Oleg Loshkin

Student # 12-13427 / loshkin.oleg.95@gmail.com

BACHELOR OF SCIENCE

Games Programming

Measuring the impact of real-time binaural sound spatialization on sound localization.

BACHELOR THESIS - MODULE CMN 6302.1

Farhan Elias, Siorak Nicolas

July 31, 2022

SAE INSTITUTE - GENEVA

Words : 13’884

I, Oleg Loshkin, certify having personally written this Bachelor thesis.

I also certify that I have not resorted to plagiarism and have conscientiously and clearly mention all borrowings from others.

Oleg Loshkin

Geneva, 31 July 2022

Summary

With the popularization of binaural audio, some video-games now provide binaural sound spatialization with claims of better sound localization.

An experiment has been conducted as part of this work to measure the variation of sound localization accuracy between a classic approach to sound spatialization and a binaural approach.

The results suggest that the use of binaural audio significantly increases localization accuracy on the azimuth but surprisingly does not improve accuracy in elevation.

The experiment needs significant improvements before the current results can be deemed reliable.

The results of the experiment also suggest that reverberations have a very significant impact on sound source localization but the small number of data samples makes this assertion uncertain.

Acknowledgements

This work has been made possible thanks to these people:

- Timothée Crettaz, a fellow audio engineering student who spiked my interest in game audio and has been a great help in introducing me to the world of audio.
- Dr. Hasan Baran Firat, a professional acoustician who has dedicated some of his time to answer my questions regarding the prospective of binaural audio as well as given a few advices for the practical experiment of this work.
- Nicolas Siorak, a great teacher and mentor whose help and enthusiasm have doubtlessly resulted in the betterment of this work.
- Mike Manco, a school assistant who has provided helpful advice for the formatting of this document.
- Elias Farhan, a very skilled teacher whose emphasis on curiosity has led me to this bachelor subject.

Although not necessarily mentioned explicitly, the following software was used to create the software used for this work's experiment:

- 3DTI Audio Toolkit
- CMake 3.23.2
- FMod
- Git and Github
- Google Docs
- Google Drive
- Google Sheets
- OpenVR

- PortAudio
- Python 3.10 with the following packages:
    - matplotlib
    - numpy
    - pandas
    - pysofaconventions
- SDL2
- Visual Studio 2019
- cereal
- dr_wav
- easy_profiler
- libsofa
- spdlog

1. Introduction

1.1 Preface

Hearing is one of our main five senses and as such informs us greatly about our environment. While we might think of sight as our most important sense, the ears tell the eyes where to look: these two senses are closely linked together.

While sight gives us the ability to perceive with great precision things that are in front of us, hearing gives us a less precise but more global description of our complete environment (Letowski & Letowski, 2011), regardless of the orientation of our head. Some research goes further into the interaction between sight and hearing and it is even a domain that might benefit greatly from advances in machine learning (Parida et al., 2021). Although we might seldom think about the complexity of our hearing, those who are forced to rely on it can testify of its sophistication: a 2012 study lead by Santani Teng for instance has studied echolocation in blind humans, finding it to be in some cases to be outperforming that of bats (Teng et al., 2012)!

In video-games today, visual elements seem to gather more attention and investment than audio however: Nvidia's push for raytracing (*RTX. It's On: Ultimate Ray Tracing & AI | NVIDIA*, 2022), Unreal Engine's push for Nanite (*Nanite Virtualized Geometry in Unreal Engine | Unreal Engine Documentation*, 2022), the race to higher refresh rates of monitors (Agomuoh, 2022), much of technical progress in the video-game industry seems to revolve around graphics.

There is no lack of potential for advances in real-time sound processing however. The use of stereo still completely overshadows the use of binaural audio outside of VR games, despite its potential to improve the listener’s experience, especially for competitive games (BLAST Premier, 2021). The real-time simulation of reverberations of arbitrary environments is still barely ever used with only a few commercial software solutions starting to become available such as Steam Audio (*Steam Audio*, 2022), which as impressive as it already is, still does leave room for improvements regarding leveraging the existing Nvidia hardware.

Most software still uses proprietary HRTF file formats but a common standard, the SOFA file format, is thankfully slowly gaining popularity (Majdak, 2013).

Generic HRTF files are still however very predominantly used as current HRTF individualization remains a complicated procedure (Cuevas-Rodríguez et al., 2019). There is still no lack of potential improvements in the domain of real-time sound simulation.

1.2 Objectives

The aim of this thesis is to establish a table to compare sound spatialization quality of two commercially available spatializers for video-games solutions publicly available today. The table is composed of the spatialization method used as column axis and mean errors on the row axis.

This table is created from measurements done during a dedicated experiment using the controllers of a VR headset.

1.3 Overview of the document

This document is split into three main sections.

Section 2. introduces the reader to the concepts necessary to understand to make sense of the results of the practical experiment conducted as part of this work and to understand the implications of these results.

Sections 3. describes the practical experiment and section 4. presents and analyses its results.

2. State-of-the-art

2.1 The physics of sound

From a physical perspective, sound is a wave of kinetic energy traveling through a medium. While sound isn’t limited to air (you can still hear sounds underwater for instance), for the purposes of this work we’ll limit ourselves to sound traveling through air and only briefly mention the phenomena of sound transmission.

Firstly, let us go over some basic concepts of sound. Periodic sounds, meaning sounds that are composed of a repeating variation in air pressure, can be described with a few parameters.

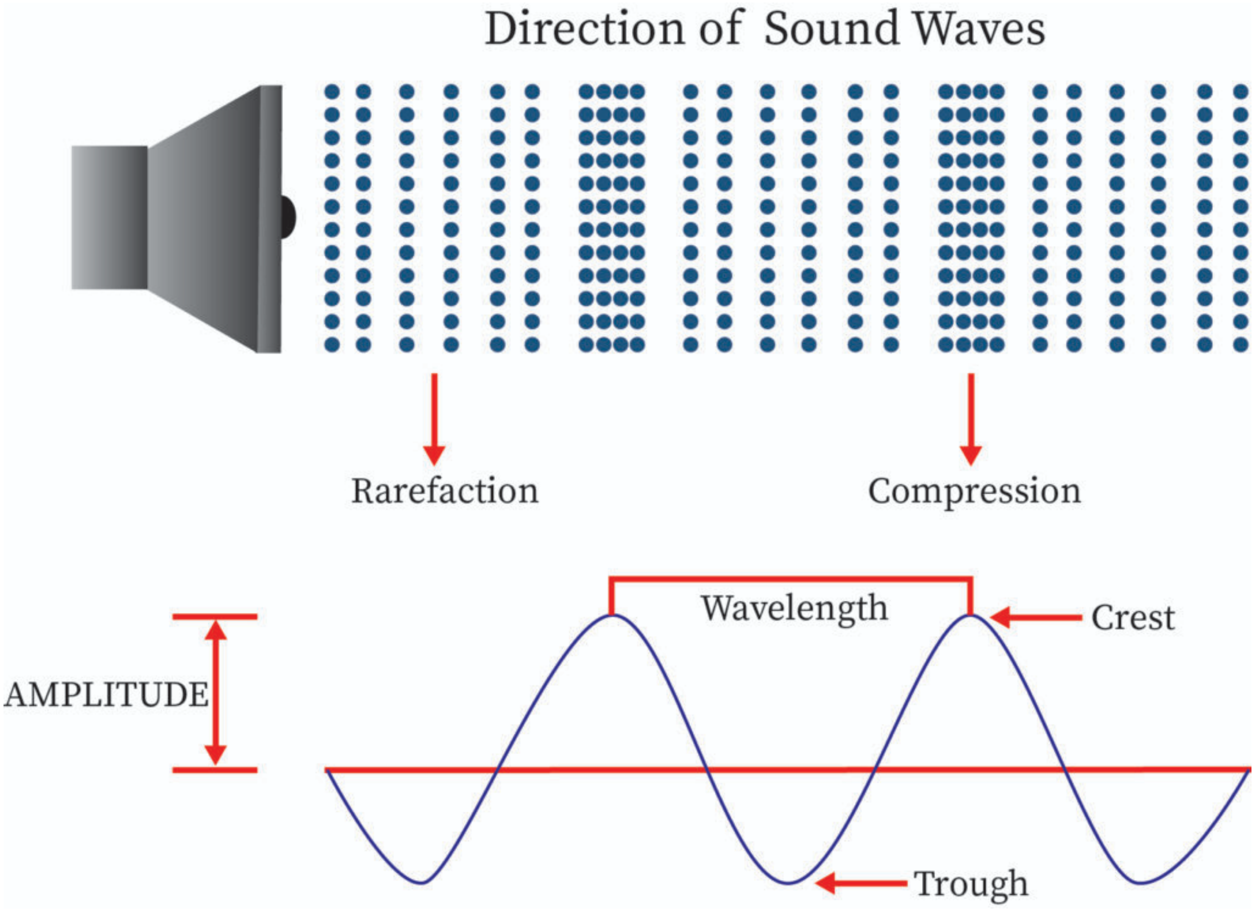


Fig.1: Evolution of sound pressure at a fixed point in space as sound travels through it.

A periodic signal has an “amplitude”: the highest amount of atmospheric pressure variation occurring as the wave propagates. A sound’s “frequency” (often referred to as its “pitch” when outside of the mathematical context) defines how many times a second the pressure oscillates between low and high pressure. The *phase* of a periodic sound signal defines how much it is “offset” on the time axis, the cosine function being equal to a sine function offset by Π on the time axis for instance.

In theory, by adding periodic signals together (also referred to as “pure tones”) one can create any sound imaginable using a “Fourier series” (‘Fourier Series’, 2022). This fact is the basis of most modern digital signal processing techniques.

It is important to point out that while in electronics we usually deal with continuous signals, ones that cannot be accurately represented by a finite amount of points on a graph, in digital signal processing, signals are discrete instead, meaning they are represented as “impulse trains”, a series of theoretically instantaneous continuous signals that have a non-zero value at very specific times (Gregory, 2014).

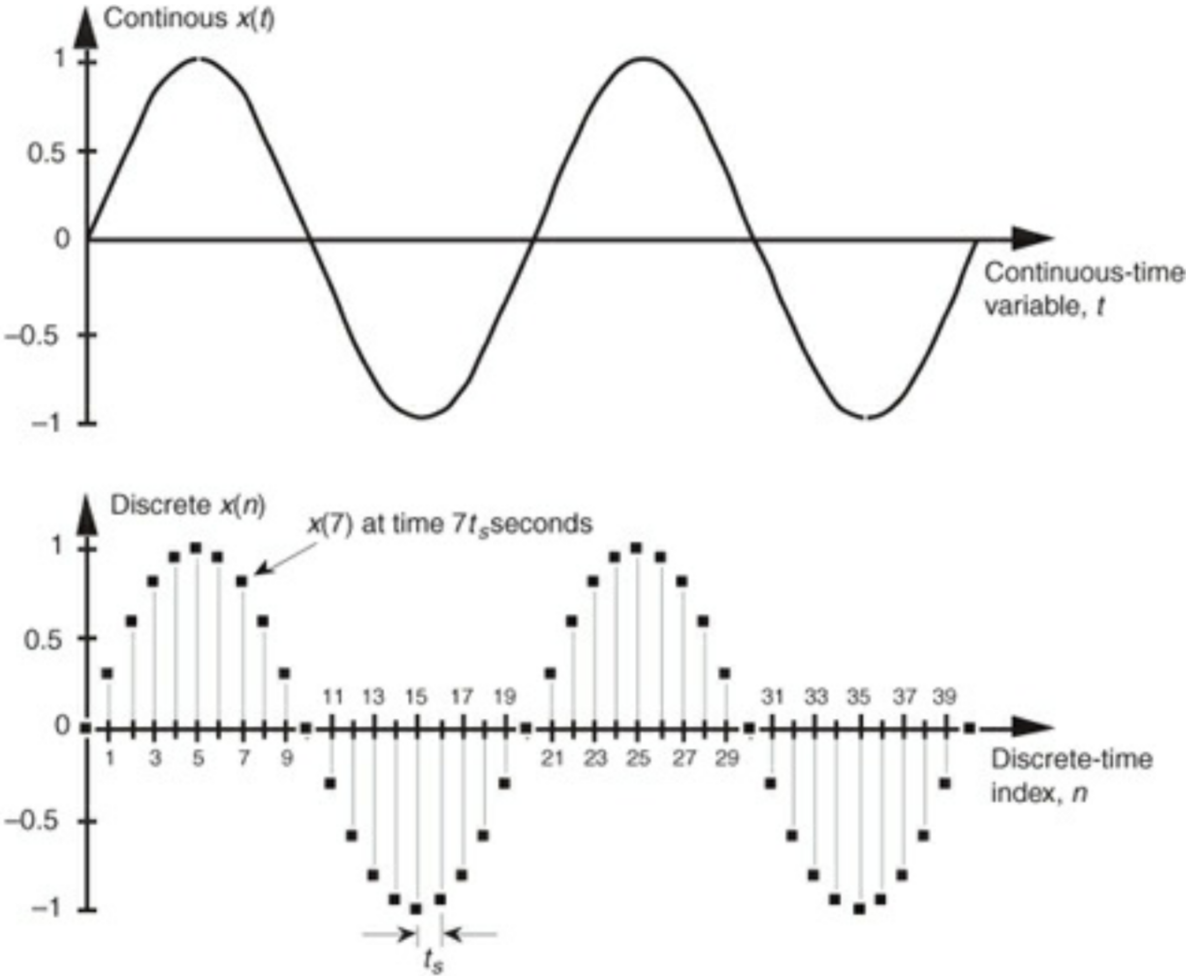


Fig.2: Representation of the same signal, above as a continuous signal and below as a discrete signal, sometimes referred to as an “impulse train”.

Sound signals are described as “pure” (also referred to as “pure tones”) when they consist of a single frequency. Sound signals composed of multiple pure tones are referred to as “complex”.

A “harmonic tone”, is a pure tone whose frequency is a positive integer multiple of another pure tone which is then called the “fundamental tone”.

As an example, let a fundamental tone be 64 Hz. Its harmonic tones would therefore be 128 Hz, 192 Hz, 256 Hz, 320 Hz, and so on to infinity.

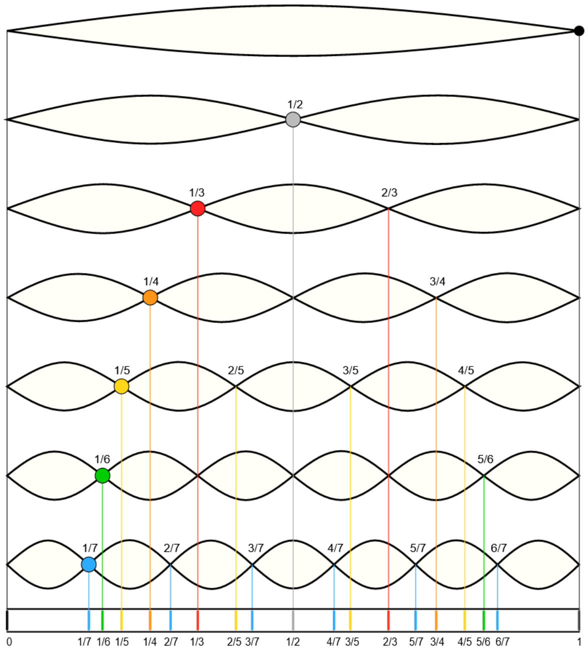


Fig.3: Illustration representing vibrating strings with the fundamental at the top and its harmonics below it.

2.1.1 Time domain and frequency domain signal representations

A signal can be represented in the so-called “time-domain” or in the “frequency-domain”.

