TOOLS FOR 6-DOF IMMERSIVE AUDIOVISUAL CONTENT CAPTURE AND PRODUCTION

F. Schweiger[1], C. Pike[1], T. Nixon[1], M. Firth[1], B. Weir[1], P. Golds[1],
M. Volino[2], M. A. Mohd Izhar[2], N. Graham-Rack[2], P. J. B. Jackson[2], A. Ang[3]

[1]BBC Research & Development, UK,
[2]University of Surrey, UK and [3]IMRSVray Ltd. UK

ABSTRACT

We present a set of tools that enable the production of immersive experiences with six degrees of freedom (6-DoF), starting from broadcast-centric audiovisual media. We use a moving 360° camera to record the light field of quasi-static backgrounds, and a camera-microphone array to capture light field video and spatial audio of foreground objects, such as actors. Audiovisual tracking produces dynamic spatial metadata for recorded sound objects, and beamforming techniques are used to provide clean audio object signals ready to be associated with other audio feeds, such as a lavalier microphone. Room acoustics are modelled and can be altered and applied to different scenes. The audio tools use the Audio Definition Model (ADM) standard, enabling object-based representation of 3D scenes. Audio workstation plug-ins enable intuitive authoring of scenes, and a headphone renderer creates spatial audio with 6-DoF listener movement adaptation. All assets are eventually composited in the Unity game engine to produce interactive 6-DoF experiences.

1 INTRODUCTION

Creating XR experiences typically requires expertise in games development and computer graphics. To achieve realistic portrayal of real-world scenes can be costly, despite recent developments in game engines. We have developed a toolset that enables production teams from a filmmaking and broadcasting background to produce immersive experiences with six degrees of freedom (6-DoF), using familiar workflows. This work was funded by *InnovateUK* as part of the *Polymersive* project under grant reference number 105168.

This paper describes a combination of production tools for background and foreground light field video and spatial audio. It discusses software to integrate these capture tools with a range of production environments, mainly using games engines such as Unity or Unreal Engine. In the short term, we aim to enable rapid 6-DoF productions in multiple genres from immersive music concerts, fashion shows, dance, theatre, some sports productions such as boxing, as well as applications for healthcare—meditation and therapy sessions, and business: rapid training and rapid crime scene reconstructions. We can also see how these tools can be used in high-budget 2D film and TV productions that require rapid pre-visualisation of complex scenes. Eventually, as the market develops, the tools described in this paper have the potential to evolve towards higher definition, higher immersion capture and production outputs that can be shown directly to audiences.

2 PROPOSED SYSTEM

In this section, we present the different components of our system and describe the production process including capture, processing, authoring and rendering.

2.1 Quasi-static Light Field Backgrounds

To capture light field backgrounds, we use a motorised arm that carries a 360° camera on a circular path with radius adjustable between 50 cm and 90 cm (see Figure 1). In most of our productions, we have used an Insta360Pro[1] although any other model would be suitable. The arm's angular velocity can be freely adjusted and is set such that during full revolution around 1800 frames are captured, which corresponds to five frames per degree. With the camera set to record at 30 frames per second, the capture process therefore takes just over a minute, allowing for a few seconds to accelerate the camera to target velocity.

Figure 1: Motorised camera rig for background capture

The recorded video then undergoes simple pre-processing steps for which we have developed dedicated prototype tools, but that can as well be carried out with existing video editing software. These steps are:

1. **Loop closure**: Trim the video so that it contains exactly one revolution, i.e., the last and first frames seamlessly transition into each other.
2. **Reprojection**: Crop and resample each 360° frame to form a 3-faced cubemap (with an upward, outward and a downward face). This mitigates the non-uniformity issues of equirectangular sampling and reduces the data size to 3/8=37.5% while maintaining the same minimum resolution. The limited field of view means, however, that the virtual camera's motion range is reduced by a factor of $\sin(90°/2) = 1/\sqrt{2} \approx 0.7$.
3. **Downsampling**: Optionally adjust the spatial resolution and the frame rate to meet the quality requirements of the final application. For example, for virtual reality experiences, a resolution of 512×1536 at 1024 frames per revolution is suitable.
4. **Encoding**: Encode the processed video in a format that is compatible across platforms. For current releases of the Unity game engine, VP8 in a WebM container is a good choice[2].

