ATK Reaper: Ambisonic Toolkit as JSFX plugins

Trond Lossius
BEK - Bergen Centre for Electronic Arts
`trond.lossius@bek.no`

Joseph Anderson
DXARTS, University of Washington
`joanders@uw.edu`

ABSTRACT

While there is a well-established workflow for stereo in DAWs, options have been more limited when working with Ambisonics. The Ambisonic Toolkit (ATK) brings together a number of tools and transforms for working with First order Ambisonic surround sound, including intriguing possibilities for spatial transforms of soundfield image. Previously these tools have only been available in the real-time processing environment SuperCollider.

Reaper is a reasonably priced and flexible DAW for spatial sound, popular among many composers and sonic artists working with spatial sound. ATK has been ported to plugins for the Cockos Reaper DAW using the JSFX text-based scripting language, with intuitive graphical user interfaces. As compared to development in C++, this has simplified the effort, and plugins are immediately available for all platforms supported by Reaper. Future maintenance is also expected to be less demanding.

1. INTRODUCTION

1.1 Spatial sound representations

Spatial sound is usually represented in one of three ways: as channel feeds, as spatial scene descriptions, or as sound field encodings. ITU 5.1 and ITU 7.1 are examples of fixed-channel distribution formats [1]. This approach becomes less practical as the number of channels increases, and also has clear limitations due to the lack of flexibility in loudspeaker positioning. Object-based spatial scene descriptions instead associate sound sources with meta-information describing location or direction. In this way an auditory scene can be described independently of loudspeaker setup, and rendered appropriately for a chosen playback system. This is the approach used for e.g., Dolby Atmos and Wave Field Synthesis [3, 2]. SpatDIF, the Spatial Sound Description Interchange Format, is an open format offering a semantic and syntactic specification for storing and transmitting such spatial audio scene descriptions [4].

Ambisonics encodes audio sources into a speaker-independent representation of the sound field called B-format [5]. Decoding is the process where an encoded sound field is translated to individual speaker channel feeds; an advantage being, decoders can be designed for different speaker arrays. Ambisonics is based on spherical harmonic decomposition of the sound field, and depending on truncation of the spherical harmonic decomposition, the B-format signal may be First order Ambisonic (FOA) or higher order (HOA) [1] [6].

Spatial resolution improves with increasing order, but so does the number of channels required for the encoded signal. A number of commercial microphones are available for recording in FOA, and free or commercial B-format recordings are available as sound effects libraries [7, 8, 9].

1.2 Spatial sound processing in DAWs

Encoding, processing and decoding of B-format signals requires support for multi-channel signals. Real-time programming environments are well-suited in this respect as they offer flexible configuration and routing of channels. Ambisonics is supported in all major real-time audio programming environments, such as Csound, Max, Pd and SuperCollider, often as 3rd party extensions to these programs.

The support for surround sound is more limited in most Digital Audio Workstation (DAW) programs. Ableton Live is primarily geared towards stereo, although Max 4 Live devices can be used for spatialisation by routing signals to several buses. Most DAWs that support surround sound, such as Adobe Audition, Cubase, Logic Pro X and Pro Tools are oriented towards fixed-channel distribution formats, with an upper limit of 6 (5.1) or 8 (7.1) channels. Digital Performer also supports 10.2, and Nuendo supports up to 13 channels in various fixed-channel distribution formats. Nuendo and Pro Tools can be extended to support object-based spatial scene descriptions. The Iosono Spatial Audio Workstation program plug-in for Nuendo extends the capabilities of this program by adding abilities for object-based manipulation of sound sources [10], and can be used with Iosono Core hardware systems for wavefield synthesis. Dolby Atmos authoring is achieved using ProTools and the Dolby Rendering and Mastering Unit (RMU). RMU provides the rendering engine for the mix stage, and integrates with Pro Tools through the Dolby Atmos Panner plug-in over Ethernet for metadata communication and monitoring. The metadata is stored in the Pro Tools session as plug-in automation [11].

The DAWs discussed so far are either limited with respect to surround sound abilities, or expensive and proprietary software and hardware systems for object-based spatial scene authoring and rendering. In contrast Reaper is a reasonably priced and flexible DAW for spatial sound [12].

---

[1] Third order Ambisonic is often abbreviated TOA.

Reaper supports tracks of up to 64 channels [2], and is exceptionally open-ended with respect to routing of channels and tracks. Reaper supports all standard fixed-channel surround sound configurations, and the ReaSurround panning plugin that is integrated with Reaper also caters for non-standard speaker configurations [13], using a spatialisation algorithm based on principles resembling VBAP [14] as well as DBAP [15]. Reaper is well-suited for Ambisonics as well, and is a preferred DAW for many composers within the Ambisonic community [16].

1.3 Plugin technologies

Several plugin specifications enable development of virtual instruments and effects that can be used with multiple hosts. VST is a plugin interface specification and SDK for Windows, Mac OSX and Linux by Steinberg [17]. It is a popular format for commercial and freeware plugins, mostly using the VST 2 SDK, and a large number of audio applications support VST under license from its creator. AudioUnits (AU) is a Mac OSX-specific plugin architecture provided by Apple as part of CoreAudio [18]. In a similar way DirextX can be used for development of Windows-only plugins [19]. LADSPA is a GNU LGPL plugin architecture mostly used on the Linux platform [20], and is gradually superseded by LADSPA version 2 (LV2) [21].

