SOUND PARTICLES

ALL YOU NEED TO KNOW ABOUT 3D AUDIO

Nuno Fonseca, PhD

www.soundparticles.com

Foreword

August 2020

Immersive Sound is the next big thing in audio. Cinema, TV, VR/AR, Videogames, Music, Museums, everyone is moving to 3D audio as a way to immerse audiences and improve their experiences. However, 3D audio brings new concepts and some new/old technologies that professionals may not understand at first, with buzzwords such as 7.1.2, VBAP, 22.2, objects, beds, B-format, AmbiX, ACN, SN3D, HRTF, HRIR.

This eBook, which is based on some talks I've done in past AES conventions, tries to summarize the most relevant information on this topic, covering **Channel-based** audio, **Object-based** audio, **Ambisonics**, and **Binaural**.

I hope you enjoy, and welcome to the world of 3D sound…

Nuno Fonseca
nuno.fonseca@soundparticles.com

About Nuno

Nuno Fonseca (PhD), is the founder and CEO of Sound Particles company, and was the creator of Sound Particles software, a 3D CGI-like audio software used in all major Hollywood studios in productions such as "Star Wars 9", "Frozen II", "Aquaman" or "Game of Thrones".

Former university professor (computer science and music technology areas), Nuno is the author of the Portuguese book "*Introdução à Engenharia de Som*" (Introduction to Sound Engineering), co-author of the Portuguese book "*Desenvolvimento em iOS*" (iOS Development), and author of more than 20 papers regarding audio research.

He is a member of AES, SMPTE, CAS, MPSE, AMPS, and member of the Audio Engineering Society (AES) Technical Committee on "Audio for Cinema".

Table of Contents

Channel-based Audio

Many audio formats, like stereo or 5.1, have a predefined number of channels and predefined speaker positions, and for that reason we call them **Channel-based Audio**. But before we dive deeper and start talking about 3D and immersive sound, it's important to have a better understanding of some basic sound formats in order to lay the foundations for 3D audio. So, let's start with the simplest format.

Stereo

Stereo tries to recreate a **frontal sound image**, using 2 speakers. If we send the same signal to both speakers, a phantom "image" will be generated, and we will perceive the sound as coming directly from the front. If we change the gain relationship between the two speakers, we can make the sound "move" between the left and right speakers.

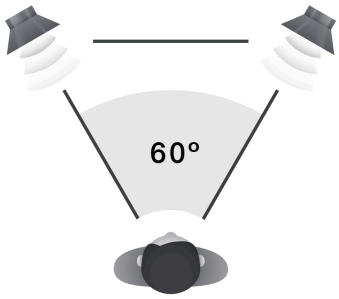

Figure 1 - Stereo Layout : 2 Speakers with an angle around 60°, creating a perfect triangle of equal sides.

With stereo, we usually consider the speakers placed with an angle around **60-degrees**. Why?

On the one hand, we want the sound image to be as wide as possible. If the angle is small (placing the speakers close to each other), the image will be too narrow and small. On the other hand, if we keep increasing the angle far beyond the 60-degrees (moving speakers apart), the frontal image starts to break up. Rather than having a phantom image in front of us, we lose the perception of a central sound, and start perceiving two independent sounds, one coming from the left speaker and another from the right speaker, creating a **hole in the middle**, where no sound is perceived.

Center Speaker

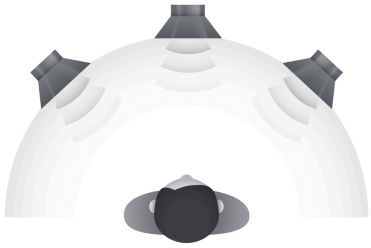

Figure 2 - Stereo layout + center speaker

At a certain point, the movie industry decided to add a central speaker, between the left and right speakers, which brings up the question “If we are able to create a sound image between a pair of stereo speakers, why do we need an additional central speaker in-between?”

Sometimes, I even listen to world-class mixers, who mixed famous music albums in surround, mentioning that they don’t use the center speaker whatsoever, because they don’t miss it. So **why**, do we need the central speaker?

What happens is that the world is **not a perfect place**, and in spatial audio world, this means that not everyone can be at the sweet spot of a room (the perfect listening position). If you are at the sweet spot, left and right speakers would be enough to create the frontal sound image between them, but there are many situations in which the listener isn’t at the sweet spot.

For example, in a movie theater, only a few have the privilege of being seated at the **sweet spot**. Most listeners will be seated too much to the side, or too much in the front or in the back. Left and right speakers are located behind the projection screen near the edges. As you can imagine, anyone who is seated on the front rows ends-up with a left-right angle wider than 60 degrees, which will break the frontal sound image, creating a hole in the middle. Also, for people seated on the sides, they will be much closer to one of the speakers than the other, creating a “distortion” on the sound image (not an audio distortion, but a perception distortion).

For all these situations there’s the need for a central speaker, which will work as an **anchor** that helps us have a more defined and stable sound image, especially for those out of the sweet spot, in the not-so-ideal seats. And this is the reason why the central speaker is so widely used in the cinema industry, where obviously lots of people are spread across a room, not being one of the few lucky ones seated at the sweet spot.

Surround Speakers

To improve the movie experience, the cinemas started to introduce additional speakers on the **sides**. Instead of exploring only the frontal sound image, a movie could now explore other sound directions by adding sounds coming from the sides, which makes the experience become much more realistic, and audiences end-up more immersed in the scene (e.g. a battlefield).

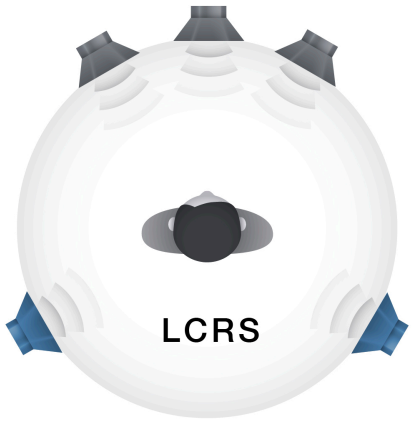

Figure 3 - LCRS layout

One of the initial formats was the **LCRS** (Left, Center, Right, Surround). With it, we started to have 5 speakers and 4 audio channels. In this case, the surround channel is only a single audio channel that is reproduced by the two surround speakers, i.e., the exact same audio is reproduced in the left surround speaker(s) and the right surround speaker(s).

Later, with formats such as **5.1**, an independence from the Left Surround and the Right Surround arose, with each one having their own audio channels.

In movie theaters, surround audio channels are not reproduced by a single speaker, but by an **array of speakers**. If you go to a movie theatre with a 5.1 audio system, you will notice many speakers spread over the lateral and rear walls. But only 2 audio channels are actually being reproduced there. This means that all speakers of the left wall and the ones on the left side of the rear wall are reproducing the exact same thing, and the same happens with the speakers on the right side, which reproduce the right surround channel.

Why do we have this high number of speakers instead of a single speaker on each side?

First, by adding more speakers we are preventing the existence of "**holes**" on the sound image. Yes, we are losing a bit of space resolution because surround sound will now become more **diffused**, almost out-of-focus, but no holes, no places where there isn't any sound.

Secondly, the world is **not a perfect place**, and once again, many people will be seated outside the sweet spot. With this speaker array approach, wherever you seat, you will still be surrounded by several speakers around you. If the movie theater used a single speaker for each surround channel, big “holes” would exist for some listeners – a person on the back will have the surround speaker on the sides (or eventually slightly on the front) and nothing would be playing on the rear; a person seated near the sides could eventually have the surround speaker sound coming from the rear direction and no sound coming from the near wall; etc.

LFE / Sub-woofer

At some point, the movie industry also wanted to increase the impact of sounds with lower frequencies on effects such as explosions or earthquakes. The problem with **low frequencies** is that humans have a lack of sensitivity on those frequency ranges, which means that for a human to hear frequencies bellow 80 Hz, these frequencies need a very powerful sound, with high SPL[1] levels, which require very powerful speakers to reproduce these types of frequencies.

