A VIRTUAL INSTRUMENT FOR IFFT-BASED ADDITIVE SYNTHESIS IN THE AMBISONICS DOMAIN

Hilko Tondock and Henrik von Coler

Audio Communication Group
TU Berlin
voncoler@tu-berlin.de

ABSTRACT

Spatial additive synthesis can be efficiently implemented by applying the inverse Fourier transform to create the individual channels of Ambisonics signals. In the presented work, this approach has been implemented as an audio plugin, allowing the generation and control of basic waveforms and their spatial attributes in a typical DAW-based music production context. Triggered envelopes and low frequency oscillators can be mapped to the spectral shape, source position and source width of the resulting sounds. A technical evaluation shows the computational advantages of the proposed method for additive sounds with high numbers of partials and different Ambisonics orders. The results of a user study indicate the potential of the developed plugin for manipulating the perceived position, source width and timbre coloration.

1. INTRODUCTION

The dynamic distribution of sound on multichannel loudspeaker systems, known as spatialization or diffusion, has a long history in electronic and electroacoustic music. With a variety of algorithms, such as Vector Base Amplitude Panning (VPAB), Higher Order Ambisonics (HOA) or Wave Field Synthesis (WFS), any recorded or synthesized sound can be placed and moved in 2D or 3D space, resulting in immersive audio experiences. Spatial sound synthesis describes methods which treat spatial aspects as an integral part of the synthesis process, usually at an early stage in the algorithm. Such procedures can create sound events with inherently connected timbral and spatial properties. This concept has been investigated for many established synthesis paradigms, such as granular synthesis [1], physical modeling [2], FM Synthesis [3] and additive synthesis [4], respectively spectral modeling [5].

The approach presented in this paper is based on an efficient IFFT-based additive synthesis in the Ambisonics domain [6, 7]. This allows the synthesis of spectra with a large number of sinusoidal components, each with individual spatial attributes. To enable the use of this sound synthesis method beyond experimental music, an implementation as a VST plugin is presented. Thus, it can be included in a generic digital audio workstation (DAW) workflow. Synthesis parameters like pitch, timbre and spatial distribution can be controlled with sequences, envelopes and other modulators. This opens new possibilities, since the production of popular music for spatial audio setups and binaural playback is continuously gaining importance.

The remainder of this paper is organized as follows. Section 2 introduces the theoretical basics of the underlying algorithm. Section 3 deals with the implementation of the algorithms in the plugin and gives a brief overview of the current possibilities to integrate the plugin into digital music production environments. Section 4 presents a user study and evaluates the collected data.

2. ALGORITHM

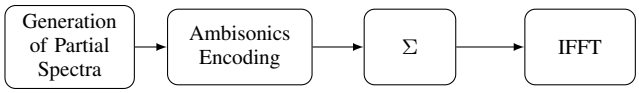


Figure 1: *Flow chart of the spatial IFFT-based additive synthesis.*

Figure 1 shows the main stages of the implemented IFFT-based additive synthesis in the Ambisonics domain. In the first step, a complex spectrum is generated for each partial. All partial spectra are then encoded into individual Ambisonics signals, which are summed to produce a single frequency-domain Ambisonics signal. Finally, each Ambisonics channel is transformed to the time domain via IFFT, resulting in an Ambisonics-encoded time domain signal. The individual steps are explained in the following sections.

2.1. Generation of Partial Spectra

In order to reduce the computing load for additive synthesis with a high number of partials, inverse DFT [8] and the inverse FFT [9] have been proposed for the signal model

$$x[n] = \sum_{i=1}^{N} a_i[n] \cos\left(\omega_i[n] T n + \varphi_i\right), \tag{1}$$

where N is the number of partials, a_i is the partial amplitude, ω_i is the partial frequency in radians, $T = \frac{1}{f_s}$ is the sampling period and φ_i the initial phase of the partial.

Partial spectra can be approximated by a small number of significant bins, around $K \approx 7$, in the magnitude spectrum, referred to as the *spectral motif* [10]. K is accountable for the approximation error and should be selected on the basis of the window properties and the desired signal-to-noise ratio (SNR). This approximation results in a computational benefit of roughly $\frac{H}{K}$, where H is the hop size, and is most effective with a suitable window that has as few decisive bins as possible. Such a window must have both a narrow main lobe and high side lobe suppression, which are conflicting requirements. Another optional approximation in [9]

expects stationary frequency, phase and amplitude of a sinusoid in one IFFT frame of size N. Hence, the associated window can be constructed as a real and even sequence, meaning symmetric with respect to the origin, thus the Fourier transform of the window is real.

In a harmonic signal, each partial corresponds to a shifted and scaled spectral motif. The inverse Fourier transform of the weighted sum of these spectral motifs produces a windowed signal in the time domain which has to be divided by the inverse window function to reverse the effect of the window. In order to concatenate frames, the overlap-add process is used, which involves weighting the time-domain signals of frames with a window that adds up to 1 in the overlapping region (e.g., a triangular window) [9]. Since the frequencies of partials in a frame are assumed to be stationary, the concatenation of adjacent frames with varying frequency components may cause modulations from phase cancellation in the overlapping regions. These distortions can be reduced by matching the phases of subsequent frames [9].

Several methods aim at improving the basic IFFT approach. Distortions can be minimized by using chirp signals instead of stationary frequencies in a frame [10, 11]. Laroche [12] presented an algorithm that avoids the overlap-add process and directly concatenates successive frames. This approach is refined by calculating optimal coefficients of the spectral motif and an optimal window function [13]. In [14], the subband sinusoidal synthesis algorithm is presented which evaluates a time-domain sinusoid at only a few samples and applies a pair of DFT to interpolate it to N samples. Aiming for higher SNR [12, 13] should be considered. If non-stationary sinusoids are mandatory, the approaches in [10, 14] will provide a foundation. The overlap-add method [9] is implemented because of its low computational cost and because there is no need for non-stationary sinusoids in the developed plugin.

