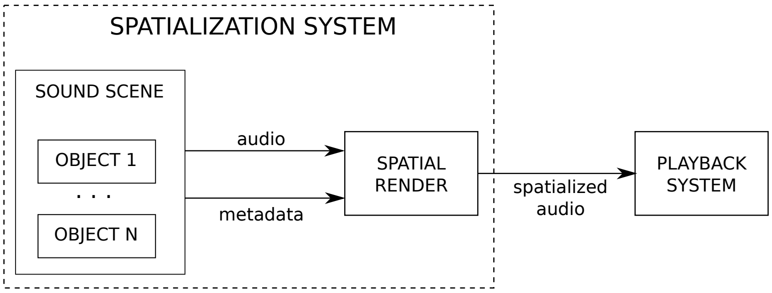


Figure 2.7: Schematic representation of a spatialization system

In his software review [51], McGee considers monolithical implementations as a positive feature for spatialization systems. The arguments given are *robustness* and *ease of compatibility*. While this statement can be accepted, a modular implementation may be beneficial for other reasons. For instance:

- Allows the reuse of existent specific software modules, as in the case of spatial renders or visualizations
- Favors adoption of standard formats

Consequently, in this chapter we will refer to the *spatialization system* as a conceptual whole, regardless of the actual implementation structure.

Finally, we must specify the *real-time* aspect of sound spatialization instruments. Although much work has been done in the offline domain, real-time interaction with spatialization systems needs holistic and specific approaches, which are not fully exploited yet.

However, for the extent of this section, we will consider spatialization systems as *real-time agnostic*, letting the real-time implications for Section 2.3 and Chapter **??**

2.2.2 Spatialization Control Parameters

According to Marshall, *"Even in the simplest implementations, sound spatialization is a multidimensional system (the basic case being 2-dimensional position and volume for each sound source) and with the development of more complex systems incorporating modelling of sound source and room parameters the dimensionality of these systems has increased dramatically"* [49]. Given an object-based spatialization system, his work presents the first systematic organization of possible control parameters, taking into account not only sound source position, but also source characteristics and environmental models.
Marshall proposes the following parameter taxonomy:

Sound Source Position Spatial coordinates (cartesian and/or polar)

Sound Source Characteristics Source size, directivity, presence, brillance

Environmental and Room model parameters Room size and presence, reverberation and early reflections, doppler effect, equalization, air absorption and distance decay

More than the validity of the proposed control parameters, it is important to notice the change of paradigm, from pure positional approaches towards a more complete sound scene. In fact, including source characteristics and room models into the classification tends towards realistic simulations of physical behaviors. This tendency is continued and explained in Section 2.2.4.

2.2.3 Spatialization System Parameters

In addition to the basic aforementioned sound spatialization parameters proposed by Marshall, McGee [51] contemplates a few other system characteristics which can be desirable for a spatialization system. These parameters are not directly related with the spatial nature of the sound, but rather relevant system features.

- Configurable speaker setup
- Arbitrary number of sound sources
- Support for various sound spatialization techniques

In addition to them, McGee also considers whether the system supports object behaviors and/or trajectories. These represent an abstraction to the raw position data, which offer a realistic and intuitive way to treat dynamic objects. This feature is further explained in the following section.

The last feature that we will comment is the compatibility with OSC interfaces. This compatibility is a standardized way access to the control parameters from any kind of Human Interface Device. We will not discuss more here about this possibility, since it will be treated extensively in Section 2.3.

2.2.4 Behavioral models and sound scene abstractions

In his paper, Schacher [59] proposes a very complete perspective to the object-based sound spatialization problem. The main idea behind his proposal is to borrow concepts from the 3D modelling area, and to consider the inherent physical nature of three-dimensional audio scenes. These are the main ideas in his proposal:

- Sound objects have not only properties, but exhibit a physical behavior
- Sound objects can have hierarchical relationships, which ease their control and conception
- The performer can be fully considered as a part of the sonic scene

If we consider the behavioral perspective, a new question is immediately opened: which kinds of dynamic behaviors should be considered? Does exist a set of canonical behaviors, in the form of a spatial language? Many classic electroacoustic composers developed their own methods, including Stockhausen, Chowning or Whishart - these basic forms usually included trajectories, circular and looping motions. Schmele [60, chapter 2.3] provides a complete review about historical developments on sound spatial dynamics. With the lack of an standardized dynamic behavior set, many current spatialization systems have implemented a variety of proposals: trajectories based on splines [31], random walks, bouncing [59], rotations, ballistic and boomerang curves, pendular movements [47] or even Trevor Wishart's spatial motion taxonomy [22, 68].

In addition to the aforementioned ideas, Schacher defines two fundamental ways of interaction with the sonic scene:

Top-Down The user controls a desired specific parameter from a given object/hierarchy level. This is the most common interaction approach, and provides direct control of the parameter.

Bottom-Up The user acts from *inside* the sound scene, and thus the inputs behave according to the implicit physical rules of the scene.

In the first case, the user's role is similar to an *omnipotent* element in the sound scene, which can modify the system in a very precise way. However, it is easy to understand that such precision can be limiting in a scenario with hundreds of parameters. Conversely, the latter case provides a more immersive experience, at the expense of less (or none) precise control of the scene.
Both paradigms offer complementary interaction models, and can be considered depending on the desired design considerations.

2.2.4.1 Behavioral Models and Spatial Sound Synthesis

Spatial Sound Synthesis refers to methods to produce sound which takes spatial dimension as a fundamental and implicit component. In words of Schmele, *"[...] we want to look at creating sounds that do not just exist in space, but* are *the space"* [60].

Many different approaches to Spatial Sound Synthesis have been proposed. We can give as example the *Spatial Swarm Granulator* [66], which considers sonic grains from a dynamic flocking approach, or the *Rapid Panning Modulation Synthesis*, described in [60], which exploits sound artifacts that result from the fast movement of a punctual sound source in circles around the listener. Other spatial synthesis proposals can be found in Schmele's thesis [60].
What is interesting to notice here is the fact that behavioral models can provide the

necessary common basis for any kind of spatial synthesis. In this way, a spatial synthesis method can be seen as a particular behavior for a sound source, and thus different methods are particular cases of dynamic behaviors. A compact description of spatial synthesis methods can be therefore achieved.

2.2.5 Scene Description Standardization and Storage

As pointed by Geier [31], one of the current drawbacks of existing spatialization systems is their lack of interoperability. Object-based sound scenes offer, indeed, independency respect to the speaker configuration, through the rendering system. However, they still depend heavily on the software system running. In fact, due to its custom and often proprietary implementations, it is not easily feasible, for example, to export a piece or even an object behavior from one system to another.
A number of proposals have been done to accomplish this independence.

SpatDIF - Spatial Sound Description Interchange Format

After different panel discussions in various *International Computer Music Conferences*, it was decided that a light, easy to implement and human-readable specification would be desirable. SpatDIF represents the development of these ideas. It can be implemented in various markup file formats (XML, JSON, YAML) and interfaced by OSC. Sound objects information is simplified into timestamps and position. More information about development and implementation can be found in [52] and [10].

ASDF - Audio Scene Description Format

ASDF is the standard developed jointly with the SSR software. In contrast to SpatDIF, it is intended to provide higher level dynamic concepts through symbolic representations, in order to avoid message overloading. It is implemented throug XML, and based on the multimedia standard SMIL. Geier's article [31] extends the information about ASDF format.

SpatOSC

SpatOSC is a different approach to the sound scene standardization. Instead of provide a common output to the spatialization system, it features a set of dynamic *translators* to the existing spatial formats (such as the ones mentioned above, or other custom implementations). In this way, compatibility may be reached without the need of reimplementing spatialization systems to comply with the desired standard. More information can be found in [69].

The widespread use of these formats can bring a number of advantages, to name some of them:

- Possibility of recording spatial scene, for offline playback or analysis
- Possibility of remote performance without worries for system compatibilities

However, it is important to notice that, at the moment of writing, all of these standards are still under development, and none has been widely adopted by existing spatialization software.

Finally, we must mention some industry-oriented sound scene standardization proposals. *MPEG-4 Part 11* specifies spatio-temporal positioning of audio-visual objects, and is implemented by *BIFS* binary format files [61]. *MPEG Surround* is a complete transmission and storage audio format: it implements audio compression with support for spatial metadata [19]. Its capabilities are further extended by the *Spatial Audio Object Coding (SAOC)* format [20].

2.2.6 Review of Real-Time Sound Spatialization Systems

In this section, we perform a comparative review among existing spatialization systems that support real-time interaction, whose results are presented in Table 2.1.

In the classification, we distinguish between the two main types of spatialization software: standalone applications, and applications based on existing Sound Processing Environments (such as SuperCollider or PureData).
It is important to notice that some of the spatialization systems are custom implementations for a given reproduction system; this is the case for Zirconium (Klangdom, Zentrum für Kunst und Medientechnologie Karlsruhe) and BEASTMulch (BEAST, Universtity of Birmingham).

The evaluation is performed by using experimental *historical* methods; more precisely, we use a combination of *Literature Search* and *Static Analysis* methods, in the sense that we revise the existing spatialization systems.