The resulting video is only several tens of megabytes in size, and can easily be stored, transmitted, and modified with standard video editing workflows, such as colour grading, rotoscoping or neural style transfer as in Gibb and Schweiger (1).

Despite this compact representation, the dataset contains a densely sampled set of rays that allows us to generate novel views from inside the capture circle (see Figure 3). While rays in the horizontal plane can simply be interpolated from captured rays, off-plane rays are not present in the data. To relate them back to available rays, depth information about the scene is required. It is very easy to position simple geometry in the Unity Editor to represent dominant scene features, such as a floor plane or walls. Photogrammetric reconstruction can help guide this process, but several manually placed planar objects typically

[1] https://www.insta360.com/de/product/insta360-pro/
[2] https://docs.unity3d.com/Manual/VideoSources-FileCompatibility.html

suffice to get a *scene proxy* that gives a good enough indication of depth. If no proxy is provided, infinite scene depth is assumed. This does not affect the horizontal component of the ray lookup, so horizontal view-dependent effects are preserved, but vertical distortions occur[3] that would be particularly noticeable when foreground objects are placed in the virtual scene. The proxy geometry is rendered with our light field shader that performs the ray lookup and interpolation.

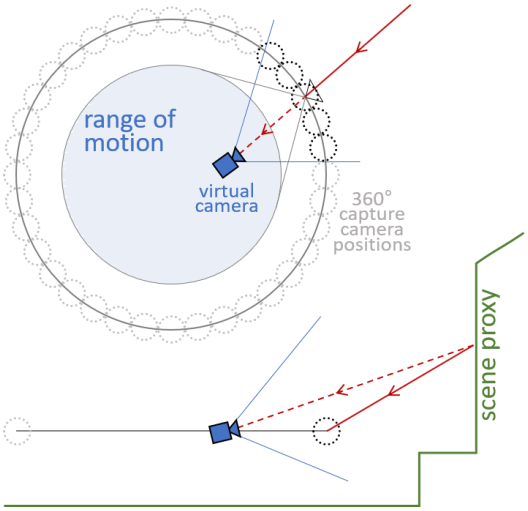
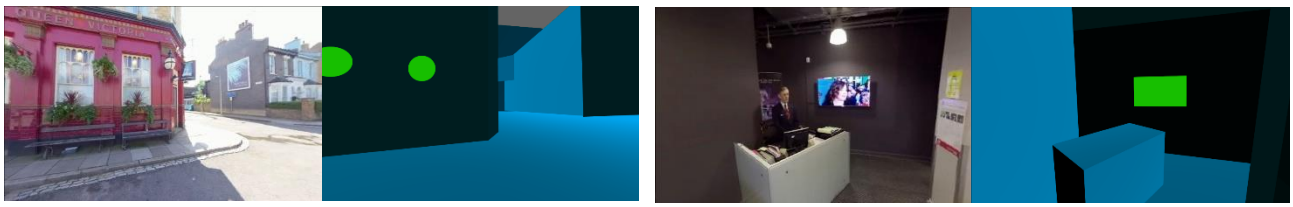


Figure 3: Top and side views of light field geometry (not to scale). Target rays (dashed red) are interpolated from source rays (solid red).

In general, the scene must be perfectly still during capture to avoid artefacts. However, indiscriminate background motion is supported if it can be replicated periodically, as long as the dynamic object remains in view over the full period length and can be sufficiently approximated by planes[4]. In Unity Editor, such objects are modelled by applying a variant of our light field shader that selects the horizontal component of source rays not based on their incident direction, but on the moment in time they were recorded. Different blending options and looping patterns (repeat/oscillate) can be chosen to achieve seamless looping and thus a higher level of realism.