2.1.1. Spectral Motif

The DTFT of a windowed sinusoid in the time domain yields the shifted Fourier transform of the window, multiplied by the complex amplitude [15, p. 797]:

$$\begin{aligned} X(e^{j\omega}) = \frac{A}{2} e^{j\varphi} W\left(e^{j(\omega-\omega_0 T)}\right) \\ + \frac{A}{2} e^{-j\varphi} W\left(e^{j(\omega+\omega_0 T)}\right), \end{aligned} \tag{2}$$

where $W\left(e^{j(\omega\pm\omega_0 T)}\right)$ is the shifted Fourier transform of the window sequence $w[n]$. Further, the DFT corresponds to equally spaced samples of the DTFT:

$$X[k] = X\left(e^{j\omega}\right)\Big|_{\omega=\frac{2\pi k}{NT}}. \tag{3}$$

The spectral motif can be created from a Fourier transform of a suitable oversampled time-domain window function, by extracting the main lobe values. The main lobe can be defined as the values between the first local minima to the left and right of the global maximum for most window functions. These values are stored in a lookup table and used later for synthesis. For this purpose, the window types Hann, Kaiser ($\beta = 8$) and Blackman-Harris are considered.

Besides the different types of windows and the window length M, it is of particular importance to emphasize the differences between true even symmetry and DFT-even symmetry windows, regarding discrete window functions [16]. True even symmetry refers to a sequence that has symmetric samples about an assumed midpoint and DFT-even symmetry relates to a true even sequence with the right endpoint removed. If the periodic continuation of a time-domain window function is symmetric with respect to the origin, the Fourier transform of this window will be real valued. In the discrete domain, if M is of even length and a symmetric window, there will be no peak value of exactly one. Instead two maximum values exist to fulfill the symmetric qualifications. In order to place the partials at the appropriate frequency location in the spectrum, it is desirable to use a window with a single maximum value of one. Also, it is a computational benefit to calculate only real values. Thus, the basis for the spectral motif is assumed to be a DFT-even symmetric sequence of an even length M.

Assuming $N = 512$ and $f_\mathrm{s} = 48000$ Hz, the frequency resolution results in $\frac{f_\mathrm{s}}{N} = 93.75$ Hz. If we assume the human ear resolves differences in frequencies of 1 Hz, a spectral motif with much finer resolution is required. Larger values for M result in a better frequency resolution in general but the main lobe width remains the same regarding the number of samples. Zero-padding is hence used to obtain an interpolated or rather oversampled version of the main lobe of the spectrum. Applying casual zero-padding (adding zeros after a signal) ruins the advantage of a real-valued spectrum. A possible solution utilizes zero-phase zero-padding to obtain a real valued spectrum, where first, the zero-frequency component is shifted to the center of the spectrum and afterwards zeros are inserted in the middle of the window without destroying the DFT-even symmetry.

Then, after the zero-phase shift is reversed, the oversampled main lobe or eventually additional side lobes of the spectrum of the window can be normalized and stored. The number of stored samples is dependent on the selected values for K and O.

2.1.2. Spectral Encoding

Since the objective is to generate real-valued signals and N is even, it suffices to calculate only $\frac{N}{2}+1$ samples of the spectrum, exploiting the symmetry properties of the DFT. The spectral motif $W[k]$ has to be placed correctly for the given frequency f_P of a partial. The floating bin location is equal to $k_\mathrm{f} = T_\mathrm{s} N f_\mathrm{P}$ in an N-point DFT. Because it is only possible to fill integer positions of k, firstly, the closest bin location is calculated by $k_\mathrm{i} = \lfloor k_\mathrm{f} + 0.5 \rfloor$ and the remaining distance $k_\mathrm{r} = k_\mathrm{i} - k_\mathrm{f}$ is stored separately.

Then, the decisive bins are calculated for $0 < k < \frac{N}{2}$ by

$$\begin{aligned} X_\mathrm{P}[k_\mathrm{i}+k] = \frac{A_\mathrm{P}}{2} W\left[\lfloor O(k_\mathrm{r}+k) + O_\mathrm{M} \rfloor\right] \cos(\varphi_\mathrm{P}) \\ + j\frac{A_\mathrm{P}}{2} W\left[\lfloor O(k_\mathrm{r}+k) + O_\mathrm{M} \rfloor\right] \sin(\varphi_\mathrm{P}) \end{aligned} \tag{4}$$

where O_M is the middle index of the spectral motif and φ_P is the current phase of the partial. Particular attention should be given to bin locations $k \leq 0$ and $k \geq \frac{N}{2}$. Directly from the definition of the DFT follows that for $k = 0$ and $k = \frac{N}{2}$:

$$X_\mathrm{P}[k] = A_\mathrm{P} W\left[\lfloor O(f_\mathrm{k}+k) + O_\mathrm{M} \rfloor\right] \cos(\varphi_\mathrm{P}). \tag{5}$$

Due to the periodicity of the DFT, frequency domain aliasing has to be considered. Based on the symmetry properties of the DFT follows for $k < 0$ and $k > \frac{N}{2}$:

$$\begin{aligned} X_\mathrm{P}[k_\mathrm{i}+k] = \frac{A_\mathrm{P}}{2} W\left[\lfloor O(k_\mathrm{r}+k) + O_\mathrm{M} \rfloor\right] \cos(\varphi_\mathrm{P}) \\ - j\frac{A_\mathrm{P}}{2} W\left[\lfloor O(k_\mathrm{r}+k) + O_\mathrm{M} \rfloor\right] \sin(\varphi_\mathrm{P}) \end{aligned} \tag{6}$$

The obtained spectrum forms the basis for the spatial encoding.