Comparative parameters are validated by its acceptance within the scientific community on the area. Specifically, they are taken from three sources:

Table 2.1: Comparative of Real-Time Spatialization System Software

	Standalone systems				SPE based systems		
	Zirkonium [14]	Spat [9]	SSR [30]	SES [51]	BEAST Mulch [2]	Spatium [11]	OM Prisma [7]
3D position	✓	✓	×	✓	✓	✓	✓
Source size	✓	?	×	✓	?	?	?
Source directivity	?	?	×	×	?	?	?
Room parameters	?	✓	×	×	?	✓	✓
Distance cue	×	✓	✓	✓	?	✓	✓
Configurable speaker setup	✓	✓	✓	✓	✓	✓	✓
Arbitrary nº 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 interaction	✓	×	?	✓	?	✓	✓
Description format	?	SpatDIF	ASDF	SpatDIF	?	(OSC-based)	SpatDIF
Platform	MacOS	MacOS	Linux MacOS	Linux MacOS	MacOS	MacOS	MacOS
License	BSD	propietary	GPL	propietary	GPL	CC	LGPL

The first one is the work of Marshall [49]; we must notice that, in their comparative table, they mix together both rendering systems and spatialization systems. The second source is the paper from McGee and Wright [51], where they review different sound spatialization systems, and provide some insights into their design. Finally, the third set of parameters are added by the author following the criteria presented in Section 2.2, taken mainly from the work of Schacher [59] and depicted in Figure 3.1.

From Table 2.1, we can extract some conclusions, in the form of qualitiative valorations:

- First of all, none of the systems integrates all the desired functionalities; each of them is centered around a specific subset of features.

- Most of the systems allow different spatial rendering techniques. Plain VBAP and its extensions (such as DBAP) is the most widespread. Higher Order Ambisonics is also commonly implemented, but usually with some restrictions. For instance, *SSR* allows only for two-dimensional representations, and *Spatium* limites the encoding to 3rd order horizontal + 1st order vertical.

- Regarding behavior and trajectories support, there is a lack of implementation, considering that these are the basis of spatialization dynamics. Even the systems that implement them only consider a small subset of predefined trajectories.

- There is an absolute lack of support for hierarchic management, as proposed by Schacher [59]. We consider that such a feature would be desirable for a live spatialization system.

- Regarding the spatial description format, it is clear that the desired standardization is still to come. Some systems start to implement SpatDIF, and only SSR implements ASDF (which was created by the same researchers). Independently from the chosen description format, it would be desirable a complete interoperability among systems.

- Availability of software is not always provided. For instance, *BeastMULCH* system is only compatible with OSX versions up to 10.4, which was released in 2005, and superceded in 2007 by version 10.5 . Another example is given by the *Sound Element Spatializer*. Despite its extense documentation (for example [51]), it is not possible to find the software on internet.

- The table also shows a huge dependency on the MacOS operative system. Even with SPE based systems, which are theoretically cross-platform, the use of specific libraries shortens the inter-platform usage. SSR, the only available system working under Linux, lacks still many features - 3D support is, without doubt, the biggest shortage.

- Finally, we must remark the importance of Free Software implementations. In a scientific research context, availability, experiment replicability and improvement capacity are fundamental considerations; privative licenses do not comply nor encourage these statements. Most of the software has free sofware licenses; however, they still show dependency towards privative operating systems.

.

2.3 Digital Musical Instruments

2.3.1 Concepts

We will begin this section by defining the term *musical instrument* as a device used to play and to produce any music, transforming in real-time (i.e. by being played) the actions of one or more performers into sound events [42]. Although it is a quite informal definition, it will be enough to conceptualize our ideas.

Along with technological developments in 20th Century, a new kind of instruments appeared, which do not rely on the acoustic laws, but are based on electronic devices. Examples of this are instruments such as the *Theremin* or the *Ondes Martenot*. These developments set the basis of the field know as *Digital Musical Instrument* design [48, chapter 2.1], which grew up after the personal computer boom and the digital synthesis standardization operated by the MIDI protocol [43, chapter 2.3]. The *New Interfaces for Musical Expression (NIME)* conference is the main reference for new instrument design or *Digital Lutherie* [43].

Digital Musical Instruments present mainly two breaking differences respect to acoustic instruments [44]:

- Since instruments are no longer depending on their building acoustic properties, they are theoretically capable of producing *any* kind of sound
- The separation between the *gestural control interface* (also referred as *input device*) and the *sound synthesis engine* is now explicit; think for example in a MIDI keyboard, unable to produce sound by itself, which must be linked to a sound synthesis module to produce any sound.

As pointed by Jordà [44], even if the control and the synthesis are decoupled, it is very difficult to design one system without knowledge about the other; thus, a parallel development is always preferred.

Figure 2.8 depicts a possible schema for Digital Musical Instrument interaction, as proposed by [64], although the same schema could be used also for *traditional* acoustic instruments. We decided for this design due to its simplicity and clearness, but alternative schemas could be also used. A review of similar proposed schemas can be found in [48].

Note that the terms *interaction* and *feedback* will be discussed and defined in Section

2.3.2. Furthermore, in this document we will refer to *gestures* as any action performed by the human body that is capable of being sensed by the interface device.

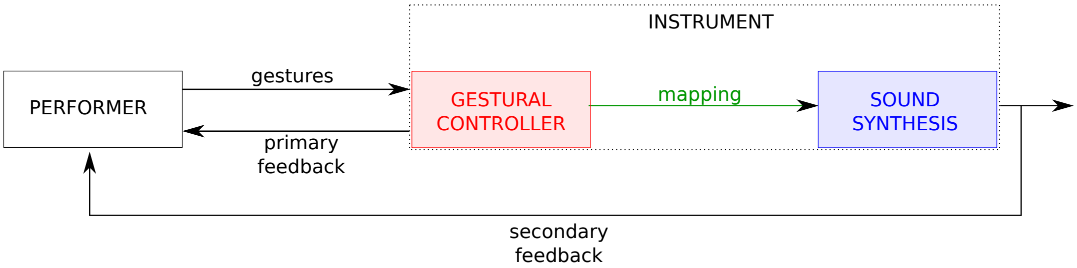


Figure 2.8: Digital Musical Instrument interaction schema as proposed by [64]

2.3.1.1 Control Interfaces

Much attention has been paid in last years into the development and design of novel control interfaces [48]. Despite the huge variety in shapes and functionalities, we can classify them conceptually using Wanderley's taxonomy [64]:

Instrument-like Controllers
Devices that tend to imitate an existing instrument

Augmented/Extended Controllers
Acoustic instruments with extra sensors that expand their functionalities

Alternate Controllers
When the device is not based on any existing instrument

It is relevant to consider the human body as included in the *alternate controllers* group. A number of position/motion trackers and depth-cameras have been created and used extensively in last years [63]. In this way, the own gestures and movements are directly mapped into the system, in a non-invasive way.

Each one of these categories has its own advantages and disadvantages, in terms of simplicity, previous knowledge, cognitive load, idiosyncrasy, etc. However, as noticed by Jordà, *"Any input device can become a good or a bad choice depending on the context, the parameter to control, or the performer who will be using it"* [44].

Multiplexation

When the amount of controllable parameters to control tends to be high, it is possible to split the interface's control scope. The concept of *multiplexing*, first introduced in HCI by Buxton [21], distinguishes between two different approaches:

- **Spatial Multiplexing** There is a dedicated interface/transducer for each specific parameter
- **Temporal Multiplexing** There are more parameters than interfaces/transducers. Therefore, their functionality is variable over time, by means of specific *state change* elements.

2.3.1.2 Mapping

In Figure 2.8 we can also observe *mapping* as the connexion between input device and sound synthesis. Mapping the the system which relates the outputs of the controller (*control parameters*) to the sound synthesis inputs (*synthesis variables*).
Although at the beginning of DMI design practice mapping was not fully considered, it was clearly shown that a good mapping strategy design was fundamental for the success of a Digital Musical Instrument [35, 37].
The first attempt into mapping strategies organization, proposed by Rovan and colleagues [58], distinguishes three types of mappings:

- **One-to-One** Each one of the control parameters affects one specific synthesis variable
- **Divergent or One-to-Many** One specific control parameter affects simultaneously many synthesis parameters
- **Convergent or Many-to-One** Many control parameters are coupled to the same synthesis variable

Of course, a big amount of proposals have been done since then, including intermediate mapping layers to abstract control parameters [37], dimensionality reduction through interpolation [33] or not explicit, generative mapping mechanisms [36].

It is interesting to notice the general tendency to map gestures with parameters in a idiosyncratic way (for example, more input energy produces a louder sound), as it is usual with acoustic instruments. Nevertheless, we can tweak these classical metaphors at our own will, in an attempt to create innovative interaction experiences.
Winkler [67] illustrates these possibilities in the following way:

> “[...] tampering with the apparent laws of physics is a luxury made possible in virtual environments. By being aware of these laws, it is possible to alter them for provocative and intriguing artistic effects, creating models of response unique to the computer. More furious and strenuous activity, for example, could result in quieter sounds and silence. [...] Such ”unnatural” correlations makes motion all the more meaningful”.

2.3.2 Interactivity

In words of Jordà, *“[In digital instruments] the performer no longer needs to control directly all these aspects of the production of sound, being able instead to direct and supervise the computer processes that control these details”*. Therefore, *“performing music with ‘intelligent devices’ tends towards an interactive dialogue between instrument and instrumentalist”* [44]. He continues defining *interaction*: *“‘Interaction’ involves the existence of a mutual or reciprocal action or influence between two or more systems”*.

However, the sole existence of a digital instrument does not automatically provide its interactive quality. According to Winkler [67], interactivity is a continuous property, and depends on (1) the amount of freedom given to the performer, and (2) the computer ability to respond in a appropriate way.

As a conclusion, digital instruments should be able to present interactivity in the same way that classical instruments do. Jordà exaggerates this concept in his sentence: *“A good instrument should also be able to produce ’terribly’ bad music, either at the player’s will or at the player’s misuse”* [44].