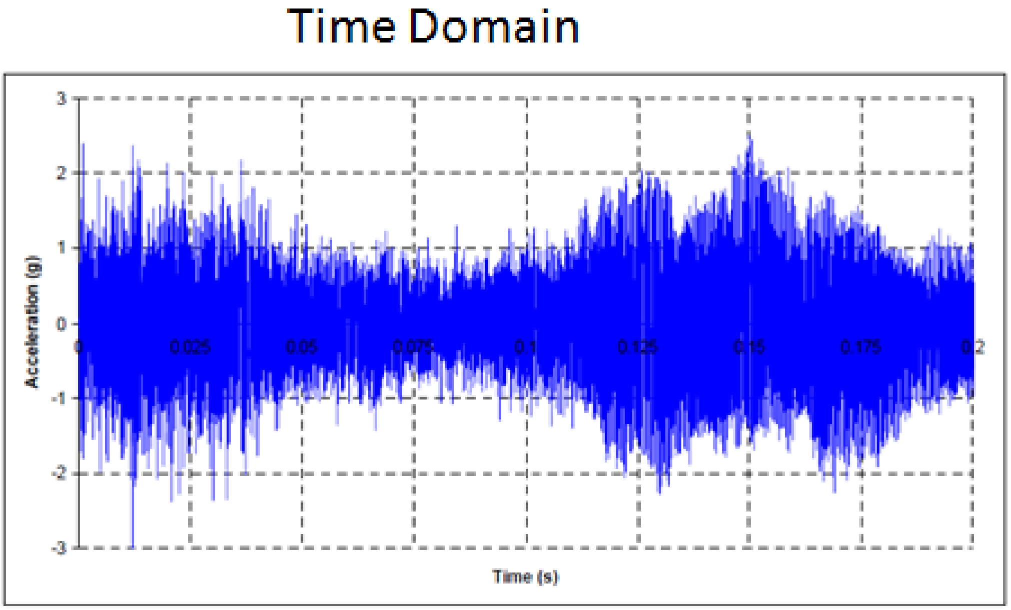


Fig.4: Time-domain representation of a signal.

The graph above represents a signal in the time domain. This means that the signal is presented as a single function evolving over time. This is a useful description to inspect the behavior of a signal as time advances. For example, one might expect the signal of a recording of a single “click” sound to have a sudden peak at the beginning of the graph (left) and to quickly diminish as time goes on. This representation however does not make it obvious what frequencies compose the signal.

Every sound can be synthesized by adding together enough pure tones, meaning simple sine signals with varying amplitudes, frequencies and phases can be used to represent any more complicated sound. The inverse is also true: any complicated sound can be decomposed into a (potentially infinite) set of pure tones using an operation called the “Fourier transform”. This means that taking as input a graph like the one above, by applying a Fourier transform to it, we get the following graph.

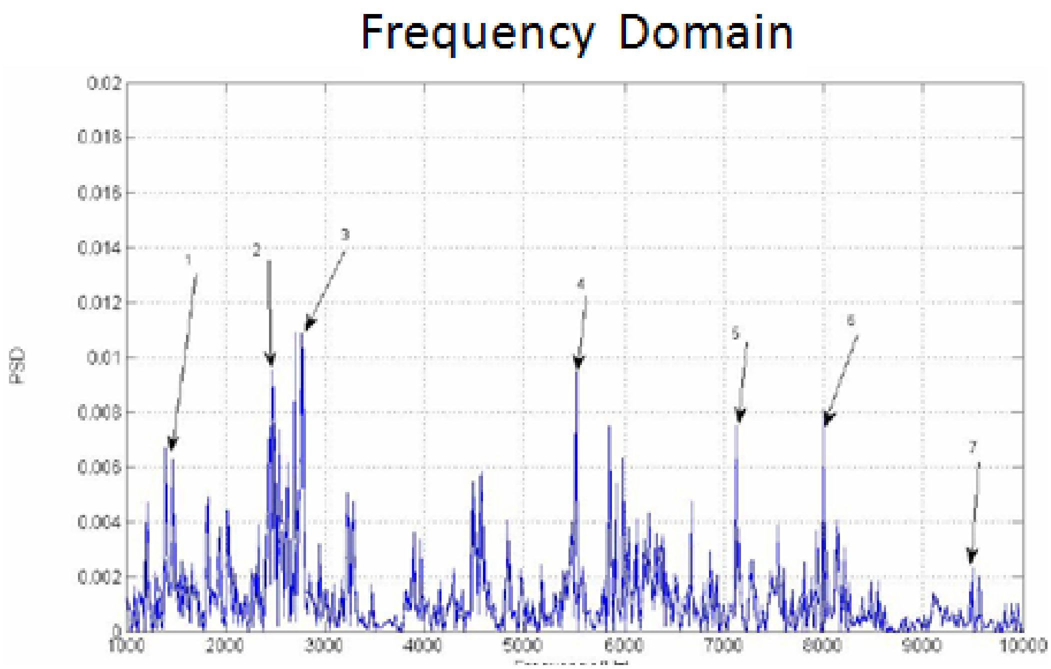


Fig.5: Frequency-domain representation of a signal obtained by applying a discrete Fourier transform (DFT) on a time-domain representation of a signal.

This new graph represents the same signal, but instead of it varying with time, it now varies with frequency. On the horizontal axis is the frequency of a pure tone that would be needed to synthesize the original signal and on the vertical axis is its amplitude, or “how much” of it is needed.

This is an extremely powerful tool that has revolutionized the domain of digital signal processing and lies at the heart of many fields of engineering, not only in digital signal processing.

This representation allows us to inspect and manipulate individual frequencies that compose a sound. If we wanted for example to get rid of the high frequencies in this sound, thanks to this representation of the sound, we would be able to create a signal that would exactly cancel out the undesired frequencies.

2.1.2 Distance attenuation and atmospheric filtering

In real-life, due to the three-dimensional nature of space, as a sound wave propagates through a medium, it will dissipate with distance according to the inverse-square law, and this without accounting for the absorption of the acoustic energy by the medium itself (“Inverse-Square Law,” 2022). This phenomenon is referred to as “distance-based attenuation”.

While modeling distance-based attenuation with an inverse-square formula is realistic, virtually every middleware allows for adjusting this effect via a scalar parameter, if not allowing to outright define a new formula to compute the attenuation. This is done for game design considerations: realism is not always convenient!

As a sound wave travels through a medium, not only does it dissipate via the simple fact of propagation, it also loses energy through friction (Russell, 2016).

Higher frequencies vibrate more quickly and therefore lose energy quicker than lower frequencies. This is why far away sounds appear “deeper” than their closer counterparts. This effect can help us with sound distance perception provided that the listener is familiar with the sound, like the muffled sound of a train in the distance.

2.1.3 Doppler effect

The Doppler effect is the familiar phenomenon one can hear as an ambulance with its siren on passes us by: when the vehicle travels towards us, the siren sound's pitch is higher than when the vehicle travels away from us. This transition can be clearly heard as the vehicle passes us by resulting in a characteristic sudden change in pitch ("Doppler Effect," 2022).
This is due to the relatively slow speed of sound in air. Since the speed of sound is more or less constant (343 meters per second in dry air at a temperature of 20°C), a moving sound emitter will produce higher pitched sounds in the direction it is traveling than the opposite direction.

This effect can intuitively be understood by looking at an illustration of the phenomenon:

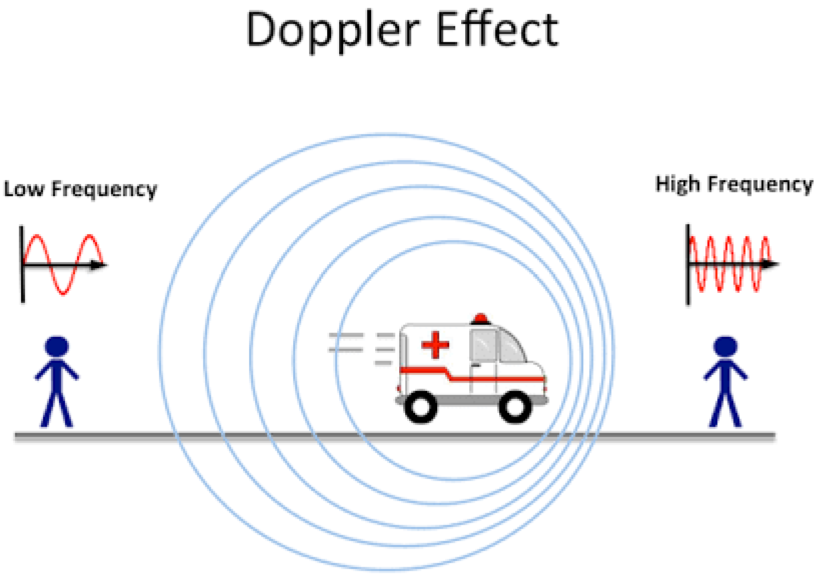


Fig.6: Illustration of the Doppler effect as an ambulance travels left to right with two listeners hearing the same sound at different pitches due to the speed of the moving sound source.

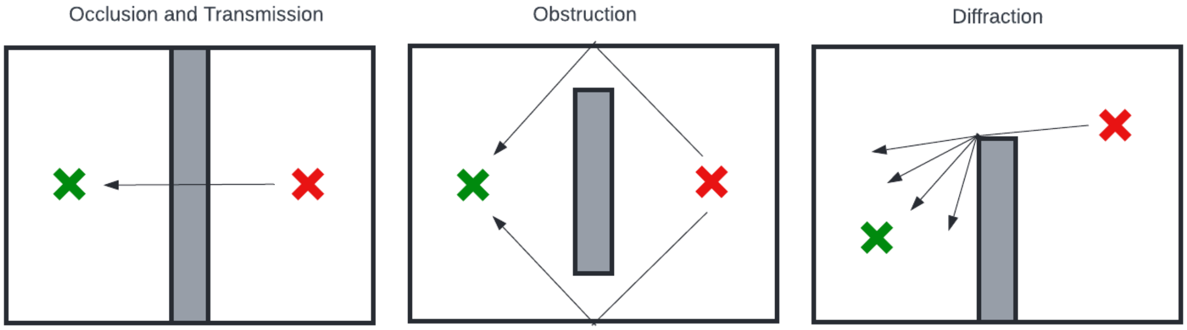
2.1.4 Occlusion, transmission, obstruction and diffraction

A sound is said to be “occluded” when no sound path exists between the sound source and the listener that wouldn’t require the sound wave to change medium (Gregory, 2014). An example of a fully occluded sound might be the muffled music coming from a car with no opened windows or doors.

The sound can still be heard through “transmission” however: the sound wave changes medium through which it propagates, from air to car door back to air in the case of the above example. Sound wave refraction occurs in such a case as well but it is usually not accounted for in real-time sound simulations. Even the phenomenon of transmission itself is seldom modeled in video-games, instead often ignoring sound transmission entirely or approximated it in some games by applying a filtering effect on the output of a sound traveling via a so-called “portalling system” (WildGamerSK, 2022).

A sound source is said to be “obstructed” when the anechoic path is entirely blocked but sound can still travel to the listener via reflection and diffraction.

Diffraction is a wave propagation phenomenon that occurs when part of a wavefront encounters an obstacle. The resulting wavefront changes direction, changing the perceived location of the incoming sound. This effect is the reason an obstructed sound gradually becomes unobstructed in real-life rather than suddenly becoming louder as soon as our ears are in the sound source’s line-of-sight. Games often ignore this phenomenon which results in jarring sound volume changes. Most modern audio engines however account for diffraction by modeling the sound source as a sphere rather than a point and gradually reintroducing the anechoic part of the sound into the final signal by computing the percentage of the sphere's visible area (*Steam Audio*, 2022).



Fix.7: Illustrations of sound propagation phenomena in the presence of an obstacle in an enclosed space.

2.1.5 The convolution operation

The convolution operation is a mathematical operation that, put simply, describes how the shape of one function is modified by another.

The convolution operation can be intuitively understood by illustrating what happens when convolving two rectangular functions:

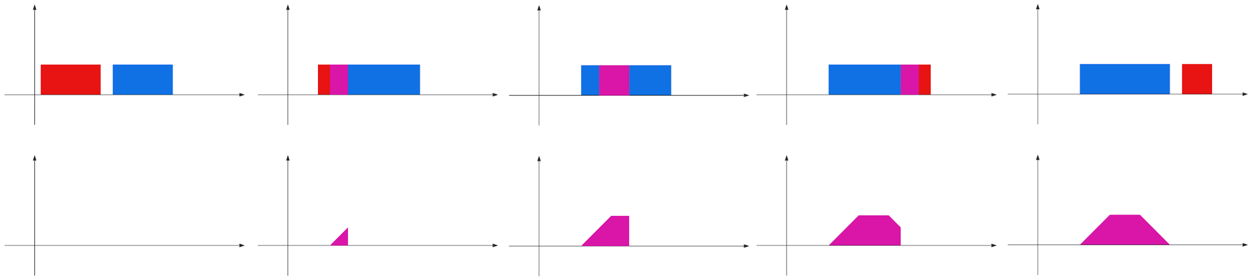


Fig.8: From left to right, each vertical pair of graphs showcases the generation of the result of a convolution (lower graphs) when convolving two rectangular functions (upper graphs).

By looking at the graphs above, one can intuitively understand that the output of a convolution is a function that describes the area under the curve common between two functions as one is “slid over” the other.

In the context of binaural sound spatialization, convolving a sound signal with a single so-called “HRIR” results in an output sound signal that sounds like the original sound coming from the location at which the original HRIR measurement has been taken (Cuevas-Rodríguez et al., 2019). A description of the HRTF and HRIR’s is given later in this document.

Convolution can (and is, in the context of video-games) be done with discrete signals and those interested are invited to watch the Youtube channel “audioease”’s very elegant explanation of it in its “Altiverb 7 guided tour” video (audioease, 2011).

2.1.6 Spherical coordinates

While the cartesian coordinate system is familiar to every high-school graduate, the spherical coordinate system is worth introducing in the context of this work.

In a spherical coordinate system, the triplet that defines a position in 3D space is composed of two angles and a distance instead of the cartesian’s X, Y and Z vectors (“Spherical Coordinate System,” 2022).

The azimuthal component, usually denoted as “φ”, defines the horizontal signed angle between an arbitrary position and the forward axis. It is often referred to as the “azimuth” or the “longitude”.

The sagittal component, usually denoted as “θ”, defines the vertical signed angle between an arbitrary position and the up axis. It is often referred to as the “elevation”, “inclination” or sometimes the “polar coordinate”.

The angles are measured in a counterclockwise, CCW, fashion, either in radians or degrees.

The third component of the triplet, usually denoted as “r”, is the unsigned distance to the arbitrary position from the origin, usually measured in meters. It is often referred to as the “radius” or the “depth”.

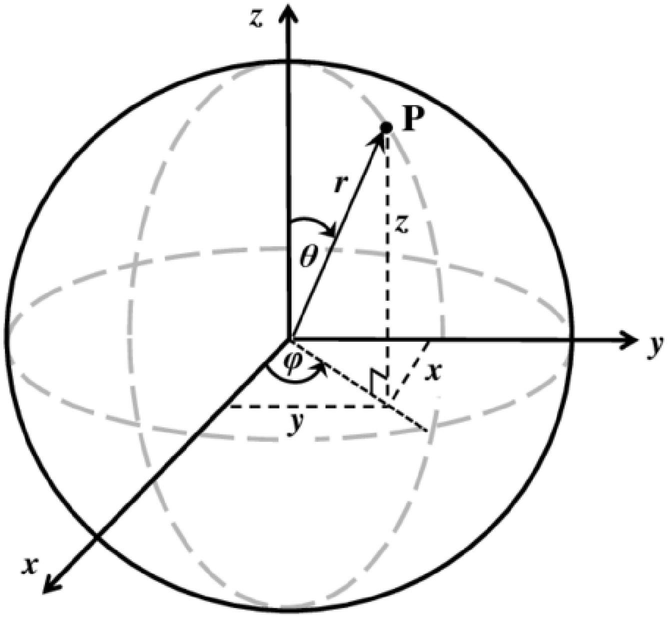


Fig.9: Illustration of a point P represented both in cartesian and spherical coordinates.