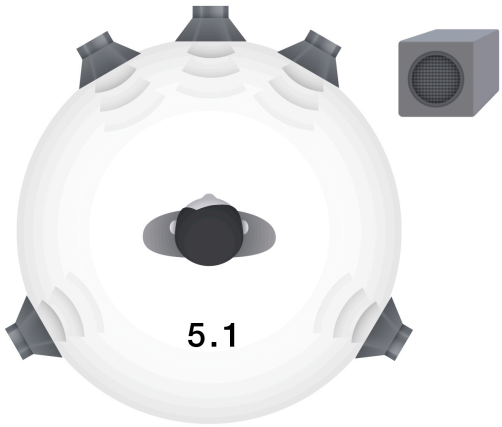

Figure 4 - 5.1 Layout Surround

In order to reproduce those frequencies, there is a need for a **lot of power** and there were two possible ways to achieve the desired result.

The first, and apparently obvious solution, would be to use **more powerful speakers** in order to have a more powerful sound system, and therefore be able to reproduce those low frequencies. This would be an expensive solution as we would have to replace all the speakers for new and more powerful ones. But more importantly, we would have to deal with the problems of the **dynamic range**. At this time, systems were analog and with low dynamic range, which meant that the range between the loudest sound and the background noise of the system was small. Yes, you could now have enough power to reproduce those low frequencies, but you were also increasing the noise level of the system, which would become more noticeable in soft passages, such as dialogue scenes.

[1] Sound Pressure Level

The second approach, and the one with more advantages at the time, would be to add an **additional speaker** only for low frequencies, and with its own audio signal. Why? First, humans are not able to detect the direction of low frequencies, which means that using a single speaker (with its single direction) would not affect the sense of space or result in a decrease of the sound image. Secondly, all other speakers will work at the same power ranges, without having dynamic range issues such as increase of noise level. Third, by having its own audio channel (**LFE**), the systems would already know that this channel should be reproduced at a higher level than the other channels (once again, increasing the power for low frequencies without affecting the dynamic range of other channels). And fourth, any existing movie theaters would only need to buy one additional speaker, instead of replacing all speakers. And with this approach, the ".1" of **5.1** was born – 5 main channels (left, center, right, left surround and right surround) and 1 channel for low frequency effects (the ".1" part of the name).

It's important to notice that **LFE** and **sub-woofer** are different concepts, LFE is the name of the audio channel that transports these low frequencies and sub-woofer is the name of the speaker that reproduces those frequencies.

Other Horizontal Layouts (6.1/7.1)

Besides 5.1, other layouts were created, by adding additional channels.

Sony **SDDS** was created in the 90's and it used a 7.1 setup, but with 5 speakers in the front: left, left center, center, right center, right, while maintaining the 2 surround channels. With the additional speakers on the front, SDDS was mainly focused in theaters with large screens, on which even 3 speakers could not be enough for the frontal rows.

Later on, Dolby created Dolby Digital EX, a **6.1** format with a rear surround speaker to increase the perception of sound on the back, which was released with "Star Wars – Episode I".

Finally, the traditional **7.1** was adopted, with 3 frontal channels and 4 surround channels, increasing the space resolution on the sides/rear.

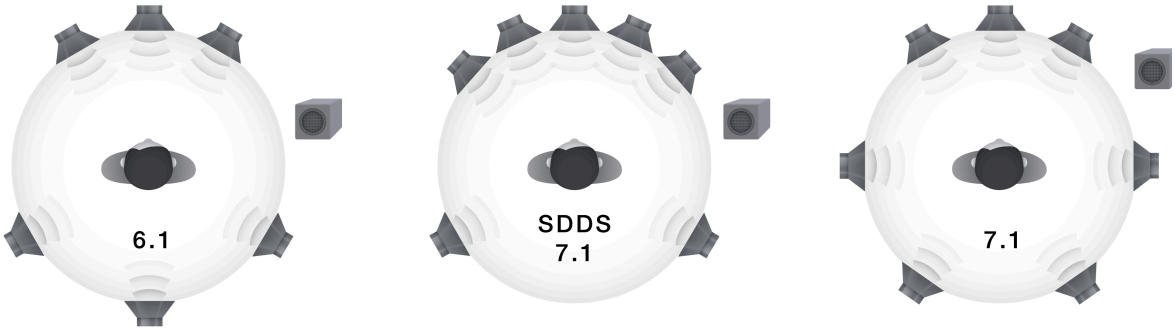

Figure 5 - Dolby Digital 6.1 layout; Sony 7.1 Layout; Nowadays 7.1 format

Immersive Audio

The expression "**Immersive Audio**" usually refers to audio systems that allow listeners to perceive sound coming from other directions besides the horizontal plane. Instead of placing speakers only on the horizontal plane (front, sides, and rear), immersive systems are able to explore the **height** component, for instance, by adding speakers on the ceiling, or even on the floor.

To explore height, several formats arrived, such as Auro 11.1/13.1, IMAX 12.0, NHK 22.2, among others.

For instance, **Auro 11.1 / Auro 13.1** considers a 3-layer speaker setup: an ordinary horizontal plane with 5.1 or 7.1, plus a height layer with a 5.0 setup with an elevation around 30°, plus a speaker above the listener (called "Voice of God", for obvious reasons). This means 5.1 + 5.0 + 1 = 11.1 or 7.1 + 5.0 + 1 = 13.1.

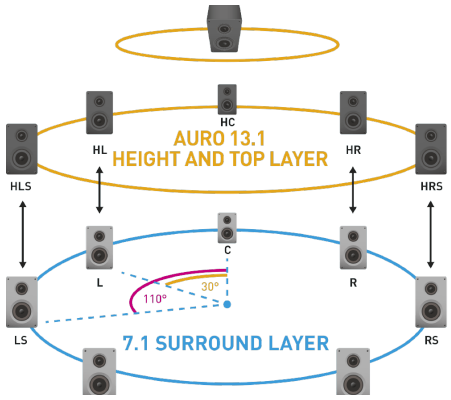

Figure 6 - Auro 13.1 (courtesy of Auro Technologies)

IMAX 12.0 has a slightly different approach. It considers 12 channels, which have the traditional 7 channels at the horizontal plane, plus 5 elevated channels to explore the height dimension, and with no LFE channels (the zero on "12.0").

NHK 22.2, the format created by the Japanese Broadcast company, goes further, with 2 LFE channels (left and right), a lower layer with 3 speakers (bellow horizontal plane, at front, with sound coming from the ground), an horizontal layer with 10 channels (5 at the front and 5 surrounds), and a height layer with 8 channels + 1 “voice of god” (3.2 + 10 + 8 + 1 = 22.2).

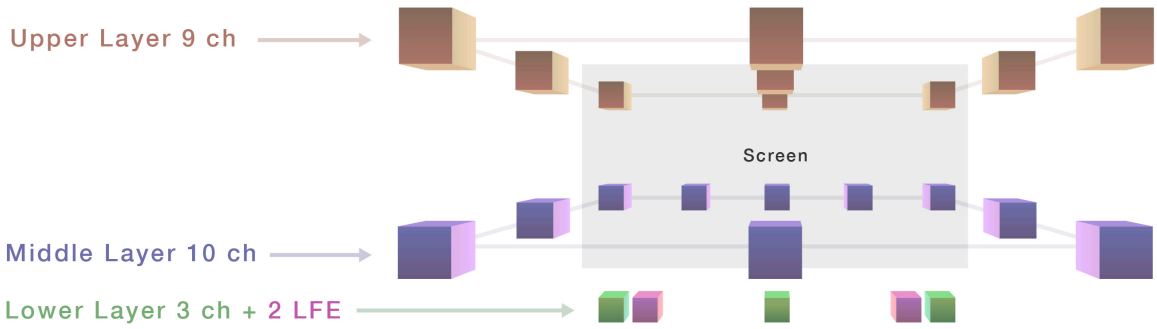

Figure 7 - NHK 22.2

3D PANNING

In the horizontal plane, sound is usually panned between 2 speakers to simulate in-between directions. When we move to an immersive setup, the same applies, but now considering triangles with 3 speakers – by placing more or less sound on each speaker, we can create the perception of sound coming from anywhere within this triangle of speakers. This panning technique is called Vector Base Amplitude Panning (**VBAP**).