If we look again Figure 2.8, we can observe that the interaction between the performer and the instrument is provided, apart from the gestures, by the so called *feedback loop* [25]. Wanderley [64] distinguishes between two feedback types:

- **Primary feedback** Visual, tactile, haptic feedback provided by the input controller
- **Secondary feedback** Auditive feedback produced by the sound generator

Furthermore, feedback can be also classified into *passive*, as a result of the physical characteristics of the instrument, and *active*, when the feedback is a somehow predefined system response to the performer. It is important to notice that, although all digital instruments show some kind of passive feedback due to their physical nature, active feedback must be properly addressed for a successful interactive design [48].

2.3.3 DMI for Sound Spatialization: Considerations

Body, movement and space

One of the most immediate ideas that come from considerating spatialization systems as instruments is the relationship between our physical three-dimensional reality, experienced through our body, and the physical reality of the spatialized sound. In words of Schmele, *"Finally, this idea, translating the physical movement of a performer into sonic movement through virtual space is an interesting, and, considering our bodily experience of space, probably the most true approach to trajectorial gestures"* [60]. The same idea can be found in words of Wanderley and Orio: *"[...] devices whose control structures match the perceptual structure of the task will allow better user performances."* ([65], referring to [38]).

The possibility of employing our body gestures, with the help of a motion tracker, as the input device is a very interesting idea, that connects with many artistic and aesthetical domains. In fact, such new proposals for interaction modes should be investigated, since sound spatialization does not present classical acoustic instrument precedents [22]; therefore, extended or alternate controllers should be favourited.

Performer roles and cognitive load

Marshall [49] differenciates between three main roles that a performer could adopt with a spatialization instrument. These roles are roughly based on their spatialization control parameters classification (see Section 2.2.2):

- **Spatial Performer** Locates sound sources in the space using gestures
- **Instrumental Performer** Performs changes in sound characteristics by performance gestures on an extended instrument
- **Spatial Conductor** Changes room and environmental parameters with gestures

We may criticise the former classification for various reasons. First, assumes an existing type of controller for each role (alternate controllers for first and third roles, and augmented interfaces for the second). Second, it is artificially limited to the control parameters of each one of its classes. We believe that a more complete and expressive approach should take into account control parameters indistinctly of their belonging to a classification abstract.

A probably more interesting approach to the performer roles is given in the same article. Marshall discusses about *cognitive load* implications with a spatialization instrument, and distinguishes again three possibilities:

- The performer is only in charge of spatialization system, letting sound generation for other performers or the computer
- The performer controls both spatialization and sound generation, through its own instrument
- The performer is in charge only of sound generation through its instrument, and spatialization is controlled in an *unconscious* way through his/her performing gestures

Again, there is the implicit assumption that a given interface type should be used. However, the interest of this proposal is the distinction between *performing only spatialization* and *performing spatialization and sound generation.* The latter case, where the spatialization is not directly controlled, can be seen as a special case of spatialization where neither the interface nor the mapping are known, or follow complex patterns.

Feedback and disembodiement

The last specific consideration for spatialization instruments is the particular characteristics of secondary (auditive) feedback. As pointed by Cannon [22], sound coming from a different location than the performer's one causes a *disembodiement* sensation that must be handled with caution, specially in the case of extended instrument interfaces.

2.3.4 Design Considerations

In addition to all aforementioned aspects of digital instruments, we can take into account some further considerations about instrument design:

- *Is the performer only able to control one aspect at once?*
  As we saw in Section 2.3.2, the performer is not more required to control in a continuous way every aspect of the sound synthesis. Therefore, the *multithread* and *shared control* paradigms are now available [25, 44]. A digital instrument should provide mechanisms to take and leave control of the processes at will.

- *Is the instrument intended for a casual user or for a potential expert?*

  First of all, it is important to notice that even the expected outcomes from interaction with a digital instrument can vary depending on the user approach: casual, novice users probably expect only a positive experience (*play with music/musical toy*), while experts may expect some kind of expressivity (*play music/musical instrument*) [41, 65]. In interactive installation contexts, social interaction is probably the most expected outcome [16].

  In the same way, instruments designed for sporadic users (which are not expected to spend hundreds of hours practising) should be (on purpose) limited enough to facilitate participation [16]; this is indeed not the case for potentially virtuosos. However, as pointed by Jordà, an instrument design that covers both apparently exclusive paradigms may be reached [40]. Figure 2.9 depicts briefly the described ideas.

  Concepts as *learning curve* [45] or instrument *efficiency* [41], which will be not discussed here, should be taken into account.

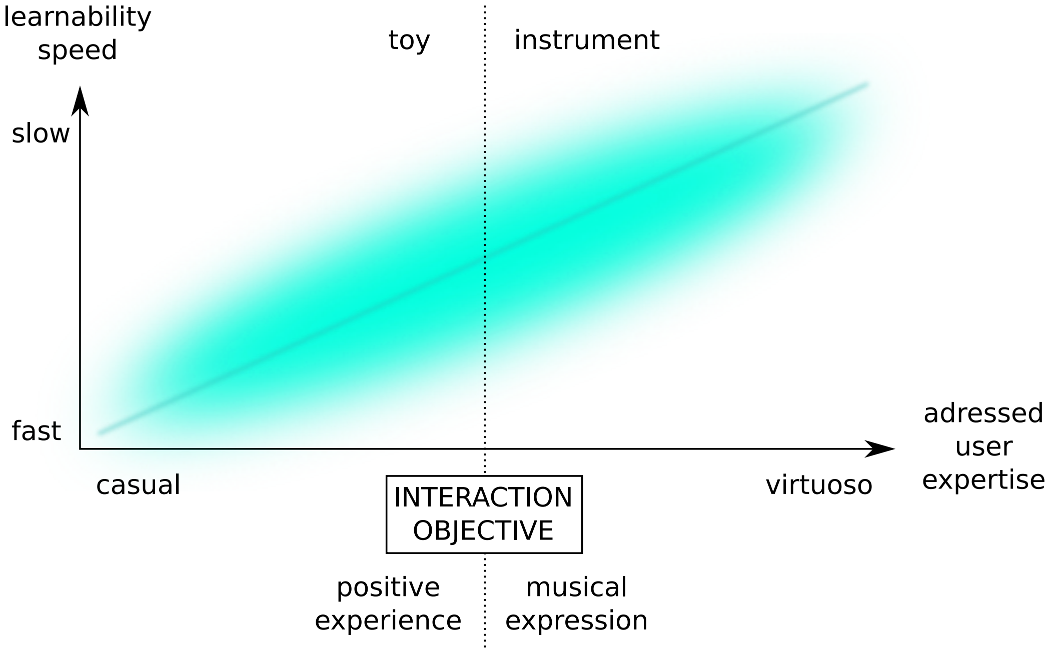


Figure 2.9: Relationship between intended user expertise and instrument learnability, based on ideas from [41] and [16]

- *How many performers may participate?*

  Traditional instruments are mainly limited to one unique performer. However, despite the innovative possibilities of digital instruments [40], the dichotomy *solo virtuoso performer / multiple casual performers* continues a general trend (as we will show in Section 2.3.5 for the specific case of spatialization instruments).

2.3.5 Review of Sound Spatialization Instruments

In this section we will present a review between different interactive spatialization instruments appeared in last years. Although the concept of *Spatialization Instrument* will be introduced and developed in the forthcoming Section 3.1, we can preliminary consider them as Digital Musical Instruments intended for sound spatialization.

Again, the evaluation methodology is based on *historical* methods. We revised the existing descriptions from publications in NIME and ICMC conferences, and contrast them against widespread parameters from interaction design area (Section 2.3.5.1) and spatialization systems (Section 2.3.5.2).

2.3.5.1 Classification according to DMI design parameters

Table 2.2 shows the comparative between those systems, classified by the criteria developed in Section 2.3 referent to Digital Musical Instruments.

TABLE 2.2: Comparison of spatialization instruments according to DMI dessign

	Multithread Shared control	Intended User	Number of performers	Rol of performer	Interface	Active feedback	Evaluation
Boko11 [17]	✓	trained	1	spatialization & synthesis	gesture tracker	×	×
Bred08 [18]	×	casual	1 / many	spatialization	tabletop	visual	×
Cann10 [22]	×	trained	1	spatialization & synthesis	extended	visual	×
Cara11 [23]	×	casual	1	spatialization & synthesis	gesture tracker	visual	×
Carl11 [24]	×	both	1	spatialization	slider	visual & haptic	×
Fohl13 [27]	✓	?	1	spatialization	gloves	×	×
John13 [39]	×	casual	1 / many	spatialization	tabletop	visual	✓
Mare07 [46]	×	trained	1	spatialization & synthesis	extended	×	×
Mars06 [47]	×	trained	1	spatialization	gesture tracker	×	×
Mars09 [49]	×	trained	1	spatialization	gesture tracker	visual	×
Nixd06 [54]	×	trained	1	spatialization	?	visual	✓
Park13 [55]	×	trained / casual	1 / many	spatialization	smartphone	visual	✓

By analizing Table 2.2 we can appreciate the following tendencies:

- Contrary to our expectations, the most of the instruments do not provide multithread and shared control mechanisms; they are rather following the classical absolute control paradigm. Systems by Bokowiec [17] and Fohl [27] does implement shared control by means of dedicated *active/inactive control* buttons. In fact, Bokowiec presents this feature as *unusual*.
  We must remark the comments given in [17] about the *King Midas problem*: the problem of letting know the system when are we performing spatial gestures, and when not. The way they implemented the solution, by means of selection/deselection mechanisms, appears to be too slow for a real-time interaction.
- Only 3 out of the 12 analyzed system reviews presented some kind of evaluation methodologies. The rest only assessed personal valorations over the system. We must take into account the fact that most of the systems were still under development, and also the **still existing lack** of a standard evaluation methodology (the reader is encouraged to read [65] for more information about DMI evaluation). Despite Greenberg and Buxton illustrate the problems that a premature evaluation might bring [34], simpler approaches such as participatory critiques or polls might provide positive design feedbacks.
- We can observe a continuity in the tendency *one intended expert performer / multiple casual performers*, which follows the dichotomy *artist performance / interactive installation*. As we already mentioned, there is no reason to follow this tendence (but also not to follow it!); we must only be aware of the possibilities that digital instruments offer.
  It is fundamental to mention the work of Park and colleagues [55] according to their innovative multi-user proposal. They designed two types of interface for a live concert scenario: one for a spatial performer on stage, and another different to the audience, accessed through their smartphones. Even if the interaction possibilities are very limited, the proposal fills an intermediate gap between individual and multi-user systems, specifying different interaction modes for each role.

