Audio Engineering Society

Conference Paper

Presented at the Conference on
Audio for Virtual and Augmented Reality
2016 Sept 30 – Oct 1, Los Angeles, CA, USA

Object-based spatial audio production for virtual reality using the Audio Definition Model

Chris Pike[1], Richard Taylor[1], Tom Parnell[1], and Frank Melchior[1]

[1]*BBC Research & Development*

Correspondence should be addressed to Chris Pike (chris.pike@bbc.co.uk)

ABSTRACT

This paper presents a case study of the production of a virtual reality experience with object-based spatial audio rendering using audio post-production tools and workflows. An object-based production was created using a common digital audio workstation with real-time dynamic binaural sound rendering and visual monitoring of the scene on a head-mounted display. The Audio Definition Model is a standardised meta-data model for representing audio content including object-based, channel-based and scene-based spatial audio. Using the Audio Definition Model the object-based audio mix could be exported to a single WAV file. Plug-ins were built for a game engine in which the virtual reality application and the graphics were authored to allow import of the object-based audio mix and custom dynamic binaural rendering.

1 Introduction

The importance of audio in virtual reality (VR) experiences has been recognised and much discussed in recent times. Tools for production and platforms for distribution of VR content with spatial audio are becoming widely available. Spatial audio can now be integrated into VR applications using software developer kits (SDKs) or plug-ins for common game engines and audio middleware systems. Many such tools are provided free of charge to encourage content creators to integrate spatial audio into the VR experience.

Despite such progress, there are still many improvements to be made in tools and the workflow for VR audio production. The tools for VR audio in games development are not readily compatible with common tools of sound designers and mixers, i.e. digital audio workstations (DAWs) and associated plug-ins. Integration between these different environments is currently lacking. This is particularly problematic when creating cinematic or broadcast-style VR and omnidirectional video (ODV) experiences, where the audio production team will often have little or no experience using game engines and middleware to author content.

This paper will present a description of tools developed to enable production of 3D sound for virtual reality in a professional broadcast environment. The tools were applied in the production of The Turning Forest VR experience which was premiered at the 2016 Tribeca

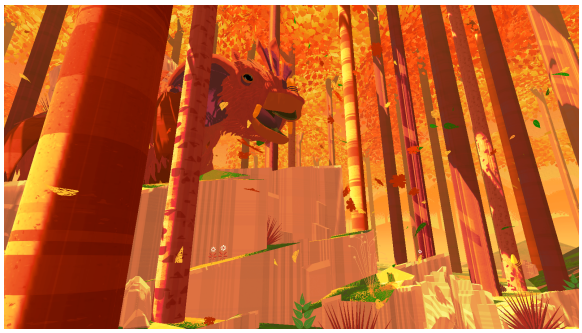

Fig. 1: In-game image from The Turning Forest

Fig. 2: 16-channel spaced hypercardioid microphone array used for forest atmosphere recording

Film Festival.

2 The Turning Forest

The Turning Forest is a narrative-led VR experience, created by BBC Research & Development and VR-TOV[1]. The aim of this production was to create a compelling immersive story using VR with a strong focus on spatial audio, to demonstrate how much it can add to the experience. It was originally created as an audio-only production as part of the freely-available S3A object-based audio drama dataset [1]. It was first mixed on a 3D loudspeaker array and then subsequently adapted for virtual reality with binaural rendering for headphones and an accompanying computer-graphics-based visual environment (see Figure 1) built in a game engine and presented in a head-mounted display (HMD).

2.1 The Story

The Turning Forest is a fantasy story, written specifically for demonstration of 3D audio. It begins in an autumnal forest where two children are playing around the viewer until one child runs away. At this point a huge monster appears to the remaining child and the viewer's role shifts, becoming the child. It becomes apparent that the monster is friendly as it smiles to reveal a beautiful musical sound and the monster's teeth become an interactive musical instrument that the viewer can play. Suddenly the forest floor fills with water and the monster lifts the child onto its back and begins to swim as the forest disappears under water. After some time they arrive onto the shore of a new frozen landscape, the winter forest. The creature disappears leaving the child alone in this unknown world as the sun sets. There are two other interactive musical components during the story. In total the piece is approximately seven minutes long.