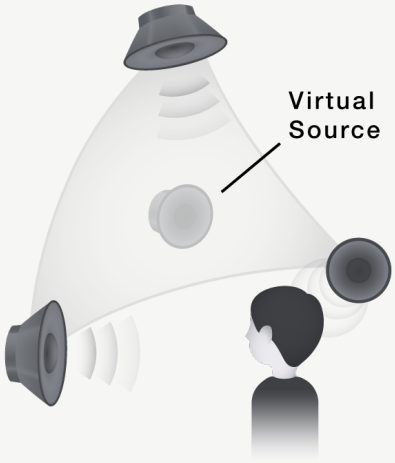

Figure 8 - 3D panning

Pros and Cons

Channel-based audio has its pros and cons. The main advantage is that it is the **perfect system** if we already know exactly the speaker layout on which the sound will be reproduced. For instance, if I know beforehand that my mix will be reproduced on a 5.1 system, with the speakers placed in those pre-established positions, that's great! We can then use the exact same conditions in the studio and make a perfect mix for it.

The main problem of channel-based audio is the **lack of flexibility**, since it is locked to a particular speaker configuration, and any changes to the output speaker layout will require a new mix. For instance, if a movie is released in Stereo, 5.1, 7.1, 11.1 and 22.2, then it will need 5 different mixes, one for each format.

This raises the question "Can we create a format that doesn't depend on the output channel layout? Can we create a format where a single mix is able to automatically adapt to different speaker layouts?". Indeed, and the answer is **Object-based** audio and **Ambisonics**.

Object-based Audio

Object-based audio started to get some attention with the release of **Dolby Atmos**. But what is object-based audio exactly?

Channel-based vs. Object-based audio

Before we get started on object-based audio, let's analyze how **channel-based audio** is created and distributed.

The basis of channel-based audio is to have a **pre-established speaker layout** - for example, stereo or 5.1, which means that we already know the layout in which the mix will be reproduced. So, we create a mix for those channels, which are then distributed (e.g. streamed, stored in a file, DVD). During reproduction it's only a matter of sending each audio signal to the corresponding speaker.

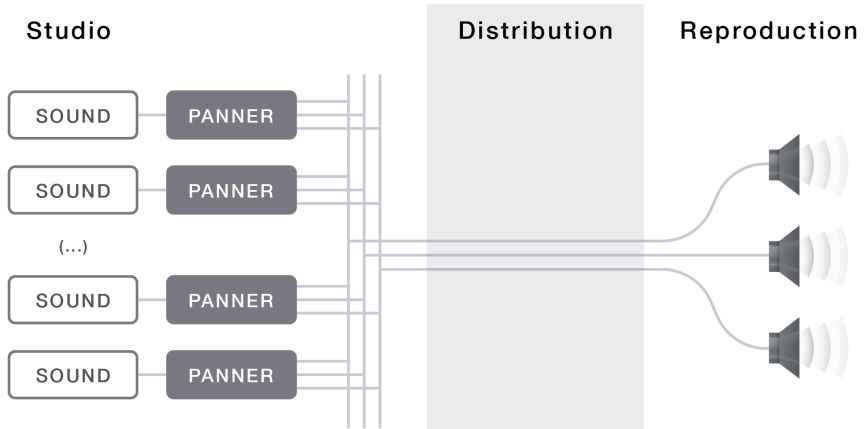

Figure 9 - Channel-based audio

In the studio, every sound passes through a **panner**, which will control how much sound should be placed on each output channel. For instance, on a 5.1 mix, if I want a sound to be positioned somewhere between center and right speakers, I use the panner to control it, which will place the correct amount of the signal on the center and right channels, but not on the remaining channels.

Also, the output of all panners is mixed together (using a **bus**) before being distributed: the left output of all panners is mixed and placed on the left channel, the same for the right channel, and so on.

Finally, the output signal is then distributed (DVD, stream, etc.) and reproduced later.

The idea of **object-based audio** is slightly different. Instead of mixing all sounds in the studio and distributing the final mix, object-based audio uses a different approach by distributing all sounds independently, which are **mixed only during reproduction**.

In the studio, you will still use a panner to position your sound, but you don't apply that panning information to the sound – you simply say where you want your sounds to be positioned. That information is distributed, and during reproduction, depending on your actual reproduction system, that panning information is actually applied to sound.

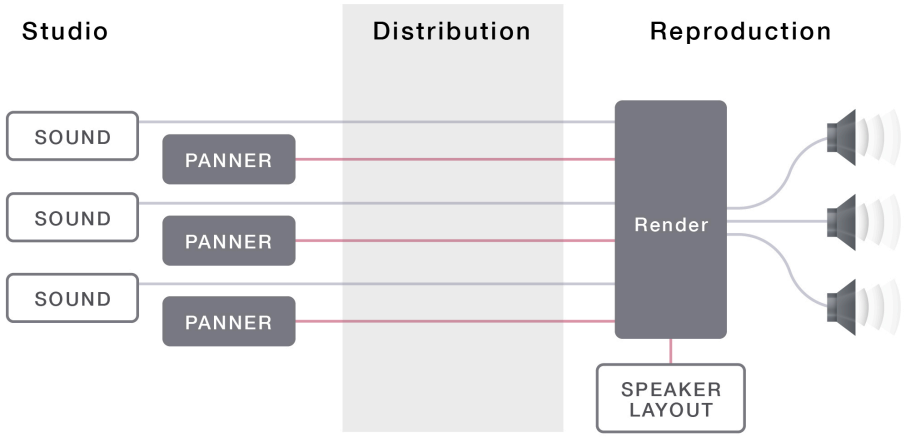

Figure 10 - Object-based audio

Essentially, you are **distributing your intentions** “I want this sound to be placed here, and this other sound to be placed there.” Then, during reproduction, the system knows how many speakers it has and their positions, and based on that, will make sure the correct signals are sent to the speakers to fulfill the intensions of the author, with everything being done in real-time.

Let’s see a practical example: imagine that you want to place a sound, once again, between “center” and “right”. With object-based audio, we distribute the audio stream of that sound, and also the desired panning position. Then, during reproduction and depending on the speaker setup that exists there, the sound is finally “rendered” to the exact speakers: in a movie theatre with 3 front speakers (“left”, “center”, “right”), the sound will be sent to “center” and “right” speakers, but if the theatre has 5 front speakers (“left”, “left center”, “center”, “right center”, “right”), then the sound will be sent to “center” and “right center”, or “right center” and “right”, depending if the sound is positioned closer to center or not.

Another example: imagine a car passing-by on the left side. With channel-based audio, the mix is entirely done at the studio. On a 7.1 system, the sound would eventually start at “left surround rear” and gradually move to “left surround side” and finish at the “left” channel. With object-based audio, if the movie theatre has 5 speakers on left wall, the sound would start at speaker 5, moving to speaker 4, moving to speaker 3, moving to speaker 2, moving to speaker 1, moving to left. But in a movie theatre with 8 speakers on the left side wall, it would do 8→7→6→5→4→3→2→1→left, taking advantages of the extra speakers to achieve a better space resolution.

Of course, when we say that the mix is only finished during reproduction, it is obvious that the re-recording mixer needs to hear what they are doing on the studio/dubbing stage, and they will have their own reproduction system that will reproduce their mix.

Objects

The panning information that is distributed for a particular sound is not a static value, but a value that **may change in time**. In this passing-by example, we only have one (mono) sound but its panning information changes through time. Also, besides the position of the sound, other data can also be taken into consideration. For instance, an object-based audio system could also use the concept of "**Size**" - is the sound to be almost a pin-point kind of sound (with only 1 or 2 speakers reproducing that sound) or do you want more spread, covering a much larger area (in the extreme situation, the sound could be reproduced by all speakers). Once again, on that passing-by example, instead of:

8→7→6→5→4→3→2→1→left

It would use something like:

8765→7654→6543→5432→4321→321left

Also, in some situations, the mixer may not be worried about the exact position of the sound, but much more interested in making sure that a particular sound is reproduced by a single speaker, telling the reproduction system to **pick the closest speaker** that exists near a particular direction, and reproduce the sound only on that speaker.

