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

The abstract should be placed at the top left column and should contain about 150-200 words.

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-oriented 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 [2]. SpatDIF, the Spatial Sound Description Interchange Format, is an important example offering a semantic and syntactic specification for storing and transmitting such spatial audio scene descriptions [3].

Ambisonics encodes audio sources into a speaker-independent representation of the sound field called B-format [4]. 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 might be first or higher order [5]. 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 first order, and free or commercial B-format recordings are available as sound effects libraries [6, 7, 8].

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, and ambisonic is supported in all major real-time audio programming environments such as Csound, Max, Pd and SuperCollider, often as 3rd party extensions to the 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-oriented 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 [9], and can be used with Iosono Core hardware systems for wave-field 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 [10].

The DAWs discussed so far are either limited with respect to surround sound abilities, or expensive and proprietary software and hardware systems for object-oriented spatial scene authoring and rendering. In contrast Reaper is a reasonably priced and flexible DAW for spatial sound [11]. Reaper supports tracks of up to 64 channels, 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, using a spatialisation algorithm based on principles resembling VBAP [12] as well as DBAP [13]. Reaper is well-suited for ambisonic as well, and is a preferred DAW for many composers within the ambisonic community [14].

1.3 Plugin technologies

Several plugin specifications enable development of 3rd party audio instruments and effects that can be used with multiple hosts. Virtual Studio Technology (VST) is a plugin interface specification and SDK for Windows, Mac OSX and Linux by Steinberg [15]. It is a popular format for commercial and freeware VST 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 [16]. In a similar way DirextX can be used for development of Windows-only plugins [17]. Linux Audio Developers Simple Plugin API (LADSPA) is a GNU LGPL plugin architecture mostly used on the Linux platform [18], and is gradually superseded by LADSPA version 2 (LV2) [19].

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 plugins format that work with their software and hardware exclusively, as both Real-Time AudioSuite (RTAS) and TDM have been discontinued and replaced by Avid Audio eXtension (AAX). Ableton and Cycling'74 jointly offer Max for Live, enabling the use of Max patches as devices in Ableton Live [20]. JSFX is a text-based scripting language for programming audio-oriented effects compiled on the fly for Cockos Reaper [21, 11].

* Juce * wdl-ol * OpenFrameworks * SonicBirth * SynthEdit (for Windows) * Faust

1.4 Plugins for ambisonic processing

* Bruce Wiggins * Danielle Courville * Flux Spat * ambit 1st, 3rd and 5th order (have been temporarily withdrawn for licensing reasons) * Hoalibrary for Faust * Harpex * Blue Ripple TOA plugins ** BlueRipple Harpex 1st to 3rd order spatial upsampler * (Wikipedia overview of software)

2. AMBISONIC TOOLKIT

The Ambisonic Toolkit (ATK) brings together a number of tools and transforms for working with ambisonic surround sound [22, 23]. 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 not focused on the problem of auralization and/or room modelling. Auralization is a complex problem beyond the scope of the fundamental algorithms of the ATK. It is worth noting, however, with the toolset provided, successful room modelling may be implemented. Interestingly enough, many composers of acousmatic music often do not move beyond the problem of auralization or room modelling.

The ATK, by addressing the holistic problem of creatively controlling a complete soundfield, 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, therefore taking advantage of the model the ambisonic technology presents. This is viewed to be the *idiomatic* approach for working with the ambisonic technique. Restated, the model of the ATK is a sound-field sound-image model rather than a sound-object sound-scene model.

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 recent years ATK has primarily been distributed as a version for SuperCollider3 [22]. Some of the underlying ideas of ATK has also been incorporated into the Blue Ripple Sound third order ambisonic plugins [24].

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 and y-axis to the left in the horizontal plane, 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 forward in the horizontal plane, and z-axis in the upwards direction. Azimuth is 0° in the forward direction, and angles increase clockwise so that 90° azimuth is to the right. Positive elevation is upwards.