While the above formats can be used to develop plugins for use with multiple hosts, there are also a number of specifications that permits 3rd part development of plugins for one hosting environment only. Avid has developed the RTAS, TDM and more recently AAX plugin format that work with their software and hardware exclusively. Ableton and Cycling'74 jointly offer Max for Live, enabling the use of Max patches as devices in Ableton Live [22]. JSFX is a text-based scripting language for programming audio-oriented effects compiled on the fly for Cockos Reaper [23, 12].

The multiple operation systems and plugin architectures poses challenges to plugin developers, that ideally might want to offer their software work with all plugin architectures and hosts in all operation systems. Developing, compiling and maintaining plugins for many parallel platforms can introduce substantial workload overhead, and several cross-platform plugin development frameworks have been developed to address this, e.g., JUCE [24], wdl-ol [25] and FAUST [26]. *SonicBirth* can be used to create AudioUnit and VST plugins for OSX and Windows by connecting objects in a graphical programming environment. Development stalled in 2007, but recent renewed efforts have resulted in a version 2 public alpha being available [27]. *SynthEdit* is a similar technology for VST plugin development on the Windows platform [28].

1.4 Plugins for Ambisonic processing

In recent years a number of plugins have emerged for Ambisonic processing. *WigWare* is a set of VST plugins for Mac and Windows that includes FOA and HOA panner and

---

[2] Suitable for 7th-order HOA.

decoders for regular and irregular speaker layouts, as well as a FOA 3D reverb [29]. Danielle Courville has developed a number of FOA plugins in SonicBirth for encoding and decoding, correcting misaligned B-Format streams, performing rotations, mixing, reverb, and decoding, as well as encoders and decoders for 2nd order and planar 5th order Ambisonic [30]. Due to the long halt in development of SonicBirth, these plugins do not work reliable in all newer versions of DAW programs. *ambiX* are cross-platform 1st, 3rd and 5th order Ambisonic processors for encoding and decoding, including binaural decoding, as well as spatial transforms [31, 32]. These plugins use the ambiX encoded signal convention; full 3D, ACN channel ordering and SN3D normalization [33] as discussed in section 3.1, but also provides a plugin for conversions to other formats. They are accompanied by *mcfx*, a number of more general multichannel plugins for equalising, delay, gain adjustment and level metering [34]. *Flux Ircam Spat* wraps the Ircam Spatialisateur multiformat room acoustics simulation and localization software as AU and VST plugins for Mac and Windows [35]. Although this plugin mainly provides object-based scene descriptions, it can render the auralization as B-format. The *HARPEX* plugin for decoding FOA uses an algorithm based on high angular resolution planewave expansion (HARPEX) in order to address issues with low angular resolution and small sweet spot that FOA suffers from [36]. Blue Ripple Sound offers a wide range of OSX and Windows VST plugins for TOA, including encoders and upmixers, decoders, and a rich set of plugins for spatial and filter-based manipulations of the encoded signal. The TOA Harpex Upsampler converts FOA material to TOA using the Harpex algorithm [37].

2. AMBISONIC TOOLKIT

The Ambisonic Toolkit (ATK) brings together a number of tools and transforms for working with Ambisonic surround sound [38, 39]. Use is targeted towards the composer of acousmatic and experimental music. The intention is for the toolset to be both ergonomic and comprehensive, providing algorithms to creatively manipulate and synthesize Ambisonic sound fields.

The tools are framed for the user to 'think Ambisonically'. By this, it is meant the ATK is focused on the problem of synthesising and processing sound fields. The model of the ATK is a sound-field sound-image model rather than a sound-object sound-scene model. By addressing the holistic problem of creatively controlling a complete soundfield, the ATK allows and encourages the composer to think beyond the placement of sounds in a sound-space and instead attend to the impression and image of a soundfield. This is viewed to be the *idiomatic* approach for working with the Ambisonic technique, and creatively leverages the model the Ambisonic technology presents.

Since 1998 the ATK has existed in a variety of forms. In its very first implementation, the ATK was deployed as a collection of Csound orchestras, and a collection of VST plugins have previously been distributed using the now discontinued SonicBirth. Development of the real-time ATK library for SuperCollider2 began in 2000, and in
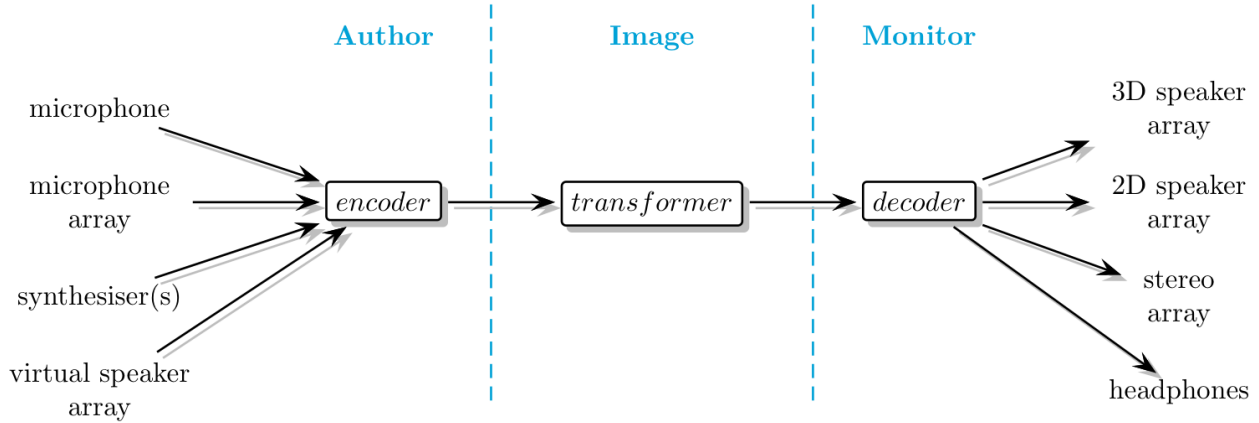
Figure 1: Ambisonic Toolkit Paradigm.

