Spatialization Overview

An overview of spatialization in Unreal Engine.

Introduction

In audio, the simulation of sound spatialization, also known as sound localization, is achieved using a wide variety of audio technologies working together. These technologies mimic natural sound phenomena such as orientation, attenuation, propagation, occlusion and obstruction, and reverb.

These phenomena are explained below:

Phenomena	Description
Orientation	The relative orientation between the listener and the sound source.
Attenuation	The distance between the listener and sound source.
Propagation	The paths a sound travels from its source to the listener.

Phenomena	Description
Obstruction and Occlusion	A sound may collide with other objects while it travels towards the listener. These objects act as obstructions to the sound.
Reverb	A sound may take different alternative paths to reach the listener, creating a perceived echo effect.

In the Unreal Audio Engine, and in game audio, the term "spatialization" usually refers to only one aspect of the overall generalized spatialization problem: the techniques and technologies used to orient the sound relative to a listener. Other considerations, such as attenuation, reverb, and occlusion, are usually discussed in that same context. This overview will use the term "spatialization" to exclusively refer to methods of orienting a sound relative to a listener.

The three main methods of spatialization are **panning**, **soundfield spatialization**, and **binaural audio spatialization**.

Panning

Panning is the oldest and simplest way to simulate spatialization. In its most basic form, panning can be achieved by adjusting the relative gain between different audio channels (speakers). Gain is a generic term used to describe the operation of either attenuating (gain < 1.0) or boosting (gain > 1.0) a sound.

For a given sound, making its gain higher in the left channel compared to the right channel will create the illusion that the sound is coming from the left. This is called stereo panning, where "stereo" means that the audio has two channels. It may seem obvious that this would produce the feeling of spatialization, but it's important to note that this illusion is created because the brain has been trained to understand that a sound to your left will sound louder in your left ear compared to your right ear. Technically speaking, this psycho-acoustic experience is referred to as the interaural level difference, where "level" is another term often used for volume.

For speaker setups that have more than two channels, such as quad, 5.1, and 7.1 surround sound, the technique for panning is the same as it is for stereo panning, but it's applied to pairs of speakers instead. When describing panning with more than two speakers, this is known as pairwise panning.

For speaker setups that use more than two channels and have speakers above and below the listening plane, the technique for panning is similar to pairwise panning. However, instead of adjusting the relative gain between two paired speakers, gain is adjusted between speaker triplets (i.e. triangulation). This type of panning is known as vector-based amplitude panning or VBAP.

Determining the Panning Value: The Listener and Speaker Geometry

Before discussing different panning techniques, is it useful to understand how the panning value is calculated based on game geometry. There are two main geometries to consider when computing the pan values: the geometry of the listener, and the geometry of the physical speaker locations.

Listener Geometry

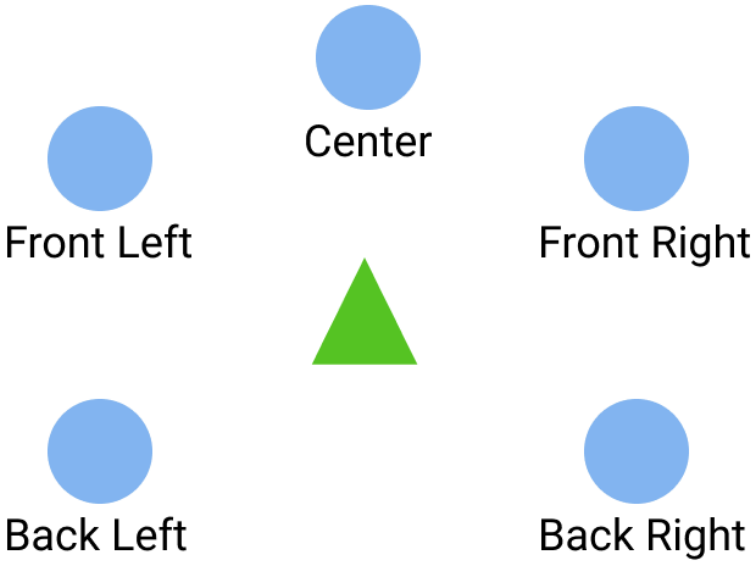
All spatialization techniques depend on the concept of having a listener. The listener is the location and orientation where a virtual listener is maintained. It can be stationary or manually controlled in a 3D environment.

