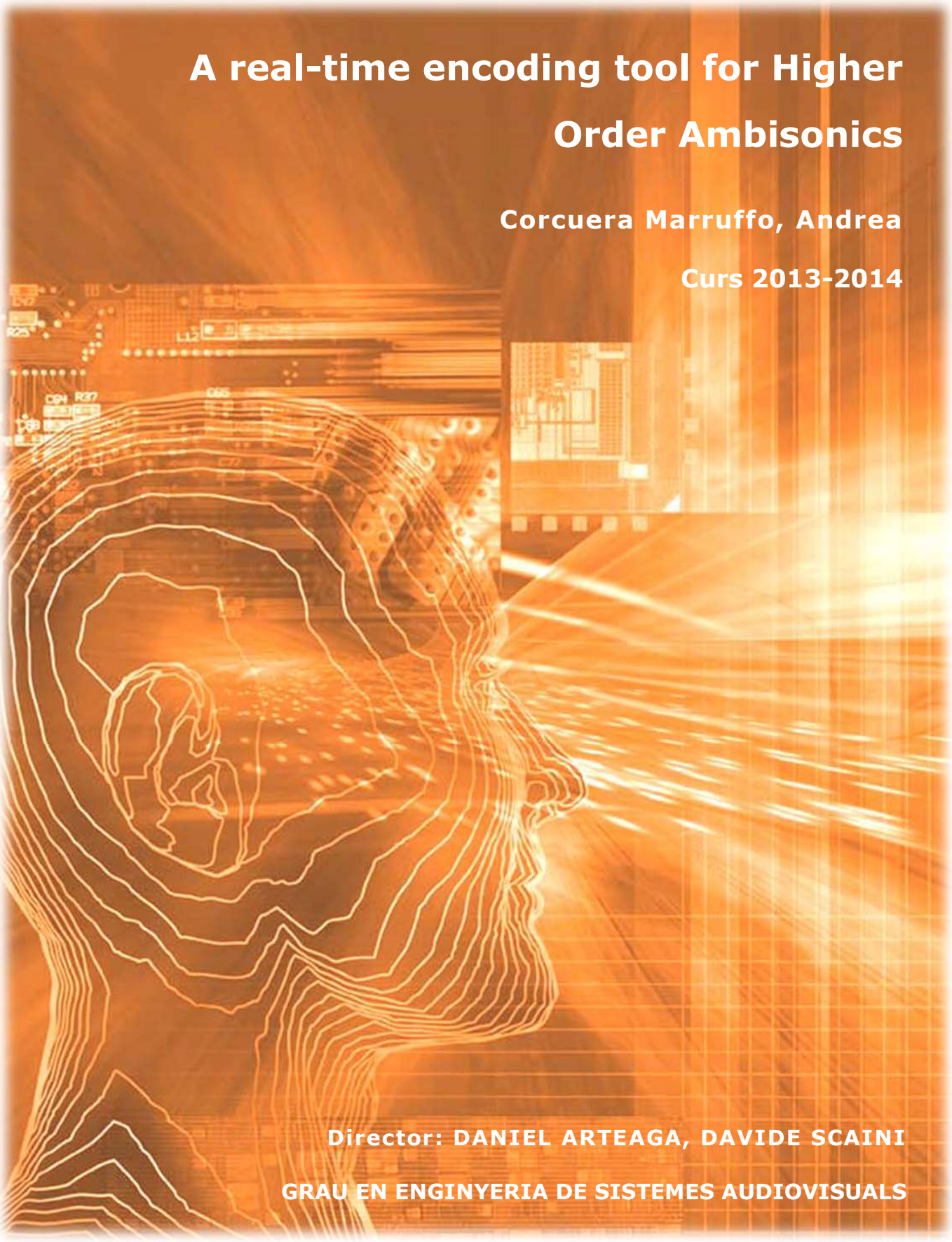
A real-time encoding tool for Higher Order Ambisonics

Corcuera Marruffo, Andrea

Curs 2013-2014



Director: DANIEL ARTEAGA, DAVIDE SCAINI

GRAU EN ENGINYERIA DE SISTEMES AUDIOVISUALS

upf. Universitat Pompeu Fabra Barcelona
Escola Superior Politècnica

Treball de Fi de Grau

Pompeu Fabra University

Escola Superior Politècnica

A real-time encoding tool for Higher Order Ambisonics

Author:
Andrea Corcuera

Supervisor:
Daniel Arteaga
Davide Scaini

2014

Abstract

This report presents the study of the spatial audio method called Higher Order Ambisonics, together with its physical and operational principles. In order to understand what is spatial audio, the human sound localization techniques are explained, and the different systems intended to create a sense of directionality are analyzed and evaluated. The traditional methods like stereo or surround are reviewed, as well as spatial audio techniques, such as channel-based binaural and VBAP (panning) systems, and the channel-independent methods WFS and Ambisonics, which are based on the sound field reconstruction.
First Order Ambisonics is an encoding method developed mainly by Michael A. Gerzon in the 1970s, based on the premise that the sound field can be represented as a superposition of plane waves. Higher Order Ambisonics, developed in the 1990s, extends this approach using more channels, meaning an improvement of the directionality and accuracy in the area where the sound-field is reproduced.
As an application of HOA spatialization, a 3D composition with point and extended sources is done by using a third order encoder. This encoder is designed and implemented with the goal of placing and moving sounds in all directions in space. For its development the programming language Pure Data has been used and a graphical interface is created to control the different sources of the developed composition.

Resumen

Este trabajo presenta el estudio del método de audio espacial llamado Ambisonics de Orden Superior, sus principios físicos y de funcionamiento. Con el fin de entender qué es el audio espacial, se explicarán las técnicas de los humanos de localización sonora, y los diferentes sistemas destinados a crear un sentido de direccionalidad se analizarán y evaluarán. Se presentan los métodos tradicionales como estéreo o surround, así como las técnicas de audio espacial, entre los cuales están los sistemas binaurales basados en canal y VBAP (panning), y los métodos independientes de canal WFS y Ambisonics, basados en la reconstrucción del campo de sonido.
Ambisonics de primer orden es un método de codificación desarrollado por Michael A. Gerzon en los años 70, basado en la premisa de que el campo sonoro puede representarse como una superposición de ondas planas. Ambisonics de Orden Superior, desarrollado en los 90, extiende este enfoque al uso de más canales, lo que significa una mejora de la direccionalidad y una precisión en el área donde el campo sonoro es reproducido.
Como aplicación de esta técnica, se realiza una composición 3D con fuentes puntuales y

extendidas usando un codificador de tercer orden. Este codificador es diseñado e implementado con el objetivo de permitir posicionar y mover sonidos en todas direcciones en el espacio. Para su desarrollo se ha usado el lenguaje de programación Pure Data, con el que se crea una interfaz gráfica para controlar las diferentes fuentes de la composición desarrollada.

Acknowledgements

En primer lugar me gustaría agradecer a mis tutores por todo el tiempo que me han dedicado, por su paciencia, su preocupación para que este proyecto alcance los objetivos deseados, y porque siempre han estado dispuestos a ayudarme en todo lo que necesitaba.

Quiero dar las gracias a mis padres, por haberme dado la oportunidad de estar en Barcelona durante cuatro años, estudiando lo que yo quería. Especialmente quiero agradecer a mi padre por su apoyo y sus consejos; sin tí, no estaría aquí ahora. A mi madre y a mi hermana Fiorella, gracias por estar siempre ahí y haber sido mi sustento en mis malos momentos.

Quiero recordar también a mis amigos y compañeros de clase con los que he pasado tantas horas en la biblioteca y he compartido tantos esfuerzos y dolores de cabeza.

Finalmente, quiero agradecer a Albert, por sus ánimos durante este tiempo y por haberme soportado cuando las cosas se ponían difíciles.

Contents

Abbreviations

HOA	**H**igher **O**rder **A**mbisonics
IID	**I**nteraural **I**intensity **D**ifference
ITD	**I**nteraural **T**ime **D**ifference
HRTF	**H**ead **R**elated **T**ransfer **F**unction
LF	**L**ow **F**requency

A mis padres y hermana

Motivation and goals

Since the beginning of the sound in cinema, technology of sound has been improved in order to offer the viewer the best cinema experience. The introduction of sound to cinema started with mono, only one speaker in the middle of the screen. Afterward, came stereo, surround sound, 5.1, 7.1 even 11.1. In the last decades, sound engineers have worked harder in order to go further, recreating the sound scene, as if the listener would be in the original place. But not only in cinema and television, spatial sound has also many applications such as in music or augmented reality.

Channel-based systems like stereo or 5.1 have been successfully welcomed by the users and they are integrated in the common electronic devices (televisions, radios, computers...), but other techniques, based in sound field reconstruction methods, are still not exploited commercially. However, this situation may change in the future. Spatial audio systems are being introduced, as can be seen in the recently incorporation to cinemas of Dolby Atmos technology.

The purpose of this report is to obtain a 3D composition by programming an encoding tool with Pure Data, easy to use and intuitive, which allows to place sound sources in 3D in space. In order to understand how this program works, the technique used in the tool is explained, and other methods of spatial audio are presented.

It is necessary to explain the mechanisms used by humans to localize sounds around themselves in order to understand what spatial audio means. For this reason, the first chapter introduces the human sound localization mechanisms as well as reproduction and recording techniques for different audio configurations. Some methods for spatial audio are reviewed, as well as their main strengths and weaknesses. Afterwards, the project focuses on HOA, which is explained in the second chapter, mainly the encoding stage. In the third chapter the developed tool is described, its operation and implementation are explained. Finally, in the fourth chapter the composition is exposed.

Chapter 1

Introduction

1.1 What is spatial audio?

When a person is in the natural environment he/she can perfectly know the localization of a sound, if its size is big or small and how far it is. An example of this is a car passing by. The car moves from his/her left to the right and the subject can distinguish in which position it is located. Closing the eyes the person can locate at any time where the car is: beside, in front, or behind, next or away from him/her. This sound changes if the object is, let's say, for example, a lorry. This object is bigger, therefore the sound is different and it is also possible to appreciate this size characteristic. In addition, the person can also hear more objects moving in this scene, each one with its particular localization and properties.

With the spatial audio tools one can produce this subjective soundscape. Spatial audio implies the generation of sound sources, or *images*, in the environment. It uses loudspeaker systems or headphones to reach this sensation.

1.2 Human sound localization

It is important to explain, first of all, the techniques that humans use to identify the location, distance and size of a sound. Imagine yourself sitting in a library. Although it is quite silent, you can perfectly hear the sound of a person walking in front of you, some people talking on your right, birds singing outside. Suddenly, a book falls down. When it occurs, its sound waves propagate in all directions and interact with the environment. They are reflected, absorbed or diffracted by the objects in the room. This interaction yields constructive and destructive interferences, and the waves reach the listener from

different directions, at different levels and times. Figure 1 illustrates how the sound waves interact with the environment. The *direct sound* comes directly from the sound source by the shortest path and normally provides the "first information". It supplies to the listener the main information about the sound localization.

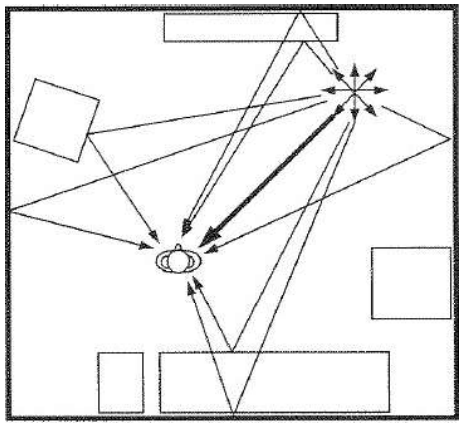


FIGURE 1.1: An example of how sound waves interact with the environment. The thick line is the direct sound, which arrives from the sound event to the listener. The thin lines are the indirect sound, arriving from many different directions [1]

Reflections and reverberation, consequence of the presence of objects and reflective walls, arrive from different directions and times. This *indirect sound* helps the listener to know some characteristics of the environment and the distance of the sound event [1].

1.2.1 Binaural localization

Our brain compares the signals received by the left and right ears to localize the sounds (mostly in the horizontal plane). The spectrum of the signal reaching each ear is different, since the amplitude and phase information differ. This binaural cues are called *interaural intensity difference* (IID) and *interaural time difference* (ITD).

Interaural time difference makes reference to the fact that when the source is to the side of the listener, the time at which the sound wave arrives at the ears differs because the signal has to travel different distances to reach the ears.

The other main cue is the *interaural intensity difference*, which arises from the fact that a sound will be louder at the ear closest to the sound event. This is because the sound wave has to travel more distance to arrive to the farther ear, and because the head absorbs and reflects the wave [1]. This *shadowing effect* caused by the head has different results depending on the frequency. Frequencies below 1500 Hz (frequency with a wavelength approximately equal to the diameter of the head) bends around the listener

due to their big wavelengths, and for high frequencies the head shadowing effect occurs at the farther ear.

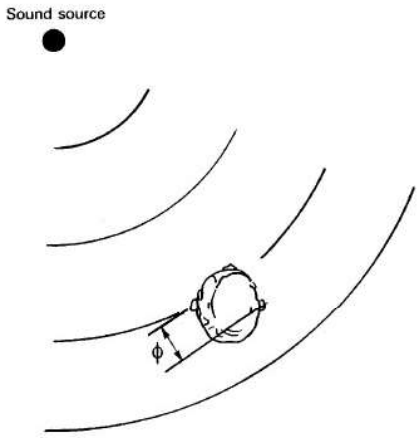


FIGURE 1.2: Sound waves arriving to a listener. ϕ is the phase differences between ears [2]

For sound localization at low frequencies (below 700Hz) it has been observed that the phase difference between the two ears (ITD mechanism) helps to discern the position of the source [3]. For low frequencies, the head is not an obstacle to this sound waves with large wavelengths, so the amplitude differences (IID) of the sound that arrives to the ears is negligible and therefore, it is not an useful cue to use in this case. However, at high frequencies (about 1500 Hz) the ITD is not effective but there is a difference in amplitude (IID), which helps to locate the source. For sounds with frequencies between 700 and 1500Hz, both differences are useful for localization.

1.2.2 Monoaural localization

True 3D implies that sounds are perceived outside the head (*externalization*). With ITD and IID the sound image only moves along the interaural axis, an axis left/right inside the head; i.e. IID and ITD are useful for *lateralization* but the listener can not determine if the sound is in front, above or behind. This region of positions where all sounds yield the same ITD and IID is called *cone of confusion* [1].