[1] http://vrtov.com

2.2 Audio Sources

The audio production features a range of different audio sources; a studio-based narration recording, location recordings of child actors, mono and stereo sound effects, multitrack music recordings and a 16-channel forest ambience recording.

The forest ambience recording was captured using a double layered circular array of radius 2.5m. Each layer of the array consisted of 8 regularly spaced supercardioid Sennheiser MKH8050 microphones. The microphones in the lower layer were 1.35m above the ground pointed outwards and the microphones in the upper layer were 2.45m above the ground and pointed upwards and outwards. The array was designed to provide decorrelated signals but still give some sense of movement (i.e. wind moving through trees). A photograph of the array setup is shown in Figure 2.

The child actors were equipped with lapel mics to capture dialogue and breathing. Foley sounds such as footsteps, rustling of leaves, and twigs snapping were captured using a Neumann 191 on a boom pole.

3 Production Tools

To enable efficient professional-quality spatial audio production for narrative-led VR, the export of dynamic

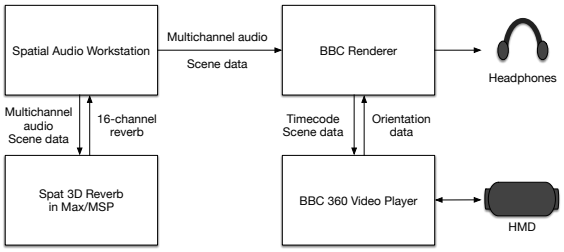

Fig. 3: Overview of tools used for audio production

spatial audio scenes (i.e. multichannel audio with time-varying scene description meta-data) from the DAW into the game authoring environment should be possible. This will allow the audio production team to create rich complex scenes that may be difficult to construct within a game engine environment alone. It is highly desirable that such tools use open-formats and lossless representations of the 3D audio content, to preserve quality, to allow for maximum flexibility in the choice of tools used at each stage of the process and to allow archiving without dependency on proprietary technologies. Real-time dynamic spatial audio rendering with head tracking should also be integrated directly into the DAW, ideally with support for monitoring ODV on a HMD. This will allow the engineer to create the scene whilst listening to and viewing it in a manner representative of the target user experience.

The production of The Turning Forest used a custom dynamic binaural rendering system and involved development of several tools to improve the VR audio production workflow, as shown in Figure 3. These tools were run as external applications to the DAW, although in future such tools may well feature natively. The Steinberg Nuendo 6.5 workstation software was used with the IOSONO Spatial Audio Workstation (SAW) plug-in used to author the object-based spatial audio scene. The SAW generated a 64-channel multichannel audio output bus on a MADI link and used the Open Sound Control (OSC) protocol to send scene meta-data. A custom rendering system developed by the BBC ran on a separate machine and received this input to generate a headphone signal by real-time binaural rendering. Additionally a 3D reverb/environment modelling tool was run externally to the DAW and a separate video monitoring application was developed which provided an interface between the audio production system and a VR HMD.

3.1 Spatial Audio Rendering Engine

The core component of VR audio systems is real-time dynamic binaural rendering. The position and orientation of the listener and sound sources within the scene are taken into account, and filtering of source sound signals is updated in real-time to create a two channel signal for headphone listening. The result should be a plausible auditory scene which corresponds to the visual virtual environment and updates rapidly to user movements and interactions.

Binaural rendering was implemented using a uniform-partitioned fast convolution engine and a set of head-related impulse responses (HRIRs) loaded at run-time from an AES69 (SOFA) file [2]. A 3-dimensional binary tree of the HRIR measurement positions is created to allow efficient nearest neighbour search. Each sound source in the scene is processed with a separate HRIR convolution. For each block of source samples to be processed, the nearest HRIR measurement angle for the source direction as seen from the current listener orientation is selected and used in the convolution. When the HRIR measurement has changed, due to source or listener movement, the convolution is performed twice using both the previous and new measurements and the two resulting signals are crossfaded over the duration of the block, thus avoiding discontinuities in the output. In addition to this process, the onset delays for the HRIRs are stored separately in the SOFA file and re-inserted after the convolution using a variable fractional delay line with a windowed-sinc interpolation filter [3]. An overview of the signal processing components is shown in Figure 4. The signal processing was implemented in a C++ library such that the same processing techniques could be used in the production rendering application and also within the game engine. This binaural rendering system was developed for use in a subjective evaluation of binaural processing techniques [4] as well as in production of spatial audio content for entertainment media.