Typically, the listener is attached to the player camera where it's assumed that the "eyes" and "ears" of a game viewport ought to be in the same location. Although this is a reasonable assumption for most situations, there can be more unusual setups depending on the game type or gameplay requirements. For example, a character might have an ability that allows them to hear things from different locations. Another example can be seen in third person games, where designers often split their listener geometry: the location of the listener is set on the character for purposes of distance attenuation, and the camera is used for purposes of panning.

Speaker Geometry

Speaker geometry involves the actual physical arrangement of speakers in the player's environment as they experience the game. For headphones, the arrangement is simple and fixed: speakers are immediately to the left and right of the listener. If the green triangle is the listener's position and orientation, the geometry of headphone speakers is the following:

For surround sound speakers, however, the arrangement is more complicated and depends on the number of speakers and the standard used to position them. The following is a typical arrangement of speakers in a 5.1 surround sound setup (the subwoofer is not pictured):



There are many standard speaker layouts and channel configurations. Typically, games make an assumption about speaker locations based on the standard configuration selected. There are methods of automatically determining speaker locations using noise bursts, sound tones, and microphone analysis, but this is almost never done for video games.

The Center Channel

The center channel is often ignored during pan calculations because it is usually reserved for non-spatialized audio, such as dialogue or interface audio.

This is typically done in cinematic mixing where important dialogue uses the center channel to ensure it is heard clearly. Since dialogue has a narrow spectral band and low power requirements, this has caused many sound systems to underpower their center channels.

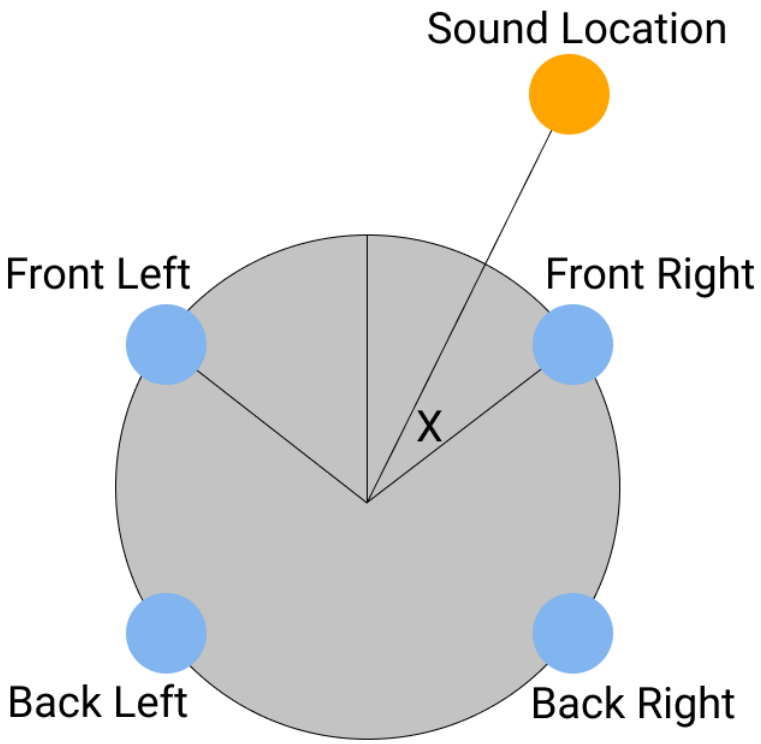
Games usually opt to disable panning through the center channel for this reason, and mostly use it for dialogue or other important audio, such as interface audio.

Computing Pan Values For Mono Sources

Given the listener and speaker geometries, a sound's relative location in the world can now be calculated.

Simplifying the speaker arrangement to ignore the center channel in a 5.1 speaker arrangement, we can conceptualize the speaker arrangement as virtual locations within a game world. These locations are relative to the listener's location in the world.