recent years ATK has primarily been distributed as a version for SuperCollider3 [38]. Some of the underlying ideas of ATK has also been incorporated into the Blue Ripple Sound Third order Ambisonic plugins [37].

3. ATK REAPER: DESIGN CONSIDERATIONS

3.1 Coordinate systems and encoding conventions

Within the spatial audio community several coordinate system conventions are being used in parallel. Theory on Ambisonic generally assume the same coordinate system conventions as acoustics, with the x-axis pointing forward, y-axis to the left, and the z-axis pointing upwards. Spherical coordinates also follows standard mathematical conventions, with $0°$ azimuth being forward, and angles increasing in the anti-clockwise direction so that $90°$ azimuth is to the left. Positive elevation is upwards. This is the coordinate system convention used by ATK for SuperCollider

SpatDIF use a navigation-based coordinate system with x-axis to the right and, y-axis pointing forward, and z-axis pointing upwards. Azimuth is $0°$ in the forward direction, and angles increase clockwise so that $90°$ azimuth is to the right. Positive elevation is upwards [4].

There are also several conventions for how encoded Ambisonic signals are represented. B-format recordings made with one type of encoding and played back using another will have severe mismatches in amplitude levels and channel order.

Classical *Furse-Malham* encoding can be used for up to third-order signals, and seem to be the preferred representation in musical applications. First order Furse-Malham encoded signals adhere to acoustics coordinate conventions. The W component is the pressure (omni or mono) component of the signal, W is the pressure gradient component in the forward-backwards direction, Y is the pressure gradient component in the left-right direction (the stereo component of an MS signal) and Z is the up-down pressure gradient component. Furse-Malham encoding is the preferred format for all of the Ambisonic plugins discussed is section 1.4 except ambiX, and Ambisonic recordings done using SoundField microphones also use this format.

Normalised 3D formulas (N3D) and *semi-normalised 3D formulas (SN3D)* can be used for higher order encoding and decoding [6], also beyond TOA. N3D component signals go beyond unity $(-1, 1)$ and can only be properly stored in sound file formats that supports floating-points, whereas SN3D component signals stay within unity.

ATK for Reaper uses First order Furse-Malham encoded signals throughout. This ensures interoperability between ATK for Reaper and other sources and processors of Ambisonic signals such as ATK for SuperCollider or recorded Ambisonic signals, as well as most other available plugins.

ATK for Reaper strives to provide a consistent and intuitive interface towards scene description parameters such as direction of sources, speakers and spatial transforms, regardless of whether the user is accustomed with acoustics or navigational coordinate systems. For this reason Cartesian coordinates are avoided throughout the plugin suite, and whenever possible the plugins offers graphical user interfaces to avoid ambiguities.

However, due to current limitations in the Reaper JSFX API, there are some considerations to be made with respect to description of azimuths. The desired behaviour is that azimuths increase anti-clockwise. At the same time it is also preferable that for sources coming from the front half circle, moving the azimuth slider to the right results in the direction of the sound also moving clockwise to the right, similar to what happens when moving a regular stereo pan pot. The Harpex and Blue Ripple plugins both functions this way. In order to achieve this, horizontal azimuth sliders will need to implemented with increasing values to the left. This is however currently not supported in Reaper JSFX, and for the time being ATK for Reaper uses azimuth values described according to a navigational coordinate system. If Reaper in the future is updated to support sliders with increasing values to the left, this design decision is likely to be reconsidered, and changed to use acoustic conventions.

3.2 Graphical user interfaces in spatial transforms

When a mono source is encoded and the resulting B-format signal is exposed to subsequent matrix-based spatial transforms as discussed in section 4.2, the resulting B-format signal W, X, Y, Z relates to the original mono signal $s(t)$ as:

$$W(t) = k_w s(t)$$
$$X(t) = k_x s(t)$$
$$Y(t) = k_y s(t) \quad (1)$$
$$Z(t) = k_z s(t)$$

The four coefficients $k_{w,x,y,z}$ indicates four degrees of freedom, and relates to gain g, azimuth ϕ, elevation θ and the degree of directness γ (versus omnipresence, as discussed in section 4.2.2) as:

$$k_w = g\sqrt{\frac{1 + sin(\gamma)}{2}}$$
$$k_x = g\sqrt{1 - sin(\gamma)}\cos\phi\cos\theta$$
$$k_y = g\sqrt{1 - sin(\gamma)}\cos\phi\sin\theta \quad (2)$$
$$k_z = g\sqrt{1 - sin(\gamma)}\sin\phi$$