2.3.5.2 Classification according to Spatialization System parameters

We can also compare the selected spatialization instruments according to their spatialization system characteristics. Results are shown in Table 2.3. Comparative criteria are taken from those discussed in Section 2.2.

Table 2.3: Comparison of spatialization instruments according to spatialization system parameters

	Periphonic	Control parameters	Trajectories	Hierarchies	Interaction mode	Spatialization technique	Spatialization system
Boko11 [17]	×	position & trajectories	✓	×	top-down	?	×
Bred08 [18]	×	position	✓	✓(groups)	top-down	(SSR)	SSR
Cann10 [22]	×	position & trajectories	✓	×	top-down	FOA	×
Cara11 [23]	×	physical model	✓	×	top-down	WFS / VBAP	×
Carl11 [24]	×	physical model	×	×	bottom-up	VBAP	×
Fohl13 [27]	×	position & trajectories	✓	×	top-down	WFS	×
John13 [39]	×	position	×	×	top-down	VBAP	×
Mare07 [46]	×	position	×	×	top-down	VBAP	×
Mars06 [47]	✓	position & environmental	✓	×	top-down	ViMiC	×
Mars09 [49]	✓	position & environmental	×	×	top-down	?	×
Nixd06 [54]	×	position	×	×	top-down	?	×
Park13 [55]	×	position / casual	×	×	top-down	custom	×

Again, we can highlight some qualitative comments based on the analysis of Table 2.3:

- First of all, and probably the most interesting idea, is that most of the systems only allow two-dimensional spatialization. As pointed by Cannon, *"There is a need to develop more sensor devices capable of performing 3D gestures"* [22]; but this fundamental lack might be also due to technical or logistic limitations (as in the case of systems using WFS). In any case, we consider this lack as a big drawback of new spatialization instruments.

- We observe a general tendency to continue with simple source location paradigm. Even if the majority of systems allow trajectories and dynamics control, more abstract control parameters would probably benefit creativity. We can take as an example the immersive system from Caramiaux [23], where the user controls macro-structural parameters such as *elasticity* or *viscosity* in a simulated particle system with its own physical rules. These parameters are mapped also to input "abstract" gestures such as energy or periodicity.

- Regarding the proposals made by Schacher [59], ony the system by Bredies and colleagues [18] allows for a partial implementation of hierarchies, in the form of source grouping.
  On the other hand, the *Sound Flinger* by Carlson and colleagues [24], despite its simplicity, is the only one which allows the *bottom-up* interaction mode. In this case, the sound scape is composed of two sound sources, with mass and attraction properties. The user can place his virtual representation through sliders and, when located in the same place that an object, move and displace it.
  We believe that, again, consideration of these ideas may benefit innovation and expressiveness.

- We find interesting that only one instrument (the one from Bredies [18]) utilizes an available spatialization system software, from the ones reviewed in Section 2.2.6. Moreover, this was not casual, since they belong to the same research group that developed the spatialization system (for more evidences, the article name is *The Multi-Touch SoundScape Renderer*).
  The rest of the instruments used custom implementations, mostly based on SPE like Pure Data or Max/MSP; and often with a notable lack of capabilities, compared to the existing dedicated software.

- Only the work by Cannon has used Ambisonics spatialization techniques. We agree with the comments expressed in his article: *"[...]yet little work has been undertaken to apply Ambisonics to augmented instrument ensembles or live improvisation"* [22]. We must notice that Cannon's implementation only used the most basic

approach (First Order Ambisonics in a two-dimensional space).
According to what we exposed in Section [**?**], we consider that **the use of High Order Ambisonics can push spatialization instruments (and systems) towards innovative and rich spatial sound experiences**.

- Finally, it is important to remark that, for both tables 2.2 and 2.3, some of the parameters were difficult to guess out of the information provided by their respective article descriptions. Of course, **it was not possible to replicate the systems, from the interface to the software** (with the implications that this fact has into controlled evaluation methodologies and experiment reproductability). However, we believe that pushing toward specific standards or agreements, as in the case of spatial descriptor formats (see 2.2.5), would favourably impact improvements and new ideas on the field. Sharing a common spatialization system software could be one of the starting points.

Chapter 3

Methodology and Implementation

As we already highlighted in Section 1.3, we are interested in analysing and improving the relationship between the specificities of real-time sound spatialization and the existing approaches performed by both spatialization software and interfaces for spatialization. Therefore, we planned two different research steps:

1. **Spatialization Instruments** (Section 3.1): We develop the ideal characteristics that a real-time sound spatialization system should feature, around the concept of *Spatialization Instrument*.
2. **Software implementation** (Section 3.2): Once we have the desired features , we find that existing spatialization systems are not designed to fully comply with these features. Furthermore, most of them are not available or deprecated. Consequently, we develop our own spatialization tool, and provide a sample interface prototype to interact with it.

3.1 Spatialization Instruments

We present in this section the concept of *Spatialization Instrument*, which was already used in the comparative review of Section 2.3.5 . We define it as follows:

> A *Spatialization Instrument* is a Digital Musical Instrument, which has the capability of manipulating the spatial dimension of the produced sound, independently of its capability of producing or manipulating the other sound dimensions.

For instance, we can use the Digital Musical Instrument as defined by Wanderley [64], which is depicted in Figure 2.8. For a Spatialization Instrument, the *Sound Synthesis* functionality will be performed by a *Spatialization System* as defined in previous sections.

This definition represents a holistic approach to the real-time spatialization problem, which takes into account the knowledge basis from the NIME area presented in section 2.3. Furthermore, the concept will be useful for the upcoming analysis performed in Sections 2.2.6 and 2.3.5.

The general proposed schema for a *Sound Spatialization Instrument* is presented in Figure 3.1. Specifically, we bring together the structural ideas showed in Section 2.2 with the interaction and interface design explained in Section 2.3. We also take ideas from schemas proposed by Wanderley [64] (Figure 2.8) and Schacher [59].

Some comments on Figure 3.1:

- First of all, notice that the colour schema agrees with the one presented in Figure 2.8. Functional building blocks are preserved, even when their location and representation are slightly different.
- We make an explicit distinction between *primary feedback* (both active and passive) and *secondary* or *auditive feedback*, in order to remark the different intermodal possibilities. In this sense, the *active feedback* goes, from a dedicated *Active feedback generator*, to both the user (as in the case of visual cues) and to the input device (as in the case of potential visual, haptic or tactile cues).
- Regarding the *Sound scene representation* system, we define *objects*, *hierarchical relationships*, *behaviors and trajectories* and *physical model* as their functional

blocks. However, we must notice that any representation based on the parameters defined in Section 2.2 would equivalent.

- The output of the *Sound scene representation* might be formatted in a standardized way, as the proposals mentioned in Section 2.2.5. As already commented, this standardization can ensure inter-render compatibility, as well as facilitate storage and transmission of the sound scene.

- The *Sound processor* block is responsible for the audio which will be spatialized. As noticed in the definition, the user might or might not have control over the sound content (apart from the spatial dimension).
  Sound can be both external (music performed by the user), internally synthesized, or a combination of both. This audio is transmitted to the *Spatial render* stage, where it will be spatialized according to the metadata provided by the *Sound scene* block.

- Due to the importance of the *mapping* stage, as highlighted in Section 2.3.1.2, we considered interesting to conceptualize it as a dedicated block within the sound spatialization system. We can observe that mapping stage links not only with the *Sound Scene*, but also with *Sound Processor* stage; as it will be explained in Section 2.3.3, the user might control either only spatialization parameters, or both spatialization and sound synthesis parameters.