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-Mulham 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, 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 [5], and as an example the ICST ambisonic externals for Max supports up to 11th order using N3D and SN3D [25]. 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. Sources are encoded as first order Furse-Malham B-format, all processing of encoded signals assumes Furse-Malham B-format signals, and this is also the assumed encoded incoming signal for decoding plugins. 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 used to 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. When describing azimuths, the navigational coordination system is used, as it is considered to give a more consistent user interaction with the interface. In Reaper, when using the faders provided as a standard interface for JSFX plugins, moving the slider to the right increases the associated value. Azimuth angles are generally in the $(-180°, 180°)$ range, and defaults to 0°. Within the $(-90°, 90°)$ range and using acoustics anti-clockwise conventions, an increase of azimuth value would cause the direction to move to the left while the slider would move to the right. With a navigational coordinate system these inconsistencies can be avoided. Another important motivation for the choice of clock-wise positive azimuths, is that description is more consistent with other such as the standard stereo panner, surround sound panners in most DAWs, including the ReaSurround plugin, rotation of the virtual microphone in the SoundField SPS200 Surround Zone plugin, Flux Spat

Exceptions: TOA and Harpex but even in these automation data values increase in the clockwise direction.

WigWare - strictly ambisonic

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:

$$\begin{aligned} W(t) &= k_w s(t) \\ X(t) &= k_x s(t) \\ Y(t) &= k_y s(t) \\ Z(t) &= k_z s(t) \end{aligned} \tag{1}$$

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:

$$\begin{aligned} k_w &= g\sqrt{\frac{1+sin(\gamma)}{2}} \\ k_x &= g\sqrt{1-sin(\gamma)}\cos\phi\cos\theta \\ k_y &= g\sqrt{1-sin(\gamma)}\cos\phi\sin\theta \\ k_z &= g\sqrt{1-sin(\gamma)}\sin\phi \end{aligned} \tag{2}$$

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:

$$\begin{aligned} \phi &= \text{atan2}(k_y, k_x) \\ \theta &= \text{atan2}(k_z, \sqrt{{k_x}^2 + {k_y}^2}) \\ g &= k_w\sqrt{\frac{2}{1+\sin\gamma}} \\ \gamma &= \text{arcsin}\Big(\frac{2{k_w}^2 - ({k_x}^2 + {k_y}^2 + {k_z}^2)}{2{k_w}^2 + ({k_x}^2 + {k_y}^2 + {k_z}^2)}\Big) \end{aligned} \tag{3}$$

This is used in the graphical user interface (GUI) of several of the plugins discussed in section 4.2 to 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 5. 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 1. 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 $\pm 90°$ 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 and 4.

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

3.3 Working with mono and stereo sources in Reaper

4. ATK REAPER: IMPLEMENTATION

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. The following sections present the available plugins.

4.1 Encoders

Omni encodes 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.

PlanarWave encodes a mono source as a directional planewave, where the direction (azimuth and elevation) of the planwave can be set.

The *Stereo* plugin encodes the left and right channels as two planewaves coming from left and right direction. The angular spread between the two waves is parameterised. Spatial transforms such as *RotateTiltTumb* (section 4.2.1) may be used to alter the direction of the planewaves.

4.2 Spatial transforms

4.2.1 Rotate, Tilt and Tumble

The *RotateTiltTumble* plugin is a first order ambisonic 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 1 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

The *DirectO* plugin (figure 2) 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

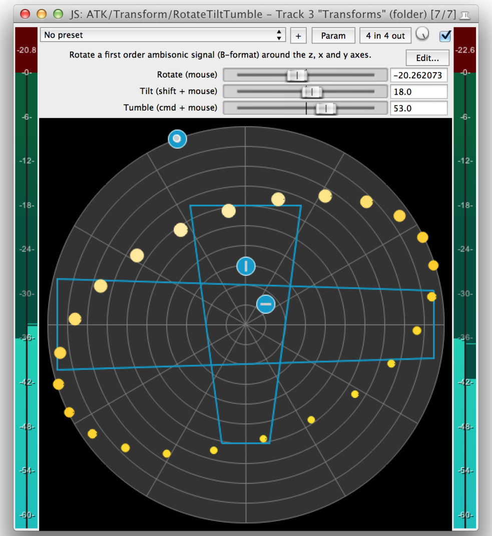

Figure 1: Rotate, Tilt and Tumble transform.