It is possible to resolve this ambiguity with the distinctive filter of the *pinnae* (external ears), head and torso, which describes the interaction between the sound and the listener's body. The sound waves that reach the subject, in addition to arrive with a different level and time, interact with the listener's *pinnae*, inner ear, head and torso. This interaction produces reflections and resonances that modify the spectrum, which is characterized by the Head Related Transfer Functions.

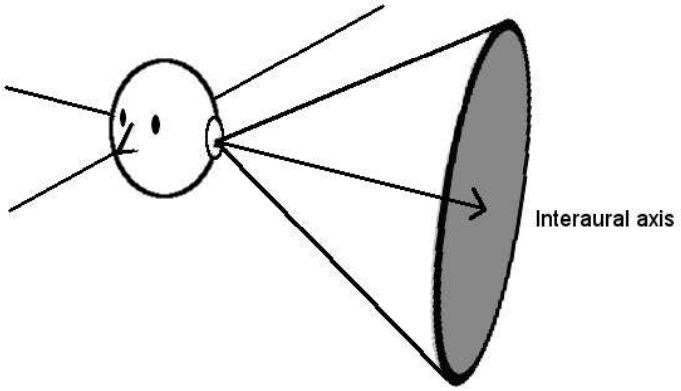


FIGURE 1.3: Cone of confusion.

1.2.3 Head Related Transfer Functions

The Head Related Transfer Functions (HRTFs) represent the transformations of the signal produced by the body's listener. Because these functions are based on their own body, each person has his own HRTFs, which differ from one person to person. The HRTFs contain the binaural differences (ITD and IID) and the filter effects caused by the *pinnae*, ear canals, head and torso.

1.3 Techniques for spatial audio

1.3.1 Binaural techniques

Head-related stereophony is based on the premise that a sound source can be positioned using the information perceived by our ears. In other words, this technique takes into account the presence of the head, and aims to re-create the acoustic pressure that one would receive in a natural listening, with headphones or loudspeakers.

A dummy head is typically used for binaural encoding. Two microphones are placed in the position of the ears [4]. This approach capture all the necessary cues for an accurate spatial perception (IID, ITD and monoaural cues) in accordance with the morphology of the head. The dummy head can be substituted by a "real head" by placing little microphones in the listener's ears. A sphere with the size of a head and microphones in the place of the ears can also substitute the dummy head (*Kugelflächenmikrofon*).

The easiest way to reproduce this recorded signals is to reproduce them via headphones, since the signals that arrive to the ears are independent. If the binaural signals are reproduced via loudspeakers (*transaural stereo*), the resulting signal arriving at the ears will be affected by the crosstalk between the emitted signals. The signal intended to arrive to one ear, would reach the another one with a different time and level. To obtain a good result using loudspeakers the signal should be processed with some crosstalk cancellation filters.

1.3.2 Discrete panning techniques

The most common panning technique is called Amplitude panning, in which the loudspeakers are fed with the same signal but with different gains. In international standards, n.m. reflects the number of channels, where n is the number of front channels and m is the number of LF channels.

Traditional methods: stereo, 5.1, 7.1...

In the initial stages, all the sound systems were monophonic, only one channel, typically reproduced by one loudspeaker. In the 1930's, a British engineer called Alan Blumlein developed a system [4, 5] that created the sensation of directionality, the stereo. He considered that the principal cues used by humans to identify the direction of a sound source were the amplitude and time differences between ears, thus he designed a system based on this premise.

Stereophonic sound is a system that has two independent audio signal channels in order to represent the natural hearing (2.0 stereo). These two signals have a different level and phase, so this method needs two or more loudspeakers, placed typically with an angle of 60º from each other. When they are reproduced, they create the impression of hearing the sound from various directions in between the loudspeakers. For listening the sound to our right, then the sound level of the right speaker should be higher than the left speaker sound, creating the illusion of directionality.

There are many different techniques to record stereo [2]. One of these is the XY or coincident-pair technique, which uses two directional microphones (tipically cardioid) located in the same place with an angle between each other. They are located as close as possible, with their capsules one over the other. In general, this technique tends to have less depth and less sense of spaciousness due to the lack of differences in time and phase.

If there are two bidirectional microphones mounted at the same point with an angle of 90º between each other, then this format is called Blumlein pair. Despite it has a

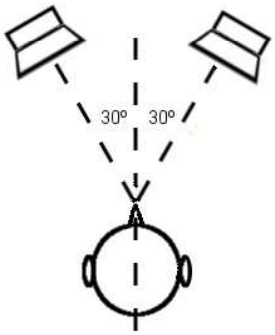


FIGURE 1.4: Stereo. Equilateral triangle with the listener located to the rear of one vertex of the triangle. It is the perfect location (*sweet spot*) to perceive the apparent position of sound sources (phantom images) in between the speakers

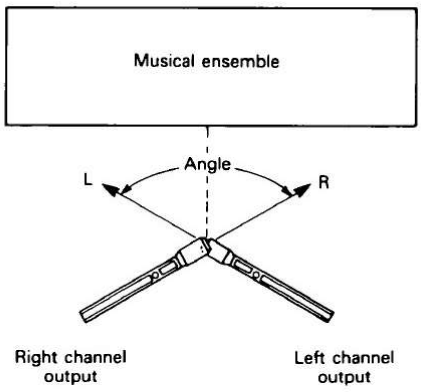


FIGURE 1.5: Coincident-pair technique. Two microphones are mounted in the same place with an determinate angle. Reproduced from [2]

wispy localization due to the antiphase result between channels, this technique provides a sharp sense of depth.

M/S (Mid-Side) technique uses two coincident microphones. The middle one is often a cardioid facing ahead and the other one is bidirectional pointing sideways [2]. The right and left channels are obtained by summing the outputs of this two capsules:

$$\begin{cases} L = M + S \\ R = M - S \end{cases}$$

Another method is called spaced-pair or AB technique. It uses two identical omnidirectional microphones spaced a distance of between 40 and 60 cm, i.e., they capture

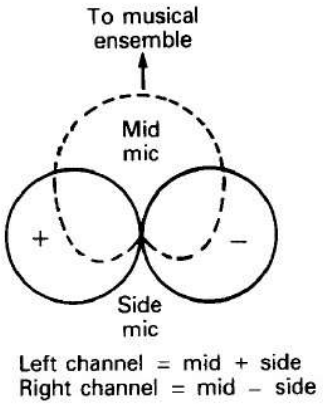


FIGURE 1.6: M/S technique. Figure reproduced from [2]

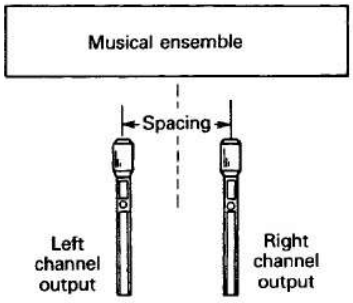


FIGURE 1.7: Spaced-pair technique. Two microphones are placed some distance apart [2].

the time of arrival information and some level differences. This method provides a less precise image than Blumlein pair but a better sensation of spatiality [2].

The ORTF method, developed in the 1960s at the *Office de Radiodiffusion Télévision Française* (ORTF), is a combination between XY and AB. It combines the spaciousness of the timing difference provided by the two microphones spaced 17cm and the level difference, since this two cardiod microphones are spaced with a 110º angle.

One of the best known surround reproduction techniques is the called "5.1 surround", proposed by the International Telecommunication Union (ITU). It consist in five independent channels (left front, centre, right front, left surround and right surround) and a sixth optional for low frequency effects [6]. The two surround speakers allow to create sound images also behind the listener while the center one helps to produces a real image, keeps the dialogue on screen even if the listener moves. This center channel was added to avoid the "hole in the middle" between speakers problem. However, for a good image and sense of envelopment, all the speakers should be the same model, and the listener must be in one specific position at the same distance from each speaker (*sweet*

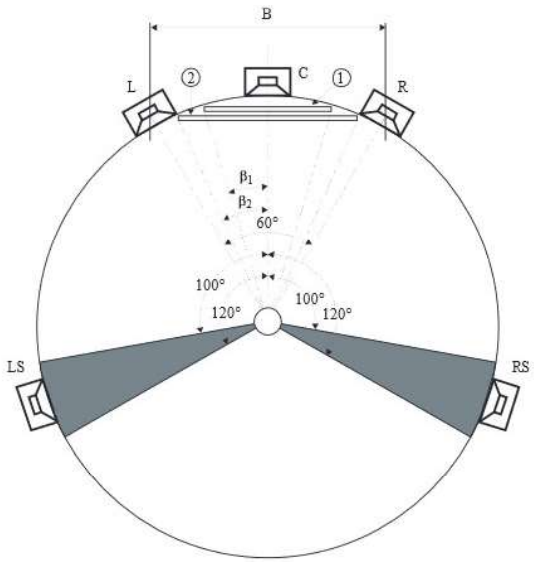


FIGURE 1.8: 5.1 configuration extracted from [6]

spot). In addition, this format does not support periphony (it is only intended to the azimuth localization).

Let's take a look at some techniques for recording in surround sound [2] :

- **SoundField 5.1 Microphone System.** This system uses one multi-capsule microphone. It produces two different sets of signals, called A-format or B-format. This microphone has four closely spaced capsules arranged in a tetrahedron which provide four signals, that the decoder translates into L, R, C, LR, RR and subwoofer outputs. A more detailed explanation will be reported later.
- **DMPMethod (Decca Tree).** This system, originally designed for recording orchestral music, has three omnidirectional microphones placed in a "T" pattern were suspended above the conductor, and a stereo spaced or coincident-pair for the surround ambiance. The side channels are panned, and the center one feeds the center channel in 5.1 setup.
- **NHK Method (Fukada Tree).** This system consist of: two near-coincident mics that feed front-left and front-right channels, a center one for the center channel, up to three pairs of mics for the rear, and a pair of spaced microphones in order to add spaciousness.

The 7.1 technique uses eight audio channels in order to improve sound localization and enhance definition. This system adds two additional speakers to the conventional 5.1, meaning a more immersive sound-field providing a good sound to all seats in the room.

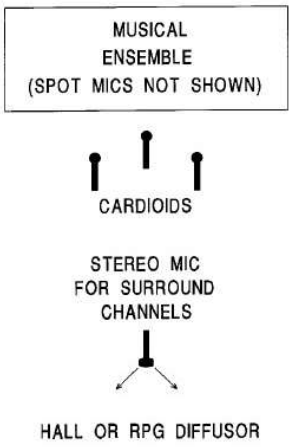


FIGURE 1.9: DMP surround sound recording method. Image reproduced from [2]

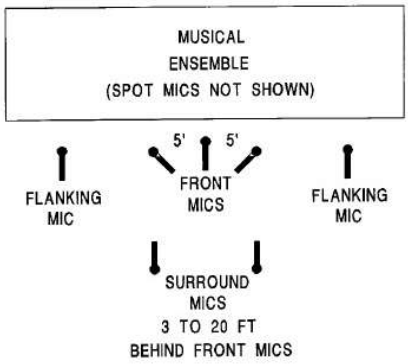


FIGURE 1.10: NHK surround sound recording method. Image reproduced from [2]

Vector Base Amplitude Panning (VBAP)

VBAP is a extension of stereophonic amplitude panning techniques, which consist of feed only one, two or three adjacent speakers for positioning virtual sources in different locations in between loudspeakers [7]. The number of loudspeakers can be arbitrary, and can be located in an arbitrary 2D or 3D configuration. The "virtual" direction of the sound determines which loudspeakers must be fed. This method is layout independent, i.e. no longer *n.m.* The key difference is that it is an object based approach, the decoder receives the audio signal and the position metadata in order to generate the signals that feed the loudspeakers, thus the sound source is placed.

In a horizontal plane, a virtual sound position is created by using the tangent law [8]. This law can be generalized for three dimensional layout in the same way (assuming the listener is located in the center of the speakers setup).

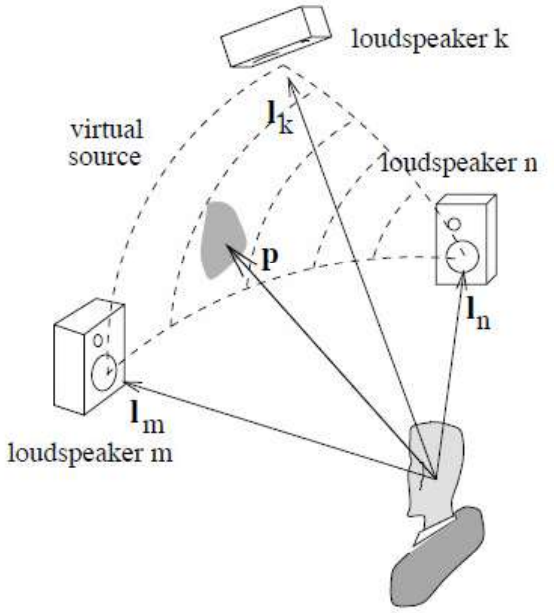


FIGURE 1.11: Three dimensional VBAP. This method use the differences between two or three loudspeakers for the panning of sources. Image extracted from [8]

1.3.3 Sound field reconstruction methods

Wave field synthesis

The aim of sound field reconstruction methods is to recreate the acoustic variables of the environment (pressure, velocity). One example is the Wave Field Synthesis (WFS) theory is based on the Huygens principle, that states that every point of a initial wave front, can be considered as a spherical secondary source of a wave which travels in all directions with the same frequency of the primary source. This virtual secondary sources can be seen together as a new sound field identical to the sound field of the primary sound [9, 10]. This principle is applied by using a large array of speakers, spaced very closely. This system recreates wave fronts that would reach a listener if he/she would be in the natural environment. In other words, the aim of this technique is no longer to create the psycho-acoustic perception, but a given sound-field. WFS is also an object based method.

Ambisonics

Ambisonis is a spatial audio theory based on the spherical harmonics decomposition of the sound field. It provides *periphony* (full surround with elevation) with only four channels in its simplest configuration. A simple Ambisonics encoder needs the source signal S and the horizontal and elevation angles of the location where the source wants

to be positioned. Thus, Ambisonics can be treated in terms of sound objects, but it is not mandatory. There is an intermediate layout independent format with a fixed number of channels which is the Ambisonics format. The encoded signal is not used to feed one independent loudspeaker. Each speaker contains weighted signals, so all the layout works together to recreate the acoustic environment. In the next chapter a more detailed description of Ambisonics is provided.

1.3.4 Pros and cons of each technique

This three approaches have different drawbacks. These disadvantages can be see in terms of: homogeneity between speakers (the sound has the same quality in all positions?), size of the listening area (how much can the listener move?), and quantity of needed loudspeakers. Depending of the application, one or another technique is chosen.