All this data that is distributed with each sound (e.g. position, size, etc.), is called "**Metadata**" (data about data). Each sound (and the corresponding metadata) is called an "**Object**", since it represents a (mono) sound object that moves through the 3D space.

With object-based audio, we are creating something that adapts to the reproducing system – a system with 10 speakers will be able to reproduce it, and a system with 50 speakers will also be able to reproduce it, taking advantage of the additional speakers to get a better space resolution.

Also, since the final mix only happens during reproduction, we can even allow the listener to **enable** or **disable** some audio objects or beds. This could be interesting in broadcast and an example is sports events, allowing the listeners to select if they want to hear the match with or without commentary, or even to select the sound according to their favorite team, by switching between mics positioned near the supporters of team A or near the supporters of team B.

Dolby Atmos

One of the most well-known systems using object-based audio is **Dolby Atmos** and therefore, we will explore it as an example of object-based audio. Nevertheless, there are other object-based audio formats, such as **MPEG-H**, **AuroMax**, **DTS:X**.

Dolby Atmos supports both object-based audio (**objects**) and channel-based audio (**beds**). An object will have its audio (mono audio clip) and the corresponding metadata, but Dolby Atmos also supports channel-based audio, by using **7.1.2** beds (7.1 + 2 height channels). Let's focus on objects first.

Dolby Atmos supports 128 independent channels, which means that we may have up to 118 audio objects plus 10 channels reserved for a 7.1.2 bed (more info on beds later on). During reproduction, Dolby Atmos supports up to **64 independent speakers**, which means that a movie theater may have up to 64 speakers which are controlled independently, meaning that each speaker can reproduce audio that is different from all other speakers. The room may even have more than 64 speakers, as long as some of them share the same audio signal.

A movie theater with Dolby Atmos support will be slightly different from a regular 5.1/7.1 room, as you can see in figure 11

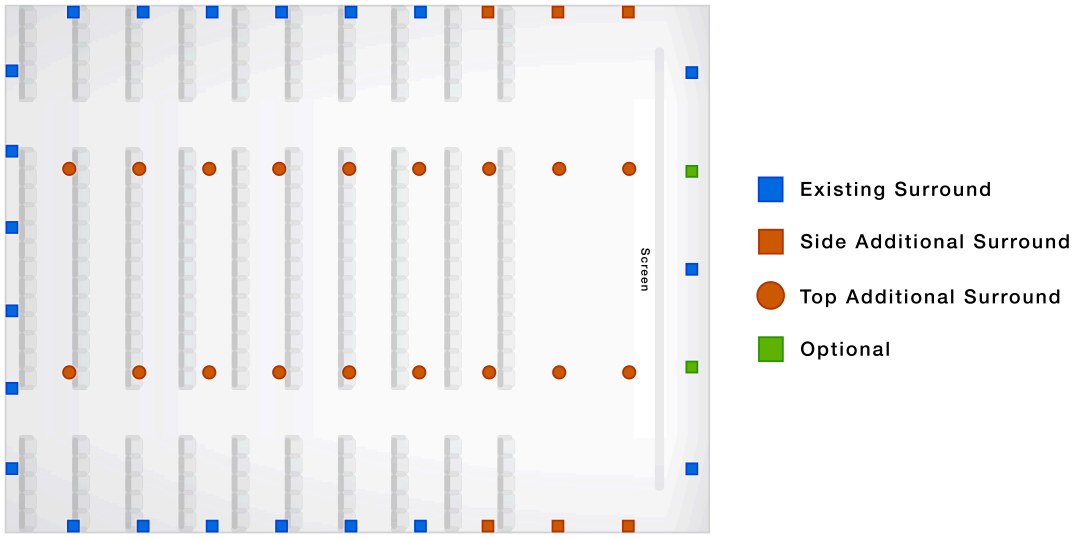

Figure 11 - Changes between a 5.1/7.1 movie theater and Dolby Atmos

The most obvious difference is the speakers on the **ceiling**. With Dolby Atmos, movie theaters start to have 2 rows of speakers on the ceiling, to start to explore the height component of sound.

In the **front**, we continue to have the “left”, “center”, “right” speakers, located under the screen, but allowing 2 optional speakers on rooms with larger screens, which are placed between center speaker and left/right speakers.

Also, up to 4 sub-woofers can be added to the room.

Finally, additional speakers are added on the **sides**, near the front. With traditional rooms, surround speakers usually start at 1/3 of the side length, occupying only the back 2/3’s of the side walls, which means that there is a gap between the screen and the initial surround speakers. With Dolby Atmos, that gap ceases to exist, and is filled with additional speakers, making sure that the entire horizontal plane is covered.

Beds

During the production of a movie, some audio content may have been already **pre-mixed**. For instance, the music score could be delivered to the re-recording mixers already mixed, or at least pre-mixed in stems (7.1 strings, 7.1 brass, etc.); or imagine a sound effect ambience that was already recorded or edited in 5.1.

Besides objects, Dolby Atmos also supports channel-based content which is called audio **beds**. This means, that besides objects, we can also include “regular” channel-based audio. Dolby Atmos uses 7.1.2 beds, which correspond to a traditional 7.1 (3 front + 4 surround channels + LFE) with 2 additional channels for the ceiling (one on the left side of the ceiling, and another on the right side of the ceiling). For instance, all existing 5.1 and 7.1 tracks can be directly mixed to the 7.1.2 bed. And even without using objects, a mixer can still explore the height component, by sending audio for the overhead speakers.

REPRESENTING LAYOUTS (E.G. 7.1.2)

During many years, we used a representation with 2 numbers to specify a layout format. For instance, **5.1** means that we have 5 main channels + 1 LFE. With the new immersive formats, we added a new number which refers to the number of channels that are on the ceiling. As such, **7.1.2** means 7 main (horizontal) channels + 1 LFE + 2 channels on the ceiling; or 9.1.4 which means 9 main (horizontal) channels + 1 LFE + 4 channels on the ceiling; and so on.

This is not a perfect system (some formats have a layer below the horizontal plane (on the floor), others may have 2 layers above the head, etc.), but at least it gives us slightly more information about the sound system.

But beds are not used only for pre-mixed material. Many movie productions use **hundreds of tracks** during the mix, and some even go above 1000 audio tracks - I still remember the first time I visited the post-production facilities at Universal, and in one of the dubbing stages someone said “this is our mixer, with 512 channels for sound effects and 256 channels for dialogue and music”. Although Dolby Atmos supports 118 objects, that would not be enough to handle all audio tracks. But beds allow us to overcome that, because we can **mix unlimited sounds** on the beds. With objects, each sound must be independent, as a way to control its panning, but with the beds that is not an issue, since we use “regular” panning to place the sounds around. As such, the re-recording mixer could use objects for the most important/ relevant sounds where accurate localization is more critical, and mixed secondary sounds in beds.

Dolby Atmos includes one 7.1.2 bed, but more beds can be added by reducing the number of available objects. For instance, we may have 118 objects + 1 bed, or 108 objects + 2 beds, or 98 objects + 3 beds, etc. But, once again, in terms of sound there isn’t any particular advantage on using more than one bed.

Pros and Cons

The main advantage of object-based audio is the **ability to adapt** to any speaker layout. By sending “intentions” to the reproduction system, we allow those intentions to perfectly adapt to an existing speaker layout. Also, and quite important, we are able to get the **maximum space resolution** out of the system – by adding more speakers, we are able to improve the space precision of the reproduction.