The pan calculation can then be described by the following diagram, where X represents the angle between the line that connects the listener's location to the right speaker, and the line between the listener location and a sound in the world.



If the angles of the speakers relative to the listener are known (or chosen according to a standard), this derived angle X can be used to compute a pan parameter that can be used in the linear pan algorithm or the equal power pan algorithm (see the **Performing Audio Panning** section below).

Pan_{Left} = X / TotalArcLength
Pan_{Right} = 1.0 - Pan_{Left}

For example, if the angle X is determined to be 15 degrees, and the total arc length between the left and right speakers is 60 degrees (30 degrees to the right and 30 degrees to the left from center), the pan parameter will be determined to be 0.25 for left and 0.75 for right.

To continue this as the sound pans around the listener, the algorithm only changes by determining which pairs of speakers need to be used to calculate the pan value. The azimuthal arrangement of each of the speakers would be used in the same way to determine the left and right pan parameters.

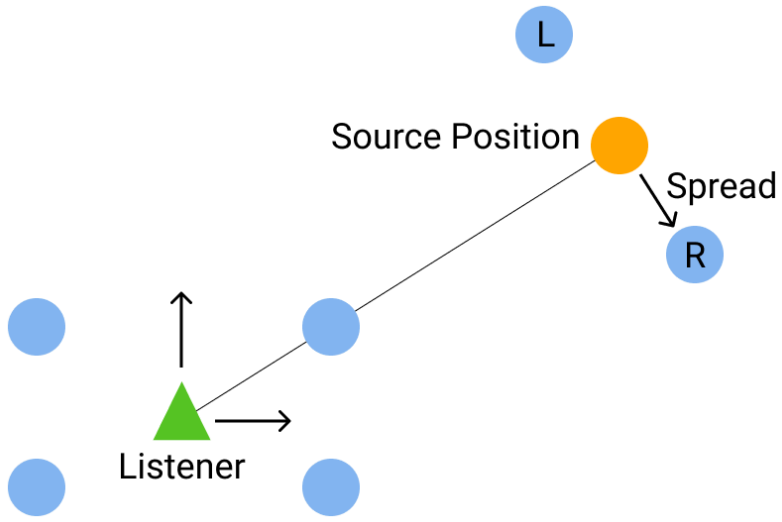
For non-azimuthal speakers (height speakers), the algorithm is similar but requires three pan parameters instead of two.

Computing Pan Values for Multi-Channel Sources

Panning is computed for multi-channel sources in a similar way to mono source panning. The difference is that, for each of the channels in the source file, a unique panning matrix (how much gain to apply to what output channel) is computed.

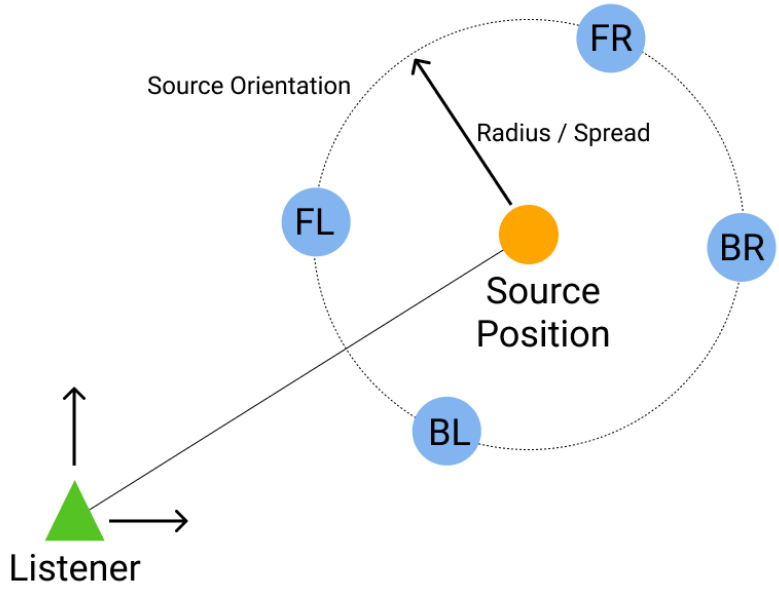
For stereo sources, each left and right channel is treated as a mono source. A spread parameter is used to determine the geometry to use for each of the source channels position. This parameter is defined as the actual spatial distance (or half-distance) between each left and right channel.