- We must remark that the proposed schema is a conceptual organization of features and capabilities. For instance, the actual spatialization system software does not need to be implemented monolithically; a modular approach can be positively considered, as discussed in Section 2.2.1,

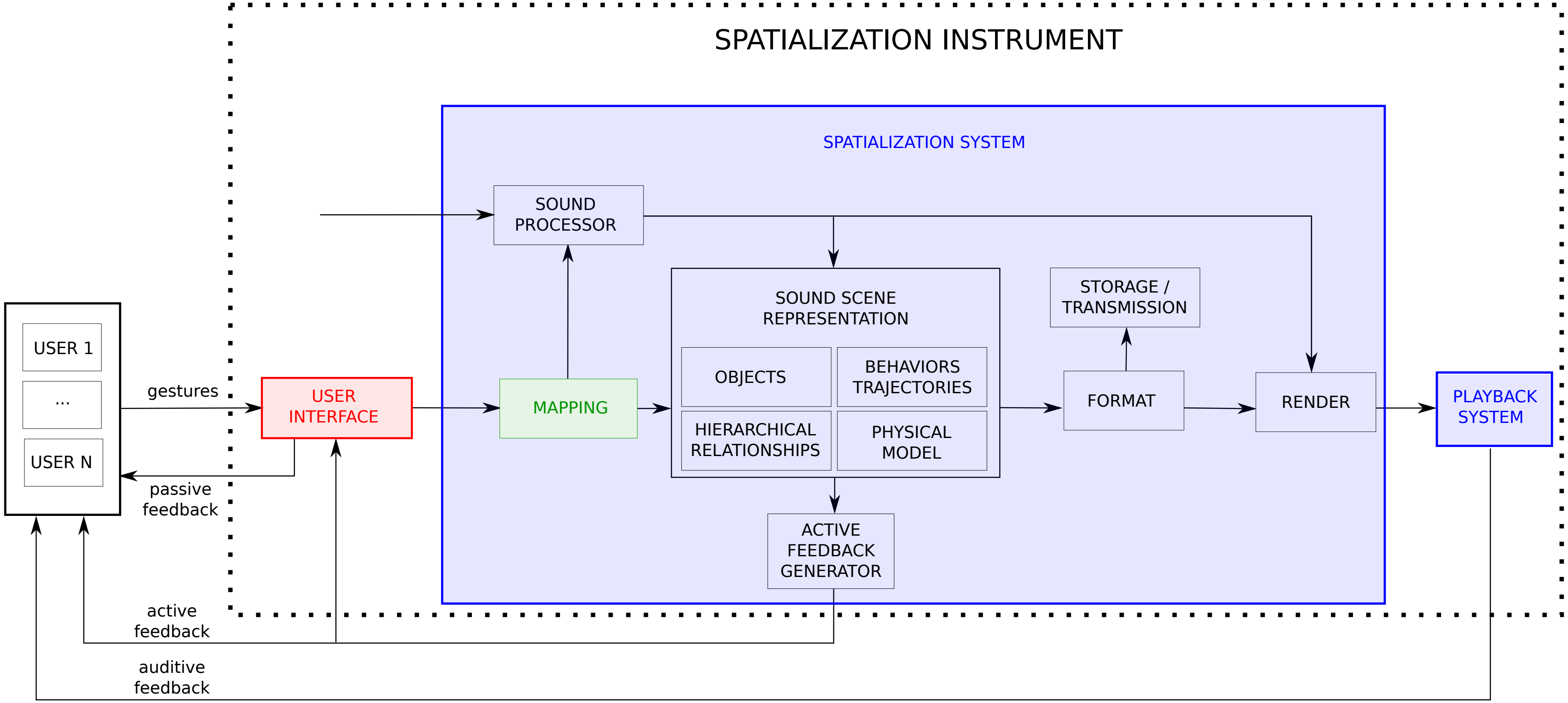


FIGURE 3.1: Proposed schema for a Sound Spatialization Instrument and its interaction

3.2 Software Implementation

In this section, we comment our spatialization system implementation, called *SCLiss: SuperCollider Live Spatialization System*. We must remark again the implementation goal: we will provide a spatialization system which will be mainly used as a tool for spatialization instruments design and performance. Therefore, the software will not be used by itself in performance time; it will provide the technical basis to allow designers to focus into the interactive design and creative aspects.

3.2.1 Design Specifications

From the conclusions provided in the previous section, we can extract a set of design specifications for our custom implementation. We will comment on them individually:

- **Device independence** We must ensure that our system can be used with any kind of control interface. The way to provide that is to comply with the standard interface protocols: HID, MIDI and, specially, OSC.
- **Mapping** We showed in Section 2.3 the importance of mapping in the interface design. We must provide access to flexible and arbitrarily complex mappings, not limiting the potential connections between interfaces and control parameters.
- **Control parameters** Related to the previous item, we must provide a big range of control parameters, including not only the basic ones reviewed in Section 2.2.2, but also any kind of parameters related with physical modelling, trajectories, groupings, sound scene, etc.
- **Feedback modalities** Capability of providing different feedback types is crucial, as showed in Section 2.3.2. On the one hand, we can provide specific visual feedback integrated in the software. On the other hand, interface protocols compatibility allows also active feedback to be generated in the interface directly.
- **Sound Scene** We must include the reviewed desired features into our sound scene simulator: trajectories, hierarchies, physical simulations and different interaction modes.
- **Spatial Render** We consider important to provide different spatialization techniques, from which the user can choose according to the specific context. However, we also consider important to encourage the use of High Order Ambisonics, due to its potential and the observed lack of full implementations and usage by existing spatialization instruments. The capability of simulating arbitrary sound

shapes can also help in this goal. Finally, we believe that the compatibility with SpatDIF format, which is being actively developed, is a determinant choice for inter-operability and storage purposes.

- **Modularity** With the SpatDIF compatibility, we can effectively split the Sound Scene and the Spatial Render blocks. In this way, they can run in different machines if the CPU load is critical, or even use one independently from the other, if a different software is preferred.
- **Free Software** As a software development in the research domain, we must open the possibility for the community to access the software in any condition, see the code, analyse it, improve it, reuse it and adapt it for their necessities. Therefore, it is mandatory to use a Free Software license, which allows for all these possibilities. Multiplatform compatibility is encouraged, but we must first ensure compatibility with free operating systems, to be consequent with the exposed ideas.

In order to implement these features, we ppted for the SuperCollider environment [50]. SuperCollider is a real-time audio processor and a object-oriented programming language.
It is free sofware and multiplatform, and there is a big community of technicians and artists using it for very diverse projects. There are libraries available that support GUI creation and standard interface protocols. It also provides a framework for audio processing units creation in C++.
Due to its interactive and object-oriented languange paradigms, it can provide the arbitrarily complex mappings and class structure that we need for our development. Furthermore, it is easily expandible by means of *quarks* or class library extensions.

3.2.2 SCLiss: SuperCollider Live Spatialization System

In this section we comment our implementation, which is named *"SCLiss: SuperCollider Live Spatialization System"*. Figure 3.2 shows the internal structure of the proposed implementation. By following the modularity criteria, we implemented three different functional blocks:

- Ambisonics Encoder
- Spatial Render
- Sound Scene

In following paragraphs we will discuss their features.

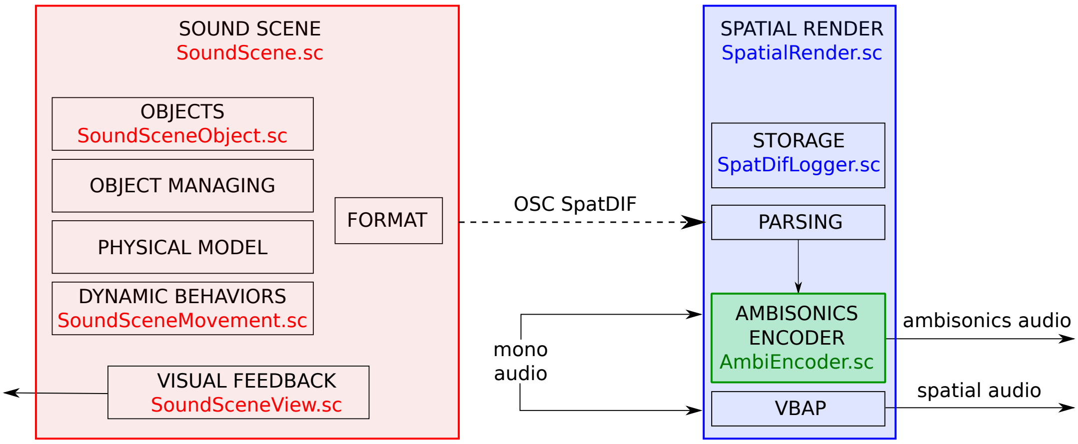


FIGURE 3.2: *SCLiss* structure and associated SuperCollider classes

SCLiss Ambisonics Encoder

The class *AmbiEncoder* implements a 3rd order Ambisonics encoder. It extends the functionality of the existing class *AmbIEM* in several ways:

- Possibility to choose several Ambisonics normalizations and channel order conventions
- Several sound source shapes available:
  - Point
  - Parallel
  - Meridian
  - Semi-meridian
  - Spherical surface

In order to assess validation of the developed Ambisonics encoder for a punctual source, its output for a set of positions was compared with the output of a *ground truth* Python encoder, provided by Barcelona Media.

Figure 3.3 shows the difference among the two versions, for a elevation angle of 0, computed along 100 azimuth values equal-spaced in the interval $[0, 2\pi)$.

Conversely, figure 3.4 shows the difference for an azimuth angle of 0, computed for 100 elevation values equal-spaced in the interval $[-\pi/2, \pi/2)$.

Each subplot shows the absolute error for a given 3rd order channel, following the ACN order convention, from left to right and from top to bottom. Error is computed, for each point, by taking the absolute value of the difference between values.