In terms of disadvantages, there are also a few things. First, we need to distribute a much **higher number of audio channels**. In movie theaters, Dolby Atmos may use up to 128 audio channels. But if we change to other mediums (streaming, blu-ray), those 128 audio channels are simply too much, forcing us to decrease the number of objects, which means that many objects must be rendered as beds, losing their advantages. Secondly, objects are **limited to a certain number**. Yes, 128 is a big number and most projects would not need them all, but in some situations that number is reached easily. For instance, Sound Particles is capable of rendering audio scenes with literally thousands of sounds (we have renders up to 1 million sounds playing at the same time). Also, some Hollywood productions use more than 1000 tracks. Yes, audio beds can handle most of those limitations, but is not a pure object-based audio approach.

CURIOSITY – DISNEY'S FANTASIA

During the production of "Fantasia", the Disney's movie from 1940 which includes the famous scene with Mickey Mouse with the Sorcerer's Hat, Walt Disney thought that it was important for audiences to have a better sound experience, due to the role of music in the movie. So, Disney and RCA engineers created "Fantasound", a multi-speaker sound system that was temporally installed in a few movie theaters (some systems even went on tour across the US), allowing audiences to watch the movie with a much more interesting sound experience. As such, Fantasia is considered to be the **first movie to use surround** sound.

There were several variations of the Fantasound system, and some of such systems included speakers on the ceiling, making Fantasia, not only the first movie to use surround sound, but also the **first movie using immersive sound**.

Technically speaking the system used 4 tracks: 3 tracks with recorded audio (3 independent sound signals) and one track with control signals, which would control how the 3 sound channels would be reproduced: at some point, channel 1 could be sent to the left speaker, channel 2 to the right speaker, and channel 3 to surrounds; and later on, channel 1 could be reproduced on the front speakers, channel 2 on the ceiling, and channel 3 on the rear; etc. Yes, it was an analog technology, but this makes Fantasia also the **first movie to use object-based audio**.

First **surround**, first **immersive**, first **object-based** audio movie, 80 years ago! There are clearly, people with a vision ahead of their time.

Ambisonics

The main goal of **Ambisonics** was to create an audio format that would be independent of the output reproduction layout – if you record something in Ambisonics, it wouldn't matter how many speakers you have during playback, or their positions, because Ambisonics would be able to adapt to it.

Although Ambisonics was created in the 70's (based on the work of Michael Gerzon), the interest in this format had a significant increase during the last decade, mainly associated with VR applications. But before discussing Ambisonics, let's understand a little better its simpler cousin: the Middle-Side technique.

Mid-Side

Although many people use the traditional **XY** stereo recording technique, where 2 cardioid mics are used – one pointing to the left side, and another pointing to the right side – some professionals prefer the Mid-Side recording technique.

Mid-Side is a widely used stereo recording technique, that consists on a cardioid microphone pointing towards the front (**Mid**) and a figure-of-eight microphone pointing to the sides (**Side**), as you can see in figure 12. With this approach, the Mid microphone will capture essentially what comes from the front. By pointing the figure-of-eight to the left side, we allow the microphone to capture sounds coming from both left and right side, but not sounds coming from perpendicular directions (such as front).

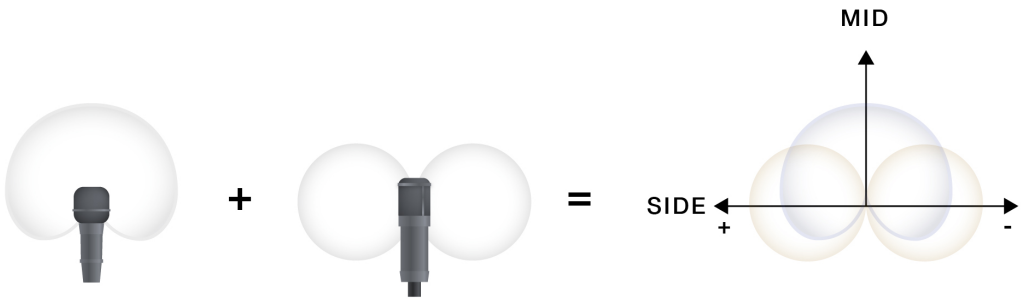

Figure 12 - Mid-side technique

As you can see, we have a microphone to capture the mono component (Mid, capturing the front), and a microphone to capture the non-mono component (Side, which doesn't capture the front).

What happens if we feed Mid-Side signals directly to two speakers? The speakers will still reproduce ordinary audio, but you will not have a stereo sound image. You need to decode it first to left/right to get the best result - a stereo sound image at the front.

To create the final stereo output, the signals from the microphones are mixed like this:

- Left signal is obtained by mixing both signals (Mid + Side)
- Right signal is obtained by mixing Mid with an out-of-phase version of Side mic (Mid – Side)

So, if you are using a Mid-Side pair and there is a sound coming from the left side, that sound will appear on the left channel (both Mid and Side mics capture that sound with the same phase), but will be very weak on the right channel (the Side mic signal will cancel the Mid signal, due to their opposite phase).

The same way that we **convert** Mid-Side into Left-Right, we can also convert Left-Right into Mid-Side[2], which means that we can convert back and forward between Left/Right and Mid/Side. Actually, this conversion is used a lot during **mastering**. Imagine that we need to apply a small compression to the main voice of a song, but you only have the final mix in stereo (no access to individual tracks). Instead of applying the compressor to the entire stereo signal, a mastering professional may convert the Left-Right into Mid-Side, apply the compressor only to the Mid channel, and convert the Mid-Side back to Left-Right. This way, the compressor is applied mainly to the sounds near the center.

Also, this Mid-Side conversion is used in some **effects units** to increase the stereo effect. If you convert a stereo signal (left-right) into a Mid-Side, and then increase the gain of Side channel, converting the result back to stereo, you will end up increasing the gain of all non-mono sounds, increasing the sense of space of the signal[3].

Ambisonics

Ambisonics is essentially a Mid-Side technique on steroids. Imagine that instead of using one figure-of-eight pointing to the sides (axis left-right), you also added a figure-of-eight mic pointing to the ceiling (axis top-down) and another figure-of-eight pointing to the front (axis front-back). And for the Mid signal, instead of a cardioid to capture the mono component, you replace it with an omni microphone, to have a full mono signal from around it. That is **Ambisonics**.

The traditional Ambisonics format, known as "**1st order Ambisonics** (**B-format**)" uses 4 audio channels:

- W – the omni component, which is as if I were capturing my scene with an omni microphone, so this channel has the audio coming from all directions.
- X, Y, Z – 3 channels corresponding to 3 figure-of-eight, each one pointing towards a different direction: Left-Right; Front-Back; Up-Down.

[2] For those that like math:
Left + Right = (Mid + Side) + (Mid – Side) = 2 Mid
Left – Right = (Mid + Side) – (Mid – Side) = 2 Side

[3] Of course, this will not work on a pure mono signal, because the Side channel will be empty.

When we talk about Ambisonics, and keep in mind that there are many different versions of Ambisonics, in general people are referring to “First Order, B-Format” (more info on this topic later).

And that’s it… Ambisonics is still **regular audio**, not some sort of mystic dark magic that makes 3D sound work. It’s simply audio channels: one with the mono component of everything, and then these 3 components: Left-Right; Front-Back; Up-Down. We can say that Ambisonics is a 3D version of Mid-Side as we have a mono component of everything and then these three components Left-Right; Front-Back; Up-Down as captured by figure-of-eight microphones.

The same way we cannot use the Mid-Side signals and reproduce them directly on the speakers (we can, but it won’t create the desired frontal image), the same happens with Ambisonics: we have 4 audio channels, we can hear their signals (its regular audio - not a weird noise-like signal) but we **need to “decode”** them before reproducing them on a speaker layout.

Ambisonics Microphones

One of the things that became clear from the beginning was that doing recordings with an Omni and 3 Figure-of-eight was **not very easy**, especially considering that capsules should stay as close as possible. Nevertheless, the same result can be obtained with a **tetrahedral microphone** with 4 cardioid capsules (as seen in figure 13).