It follows that if the coefficients $k_{w,x,y,z}$ of the transformed signal are known, gain, azimuth, elevation and directness can be calculated as:

$$\phi = \text{atan2}(k_y, k_x)$$
$$\theta = \text{atan2}(k_z, \sqrt{k_x{}^2 + k_y{}^2})$$
$$g = k_w\sqrt{\frac{2}{1 + \sin\gamma}} \quad (3)$$
$$\gamma = \arcsin\left(\frac{2k_w{}^2 - (k_x{}^2 + k_y{}^2 + k_z{}^2)}{2k_w{}^2 + (k_x{}^2 + k_y{}^2 + k_z{}^2)}\right)$$

This is used in the graphical user interface (GUI) of several of the plugins discussed in section 4.2 to analyse and illustrate the effect of the spatial transforms. The initial $k_{w,x,y,z}$ coefficients are calculated independently for an array of points distributed at equidistant angles in the horizontal plane, and then exposed to the same transform as the audio signal will be. The transformed coefficients are analysed according to equation 3 and displayed as a series of monochrome circles in the GUI against a dark circle indicating the sound field hemisphere as seen from above. Colour hue depends on gain with 0 db being orange, signals with increased gain becoming red, and signals with reduced gain venturing towards green and blue, as can be seen in figure 6. Elevation is indicated by adjustments to lightness and saturation, so that the circle gets brighter in the upper and darker in the lower hemisphere. Additionally the radius of the circle increases slightly with increasing elevation, as seen in figure 2. Azimuth is indicated by angular direction of the centre of the circle relative to the display hemisphere. The distance from the centre of the hemisphere depends on elevation and directness combined, so that a transformed source that is fully directional and with 0° elevation displays as a small non-transparent circle located at the outer edge of the hemisphere. With decreasing directness, or increasing absolute elevation the centre of the circle moves inwards, and a transformed source with an elevation of ±90° and/or no directness (an omnipresent

signal) will be located in the centre of the displayed hemisphere. Degree of directness affects the radius of the circle as well, but in a more radical way than elevation. A fully directional transformed source will be indicated as a small non-transparent circle, and with decreasing directness radius increase so that it equals that of the display hemisphere for a omnipresent source. Additionally the circle becomes more transparent with decreasing directness. This way the visualisation helps communicate that an omni-present source does not sound as if it is a focused source located at the centre, rather the sound will appear to be arriving from all directions. Examples of less directional sources can be found in figures 3-5.

Depending on the plugin, one or more bright blue knobs in the GUI can be used to control azimuth and degree of transformation.

4. ATK REAPER: THE SET OF PLUGINS

The source code for ATK for Reaper is implemented as a set of Reaper JSFX source files, one for each plugin, and a shared library file containing mathematical constants and conversions, matrix operators and graphics calls used by several of the plugins, ensuring a DRY (Don't Repeat Yourself) programming approach. The various plugins are sorted into subfolders by categories *Encode*, *Transform* and *Decode*, making it easy for the user to understand the scope of each of the plugins when selecting a new plugin in Reaper. The following sections present the available plugins.

4.1 Encoders

Most users approaching Ambisonics are usually presented with two avenues to author an Ambisonic soundfield: capture a natural soundfield directly with a Soundfield microphone, or author a planewave from a monophonic signal. The ATK provides a wide palate of authoring tools.

The *PlanarWave* plugin is a classic encoding of a mono source as a directional planewave, where the direction (azimuth and elevation) of the planewave can be set. *Omni* is a plugin encoding a mono signal as an omnidirectional soundfield. An omnidirectional soundfield can be regarded in two ways: A soundfield with an infinite number of planewaves arriving in all directions, or a soundfield with no directions. In a well aligned, dampend studio environment, this usually sounds "in the head", while in concert hall listening it usually appears as omnipresent. To control the soundfield, spatial transforms can be applied using the *FocusPressPushZoom* plugin (see section 4.2.3) to either "push" or "focus" an omnidirectional soundfield into a planewave, giving the sound field an angle of arrival. Strictly this plugin is redundant in Reaper, as it is sufficient to route the mono source channel to the first W channel of the next plugin in the processing chain. It is included for completeness, and because the inclusion of this plugin in the effects chain helps exposing the logics behind the processing graph assembled for the track. The *Stereo* plugin encodes the left and right channels as two planewaves coming from left and right directions. The angular spread between the two waves is parameterised. Spatial transforms

such as *RotateTiltTumble* (section 4.2.1) may be used to alter the direction of the planewaves. Additional upmixers are provided for a number of virtual speaker arrays; *Quad*, *5.0* and *7.0* for the standard ITU layouts, *Pantophonic* for 2D and *Periphonic* for 3D.

A few encoders remain to be implemented: The *PseudoInverse* encoding technique provides the greatest flexibility, and can be used with both microphone arrays and synthetic signals. In the absence of a Soundfield microphone, this feature gives the opportunity to deploy real-world microphone arrays (omni, cardioid, etc.) to capture natural soundfields. With synthetic signals, pseudoinverse encoding is usually regarded as the method of choice to generate spatially complex synthetic Ambisonic images. In combination with the ATK's imaging tools these can then be compositionally controlled as required. *ZoomH2* adapts *PseudoInverse* for this popular portable recorder.