(a) (b) (c) (d)

Figure 2: Views rendered from two background light fields (a,c) and the corresponding scene proxies in Unity Editor (b,d). Static scene proxy elements are tinted cyan, looped elements are in green.

2.2 Foreground Light Field Video

To capture dynamic foreground elements, such as people, a sparse video camera array was developed, shown in Figure 4. The video array consists of 11 machine vision cameras[5] set into a custom laser cut acrylic sheet providing rigid and precise positioning of the camera sensors. The array is lightweight and compact, approximately 50cm×40cm, to allow mounting to a standard studio tripod. The principal camera, located at the centre of the array, provides a reference view to an operator enabling creative decisions such as shot composition to be made in a conventional way. The 10 auxiliary cameras that

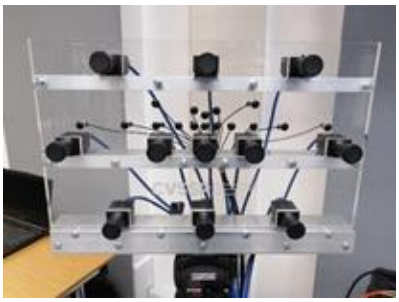


Figure 4: Camera-microphone array for foreground capture

[3] Without scene proxy, objects generally appear too tall, and do not scale according to their relative distance when the virtual camera moves towards them.
[4] This works well for planar surfaces such as TV screens or bodies of water, but also for distant objects such as plants swaying in the wind or passing birds/traffic/pedestrians in the distance.
[5] https://www.flir.com/products/grasshopper3-usb3/?model=GS3-U3-51S5C-C

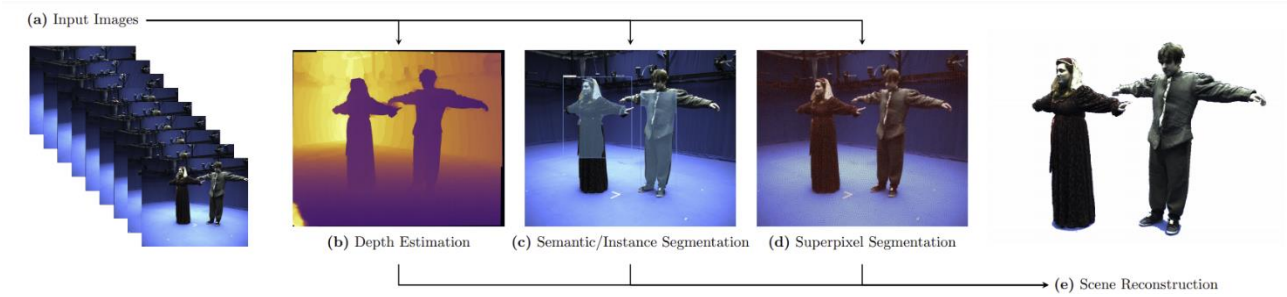


Figure 5: Visual processing pipeline stages

surround the central principal camera support 3D reconstruction of the principal camera view and do not require explicit monitoring by the operator. The configuration of the camera array has been specifically designed to support production of seated immersive content to allow users to experience six degrees of freedom within a limited volume.