Figure 13 - Tetrahedral microphones with 4 capsules: Sennheiser AMBEO VR mic (Courtesy of Sennheiser) and RØDE NT-SF1 (Courtesy of RØDE Microphones)

Using this 4-capsule setup - Left-Front-Up (LFU), Right-Front-Down (RFD), Left-Back-Down (LBD) and Right-Back-Up (RBU) - we can obtain 4 signals which are later on converted into the regular omni and 3 figure-of-eight channels. The signals that comes directly from the capsules of this tetrahedral microphone (LFU, RFD, LBD, RBU) is designated as “**A-format**”, while the traditional signals (omni + 3 figure-of-eight) are designated as “**B-format**”.

Nonetheless, through some math we can **convert** an A-format into B-format. For example, if we mix the signal from all capsules, we get the signal equivalent to the Omni. If we subtract the back capsules to the front capsules, we get the equivalent to the Front-Back, and so on.

- W = LFU + RFD + LBD + RBU
- X = LFU + RFD - LBD - RBU
- Y = LFU + LBD – RFD - RBU
- Z = LFU + RBU – RFD - LBD

Although capsules are close to each other, that small distance can still have some impact in some frequencies, because the signal doesn't arrive at all capsules at the same time. As such, some systems/plugins can do some even more advanced signal processing during this format A → format B conversion, trying to compensate for those minor effects.

Once again, although the sound is captured in A-format, it is important to convert it, **as soon as possible**, into B-format, since that is the standard format for Ambisonics.

USE AMBISONICS MICROPHONES, BUT DON'T FORGET ABOUT THE LESSONS FROM STEREO

Ambisonics microphones are a very interesting approach for recording audio, and they are able to record much more space information than other mono or stereo microphones. **But** it is important to remember 2 essential things.

First, sometimes people place too much confidence on the results of those microphones. Imagine a movie production that will be released in stereo. You wouldn't be too naive to the point of thinking that placing a single stereo mic on a camera would be enough, right? The same thing happens with 3D sound productions – **it's not only a matter of adding an Ambisonics mic** to a 360 camera. You will need to place mics close to the dialog (even if they are only mono), you will need to add better sound effects during the mix, you will need postproduction, etc.

Secondly, the engineers that receive GRAMMY awards for their fantastic orchestra recordings, don't use a simple XY stereo mic placed somewhere. Usually several microphones are positioned at different places, to use **time as a space tool**. Once again, for immersive recordings, Ambisonics microphones do a great job (and are definitely better than XY and MS stereo pairs), but if you want to excel, you probably need to use microphones spaced apart to also use time as a space tool.

High Order Ambisonics (HOA)

So, can we get an accurate 3D sound image using only 4 audio channels? Well, **not really**. Although Ambisonics does a good job reproducing a 3D sound environment using only 4 audio channels, its space resolution is not very high. This means that each sound will be slightly **blurred** in terms of direction. But **High Order Ambisonics (HOA)** can fix this.

With High Order Ambisonics (HOA), instead of the usual 4 audio channels of first order Ambisonics (FOA), we add more audio channels to increase its space resolution: a total of 9 audio channels in 2nd order Ambisonics (SOA), 16 channels in 3rd order Ambisonics (TOA), 25 channels in 4th order Ambisonics, 36 channels in 5th order Ambisonics, 49 channels in 6th order Ambisonics, and so on.

By adding more channels, we increase the **detail** of the sound image. These new channels will continue to carry audio signals, but with strange polar behaviors (see figure 14), instead of the traditional figure-of-eight diagrams of 1st order. Mathematicians call this **spherical harmonics**, because the same way that a sound can be decomposed in its harmonics, a 3D soundscape can be decomposed in its spherical harmonics.

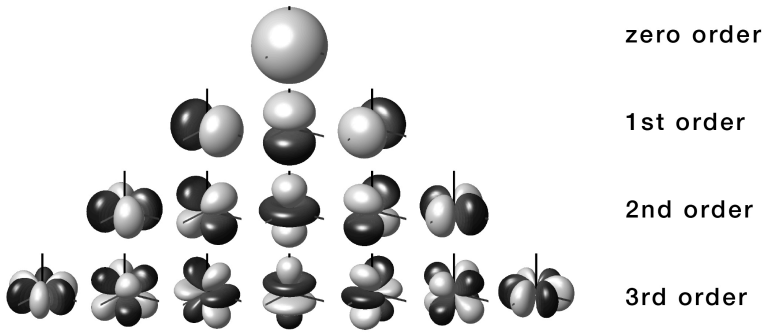

Figure 14 - Polar diagrams of the several Ambisonics channels (Courtesy of Franz Zotter)

Although we don't have microphone capsules that natively capture sound with these exotic directivity patterns, we can use special **microphones** with a high number of capsules, to derive such signal, such as the Eigenmike® (see figure 15).

Figure 15 – Eigenmike® microphone, capable of recording 4th order Ambisonics (courtesy of MH Acoustics)

Many Ambisonics Variations

Something very important to take into account when we talk about Ambisonics are its "**many flavors**". As such, we can easily make the mistake of using the wrong format or to interpret the format in the wrong way. Therefore, there are 4 important parameters that define which is the version of Ambisonics we are talking about:

- Order (1st, 2nd, …)
- Format (A, B, …)
- Component Order (ACN, FuMa, SID)
- Normalization (SN3D, N3D, FuMa, …)

The first thing to look at is what is the Ambisonics **order**[4]: is it First Order, Second Order, Third Order, etc. This is easy to identify because it depends on the number of audio channels we have:

1st Order	4 audio channels
2nd Order	9 audio channels
3rd Order	16 audio channels
4th Order	25 audio channels
5th Order	36 audio channels
6th Order	49 audio channels
…	…
nth Order	$(n+1)^2$ audio channels

The second thing to identify is the Ambisonics **format**, whether it's "B-format" or "A-format" (there are other formats, but they are very rare). The audio recorded directly from an Ambisonics microphone is "A-format" (each channel represents the audio from a particular capsule), but it should be converted[5], as soon as possible, to "B-format", which is the usual Ambisonics format, and the one most software knows and is expecting to receive.

[4] There is also the concept of a **mixed order**, but quite rare nowadays, which allows us to have more resolution at the horizontal plane, but less (or none) resolution on the height component.

[5] Your Ambisonics microphone should include a plugin to convert A-format into B-format.

The third thing to pay attention to is the **Component Ordering**, i.e., the order of the audio channels – unfortunately there are different ways of ordering audio channels inside Ambisonics, but the good news is that most of them are less common. Originally, channels were ordered in a particular way (W, X, Y, Z), but with the creation of High-Order Ambisonics, it was important to create an order of channels that would scale well when used in more complex scenarios with much more channels, so **ACN** order was born. For instance, with ACN your channel order for 1st order Ambisonics would be W, Y, X, Z; but with Fu-Ma will be W, X, Y, Z. Modern Ambisonics platforms use ACN channel order, but be careful when using older Ambisonics platforms, which could still use the old **Fu-Ma** order. There is also a **SID** approach, but almost inexistent. If you experience Ambisonics with wrong directions – sounds that should come from the left but are played at front, etc. – that is most likely an issue with the channel order.

The fourth important parameter is **normalization**, which refers to the gain of each channel. Once again, there are several approaches (**SN3D**, **N3D**, **maxN** / **Fu-Ma**[6]), but currently **SN3D** is the most common, because of one particular reason – with SN3D, the first channel (omni) is always the channel with the highest level, which means that if you are mixing or recording it, you only need to pay attention to the first channel in terms of clipping (if it doesn't clip, the other will not clip either). With other normalization approaches, such as N3D, it isn't granted that the omni channel was the one with the highest level, at least for all sound directions.

When asking for these parameters, someone may mention "**AmbiX**", which is a special audio file format, for Ambisonics, that uses B-format, ACN, SN3D. Sometimes people refer to AmbiX, not because of the use of that file format, but simply to say "B-format, ACN, SN3D".

So, be extra careful, and make sure you don't mix different "flavors" of Ambisonics.