2.2. Ambisonics Encoding

Ambisonics is a surround sound technology that was developed in the 1970s [17, 18] and was further investigated, for example, in [19]. It is a spatial audio representation that encodes incoming sound sources in three dimensions, allowing for the capture and reproduction of a full-sphere sound field.

The first-order Ambisonics B-format refers to encoding pressure and velocity at the origin of a sound field to four channels. However, first-order Ambisonics is limited in terms of spatial resolution. To resolve this issue, high-order Ambisonics (HOA) is calculated using higher numbers of so called spherical harmonics. These are functions on the surface of a sphere that can describe the distribution of sound pressure. A comprehensive summary of Ambisonics-related theory can be found in [20].

A real-valued set of spherical harmonics can be defined

$\forall(n,m) \in \mathbb{N}_N$ by

$$Y_n^m(\theta,\phi) = N_n^{|m|} P_n^{|m|}(\sin\theta) \begin{cases} \cos(m\phi), & m \geq 0 \\ \sin(m\phi), & m < 0 \end{cases} \tag{7}$$

where the elevation angle θ is 0 at the horizontal plane and positive in the upwards direction, while the azimuth angle ϕ is 0 pointing in face direction and increases counter-clockwise. $N_n^{|m|}$ represents a normalization constant. The associated Legendre functions P_n^m are defined by

$$P_n^m(x) = (-1)^m \left(1-x^2\right)^{\frac{m}{2}} \frac{d^m}{dx^m} P_n(x), \tag{8}$$

with the Legendre polynomials P_n which can be expressed in the Rodrigues representation

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} \left(x^2-1\right)^n. \tag{9}$$

There are various conventions for ordering and normalizing Ambisonics channels. The ambiX format [21] is a widely used standard and is implemented in the proposed algorithm. It specifies the Ambisonics channel number (ACN) for channel ordering:

$$ACN(n,m) = n^2 + n + m + 1 \tag{10}$$

and the Schmidt semi-normalized (SN3D) convention

$$N_n^m = \begin{cases} 1, & m = 0 \\ (-1)^n \sqrt{2\frac{(n-m)!}{(n+m)!}}, & m \neq 0 \end{cases} \tag{11}$$

is proposed in terms of normalization.

2.2.1. Displacement Function

In the proposed synthesis approach, Ambisonics encoding takes place in the spectral domain, after generating the complex spectra and before applying the IFFT. Each partial of the additive synthesis is assigned individual angles θ and ϕ. Equation 7 determines the spatial gains for each Ambisonics channel, depending on these angles. The previously calculated real and imaginary parts of the spectra are then scaled by these spatial gains.

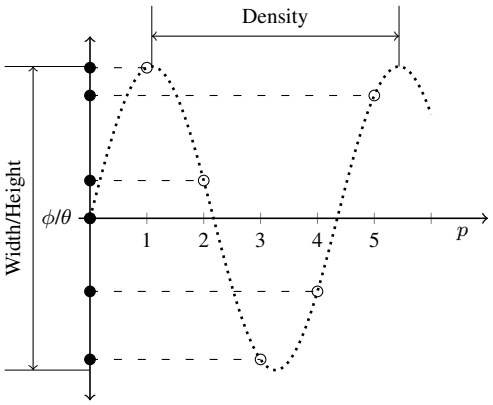


Figure 3: *Example for the placement of the first six partials of the sine type displacement function.*

With increasing number of partials, individual control of their positions is not possible. Hence, various displacement functions can be used to distribute the partials with few meta-parameters. The standard sinusoidal displacement function for a single dimension is defined as:

$$D[p] \coloneqq \frac{H}{2} \sin\left(\frac{2\pi S p}{P}\right) \tag{12}$$

where H is the width or height, S determines the horizontal or vertical dispersion pattern, P is the total number of partials and p is the partial index. The resulting values are added to the global azimuth or elevation angle of the sound, as shown in Figure 3.

2.3. IFFT

2.3.1. Overlap-add

After Ambisonics encoding, an IFFT is aplied to each Ambisonics channel. The resulting time-domain signal needs to be divided by the window function that was applied to reverse the windowing effect. However, this results in large amplitude values at the boundaries of the frame, which can cause distortion when overlapping with successive frames in the overlap-add process. To ensure equal energy in the overlapping sections, a proper weighting function must be applied. The triangle window is often used for this purpose and the calculation of the inverted synthesis window and the amplitude weighting function can be combined. To sufficiently reduce distortion at the edges, it is recommended to choose the hop size H less than or equal to 25% of the frame size [22].