For this work, measures of azimuth are limited to the range [-Π;+Π] (or [-180°;180°] in degrees) and sagittal measures are limited to the range [0;+Π] (or [0;180°] in degrees).
Note also that confusingly, some sources invert the “φ” and “θ” notations.

2.1.7 Ambisonics

Ambisonics is an audio signal format developed in the 1970’s by the British National Research Development Corporation (“An Introduction to Ambisonics,” 2017). As its name indicates, it is capable of representing a full sphere of a sound-field. The term “ambisonic” itself is composed of the latin “ambi-”, signifying “on all sides” and “sonic”, signifying “sound”.

Ambisonic audio is recorded using a special type of microphone composed of four (or more for higher-order ambisonics) microphone capsules, each affixed to the surface of a tetrahedron (the three-dimensional simplex). The raw output of each capsule together composes what is called the ambisonic “A-format”.

Once encoded, this data takes on the form of the ambisonic “B-format”, a representation of the sound-field using the 0th order spherical harmonic for the W channel and the three spherical harmonics for the X, Y and Z channels. The mathematical details of the B-format go well beyond the scope of this work (and the author’s understanding).
This configuration is referred to as “first-order ambisonics” and is the most relevant for the purposes of real-time sound spatialization. Higher-order ambisonics can be used but their use increases the computational cost. It is noteworthy that the 3DTI Audio Toolkit used in the practical experiment of this work implements their own variations of the B-format that can reduce the number of B-format channels to process from four to three or one, at the cost of spatialization quality.

While this format was historically underused, it is now making a resurgence in some audio applications, among which is video-game audio. This is arguably due to a very useful characteristic of the B-format: it is playback setup agnostic. This means that it encodes an ambisonic sound signal in a generic way which can then be “downmixed” to a target playback devices setup, such as mono, stereo or various surround or so-called “object-based” sound systems. This genericity has also allowed the conception of “virtual ambisonics” presented later in this work (Cuevas-Rodríguez et al., 2019).

2.2 The perception of sound

We have given a physical definition of sound earlier in this work but it is worth defining it again in the context of human listening. In this new context, sound is the sensation of hearing and some new concepts are worth explaining.

The “timbre” or “color” of a sound is a very vague term used to describe the qualities of a sound that do not directly have their origin in its pitch: a note played on a piano will indeed sound very different from the same note played on a guitar, this stems from a difference in timbre.

While the term has no precise definition, it is generally agreed that the reverberations and the so-called “envelope” of a sound signal at least partly define its timbre (“Timbre,” 2022).

The reverberations of the body of a grand piano for instance will result in a very different timbre than those of the body of a violin.

The envelope of a sound is a term used to describe the behavior of a sound in the time-domain. The sudden plucking of the string of a harp indeed sounds very different from the smooth raise in volume of an alphorn.

While these terms have been described in the context of musical instruments, they still apply to every sound that is not a pure tone. One can tell from the reverberations whether they are in a large cathedral or inside a claustrophobic closet from the color of sounds for instance.

2.2.1 Localization cues

Localizing a sound involves the complex task of extracting spatial information from a set of sound signals composed of a lot of irrelevant information, a process that is still the subject of research. This document will not explain the myriad of sound processing processes involved in human hearing such as sound source identification, neural coding of sound or selective auditory attention (Yost et al., 2008). We will instead solely focus on sound localization, meaning the process by which an approximate spatial position of a sound can be deduced by the brain.

The extraction of this information is done via so-called “cues”, patterns in sound signals that our brain has learnt and/or evolved to detect to infer spatial information from signals.

These can be segregated into “monaural”, “interaural” and “dynamic” cues when studying psychoacoustics but it is important to remember that the brain does not make such distinctions during listening, all of these cues are processed simultaneously and used together to extract information (“Sound Localization,” 2022).

2.2.1.1 Monaural cues

One might be surprised to realize that they can still localize the origin, or at least the direction of a sound, fairly accurately when they plug one of their ears. After all, if sound behaves in many ways similarly to light, why do we not lose all sense of space when we plug one ear the same way we lose all sense of depth when we close one eye (provided we cannot rely on object familiarity)?

This is because not all cues are binaural, meaning they don’t all require two ears. As sound propagates from the source to the eardrum, it is affected by reflective surfaces. Sound waves, as all waves interact with each other in a phenomenon known as “interference”.

As a sound propagates, it is reflected and diffracted by geometry, such as the walls of a room, the torso of the listener or the pinnae. The interaction of all these waves creates a unique pattern in the sound signal by the time it arrives at the eardrum. This unique pattern allows the brain to estimate the origin of the sound, even with a single ear (Ballou, 2015). As each individual’s anatomy is different, these monaural cues vary significantly from person to person and one of the challenges of accurate sound spatialization is to be able to take into account these anatomic variations.

Due to the fact that our pinnae gets in the way of sound coming from behind us, monaural cues are also the ones allowing us to distinguish between sound coming from in front of us or from behind us, as well as whether they are coming from above or below us.

Monaural cues don’t only inform us about the origin of a sound but also about the environment. We are able to tell, with our eyes closed, whether we are in a tiny closet or in a large hall for instance thanks to monaural cues. In this case, the amount of reverberations is one of the carriers of such information.

2.2.1.2 Interaural cues

While monaural cues provide a great amount of information to the brain about the environment, the most useful cues for locating a sound are interaural ones (also referred to as “binaural” cues).

Sound is, relatively speaking, slow. It is slow enough for the brain to be able to deduce the direction of origin of a sound. This is what is known as the interaural time difference, ITD, and interaural level difference, ILD.

If the signal arriving at one ear is nearly identical to the one arriving at the other, with only a slight shift in the signal’s phase, meaning it is “offset” on the time axis, it is safe to assume that these two signals correspond to the same acoustic event. The phase variation therefore can be used by the brain to approximate the incoming direction of the sound. This is the ITD.

Similarly, if the signal arriving at one ear is nearly identical to the one arriving at the other, with only a slight difference in the magnitude of the signal, meaning how “loud” it is, it is safe to assume that these two signals correspond to the same acoustic event as well. The magnitude variation can therefore be used by the brain to also approximate the direction of the incoming sound. This is the ILD (Blauert, 2013).

Together, these cues are the main sound spatialization tool our mind has, which explains the success of stereo audio despite its total lack of accounting for monaural cues.

2.2.1.3 Dynamic cues

Finally, sounds are never isolated “impulses”, meaning signals of length nearing 0. Most sounds are emitted continuously as they move or as we move our head. Hearing a sound over time allows our brain to hone in on the exact location of a sound as more and more data accumulates about its location. These are called dynamic cues and they are very important for sound localization. A listener will have a much harder time locating a short sound without moving their head than if the sound were longer and the listener could rotate their head (Ballou, 2015).

This is the reason behind the success of audio in VR: no other popular entertainment medium allows for tracking of the head and therefore allows for sound spatialization as convincing as VR does.

2.2.2 Effect of the environment on localization

The environment doesn’t have a direct effect on sound localization: it is the ITD of the first wavefront that dominates the localization of a sound (concept referred to as the “precedence effect”). However, reverberations do contribute heavily to our perception of a space. We can, for instance, easily determine from the color of a sound whether it is played inside a large cathedral or a tiny closet.

This additional information conveyed by reverberations together with the listener’s mental model of the space may allow them to more precisely localize the origin of a sound. As an example, one might deduce that a highly reverberant sound heard from a building with a single opened window while the listener is outside comes precisely from the opened window rather than any other that may lay in the general direction from which a sound has been perceived.

It is also noteworthy that sound simulations that account only for the anechoic part of a sound tend to result in a “centering” of the perceived acoustic events as has been observed in a few experiment sessions conducted as part of this work. This is most probably a psychoacoustic effect that comes from the fact that large spaces tend to be more reverberant, so a sound spatialized without any reverberations appears closer than it normally would in real-life.

2.2.3 The near-field effect

In the context of binaural audio, the “near-field effect” refers mainly to the acoustic shadowing created by the presence of the listener’s head.

Due to the scale of acoustic waves (~2 centimeters to 17 meters), when a sound source is near to a human head, the sound changes drastically due to acoustic shadowing phenomena created by the human anatomy (“Acoustic Shadow,” 2022): the head prevents the propagation of high-frequency sounds which results in a noticeable spectral difference between the sound signals reaching the ears. This difference contributes greatly to our ability to localize very close sounds.

The reader might be familiar with the so-called popular “ASMR” recordings that take advantage of this phenomenon to create recordings that some find relaxing.

The exact distance at which the near-field effect begins seems to vary in every source documenting it but for the purposes of this work, it is considered that it becomes audible from 1 meter away from the listener and below, the same distance as is used by Marià et al.

2.2.4 Reversals

Monaural cues provide us with information necessary to discriminate between sounds coming from the front and back. Without them, discriminating between would be impossible, as is suggested by Spherical Head Models (SHM) which are in essence the basis of sound spatialization based on audio panning (Cuevas-Rodríguez et al., 2019).

Usually however, even those simplified spatialization models apply a filter to simulate the effect of the pinnae on sound propagation, though usually only on the azimuthal plane as is the case with FMod’s default spatializer.

Despite this, especially in reverberant environments, confusion can occur that results in the sounds being perceived behind the listener when the actual position of the sound lies in front of the listener, and vice-versa. This phenomenon is called a “front-back reversal”, FBR.

Much less studied reversals are the up-down reversals and the very rare left-right reversals that follow the same principle.

2.3 Spatialization

Sound spatialization is the process that embeds spatial information into a sound signal that allows the listener to localize it. Sound spatialization is therefore an artificial process done by a machine, more precisely by an audio spatializer in the context of video-games.

Classically, sound spatialization is done on the azimuthal plane via panning, a process that splits the audio signal between multiple sound drivers to spatialize a sound. The Vector-Based Amplitude Panning, VBAP is one panning method, one that is used in FMod by default (Pulkki, 1997).

More recently however, we've seen a popularization of the so-called "binaural" spatialization, a family of sound spatialization methods that use convolution with impulse responses to produce the spatialized sound.

2.3.1 Describing sound

When working with sound for real-time sound spatialization, we often discriminate between some of its components. From the most global point of view, sound can be separated into two parts: the "anechoic path" and the "reverberations". In the world of video-game development, an additional separation is made between the so-called "early reflections" and "late reverberations" that compose the overall reverberations.

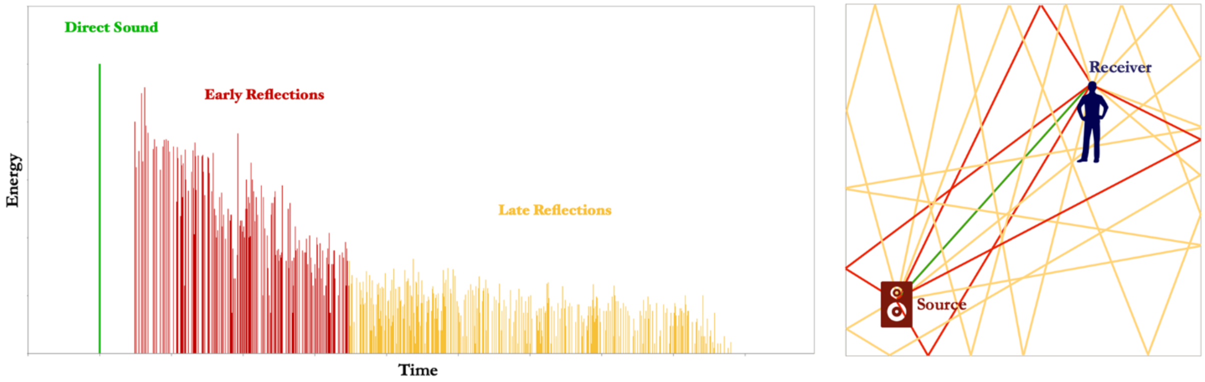


Fig.10: Illustration of the anechoic, early reflections and late reverberation parts of a sound.

The “anechoic path” (also called the “direct sound”) is the path taken by sound that directly travels from the sound source to the listener without reflecting off of anything nor being blocked or filtered by anything. Professionals in the audio domain often refer to this component as the “dry” part of a sound.
This component of the sound is the carrier of the ITD and ILD and therefore plays a crucial role in the listener’s ability to locate the direction from which a sound reaches them (Fırat et al., 2022).

The complement of the anechoic path is the reverberations of a sound: the sum of all sound waves that have originated at the source of the sound but have not taken the anechoic path. Those typically arrive 50 to 80 ms after the anechoic part of the signal (Fırat et al., 2022). These are sound waves that have bounced around the environment to eventually make their way to the listener’s ears after a practically unpredictable delay and filtered by the path they’ve taken, such as by bouncing off of matter surfaces, diffusing the sound wave, or by bouncing many times, gradually dispersing their energy, resulting in a lower volume by the time they reach the listener’s ears. Professionals in the audio domain often refer to this component as the “wet” part of a sound.
This component of the sound provides us with information about our global environment, it is thanks to this part of the sound that we can tell without the need of sight that we might be in the great hall of a train station, the claustrophobic space of a tiny elevator or in the middle of valley bracketed by echoing mountains.

The early reflections composing the reverberations give us clues of the size of the space we’re in: the larger an enclosed space is, the greater the time delay before the arrival of these reflections to the ears.

The late reverberations on the other hand tell us much about the material of the space surrounding us (Fırat et al., 2022): a room with walls lined with curtains will absorb reverberations much more quickly than a room with marble walls, resulting in a much shorter “tail” of the sound.

It is worth noting that the “order” of a reverberation refers to the number of times a sound wave has reflected off of obstacles before reaching the listener’s position.

While the anechoic part, as well as the early reflections to a certain extent, might be reasonably cheaply and accurately simulated in the span of a single game frame (usually 16.6 milliseconds), simulating the extremely chaotic late reverberations accurately is simply not doable in real-time.

This is why many technical solutions for simulating sound often offer the possibility to tweak the simulation. In 3DTI for instance, the anechoic path can be simulated in high-quality whereas the late reverberations are simulated using convolution with a BRIR or, cheaper still, using a so-called “feedback delay network”, FDN, to imitate reverberations at the expense of acoustic accuracy (Funkhouser et al., 2003).

A feedback delay network is a digital signal processing circuit, often used to cheaply simulate late reverberations of a sound. It does not model physical sound phenomena, instead simply feeding back its own output for further processing after a delay. This results in a reverberation-like sound effect. These circuits can usually expose arbitrary parameters for sound designers to tweak to obtain a desired result.

2.3.2 Sound filtering

Conceptually, filtering is a modification of a signal that attenuates, amplifies or applies a phase-shift on a subset of frequencies that compose it and lets all other frequencies pass unchanged. A filter can be digital as is the case for video-game audio processing, analog or natural.

The subset of frequencies modified by the filter are called the "stop-band" and the ones unchanged are called the "pass-band"(s). The frequency at which the filter begins transitioning from stop-band to pass-band is called the "cutoff frequency" and the steepness of the transition slope is referred to as "roll-off". The middle of the pass-band is the "critical frequency".