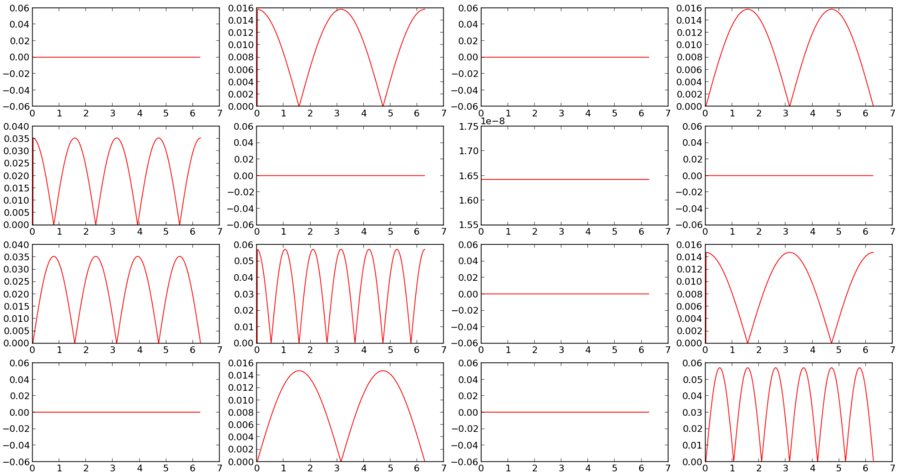


FIGURE 3.3: AmbiEncoder error in azimuth for a punctual source

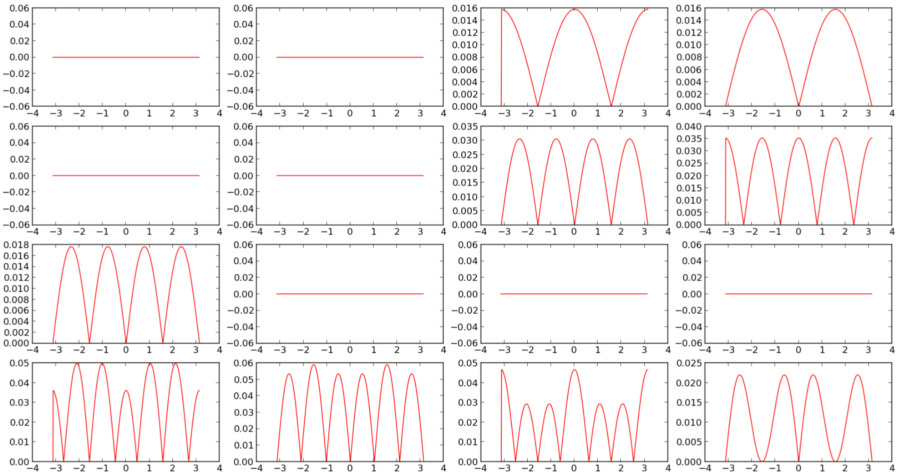


FIGURE 3.4: AmbiEncoder error in elevation for a punctual source

We can observe that the overall error is around 10^{-2}, which in any case might cause a perceptual difference. The sinusoidal-like shapes suggest that the difference is due to the different internal implementations of trigonometric functions between Python and SuperCollider.

For the rest of the implemented shapes, no theoretical validation was provided, due to the lack of *ground truth* code. However, subjective listening tests were performed by expert listeners, who stated a preliminary validation.

SCLiss Spatial Render

The spatial render capabilities are provided by the class *SpatialRender*. The user can choose among several spatialization techniques:

- HOA (via *AmbiEncoder* class)
- VBAP (with the existing class *VBAP*)

We must notice that, in the case of HOA, the audio output is in HOA Format; the user must use an Ambisonics decoder in order to obtain the spatialized audio. For these spatialization techniques, the spatial render features also distance cues, simulated by attenuation and low-pass filtering.

The class *SpatialRender* receives two kind of inputs: SpatDIF OSC messages with the required metadata, and audio channels through JACK [5]. There exists also the possibility of recording all SpatDIF messages for storage and subsequent playback, thanks to the *SpatDifLogger* class.

SCLiss Sound Scene

The class *SoundScene* implements and manages the sound scene simulation. It serves as a container for the different objects, implemented by the class *SoundSceneObject*. Objects can be either sound sources (related to a certain JACK channel), or user representations (in order to allow bottom-up interaction). Objects can be controlled jointly by means of groups, and can have different dynamic behaviors, implemented by the *SoundSceneMovement* class:

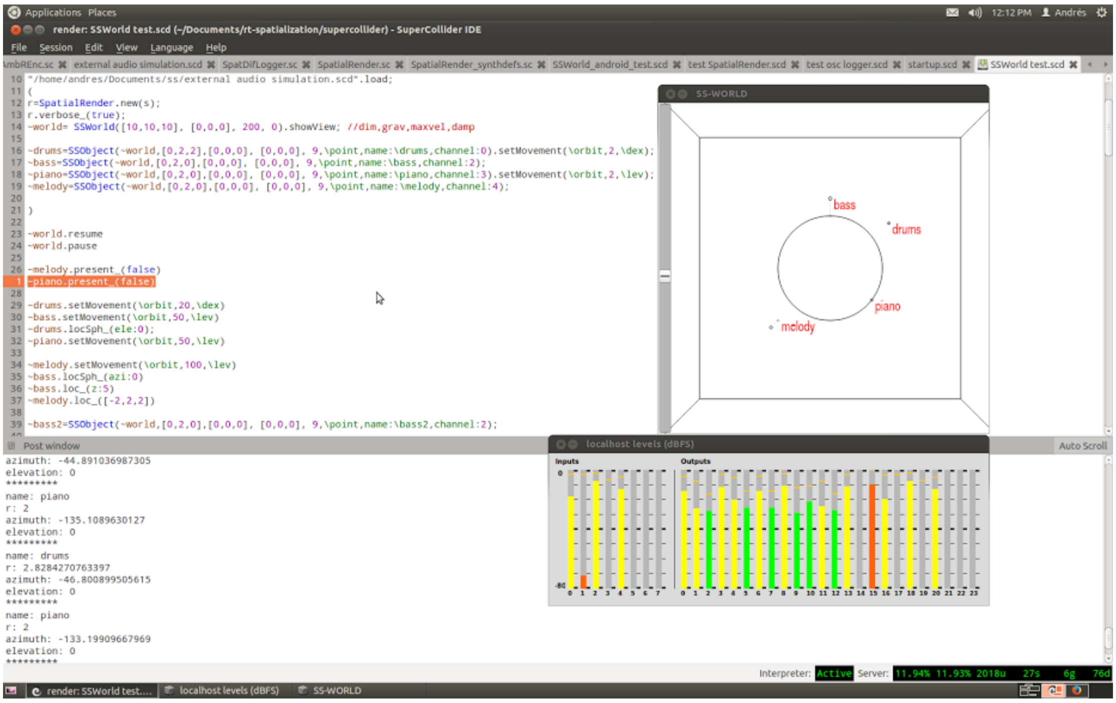
- Static
- Linear Motion
- Random Motion
- Brownian Motion

- Simple Harmonic Motion
- Orbital Motion

Furthermore, the physical model allows to configure several aspects of the sound scene, such as room parameters, physical constants (gravity, medium viscosity, wall friction), select object's physical properties and interact with them by means of forces.

All information simulated in the sound scene is sampled at a configurable rate, SpatDIF formatted and sent via OSC to the *SpatialRender* class or other compatible renders.

Finally, the sound scene provides visual feedback by means of the *SoundSceneView* class. We can observe the kind of visualization provided in Figure 3.5.



```
"/home/andres/Documents/ss/external audio simulation.scd".load;
(
r=SpatialRender.new(s);
r.verbose_(true);
~world= SSWorld([10,10,10], [0,0,0], 200, 0).showView; //dim,grav,maxvel,damp

~drums=SSObject(~world,[0,2,2],[0,0,0], [0,0,0], 9,\point,name:\drums,channel:0).setMovement(\orbit,2,\dex);
~bass=SSObject(~world,[0,2,0],[0,0,0], [0,0,0], 9,\point,name:\bass,channel:2);
~piano=SSObject(~world,[0,2,0],[0,0,0], [0,0,0], 9,\point,name:\piano,channel:3).setMovement(\orbit,2,\lev);
~melody=SSObject(~world,[0,2,0],[0,0,0], [0,0,0], 9,\point,name:\melody,channel:4);

)

~world.resume
~world.pause

~melody.present_(false)
~piano.present_(false)

~drums.setMovement(\orbit,20,\dex)
~bass.setMovement(\orbit,50,\lev)
~drums.locSph_(ele:0);
~piano.setMovement(\orbit,50,\lev)

~melody.setMovement(\orbit,100,\lev)
~bass.locSph_(azi:0)
~bass.loc_(z:5)
~melody.loc_([-2,2,2])

~bass2=SSObject(~world,[0,2,0],[0,0,0], [0,0,0], 9,\point,name:\bass2,channel:2);
```

```
azimuth: -44.891036987305
elevation: 0
*********
name: piano
r: 2
azimuth: -135.1089630127
elevation: 0
*********
name: drums
r: 2.8284270763397
azimuth: -46.800899505615
elevation: 0
*********
name: piano
r: 2
azimuth: -133.19909667969
elevation: 0
*********
```

Figure 3.5: SCLiss: Screenshot

3.2.3 Sample Application

In this section we will describe the prototype interface that we developed, which will cover two goals:

- Exemplify the *SCLiss* library in a working environment, and complement its description

- Serve as a basis for a further case study evaluation

For the interface design, we took into account the design considerations reviewed in Section 2.3.3. Our idea consists of an instrument for a solo performer, which might not be a trained/expert user, and who is able to control both spatialization and other sound characteristics. In order to make the instrument simple, and do not overload the user with multitude of sound parameters, we used a spatial multiplexing approach.

On one hand, we decided to devise a sequencer-like software, with predefined samples. The software is capable of synchronously reproduce the tracks, and perform mute/solo into the tracks. For this tasks, we use a MIDI keyboard interface with *pads*; the user does not need to continuously take care of the produced sound, and furthermore all functionalities are accessible rapidly with one only hand.

On the other hand, we opted for an Android interface for the spatialization control. It allows to be controlled with one hand, it might provide different active feedback modalities, it can run custom software and, principally, it provides a variety of physical sensors (accelerometer, gyroscope, etc). Since we are controlling sound sources in a three-dimensional space, we believe that a device with three-dimensional sensors can provide a good metaphor, thus making interaction more intuitive. It can also reduce the user's cognitive load, since humans are already used to three-dimensional motion.

Figure 3.6 depicts the full spatialization instrument implementation.