2.3.2. Phase Adjustment

Phase adjustment is a technique that can be used to attenuate amplitude modulations that result from phase cancellation due to destructive interference of adjacent frames. The specific implementation depends on which signal information is stored and whether there is any frequency variation at all. In this case, only the last phase information for each partial is stored, with the initial phase of a partial located at $N/2$. The points at which adjacent overlapping frames have the same energy are $\frac{1}{2}(N+H)$ and $\frac{1}{2}(N-H)$, respectively. First, the initial phase at $\frac{1}{2}(N+H)$ of the current

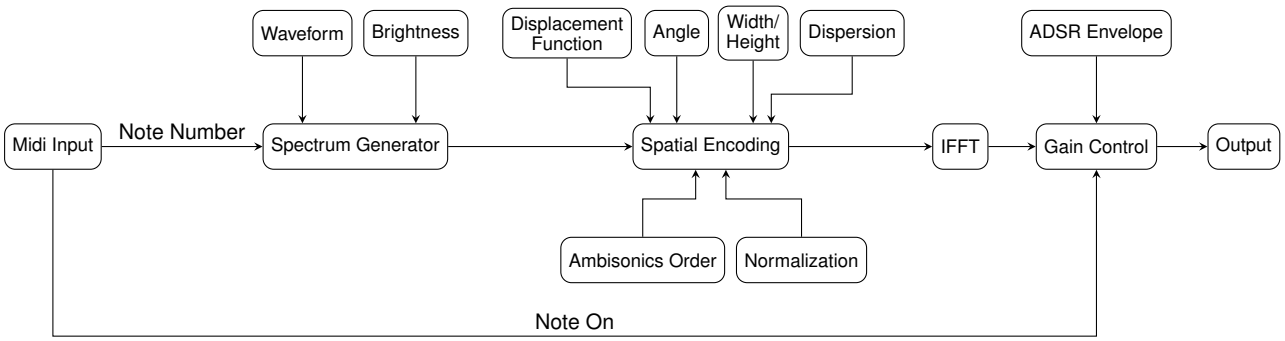


Figure 2: *Signal flow of the plugin.*

partial is calculated as follows:

$$\varphi_t\left[\frac{N}{2}\right] = \left(\varphi_{t-1}\left[\frac{1}{2}(N+H)\right] + \pi f_t H T_\mathrm{s}\right) \bmod 2\pi, \quad (13)$$

This value is then used for partial placement, and the new phase is stored for the next callback, calculated by:

$$\varphi_t\left[\frac{1}{2}(N+H)\right] = \left(\varphi_t\left[\frac{N}{2}\right] + \pi f_t H T_\mathrm{s}\right) \bmod 2\pi. \quad (14)$$

If the frequency remains constant throughout the lifetime of a partial, the calculation reduces to a single floating-point remainder function call. However, it is important to note that the adjustment only works effectively when the frequencies of consecutive frames are close together.

3. IMPLEMENTATION

3.1. Plugin

Figure 2 shows the signal flow of the plugin. Table 1 lists the adjustable parameters and their range of values. This first version of the plugin features the basic waveforms triangle, sawtooth, and square wave. All these waveforms are well suited for the production of lead and bass sounds in popular electronic music, especially when the high frequency content is changed over time by temporal envelopes. A pure sine wave has been added as an anchor for the test phase, since it emphasizes artifacts and distortions. The pitch of the waveform is controlled by MIDI. Since the plugin itself does not feature pitch modulation capabilities, a signal just consists of stationary sinusoids after being generated based on the note-on event. Assuming stationary sinusoids, a real-valued spectral motif can be constructed. Frequency-domain additive synthesis can handle aliasing by constraining the maximum permitted frequency of a partial below half of the sampling rate. To get control over the overtones, the Brightness parameter is implemented as follows:

$$A[p] = A[p]e^{-\frac{p}{d}}, \quad (15)$$

where $A[p]$ is the amplitude of the current partial index p and d is an adjustable damping factor.

The plugin is based on the JUCE[1] framework and integrates the KFR[2] library for FFT/IFFT and is made publicly available[3].

[1]https://github.com/juce-framework/JUCE
[2]https://github.com/kfrlib/kfr
[3]https://github.com/ringbuffer-org/spadd

Table 1: *Plug-in parameters and range of values.*

Parameter	Range of Values
Waveform	Sine, Triangle, Sawtooth, Square, Noise
Noise Density	1 to 10000
Brightness	0.5 to 250
Distance	1.0 to 100.0
Horizontal Displacement Function	Sine, Cosine, Sawtooth, Square
Azimuth Angle	-180 to 180 degree
Width	0 to 180 degree
Horizontal Dispersion	0 to 300
Vertical Displacement Function	Sine, Cosine, Sawtooth, Square
Elevation Angle	0 to 90 degree
Height	0 to 90 degree
Vertical Dispersion	0 to 300
Ambisonics Order	0^{th},1^{st}, 2^{nd}, 3^{rd}
Normalization	SN3D, N3D
Gain Attack	0 to 5 s
Gain Decay	0 to 8 s
Gain Sustain	0 to 1
Gain Release	0 to 8 s