Binaural techniques

As previously mentioned, binaural technique uses the information provided by HRTF functions, which are related with each person. Therefore, despite this technique provides a very accurate sense of spatiality, this method uses a subjective measure. So for other people, with different head and torso shape, the performance would change. In addition, if the person moves, this movement must be registered for update the HRTF filters in order to maintain the absolute position of the sound, otherwise the sound location would be wrong. This problem could be solved with *head tracking*, which means that the movements are monitored, but this implicates higher costs [4].

The same problems with individual differences in HRTFs appear in transaural stereo. In addition, despite the advantage of the use of only two loudspeakers, it has the drawback of having a small sweet spot.

Panning techniques

The advantages of these techniques are that they are very simple to implement, they provide a good localization within a large sweet spot, and can be easily modified for different loudspeaker layouts. However, these methods are not homogeneous, that is, the sound source has a better resolution when the source location falls in the position of a loudspeaker. In addition, sometimes the listener can have the impression that the sound source "jumps" from one speaker to another along a panning [7].

Sound field reconstruction techniques

The main advantage of these methods is that there are independent from the loudspeaker layout. In contrast to the traditional stereo or surround, which are channel-based, these

techniques can encode the signals regardless of the reproduction layout and they aim to fully reconstruct the sound-field instead to create a phantom image, thus these techniques are homogeneous.

Problems with Ambisonics technique include: its sweet spot is quite small and has poor localization at its low orders [7].

Systems based on WFS generate a wave field with a big listening area (large *sweet spot*) and it has a high precision for positioning sound sources. However, this method has a high cost since it involves the use of a large number of loudspeakers (infinite planes with infinite sources). In addition, artifacts occurs at frequencies whose wavelengths are smaller than the distance between speakers.

Chapter 2

Ambisonics

Ambisonics is a spatial audio technique developed in the 70s mostly by Gerzon. It provides a three-dimensional sound reproduction, since it includes height information. As mentioned above, Ambisonics is a spatial audio theory based on the decomposition of the sound-field in spherical harmonics. But, what does it mean?

2.1 Mathematic fundamentals

The mathematical representation of Ambisonics for encoding a sound field is based on the directional properties given by the spherical harmonics, and the assumption that the sound field can be decomposed in a superposition of plane waves since it is considered that the loudspeakers are far away from the listener.

For the purpose of analyze this combination of plane waves, the wave equation is used. This equation can be represented in the spherical coordinate system, where θ (azimuth) and ϕ (elevation) angles provide the directional information. This leads to the Fourier-Bessel series [11, 12]:

$$p(\vec{r}) = \sum_{m=0}^{\infty} j^m j_m(kr) \sum_{n=-m}^{m} B_{mn} Y_{mn}(\theta, \phi) \tag{2.1}$$

where the wave number is $k = 2\pi f/c$, $j_m(kr)$ is the spherical Bessel function , B_{mn} represents the Ambisonics channels, and the angular functions $Y_{mn}(\theta, \phi)$ are called spherical harmonics. These real spherical harmonics are defined as follows:

$$Y_{mn}(\theta,\phi) = N_{mn}P_{m|n|}(\sin\phi) \times \begin{cases} \cos(|n|\theta) & \text{if } n > 0 \\ \sin(|n|\theta) & \text{if } n < 0 \\ 1 & \text{if } n = 0 \end{cases} \tag{2.2}$$

where P_{mn} are a Legendre function with degree n and order m and N_{nm} is the normalization factor[1].

The set of spherical harmonics form an orthonormal basis of the space of functions defined over the sphere. Namely, the integral of any two spherical harmonics is 0 if the harmonics are different, 1 if they are identical:

$$\int_{sphere} d\theta d\phi \cos\phi Y_{mn}(\theta,\phi)Y_{m'n'}(\theta,\phi) = \delta_{mm'}\delta_{nn'} \tag{2.3}$$

where $\delta_{mm'}$ and $\delta_{nn'}$ are the Kronecker delta.

$$\delta_{mn} = \begin{cases} 1 & \text{if } m = n \\ 0 & \text{if } m \neq n \end{cases} \tag{2.4}$$

The encoding Ambisonic functions arise from the decomposition of a plane wave in spherical harmonics. The sound field reproduction is restricted to the origin of the sphere centered on the sweet spot (origin of the coordinate system), thus it is assumed that there are no virtual sources inside this area. In practice, this representation of a field can't have an infinite number of components, since it would involve an infinite number of microphones/loudspeakers for recording/reproduction, hence the function is expressed up to an order M:

$$\tilde{p}(\vec{r}) = \sum_{m=0}^{M} j^m j_m(kr) \sum_{n=-m}^{m} Y_{mn}(\theta_S,\phi_S)Y_{mn}(\theta,\phi) \tag{2.5}$$

The field approximation is obtained by the decomposition (2.5), where it is described by the coefficients B_{mn}. An Ambisonic 3D representation is closely related to this expression, where the coefficients are the angular functions $Y_{mn}(\theta,\phi)$ (the radial functions are no codified). Therefore, the first four components can be recognized: the pressure $W = B_{00}$, and the acoustic velocity components (or its gradient) $X = B_{11}$, $Y = B_{1-1}$

[1]The normalization used in this project is "normalized 3D", so the $Y_{mn}(\theta,\phi)$ function will be denoted with the exponent tag $^{(N3D)}$

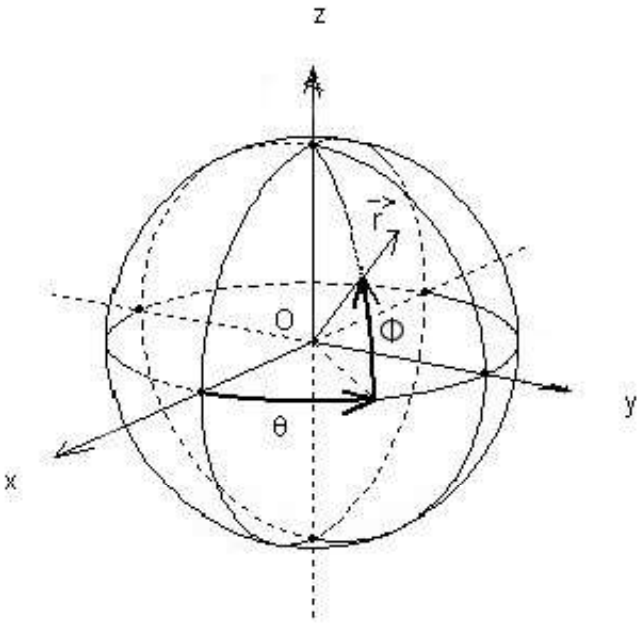


FIGURE 2.1: Spherical coordinate system where the radius is r, the azimuth angle is represented by θ and the elevation angle is ϕ. Extracted and modified from [12]

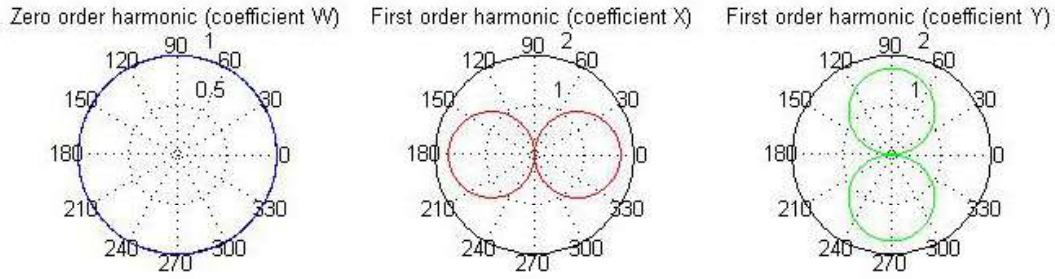


FIGURE 2.2: Spherical harmonics in 2D

and $Z = B_{10}$. The two-dimensional Ambisonics components B_{mn} come from the Fourier series, as can be seen in the Figure 2.2.

2.2 Ambisonics encoding

The spherical harmonic functions $Y_{mn}(\theta, \phi)$ define the Ambisonics encoding functions. Therefore, the Ambisonics channels are given by the expression:

$$B_{mn}(t) = S(t)Y_{mn}(\theta_S, \phi_S) \tag{2.6}$$

where the $Y_{mn}(\theta_S, \phi_S)$ are the encoding coefficients, related to the direction of the given sound source $S(t)$. Hence, to encode a source, some gains, which are the spherical harmonic functions calculated in (θ_S, ϕ_S), are applied to the signal $S(t)$.

B-format

For a first order, the signal Ambisonics components are computed as follows [13] (N3D normalization):

$$\begin{cases} W = B_{00} = SY_{00}(\theta_S, \phi_S) = S \\ X = B_{11} = SY_{11}(\theta_S, \phi_S) = S\sqrt{3}\cos\theta_S \cos\phi_S \\ Y = B_{1-1} = SY_{1-1}(\theta_S, \phi_S) = S\sqrt{3}\sin\theta_S \cos\phi_S \\ Z = B_{10} = SY_{10}(\theta_S, \phi_S) = S\sqrt{3}\sin\phi_S \end{cases} \tag{2.7}$$

However, this is not the traditional normalization in first order Ambisonics, where ussually the Furse-Malham normalization is used. These four signals, called B-format, are equivalent to an omnidirectional and three figure-of-eight microphones:

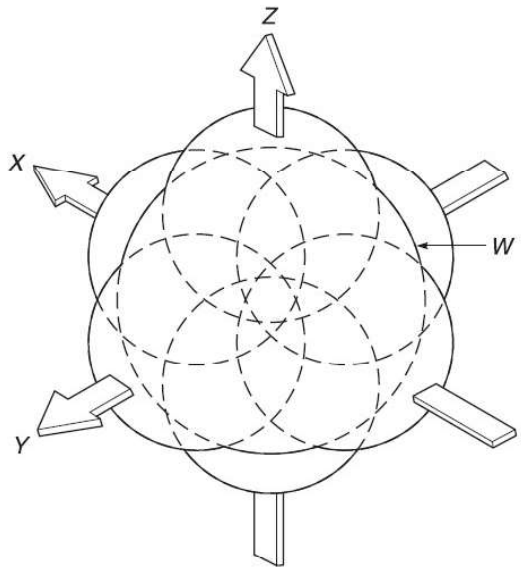


FIGURE 2.3: B-format components: omnidirectional pressure component W, and the three velocity components X, Y, Z. Extracted from [4]

- Omnidirectional component (pressure signal) from the omnidirectional microphone
- A front-back information (velocity component) from the figure-of-eight microphone pointing forward

- A left-right information (velocity component) from the figure-of-eight microphone pointing side to side
- A up-down information (velocity component) from the figure-of-eight microphone pointing upward

Thus, first order Ambisonics can be recorded using four microphones: one omnidirectional for the pressure and three for the velocity measurement.

It can be used equivalently, as mentioned above, the Soundfield microphone developed by Craven and Gerzon. The four signals measured by the four microphones arranged in a tetrahedron are called *A-format*: LF (left-front), LB (left-back), RF (right-front) and RB (right-back).

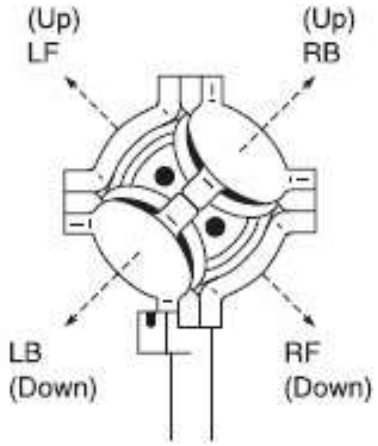


FIGURE 2.4: A-format microphone directions: LF, LB, RL, RB. Four microphones placed on the faces of a tetrahedron. Extracted from [4]

A-format must be processed since it can not be used directly. B-format components can be deduced from A-format. In case that the reproduction is horizontal, only three of this four components are needed. Z component is not used and the elevation angle equals zero.

Higher order Ambisonics encoding

As can be seen in the image 2.5, each additional set of higher order components improve the directional localization, so the sound field reconstruction gets more accurate if the order is increased.

For periphonic Ambisonics there are $(2m+1)$ components for each order m, so the total number of channels for a M order is $N = (M+1)^2$ for 3D reproduction and $N = (2M+1)$

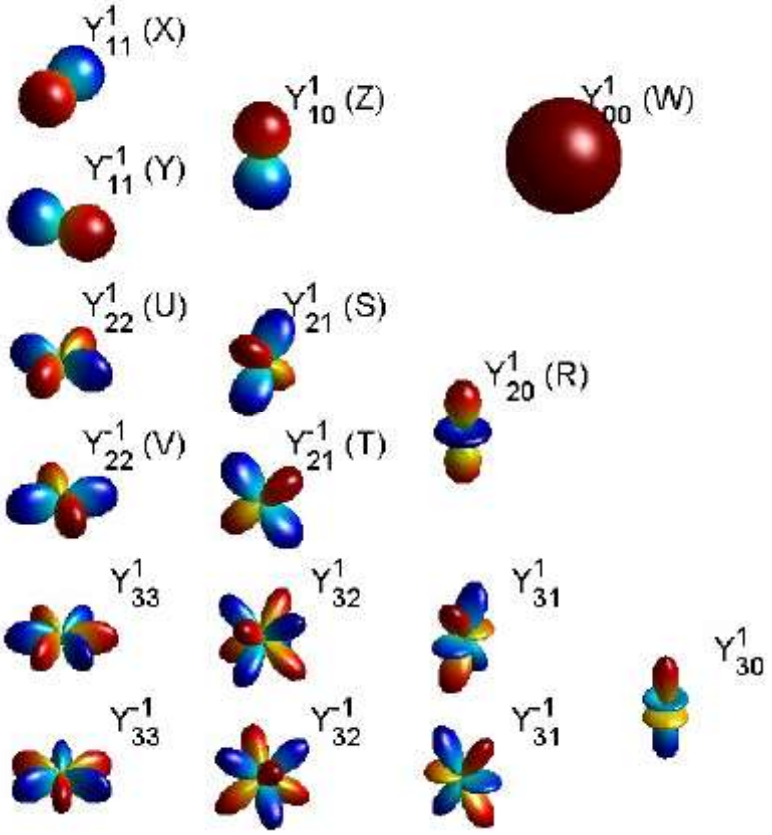


FIGURE 2.5: Spherical harmonics up to 3 degree. Reproduced from [12]

for 2D reproduction. This project deals with the third order, so, in this case, the number of components will be $N = (3 + 1)^2 = 16$.