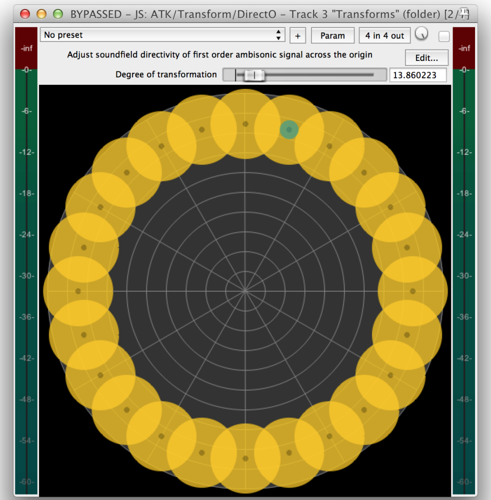

Figure 2: DirectO transform plugin interface.

transform, the signal becomes less directional, and with a transform of 90° the signal becomes omnipresent.

Similarly, the *Direct* plugin (figure 3) adjusts the sound field directivity of a first order ambisonic signal (B-format) across an arbitrary plane.

4.2.3 Focus, Pull, Push, Zoom and Dominate

The *FocusPushPressZoom* plugin provides a unified interface to four different First Order Ambisonic (FOA) focus, press, push and zoom transformer

Apply focus to a first order ambisonic signal (B-format) along an arbitrary axis.

Apply press to a first order ambisonic signal (B-format) along an arbitrary axis.

Apply dominance to a first order ambisonic signal (B-format) along an arbitrary axis. Apply zoom to a first order ambisonic signal (B-format) along an arbitrary axis.

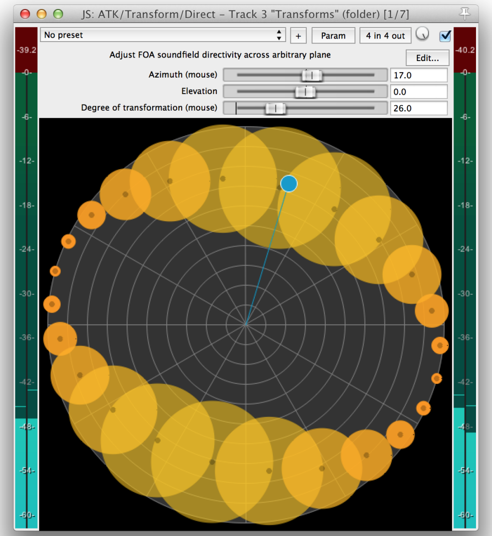

Figure 3: Direct transform plugin interface.

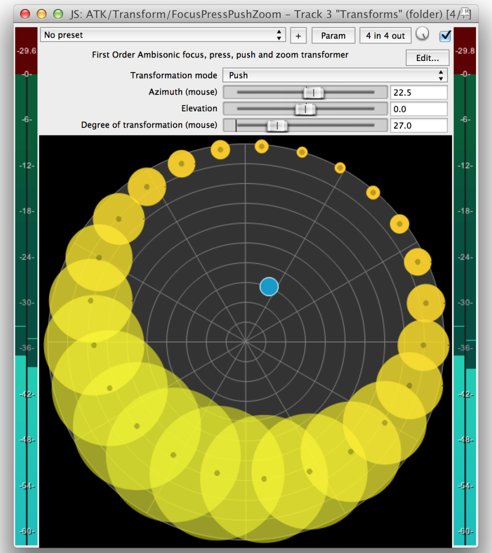

Figure 4: Push transform plugin interface.

Description: the *Dominance* gain, in dB, applied on the x-axis. Positive values increase the gain at front center to +gain, while decreasing the gain at back center to -gain, simultaneously distorting the image towards front center. Negative values of gain invert this distortion, distorting the image towards back center. The default, 0, results in no change.

Description: the angle of distortion in radians. Positive values focus on the front center of the image, and at pi/2 collapse the soundfield to mono, reducing the gain at back center to -inf dB. Negative values focus on back center. The default, 0, results in no change.

Description: the angle of distortion in radians, from -pi/2 to pi/2. Positive values push to the front center of the image, and at pi/2 collapse the soundfield to mono. Negative values push to back center. The default, 0, results in no change.