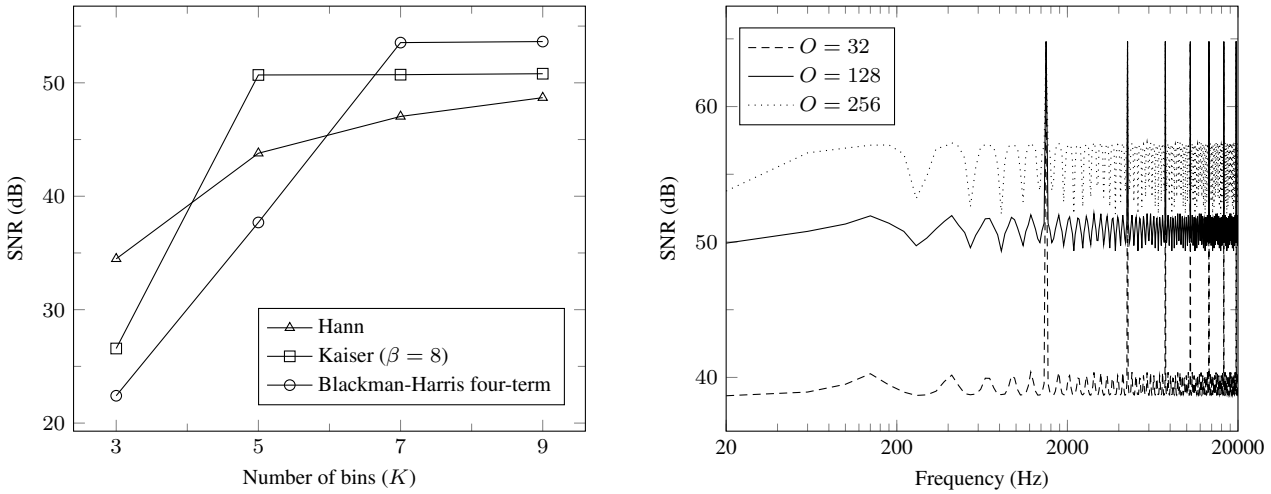
3.1.1. Distortion Analysis

Distortion artifacts mainly stem from the overlap - add process, which are minimized here by calculating an appropriate phase adjustment. Other adjustable sources of distortion are O and K, regarding the type of window function. In order to inspect the distortion and to choose suitable parameters for the implemented algorithm, the SNR is calculated as

$$\mathrm{SNR} = 10\log_{10}\left(\frac{E_\mathrm{signal}}{E_\mathrm{noise}}\right) \quad (16)$$

where E is the energy of a signal $x[n]$ of length L:

$$E = \sum_{n=0}^{L-1} |x[n]|^2. \quad (17)$$



(a) (b)

Figure 4: *Signal-to-noise ratio for (a) different window functions with increasing number of bins (K) and fixed overlap (O = 128) and (b) for the Kaiser window (β = 8) with K = 5 dependent on O and the frequency.*

E_{signal} in this case takes a sinusoidal reference signal consistent with the phase of the synthesized output signal as a basis and E_{noise} relates to the difference between the reference signal and the output signal. The first output frame is discarded because there is no overlap-add data yet. Note that this property generally affects transient signals, e.g. for $H = 256$ and $f_s = 48000$ Hz an attack time of 5.3 ms occurs. Figure 4(a) shows the SNR for different windows dependent on K with fixed O.

As expected, window functions with a narrower main lobe converge towards a maximal possible SNR earlier, whereas window functions with a higher side lobe suppression reach a higher maximal SNR at the expense of higher K. Hence, to achieve even higher SNR, a Kaiser window with $\beta > 8$ or a Blackman-Harris window with more than four coefficients could be applied. The SNR is also dependent on O and the frequency, as shown in Figure 4(b). The peaks of the curves emerge, if the frequency matches an exact bin location. Note that O does not have to be a power of two in general but can be computed efficiently. An SNR ≥ 40 dB was considered to be sufficient to exclude audible artifacts in [12]. Hence, the Kaiser window ($\beta = 8$) with $K = 5$ and $O = 256$ and the Blackman-Harris four-term window with $K = 7$ and $O = 128$ are taken into account during implementation.

3.1.2. Performance Analysis

Music production setups usually utilize either internal or external sound cards (interfaces) for enhanced usability and performance. Audio drivers typically provide a range of buffer sizes between 16 and 2048 samples, which leads to round-trip latencies between 0.7 and 43 milliseconds, assuming $f_s = 48000$ Hz. Measurements were taken on an Intel Core i7-7700HQ, representing today's average consumer CPU. The benchmark test utilizes a scope based timer, containing only the frequency and time-domain specific process functions, that means all function calls which are the same for both methods are excluded, e.g. constructing the basic signal information or midi processing. The plugin was embedded in a digital audio workstation during testing and ran over 2000 callbacks. Random partials were generated every callback, taking into account the branches which stem from (4), (5) and (6). Figure 5 contains the results of the measurements.

As Figure 5(a) shows, the frequency-domain approach is faster than the time-domain approach above 10 partials. 100 partials are sufficient for synthesizing the deterministic part of sounds from most musical instruments, considering the upper frequency limit of our auditory system and the fundamental frequencies of musical sounds. For 100 partials, the frequency domain approach is roughly twice as fast as the time domain approach ($0.239\,\text{ms}$ / $0.112\,\text{ms}$). For low frequency sounds, especially synthetic ones, more partials may be necessary. If non-harmonic partials are considered in order to create stochastic components, the number of partials can be increased to several hundreds [23]. For 1000 partials, the performance ratio between time domain and frequency is about 6 ($1.783\,\text{ms}$ / $0.295\,\text{ms}$), which indicates a significant advantage of the IFFT approach.

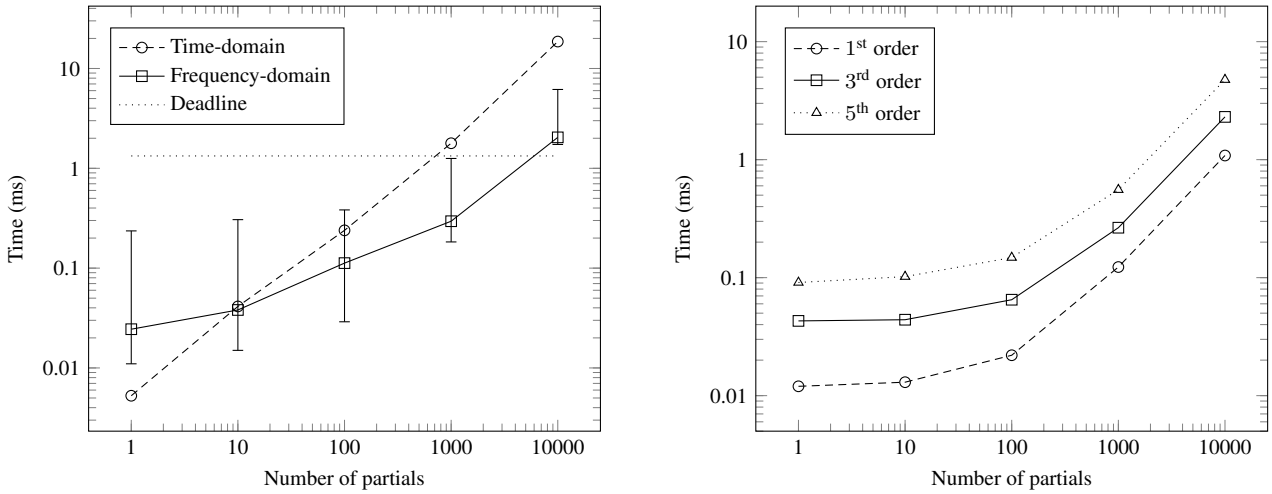
As shown in Figure 5(b), the performance load increases with the Ambisonics order, independent of the number of partials.