The higher order coefficients can be obtained by calculating the expression for the real spherical harmonics of higher than one order. A calculation of the Ambisonics components up to third degree (m=3) is reported in the table 2.1

The Ambisonics channel order used in this project is the ACN (*Ambisonic Channel Number*). Thus, the order changes from the traditional B-format (W, X, Y, Z) to: *W, Y, Z, X*. And for second and third order *V, T, R, S, U* and *Q, O, M, K, L, N, P* accordingly.

Extended sources

In case of need a source covering an area larger than a point (for example, in a practical case in which the idea is to recreate a concert with audience placed in most of the front

Order	**(m, n)**	**B_{mn}**	$\mathbf{Y_{mn}^{(N3D)}}(\theta, \phi)$
0	(1, 0)	W	1
1	(1, -1)	Y	$\sqrt{3}\sin\theta\cos\phi$
	(1, 0)	Z	$\sqrt{3}\sin\phi$
	(1, 1)	X	$\sqrt{3}\cos\theta\cos\phi$
2	(2, -2)	V	$\sqrt{15}/2\sin(2\theta)\cos^2\phi$
	(2, -1)	T	$\sqrt{15}/2\sin(\theta)\sin(2\phi)$
	(2, 0)	R	$\sqrt{5}(3\sin^2\phi) - 1$
	(2, 1)	S	$\sqrt{15}/2\cos(\theta)\sin(2\phi)$
	(2, 2)	U	$\sqrt{15}/2\cos(2\theta)\cos^2\phi$
3	(3, -3)	Q	$\sqrt{7}\sqrt{5/8}\cos(3\theta)\cos^3\phi$
	(3, -2)	O	$\sqrt{7}\sqrt{5/8}\sin(3\theta)\cos^3\phi$
	(3, -1)	M	$\sqrt{7}\sqrt{15}/2\cos(2\theta)\sin(\phi)\cos^2(\phi)$
	(3, 0)	K	$\sqrt{7}\sqrt{15}/2\sin(2\theta)\sin(\phi)\cos^2(\phi)$
	(3, 1)	L	$\sqrt{7}\sqrt{3/8}\cos\theta\cos\phi(5\sin^2\phi - 1)$
	(3, 2)	N	$\sqrt{7}\sqrt{3/8}\sin\theta\cos\phi(5\sin^2\phi - 1)$
	(3, 3)	P	$\sqrt{7}\sin\phi(5\sin^2\phi - 3)/2$

TABLE 2.1: Ambisonics normalized 3D encoding functions for point sources

part, so the source is no longer a point), the computations vary a little. The coefficients for extended sources can be obtained by copying the same point source in all area where the sound is required -not a very efficient method- or by applying an integral over the desired part of the sphere as can be seen in the following expression:

$$B_{mn} = 1/A \int_S d\theta d\phi \cos\phi Y_{mn}(\theta, \phi) \tag{2.8}$$

where S is the angular surface covered by the extended source and A is the area of this surface (in stereoradians)

In the case of this report, a *ring source* is implemented, thus the area that must be integrated has the shape of a ring. The coefficients for this kind of source are reported in Table 2.2.

2.3 Ambisonics decoding

Since Ambisonics encoding signals can not be reproduced directly through the loudspeakers – because one specific channel is not intended to be reproduced in one particular speaker (like in stereo or 5.1 system) – it is necessary to have a specific decoder associated to the loudspeaker layout. In this way the encoded composition can feed any

Order	**(m, n)**	**$\mathbf{B_{mn}}$**	**$\mathbf{B_{mn}}(\theta, \phi)$**
0	(1, 0)	W	1
1	(1, -1)	Y	0
	(1, 0)	Z	$\sqrt{3}\sin\phi$
	(1, 1)	X	0
2	(2, -2)	V	0
	(2, -1)	T	0
	(2, 0)	R	$1/2\sqrt{5}(-1+3\sin(\phi)^2)$
	(2, 1)	S	0
	(2, 2)	U	0
3	(3, -3)	Q	0
	(3, -2)	O	0
	(3, -1)	M	0
	(3, 0)	K	$-1/4\sqrt{7}(1+5\cos 2\phi)sin(\phi)$
	(3, 1)	L	0
	(3, 2)	N	0
	(3, 3)	P	0

TABLE 2.2: Ambisonics normalized 3D encoding functions for a ring source

layout, because encoding and decoding process are independent from each other. The only condition is that the number of loudspeakers L must be equal or greater than the number of Ambisonics channels K.

$$L \geq K$$

That is, for first order Ambisonics, at least four loudspeakers are necessary to achieve periphony (3 dimensions) and three for 2 dimensions (Z channel discarded). However, it is desirable that the number of loudspeakers is more than the number of Ambisonics channels in order to improve the accuracy of sound localization.

2.3.1 Physical decoding

Ambisonics decoding is based on the so called "re-encoding principle", which implies the reconstruction of the encoded components at the center O of the array [12]. These signals can be translated into pressure terms, $p_i = S_i$, which contribute to the total pressure (measured at the center of the system), $p = \sum p_i$. It is assumed that loudspeakers are far away from the center point, and the sound field can be obtained by adding coherently the contributions of these plane waves emitted by the loudspeakers. A linear combination of this components (superposition principle), which will feed the speakers, is found. This combination can be applied to the velocity vector, that is defined by Gerzon in the following form [13]

$$\tilde{V} = \frac{\sum_{i=1}^{N} G_i \vec{u}_i}{\sum_{i=1}^{N} G_i} = r_v \vec{u}_v \tag{2.9}$$

where G_i is the gain of the loudspeaker i and $\vec{u}_i$ is the unitary vector which represents the wave direction.

This principle gives a matrix, called "re-encoding matrix", which is used to get the recomposed pressure and velocity components. This signals are obtained as follows:

$$\tilde{B} = CS \tag{2.10}$$

where $\tilde{B} = [\tilde{B}_{00}\tilde{B}_{1-1}\tilde{B}_{10}...\tilde{B}_{mn}...]^T$ is the vector with the encoded Ambisonics channels, $S = [S_1 S_2 ... S_N]^T$ is the vector containing the signals and $C = [c_1 c_2 ... c_N]^T$ is the reencoding matrix, which elements are the associated spherical harmonics $Y_{mn}(\theta_S, \delta_S)$ (encoding gains) to the loudspeakers directions. This matrix has $K = (M+1)^2$ rows and as many columns as loudspeakers [11]:

$$c_i = \begin{bmatrix} Y_{00}(\theta_i, \delta_i) \\ Y_{1-1}(\theta_i, \delta_i) \\ Y_{10}(\theta_i, \delta_i) \\ ... \\ Y_{mn}(\theta_S, \delta_S) \\ ... \end{bmatrix} \tag{2.11}$$

Thus, in order to guarantee $\tilde{B} = B$, the decoding equation can be inverted as follows:

$$S = C^{-1}B, \tag{2.12}$$

where C^{-1} is the inverse of C. This ensures that the number of loudspeakers L is equal to the number of Ambisonics channels N, because the matrix can be inverted only if it is a square matrix.

Since the number of loudspeakers is usually higher than the number of channels, $N \geq L$, this inversion can be done using the pseudoinverse matrix of C, the so-called *"decoding matrix"*,

$$D = pinv(C) = C^T(CC^T)^{-1} \tag{2.13}$$

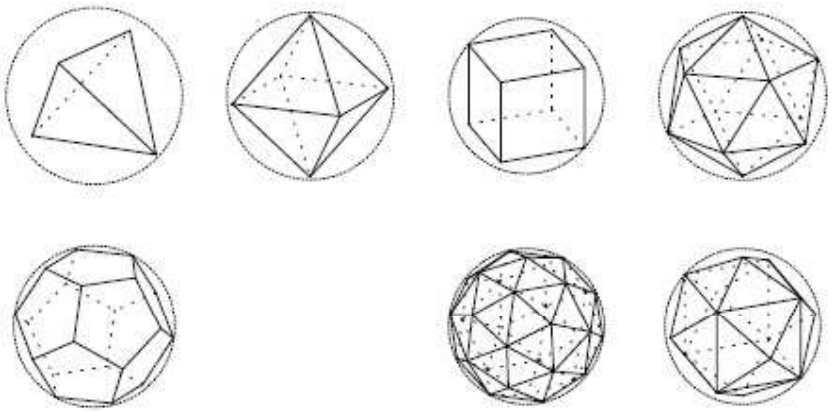


FIGURE 2.6: Regular polyhedron, called platonic solids. From top to down, the regular polyhedron: tetrahedron, octahedron, cube, icosahedron, dodecahedron. Then, two examples of semi-regular shapes, a product of a concatenation of dodecahedron/icosahedron, and cube/octahedron. Image extracted from [13]

Thus

$$S = DB, \tag{2.14}$$

For regular speakers layouts, which follow the shape of a *platonic solid* (2.6), the matrix CC^T is diagonal, and the equation is simplified [13] [14]

$$D = \frac{1}{N} C^T \tag{2.15}$$

The basic decoding has good results for low frequencies (below 700 Hz approximately) and for a local reconstruction (centered head). But for a higher frequency domain, where is assumed an incoherent sum of the speakers signals, a psychoacoustic criteria, introduced by Gerzon, is preferable.

2.3.2 Psychoacoustic decoding

Since it is assumed an incoherent sum of plane waves for medium and high frequencies (short wavelengths) and the center of the system is the measuring point, energy vector, defined as follows, seems to be the best localization predictor [13]

$$\vec{E} = \frac{\sum_{i=1}^{N} |G_i|^2 \vec{u}_i}{\sum_{i=1}^{N} |G_i|^2} \tag{2.16}$$

where G_i is the gain of the loudspeaker i and $\vec{u}_i$ is the unitary vector which defines the speaker wave direction. This "energy vector" is an estimator of the intensity vector under the incoherent summation hypothesis (which applies at high frequencies).

The decoding optimization can be achieved by simply applying gains to the frequency bands of the B_{mn} coefficients before the *basic decoding* process. A set of gains is applied to the "decoding matrix":

$$D = \frac{1}{N} C^T Diag([...g_m...]) \tag{2.17}$$

Thus,

$$S = \frac{1}{N} C^T Diag([...g_m...])B \tag{2.18}$$

The equation can be simplified, obtaining a equivalent ambisonics panning function:

$$f(\theta, M) = \frac{1}{n}(g_0 + 2\sum_{m=1}^{M} g_m \cos m\theta) \text{ (for 2D reproduction)} \tag{2.19}$$

$$f(\theta, M) = \frac{1}{n}\sum_{m=1}^{M}(2m+1) g_m P_m \cos\theta \text{ (for 3D reproduction)} \tag{2.20}$$

where θ is the angle between the loudspeaker and the sound direction. In regular layouts the Ambisonics decoding consist in generating virtual cardioid or hypercardioid microphones of order N pointing to the loudspeakers.

It is important to introduce the r_E index, the directionality coefficient of the energy, which characterizes:

- The accuracy of the sound image localization
- Image robustness (outside the sweet spot)
- Spatial quality (width of the point source)

The magnitude and direction of the energy vector r_E are defined as follows:

$$r_E \vec{u}_E = \frac{\sum_{i=1}^{N} |G_i|^2 \vec{u}_i}{\sum_{i=1}^{N} |G_i|^2} \tag{2.21}$$

For a regular layout:

$$r_E = \begin{cases} \dfrac{2M}{2M+1} & \text{2D} \\ \dfrac{M}{M+1} & \text{3D} \end{cases} \tag{2.22}$$

The optimal case is $r_E = 1$, associated to a fully localized sound source whereas $r_E = 0$ indicates a completely delocalized source. One can easily see that the r_E index converges towards 1 as the order goes to infinity.

max $\mathbf{r_E}$

In order to optimize the global reconstruction (high frequencies and non-centered head), $maxr_E$ decoding is based in focusing the energy signals in the $\vec{u}_S$ direction, this is, maximize of the energy module r_E. In this case, the r_E is computed by:

$$r_E = \begin{cases} \cos\dfrac{\pi}{2M+2} & \text{(2D)} \\ P_{M+1} & \text{(3D)} \end{cases} \tag{2.23}$$

where P_{M+1} is the biggest root of Legendre polynomial of M+1 degree.

And the g_m coefficients are computed by:

$$g_m = \begin{cases} \cos\dfrac{m\pi}{2M+2} & \text{(2D)} \\ P_m(r_E) & \text{(3D)} \end{cases} \tag{2.24}$$

In-phase

The loudspeakers' contributions interact with each other and sometimes the sound waves can be out of phase with respect to others. These out-of-phase speakers can be problematic in large rooms where the listeners are spread over a big area. This problem can be solved with the *in-phase* decoding, which implies the production of equally phased signals.

For the *in-phase* decoding algorithm, the gains applied to the B_{mn} components are:

$$g_m = \begin{cases} \dfrac{M!^2}{(M+m)!(M-m)!} & \text{(2D)} \\ \dfrac{M!(M+1)!}{(M+m+1)!(M-n)!} & \text{(3D)} \end{cases} \tag{2.25}$$

Non-regular layouts

Order	**Basic**	**max $\mathbf{r_E}$**	**In-phase**
1	0.5	0.577	0.5
2	0.667	0.775	0.667
3	0.75	0.861	0.75
4	0.8	0.906	0.8

TABLE 2.3: r_E index of the 3D decoding solutions (for regular configurations)

For non-regular layouts this process is more complex. Despite the basic decoding can be done by the pseudo-inverse method, the psychoacoustic decoding needs a non-linear search method.

2.3.3 Performance of decoders

The *basic* decoding method is the best choice for local reconstruction since it provides the best results for low frequencies at the sweet spot. However, at high frequencies the performance of this decoding algorithm decreases in comparison with the other two methods. *max* $\mathbf{r_E}$ provides the best results for high frequencies.

In phase decoding has a worst localization (less r_E) but it has a large sweet spot, so it is the best option for large audiences, where the listeners are outside from the sweet spot, or very close to the loudspeakers.