VR Application

With Ambisonics it's easy to apply **rotations**. Imagine that you have your 3D scene, but you want to rotate it (e.g. 60° to the left, or 10° higher). With Ambisonics, rotation is as simple as applying a **matrix**, i.e., applying a formula that takes all existing channels and create the new rotated channels.

[6] The short name for Furse-Malham

This feature makes Ambisonics perfect for VR applications. Why? When you look around on a VR application, you are essentially rotating the camera: looking to the side (**yaw**), looking up and down (**pitch**) or tilting your head (**roll**). As such, it is easy for the VR application to take the Ambisonics signals and rotate them accordingly to the rotation of your head, obtaining a sound that is coherent with the image.

On top of this, Ambisonics is able to reproduce 3D sound using a small number of audio channels (4 audio channels), which is great for all platforms, especially for web and mobile. As such, Ambisonics had a significant boost in public interest with the rebirth of VR applications over the last years.

After rotation is applied, VR applications simply convert the Ambisonics signals to **binaural**, allowing users to listen to 3D sound using headphones. But we will talk more about binaural on the next chapter.

Pros and Cons

The biggest advantage of Ambisonics is its **flexibility** to adapt to any speaker layout. Secondly, it can reproduce a 3D audio scene using a **small number of audio channels** (starting with only 4 channel). And third, Ambisonics can **scale** to more channels to improve space resolution.

The disadvantage is that it requires a **high number of channels if you want a good space resolution**, because the traditional “first order” will have a reduced space resolution. As such, you probably need 3rd order (with its 16 audio channels) or above to really start to have good space resolution, and even higher (e.g. 6th order, with its 49 audio channels) if you want something comparable with object-based audio.

Binaural

When someone mentions immersive sound, most people automatically think of a room filled with speakers everywhere – in the front, the sides, above you, etc. Nevertheless, there is a special type of immersive sound approach that only requires a very simple setup: **headphones**. But can we really simulate 3D sound with headphones?

Well, the truth is that everyone is able to experience 3D sound using only **2 ears**, right?! I may be on a fully installed Dolby Atmos theater, surrounded by 64 speakers, listening to a fantastic sound experience, but in the end, my brain is only listening to two audio signals: my left and right ears. As such, it should be possible to reproduce a fully 3D sound experience using headphones, as long my ears continue to listen to the exact same signals as if I were on the actual room filled with speakers.

The reproduction of 3D sound using headphones is called **Binaural** audio, and it doesn't require any special type of headphones, being a fundamental tool for anyone working in immersive sound.

Perception

In order to understand how binaural sound works, we need to understand how humans perceive the direction of a sound. For instance, imagine a sound playing somewhere in the right side as seen in the image (figure 16).

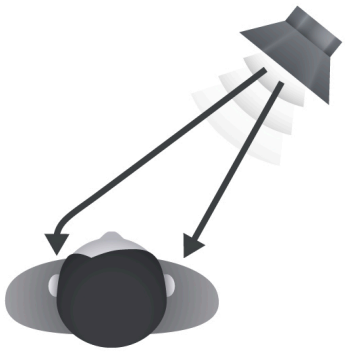

Figure 16 - A sound playing at the right side

The sound will be louder in the right ear and softer in the left ear, which is a fundamental clue for the brain to know that the sound is coming from the right side. Technically, we call this Interaural Level Difference (**ILD**).

Also, the sound will arrive first to the right ear, and slightly later to the left ear, which depending on the size of the head and the direction of the sound, could correspond up to 1 ms of delay. This difference of timings between ears is called Interaural Time Difference (**ITD**).

But these two behaviors don't explain everything. For instance, how do we detect the difference between front, top or rear sounds? In these 3 situations, the sound arrives with the same levels and at the same time to both ears.

Well, we have some special, highly designed accessories, called ears. Depending on the direction of the sound source, the sound will bounce on different parts of the outer ear (**pinna**), creating minor reflections, that help the brain identifying the direction of sounds. We know that when we mix a sound with a delayed version of it, some frequencies will be reinforced while others will be attenuated. As such, the ears will act as a small **equalizer** that changes their frequency response depending on the direction of the sound. The reflections from the **shoulders** area will also complement the reflections of the outer ear, once again making a smaller change on the frequency response of the arriving sound.

Finally, there will be also a small **shadow effect** on the opposite ear. Higher frequencies, due to their smaller wavelength, will have more difficulty to "bend" and enter shadow areas, which means that a furthest ear will also have a slightly decrease on the high frequencies.

HRTF

All of these factors help our brain to detect the direction of a sound, and as we can see, depend mainly on gains, delays, and filters. So, the next question is, how can we capture (or represent) the overall response of a particular sound direction arriving at a human head?

The same way that we use impulse responses to capture the reverb of an existing room, which is made of reflections (filtered reflections to be more exact[7]), we can use the same concept to capture the impact of a particular sound direction at a human ear. These impulse responses are called Head-Related Impulse Responses (**HRIR**), which are also known as Head-Related Transfer Functions (**HRTF**). HRIR's and HRTF's are the same thing. The only difference is that HRIR's are on the time domain (like a waveform), and HRTF's are at the frequency domain (like a spectrogram).

During the years, several scientists have created collections of these impulses, called **HRTF datasets**, by using two approaches:

- using miniature microphones placed on the ears of real people, one capsule in each ear;
- using dummy heads, with or without torso, with accurate representations of ears, and microphone capsules located at the internal position of the ear (see figure 17);

Using these approaches, scientists will then capture the left and right impulse response of a series of different positions around the head. Based on those datasets, software developers can then create binaural plugins and similar software (game audio engines, etc.) to create the illusion of a sound coming from a particular position.

Figure 17 - A binaural microphone (Neumann KU 100, courtesy of Neumann)

[7] For instance, concrete will reflect sound with a frequency response that is different from a reflection from a carpet.

For instance, if I have a sound of an helicopter (in mono) and I want the listener to feel that the helicopter sound is coming from a specific direction, all I need to do is to use a HRTF dataset and find a recorded position that is close to the direction that I want (or even interpolate between positions of the dataset). Then I take the pair of impulse responses of that position (a left impulse response and a right impulse response) and apply them to the helicopter sound, creating 2 sounds: the left sound (the left impulse response applied to the helicopter sound) which will be sent to the left side of the headphones, and the right sound (the right impulse response applied to the helicopter sound) which will be sent to the right side of the headphones. By the way, the act of applying an impulse response to an audio signal is called **convolution**. With this, listeners will perceive the helicopter as coming from that direction.

Although most people use binaural techniques during post-production or reproduction, it's perfectly possible to do **binaural recordings** as well. Once again, if I place miniature microphones on a person's ears[8], or if I use a dummy head (like the one in figure 17) placed on the audience on a concert hall, I will be able to record audio natively in binaural. Then, by sending those 2 audio channels to headphones, you would be able to listen to the same 3D audio scene as if you were there.

HRTF individualization

So, does it mean that we are able to get realistic 3D sound over headphones for everyone? Unfortunately, **not for everyone**.

Each person has a different head, different ears, and other physical aspects that are **unique** to each one of us, which means that there will be slight variations on the way each person's ears will affect the arriving sound. And each brain has spent many years listening to audio with those personal characteristics, not others.

With speakers we don't have that issue. The sound comes out of the speaker, and although each person's ears will hear it on a slightly different way, that slightly different way is the regular way for that person's brain (the sound coming from that direction always sounded like that). The problem with headphones with other people's HRTFs is that the brain listens to sound in a way that is slightly different and uncommon, and that may seem strange.

[8] These microphone capsules should be placed as internally as possible, otherwise, they will not capture the true response of the ear.

If I listen to some binaural audio, which was created with an HRTF dataset captured from someone else, I can either have a great experience with a perfect 3D sound, or, my brain could have a difficult task, trying to interpret **conflict information** due to the physiological differences between me and the original HRTF person.

For instance, for the same position (front-center), figure 18 shows the differences on the sound that reaches the inner ears of different individuals. As you can see, the differences between different people are quite huge, even above 30 dB in some cases! Some have big dips at 6 kHz while others have those dips above 10 kHz.