3.2. DAW Integration

Typically, working with digital audio material involves using a digital audio workstation (DAW) for recording, editing or synthesizing audio content. DAWs mostly support one or more plugin formats, providing flexibility by enabling the use of third-party effects and instruments. For the developed plugin, the main requirements are a graphical user interface, support on all major platforms, multichannel capabilities, and a free licensing model. VST 3[4] and CLAP[5] have been considered in this context, but since the JUCE framework does not natively support CLAP, the plugin uses VST 3.

To use Ambisonics, the setup has to support multichannel layouts, which means the DAW must be able to process multiple input and output sample streams simultaneously. However, many DAWs

[4]https://steinberg.net/developers
[5]https://cleveraudio.org



(a) (b)

Figure 5: *Benchmark results of (a) 3rd order Ambisonics for the frequency-domain (K = 7, O = 128 and N = 256) and the time-domain (naive wavetable) with the deadline indicating a buffer size of 64 samples and f_s = 48000 Hz. The minimum and maximum values are shown only for the frequency-domain due to transparency. (b) Comparison of different Ambisonics orders for the frequency-domain (K = 7, O = 128 and N = 1024) and a buffer size of 256 samples.*

do not support proper multichannel workflow. Reaper[6] includes extensive multichannel support but does not provide extended parameter modulation capabilities easily. Ableton Live[7] has no native support for multichannel plugins, but it integrates Max[8] as Max4Live, allowing the integration of multichannel plugins. The only drawback is that the GUI of the plugins cannot be directly accessed anymore, and must be controlled through Max4Live parameters. However, comprehensive parameter modulation is a preferable feature for the user study.

Since Abisonics decoding is not integrated into the plugin, an external decoder has to be inserted after the synthesis plugin. At this stage, the AllRADecoder[9] of the IEM plugin suite is used with 3rd order Ambisonics.

4. USER STUDY

A user study was conducted to investigate the perception of the source spread in relation to the localizability. User feedback and open responses were collected to evaluate additional and general aspects of the instrument.

4.1. Setup and Procedure

A total of 12 participants with a mean age of 27.25 years (SD = 3.6 years), took part in the study. They were seated at a desk in the center of a 21 channel loudspeaker system in a dome configuration.

The developed synthesizer plugin (Blackman-Harris four-term window with $K = 7$ and $O = 128$) and the AllRADecoder were integrated into a Max4Live instrument running in Ableton Live 11 on Windows 10. The Ableton Push Controller was used in the study, and parameters were automated by Ableton tools like LFO and Shaper. As the plugin is integrated in Max4Live, only essential parameters were made accessible through the GUI. Control over Ambisonics order and normalization, the gain envelope, and selection of the displacement function (fixed to sine type) were non-adjustable and hidden from the user.

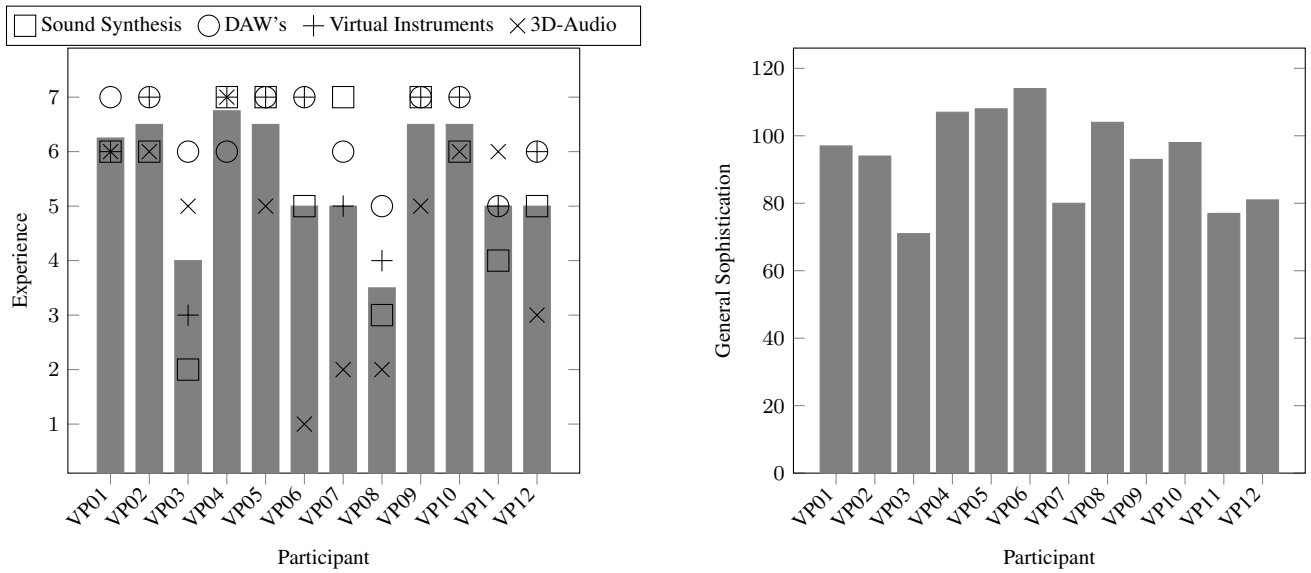
In the first part of the study, test subjects used the Ableton Push controller to manually adjust the spatial and timbral parameters. In the second part, parameters were automated by envelopes and low-frequency oscillators. Finally, the test subjects were able to give open feedback. The subjects' experience was evaluated using a seven-point balanced Likert scale ranging from "Completely Disagree" (1) to "Completely Agree" (7) and the general sophistication of the Gold-MSI [24, 25]. The parameter settings were assessed on a five-point Likert scale, and the perceptual qualities were named based on [26]. Higher values stand for a greater development of the corresponding quality of perception.