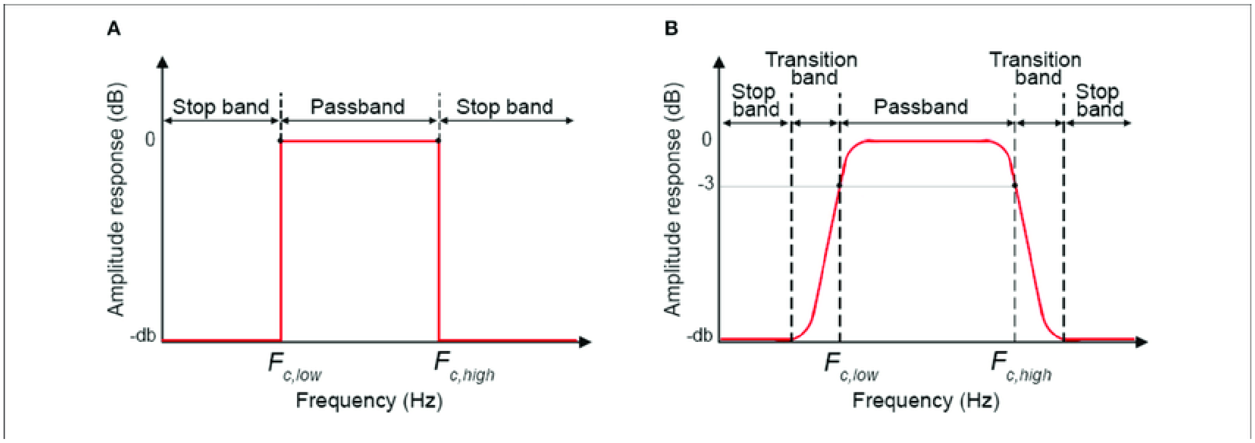


Fig.11: Two graphs describing the response of an ideal filter (left) and that of a real filter (right).

Some basic examples of such filters are the "low-pass" filter, which attenuates high frequencies, or a band-pass filter, which attenuates all frequencies outside of a selected frequency range.

It is worth noting that the "order" of a filter refers to the steepness of the transition between a filter's stop and pass bands. (Ballou, 2015)

2.3.3 Impulse responses

An impulse is a signal of non-zero amplitude the duration of which nears zero. Put simply, think of it as a short and dry sound such as a “click” or a clap of hands.

This kind of signal is very useful due to its simplicity: if you know that when you emit the sound, it approximates an instant non-zero signal, you know that any difference between this simple signal and whatever you might record with a microphone of the same acoustic event is the result of the sound propagating in the environment the sound has been emitted in.

The recording of such an impulse constitutes an “impulse response” and the resulting signal describes the behavior of any sound as it travels from sound emitter to sound listener.

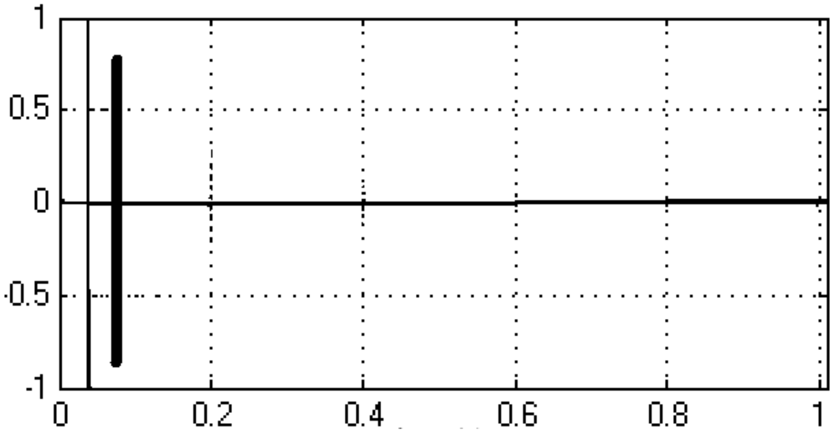


Fig.12: Author’s mockup of an original impulse signal.

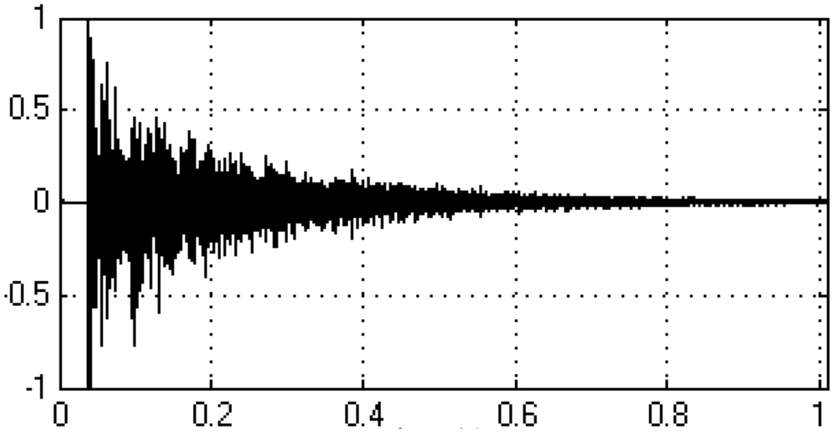


Fig.13: Example of a real impulse response (recording of the mockup as heard by a microphone) of a concert hall.

Note that in practice, a so-called “sweep” sound is played back and then processed to obtain an impulse rather than an impulse to avoid damaging the equipment and the listener’s ears.

2.3.4 HRTF

The head-related transfer function, HRTF, is the fourier transform of an HRIR. Convolving a signal with the HRTF produces the sound signal spatialized at the location the original HRIR has been recorded at.

Alternatively, it is also the colloquial term used to refer to binaural audio spatialization, or alternatively again, the set of all individual HRTF's generated from the set of HRIR's measured inside an anechoic chamber as explained above, which can create a lot of confusion.

It is composed of a set of HRIR's, head-related impulse responses, which encode the effect that the anatomy of a listener has on sound propagation (and that effect only!).

An impulse response describes how a particular environment affects a sound wave as it propagates through it. The head-related impulse response is the exact same concept with the microphone placed inside the ear canal of a listener.

However, if the response were to be recorded in a reverberant environment, the impulse response would not only describe the effect of the participant's anatomy on sound waves but also the effect of the reverberant environment. This is not desired for head-related impulse responses, as its name indicates.

This is why HRIR measurements are usually made inside of an “anechoic chamber”, specialized rooms designed to absorb as much reverberation as possible which results only in the anechoic part of the sound reaching the recording device.

It is very important to point out that the HRIR changes with the relative location of a sound source. As a sound source is rotated about the listener, the effect of the listener's anatomy will also change. This is why when recording HRIR's, the result is an array of impulse responses, usually named in a manner to allow localization of individual sound emitters around the listener, via the use of spherical coordinates for instance.

To record a set of HRIR’s (sometimes confusingly referred to collectively as “the HRTF”), the listener is placed in the middle of an anechoic chamber on a rotating platform with a set of speakers arranged in a semi-circular array. A small microphone is placed inside each ear of the listener to record everything heard.

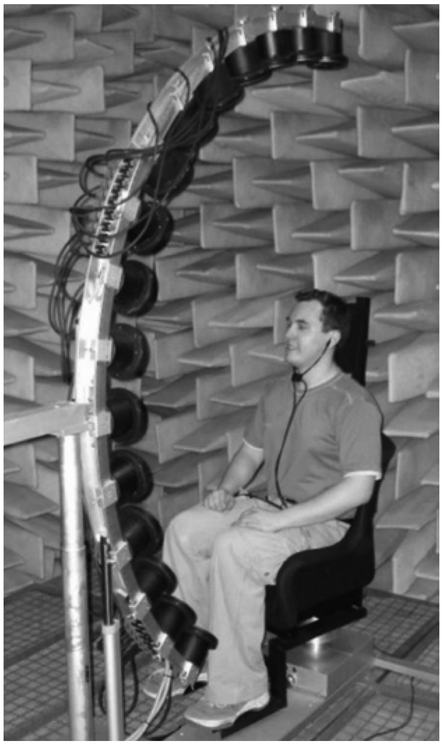


Fig.14: A listener is sitting on a rotating platform in front of a semi-circular arrangement of speakers for the purposes of measuring his HRIR’s.

2.3.5 BRIR

BRIR stands for binaural room impulse response. Whereas HRIR's encode the effect that a listener's anatomy has on propagating sound (and that effect only), BRIR's similarly encode the effect of an environment on sound propagation. Depending on the setup, this may or may not include the effect of the listener's anatomy as well (3DTI notably requires the anechoic part of the impulse response to be removed).

Just like the HRTF, it is a set of impulse response measurements to a sweep, played from an array of speakers but this time arranged in a semi-spherical manner. The microphone is placed in the middle of the room and similarly rotated at regular increments once all speakers have individually played the sweeping sound, until a full sphere of measurements has been recorded.

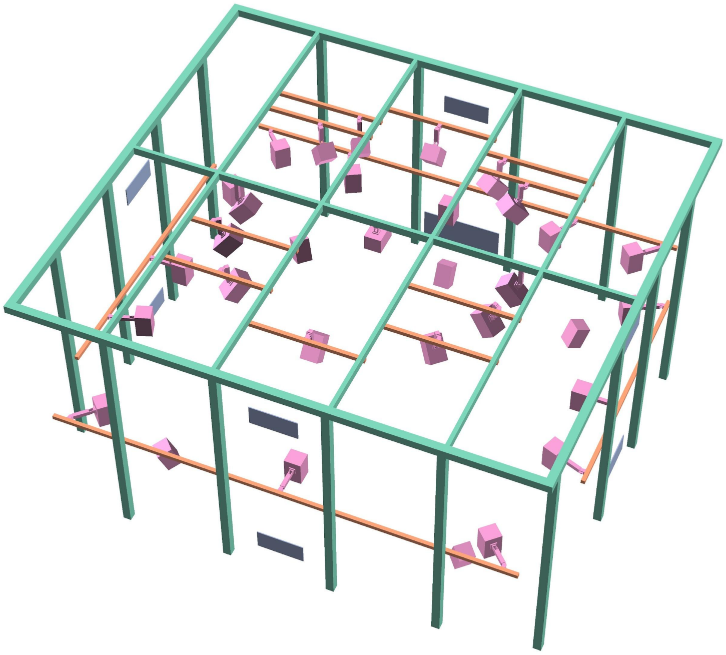


Fig.15: 3D model of the BRIR measuring setup used for the recording of the bbcrd-brirs dataset.

2.3.6 Propagation paths enumeration

While stereo and surround sound spatialization techniques result in sounds that can be localized, they are not necessarily representative of the environment through which sound had to propagate. To simulate the effect of the environment on sound propagation so-called “propagation path enumeration” methods are used to model how a sound propagates through an environment and these results are then used in the mixing of the reverberations to create a sound signal that sounds grounded in the virtual world’s space (Funkhouser et al., 2003).

Geometric sound propagation paths enumeration methods, usually referred to as “Geometric Acoustics” methods, GA, are those that make the assumption that the wavelength of sounds is small enough not to be affected by obstacles, making it possible to model sound propagation using ray theory. As such, this modeling approach is fairly convincing for high frequency sounds but fails to take into account the non-negligible effects that stem from the relatively long wavelength of lower frequency sounds.

With geometric acoustics techniques, ray propagation paths can typically be pre-computed for static environments to alleviate the computational load of the simulation in real-time as is done in SteamAudio’s Phonon sound spatializer.

Image source methods are ones best suited for very simple environments (so-called “shoeboxes”) and are notably quick as they enumerate propagation paths by using the symmetric properties of convex regular spaces.

Geometric sound propagation methods are usually used for the anechoic and low-order reverberations as computation of high-order reverberations increases in cost in an exponential manner.

2.3.7 Virtual ambisonics rendering

While simple binaural rendering is enough to spatialize a sound in a very convincing manner, it is not without its drawbacks.

Firstly, convolution is an expensive operation to perform in real-time. Partial baking of such data is possible (Cuevas-Rodríguez et al., 2019) but the cost of sound spatialization increases exponentially.

This is where usage of so-called “virtual ambisonics” comes into play. Virtual ambisonics is a technique whereby all the sound signals prior to convolution are encoded into a single ambisonic B-format signal which is then convolved with an impulse response rather than convolving each sound signal individually.

While convolving a B-format signal is initially more computationally intense than convolving a simple mono signal, the beauty of this approach is that the increase of the number of sound sources only affects the performance in the spatialization stages prior to binauralization, the performance cost of binauralization itself remains the same.

This has allowed the 3DTI middleware for instance to have a sub-linear processing cost increase as the number of spatialized sound sources grows (Cuevas-Rodríguez et al., 2019).

Note that 3DTI offers different simulation qualities that allow compromise between quality and computational efficiency. Note also that unlike the I3DL2, the usage of BRIR files implies that the overall environment cannot be configured via 3DTI directly. Room impulse response synthesizers hoever exist and can surely be used with this aim.

2.3.8 Importance of binaural sound

Despite the positive perception of binaural audio and the marketing efforts of audio hardware providers, the limited existing studies investigating the effect of binaural audio on user experience report mixed results (Pike, 2019).

Studies do suggest that a justified and well implemented use of binaural audio can lead to a qualitatively better listening experience but they also show that inappropriate use of the technology tends to result in a worsening of the listening experience.

It is therefore important to motivate the use of binaural audio by weighing the potential improvement of the listening experience against the added complexity of production and processing in the case of video-games. If done correctly however, binaural rendering can significantly improve the player’s experience. It is also worth noting that existing studies investigating the effect of binaural audio on the listener’s experience have been focusing on non-interactive listening experiences, such as music listening or the watching of television series. Due to the dynamic nature of video-games, binaural audio might arguably have a greater impact on the user experience than other media.

2.3.9 Interview with Dr.Firat

A written questionnaire has been sent to Dr.Hasan Baran Firat, a practicing acoustic consultant based in Turkey with a Bachelor of Science in Mechanical Engineering, a Master’s degree in Building Physics, a Doctorate in Industrial Design and Cultural Heritage as well as a passion for the preservation of Turkish musical heritage.

Dr. Firat is currently working on leveraging virtual reality (VR) technology to create historically accurate soundscapes: auditory experiences that allow the user to experience the sounds of historical places in the past, such as the city of Istanbul.

From this interview have surfaced insights relevant for the current work. Firstly, Dr.Firat has been very enthusiastic about the prospectives stemming from the popularization of VR and the impact it will have on the popularization of binaural audio.

Dr.Firat has observed significant progress in sound simulation techniques standard in the video-games industry but still believes that a change of perspective is necessary in the creation of audio engines design with greater emphasis on physical accuracy.

According to Dr.Firat, advances in real-time diffraction simulation, early reflections simulation and sound scattering simulation can be important areas for the video-games industry (Firat, 2022).

Relating to this work, Dr.Firat has pointed out the influence reverberation might have on the results of the experiments. This factor has indeed shown to affect the participant's localization ability in a significant manner. Therefore, for most of the experiment's sessions, sound has been spatialized by taking into account only the anechoic part of sounds.

2.4 Hardware and Software

The "game engine" is the overarching code structure used to run a given game. A game needs a lot of various code to run which is usually separated into "modules", separate and usually modular sets of code (hence the name). The structure managing the operation and inter-operation of these modules is the game engine.

Such modules can be an input manager, a rigid body physics simulator, an animation engine, a logger and so on. The module of interest for this work is the audio engine.