The rendering software used different source representations, signalled in the scene meta-data, to allow multiple simultaneous rendering methods. Besides the basic binaural renderer using HRIR convolution, a binaural room scanning [5] renderer was used to render sources through measured binaural room impulse responses (BRIRs). A basic stereo panning renderer was also used to allow sources to be rendered inside the listener's head and not be affected by their orientation. This approach was used for the story narration.

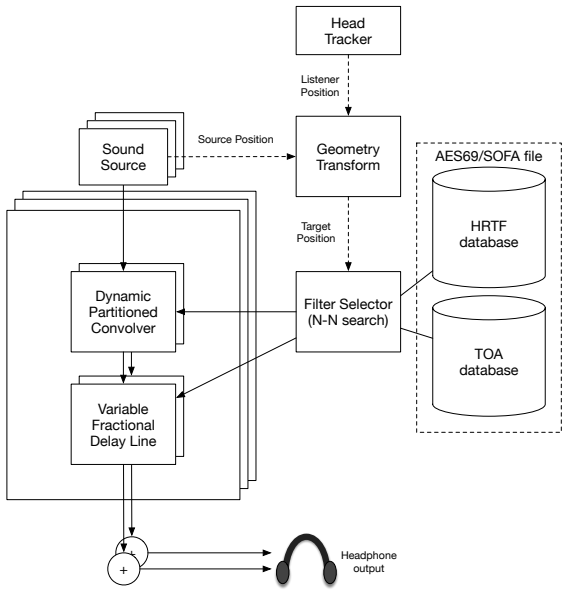

Fig. 4: Overview of the dynamic binaural rendering algorithm used in The Turning Forest VR

3.2 3D Reverb

Since no 3D reverb was available as a plug-in for a DAW, an external 3D reverb bus was created using the Spat software package[2] in Max/MSP. Spat is a parametric reverb tool with a set of perceptual controls mapping to acoustic parameters of the virtual environment [6]. It was configured to generate a 16-channel reverb bus of virtual loudspeaker sources with positions evenly distributed over a unit sphere. The virtual loudspeakers of the reverb bus were each rendered as a separate binaural source by the BBC rendering system. Spat was configured not to include the direct sound in the reverb bus, since the direct sound of each source was processed with a separate HRIR convolution in the renderer. The send level for each dynamic source object into the reverb bus could be separately controlled within the workstation and their positions were updated in synch with the direct sound rendering, using OSC messages to communicate the scene data between applications.

[2] `http://forumnet.ircam.fr/product/spat-en/`

3.3 Audio Import/Export via the Audio Definition Model

A key component of the workflow was the use of the Audio Definition Model (ADM) (Recommendation ITU-R BS.2076 [7]), to represent a complex object-based 3D audio scene within a single WAV file. The use of the ADM enabled an audio-led process in which the complete audio scene could be transferred from the DAW into the game engine and the visual scene then built to complement this.

The Audio Definition Model is a standardised meta-data model for representing advanced audio content formats. An XML representation of the ADM can be included in the *axml* chunk of a WAV file in the Broadcast Wave Format (BWF) [8], which contains the associated audio data. Object-based, channel-based and scene-based (i.e. higher-order ambisonics) 3D audio can be represented, as well as pre-rendered binaural signals and matrixed formats. For object-based content, a detailed set of parameters are available to describe for example the source position and extent in 3D space.

In next-generation broadcast systems, the ADM and its inclusion in BWF files will provide a standardised open format for archive and production exchange of advanced audio formats. It is intended that these files can be created directly from audio production software such as a DAW and provided as input to advanced audio distribution codecs that feature as part of ATSC and DVB standards for consumer delivery. Standardisation of baseline rendering algorithms that interpret the ADM parameters and generate loudspeaker driving signals is part of ongoing work in ITU-R. The capability to represent dynamic 3D audio scenes makes the ADM also useful in VR production.