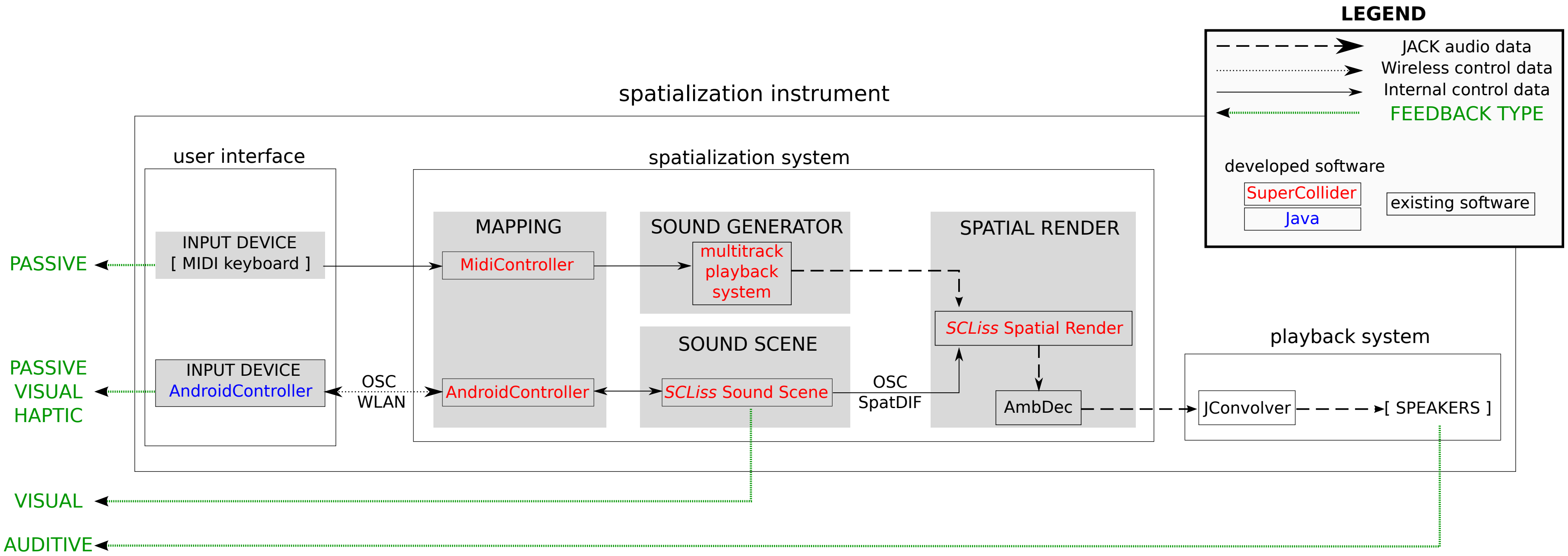


Figure 3.6: Spatialization Instrument: Implementation and Feedback modes

As we can observe, it follows the proposed schema described in Figure 3.1:

- **Spatialization System** The ground functionality (Sound Scene and Spatial Render) is provided by the *SCLiss* library. We might notice, as already mentioned, that the HOA Format *AmbiEncoder* output is redirected into an Ambisonics decoder, in this case the AmbDec software [1].
  The sound to spatialize consists on a set of mono tracks, specially produced for the instrument. The *multitrack playback system* is a custom code that works as a sampler, with synchronous reproduction and mute/solo capabilities.
  Finally, we observe that the mapping stage is splitted into two codes, one for each control interface.
- **Playback System** We connect the spatialized audio to the JConvolver software [6], which compensates the room acoustics effect. From there, the audio is sent to the speakers using the MADI protocol.
- **User Interface** There are two user interfaces: a MIDI keyboard-type controller, dedicated to the sound parameters, and an Android device, responsible for the sound spatialization.
- **Feedback** We tend to provide a variety of feedback modalities. Apart from the evident auditive feedback from the speakers, and the passive intrinsic to the user interfaces, we provide two more different feedback paths. One is the aforementioned visual feedback from *SoundScene* class, which can be shown through a projector or monitor. The other one is the active feedback which can potentially come from the Android device: visual and haptic/vibrotactile.

We implemented a set of different mappings and predefined situations, which are still in the iterative design process, but still can serve for *demo* purposes.

Finally, we must remark the fact that the whole system is Free Software licensed; not only the self developed implementations, but also the existing software used. In this way, we encourage availability, reproducibility and potential improvement from both the author and the community. As already mentioned, the source code can be found in [8].

3.2.4 Software Evaluation

In order to evaluate the implemented software, we must take into account the following issues:

- **Specific context** The system is an answer to a very specific problem (live sound spatialization). Because of that, it is intended for a specific user profile. Evaluating the system by people with no experience in the field might lead to wrong results, as well as time-consuming learning processes.
- **Implementation goal** SCLiss is not a goal by itself. It is a software tool intended for helping in the development of new spatialization instruments. Therefore, it would be desirable to evaluate not only its features, but also within the instrument development cycle context.
- **Artistic tool** As already mentioned, the implementation is a software tool intended for a range of artistic/creative situations. Consequently, it is difficult to measure its validity by means of traditional HCI methods based on *tasks*, from which measure *usability* or *efficiency*. It might be more adequate to measure it in terms of *usefulness* or *creativity enhancement*. The controversial article from Greenberg and Buxton [34] illustrates this difficulty.

As is the case in the design and HCI fields, the development methodology is based on the *iterative design cycle*: design, implementation and evaluation are performed repeatedly, until a certain development point is reached. In our case, we applied a two-step evaluation methodology, depending on the development stage; similar approaches can be found in the software evaluation literature (see, for instance, [29]).

Summative Evaluation It is performed in each step of the iterative design process, and can be related to the question *“Why is it bad?”*. In our case, we decided to use *assertive* evaluation: the developers where evaluating the system as it was in its developing stage. The main goal was to ensure that devised features were implemented. Formative evaluation was applied until the system fulfilled the requirements.

Formative Evaluation It is applied when the system development is almost accomplished, so it will not be affected radically. It is related to the question *“Which one is better?”* or *“How good is it?”*. Due to the aforementioned evaluation issues, we decided the appropriateness of a *field study*, to get information about the *usefulness* of the system. Since the software is available on internet [8], we encourage scientists and artists working in the spatialization instruments field to use it for their developments. We can devise to gather both subjective qualitative data from interviews and quantitative data from questionnaires. The same procedure might be used to perform a broad analysis, considering all existing real-time spatialization systems.

Chapter 4

Conclusion

4.1 Contributions

According to the goals defined previously in Section 1.3, and developed in Section 3, we enumerate the contributions of the carried research:

- Proposal of a holistic approach to analyse real-time sound spatialization, applying the existing knowledge in the HCI and DMI design fields, by means of the *spatialization instrument* concept.
- Critical review of existing real-time spatialization systems and spatialization instruments presented in last years in NIME and ICMC conferences.
- Development of a State of the Art spatialization system, encouraged for spatialization interface design and experiment replication, as well as a spatialization instrument prototype for individual music performance.

The developed software, the *SCLiss* library, contributes in following ways to the existing spatialization systems:

- First Linux-compatible 3D spatialization system
- First implementation of extended sources in HOA
- First SpatDIF-compatible implementation in SuperCollider
- First bottom-up approach in spatialization systems

4.2 Future Work

Based on the developed work, we devise a number of relevant related topics that can be carried in subsequent research:

- Evaluate the proposed framework for spatialization instruments development, in the form of a field-study among specialist users (as proposed in Section 3.2.4). The developed spatialization interface prototype might be also refined and used as a case-study.
- Develop spatialization instruments with the developed knowledge and tools, for different contexts and situations, which can highlight the potential of interactive 3D audio spatialization.
- Continue the research into the extended sources simulation in HOA, and complement it with perceptual evaluations.
- Refine the *SCLiss* implementation. Some of the devised improvements might be the compatibility with Binaural techniques, the implementation of 3D audio effects (spatial granulator, spectral splitter, etc), or the integration with the ongoing *Ambisonics Toolkit* developement.

Bibliography

[1] Ambdec, July 2014. URL `https://packages.debian.org/sid/ambdec`.

[2] Beastmulch, March 2014. URL `http://www.birmingham.ac.uk/facilities/BEAST/research/mulch.aspx`.

[3] Dolby atmos, March 2014. URL `http://www.dolby.com/us/en/consumer/technology/movie/dolby-atmos.html`.

[4] Game of life foundation, July 2014. URL `http://gameoflife.nl/en/`.

[5] Jack audio connexion kit, July 2014. URL `http://jackaudio.org/`.

[6] Jconvolver, July 2014. URL `http://kokkinizita.linuxaudio.org/linuxaudio/`.

[7] Omprisma, March 2014. URL `http://www.idmil.org/software/omprisma`.

[8] Scliss github repository, July 2014. URL `https://github.com/andresperezlopez/rt-spatialization/`.

[9] Spat, March 2014. URL `http://forumnet.ircam.fr/product/spat/`.

[10] Spatdif, March 2014. URL `http://www.spatdif.org`.

[11] Spatium, March 2014. URL `http://spatium.ruipenha.pt/`.

[12] Wave field synthesis, March 2014. URL `http://en.wikipedia.org/wiki/Wave_field_synthesis`.

[13] Wfscollider, March 2014. URL `https://github.com/GameOfLife`.

[14] Zirkonium, March 2014. URL `http://www.zkm.de/zirkonium`.

[15] Augustinus J Berkhout, Diemer de Vries, and Peter Vogel. Acoustic control by wave field synthesis. *The Journal of the Acoustical Society of America*, 93(5):2764–2778, 1993.

[16] Tina Blaine and Sidney Fels. Contexts of collaborative musical experiences. *Proceedings of the 2003 conference on New interfaces for musical Expression*, 2003.

[17] Mark A Bokowiec. V'OCT (Ritual): An Interactive Vocal Work for Bodycoder System and 8 Channel Spatialization. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, number June, pages 40–43, 2011.