As mentioned above, the obtained decoding equations can be compared to the polar patterns in a microphone. A set of this virtual microphones associated to the decoding methods can be seen below:

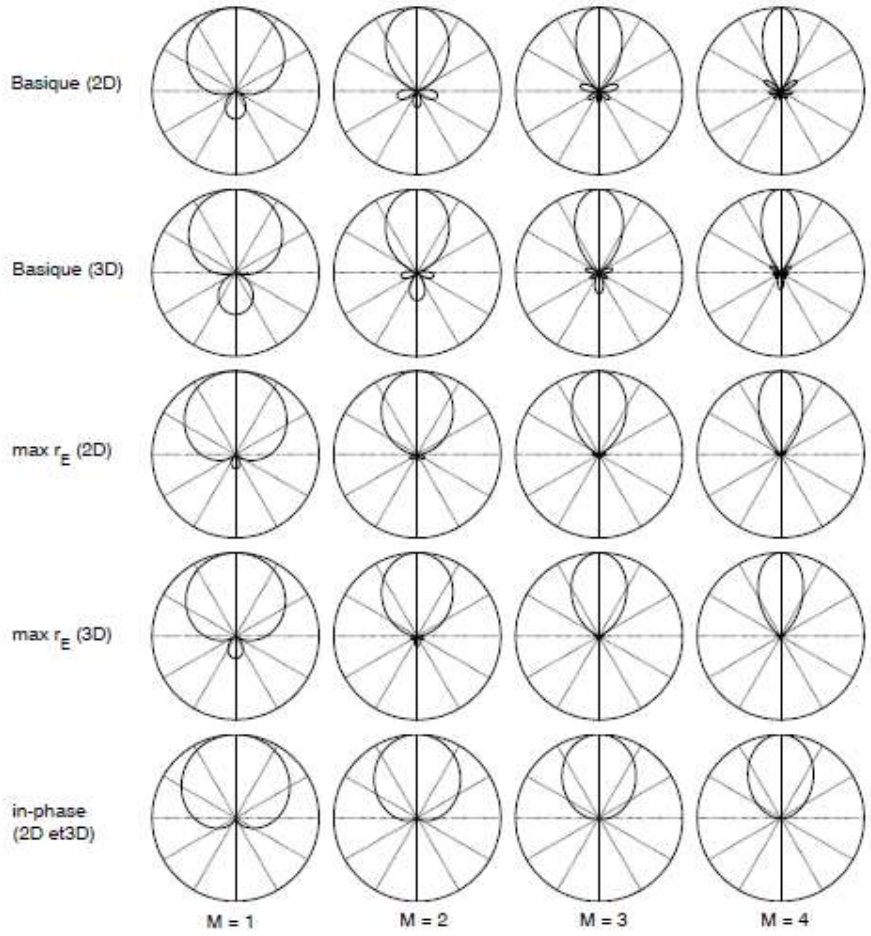


FIGURE 2.7: Directivity diagrams associated to the different decoding methods [13]

Chapter 3

Tool for encoding Ambisonics

3.1 Motivation and goals

The final goal is to obtain 3D compositions with as many sound sources as required, placed in all directions in space. For this reason, a real time encoder tool which allows to place and move sources in all the room will be programmed. The user will only need to load the mono sound as a point or *ring* source, and then he/she will be able to set the position of the source.

The aim is to get a program adaptable to the needs of the user, which can be modified and customized as required. This tool needs to be very intuitive and the interface needs to be easy to use with the possibility to see a graphical representation of the sound sources in the room. In addition, if the user wants to save the locations of the sounds for a future session, he/she will be able to do it by recording the data in text files.

3.2 Methodology

Pure Data is the tool chosen to implement the Ambisonics encoder. Pure Data (called PD from here on) is a visual programming language, created for multimedia purposes by Miller Puckette (the original version is called PD-Vanilla). A more developed version, PD-Extended, includes libraries and extensions and it is used in this project.

This open source software is based on *objects*, linked together like a diagram showing the sound/data flow, in a *patch*, an unit where the code is written. This patches can have subpatches inside them, with inputs and outputs.

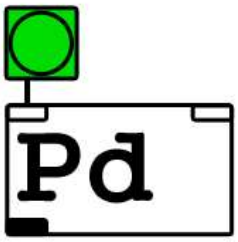


FIGURE 3.1: PD-extended

The tool programmed consist of one main patch (the interface) which contains different panels, one large horizontal placed on the top, which controls all the sources, and many vertical panels related with the different tracks. The default design contains six different tracks. One of the vertical panels, located at the bottom left, is the ring source track, another placed at the bottom right is the "external" composition track, and there are also many vertical panels related to the point sources.

Additional point (or ring) sources may be added, if necessary, by adding instances of *abstractions*, subpatches that can be reused, as will be shown. It has been programmed in this way in order to make the tool more adaptable since the program will respond to the needs of the user, who will be able to add or remove tracks as needed.

As can be seen, this tool only includes the encoding step, since the decoding process is done by the *Ambdec* program, an Ambisonics decoder developed by Fons Adriaensen. The irregular Ambisonics decoding coefficients are provided by Barcelona Media and obtained using the idhoa software.

3.3 Design

As said before, the interface is structured in panels. The first one, horizontal, controls all the tracks globally whereas each vertical section is related to one specific track and controls it locally.

The first panel, in the upper left corner, controls the window that shows the visualization of the sound scene. It contains two toggles, one for the window and another for showing the sources.

When the user checks the "On/Off" toggle, a window appears, showing the edges of a cube with one blue face, which represents the room and its floor, over a black background. The "Show all" toggle displays all sources inside the cube. Each point source is represented by a little sphere, which can be shown or not by checking the "Show" toggle

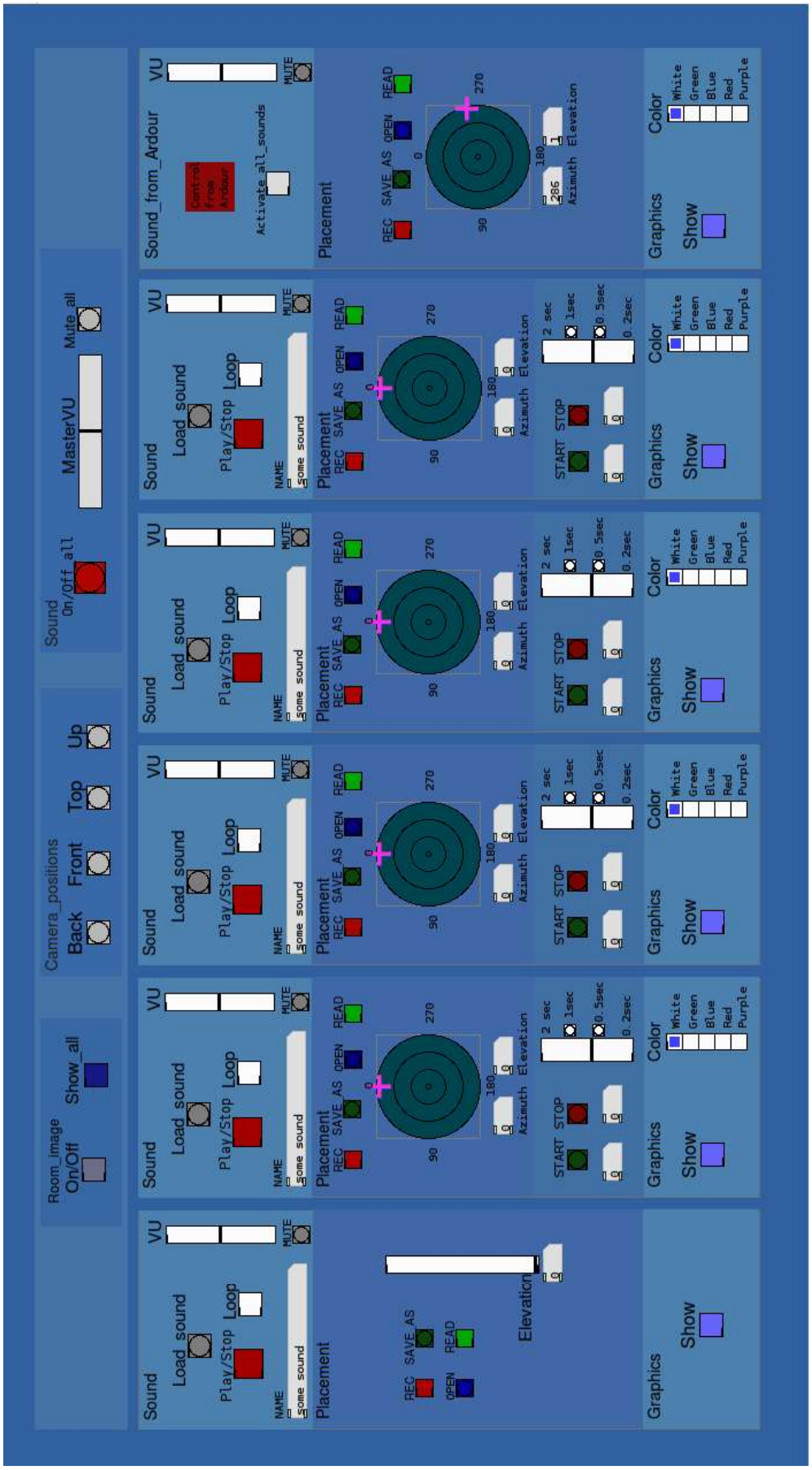


FIGURE 3.2: Interface with one ring source, one source taken from a digital audio workstation and four point sources

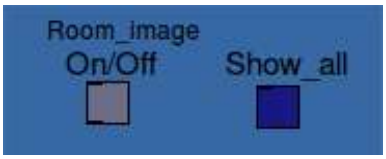


FIGURE 3.3: Panel which turns on/off the graphics

placed in its corresponding panel, and whose color can be changed by the user (between five different possibilities). The ring source is represented by a blue disk.

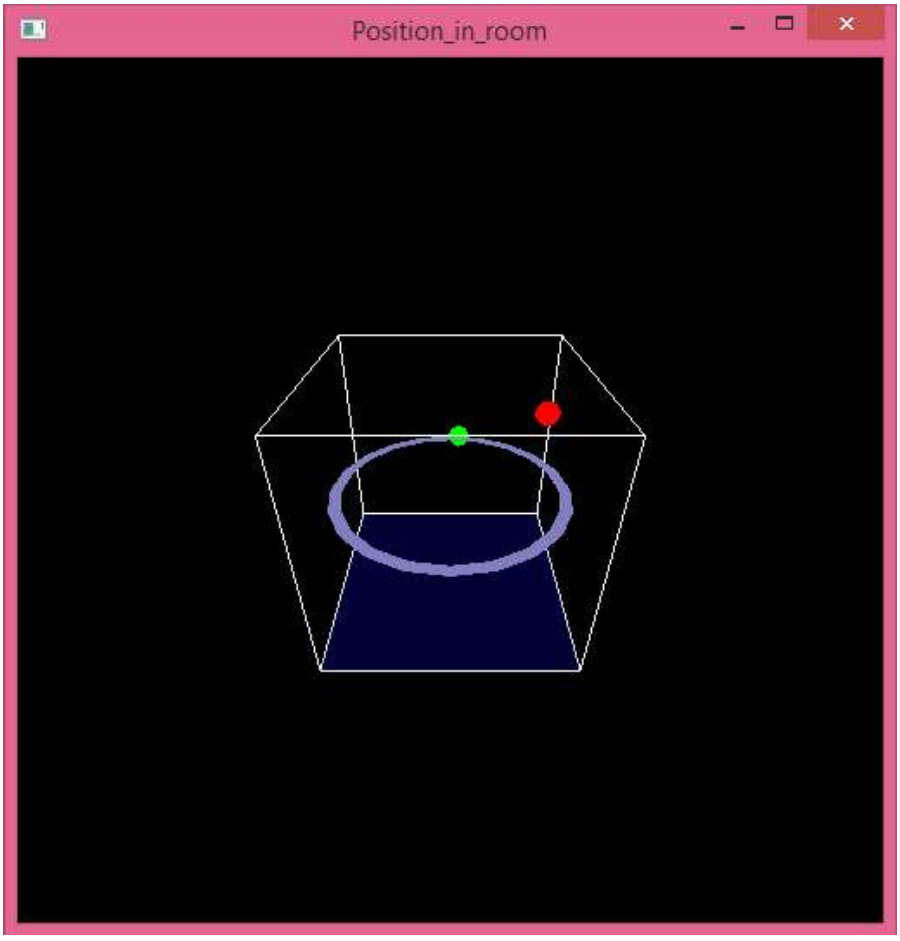


FIGURE 3.4: Representation of the room, two point sources and the ring source

Despite the default view is from the "top" when the window is opened, the room's view can be changed to top, front, back or top by clicking the buttons placed in the upper subsection with the words "Camera position".

The different views can be seen in the following image:

The third subsection controls the sound globally, i.e., the "On/Off all" button, turns all sounds and, if there are, the saved positions, on and off. It is also used if the goal is to

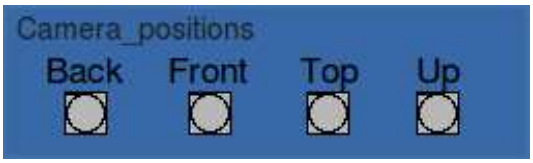


FIGURE 3.5: Buttons used to change the view

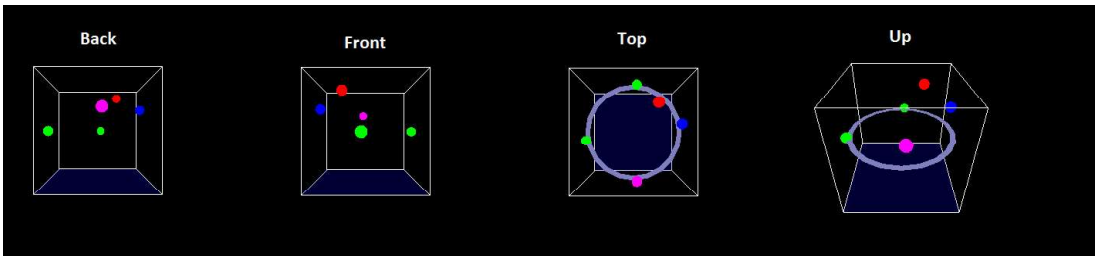


FIGURE 3.6: Different options for viewing the room: back, front, top, up

record the sound locations while all sounds are being reproduced, in which case it will be specified on the particular tracks.

FIGURE 3.7: Sound control panel. It contains the VU, a *mute* button, and the button for turn on/off all sounds

In the patch 3.2 the user has the possibility to place up to four point sources and one ring source. However, if more tracks are needed, the user only need to add more abstractions called `source`, and another sound may be used. This abstraction is a patch which can be used as many times as necessary and it will have the same characteristics every time it's applied.

Every track has its own buttons to manage the sound positions and other features. The buttons are arranged in panels subsections which are distinguished by their colors.

Point sources

In the first panel 3.8, there are four buttons and one vertical slide controlling the volume of the track. The large grey button is used, as said in its label, to load the sound. When it is pressed, a file browser appears on the screen and the user can choose the audio file, whose path will be appear in the box below. This path can be removed and changed to another name, if desired. The small grey button is used for muting the track and

the red one reproduces/stops the track. If the loop toggle is marked, the sound will be replayed until the user press the stop button.