The rest of the encoders that remain to be ported depend on kernel convolution. *Super* is an Ambisonic super stereo encoder, the classic method for encoding stereophonic signals into B-format. The *UHJ* encoder offers access to numerous published recordings for periphonic (2D) audition. *Spread* encodes a monophonic signal into B-format by smoothly rotating the signal across the soundfield sphere, by frequency, while *Diffuse* randomise the phase of the incoming monophonic signal across the soundfield sphere, by frequency.

4.2 Imaging transforms

For the artist, the real power of the ATK is found in the imaging transforms. These are spatial domain filters which reorient, reshape or otherwise spatially filter an input soundfield. As discussed in section 1.4 there are several plugins available for spatial transforms. The ATK provides a much wider and comprehensive toolset.
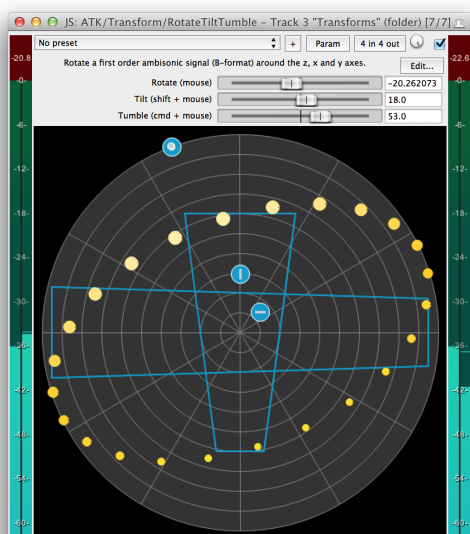
4.2.1 RotateTiltTumble

Figure 2: Rotate, Tilt and Tumble transform.

The *RotateTiltTumble* plugin is a FOA multi-axes rotation transformer, with *Rotation*, *Tilt* and *Tumble* transforms provided in the sequence indicated by the plugin name. If the user wants to change the order of transformations, it is possible to daisy-chain two or more instances of the plugin. This transform does not affect directness of the signal, and the screenshot in figure 2 demonstrates how variations in saturation, lightness and radius of the displayed transformed sources serve to illustrate their vertical position. The separate effect of the *Tilt* and *Tumble* transformations is indicated by the two blue planes in the interface.

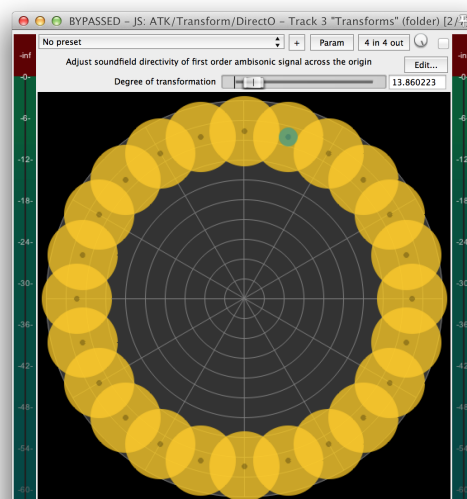### 4.2.2 Direct and DirectO

Figure 3: DirectO transform plugin interface.

The *DirectO* plugin (figure 3) adjusts the sound field directivity of a First order Ambisonic signal across the origin. It is a spatial low-pass filter; with increasing degree of transform, the signal becomes less directional, and with a transform of $90°$ the signal becomes omnipresent.

Similarly, the *Direct* plugin (figure 4) adjusts the sound field directivity of a FOA signal across an arbitrary plane.

4.2.3 FocusPushPressZoom and Dominate

The *FocusPushPressZoom* plugin provides a unified interface to the four different spatial transforms *Focus*, *Press*, *Push* and *Zoom*. The degree of spatial transform is expressed as angles in the range from $0°$ to $90°$. At $0°$ no transform is applied, while at $90°$ all four transforms cause the soundfield to collapse to a single mono planewave. All four transforms shift the sound field in a preferred direction by altering directness, gain and direction of the incoming signals depending on their original directions, but the four transforms differ in terms of what mathematical transform is applied. The results will differ subtly or more distinctly depending on which of the transforms is applied. Having all four transforms available in the same plugin makes this an expressive quality that can be explored in the creative process. Two of the transforms can be seen in figures 5 and 6.
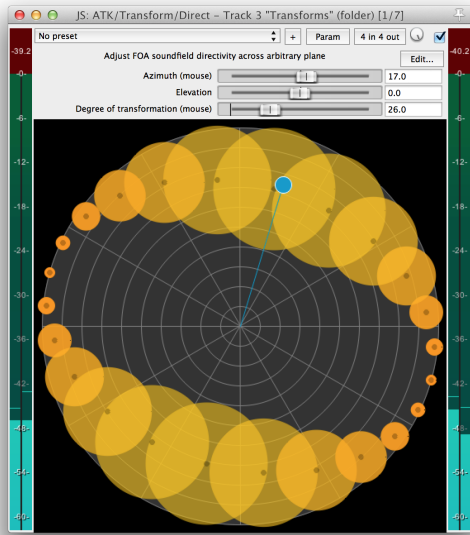
Figure 4: Direct transform plugin interface.

Figure 5: Push transform plugin interface.

The *Dominance* plugin increases the gain of signals coming from the preferred direction while decreasing the gain of signals originating from the opposite side. In the Reaper plugin gain in the preferred direction can be boosted by up to 24 dB. While the *Focus*, *Push*, *Press*, and *Zoom* transforms do not generally have any major impact on the overall sound level of the soundfield, *Dominance* can result in major alterations to sound levels. For this reason, as well as due to the difference in how the degree of transform is expressed (decibels rather than angles) the *Dominance* transform has not been integrated into the same plugin as the other four transforms.