Note that the orientation of the left and right channels relative to the listener needs to be decided. Typically, for stereo spatialization, the left and right channels are oriented such that they are always orthogonal to the vector pointing from the listener to the source's position.



For channel sources higher than stereo, which are typically even-value counts (4, 6, or 8), a similar algorithm can be employed. Instead of the spread parameter being defined as the distance (or half-distance) from the left and right channels, they are defined as the radius of

a circle centered on the source's position. In this case, the source channel virtual points are typically points on the edge of the circle, spread equally.



For higher channel spatialization, since there is no clear source orientation preference (like there was with stereo sources, where we can just prefer they maintain orthogonal orientation), we need to also supply a separate vector from which to orient the channel positions. Usually, the source's own orientation vector is used to orient the source channel positions.

Performing Audio Panning

There are two main methods of performing audio panning: linear panning and equal power panning.

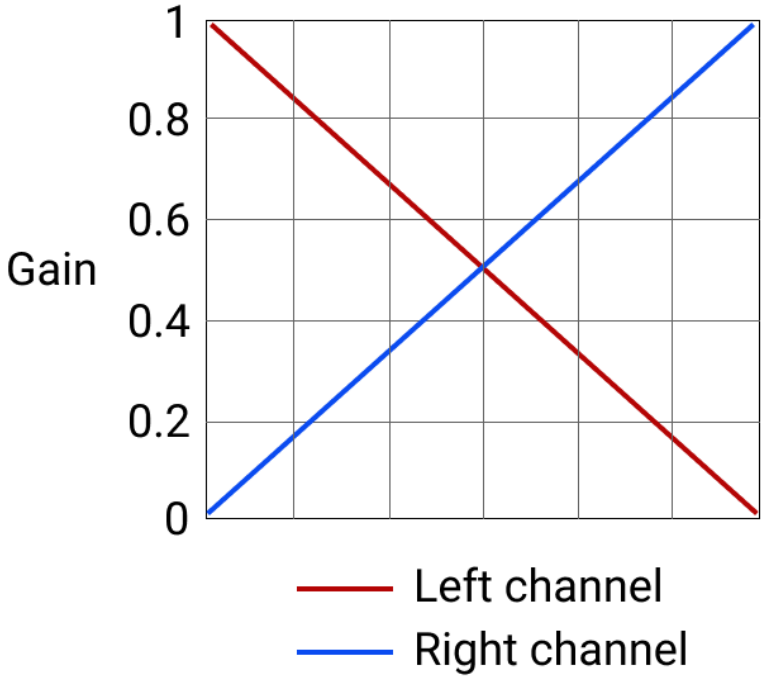
Linear Panning

Linear panning is the simplest panning technique where the relative gain of the sound between paired channels is linearly interpolated and the total gain is held constant.

When using two channels (left and right), the following gain computation equations are used:

$Gain_{Left} + Gain_{Right} = 1$
$Gain_{Left} = X$
$Gain_{Right} = 1 - X$

Linear panning, visualized, looks like like this:



Linear panning is straightforward to compute but it has a fundamental drawback: loudness (volume), which is the term used to describe the perceptual experience of sound amplitude, is not determined by the actual amplitude of an audio signal but by the power of the audio signal. Power is equal to the square of the signal's amplitude. If the amplitude, or gain, of a sound is X, the power for panning calculations is computed by the following equations:

$$Power_{Left} = Gain_{Left}^2 = X^2$$
$$Power_{Right} = Gain_{Right}^2 = (1 - X)^2$$

And the total power is computed by adding up the computed power in the left and right channels:

$$Power_{Total} = Gain_{Left}^2 + Gain_{Right}^2$$
$$Power_{Total} = X^2 + (1 - X)^2$$

Using the gain values with this equation demonstrates that the power is not constant. The power, and thus the loudness, will drop when the sound is panned in the middle (X = 0.5) vs. the sides (X = 0.0 or X = 1.0).