4.2. Results

The results of the users' expertise are shown in Figure 6. Figure 6(a) shows the self-reported experience regarding sound synthesis, DAWs, virtual instruments, and 3D-audio. The scores of the general sophistication of the Gold-MSI Figure are displayed in Figure 6(b).

Figure 7 presents the results for the auditory qualities that were examined in relation to spatial expansion. A paired t-test shows a statistically significant difference in the localizability between the conditions with and without spatial spread of the sound source (t(11) = 2.14, p = .028, one-tailed). Specifically, the mean localizability score was higher without spatial extension (M = 4.25, SD = 0.7) than with spatial extension (M = 3.5, SD = 1.1). The continuous change of the horizontal and vertical dispersion were examined for roughness (M = 2.83, SD = 0.99) and degree-of-liking (M = 3.67, SD = 0.75).

[6] https://www.reaper.fm
[7] https://ableton.com/live
[8] https://cycling74.com/products/max
[9] https://plugins.iem.at



(a) (b)

Figure 6: *Experience of the participants of the user study. (a) Self-reported expertise of specialized topics where the gray bar indicates an average value and (b) determination of the general sophistication of the Gold-MSI with a maximum attainable test score of 126.*

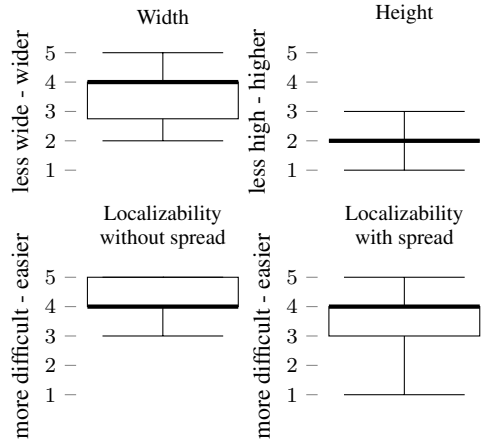


Figure 7: *Results of the investigated perceptual qualities in relation to spatial expansion.*

In the open responses it was noted that sounds were more difficult to localize in the vertical plane, and the Height parameter was not perceived as particularly influential by some subjects. A test subject explicitly mentioned a phaser effect occurring with extreme parameter modulations. Another test subject described the modulated expansion of the signal as more of a tone coloring and only to a limited extent as a spatial expansion. The changing overtone distribution emanating from the displacement function, depending on the number of partials, was explicitly rated as good by one test subject.

4.3. Discussion

Although sine waves are in general harder to localize, the results show an influence of angles and spread on the perception of the synthesized sound. Figure 7 and the t-test indicate that the spatial expansion of the signal works to a certain extend. According to the results, sources with increased spread are harder to localize. This holds true for most real or virtual sound sources and validates the source widening effect in the proposed approach.

The weaker perception of height extension, compared to width, has also been noted by some of the test subjects. This can be attributed to the loudspeaker setup, which is more sparse along the elevation axis. In addition, the human auditory system has a higher resolution for the azimuth than for the elevation of sound source positions.

The temporal change of the spread S can provoke a flanger-like effect. While the resulting tone-coloring effect is perceived as rough, the roughness is not necessarily perceived as unpleasant. In conclusion, the resulting effect has timbral qualities, which may not be achieved without the spatial modulation. Considering one isolated Ambisonics channel, moving notches in the amplitude spectrum occur which can be compared to a moving comb filter. Partials that are present in one channel at a point in time, are attenuated in other channels. Thus, the corresponding frequency modulated comb filters follow the same frequency but with temporal shifts. The flavour of the tonal coloration varies with the type of the displacement function.

5. CONCLUSION

The presented synthesizer plugin makes IFFT-based additive synthesis in the Ambisonics domain available in conventional music production workflows on the DAW. The chosen approach demonstrates a sufficient SNR and the performance analysis reveals an

expected advantage over additive synthesis in the time-domain. Furthermore, the IFFT does not represent a bottleneck with respect to the amount of calculations required for each channel of HOA. Therefore, signals with a high overtone density can also be efficiently synthesized for higher Ambisonics orders.

The user study results indicate that the chosen method for spatial distribution can be used to influence source position and width, as well as tone coloration. For a more detailed investigation, the experimental design needs to be changed and more test users should be included.

The integration of multichannel capabilities is still in its early stages in many DAWs, which makes it difficult to use plugins for Ambisonics and related technologies without additional customization. Considering the rise of spatial audio production techniques in music production, movie sound, extended reality (XR) and video games, this problem is likely to disappear in the near future. Synthesizers with an increased focus on spatial abilities are thus predestined to become a standard in these domains. Future work will focus on other waveforms than the basic ones used in this version. This will also include partial trajectories from previously analyzed recordings of musical instruments.