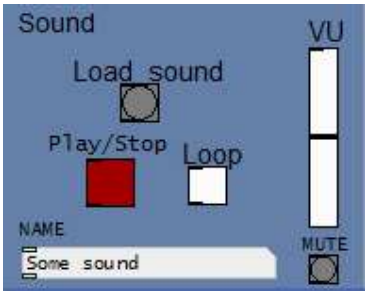


FIGURE 3.8: Sound buttons. This panel is the same for all sources

Panning point sources The second section includes the location features. For point sources, the [cube sphere] object is used due to its intuitive operability. This graphical object is a projection of the room in 2D and the sound position is represented by a purple cross 3.9. It can be moved by clicking on it or by changing the two numbers (corresponding to azimuth and elevation angles in degrees) below the object. The object can be thought of as a central dome placed above the listener's head, thus if the purple cross moves around the circle, the sound moves around the listener (the azimuth angle is modified) whereas if the movement is "into" the circle, following the radius, the sound goes up, (the center of the circumference is the maximum value reached in elevation, 90º, just above the head). Since the object returns two floats, and that does not make sense since the listener will not notice of such small changes, the floats are turn into integers and for simplicity, the azimuth angle is modified to be always positive, between 0 and 360º.

Special movements If the user wants to move the sound very fast (like jumping from one position in azimuth to another), he/she can use the buttons placed below the azimuth and elevation angles. The person only needs to set two angles and a number of seconds in the slider, which determines the velocity from one position to another, i.e. the sound will jump from one position to another every X seconds. Then, the user press the button "Start", and the positions will start to change.

Storing panning information If the user wants to store the positions used in one session, there is the opportunity of saving this information. As shown in the image 3.9, there are four buttons over the [cube sphere] object. The first one is used for recording the positions. The user must check the toggles of the tracks that wants to record and then he/she clicks on the "On/Off All" button for reproduce all the sounds. Only the

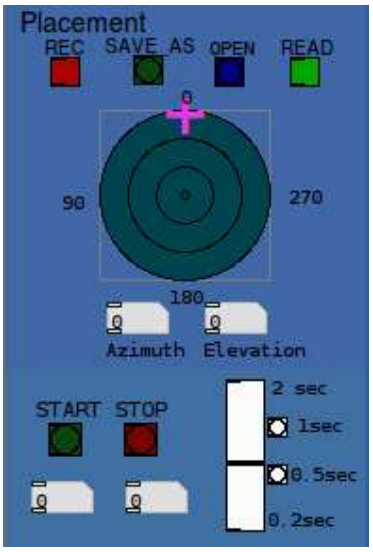


FIGURE 3.9: Placement part of the sound source panel

sound positions with the "REC" toggle checked are recorded. If the user is satisfied with the recording, then the person presses the button "SAVE AS" and a file-browser appears, where the name of the file must be written. If, on the other hand, the user wants to recover the positions from a previous session, then he/she has to click on the "OPEN" button. When it is pressed, a file-browser appears and the file can be chosen. After choosing the file, the button turns into green color automatically, indicating that there is one file charged, and the button "READ" is checked, which means that the file will be read when the sound is reproduced. Then, when the track is reproduced, in order to indicate that the file is being read, the "OPEN" button blinks. If the file is chosen but the user does not want to use it, the only thing that he/she has to do is deselect the "READ" toggle and the positions may be modified again. This buttons are arranged in such a way only one mode can be selected at time, i.e., if the *read* mode is chose, the *recording* mode can not be selected too, and viceversa.

Graphics The third section controls the graphics. As explained above, when the big toggle placed in the upper left is pressed, a window with a representation of the room appears. The point sources are represented as little spheres, whose color can be changed to white, blue, red, green or purple with the *Vradio* in the lower right corner of this section. The user can choose to show or not to show this sources by clicking on the "Show" toggle.

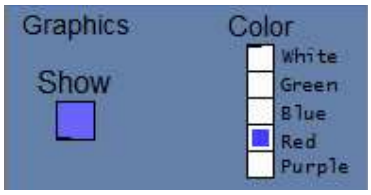


FIGURE 3.10: Point sources graphics section. The toggle turns on/off the point source graphics and the VRadio changes the colors of the spheres

Ring source

The ring source panels have almost the same options as the point sources. The difference lies in the room representation and the graphics color, which can not be changed (unnecessary because there is only one ring source).

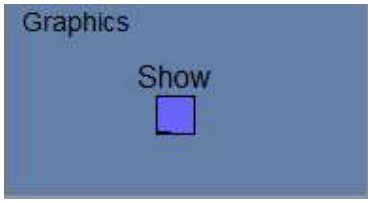


FIGURE 3.11: Ring source graphics panel. The toggle turns on/off the ring source graphics

Moreover, since the only information needed is the elevation, a slider is used instead the [cube sphere] object.

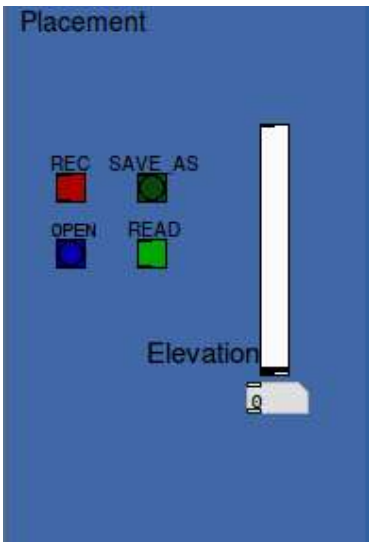


FIGURE 3.12: Ring source placement panel. The Vslide changes the elevation value and the buttons are used to read, save, record or read the positions of the source

3.4 Implementation

3.4.1 Sound control

All the processes needed to move the source, recording positions, control the sound or the graphics at a local level, are inside the sources abstractions. Most of the operations are inside subpatches in the abstractions, which allows an easier management of the code.

Inside the subpatch pd play/stop the sound is turn on/off, and the loop is applied if the toggle indicates that.

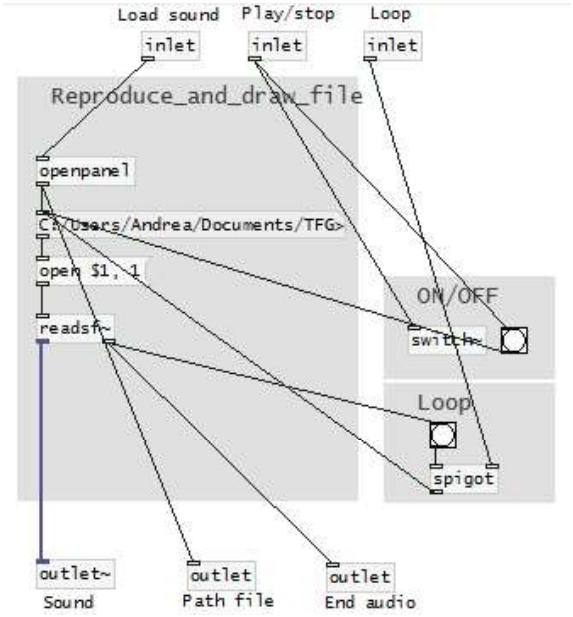


FIGURE 3.13: Subpatch where the sound file and loop are turn on/off

In order to have an easy-to-manage interface, some operations are done. The output of the subpatch pd onOffButtons determine if the toggles are checked or not, for example, when the sound file ends, the "On/Off" toggle is deactivated, so it is a good indicator for the user.

An overview of the subpatches used and operations done inside the abstraction can be seen in the Figure 3.15. For the ring source the same processes are done, with the difference that only elevation information is needed.

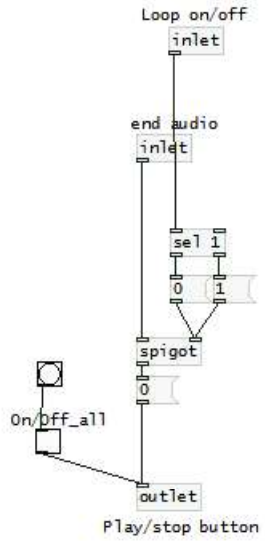


FIGURE 3.14: Subpatch which controls the checked toggles

3.4.2 Graphics

This graphics has been created with the help of the GEM(Graphics Environment for Multimedia) library. GEM, which is based on openGL, allows to create and manipulate 2D/3D graphics, process video and images. In this case, it has been used in order to display 3D images of the sound sources.

The user can control the positions with a slider for the ring source, and with the cube sphere for the point sources. The initial values for this object are inside the preferences cubeSphere, where the size, and colors are defined.

For the "jumping" utility, a simple metro object is used, where the user can change the speed.

The window and the room shape are created inside the abstraction placed in the left corner. When the "On/Off" toggle is checked, one "1" is sent to the [sel 1] object, which verifies if the input is a "1". If the input matches, a bang goes to one trigger, which outputs several bangs. One of these, arrives to the "create, 1" message, input of [gemwin] object, which allows to generate the window and starting the rendering process.

The other bangs go to the objects cube and square. The first one is used to get a cube shape whereas the second object creates a blue square (the alpha channel has been added for aesthetic purposes). Finally, the last bang is used to define the default view.

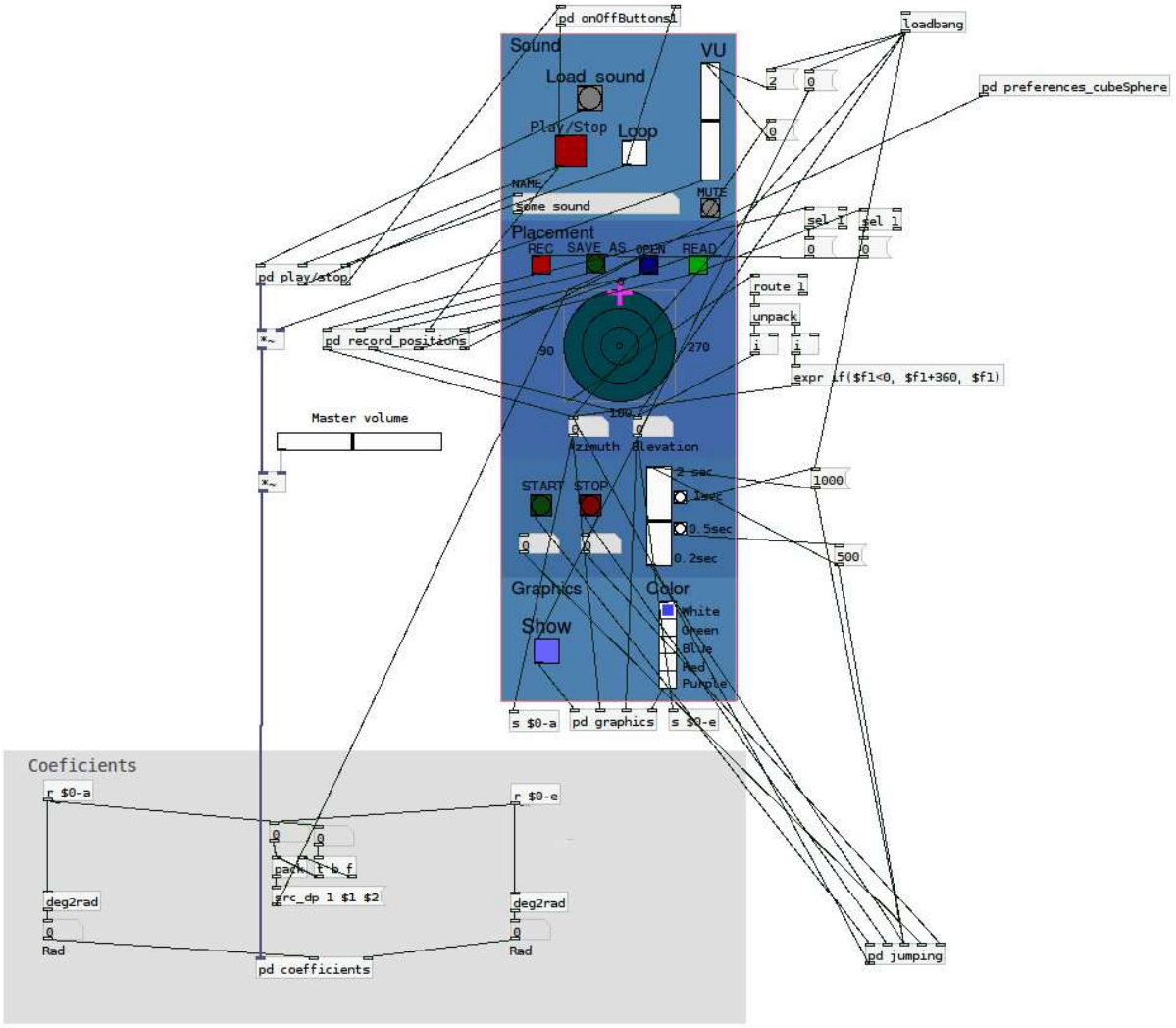


FIGURE 3.15: Point source abstraction

If the toggle is deactivated or the "ESC" key is pressed, the window closes (a "0, destroy" message is sent, which means, stop rendering, close the window).

The "Show all" toggle send a message (1 or 0) to all the "Show" toggles in the tracks, so if the button is checked, all sources will be displayed on the screen.

The different views are possible by changing the viewpoint, i.e., translating the camera position. Depending on which button is pressed, a bang is sent to a message, which arrives to the gemwin object. This message contains 9 arguments related to the position of the camera (x, y, z).