The following is the power, or perceived loudness, in the center of a pan between left and right channels:

$$Power_{Total} = (0.5)^2 + (1 - 0.5)^2$$
$$Power_{Total} = 0.25 + 0.25$$

$Power_{Total} = 0.5$

While the following is when the sound is either to the right or the left:

$Power_{Total} = (1)^2 + (1 - 1)^2$
$Power_{Total} = 1 + 0$
$Power_{Total} = 1$

This change in power during panning will affect the game's overall audio experience, as the sound will appear to move through the speaker field non-linearly.

For stereo (headphones or speakers), the sound will appear to "stick" to the edges. For surround sound panning, as a sound crosses over a speaker location, it will get loud and then dip in loudness between speakers. This creates uneven loudness as a sound rotates or moves around the listener.

Since gain reduction is a primary method for the simulation of distance attenuation, this gain reduction can also cause sounds to be interpreted by the brain as moving closer and further away from the listener, even though the audio engine's intention is that the sound is merely moving around the listener.

Equal Power Panning

The equal power panning method counters the loudness dips present in linear panning by boosting the loudness perfectly to maintain constant power. This technique retains constant loudness by holding the power constraint during the pan, instead of holding the amplitude constant.

Returning to the total power equation, we solve for X where the result is that the total power is constant for every value of gain, X (i.e. 1.0).

$Power_{Total} = Gain_{Left}^2 + Gain_{Right}^2 = 1.0$

There are several possible solutions to this equation. In audio, a square root solution (called the square-root panning law) or a sine / cosine equation (called the cosine panning law) can be used.

If X is a panning parameter number between 0.0 and 1.0 (where 0.0 is fully left and 1.0 is fully right), the following gain computation equations are used to maintain equal power.

For the square-root panning law, the left and right channel gains are computed as:

$Gain_{Left} = \sqrt{X}$
$Gain_{Right} = \sqrt{(1 - X)}$

Where it can be seen that the total power is always maintained constant:

$Power_{Total} = (\sqrt{X})^2 + (\sqrt{(1-X)})^2$
$Power_{Total} = X + (1-X)$
$Power_{Total} = 1$

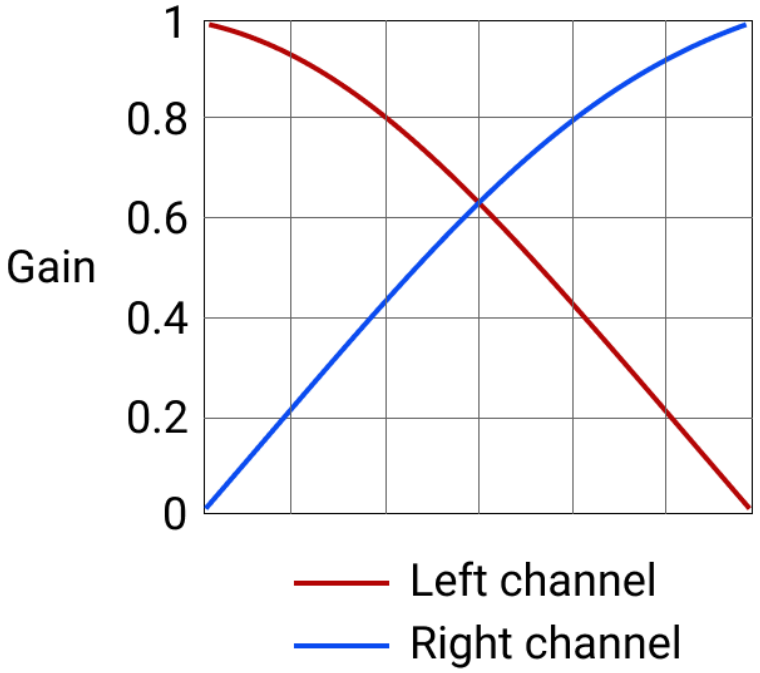
For the cosine panning law, the left and right channel gains are computed as:

$Gain_{Left} = Sin^2(X)$
$Gain_{Right} = Cos^2(X)$

To see that this also satisfies the requirement to maintain constant power, recall the trigonometric identity that defines a unit circle (a circle of radius 1):

$Sin^2(\theta) + Cos^2(\theta) = 1$

Graphing what this panning law outputs, it can be seen that the actual gain values are not linear:



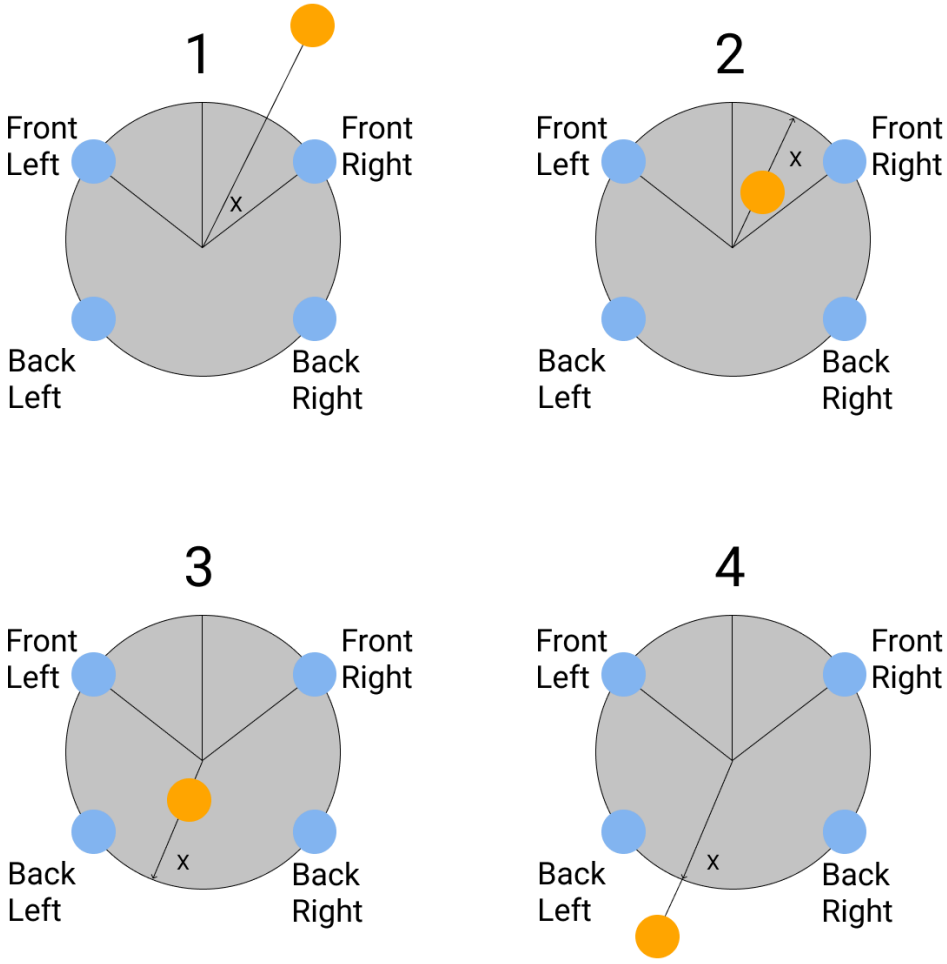
For panning calculations involving a third dimension, such as height speakers, the equal power constraint would be calculated using the following equation:

$Power_{Total} = Gain_A^2 + Gain_B^2 + Gain_C^2 = 1.0$

Note that most game audio engines forgo this calculation when panning to height channels and instead use object-based rendering methods, such as Dolby Atmos and DTS:X, which are rendered outside of audio engines. Audio rendering for game engines becomes significantly more complex and computationally costly the larger the number of channels used.

The Problem of Vector Flipping

One issue with the panning algorithm is the case where a moving sound source traverses across the listener, such that the vector pointing from the listener to the position of the source instantly changes direction.