Since the DAW software used in the production of The Turning Forest does not support export of an object-based 3D scene to an ADM BWF file, this export capability was implemented in the BBC rendering application. Open-source C++ libraries for reading and writing ADM meta-data and BWF files have been published by the BBC [9]. Using these tools, the BBC rendering application could be configured to generate an ADM BWF file in real-time, recording the multichannel audio input and received scene meta-data messages and storing them in the ADM format. This process required a separate configuration file, as shown in Figure 5, providing the content and object name labels, since

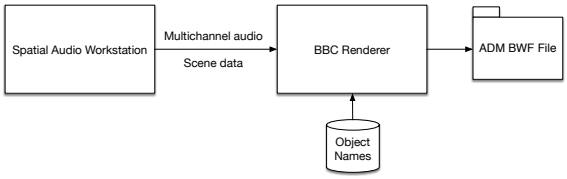

Fig. 5: ADM BWF export from the audio production system

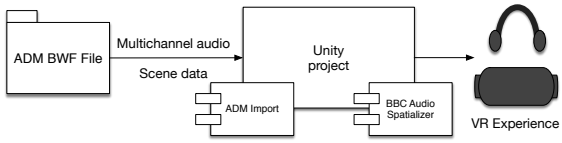

Fig. 6: ADM BWF import to the game engine

these were not accessible from the DAW via the OSC control interface. A DAW implementation of ADM export would have the advantage that offline rendering would be possible, also object or channel names could potentially be taken directly from the project data.

3.4 ODV Monitoring

Whilst the production was audio-led, it was also useful to be able to edit the audio mix and spatial scene whilst monitoring the visual scene. Typically in audio post-production for broadcast or cinema, a video track is used in the DAW for monitoring. An additional external software application was developed to present ODV, exported from the game engine, both on a regular monitor and in a VR HMD, as shown in Figure 7. This allowed for fine tuning of the audio mix to the video, it has proved very useful in more visually-led production processes.

The video playback was synchronised to the audio using MIDI timecode. Also the audio object positions were sent using OSC to the video player and overlaid on the video display to allow the operator to align the direction of sound sources with that of visual objects. The video player application also sent the HMD orientation data to the rendering application to allow the binaural rendering to update according to head orientation. When the video monitoring application was not being used, tracking data was instead provided by an optical tracking system, using markers placed on the headphones.

Fig. 7: Monitoring ODV in HMD during object-based spatial audio production

4 Game Engine Integration

The Unity[3] game engine was used to create the VR application and associated computer graphics. To enable the workflow integration between the spatial audio production system and the game engine, the software libraries used for binaural rendering and ADM handling were built into plug-ins for Unity, as indicated in Figure 6. This allowed the ADM WAV file generated by the audio production system to be imported into a Unity project, generating a set of game audio objects with animation data corresponding to the audio and object meta-data within the file. The animated spatial parameters within the engine were then used to control the dynamic binaural rendering algorithm through the Unity Audio Spatializer SDK. Using the BBC implementation ensured that the same binaural audio rendering was performed as in the production system. However it is worth noting that the decoupling of the rendering and authoring components means that an alternate spatialisation system could be used, for example one targeting efficient audio rendering for low-power devices. Using these plug-ins the team developing the computer graphics could design and animate the scene according to the spatial sound mix, the first version of which was created before the visual content was commissioned.

As previously mentioned, a third-party plug-in was used to export ODV from the game engine. This allowed the audio mix to be refined in the DAW whilst

[3] https://unity3d.com/

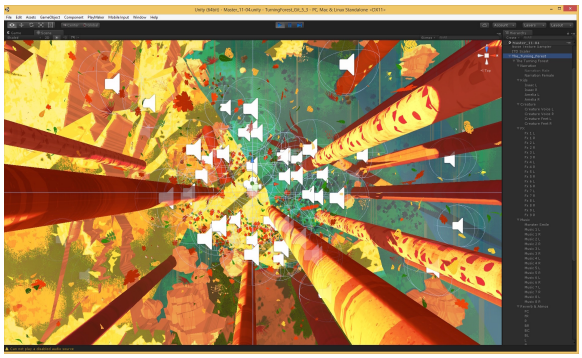

Fig. 8: Audio sources imported into Unity environment

monitoring the ODV as the visual content was developed and then reimported into the Unity project, providing a feedback loop in the production process.

5 Discussion

The described tools enabled the efficient production of complex 3D sound scenes using object-based techniques to create a rich and immersive listening experience in VR. The use of the Audio Definition Model standard enabled rapid transfer of these scenes between the DAW and the game engine. However there are still challenges to be addressed.