At a local level, the graphics are managed in the object pd graphics, which is inside each abstraction. For point sources, the object has four inlets, two for the azimuth and

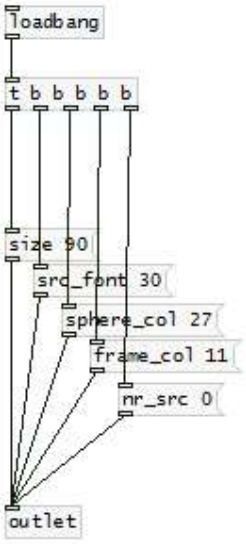


FIGURE 3.16: Init values for the cube sphere object

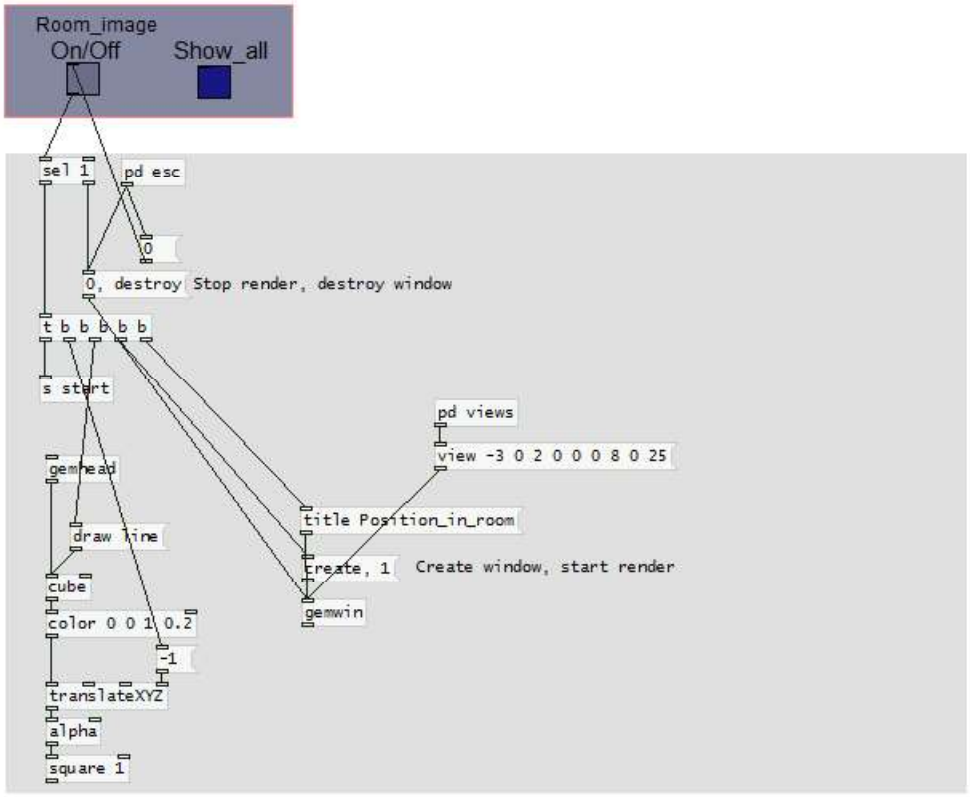


FIGURE 3.17: Graphics abstraction. In this patch the window containing the graphics is created, as well as the empty "room"

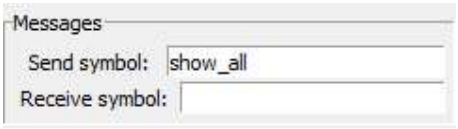


FIGURE 3.18: "Send all" toggle properties

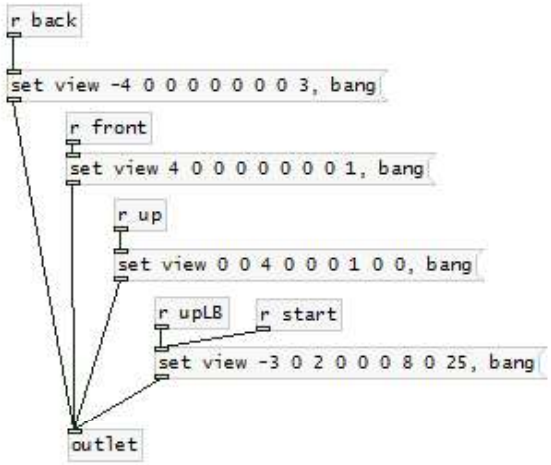


FIGURE 3.19: Messages sent to the gemwin object in order to change the viewpoint

elevation angles, one for choosing the color, and the other one is for turn on/off the graphics.

The user can choose the color of the source, by clicking the buttons in the VRadio object placed in the "Graphics" panel. This object sends a message with a different number depending on the button pressed. If the first button in the array is pressed, a message with one "0" arrives to the select object, which sends a "bang" to the first outlet, which is connected to a message containing three RGB float values (for example 1,1,1 for white and 0,0,1 for red). These numbers are sent to the color object, which sets the color of the sphere.

In order to show the movements of the source, some calculations are performed. The sphere can be moved with the object translateXYZ, whose inputs are the three Cartesian coordinates. As the user is working in spherical coordinates, the values must be converted into Cartesian coordinates before sending them to the object. This calculation is done as follows:

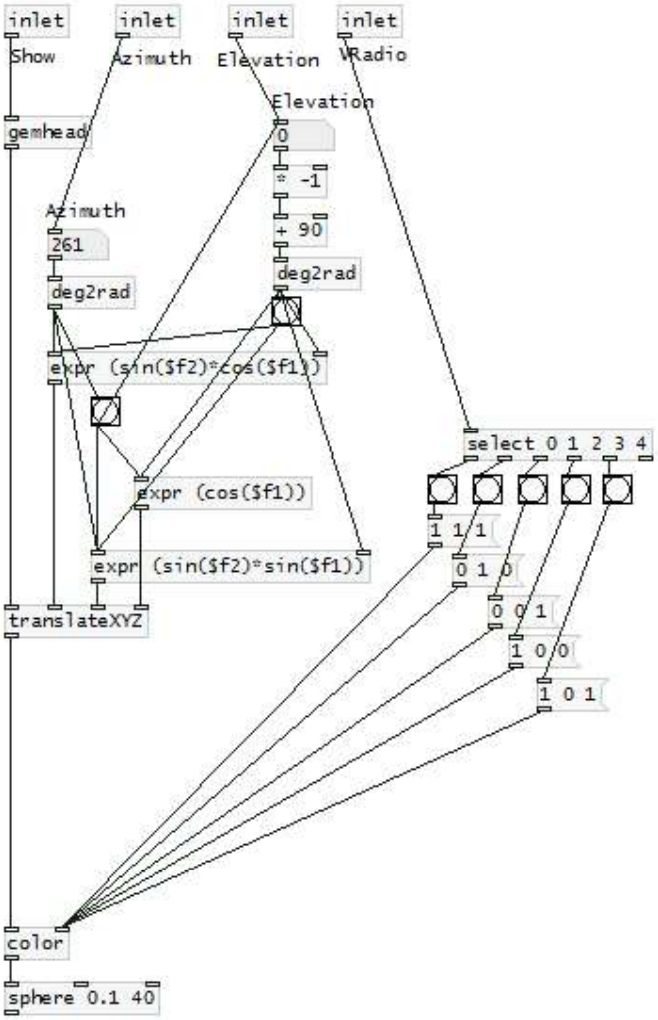


FIGURE 3.20: PD graphics. This subpatch contains all objects and calculations to show the sources

$$x = \cos\theta\cos\phi$$
$$y = \sin\theta\cos\phi$$
$$z = \sin\phi$$

The two angles are converted from degrees to radians, and the latitude values are modified to be measured in the reverse order, (from 90 to 0 and not from 0 to 90) since the angle is measured between the axis y and the sound position.

The translation and the color change are sent to the sphere 0.1 40 object, which has a size of 0.1 and it is divided into 40 segments. With this process, a little sphere related

to the position of the sound source is created.

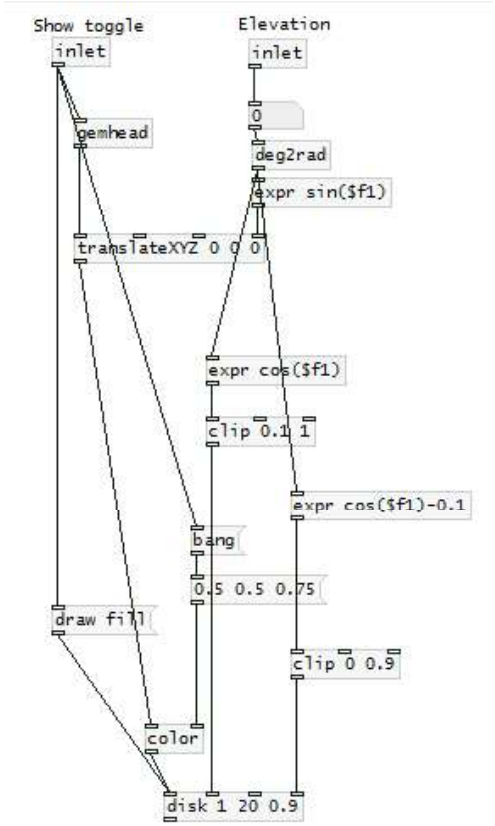


FIGURE 3.21: PD graphics for a ring source

For ring sources, the calculations are different. In order to represent it, the disk object is used, and it is moved along the z-axis depending on the elevation value given by the user. Like in the point sources, the angles must be converted into radians before sending them to the translateXYZ object (in this case only the z argument is modified). Then, three arguments must be arrive to the disk object: outer radius, number of segments, and hole size or inner radius. As the disc moves, these values should be modified accordingly. Therefore, the disk size (outer radius) is the *cos* of the elevation value, and the number of segments remains static.

3.4.3 Coefficients

As seen above, the ambisonics components are obtained as a result of multiplying the signal by the angular-dependent coefficients. These operations are inside the pd coefficients subpatch.

Point sources

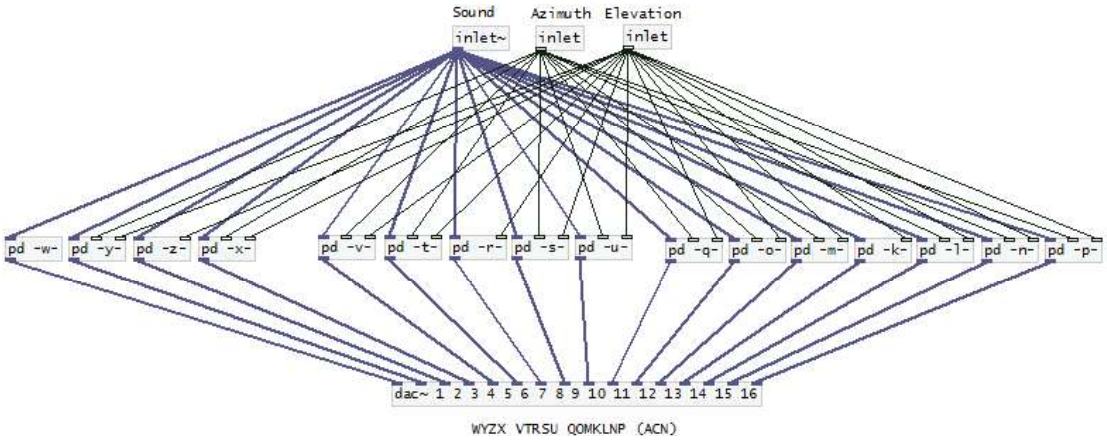


FIGURE 3.22: pd Coefficients subpatch. As shown, the channels follow the ACN order

For point sources, it is necessary to have the signal and two angles (azimuth and elevation), thus the subpatch has three inputs. Inside this subpatch, each coefficient is computed separatly. The angles are converted to radians before computing the coefficients.

This process is also performed by the "jumping" angles, which change due to the calculations done inside the pd jumping subpatch, which contains a metro with a variable speed.

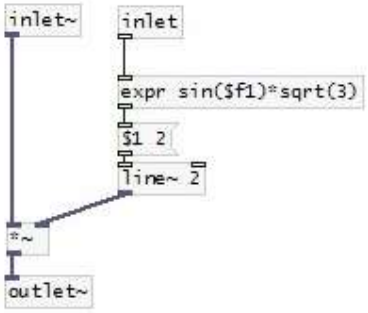


FIGURE 3.23: Z coefficient. To compute Z, it is only necessary to have two inputs: signal and elevation

A [line] object is added in order to remove clicks in the sound due to abrupt changes in the coefficients (if it moves too fast).

Ring source

In this case, the channels are computed in the same way as with point sources, but the calculation is simpler, since only information about elevation is needed.

3.4.4 Positions recording

If the user wants to automate the locations of sounds, the only thing he/she has to do is check the "REC" toggle and then, with the "Save as" button, store them under a name. For reading the positions from another session, the user must open the desired files and then, this information will be used for placing the sounds.

All the objects and procedures used to store and read the positions are included inside the `pd record_positions` subpatch. It has five inlets, four are the buttons which control the files ("Open", "Save as", "Rec" and "Read" buttons) and the other one is the "Play/Stop" toggle which turn on/off the sound of the track. This patch generates four outlets, two of them change the values of the angles when the file is being read, another changes the color of the "Open" button, and the other controls the "Read" toggle.

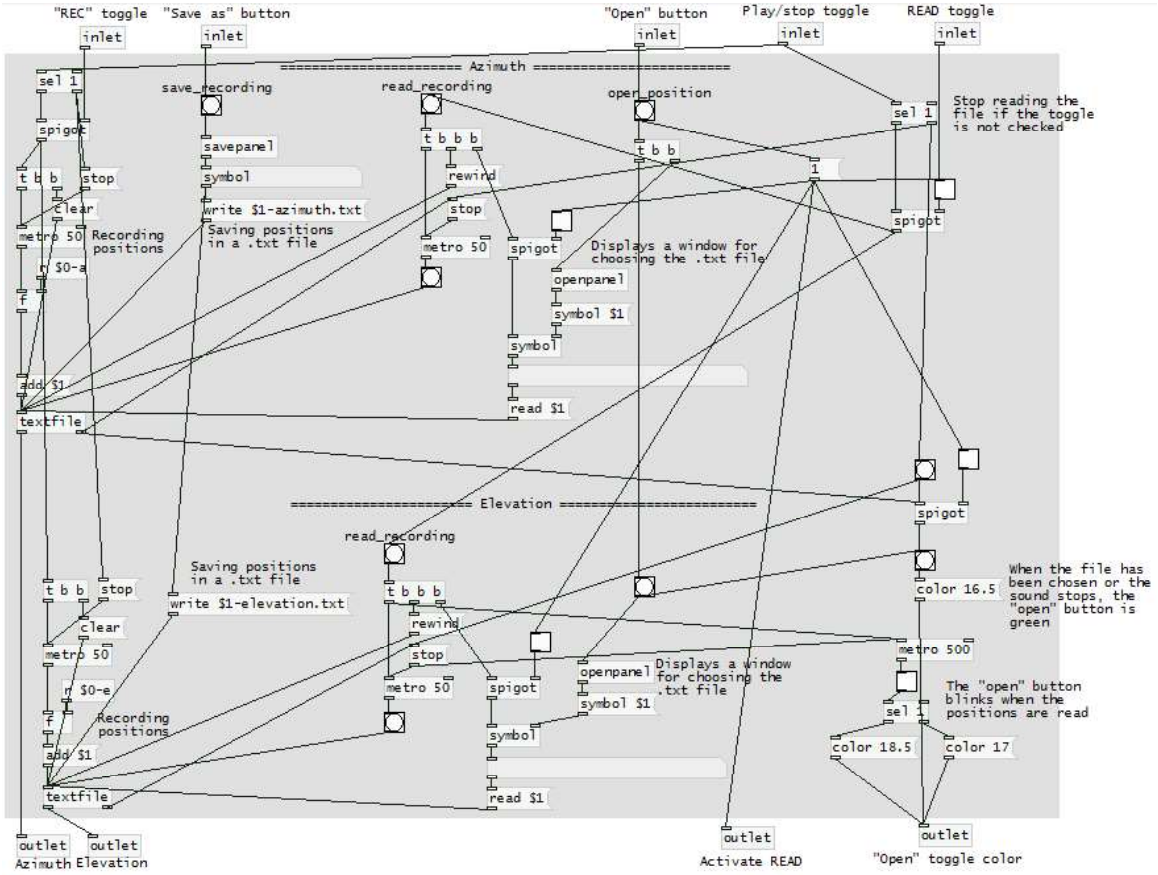


FIGURE 3.24: Subpatch where the steps for recording, saving and reading the source positions take place

The elevation and azimuth values are written in different .txt files by using the textfile object. To record them, a "clear" message must be sent first, in order to delete all inside the object, and then, an "add $ 1" message is sent, where the $1 argument corresponds to the angle value . This number, sent in the main patch, arrives the object r $0 -a, or r $0 -e for the elevation, and is sent to the left input of the object f, which store the number inside, until a bang outputs it. This bang is generated by the metro 50 object, which sends a "bang" each 50 ms. Therefore, the angles values are recorded each 50 ms (it has been thought that the movements done by the user are not faster than 50 ms).