Since this vector is primarily used to determine gain values in a speaker array, it will result in a sudden change in gain from one moment to the next, as the sound moves from one side of the listener to the other. In the above diagram, between cases 2 and 3, the blue source moves across the listener origin and the vector which is used to compute gain values jumps from pointing forward to pointing behind. This "vector flip" will cause a discontinuity in the

audio signal, which is perceived as a click or pop and is a classic edge case many engines which utilize 3D panning will fail to account for.

The primary method of avoiding this discontinuity is to utilize an omni-directional blend to all speakers or a portion of speakers to "smooth out" any discontinuities that might happen due to this vector-flip. Typically, there will be a user-defined radius below which the sound source will blend to an omni-directional pan. This removes the ability to localize the sound during this transition.

When panning sound sources that are within the virtual radius of the speaker configuration, there is no way to spatially differentiate (via gain values in speaker channels) a sound source between cases 1 and 2 and between cases 3 and 4. Thus "near-field" panning is traditionally problematic.

Soundfield Spatialization

Another spatialization technique ideal for speaker-based spatialization is to use a spherical harmonic representation of a sound field. The preferred term used to describe this method is "sound fields", but they are also widely known as "ambisonics".

Spherical harmonics are a wave-based way to represent vibrations of a field in three dimensions (spherically). They are used in many domains where an arbitrarily high resolution of spatial information is necessary.

Conceptually, spherical harmonics are a spatial equivalent to the Fourier theorem, where any periodic function can be represented as a series of sinusoids of various frequencies and amplitudes. In this way, spherical harmonics can be thought of as a series of functions which can precisely define a three-dimensional field.

Practically speaking, the order of a spherical harmonic (or soundfield / ambisonics) representation describes how well-resolved the spatial representation is. The higher the harmonic order, the more terms in the spherical harmonic expansion are included, and the more well-resolved (sharper) the spatial representation is. However, the more orders used, the more expensive they are to compute and the more memory is required to represent them.

The benefits of using sound fields over panning is that they tend to have a significantly better result in localizing sound sources between speaker positions. Where the quality of panning

methods increases linearly with the number of speakers used, sound fields can achieve very high quality results with very few speakers.

Another significant benefit of sound fields is that they encode and preserve their spatial information independent of speaker or channel configuration. They retain their spatial information in a channel-agnostic way under rotational transformations, mixing, and so on.

3D sound sources can be encoded to a sound field representation, mixed with other sound field formats, and then, at the last step in the rendering pipeline, decoded to match a precise output channel format. This means that audio content can be pre-baked in a sound field format and decoded locally to a particular local player's hardware configuration that is ideal for their listening environment. Without this ability, multi-channel sound sources need to be downmixed (e.g. 7.1 content downmixed to stereo) to fit a local hardware configuration. This downmixing process inevitably loses important spatial information. Sound fields, on the other hand, would merely decode to whatever channel configuration exists.

It's important to note that sound fields on their own can't be directly listened to in a way that would be identified as spatial. They require a decoding step to a traditional speaker / channel format, where the sound field is essentially resolved to a set of channel gains that are applied to channels similar to the way panning works. Interestingly, sound fields will usually result in non-zero gain values that are spread around a surround-sound channel configuration even though they will audibly sound like they are spatialized in any given direction.

Finally, it's possible to capture a real-world sound field using particular microphone configurations which capture audio in a way that is directly encoded into a sound field format. Usually, sound field microphone arrays record sound fields of lower order due to hardware and microphone cost, but expensive, higher-order sound field microphone arrays do exist. This technique makes sound field recordings of environmental audio particularly powerful as a technique to gather complex and spatial source material.

Binaural Audio Spatialization

If the audio is designed for headphones, then a variety of psycho-acoustic phenomena can be taken advantage of to increase the quality of spatialization. In audio technology, it's more common to hear this type of spatialization referred to as binaural spatialization (as opposed to headphone spatialization). The term "binaural" comes from the fact that humans (and most animals) have two ears. It is the acoustical equivalent to the term binocular.

There are three main components to the way your ears physically localize sound binaurally: **interaural level difference**, **interaural time delay**, and **spectral shadowing**.