4.2.4 Mirroring

The *Mirror* plugin mirrors the soundfield across an arbitrary plane, while the *MirrorO* plugin mirrors the sound field across Origo.

4.3 Near-field corrections

Proximity facilitates the introduction of the proximity effect to encoded signals. At extremes, the proximity effect introduces a strong bass boost, as well as phase differences. The proximity effect can be an important contributor to perceptions of nearness. Distance is described in meters, and to prevent bass boost from getting out of hand, the lower limit is set to 10 cm (0.1 m).

NearFieldCompensation facilitates the reduction or removal of the proximity effect from encoded signals. Near-field compensation is usually used in conjunction with decoders to compensate for the distance of loudspeakers on playback in loudspeaker rigs with smaller diameter. Additionally, NearFieldCompensation can also be used to reduce the proximity effect found in nearfield recordings.

Proximity and *NearFieldCompensation* respectively apply an integrator or high pass filter [3] to the first order components of the encoded signal. The *Proximity* filter un-

---

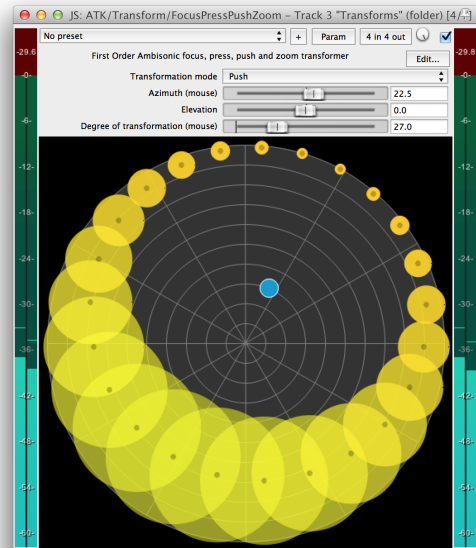[3] A 1th-order high pass filter is the reciprocal of a 1th-order integrator.

does *NearFieldCompensation* given the same distance argument.

4.4 Transformations between A-format and B-format

Balance and phase relations between the four channels of the FOA signal are critical for the spatial information. If one or more encoded B-format channels are processed using filters, distortion, delays, or some other audio effect, the spatial information of the encoded signal will most likely be distorted or otherwise lost, often resulting in distorted or unintended spatial results. This will tend to be the case if processing effects the individual B-format channels individually, i.e., if un-synced changes in gain or phase occur.

The *BtoA* plugin performs matrix-based decoding of the B-format signal to A-format, while *AtoB* performs a similar encoding of A-format into B-format. When used in conjunction, the intermediate A-format signal is well-suited for effect processing which preserve the spatial encoding found in the original input B-format signal.

On its own, *AtoB* is often used to author immersive, periphonic (3D) decorrelated soundfields.

4.5 Decoders

Perhaps one of the most celebrated aspects of the Ambisonic sound technique has been its design as a hierarchal reproduction system, able to target a number of varying loudspeaker arrays. The ATK provides a wide palate of optimised decoders.

The *Mono* virtual microphone decoder returns a single monophonic channel, and can be used to "listen in" to the soundfield at the specified azimuth and elevation. The soundfield may be decoded to *Stereo* using a pair of virtual microphones. *Quad* is an optimised quadraphonic decoder with variable loudspeaker angle, *5.0* uses Bruce Wiggins optimised ITU 5.0 recorders [40], *Pantophonic* is a regular
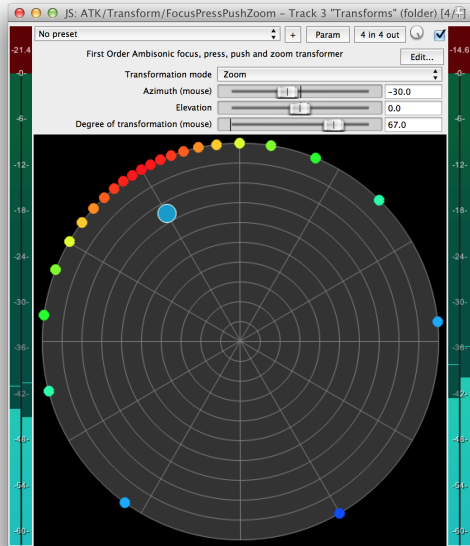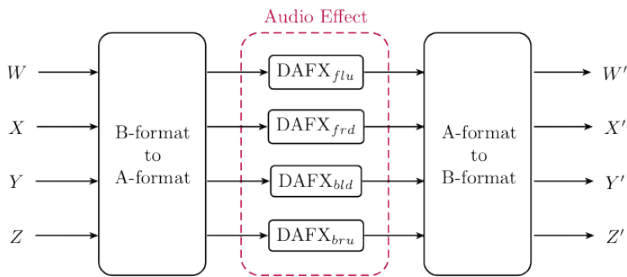
Figure 6: Zoom transform interface.

Figure 7: Process effects in A-format.

2D polygon decoder and *Periphonic* is a regular cylindrical decoder for 3D dual ring. *Diametric* is Gerzon's classic decoder suitable for varied periphonic and pantophonic loudspeaker arrays [41]. While the regular decoders will be suitable for many users, diametric decoding enables the greatest flexibility, and allows the user to design substantially varying semi-regular arrays suitable for a wide variety of playback situations. Meeting all the criteria outlined by Gerzon to qualify as Ambisonic, this decoder is a good choice for full 3D critical studio listening.