4.2.4 Mirroring

4.3 Near-field corrections

DISCUSS: I'd like to fuse these two into one plugin, with an additional pop-up menu parameter to choose which one to use. Alternatively, we could find a way of using only one slider, and use the "negative range" to draw the source nearer, and the "positive range" for moving the source away.

4.4 Transformation between A-format and B-format

DISCUSS: Can Joseph help with with this subsection, so that the philosophy of it can be communicated clearly?

4.5 Monitoring

4.6 Decoders

DISCUSS: I need to get a more firm understanding of what speaker layout etc. each of the encoders works with.

DISCUSS: Can all of the matrix decoders (mono - stereo - quad - pantophonic - periphonic - diametric) be wrapped into one plugin, with a parameter for choosing between them? If so, would it be useful to have some kind of visualisation of the speaker layout in a GUI in order to make it more intuitive to understand what the different layouts are?

DISCUSS: IMHO the best thing would be to implement the shelf filter as a separate plugin, that is unless all of the decoders can be catered for in the same plugin. It's more DRY (don't repeat yourself) and also implies that the shelf filter is available for use in other contexts as well.

DISCUSS: The tickets in tracker says "There is also a question as to whether NFC should be included for user convenience.". IMHO it's better to keep that a separate plugin.

DISCUSS: I need more details on how HRTF and UHJ decoders work in order to implement them properly

4.7 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 4.60 or newer, and will be available for download from the Ambisonic Toolkit web site in the near future [26].

5. DISCUSSION

* Maintenance overhead when developing C/C++ plugins: Compiling for multiple processors and platforms * Future of several plugin standards can be questioned, e.g. AU * If doing this, it seems beneficial to use a intermediate library such as Faust, Juce or wdl-ol. * JSFX: ** Pro: *** No compiling required, works with all platforms and processors supported by Reaper *** Less vulnerable to changes in the plugin specifications themselves. At the end of 2013 Steinberg stopped maintenance and distribution of the VST

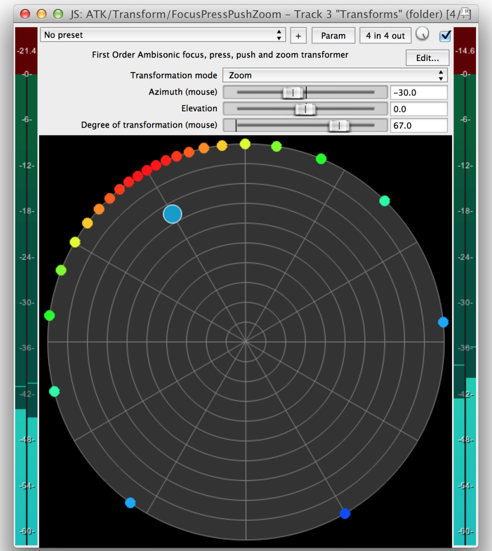

Figure 5: Zoom transform interface.

2 SDK, instead requiring future developers to use VST 3. The future of AU is somewhat uncertain, as the SDK has not been updated since 2007, and AU plugins can not be distributed via AppStore. *** Can focus only on the specific processing and interfacing ** Con: *** Only work with Reaper, but then again it is establishing itself as a preferred DAW within the ambisonic community. *** Graphics seems very slow, big CPU overhead when plugin GUIs are open

Acknowledgments

The porting of ATK to Reaper started in 2013 as part of the process of developing new artistic works for an exhibition carried out within the framework of the artistic research project Re:place at Bergen Academy of Arts and Design, funded by the Norwegian Artistic Research Programme. Stian Remvik has provided valuable feedback on user interface design.

6. REFERENCES

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

[2] 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

[3] 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.

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

[5] 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.

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

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

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

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

[10] 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

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

[12] 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/

[13] 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.

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

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

[16] 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

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

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

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

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

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

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

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

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

[25] J. C. Schacher, "Seven Years of ICST Ambisonics Tools for MaxMSP – a Brief Report," in *Proc. of the 2nd International Symposium on Ambisonics and Spherical Acoustics*, 2010.

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