[18] Katharina Bredies, Nick Alexander Mann, Jens Ahrens, Matthias Geier, Sascha Spors, and Michael Nischt. The multi-touch SoundScape renderer. In *Proceedings of the working conference on Advanced visual interfaces - AVI '08*, page 466, New York, New York, USA, 2008. ACM Press.

[19] Jeroen Breebaart, Sascha Disch, Christof Faller, Jürgen Herre, Gerard Hotho, Kristofer Kjörling, Francois Myburg, Matthias Neusinger, Werner Oomen, Heiko Purnhagen, et al. Mpeg spatial audio coding/mpeg surround: overview and current status. In *Audio Engineering Society Convention 119*. Audio Engineering Society, 2005.

[20] Jeroen Breebaart, Jonas Engdegård, Cornelia Falch, Oliver Hellmuth, Johannes Hilpert, Andreas Hoelzer, Jeroen Koppens, Werner Oomen, Barbara Resch, Erik Schuijers, et al. Spatial audio object coding (saoc)-the upcoming mpeg standard on parametric object based audio coding. In *Audio Engineering Society Convention 124*. Audio Engineering Society, 2008.

[21] William Buxton. There's more to interaction than meets the eye: Some issues in manual input. *User centered system design: New perspectives on human-computer interaction*, 319:337, 1986.

[22] Joanne Cannon and Stuart Favilla. Expression and Spatial Motion: Playable Ambisonics. In *Proceedings of the 2010 Conference on New Interfaces for Musical Expression*, pages 120–124, Sidney, Australia, 2010.

[23] B. Caramiaux, S. Fdili Alaoui, T. Bouchara, G. Parseihian, and M. Rébillat. Gestural Auditory and Visual Interactive Platform. In *Proc. of the 14th Int. Conference on Digital Audio Effects (DAFx-11),*, pages 1–4, Paris, France, 2011.

[24] Chris Carlson, Eli Marschner, and Hunter McCurry. The Sound Flinger: A Haptic Spatializer. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, number June, pages 138–139, 2011.

[25] J. Chadabe. Interactive Composing: an Overview. *Computer music journal*, 1984.

[26] J Daniel. *Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia.* PhD thesis, 2001.

[27] Wolfgang Fohl and Malte Nogalski. A Gesture Control Interface for a Wave Field Synthesis System. In *New Interfaces for Musical Expression*, pages 1–6, 2013.

[28] Edward Foley and Mark Paul Bangert. *Worship music: a concise dictionary.* Liturgical Press, 2000.

[29] Günther Gediga, Kai-Christoph Hamborg, and Ivo Düntsch. Evaluation of software systems. *Encyclopedia of computer science and technology*, 45(supplement 30):127–53, 2002.

[30] Matthias Geier and Sascha Spors. Spatial Audio with the SoundScape Renderer. In *VDT Convention*, 2012.

[31] Matthias Geier, Jens Ahrens, and Sascha Spors. Object-based Audio Reproduction and the Audio Scene Description Format. *Organised Sound*, pages 219–227, 2010.

[32] Michael A Gerzon. Surround-sound psychoacoustics. *Wireless World*, 80(1468): 483–486, 1974.

[33] Camille Goudeseune. Interpolated mappings for musical instruments. *Organised Sound*, 7(02):85–96, January 2003.

[34] Saul Greenberg and Bill Buxton. Usability evaluation considered harmful (some of the time). page 111, New York, New York, USA, 2008. ACM Press.

[35] Andy Hunt and Ross Kirk. Mapping Strategies for Musical Performance. *Trends in Gestural Control of Music*, 21:231–258, 2000.

[36] Andy Hunt and Marcelo M. Wanderley. Mapping performer parameters to synthesis engines. *Organised Sound*, 7(02):97–108, January 2003.

[37] Andy Hunt, MM Wanderley, and Matthew Paradis. The importance of parameter mapping in electronic instrument design. In *New Interfaces for Musical Expression*, pages 1–6, 2002.

[38] Robert JK Jacob, Linda E Sibert, Daniel C McFarlane, and M Preston Mullen Jr. Integrality and separability of input devices. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 1(1):3–26, 1994.

[39] Bridget Johnson and Ajay Kapur. Multi-touch interfaces for phantom source positioning in live sound diffusion. In *New Interfaces for Musical Expression*, page 4, 2013.

[40] S Jorda. Improvising with Computers: A Personal Survey (1989-2001). *Journal of New Music Research*, 2002.

[41] S Jorda. Digital Instruments and Players: Part I – Efficiency and Apprenticeship. In *Proceedings of the International Computer Music Conference*, pages 706–710, 2004.

[42] S Jorda. Digital Instruments and Players: Part II–Diversity, Freedom and Control. In *Proceedings of the International Computer Music ...*, 2004.

[43] Sergi Jorda. *Digital Lutherie: Crafting musical computers for new musics performance and improvisation.* PhD thesis, Universitat Pompeu Fabra, 2005.

[44] Sergi Jorda. Interactivity and live computer music. In N. Collins and J D'Escrivan, editors, *The Cambridge Companion to Electronic Music*, chapter 5, page 312. Cambridge University Press, Cambridge, UK, 2007.

[45] Daniel J. Levitin, Stephen McAdams, and Robert L. Adams. Control parameters for musical instruments: a foundation for new mappings of gesture to sound. *Organised Sound*, 7(02):171–189, January 2003.

[46] Georgios Marentakis, Nils Peters, and Stephen McAdams. Dj Spat: Spatialized interactions for DJs. *In Proceedings of the 2007 International Computer Music Conference (ICMC)*, pages 360–363, 2007.

[47] M Marshall, N Peters, AR Jensenius, J. Boissinot, Marcelo M. Wanderley, and J. Braasch. On the development of a system for gesture control of spatialization. 2006.

[48] Mark T Marshall. *Physical Interface Design for Digital Musical Instruments.* PhD thesis, McGill University, 2008.

[49] Mark T Marshall, Joseph Malloch, and Marcelo M Wanderley. Gesture Control of Sound Spatialization for Live Musical Performance. In *Gesture-Based Human-Computer Interaction and Simulation*, number 1, pages 227–238. 2009.

[50] James McCartney. Rethinking the computer music language: Supercollider. *Computer Music Journal*, 26(4):61–68, 2002.

[51] Ryan M. McGee and Matthew Wright. Sound Element Spatializer. In *Proceedings of the International Computer Music Conference 2011.* Ann Arbor, MI: MPublishing, University of Michigan Library, 2011.

[52] Chikashi Miyama, Jan C. Schacher, and Nils Peters. SPATDIF LIBRARY – Implementing the spatial sound descriptor interchange format. Technical Report 3, 2013.

[53] Davitt Moroney. Alessandro striggio's mass in forty and sixty parts. 2007.

[54] JJ Nixdorf and David 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*, pages 318–321, 2006.

[55] Saebyul Park, Seonghoon Ban, and Daeryong Hong. Sound Surfing Network (SSN): Mobile Phone-based Sound Spatialization with Audience Collaboration. In *New Interfaces for Musical Expression 2013*, pages 111–114, 2013.

[56] Nils Peters, Georgios Marentakis, and Stephen McAdams. Current Technologies and Compositional Practices for Spatialization: A Qualitative and Quantitative Analysis. *Computer Music Journal*, 35(1):10–27, March 2011. ISSN 0148-9267. doi: 10.1162/COMJ_a_00037. URL http://www.mitpressjournals.org/doi/abs/10.1162/COMJ_a_00037.

[57] Ville Pulkki et al. *Spatial sound generation and perception by amplitude panning techniques*. Helsinki University of Technology, 2001.

[58] Joseph Butch Rovan, Marcelo M Wanderley, Shlomo Dubnov, and Philippe Depalle. Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance. In *Proceedings of the Kansei-The Technology of Emotion Workshop*, 1997.

[59] Jan C. Schacher. Gesture control of sounds in 3D space. In *Proceedings of the 7th international conference on New interfaces for musical expression - NIME '07*, pages 358–362, New York, New York, USA, 2007. ACM Press.

[60] Timothy Schmele. Exploring 3D Audio as a New Musical Language. Master's thesis, Universitat Pompeu Fabra, 2011.

[61] Julien Signes, Yuval Fisher, and Alexandros Eleftheriadis. Mpeg-4's binary format for scene description. *Signal Processing: Image Communication*, 15(4):321–345, 2000.

[62] Sascha Spors, Rudolf Rabenstein, and Jens Ahrens. The theory of wave field synthesis revisited. In *124th AES Convention*, pages 17–20, 2008.

[63] Gabriel Vigliensoni and Marcelo M Wanderley. A Quantitative Comparison of Position Trackers for the Development of a Touch-less Musical Interface. *New Interfaces for Musical Expression*, 2012.

[64] MM Wanderley. Gestural control of music. *International Workshop Human Supervision and Control in Engineering and Music*, pages 632–644, 2001.

[65] MM Wanderley and Nicola Orio. Evaluation of input devices for musical expression: Borrowing tools from hci. *Computer Music Journal*, 26(3):62–76, 2002.

[66] Scott Wilson. Spatial swarm granulation. In *Proceedings of the 2008 International Computer Music Conference*. SARC, ICMA, 2008.

[67] Todd Winkler. Making motion musical: Gesture mapping strategies for interactive computer music. *ICMC Proceedings*, 1995.

[68] Trevor Wishart. *On Sonic Art*. Hardwood Academic Publishers, 1996.

[69] Mike Wozniewski, Zack Settel, and Jeremy R Cooperstock. A framework for immersive spatial audio performance. In *New Interfaces for Musical Expression 2006*, pages 144–149, Paris, France, 2006.