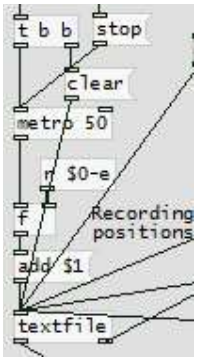


FIGURE 3.25: Detail of the subpatch for recording the elevation values

The values are stored inside the textfile object, but it needs one message to create the file. It is useful if the user is not sure if he/she is satisfied with the recorded angles and wants to reproduce them before saving them. If the user is convinced that the result is correct, then he/she can store the positions by pressing the button "Save as", and a "bang" is sent to the savepanel object, which allows to display a window where the user is able to type the name, without any extension. Automatically, the extension .txt is added to the name written by the user, and one file is created by sending the "write $1-azimuth.txt" message ("write $1-elevation.txt"), which generates a file with the name given by the user and a specification of the angle (for example, if the user types session1, the files will be "session1-azimuth.txt" for the azimuth values and "session1-elevation.txt" for the elevation angles).

When the "Open" button is pressed, two windows are displayed in order to choose the file for azimuth and elevation (the user must choose the same file two times, one for azimuth and other for elevation). When the "Read" toggle is pressed, the file name is sent to the "read $1" message, whose argument is the name of the file. It is necessary to send a "rewind" message before reading it, in order to start from the first number in the file, and a metro 50 object determines the velocity at the file is read. Then, the values are output by the textfile object and sent to the outlets. This outlets are connected to the input of the cube Sphere object, which means that the position of the source will be changed in the graphic.

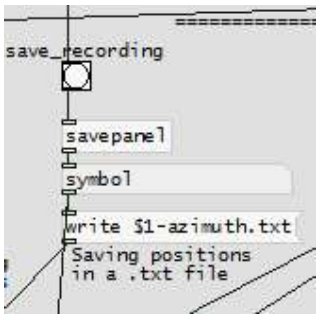


FIGURE 3.26: When the user clicks on the "Save as" button, a bang is sent and a window appears, then the user types a name, a extension is added to the name, and the file is created by sending the message to the textfile object

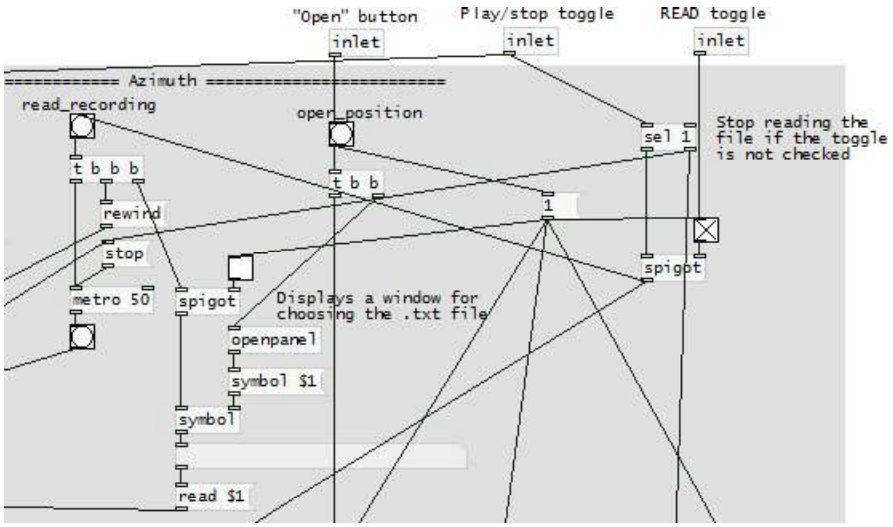


FIGURE 3.27: When the user clicks on the "Open" button, two windows are displayed, where the user can choose a file. Then, if the "Read" toggle is checked, the file will be read

In addition, when the "Open" button is checked, its color changes from blue to green, and when the file is being read, the button blinks from light blue to bright blue. This blink is made with a metro 500 object, which changes the color each half second.

The "Read" toggle is checked when a file is opened, and it decides if the file is read or not. In addition, if the "Play/Stop" button turns off, the reading process stops.

For the ring source, only one file must be saved, whose name is given by the user, plus the "-rings.txt" extension.

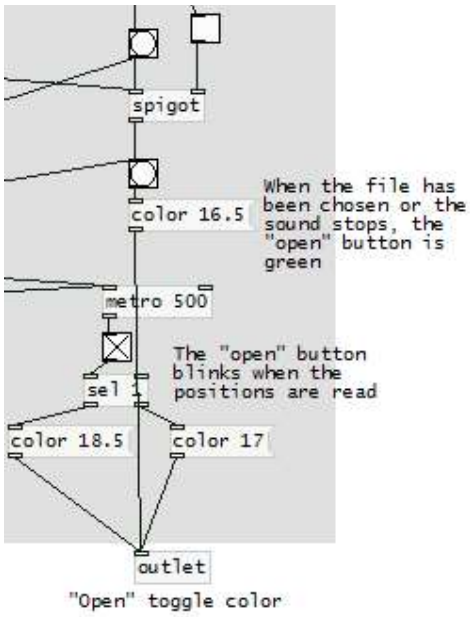


FIGURE 3.28: The "Open" button has four different colors depending if the file has been chosen, if there is not any file chosen or if the file is being read

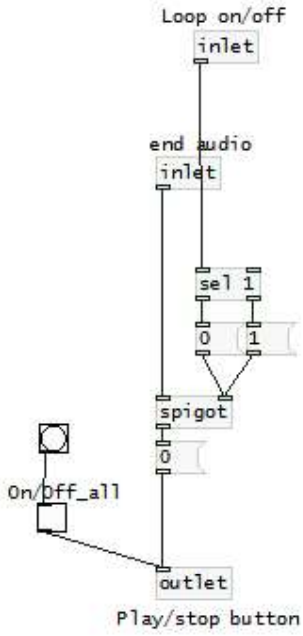


FIGURE 3.29: The "Open" button has four different colors depending if the file has been chosen, if there is not any file chosen or if the file is being read

3.5 Connecting an external Digital Audio Workstation with Pure Data

There is one more utility that allows to connect an external Digital Audio Workstation with PD, as if the composition made in the DAW were a point source. In this case, the user has to make some modifications. He/she will need:

- A DAW
- PD-extended
- JACK
- a2jmidid
- Adding one source ardour abstraction to the main program

The Digital Audio Workstation used in this report is called Ardour, which is free software and allows to record, edit, mix and master sound.

In order to conect the DAW with PD, Ardour must send a MIDI signal when the track starts in order to synchronize the two programs. This can be done by checking the "Send MIDI clock" button in the *Preferences* panel (only in Ardour3). Pure Data receives this signal if the appropriate ports are connected by JACK using a2jmidid, which creates a bridge between the ALSA-MIDI ports and the JACK ports. Finally, the sound output of Ardour must be the audio input of PD-extended (this connection is done in JACK). The MIDI clock signal determines the time when the track starts. The source can be placed and moved as the others point sources. The only difference is the first section of the panel, since it has no "Start/Stop" button. When the user wants to reproduce the file, he/she has to go to Ardour and click the play button. The audio file will be streamed to PD, where it will be processed in order to be encoded in third order Ambisonics, adding the spatial information. If the user wants to synchronize all files, he/she has to check the "Activate all sounds" toggle and then play the sound from Ardour. This action will activate all the tracks in the PD composition.

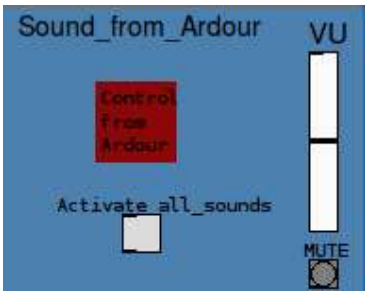


FIGURE 3.30: Detail of the sound control section in the Ardour source

Chapter 4

Practical/artistic composition

In order to show a demonstration of the operation of the tool programmed, some compositions have been done. These pieces have the objective to exploit the potential of spatial audio, the first one by creating a soundscape whereas the second one is a music example.

In the first composition the listener is inside a room with really noisy neighbors. First, one neighbor is doing some construction works and the machines sound very loud. This event awakes a baby who was sleeping in the apartment next door and he starts to cry. The mother comes to try to comfort her baby. In other flat, these sounds fright a dog which starts to bark and move in the flat of its owner. In each adjoining flat some event happens, producing a particular soundscape. As can be seen, it is a common situation where the sounds come from many different places.

The sounds have been downloaded from freesound.org, processed and mixed with Audacity and Ardour. A low-pass filter has been applied to the audio files in order to get a more realistic sound, as if the sources were on the other side of the wall.

The second composition is a simple piece of music, composed in C Major, for a string orchestra, piano, guitar, harp and flute. As said before, the principal goal is to get a composition which fully exploits the capacities of the spatial audio and takes advantage of the tool potential. Despite it is a music composition, there has not been an emphasis on getting a perfect musical structure, but it has been more important to place and move the sounds. The musical composition has been created in Cubase (a DAW developed by Steinberg company) with the Cinematic string library (for strings), Berlin Woodwings Exp. B (flute), Imperfect samples Fazioli Ebony (piano), Cinesamples CinePerc core (timpani and cymbals) and Cinesamples CinePerc Pro (maraca) libraries.

The result obtained has been a composition, with many sources placed around the room and creating a particular soundscape as well as one musical piece whose source positions move following the music, surprising the listener.

Chapter 5

Conclusions

After the completion of all phases of development of this project, I think I have met most objectives set at the beginning. The HOA basis has been understood, and an application of this technique has been developed and also used at an user level, since a composition has been done by using the programmed tool.

To start, the operation and the principles on which High Order Ambisonics technique relies have been explained. In the first chapter we have seen what the spatial audio is, and the cues used by the humans to localize sound in space. These cues are important because they justify the basis of the spatial audio techniques, which simply try to create the cues in the listener's head (discrete panning techniques, binaural technique) or aim to recreate the physical field of the original scene (WFS and Ambisonics). In this phase of the project I have been able to understand in depth spatial audio technique and in particular HOA theory.

I have dealt with the encoding step by programming a tool which allows to place and move sounds inside a room. This programming process helped me to have a structured vision of the Ambisonics system, in addition to learn more about PD and the Linux operation. I have become familiar with LaTeX and with the development tools of audio in open environments, such as Ardour and PD. I have improved my PD skills, since all the tool has been programmed almost from scratch and I had to solve problems I didn't faced before. Creating the graphics was hard work for me because I had never worked before with the GEM library, but I consider that it was a great opportunity to learn its usage and finally I obtained the desired results.

The goal was to obtain a tool easy to use and adaptable to the needs of the user, and this objective has been achieved since the interface can be modified as required (within the limits of the tool, only point and ring sources). The tool can be improved in the future

by adding more features, like graphical utilities such as a timeline, or the possibility to move many sources at the same time. In addition, it would be very useful if the user could move in the timeline without the need to listen all the tracks from the beginning.

The result obtained (in this case, a sound composition) assures the good performance of the tool. The development of this piece has been an opportunity to work also with the "artistic" part of this system but also a challenge since it is not a common traditional stereo composition.

Bibliography

[1] Gary S. Kendall. A 3d sound primer: Directional hearing and stereo reproduction. *Computer Music Journal*, 19(4):23–46, Winter 1995.

[2] Bruce Bartlett with Jenny Bartlett. *On-Location Recording Techniques*. Focal Press, 1999. ISBN 0240803795.

[3] Michael Gerzon. Surround-sound psychoacoustics. criteria for the design of matrix and discrete surround-sound systems. *Wireless World*, December 1974.

[4] Francis Rumsey. *Spatial Audio*. Focal Press, 2001.

[5] Russell Burns. Blumlein and the birth of stereo. *IEE review*, November 1999.

[6] Radiocommunication Sector of ITU. Multichannel stereophonic sound system with and without accompanying picture. Technical report, 2012.

[7] Larcher Veronique Jot, Jean-Marc¡ and Jean-Marie Pernaux. A comparative study of 3-d audio encoding and rendering techniques. In *Audio Engineering Society Conference: 16th International Conference: Spatial Sound Reproduction*, Mar 1999.

[8] Ville Pulkki. Spatial sound generation and perception by amplitude panning techniques. Technical report, Helsinki University of Technology Laboratory of Acoustics and Audio Signal Processing, Helsinki, 2001.

[9] G A Delft. Wave field synthesis and analysis using array technology. In *Proc. 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, volume Vi, pages 15–18, 1999.

[10] K. Brandenburg, S. Brix, and T. Sporer. Wave field synthesis. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2009*, pages 1–4, May 2009.

[11] Jérôme Daniel. Spatial sound encoding including near field effect: Introducing distance coding filters and a viable, new ambisonic format. *AES 23rd International Conference*, 23, 2003.

[12] Jérôme Daniel, Sebastien Moreau, and Rozenn Nicol. Further investigations of high-order ambisonics and wavefield synthesis for holophonic sound imaging. In *Audio Engineering Society Convention 114*, Mar 2003. URL http://www.aes.org/e-lib/browse.cfm?elib=12567.

[13] Daniel Jérôme. *Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia*. PhD thesis, Université Paris 6, Paris, 2001.

[14] Martin Neukom and Jan C Schacher. Ambisonics equivalent panning. Technical report, Zurich University of the Arts (Institute for Computer Music and Sound Technology), Zürich.