Interaural Level Difference

Interaural Level Difference (ILD) is the component of spatialization perception derived from the fact that ears are spread apart in space physically, and the volume (power) of sounds drops off as a function of distance. This means that a sound to the left of your head will sound slightly louder to your left ear than the right ear.

ILD is the primary perceptual reason why panning audio works and is a significant component to binaural spatialization. The primary factor that influences the ILD effect is the size of a person's head (or, more specifically, the distance between their ears).

Interaural Time Delay

Interaural Time Delay (ITD) is the component of spatialization perception derived from the fact that ears are spread apart in space physically, and it takes time for sound to travel through the air. This means that a sound to the left of your head will arrive at your left ear slightly faster than it will arrive at your right ear.

This effect is extremely small but detectable by the brain, and is a significant component to sound localization. Similar to ILD, the primary factor that influences the ILT effect is the size of a person's head.

Spectral Shadowing

Sound is a wave that diffracts when it comes in contact with a medium or obstruction. This diffraction can be quite complex and result in tiny variations and fluctuations in sound. This variation can occur in terms of loudness and frequency filtering.

If the spectral shadowing of a sound is consistent and predictable as a function of angle to your ears, the brain can interpret that information and derive valuable location information. This is one of the reasons why ears are oddly shaped and have little rivulets. These complex shapes are unique to each person but have evolved to help the brain with disambiguating and fine-tuning spatial information due to this spectral shadowing.

This effect is extremely individualized, and therefore very difficult to simulate in audio engines.

The Head-Related Transfer Function (HRTF)

The Head-Related Transfer Function, or HRTF, is used to combine all the psycho-acoustic factors that go into binaural spatialization together.

HRTFs are data-derived filters that are built by recording impulses as a function of angle, relative to a representative listener geometry.

A dummy head is used that is precisely the average head size (for a human) and average ear shape. Microphones are inserted into the ears of the dummy head, and impulses (tiny noise bursts) are recorded as a function of angle. Below is an example of a dummy head with a modeled ear with microphones placed inside the ear canal.

The derived filter is called Impulse Response (IR). As long as the system is time-invariant (doesn't change over time) and linear (doesn't have a non-linear mapping of inputs to outputs), IRs are filters that can represent all the complex details of that system without actually knowing all the internal details.

Since a person's ear shape and arrangement of parts is unique, a common IR set from a standard dummy will not result in the most accurate experience for many people. For this reason, many different IR data sets are produced to compensate for this.

Using HRTFs

By recording how a system modifies a single audio sample, subsequent samples can be processed by the recorded data set (by a process called convolution) and it will be as if those samples were processed by the original system.

HRTF data sets store IRs as a function of angle relative to the head. This is why a game audio engine uses the angle of the sound relative to the listener to look up the closest data set match at that angle. Some HRTF rendering techniques snap to the nearest IR and others perform interpolation between IR sets to get a more continuous IR.

The selected IR data set is then convolved with the input audio, and the output takes into account ILD, ITD, and the spectral characteristics of the actual shape of the ears and head.

The result is usually significantly more localizable for sounds than any one technique implemented on its own.

The downside of HRTF rendering (and binaural methods in general) is that it only works well with headphones. The filters and the delays break down when heard on stereo speakers, and the actual feeling of spatialization can be much worse than if traditional panning was used.

Hybrid Approaches

There are spatialization techniques that combine different elements of the above spatialization methods.

A common hybrid approach does the following:

- Encodes spatial information of individual sound sources into a channel-agnostic sound field.
- Mixes those sound field sources together.
- Decodes the sound field into an arbitrary virtual speaker / channel configuration.

Instead of outputting the decoded audio out to actual surround sound hardware, it outputs the virtual speaker audio into a binaural rendering process. This is a method widely used by "3D Audio" headphones that take surround sound output from a game and make it sound more three-dimensional for headphones. Some software techniques also use this method because it can result in overall lower CPU costs to get the benefits of HRTF spatialization.

Third-Party Plugins

Unreal Engine provides several third-party spatialization plugins. For additional information, refer to the third-party's official documentation.

Plugins handle channel management which may produce undesired results with multi-channel sources, as Unreal Engine only supports spatialization for mono and stereo sources.