Prior to each capture session, the camera array is calibrated using chart-based calibration procedure described in Hartley and Zisserman (2), which models the internal camera properties and geometry of the array. Synchronised video is captured and used as input to the 3D reconstruction pipeline, shown in Figure 5, which consists of four key stages:

1. **Depth Estimation**: Pairwise depth maps are exhaustively computed for all stereo camera pairs within the array, as per Hirschmüller (3). These pairwise depth maps are integrated to maximise completeness based on stereo matching confidence. The process produces a high-quality per-camera depth map used to model the scene geometry.
2. **Semantic Segmentation**: Mask-RCNN by He et al. (4) is used to provide a semantic classification and instance ID for each pixel in input images. The semantic and instance segmentation provide an initial coarse segmentation for each scene object which is refined further in a later processing stage.
3. **Superpixel Segmentation**: Superpixels are clusters of image pixels that share common characteristics, such as intensity, and offer an efficient way to represent an image. Utilizing the superpixel structure, e.g., superpixels which share a common edge, allows a reduction in the complexity of image-based processing by combining pixels with similar characteristics. Simple Linear Iterative Clustering (SLIC) algorithm by Achanta et al. (5) is used to process the captured images in a superpixel structure used as input for the object reconstruction stage.
4. **Object Reconstruction**: The final stage of the visual processing is to convert the per-camera depth maps, semantic and instance segmentation and super-pixel segmentation (computed in stages 1–3) into a set of 3D scene objects. This is posed as a multi-label graph cut optimisation problem in which each super-pixel is assigned a depth and object label.

2.3 Audiovisual Tracking

In object-based audio, the audio for each object is transmitted together with metadata describing essential attributes and properties of the sound source such as its position over time throughout the scene. The positional metadata can be extracted automatically by performing sound source tracking. Sound source tracking can be achieved from recordings collected via two or more microphones such as the microphone-array. However, acoustic signals are

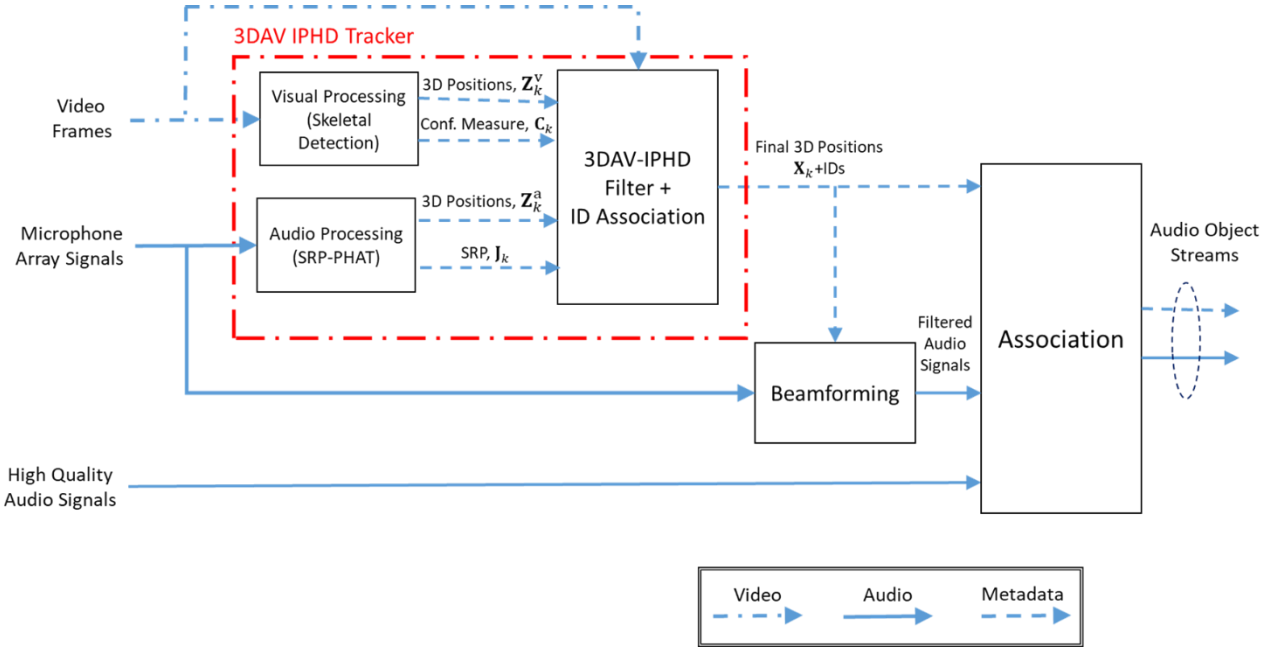


Figure 6: Block diagram of the overall system for audio objectification

susceptible to noise and by relying on audio cue only, degradation in tracking performance can be expected in adverse acoustic environments involving multiple audio objects. Therefore, we propose to fuse audio and visual cues for robust sound source tracking.