6. REFERENCES

[1] A. McLeran, C. Roads, B. L Sturm, and J. J Shynk, "Granular sound spatialization using dictionary-based methods," in *Proc. 5th Sound and Music Computing Conf. (SMC)*, 2008.

[2] A. Müller and R. Rabenstein, "Physical modeling for spatial sound synthesis," in *Proc. Digital Audio Effects (DAFx-09)*, 2009.

[3] R. McGee, "Spatial modulation synthesis," in *Proc. Int. Computer Music Conference (ICMC)*, 2015.

[4] D. Topper, M. Burtner, and S. Serafin, "Spatio-operational spectral (SOS) synthesis," in *Proc. Digital Audio Effects (DAFx-02)*, Singapore, 2002.

[5] H. von Coler, *A System for Expressive Spectro-spatial Sound Synthesis*, PhD Thesis, Technische Universität Berlin, Berlin, 2021.

[6] C. Verron, M. Aramaki, R. Kronland-Martinet, and G. Pallone, "A spatialized additive synthesizer," in *Proc. Inaugural Int. Conf. Music Commun. Science (ICoMCS)*, Sydney, Australia, Dec. 5-7, 2007.

[7] C. Verron, M. Aramaki, R. Kronland-Martinet, and G. Pallone, "A 3-D immersive synthesizer for environmental sounds," *IEEE Trans. Audio, Speech and Language Processing*, vol. 18, no. 6, pp. 1550–1561, Aug. 2010.

[8] H. A. Chamberlain, "Experimental fourier series universal tone generator," *J. Audio Engineering Society*, vol. 24, no. 4, pp. 271–276, May 1976.

[9] X. Rodet and P. Depalle, "Spectral envelopes and inverse FFT synthesis," in *Proc. 93rd AES Conv.*, San Francisco, CA, US, Oct. 1-4, 1992.

[10] M. Goodwin and A. Kogon, "Overlap-add synthesis of nonstationary sinusoids," in *Proc. Int. Computer Music Conf.*, Banff, Alta., Canada, 1995.

[11] M. Goodwin and X. Rodet, "Efficient Fourier synthesis of nonstationary sinusoids," in *Proc. Int. Computer Music Conf.*, San Francisco, CA, 1994.

[12] J. Laroche, "Synthesis of sinusoids via non-overlapping inverse Fourier transform," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 4, pp. 471–477, Jul. 2000.

[13] R. Kutil, "Optimized sinusoid synthesis via inverse truncated Fourier transform," *IEEE Trans. Audio, Speech and Language Processing*, vol. 17, no. 2, pp. 221–230, Feb. 2009.

[14] X. Wen and M. Sandler, "Fast additive sinusoidal synthesis with a subband sinusoidal method," *IEEE Signal Processing Letters*, vol. 20, no. 5, pp. 467–470, May 2013.

[15] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Pearson Higher Education, Inc., Upper Saddle River, NJ, US, third edition, 2010.

[16] F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proc. IEEE*, vol. 66, no. 1, pp. 51–83, Jan. 1978.

[17] M. A. Gerzon, "Periphony: With-height sound reproduction," *J. Audio Eng. Soc*, vol. 21, no. 1, pp. 2–10, 1973.

[18] P. Fellgett, "Ambisonic reproduction of directionality in surround-sound systems," *Nature*, vol. 252, no. 5484, pp. 534–538, 1974.

[19] J. Daniel, R. Nicol, and S. Moreau, "Further investigations of high-order ambisonics and wavefield synthesis for holophonic sound imaging," in *Proc. 114th AES Conv.*, Amsterdam, Netherlands, Mar. 22-25, 2003.

[20] T. Carpentier, "Normalization schemes in ambisonic: Does it matter?," in *Proc. 142th AES Conv.*, Berlin, Germany, May 20-23, 2017.

[21] C. Nachbar, F. Zotter, E. Deleflie, and A. Sontacchi, "AmbiX – a suggested ambisonics format," in *Proc. Ambisonics Symp.*, Lexington, KY, Jun. 2-3, 2011.

[22] U. Zölzer, Ed., *DAFX: Digital audio effects*, chapter Spectral processing, p. 408, J. Wiley & Sons, The Atrium, Southern Gate, Chichester PO19 8SQ, UK, second edition, 2011.

[23] Adrian Freed, Xavier Rodet, and Philippe Depalle, "Synthesis and control of hundreds of sinusoidal partials on a desktop computer without custom hardware," in *ICSPAT (International Conference on Signal Processing Applications & Technology*, 1992.

[24] D. Müllensiefen, B. Gingras, J. Musil, and L. Stewart, "Measuring the facets of musicality: The goldsmiths musical sophistication index (Gold-MSI)," *Personality and Individual Differences*, vol. 60, pp. S35, 2014.

[25] N. K. Schaal, A. R. Bauer, and D. Müllensiefen, "Der Gold-MSI: Replikation und Validierung eines Fragebogeninstrumentes zur Messung musikalischer Erfahrenheit anhand einer deutschen Stichprobe," *Musicae Scientiae*, vol. 18, no. 4, pp. 423–447, 2014.

[26] A. Lindau, V. Erbes, S. Lepa, H. Maempel, F. Brinkmann, and S. Weinzierl, "A spatial audio quality inventory for virtual acoustic environments (SAQI)," *Acta Acustica united with Acustica*, vol. 100, no. 5, pp. 984–994, Sep./Oct. 2014.