The *UHJ* decoder and binaural decoders, using measured and synthetic HRTFs, requires kernel convolution, and remains to be implemented. While the virtual microphone stereophonic decoder is very easy and convenient, for production work the authors advise using the *UHJ* decoder, once the port has been accomplished.

The relevant decoders supports several decoding modes: The 'dual' decoder is a dual-band psycho-acoustically optimised decoder, and the optimum choice for small scale studio or domestic settings [42]. 'single' is suitable for larger, mid-scale environments. 'velocity' returns 'strict soundfield' (aka 'basic') decoding, and is not preferred for First order Ambisonics in most circumstances. 'controlled' returns 'controlled opposites' decoding (aka 'in phase'), which is often preferred in large-scale, concert environments.

4.6 Licensing and distribution

ATK for SuperCollider3 is distributed as open-source using the GNU General Public License Version 3 (GNU GPL). This license is incompatible for use with proprietary software such as Reaper, and for this reason ATK for Reaper is distributed using the GNU Lesser Public License Version 3 (GNU LGPL). ATK for Reaper requires Reaper 4.61 or newer, and will be available for download from the Ambisonic Toolkit web site in the near future [43].

5. DISCUSSION

While there is a well-established workflow for stereo on DAWs, options have been more limited when working with Ambisonics. From a compositional viewpoint, Ambisonic field recordings sometimes can leave the sonic artist with an impression of 'take it or leave it', as the sound image can be spectrally as well as spatially quite 'full'. In addition to rich sets of encoders and decoders, Ambisonic Toolkit offers intriguing possibilities for spatial transforms of the soundfield image, and for this reason it has been attractive to port ATK to a plugin format making it available in DAW workflows in addition to SuperCollider. When porting ATK the prime question is whether to implement plugins in C++ using plugin architectures such as VST or AudioUnit. If doing this, it seems beneficial to use a intermediate library such as Faust, Juce or wdl-ol, but it will still require substantial overhead in terms of having to compile and test the plugins for multiple processors and platforms. At the same time the plugins are vulnerable to future changes in OS and plugin platforms. At the end of 2013 Steinberg stopped maintenance and distribution of the VST2 SDK, instead requiring future developers to use VST3. The future of AudioUnit currently is somewhat uncertain, as the SDK has not been updated since 2007, and AU plugins can not be distributed via AppStore.

Reaper is a reasonably priced and flexible DAW for spatial sound, popular among many composers and sonic artists working with spatial sound, and provides the script-based language JSFX for plugins. Using JSFX there is no need to compile, and plugins immediately work with all platforms and processors supported by Reaper. As long as Reaper continues to be available, maintenance of the plugins are not expected to be demanding, as Reaper itself will take care of changes to e.g., operation systems. Using JSFX has been speeding up the development process substantially, and development has been able to primarily focus only on the specific processing and interfacing. One disadvantage is that graphics in plugins with custom GUIs seem to be demanding on system resources, and the program indicate major CPU overhead when plugin GUIs are open. However, once the GUI window is closed, the plugins do not seem to require any extraneous processing resources.

6. REFERENCES

[1] ITU, "Recommendation BS. 775: Multi-channel stereophonic sound system with or without accompanying picture," International Telecommunications Union, Tech. Rep., 1993.

[2] A. Berkhout, D. Vries, and P. Vogel, "Acoustic Control by Wave Field Synthesis," *J. Acoust. Soc. Am.*, vol. 93, pp. 2764–2778, 1993.

[3] Dolby Laboratories, Inc, "Dolby Atmos. Next-generation audio for cinema. White paper Issue 3," Dolby Laboratories, Inc, Tech. Rep., 2014. [Online]. Available: http://www.dolby.com/uploadedFiles/Assets/US/Doc/Professional/Dolby-Atmos-Next-Generation-Audio-for-Cinema.pdf

[4] N. Peters, T. Lossius, and J. Schacher, "The Spatial Sound Description Interchange Format: Principles, specification, and examples," *Computer Music Journal*, vol. 37, no. 1, pp. 11–22, Spring 2013.

[5] M. A. Gerzon, "Ambisonics in multichannel broadcasting and video," *J. Audio Eng. Soc.*, vol. 3, no. 11, pp. 859–871, Nov. 1985.

[6] J. Daniel, "Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia," Ph.D. dissertation, Université Paris 6, 2001.

[7] K. Farrar, "Soundfield microphone: design and development of microphone and control unit," *Wireless World*, pp. 48–50 (Oct.), 99–103 (Nov.), 1979.

[8] E. Deleflie. (2014) Ambisonia. [Online]. Available: http://ambisonia.com/

[9] D. Darcourt. (2014) SurLib. [Online]. Available: http://www.surround-library.com/

[10] IOsONO GmbH, "Spatial audio workstation 2 operation manual," IOsONO GmbH, Tech. Rep., 2012.