An audio engine is an overarching structure responsible for all things related to sound. Jason Gregory describes the responsibilities of an audio engine as such in his “Game Engine Architecture” book:

“The primary tasks performed by the 3D audio engine are as follows:

1. Sound synthesis is the process of generating the sound signals that correspond to the events occurring in the game world. These might be produced by playing back pre-recorded sound clips, or they might be procedurally generated at runtime.

2. Spatialization produces the illusion that each 3D sound is coming from the proper location in the game world, from the point of view of the listener. Spatialization is accomplished by controlling the amplitude of each sound wave (i.e., its gain or volume) in two ways:
    a. Distance-based attenuation controls the overall volume of a sound in order to provide an indication of its radial distance from the listener.
    b. Pan controls a sound’s relative volume in each of the available speakers in order to provide an indication of direction from which the sound is arriving.

3. Acoustical modeling heightens the realism of the rendered soundscape by mimicking the early reflections and late reverberations that characterize the listening space, and by accounting for the presence of obstacles that partially or completely block the path

between the sound source and the listener. Some sound engines also model the frequency-dependent effects of atmospheric absorption and/or HRTF effects.

4. Doppler shifting may also be applied to account for any relative movement between a sound source and the listener.

5. Mixing is the process of controlling the relative volumes of all the 2D and 3D sounds in our game. The mix is driven in part by physics and in part by aesthetic choices made by the game’s sound designers.”

On a modern audio engine, this list of responsibilities might also include streaming large audio assets from the hard-drive, modeling sound propagation phenomena and managing input and output audio devices.

“Audio rendering” is the set of responsibilities of an audio engine relevant to this work. Audio rendering (sometimes referred to as “auralization” in academia, this term usually puts emphasis on the physical accuracy of a model) is the process of modeling the interaction of a sound propagating within an environment and the listener.
Note that in video-games, the aim of audio rendering is almost never to simulate real-life phenomena perfectly, doing so would often have undesirable effects on a player’s experience. In video-games audio rendering, suspension of disbelief and real-time efficiency are the goals. Therefore, game design will usually dictate the level of complexity of audio rendering and not all audio renderers need to model the complex physical and listening phenomena of real-life, nor even need to be simulating audio in 3D! Sound models that put emphasis on the listener's experience in this way are referred to as “perception-based” models.

For this work however, we are specifically interested in audio renderers that take into account listening phenomena that will be referred to as from now on as “binaural-based” renderers. Binaural-based renderers are therefore ones that take into account the human HRTF and the environment’s BRIRs, but do not necessarily model sound propagation phenomena, such as obstruction or diffusion.

Some of the steps involved in a binaural-based rendering pipeline can be quite computationally-heavy (Wefers, 2015), especially the processing of early reflections in a binaural-based renderer, which can involve the interpolation of BRIRs, encoding signals to the ambisonic’s B-format and a convolution as is the case for the 3DTI Audio Toolkit renderer (Cuevas-Rodríguez et al., 2019). Doing so every frame would be too costly for a real-time application (~25 milliseconds to process early reverberations on the author’s setup).

Luckily, the fixed sampling rate of audio files and the constancy of the “speed” of time, this cost can be spread out over multiple frames, <3 frames at most assuming a 2048 audio buffer size at 44’100 samples per second at a framerate of 60, if the audio hasn’t been processed by then, the hardware buffers will underrun. This would still seem dire were it not for the fact that modern CPU’s put great emphasis on multithreading and an audio renderer is usually run on a separate thread, leaving in theory <48 milliseconds of processing time (Gregory, 2014).

These budgets are relatively small, but are largely underused by classic audio renderers since renderers like FMod’s were optimized over the years of development and designed often for by now outdated standards.

It is therefore of the author’s opinion that more advanced sound rendering can be used in modern video-games with minimal impact on processing budget.

2.4.1 Audio middlewares

2.4.1.1 3DTI Audio Toolkit

The 3DTI Audio Toolkit is the result of an academic endeavor by Marià et al., a research funded by the European Union's Horizon 2020 research and innovation programme. The open-source toolkit provides a state-of-the-art implementation of a binaural-based audio spatializer.

It is worth noting however that 3DTI expects the HRTF and BRIR files to contain data with very specific requirements, making a processing step necessary in order to use arbitrary HRTF and BRIR files. Details of this can be found in the toolkit's academic publication.

The academic publication accompanying the source code of 3DTI Audio Toolkit describes the spatialization process in great detail. For the needs of this work however, only a summary is presented here of the spatialization pipeline of the middleware.

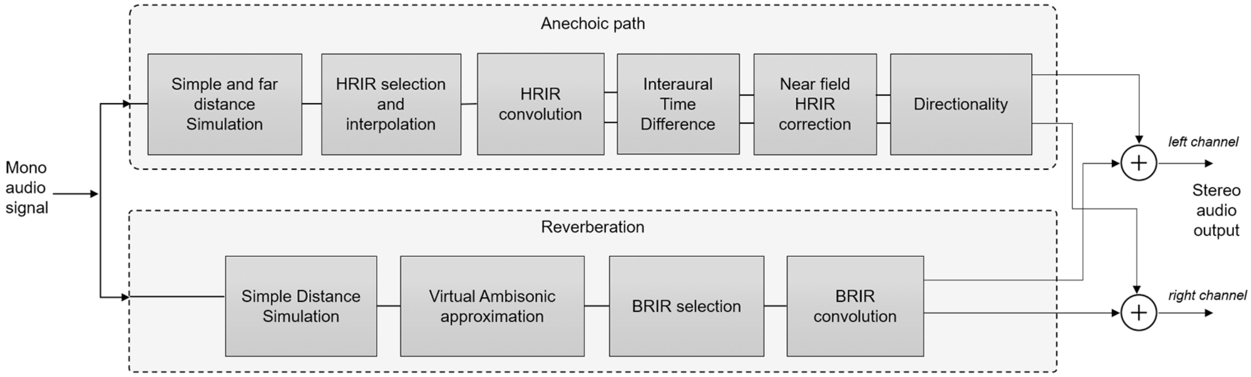

Anechoic path
Simple and far distance Simulation
HRIR selection and interpolation
HRIR convolution
Interaural Time Difference
Near field HRIR correction
Directionality
Mono audio signal
left channel
Stereo audio output
right channel
Reverberation
Simple Distance Simulation
Virtual Ambisonic approximation
BRIR selection
BRIR convolution

Fig.16: From left to right, the diagram describes the processing of a mono audio signal from its raw state up until its stereo representation is sent over to the playback device.

The 3DTI Audio Toolkit design discriminates between the anechoic, early reflections and late reverberation parts of a sound. The anechoic and optionally the early reflections are simulated in high-quality via HRTF convolution. The late reverberations are simulated using a simple FDN.

For a given mono sound signal, it is first split in two, one copy of the signal used for processing the anechoic part of the sound, and the other for the reflections.

For the anechoic part, the signal is attenuated using an inverse-square function with source-listener distance as input, and spectrally filtered to simulate the effect of atmospheric filtering. It is then convolved with an interpolated HRIR which yields a stereo signal.

Once the anechoic part is convolved, the ITD is computed. Both the ITD and the ILD are embedded into the sound signal, optionally along with the near-field effect.

The middleware processes the early reflections part of the sound in a parallel manner. The signal for simulating reverberations similarly starts with an attenuation pass, but from there follows a very different spatialization approach: that of virtual ambisonics.

Rather than processing the reverberations for each sound source separately (which very quickly becomes extremely computationally expensive), each attenuated reverberation is encoded into the ambisonic B-Format. Once every sound active during the current engine frame has been thusly encoded, the signals are combined into a single B-Format signal which is then convolved with an appropriate BRIR. This allows for surprisingly efficient and high-quality reverberation simulation.

The B-Format signal is then decoded back to stereo and is combined with the anechoic part to produce the final spatialized sound for playback by the user application.

Note that due to the binaural approach to sound spatialization, unlike FMod, the 3DTI Toolkit does not offer real-time simulation of arbitrary obstruction, occlusion or diffusion, nor does it model the Doppler effect, all of which are very important phenomena for video-games.
Despite its very impressive performance for binaural sound rendering, it does not offer a full set of features needed for video-game audio, however due to its open-source nature, one can easily integrate the middleware for use in an overarching audio engine as one spatialization step among many.

This middleware is a perfect example of binaural-based real-time audio rendering with its use of the SOFA file format allowing for extensive parametrization of binaural rendering with parameters rooted in well established psychoacoustic concepts.

2.4.1.2 FMod

Although the exact implementation of the default FMod spatializer is not known, the C API’s documentation clearly states that it implements the I3DL2 standard as its default sound spatialization process.
The exact implementation of the standard is not known, however the description of the parameters for the spatializer, the standard itself and FMod’s own documentation helps us to make some educated guesses.

The I3DL2 standard is a perception-based sound spatialization standard designed for real-time applications running on contemporary hardware. It is the parametric sound spatialization method used by default in FMod.
The standard discriminates between anechoic part, early reflections part and late reverberations part of sound and provides tuning parameters to tweak the filtering of the sound as a function of

sound-listener distance (both in amplitude only and spectrally in the form of atmospheric filtering).

The standard does not itself provide an explicit callback for implementing a custom distance-based attenuation but merely exposes a scalar multiplier to the user to tweak the output of an inverse-square function. The same distance-based attenuation function is used both for the anechoic and the reflected parts.

The standard also exposes parameters that can be used to suggest the size of the environment via time delays and via control of the tail of the sound.

It also exposes parameters to suggest the overall material that composes a scene. These parameters are used to determine the specularity of reflections, however the exact usage of the parameter is implementation-dependent.

The standard also defines parameters to control transmission, obstruction and diffusion effects. It however does not explicitly define what sound path enumeration method should be used.

Despite its age and parametric nature, this standard gives convincing results when configured appropriately.

In addition to the features defined in the I3DL2 standard, the documentation of FMod states the simulation of the Doppler effect and mentions the VBAP as the panning method.

From there on, FMod sends the spatialized sound to the playback device in an unknown manner. For ThreeDTI, the application services the audio via the RendererManager::ServiceAudio_() callback registered with PortAudio at initialization.

This standard is a perfect example of a purely parametric sound propagation model: it does not include any physically-based parameters, instead focusing explicitly on real-time performance on relatively limited hardware: indeed, the standard was first presented at the Game Developers Conference in California in March 1999!

2.4.2 Digital signal processing

Digital signal processing, DSP, is “ a subfield of signal processing that is concerned with the electronic manipulation of audio signals.” (‘Audio Signal Processing’, 2022).

In a more practical sense, digital signal processing is the engineering field giving us the tools to manipulate and represent digitized signals (such as a .wav recording of a sound for instance).

In DSP, signals are therefore represented in a discrete manner rather than as continuous signals as in the case when working with electronics.

2.4.2.1 Pulse-code modulation

In digital signal processing, for a signal data to be usable, it takes on the form of pulse-code modulation, PCM. PCM is an uncompressed representation of a sound which consists of an array of “samples”, signed values typically as signed integers of varying size.

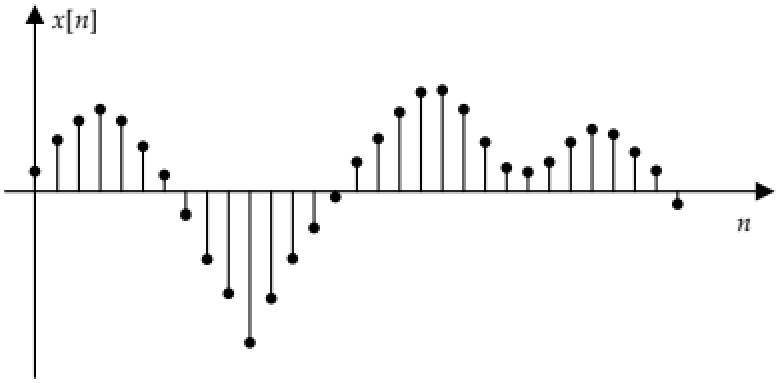
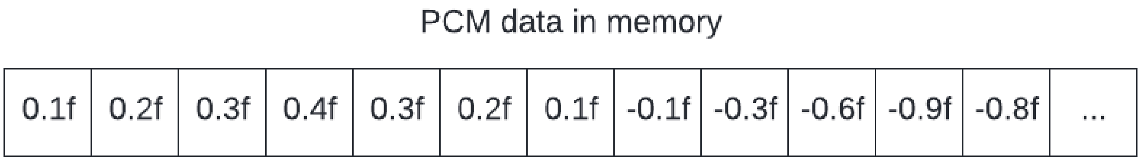


Fig.17: Example of PCM data represented as a graph consisting of discrete impulse functions.



PCM data in memory

0.1f	0.2f	0.3f	0.4f	0.3f	0.2f	0.1f	-0.1f	-0.3f	-0.6f	-0.9f	-0.8f	...

Fig.18: The same data as it would be laid out in the computer’s memory.

2.4.2.2 Sample rate

Each sample represents the amplitude of the signal at a particular point in time and the whole array represents a discrete approximation of the original continuous signal. The number of samples composing a single second of audio data can vary but there are typically 44’100 samples (the Compact Disk audio standard) for a sound file meant for ordinary playback, or 48’000 samples (the DVD audio standard) or higher for high-quality sound files meant applications where sound is of great importance, such as movies or music recordings. For applications where latency is great and a loss of sound quality is tolerable, lower sampling rates can be used, as is the case in VoIP applications (such as Skype or Discord) where a sampling rate of 16’000 is used.

2.4.3 Audio servicing

A computer typically has one (or more) audio input device and multiple audio playback devices. Using them is done via so-called “device drivers”, software dedicated to interaction between a user and the device. A typical application rarely communicates with a driver directly because that would make it dependent on the driver. Instead, interaction is usually done via a cross-platform audio API that handles the heterogeneous interactions with different drivers by providing instead an abstract interface.

PortAudio is one such API. It is an open-source, cross-platform audio API written in C that allows the user applications to interact with audio recording and playback devices. It was used in this project to play the sounds processed by the 3DTI Audio Toolkit since it does not integrate any way to interact with playback devices by itself.

Establishing communication with a playback device is done by creating a “stream”, an abstract object representing a connection between an application and the device through which data can be passed.

Once communication with a playback device has been established, data needs to be sent for playback. This is referred to as “servicing the audio” and can be done in two ways.

Servicing audio via a “callback” is an audio servicing method whereby audio is transmitted to a device upon arbitrary request. A function needs to be registered with the audio API that copies a sound signal into a buffer and the API will then call the function whenever new audio data is needed.

Servicing the audio via “blocking I/O” is an audio servicing method whereby the user is responsible for providing audio data to the API at regular intervals. The contents of the stream can therefore be visualized as a FIFO queue.

The implementation for this work’s practical experiment uses the former method.

2.4.4 Considerations for video-games

For video-games, realistic sound simulation is most important for first-person or third-person perspectives. For top-down perspectives, the lack of familiarity gives us plenty of opportunities to simplify or outright omit sound propagation phenomena. For first-person perspectives especially however, due to the familiarity of perceiving sounds from such a point of view makes the omission or simplification of certain sound phenomena jarring.

By far the most important sound phenomena to simulate for a first or third-person video-game are those that stem from reverberations such as occlusion and diffusion. Despite not contributing to sound localization in a direct manner, the cues embedded in reverberations can be of crucial importance, especially for competitive video-games.

Transmission and more importantly obstruction can also be very important phenomena to simulate, otherwise the player would be left to hear sounds coming through solid objects which can be very detrimental to the player's experience.

For games with fast-moving sources, not accounting for the Doppler effect can be jarring.

As for human sound listening phenomena, accounting for interaural cues is primordial, be it via the use of HRTF or via panning.