The ADM export relied on an external application, this made precise synchronisation difficult. In future ADM export directly from plug-ins such as the Spatial Audio Workstation or the DAW itself is desirable. Likewise it would be convenient to be able to export ADM data from the game engine and import it back into the DAW, so that certain source movements could be made to precisely track a graphical animation. This was an audio-led production since it started as an audio-only piece and the design of the workflow reflects that. The audio workstation tools themselves are well suited for mixing object-based audio to ODV but a CG project that is more visually led really requires export from the game engine to the DAW.

The approach of monitoring an ODV export from the full 3D visual environment presents some limitations. It is not possible to preview the effects of listener position translation, only orientation changes, and authoring the distance of sound sources using an ODV viewer is particularly challenging.

There are basic usability issues when using a HMD to monitor during audio production, interacting with the DAW is not possible without removing the HMD. A rapidly accessible pass-through camera or use of an augmented-reality display overlay would make audio editing whilst monitoring the video easier. Alternatively user interfaces to control the DAW and spatial audio authoring within a virtual reality environment may be beneficial. Control of 3D audio source positioning in a DAW already presents a user interface challenge [10], peripheral controllers for VR systems may be integrated into audio production tools in future to allow more effective 3D sound positioning than with current tools.

The Turning Forest is temporally linear, however much VR content will be non-linear and more interactive which presents more significant challenges for production workflows that integrate traditional audio post-production tools and game engines. As an example, in this production the 3D environment model was not parameterised within the game engine but rendered to audio signals in the DAW. This is ok for linear content, at least when sources are not very close to the listener, but for interactive content a real-time environment simulation is needed.

The Spatial Audio Workstation plug-in did not allow for authoring of more advanced audio object parameters within the ADM such as source size and diffuseness. In future authoring and rendering of such parameters could be considered. The ADM is capable of representing groupings of related objects, however in the current implementation the scene was flattened on import to the game engine. In future hierarchical relationships will be preserved so the audio scene is more easily navigated.

It is hoped that the production tools described in this paper are in future superseded by features implemented directly within common audio and game production tools. In the mean time, it is hoped that these custom enhancements to the VR production workflow can serve to demonstrate the value that such features can provide.

6 Acknowledgements

The initial audio production of the Turning Forest was supported by the EPSRC Programme Grant S3A: Future Spatial Audio for an Immersive Listener Experience at Home (EP/L000539/1). The authors would like

to thank the S3A project team for their support and the many people involved in the production of The Turning Forest VR, particularly Oscar Raby, Katy Morrison and Zillah Watson.

References

[1] Woodcock, J., Pike, C., Melchior, F., Coleman, P., Franck, A., and Hilton, A., "Presenting the S3A Object-Based Audio Drama dataset," in *AES 140th Convention*, Paris, France, 2016.

[2] "AES69-2015 AES standard for file exchange - Spatial acoustic data file format," 2015.

[3] Smith III, J. O. and Gossett, P., "A Flexible Sampling-Rate Conversion Method," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, San Diego, 1984.

[4] Pike, C., Melchior, F., and Tew, A. I., "Descriptive analysis of binaural rendering with virtual loudspeakers using a rate-all-that-apply approach," in *AES Conference on Headphone Technology*, Aalborg, Denmark, 2016.

[5] Horbach, U., Karamustafaoglu, A., Pellegrini, R. S., Mackensen, P., and Theile, G., "Design and applications of a data-based auralization system for surround sound," in *AES 106th Convention*, Munich, 1999.

[6] Jot, J.-M., "Real-time spatial processing of sounds for music, multimedia and interactive human-computer interfaces," *Multimedia Systems*, 7(1), pp. 55–69, 1999, ISSN 0942-4962, doi: 10.1007/s005300050111.

[7] ITU Radiocommunication Sector, "Recommendation ITU-R BS.2076 - Audio Definition Model," 2015.

[8] European Broadcasting Union, "EBU Tech 3285 - Specification of the Broadcast Wave Format (BWF)," 2011.

[9] BBC Research & Development, "Audio Definition Model Software," 2015, `http://www.bbc.co.uk/rd/publications/audio-definition-model-software`.

[10] Melchior, F., Pike, C., Brooks, M., and Grace, S., "On the use of a haptic feedback device for sound source control in spatial audio systems," in *AES 134th Convention*, Rome, 2013.