[11] Dolby Laboratories, Inc, "Authoring for Dolby Atmos cinema sound manual," Dolby Laboratories, Tech. Rep., 2013. [Online]. Available: http://www.dolby.com/uploadedFiles/Assets/US/Doc/Professional/Authoring_for_Dolby_Atmos_Cinema_Sound_Manual(1).pdf

[12] Cockos Inc. (2014) About REAPER. [Online]. Available: http://www.reaper.fm/about.php

[13] G. Francis, "Up and running: A REAPER user guide," Cockos, http://reaper.fm/userguide.php, Tech. Rep., March 2014.

[14] V. Pulkki, "Virtual sound source positioning using vector base amplitude panning," *Journal of the Audio Engineering Society*, vol. 45(6), pp. 456–466, June 1997. [Online]. Available: http://lib.tkk.fi/Diss/2001/isbn9512255324/

[15] T. Lossius, P. Baltazare, and T. de la Hogue, "DBAP - Distance-based amplitude panning," in *Proceedings of the International Computer Music Conference 2009, Montreal. International Computer Music Association.*, 2009.

[16] B. Wiggins. (2012) Teaching. [Online]. Available: http://www.brucewiggins.co.uk/?page_id=215

[17] Steinberg Media Technologies GmbH. (2014) 3rd party developer area. [Online]. Available: http://www.steinberg.net/en/company/developer.html

[18] Apple Inc., "Audio Unit programming guide," Apple Inc., Tech. Rep., 2014. [Online]. Available: https://developer.apple.com/library/mac/documentation/MusicAudio/Conceptual/AudioUnitProgrammingGuide/Introduction/Introduction.html

[19] Microsoft. (2014) DirectX software development kit. [Online]. Available: http://www.microsoft.com/en-us/download/details.aspx?id=6812

[20] R. Furse. (2007) Linux audio developer's simple plugin API (LADSPA). [Online]. Available: http://www.ladspa.org/

[21] LV2 developers. (2014) LV2. [Online]. Available: http://lv2plug.in/

[22] Ableton. (2014) Max for Live. [Online]. Available: https://www.ableton.com/en/live/max-for-live/

[23] Cockos Inc. (2014) JSFX programming reference. [Online]. Available: http://www.reaper.fm/sdk/js/js.php

[24] R. M. S. Ltd. (2014) About juce. [Online]. Available: http://www.juce.com/about-juce

[25] O. Larkin. (2014) Wdl / iplug - oli larkin edition. [Online]. Available: https://github.com/olilarkin/wdl-ol

[26] J. Smith III, *Audio Signal Processing in Faust*, Center for Computer Research in Music and Acoustics (CCRMA), Department of Music, Stanford University, Section 4, 2012. [Online]. Available: http://ccrma.stanford.edu/~jos/aspf/aspf.pdf

[27] Makira. (2014) Sonicbirth v2 - alpha. [Online]. Available: http://sonicbirth.com/

[28] J. McClintock. (2011) SynthEdit - make your own synth. [Online]. Available: http://www.synthedit.com/

[29] B. Wiggins. (2010) Wigware. [Online]. Available: http://www.brucewiggins.co.uk/?page_id=78

[30] D. Courville. (2012) B2x plug-in suite for mac osx. [Online]. Available: http://www.radio.uqam.ca/ambisonic/b2x.html

[31] M. Kronlachner, "Ambisonics plug-in suite for production and performance usage," in *Proceedings of the Linux Audio Conference, Graz*, 2013.

[32] M. Kronlachner and F. Zotter, "Spatial transformations for the enhancement of ambisonic recordings," in *Proceedings of the 2nd International Conference on Spatial Audio, Erlangen*, 2014.

[33] C. Nachbar, F. Zotter, E. Deleflie, and A. Sontacchi, "Ambix - a suggested ambisonics format," in *Proceedings of the Ambisomics Symposium, Lexington*, 2011.

[34] M. Kronlachner. (2014) mcfx v0.2 – multichannel audio plug-in suite. [Online]. Available: http://www.matthiaskronlachner.com/?p=1910

[35] IRCAM and Flux, "Spat multiformat room acoustics simulation and localization processor. manual," IRCAM and Flux, Tech. Rep., 2010.

[36] S. Berge and N. Barrett, "High angular resolution planewave expansion," in *Proceedings of the 2nd International Symposium on Ambisonics and Spherical Acoustics, Paris*, 2010.

[37] Blue Ripple Sound Inc. (2014) Pro audio products. [Online]. Available: http://www.blueripplesound.com/product-listings/pro-audio

[38] J. Anderson, "Introducing...the Ambisonic Toolkit," in *Proceedings of the Ambisonics Symposium, Graz*, 2009.

[39] J. Anderson and J. Parmenter. (2011) The ambisonic toolkit. [Online]. Available: http://www.ambisonictoolkit.net/

[40] B. Wiggins, I. Paterson-Stephens, V. Lowndes, and S. Berry, "The design and optimisation of surround sound decoders using heuristic methods," in *Proceedings of UKSIM 2003: Conference on Computer Simulation*, Cambridge, England, 2003.

[41] M. A. Gerzon, "Practical periphony: The reproduction of full-sphere sound," in *Proceedings of the 65th Audio Engineering Engineering Society Convention*, London, 1980.

[42] A. Heller, R. Lee, and E. Benjamin, "Is my decoder ambisonic?" in *Proceedings of the 125th Audio Engineering Society Convention*, San Francisco, 2008.

[43] The Ambisonic Toolkit developers. (2014) The Ambisonic Toolkit : Tools for soundfield-kernel composition. [Online]. Available: http://www.ambisonictoolkit.net/