Distance-based attenuation is also crucial to allow the player to localize sound. Atmospheric filtering can (depending on the game design) also be very important for video-games taking place in very large environments for the player to gauge the distance to the sound source.

Unless the game design requires so, the near-field effect can be safely ignored since for the greatest part of video-games sound sources are never located closer than 1 meter away from the listener.

Ultimately, the choice in the sound simulation depends largely on the game design needs of the game. Given the elements presented in this section, it becomes apparent that an appropriate and well implemented use of binaural audio not only has the potential to improve the realism of a video-game experience but also exposes new tools by which game designers may inform and guide the player through the game.

3. Practical project

Detailed here is the unfolding of the experiment, its goals as well as its limitations.
The goal of this experiment is to gather information on how well each identified spatialization method allows the listener to localize a sound.

3.1 General description

The practical component of this work consists of an experimental comparison of the quality of sound spatialization done via two approaches: a “classic” approach versus a “binaural” approach.

The classic approach to spatialization consists of a collection of various ad-hoc sound spatialization methods whose design puts emphasis on real-time computation efficiency. These methods have been created over the years as answers to game design requirements and are only very loosely based on the reality of sound propagation phenomena. Due to their arbitrary nature, the implementations of the sound simulations may vary widely and are often proprietary.

Well known examples of software implementing such solutions for sound spatialization via middlewares include many modern game engines such as the Unreal Engine or Unity.
On the other hand, the binaural approach to sound spatialization is a recent development in the video-gaming industry (*List of Video Games with Binaural Audio*, n.d.) made possible thanks to the increase in computational power of CPU’s, the multi-threading emphasis of the new CPU’s as well as recent developments in optimization of real-time signal convolution (Wefers, 2015), (Gregory, 2014).

This second approach to sound spatialization is more in-line with physical sound propagation models and psychoacoustical listening models but up until recently has required too much computing resources to be viable for real-time applications.

We see the gaming industry, as well as the consumers, become more aware of the effect of high-quality sound spatialization, the cornerstone of which is the concept of “binaural” audio, the definition of which is unfortunately greatly obfuscated by confusing and at times contradictory marketing. In essence, binaural sound spatialization is one that takes into account the physiology of the listener as well as the overall effect of the environment on the sound. This is accomplished via convolution of sounds with impulse responses of the listener and of the environment.

In comparing these two approaches to sound spatialization, we will be able to demonstrate the importance of developing the field of real-time, physically-accurate sound spatialization.

3.2 Definition of the used metrics

The “quality” of sound spatialization being very subjective, the author has decided instead to narrow the definition of quality to what can be viewed as one of the key components of it: the accuracy of sound localization.

In this experiment, each sound has a physical position in cartesian 3D space. The frame of reference is configured to have the origin of the space positioned roughly at the position of the test subject’s head.

3.2.1 Mean euclidean localization error

The main quantitative metric used to measure spatialization quality is the mean euclidean localization error, defined as such:

$$errEuclidean = \frac{1}{N}\sum_{i=1}^{N} \Delta p_i$$

Where Δp is the euclidean distance between the perceived position p' and actual position p.

3.2.2 Mean azimuthal localization error

Since human perception of sound directionality in the azimuth is the most precise, a mean azimuthal localization error is computed from the same experiment results that describes only the error on the azimuth. It is important to make this separation since classic sound spatialization methods typically handle elevation in a very limited manner or not at all. It would therefore be useful to examine the localization error in the azimuth plane only to gain more meaningful insights from the experiment’s results.

The azimuthal localization error is defined as such:

$$errAzimuthal = \frac{1}{N}\sum_{i=1}^{N} \Delta\varphi_i$$

Where $\Delta\varphi$ is the angle between the azimuth components of p' and p converted to spherical coordinates defined by using the formulas:

$$r = \sqrt{x^2 + y^2 + z^2} \, ,$$

$$\varphi = tan^{-1} \frac{y}{x} \, ,$$

$$\theta = tan^{-1} \frac{\sqrt{x^2+y^2}}{z}$$

Where x, y and z are the components of a 3D position p.

3.2.3 Mean sagittal localization error

Similarly, the sagittal localization error is also separated for analysis in the same manner:

$$errSagittal = \frac{1}{N} \sum_{i=1}^{N} \Delta\theta_i$$

Where $\Delta\theta$ is the angle between the elevation components of p' and p converted to spherical coordinates.

3.2.4 Mean depth localization error

Finally, the depth localization error is also separated for analysis. This component may be interesting to inspect since it is well known that auditory depth perception in humans is non linear.

It is similarly defined as such:

$$errDepth = \frac{1}{N}\sum_{1}^{N} \Delta r$$

Where Δr is the difference between the radius components of p' and p converted to spherical coordinates.

3.3 Limitations

To keep the quantitative results of the experiment meaningful, a number of limitations have been imposed.

Firstly, only stationary sounds are used. This ensures that the participants are only asked to identify a single position for the origin of the sound which should result in more accurate position identification by the participants.

Conversely, rotations of the head are not taken into account. These two limitations eliminate the influence of dynamic cues from the results of this experiment. While dynamic cues provide the listener with significantly better localization, eliminating them makes the experiment more easily reproducible since tracking of the listener’s head rotation is no longer necessary.

Thirdly, the range of randomized sound locations is limited to [70°;123°] in elevation and [0.33;1.9] meters in distance from the listener. This is to ensure that the participants are able to report the perceived location of a sound by physically moving the VR controller to the perceived origin of the sound.

Finally, it is worth noting that while the 3DTI Toolkit is an open-source middleware, FMod is not. This means that for FMod, we are left to deduce the spatialization methods used based solely on the API's usage and documentation.

3.4 The hardware used

All the tests were performed on a machine with the following specifications. The controllers of a VR headset were used only as a way for participants to indicate the perceived origin of the sound. A studio quality set of headphones with a flat frequency response were used to play the spatialized output.

3.4.1 Computer

- CPU: AMD Ryzen 7 5800H with Radeon Graphics @ 3.20 GHz
- GPU: NVIDIA GeForce RTX 3070 Laptop GPU with 8 Gb of VRAM
- RAM: 32.0 GB @ 3200 MHz
- OS: Windows 11 for x64 architecture

3.4.2 VR headset

- Product name: HTC Vive Pro
- Screen: Dual AMOLED 3.5" diagonal
- Resolution: 1440 x 1600 pixels per eye (2880 x 1600 pixels combined)
- Refresh rate: 90 Hz
- Field of view: 110 degrees

3.4.3 Playback device

- Product name: Beyerdynamic DT 770 Pro 80 ohm
- Open/Closed: Closed
- Fit Style: Circumaural (Around the Ear)
- Noise Attenuation: Passive Noise Isolating
- Frequency Response: 5Hz-35kHz
- Impedance: 80 ohms

3.5 Sound used

The sound used in this experiment was recorded in a professional studio to obtain the dries possible sounds with the help of Timothée Crettaz, a fellow Audio Engineering bachelor student. Usage of a sound recorded by the author allows us to make a direct comparison between the localization errors resulting from middleware spatialization with a control scenario, the recorded sound being reproducible in real life.

The sound used is “olegSpeech_44100Hz_32f.wav” and is located in the “resources/AudioSamples/” folder of the repository. It is a brief speech of the author’s voice. A speech sound has been chosen because human hearing is most sensitive to sound around the frequency range of speech.

3.6 Experiment’s environment

The experiment unfolds in a dedicated 2x2x2 space, with the origin of the space defined to be at 1x1x0 of the space (middle of the space, on the floor). A chair is placed at the origin for the participant to sit on. The VR headset is placed below the chair. A set of headphones and a

VR controller is placed on the chair for the participant to use. The conductor's PC is set up off to the space, just outside of it. Two of the HTC Vive Pro's tracking basestations are placed at the edge of the dedicated space, focusing on the center of the space.

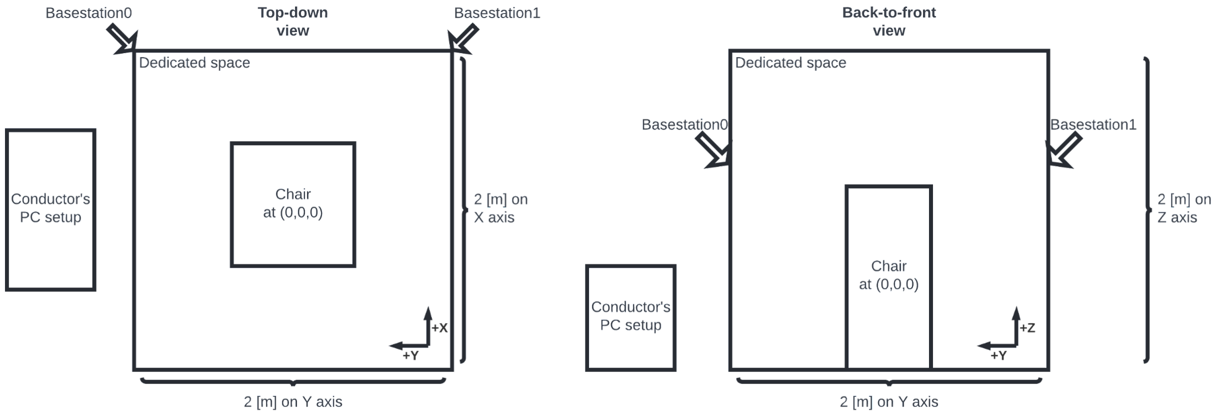


Fig.19: Diagram of the experiment's environment viewed in two perspectives: a top-down view (left) and a back-to-front view (right).

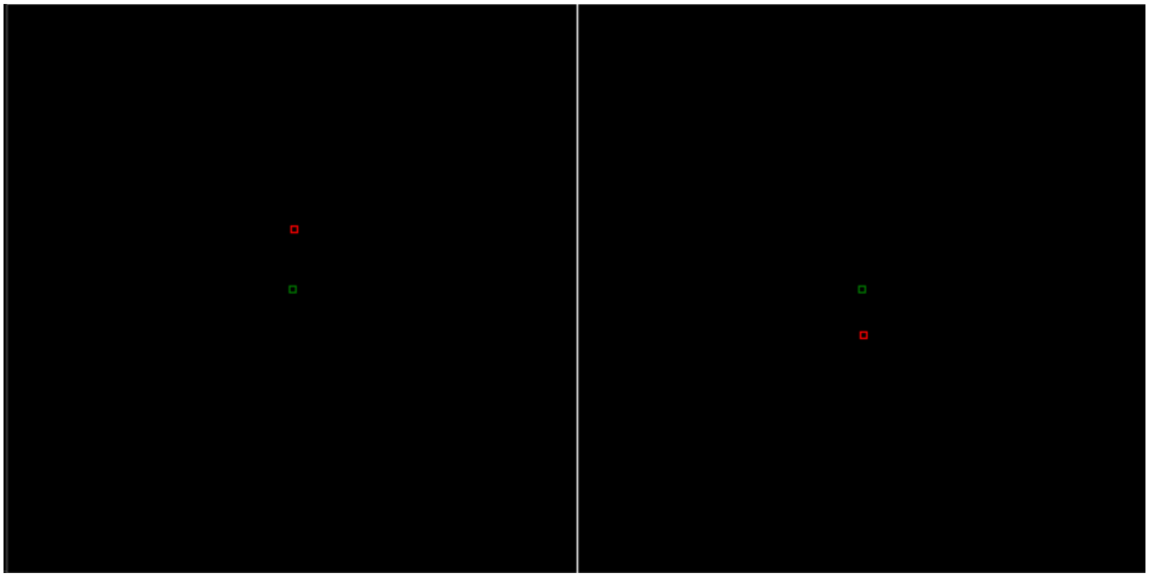


Fig.20: The same two views as above are shown when the experiment's application is running. The green square marks the listener's position and the red square marks the sound source position. The current image shows a sound source located below and in front of a listener.

3.7 Experiment’s procedure

In this section, the author who conducts the experiment is referred to as the “conductor” and the test subjects of the experiment are referred to as the “participants”.

The experiment is split into two scenarios:

1. The “control” scenario is used to gather data on the localization performance of participants in real-life.
2. The “spatialization method comparison” scenario is used to gather data on the spatialization quality of the chosen sounds spatialization methods.

For conducting the control scenario of the experiment, the participant and the conductor proceed thusly:

1. The goal and unfolding of the experiment is explained to the participant.
2. The participant is sat down on the chair in the middle of the room.
3. The participant is instructed to put on the headphones, close their eyes and to keep their head still. They have been instructed during step 1 to take the headphones off when the noise stops but not to open their eyes.
4. The conductor switches on the noise over the headphones to cover their movements.
5. The conductor moves to an arbitrary location around the participant within a 2 meters radius in both the azimuthal plane as well as in elevation relative to the participant.
6. The conductor stops the noise over the participant’s headphones. The participant then takes off the headphones as instructed during step 1 while keeping their eyes closed.
7. The conductor logs the approximate position of their head using their VR controller.
8. The conductor recites the following phrase three times while facing the participant to avoid auditive differences that might arise from the directionality of sound propagation

when spoken by a person: “Le son est la sensation auditive créée par un mouvement vibratoire dans l’air.”

9. The participant puts the headphones back on as instructed during step 1 and the conductor switches the noise back on.
10. The conductor moves back to the participant and taps them on their shoulder to indicate that they can open their eyes and take off the headphones as has been explained to them during step 1.
11. The participant moves to the perceived location of the sound and indicates the 3D position by placing their VR controller at the location and squeezing the trigger, thus logging the position.
12. The participant goes back to the chair in the middle of the room and the experiment continues from step 3.

For conducting the spatialization method comparison scenario of the experiment, the participant and the conductor proceed thusly:

1. The goal and unfolding of the experiment is explained to the participant.
2. The participant is sat down on the chair in the middle of the room.
3. The participant is instructed to put on the headphones and keep their head still.
4. The conductor sets the virtual position of the sound to be played randomly using the provided facilities.
5. The conductor sets the spatialization method randomly using the provided facilities.
6. The conductor plays a spatialized recording of the following sentence three times: “Le son est la sensation auditive créée par un mouvement vibratoire dans l’air.”
7. The participant is then instructed to take off the headphones.

8. The participant moves to the perceived location of the sound and indicates the 3D position by placing their VR controller at the location and squeezing the trigger, thus logging the position.
9. The participant goes back to the chair in the middle of the room and the experiment continues from step 3.

Ideally, the number of iterations for the control scenario and the spatialization method comparison scenario are identical. It is worth noting that the current implementation of random selection of a spatialization method can be improved by enforcing an equal distribution of spatialization methods among a given count of iterations to ensure that the data gathered using the two spatialization methods used can be directly compared without the need to account for the representation disparity between the two spatialization methods.

From this point on, all the data has been gathered and can be stored and visualized using the “positionsVisualizerControl.py” and “positionsVisualizerSpatialized.py” scripts for the control and the spatialization method comparison scenarios respectively.

The data is then transcribed to “Results.csv”, a comma-separated values file encoded in UTF-8.

3.8 The technical implementation

3.8.1 Generating a Visual Studio solution with CMake

An out-of-source CMake build is required located in a folder named “build” placed at the repository’s root. This is required for the “moveDlls.bat” to work properly as well as by the compiled application to correctly load the .wav files from the “resources/AudioSamples/” folder.

All files generated from compiling the project and running the compiled application are located in the subfolders of the “build/” folder.