If you apply an HRTF that is close to the way your ears behave, the brain will acknowledge that - “I know this equalization! It happens every time the sound comes from that direction” and everything works as it should. Other times, the HRTF that you are using is somehow closer to the way your ears behave, but from a different direction - “This equalization looks familiar… I think the sound is coming from that direction”, which gives you a wrong information - for instance, someone applies a HRTF from a frontal sound, but the brain acknowledge that as being at the back (which is called front-back confusion), or at a different elevation. Some other times, the HRTF is so different from what your brain knows, that it becomes confused - “I don’t detect any familiar equalization”, and your brain doesn’t give it any particular direction, and you feel that the sound comes from inside your head.

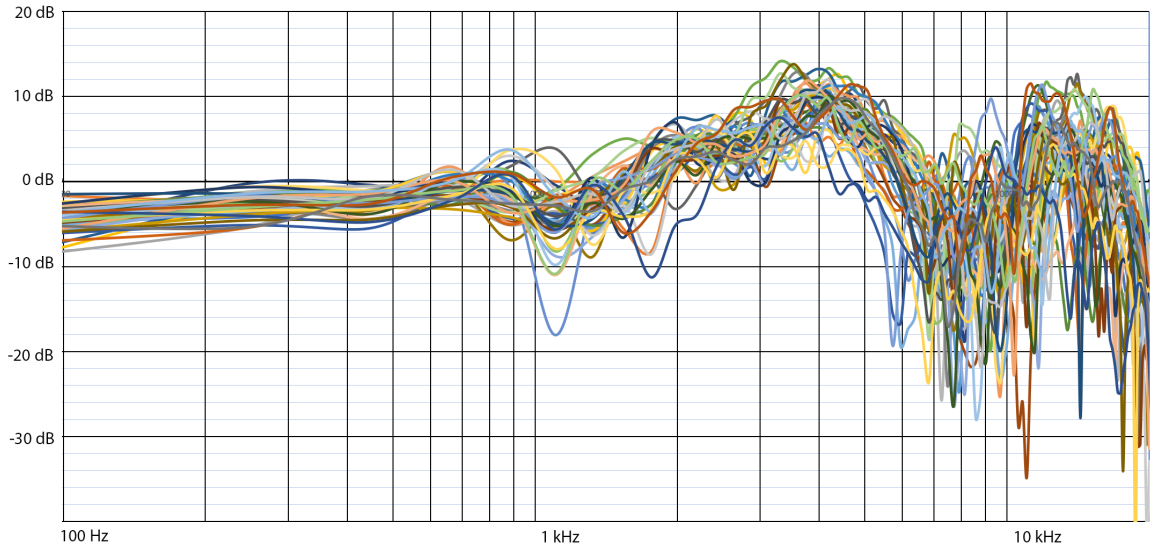

Figure 18 - The HRTF of the left ear of 45 individuals for the exact same position: front center (azimuth = 0, elevation = 0)

Once again, for some people the perception of a particular binaural sound will be tremendously realistic, for others somehow realistic (but still better than stereo), and for some the sound won't be realistic at all. For instance, when I listen to binaural audio, most of the times, all front sounds seem elevated, probably because my ears are slightly different from the common ones.

Of course, many companies try to choose the best available HRTF dataset to use on their products, which means choosing the HRTF dataset that satisfies the biggest percentage of listeners. But, even with a fantastic HRTF dataset, there will always be a **percentage of listeners that will not be able to fully enjoy** the binaural audio with that dataset.

This means that we need some sort of **HRTF personalization**: or by allowing the listener to **choose** the best HRTF dataset among several choices (choosing the one that was recorded with someone with similar physical characteristics) or having access to a **personal HRTF**. Of course, only a few are lucky enough to capture their own HRTF on a lab, but some companies are offering HRTF personalization, based on photos, videos or even 3D scans of a person's head. For those cases, where a personalized HRTF dataset exists, there's something called **SOFA** which is a file format defined by AES (Audio Engineering Society) that allows a person to have their own HRTF file and use it on a system that supports it.

Using Binaural

It's important to refer once again, that binaural audio only works if reproduced with **headphones**. If you reproduce its audio with stereo speakers, you may eventually get an acceptable stereo sound, but not a 3D experience, or anything close.

The second important thing is that all the different approaches that we mentioned earlier – channel-based audio, object-based audio, and Ambisonics – can be **converted to Binaural**.

For instance, you can convert a 5.1 or a 22.2 signal into binaural, or render Dolby Atmos in binaural, or convert a 3rd order Ambisonics signal to binaural. Actually, this Ambisonics to Binaural conversion is what happens in most VR applications. Although VR could be experienced with speakers, in most cases the Ambisonics audio signal of a VR scene is converted to binaural and delivered to the headphones of the VR headset.

SOUNDBARS

Binaural was created to reproduce 3D sound with headphones, but a lot of research was done to try to recreate it over speakers, and many sound bars use some of those concepts, allowing people to experience 3D sound by using a sound bar on the front.

Most sound bars are able to reproduce sound that is more immersive than regular stereo, but worse than using a full speaker layout, especially for people not sitting on the absolute sweet spot. Nonetheless, it could be an interesting alternative for people that want a 3D sound experience but cannot install a full 3D speaker setup.

Pros and Cons

The biggest advantage of binaural is the ability to reproduce 3D sound over a **very simple setup** – using headphones - no need for a huge number of speakers; you can take your 3D sound experience wherever you go; cheap and efficient setup.

The biggest disadvantage is that it will not work as it should for many people, due to the lack of individualization, which will result in **front/back confusion**, **elevated sound** (especially at front) and a sense of sound "**inside the head**". And second, it doesn't support **speakers**.

Once again, binaural is a great approach, as long as people understand its needs in terms of individualization.

Conclusion

We have seen 4 different approaches regarding 3D sound: **Channel**-based audio, **Object**-based audio, **Ambisonics** (aka scene-based audio), and **Binaural**. Each one of those approaches has advantages and disadvantages, and there isn't a single one that outperforms the others in all situations, which means that you need to choose it carefully.

Channel-based audio is perfect if you have a fixed speaker layout but is not flexible for multiple layouts. **Object**-based audio gives you the best space resolution with flexible layouts, but you need to distribute each audio stream separately, which could be an issue if you have many sound sources. **Ambisonics** does a nice job reproducing 3D sound with a small number of output channels, but if you want good space resolution you will need a high number of channels. **Binaural** is the approach for headphone reproduction, but it doesn't work with speakers and we have the individualization problem. And, let's not forget the **hybrid** approaches, which try to mix the best of two worlds. For instance, Dolby Atmos used both Beds (channel-based) and Objects (object-based), and most VR applications use both Ambisonics and Binaural.

By now I hope that you have a better understanding regarding 3D sound: how it works and how we can use it; what are the different systems and what they are used for; and understand better this world of 3D audio that is now exploding everywhere from cinema to TV, from videogames to music, from VR to many other applications. These are wonderful times for everyone that loves spatial audio… and you can always use **Sound Particles** software as your playground, which supports most of these formats.

SOUND PARTICLES

Sound Particles is the **Ultimate 3D Audio Software** and has been used by all major **Hollywood** studios in productions such as **Game of Thrones, Star Wars 9 and Frozen 2**

With virtual microphones, you are able to render audio in all formats, from mono to 22.2, from 6th order Ambisonics to any custom layout.
Sound Particles is Hollywood's secret weapon to create epic sound design. One of its unique features, is the ability to use particles systems to generate thousands of sounds around you.

Do you want to create a "battlefield"? Simply create a particle group with 10.000 particles (sound sources), import 200 war-related sound effects from your sound library, add random movements and random audio effects, and capture the result with a virtual microphone – the entire process in a few minutes.

Try the unlimited time demo now

Sound Particles is free for teachers, students and schools

Sound Particles Lda
IDDNET, Rua da Carvalha, 570, 2400-441, Leiria, Portugal
www.soundparticles.com • info@soundparticles.com • +351 244 859 465