Two options are provided when building the project with CMake:

- RUN_WITHOUT_VR: Enabling this option bypasses OpenVR. Useful for using the application without a VR headset.
- SEED: Setting this field to a numerical value other than 0 sets the seed for the randomized positions.
- SIMULATE_REVERB: Enabling this option spatializes sounds with reverberations in addition to the anechoic part.
- USE_EASY_PROFILER: Setting this option to true enables profiling of the application with easy_profiler and the serialization of a .prof file under “build/profilingData/” which can be loaded in the easy_profiler’s GUI application located under “thirdparty/easy_profiler/bin/profilerApp/”.

3.8.2 Overview of the solution

The Visual Studio solution is composed of two Visual Studio projects:

- AudioEngineStatic: this project builds a static library from the source files that is used by the application to spatialize a sound. It is composed of a few files:
    - BSCommon: these files contain utilities used by both Visual Studio projects and namely define the application’s cartesian and spherical structures among others.
    - Fmod_AudioRenderer: these files contain the implementation of a wrapper around the FMod C API.
    - ThreeDTI_AudioRenderer and ThreeDTI_SoundMaker: these files contain wrappers for the 3DTI Audio Toolkit API.
- ExperimentApp: this project contains the code for the application itself that uses the spatialization wrappers implemented in AudioEngineStatic. It is composed of multiple files:
    - Application: these files contain the highest level representation of the program. It is responsible for updating all of its components and binds inputs to various actions at startup. This is the class a user interacts with from any external code.
    - Logger: a component of the Application class. Responsible for writing messages to the log file.
    - OpenVrManager: a wrapper around the OpenVR C API that is responsible for gathering events from the VR headset and relaying them to the Application. It is a component of the Application.
    - RandomEngine: a wrapper around the standard C++ random number generator types. It is responsible for providing pseudo-random positions and selecting a random type of spatialization approach. It is a component of the Application.

- RendererManager: a component of Application that switches between the classic and binaural approaches to sound spatialization. Responsible for updating the Fmod_AudioRenderer and ThreeDTI_AudioRenderer as well as playing back the spatialized sound once it has been processed by ThreeDTI_AudioRenderer. The FMod C API does not allow the user to provide their own audio processing callback, the FMod audio engine directly outputs the spatialized signal to a playback device.
- SdlManager: a wrapper around the SDL C API responsible for processing SDL events and relaying them to the Application. It is a component of the Application.

The application can be built in Debug or Release mode, as a x64 application.

4. Results

The above-described experiment has been performed with 11 participants, resulting in a total of 195 data samples. These have been split into 5 categories, referred to from now on as "scenarios":

1. The "control" scenario consisting of 55 samples is the set of measurements taken during the control part of the experiment and represents the performance of participants in real-life.
2. The "3DTI no reverb" scenario consisting of 62 samples is the set of measurements taken during the spatialization part of the experiment using the 3DTI Audio Toolkit for spatialization with reverberations disabled.
3. The "FMod no reverb" scenario consisting of 48 samples is the set of measurements taken during the spatialization part of the experiment using the FMod C API for spatialization with reverberations disabled.
4. The "3DTI with reverb" scenario consisting of 18 samples is the set of measurements taken during the spatialization part of the experiment using the 3DTI Audio Toolkit for spatialization with reverberations enabled.
5. The "FMod with reverb" scenario consisting of 18 samples is the set of measurements taken during the spatialization part of the experiment using the FMod C API for spatialization with reverberations enabled.

It is worth noting that some participants had an audio engineering background (some audio engineering students and teachers) and some no notions of binaural-based rendering (games programming students).

Initially, no separation was made between samples acquired with or without reverberations enabled, however (as predicted by Dr.Firat) it quickly became evident that reverberations have a significant influence on the participant's sound localization ability. The data has therefore been segregated based off not only on the renderer used but also based off the presence of reverberations. The aim of this experience being that of comparing mainly the effect of the anechoic part on sound localization, the "3DTI no reverb" and "FMod no reverb" samples are more numerous than their "with reverb" counterparts.

Spreadsheets are provided within the "csvTables" folder of the repository with are 7 files within:

1. RawInput.csv:

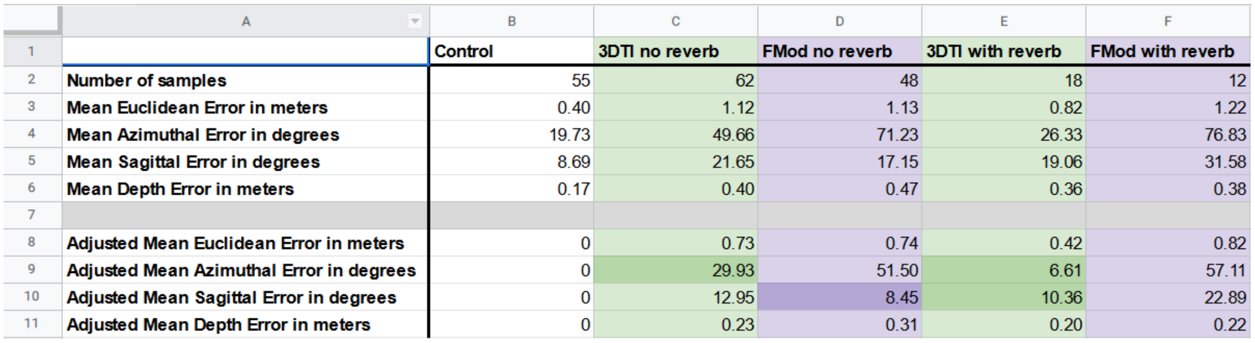
   This table contains the unprocessed results from the experience sessions.

2. Control.csv, ThreeDTI_noReverb.csv, FMod_noReverb.csv, ThreeDTI_withReverb.csv, FMod_withReverb.csv:

   These tables contain the processed results from their respective scenarios. These tables contain entries for the Mean Euclidean, Azimuthal, Sagittal and Depth errors for each sample as well as a table that displays the mean values of these errors for the scenario.

3. Summary.csv:

   This file contains two tables: one with the unaltered mean errors for each scenario and the other with the "adjusted" mean errors. The latter lists the mean errors with the mean errors of the control scenario subtracted. This second table aims to represent only the errors stemming from the usage of the middlewares. However, since the control scenario has not been done inside an anechoic chamber and the reverberation parameters for the 3DTI and FMod spatializers have not been configured to match those of the control scenario, this adjustment might not perfectly ensure that the numbers thusly obtained stem from the spatializer choice only.



	A	B	C	D	E	F
1		Control	3DTI no reverb	FMod no reverb	3DTI with reverb	FMod with reverb
2	Number of samples	55	62	48	18	12
3	Mean Euclidean Error in meters	0.40	1.12	1.13	0.82	1.22
4	Mean Azimuthal Error in degrees	19.73	49.66	71.23	26.33	76.83
5	Mean Sagittal Error in degrees	8.69	21.65	17.15	19.06	31.58
6	Mean Depth Error in meters	0.17	0.40	0.47	0.36	0.38
7						
8	Adjusted Mean Euclidean Error in meters	0	0.73	0.74	0.42	0.82
9	Adjusted Mean Azimuthal Error in degrees	0	29.93	51.50	6.61	57.11
10	Adjusted Mean Sagittal Error in degrees	0	12.95	8.45	10.36	22.89
11	Adjusted Mean Depth Error in meters	0	0.23	0.31	0.20	0.22

Fig.21: Results of comparative analysis.

4.1 3DTI “no reverb” versus FMod “no reverb” scenarios

The use of binaural sound spatialization via 3DTI has not resulted in an improvement as significant as the author expected. For scenarios without reverberations, the Mean Euclidean Error is practically identical between the two spatializers.

More surprisingly however is the increase in the Mean Sagittal Error when using the binaural-based renderer: one indeed would expect a binaural approach to yield significantly better elevation perception than it’s classic panning counterpart that (as to the author’s knowledge) does not encode any elevation information into the sound signal. Despite this, the use of the binaural spatializer has yielded in a greater error than the classic one. The FMod spatializer has an associated 8.45° Adjusted Mean Sagittal Error whereas the 3DTI spatializer has an associated 12.95° Adjusted Mean Sagittal Error, a 53% increase in inaccuracy.

This error increase on the sagittal plane is however compensated by a very significant improvement on the azimuthal plane. Usage of the FMod spatializer yielded an Adjusted Mean Azimuthal Error of 51.5° whereas the usage of the 3DTI spatializer yielded an Adjusted Mean

Azimuthal Error of 29.93°, an improvement of 72% in azimuthal accuracy. Anecdotally, this difference was very perceivable in person: the classic spatialization approach works great for spatializing sound sources at >+45° and <-45° in the azimuthal plane but localization was much more difficult for sources located well in front or behind the listener.

Finally, depth perception might be somewhat improved with the use of a binaural renderer but the difference in Adjusted Mean Depth Errors is too small to assert it with certainty: 0.23 meters for 3DTI versus 0.31 meters for FMod. An increase in the number of data samples might lead to more conclusive results.

4.2 3DTI “with reverb” versus FMod “with reverb” scenarios

While comparison of a listener’s localization ability for spatialized sound with or without reverberations accounted for was not the primary aim of this work, a small set of samples has been gathered for comparison due to the realization of the importance of reverberation for sound localization. As such, the dataset only contains a total of 30 samples and the errors computed from it are probably not very representative, however they still might be of use in showing some general trends.

Unlike for their “no reverb” counterparts, the “with reverb” scenarios have shown a major decrease in Adjusted Mean Euclidean Error with the use of a binaural spatializer: the use of FMod has yielded a 0.82 meters error whereas the use of 3DTI has yielded 0.42 only, an increase in accuracy of 95%.
More impressive however is the difference between the Adjusted Mean Azimuthal Errors: while the use of FMod has resulted in a slight worsening of accuracy of 57.11°, the use of 3DTI has yielded 6.61°, a whopping increase in accuracy of 734%!

The use of a binaural renderer similarly shows a 121% increase in accuracy for the Adjusted Mean Sagittal Errors: 10.36° and 22.89° for 3DTI and FMod respectively.

Finally, the error difference for Adjusted Mean Depth Errors between 3DTI and FMod is too small to be significant (2 centimeters only).

While these results are impressive, it is important to keep in mind the very small size of the dataset. A set of experiment sessions specifically aimed at gathering data for these scenarios would be necessary before giving any serious credibility to the current results.

4.3 “no reverb” versus “with reverb” scenarios

If the results from both of these sets of samples are representative of actual participant performance on a larger scale, a binaural-based renderer offers a clearly superior sound spatialization. If the impressive results of the “with reverb” scenarios however are simply due to a very possible statistical fluke or some other bias introduced by an insufficiently rigorous experiment procedure, this would leave us with fairly mixed results. While the increase in azimuthal accuracy is significant, one of the main selling points of the use of a binaural-based spatializer is the assumed increase in the perception of elevation. These results however illustrate a lack of diminution in the sagittal error. On the contrary, they indeed display an increase in sagittal error, a very strange observation considering that the FMod spatializer outright does not embed any elevation information into the spatialized sound signal (as of the knowledge of the author). This very odd observation might be due to a statistical fluke or perhaps due to a use of a non-individualized HRTF file.

4.4 Further research and improvements

Although there are further avenues for research that would require a modification of the experiment's procedure and the gathering of new data, there is still much that can be done by analyzing the existing data samples. No individual participant analysis has been presented here but in the gathering of data, intriguing observations were made.

Some participants for instance had the tendency to "center" the perceived sounds on themselves, meaning they would consistently place perceived sounds much closer to themselves than other participants (among who is the author). This phenomenon is not due to a lack of clarity in the potential range of positions a sound might occupy, the ~1.5 meters radius from the listener has consistently been made explicit to the participants.

Others would consistently place the majority of sounds behind their head (even when accounting for any leaning of the participant's head).

Some still would perceive a significant change in elevation during the 3DTI with reverb scenario that would disappear in the 3DTI no reverb scenario.

The collected data would also allow one to count the number of front-back, up-down and left-right reversals and the endeavor had already been started by the author before deciding on limiting this work to the analysis of mean errors as described earlier in this document.

As the gathering of data has proceeded, many avenues of experiment procedure improvements became clear.

Firstly, the results of this work would have been much more conclusive if more than one middleware was used to represent the binaural and classic approaches to sound spatialization. The very limited set of middleware used increases significantly the likelihood of introducing biases rooted in the implementations of the middlewares. Wwise and The Miles Sound System might have been used to further represent the classic approach to sound spatialization and SteamAudio’s Phonon and Resonance Audio might have been used to represent binaural-based spatializers used by the gaming industry. The Institute of Technical Acoustics’ VA and EVERTims are also binaural-based spatializers that have been suggested by Dr.Firat as examples of academy standard sound spatializers.

Secondly, a more rigorous experiment procedure might have improved the accuracy of the gathered results: the 3D location of sounds used by the spatializers is relative to the VR headset position and orientation. For the sake of experiment procedure’s ease, the headset has simply been placed below the chair upon which sits the participant and they are instructed to keep their head still in roughly the same planar position as the headset below them. This however is not ideal as participants may inadvertently turn their head and lean slightly forwards or backwards in the chair. This is particularly problematic for participants that have shown a tendency to perceive the location of the spatialized sounds very close to their head.
Additionally, reverberation phenomena have on a few occasions resulted in data samples for the control scenario that were clearly outliers compared to the median localization errors.

While individualized HRTF files are not common among video-game players, for the sake of this experiment it might have been beneficial to take HRTF measurements for each participant to ensure that the binaural spatializer produces optimal results. This however would be a very labor and time intensive improvement to implement (not to mention an access to an anechoic chamber). Alternatively, the usage of a generic HRTF might have alleviated the issue.

Similarly, the use of a BRIR set created specifically for this experiment would have greatly improved the meaningfulness of “with reverb” scenarios. This however would also be very labor intensive and would also not be representative of the actual use of binaural spatializers: recording of BRIR's for a video-game project, as of the knowledge of the author, is exceedingly rare.

Another potential improvement, this time of the technical solution implemented, would have been to stratify the randomized choices of spatializers to ensure an even sample distribution in the results.

As mentioned above, the experiment does not take into account the head rotations of the participants. While this does simplify the experiment's procedure and does make the experiment more consistently repeatable, this decision was not made entirely voluntarily by the author.

FMod, the 3DTI Audio Toolkit and OpenVR all use different axis bases for their cartesian coordinate systems. While conversion of positions from one basis to another is trivial, the conversion of transform matrices, rotation matrices and quaternions requires mathematical understanding that is, as of the making of this document, beyond the understanding of the author.

This lack of understanding has resulted in absurd localization of sounds once the rotation of the head was taken into account. Implementing functioning rotations of the head is by no means unachievable but due to time constraints of the project, it was decided to simplify the scope of the project by restricting the rotation of the participant's head.

Initial plans for the project were to measure the individual impact of various sound cues on localization. This idea was abandoned both due to time constraints as well as due to the fact that realization of such an experiment would require the creation of an implementation-independent way to specify the desired simulation parameters for a spatialization middleware that may or may not support them. More than requiring understanding of sound spatialization, this would require more extensive software design knowledge than what is possessed by the author.

Initially, various types of sounds were to be spatialized in order to provide insight into the influence of the nature of a sound on the localization abilities of the listeners. However, this idea was abandoned since it would require lengthier experiment sessions to obtain reliable results.

4.5 Conclusion

It is worth pointing out that despite the very sub-optimal parameterization of the 3DTI renderer (no individualized HRTF files, a BRIR set not representative of the actual environment), it has still performed better than FMod (with the strange exception of the mean sagittal error). This combined with the known qualitatively better sound perception resulting from binaural sound can be enough of an argument to motivate adopting binaural-based sound rendering as the default sound spatialization approach for most games, the counter-arguments for this being the relative lack of maturity of this technology and the greater computational cost of spatialization. This latter point might or might not be relevant since most game engines dedicate a separate thread for audio processing, but a dedicated study would be required to provide an estimate for the increase of computational cost inherent to binaural spatialization.

Inspecting the 3DTI Audio Toolkit’s implementation, it becomes evident that anatomically-based parameters provided by HRTF files are at the heart of nearly the whole spatialization pipeline of the spatializer. It could therefore be advantageous to use the 3DTI’s pipeline as a basis for more elaborate real-time sound spatializers that aim for realism.

If the synthesization of individualized HRTF files and BRIR sets is made quick and easy by any potential future research, the likes of 3DTI's rendering pipeline might become the basis for future generations of real-time sound spatializers. The adoption of a more rigorous format of HRTF and BRIR files would also increase the usefulness of binaural renderers, currently almost every renderer requires the data to be either presented in a proprietary format or adjusted to fit the needs of the spatializer's implementation.

Overall, the domain of physically-based real-time simulation is ripe for innovation. Some current industry-standard middleware is still by default using technology that is two decades old, all the while interest in physically accurate sound rendering grows, both public and corporate. Additionally, open-source solutions for binaural-based spatializers are available and waiting to be used in more elaborate audio engines that might integrate them into their own pipelines, resulting in quantitatively and qualitatively better sound listening experience while still providing the sound phenomena simulation video-games make heavy use of, such as the Doppler effect or occlusion.

5. Glossary

API:

Application Programming Interface, the part of a technical solution exposed for use by a user application.

ASMR:

Autonomous Sensory Meridian Response, refers to the tingling sensation that usually begins on the scalp and moves down the back of the neck and upper spine.

BRIR interpolation:

Methods to “mix” together multiple BRIRs to approximate a non-existent BRIR as it would have been recorded between two or more existing BRIR positions.

CPU:

Central Processing Unit, the generalistic computer processor of a personal computer.

DFT:

Discrete Fourier Transform, an adaptation of the Fourier Transform made to operate solely on integer values.

DSP:

Digital Signal Processing, the field of mathematics that studies the manipulation of discrete signals. Alternatively, Digital Signal Processor, a processor dedicated to digital signal processing.

Debug and Release modes:

Two ways of compiling the source code of a program, the former generating machine code that is usually slower but allows to track down issues in the program and the latter generating optimized machine code.

FBR:

Front-Back Reversal, the phenomenon whereby a listener perceives the auditory event as coming from their back despite the location of the acoustic event being located in front and vice-versa.

FDN:

Feedback Delay Network, a family of computationally cheap algorithms that produce reverberation-like auditory effects.

FIFO:

First In, First Out, a method of organizing the manipulation of a data structure where the oldest entry is processed first.

GPU:

Graphics Processing Unit, a specialized electronic circuit dedicated to accelerating visual processing algorithms.

I3DL2:

Interactive 3D Audio Rendering Guidelines Level 2.0 (not actually an acronym), a 1999 standard that defines how an audio signal should be spatialized. The standard is designed for computational efficiency.

ILD:

Interaural Level Difference, an interaural sound localization cue.

ITD:

Interaural Time Difference, an interaural sound localization cue.

PC:

Personal Computer.

PCM:

Pulse-Code Modulation, the common manner of representing a discrete signal.

SHM:

Spherical Head Model, a family of sound propagation simulations that represent the listener’s head as a perfect and featureless sphere.

VBAP:

Vector Base Amplitude Panning, a sound panning algorithm used for its genericicity and its smallest-possible number of speakers activation which results in good sound directionality.

VR:

Virtual Reality, the family of technologies designed to map real-life movements and positions to a virtual space.

Visual Studio solution:

File used by the Visual Studio IDE (Integrated Development Environment) to represent the most global working unit of a technical solution. A solution is composed of projects.

Baking:

A technical term referring to the act of computing and storing data in advance that will be used to accelerate real-time computations.

Basestation:

Steam’s name given to the devices used to illuminate an area with infrared light to allow the tracking of VR devices.

Blocking I/O:

In the context of audio servicing, refers to a method of providing data for playback to (or recording from) a device in a manner that blocks the processing of the program until the audio is serviced.

Callback:

In the context of programming, refers to a function that is “to be called back” at a later time when it is needed by a user program.

Data stream:

In the context of programming, refers to a data structure that allows continuous exchange of data between two programs.

Flat frequency response:

In the context of headphones hardware, refers to a design of the headphones that aims to provide the sound in the least altered manner possible.

Framerate:

In the context of video-game engines, refers to the frequency at which the game iterates. Also referred to as FPS (Frames Per Second).

Middleware:

A technical solution that provides the user program a service that communicates with hardware in an opaque manner.

Mono audio:

Monophonic audio signal, a single-channel signal.

Playback device:

Physical audio device used to play back audio signals sent to it. Speakers and headphones are examples of playback devices.

Real-time and off-line:

Distinction made when discussing computation. Real-time computation is one that is able to be done in a perceptually continuous manner (under 1/10 of a second generally, but usually under 1/60 of a second in the context of video-games). Off-line computation refers to baking.

Refresh rate:

In the context of computer monitors, refers to the frequency at which the monitor redraws the image on the screen.

Shoebox model:

Physical models that represent an environment as a rectangular parallelepiped. Simplifying reality in such a way allows for significant optimizations.

Stereo audio:

Stereophonic audio signal, a 2-channel signal capable of transmitting a signal that embeds interaural localization cues.

Sweep:

A popular method used to measure the impulse response of an environment. The method uses a “sweeping” sound that starts at 20 Hz and goes to 20 kHz over a few seconds. The recording is then processed to obtain an impulse response.

Tail of a sound:

The later part of a sound made up of late reverberations, named so due to its tail-like appearance when visualized in the time-domain.

Threads and multi-threading:

Multi-threading refers to the ability of modern computers to run many programs in parallel. A thread is a single program running in parallel with others.

Wrapper:

In the context of software engineering, a wrapper is an entity that encapsulates and hides the underlying complexity of another entity.

x64 application:

A computer program made to run on x64 computer architecture.

6. Figures

Fig.1 (p.13): *Sound Waves Propagation Illustration Stock Vector—Illustration of scientific, sinusoidal: 60244711*. (n.d.). Retrieved July 21, 2022, from https://www.dreamstime.com/stock-illustration-sound-waves-propagation-illustration-design-image60244711

Fig.2 (p.14): Maël, F. (2019). *Introduction to Continuous Signal Processing* -. https://maelfabien.github.io/machinelearning/Speech6/#3-discrete-vs-continuous

Fig.3 (p.15): Harmonic. (2022). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Harmonic&oldid=1089309050

Fig.4 (p.16) and **5 (p.17)**: *How to convert time domain data into frequency domain data?* (n.d.). Retrieved May 27, 2022, from https://www.researchgate.net/post/how_to_convert_time_domain_data_into_frequency_domain_data

Fig.6 (p.19): Paik, A. (2021). *Doppler effect and its application*. https://doi.org/10.13140/RG.2.2.13553.61282

Fig.7 (p.21) and **8 (p.22)**: Author's own creation.

Fig.9 (p.24): Anthonys, G. (2014). *Application of translational addition theorems to the study of the magnetization of systems of ferromagnetic spheres* [PhD Thesis].

Fig.10 (p.32): Fırat, H. B., Maffei, L., & Masullo, M. (2022). 3D sound spatialization with game engines: The virtual acoustics performance of a game engine and a middleware for interactive audio design. *Virtual Reality*, *26*(2), 539–558. https://doi.org/10.1007/s10055-021-00589-0

Fig.11 (p.35): Pinti, P., Scholkmann, F., Hamilton, A., Burgess, P., & Tachtsidis, I. (2019). Current Status and Issues Regarding Pre-processing of fNIRS Neuroimaging Data: An Investigation of Diverse Signal Filtering Methods Within a General Linear Model Framework. *Frontiers in Human Neuroscience*, *12*. https://doi.org/10.3389/fnhum.2018.00505

Fig.12 (p.37) and **13 (p.37)**: Topa, M., Toma, N., Popescu, V., & Vasile, T. (2008). *Evaluation of All-Pass Reverberators*. 339–342. https://doi.org/10.1109/ICECS.2007.4510999

Fig.14 (p.39): Bujacz, M., Skulimowski, P., & Strumillo, P. (2011, June). *Sonification of 3D scenes using personalized spatial audio to aid visually impaired persons*.

Fig.15 (p.40): *bbc/bbcrd-brirs: An impulse response dataset for dynamic data-based auralisation of advanced sound systems*. (n.d.). Retrieved June 1, 2022, from https://github.com/bbc/bbcrd-brirs

Fig.16 (p.48): Cuevas-Rodríguez, M., Picinali, L., González-Toledo, D., Garre, C., de la Rubia-Cuestas, E., Molina-Tanco, L., & Reyes-Lecuona, A. (2019). 3D Tune-In Toolkit: An open-source library for real-time binaural spatialisation. *PLOS ONE*, *14*(3), 1–37. https://doi.org/10.1371/journal.pone.0211899

Fig.17 (p.52): Gregory, J. (2014). *Game Engine Architecture, Second Edition*. Taylor & Francis. https://books.google.ch/books?id=MCQbBAAAQBAJ

Fig.18 (p.53), **19 (p.63)**, **20 (p.63)** and **21 (p.72)**: Author’s own creation.

7. Bibliography

Acoustic shadow. (2022). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Acoustic_shadow&oldid=1086637233

Agomuoh, F. (2022, May). *The next PC display refresh rate milestone could be 480Hz*. Digitaltrends. https://www.digitaltrends.com/computing/the-next-pc-display-refresh-rate-milestone-could-be-480hz/

Audio signal processing. (2022). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Audio_signal_processing&oldid=1081663633

audioease (Director). (2011, November). *Altiverb 7 guided tour*. https://www.youtube.com/watch?v=EpzNgP8uThs

Ballou, G. (2015). *Handbook for Sound Engineers* (Fourth Edition). Taylor & Francis. https://books.google.ch/books?id=t__vBgAAQBAJ

BLAST Premier (Director). (2021, April). *CSGO Tips and Tricks to outplay your opponents with audio EPOS Audio Advantage #2*. https://www.youtube.com/watch?v=d0h2s8RQjGQ

Blauert, J. (2013). *The Technology of Binaural Listening*. Springer Berlin Heidelberg. https://books.google.ch/books?id=izi8BAAAQBAJ

Cuevas-Rodríguez, M., Picinali, L., González-Toledo, D., Garre, C., de la Rubia-Cuestas, E., Molina-Tanco, L., & Reyes-Lecuona, A. (2019). 3D Tune-In Toolkit: An open-source library for real-time binaural spatialisation. *PLOS ONE*, *14*(3), 1–37. https://doi.org/10.1371/journal.pone.0211899

Doppler effect. (2022). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Doppler_effect&oldid=1092798450

Firat, H. B. (2022, June 15). *Interview with Dr. Hasan Baran Firat* [E-mail]. https://loshkinoleg.github.io/Interviews/Interview_With_Dr_Hasan_Baran_Firat/Interview_With_Dr_Hasan_Baran_Firat

Fırat, H. B., Maffei, L., & Masullo, M. (2022). 3D sound spatialization with game engines: The virtual acoustics performance of a game engine and a middleware for interactive audio design. *Virtual Reality*, *26*(2), 539–558. https://doi.org/10.1007/s10055-021-00589-0

FMOD. (n.d.). Retrieved July 6, 2022, from https://www.fmod.com/

Fourier series. (2022). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Fourier_series&oldid=1086936212

Funkhouser, T., Tsingos, N., & Jot, J.-M. (2003). *Survey of Methods for Modeling Sound Propagation in Interactive Virtual Environment Systems*. https://www.cs.princeton.edu/~funk/presence03.pdf

Gregory, J. (2014). *Game Engine Architecture* (Second Edition). Taylor & Francis. https://books.google.ch/books?id=MCQbBAAAQBAJ

Inverse-square law. (2022). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Inverse-square_law&oldid=1095352473

Letowski, T., & Letowski, S. (2011). Localization Error: Accuracy and Precision of Auditory Localization. In *Advances in Sound Localization*. https://doi.org/10.5772/15652

List of video games with binaural audio. (n.d.). Google Docs. Retrieved July 26, 2022, from https://docs.google.com/document/d/1uAklgDNC_LBOYSBbIzBeso6I4laForovIWiTYx6Ss8k/edit?usp=embed_facebook

Majdak, P., Iwaya, Y., Carpentier, T., Nicol, R., Parmentier, M., Roginska, A., Suzuki, Y., Watanabe, K., Wierstorf, H., Ziegelwanger, H., & Noisternig, M. (2013, May). Spatially Oriented Format for Acoustics: A Data Exchange Format Representing Head-Related

Transfer Functions. *134th Audio Engineering Society Convention 2013*. https://www.sofaconventions.org/mediawiki/index.php/SOFA_specifications

Mutanen, J. (n.d.). *I3DL2 and Creative EAX*. Retrieved July 22, 2022, from https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.108.4379&rep=rep1&type=pdf

Nanite Virtualized Geometry in Unreal Engine Unreal Engine Documentation. (2022, May). https://docs.unrealengine.com/5.0/en-US/nanite-virtualized-geometry-in-unreal-engine/

Parida, K. K., Srivastava, S., & Sharma, G. (2021). Beyond Mono to Binaural: Generating Binaural Audio from Mono Audio with Depth and Cross Modal Attention. *CoRR*, *abs/2111.08046*. https://doi.org/10.48550/arXiv.2111.08046

Pike, C. W. (2019). *Evaluating the Perceived Quality of Binaural Technology* [University of York, Electronic Engineering]. https://etheses.whiterose.ac.uk/24022/

PortAudio—An Open-Source Cross-Platform Audio API. (n.d.). Retrieved July 22, 2022, from http://www.portaudio.com/

Pulkki, V. (1997). Virtual sound source positioning using vector base amplitude panning. *Journal of the Audio Engineering Society*, *45*(6), 456–466.

RTX. It’s On: Ultimate Ray Tracing & AI NVIDIA. (2022, May). https://www.nvidia.com/en-us/geforce/rtx/

Russell, D. A. (2016, March 22). *Absorption and Attenuation of Sound in Air*. Acoustics and Vibration Animations. https://www.acs.psu.edu/drussell/Demos/Absorption/Absorption.html

Sound localization. (2022). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Sound_localization&oldid=1098170712

Spherical coordinate system. (2022). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Spherical_coordinate_system&oldid=1099454686

Steam Audio. (n.d.). Retrieved May 25, 2022, from https://valvesoftware.github.io/steam-audio/

Teng, S., Puri, A., & Whitney, D. (2012). Ultrafine spatial acuity of blind expert human echolocators. *Experimental Brain Research*, *216*(4), 483–488. https://doi.org/10.1007/s00221-011-2951-1

Timbre. (2022). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Timbre&oldid=1081993082

Virostek, P. (2017, March). An Introduction to Ambisonics. *Creative Field Recording*. https://www.creativefieldrecording.com/2017/03/01/explorers-of-ambisonics-introduction/

Wefers, F. (2015). *Partitioned convolution algorithms for real-time auralization*. Logos Verlag Berlin. https://books.google.ch/books?id=IA-bCgAAQBAJ

WildGamerSK (Director). (2022, March 13). *DEAD SPACE REMAKE New Gameplay Demo 18 Minutes 4K*. https://www.youtube.com/watch?v=YNR6-a5qO9U

Yost, W. A., Popper, A. N., & Fay, R. R. (2008). *Auditory Perception of Sound Sources*. Springer. https://books.google.ch/books?id=w9p7t7MOBygC