The block diagram of our overall system to automatically produce audio object streams is depicted in Figure 6. There are two main stages: sound source tracking, and associating metadata (the estimated 3D positions from the first process) with the high-quality audio feeds captured using close microphones. Beamforming is invoked to assist the association stage by providing spatially filtered signals based on the estimated positions from sound source tracking. The final output is given in the form of audio object streams following the ADM standard (see Section 2.4).

Sound source tracking is performed in the first stage by processing the audio-visual data captured from our camera-microphone array rig as shown in Figure 4. The camera-microphone array rig consists of an 11-element light-field camera-array and a 16-element microphone-array. At the beginning of this stage, both visual and audio data are processed independently in parallel. In visual processing, OpenPose by Cao et al. (6) is employed to provide 2D detections of the sound sources, e.g. mouths. Then, the 2D detections are sorted using the sorting algorithm by Malleson et al. (7) and finally, triangulation is performed using the sorted detections from all the camera views to estimate the 3D positions of the sound sources. Meanwhile, the audio data is processed using the Steered Response Power Phase Transform (SRP-PHAT) method by DiBiase et al. (8) and then filtered by Kalman filtering to yield the 3D position estimates of the sound sources.

Both the estimated positional data from audio and visual processing are fused using the probability hypothesis density (PHD) filtering framework by Mahler (9). The PHD filter is commonly used for tracking unknown and varying number of multiple targets. Our proposed tracker is based on the multiple-sensor PHD filter explicitly the iterated-corrector PHD (IPHD) filter by Mahler (10). We develop the 3D audio-visual IPHD (3DAV-IPHD) filter and the preliminary work of the 3DAV-IPHD filter was presented in Mohd Izhar et al. (11). Since

then, we have improved the filter especially by exploiting the colour information from the video frames and by correcting the distance information of the audio data. Both estimates from audio and visual processing are used in the prediction step of the filter explicitly in relocating existing particles and distributing new particles. The update step of the filter is performed two times iteratively by first updating the weight of the particles based on the 3D position estimates from visual processing and then followed by the colour-based likelihood. After the update step, the particles are resampled and clustered. The final position is estimated based on the cluster and an ID is assigned to the estimated position using colour measurements.

2.4 Spatial Audio Tools

In XR experiences, realistic interactive spatial audio is an important component to achieving a sense of presence and immersion. XR audio scenes often incorporate many elements, each of which may be in a variety of formats: either independent sound sources with defined spatial characteristics, or integrated multichannel spatial representations, using ambisonics or traditional channel-based surround sound formats.

The Audio Definition Model (ADM) (12) is a standardised metadata model that can be used to represent combinations of such audio objects. It was developed by the broadcast industry to support use of these various techniques for spatial audio in television and radio programmes, but also to enable audio personalisation through user-controlled adaptation of the object composition. The ADM defines a time-linear composition of audio objects and so provides clear benefits when creating time-linear components of an experience. The ADM parameters can also be adapted in real-time to represent scenes with interactive non-linear elements. The ADM has been used in production of spatial audio for virtual reality experiences, cf. Pike et al. (13). We have developed this approach further with the tools described in this section.

Binaural headphone renderer

A standardised rendering algorithm exists to allow reproduction of ADM-defined content on loudspeaker arrays, from stereo to 22.2 as per ITU-R (14). This system is often called the EBU ADM Renderer (EAR). The standard does not provide methods for rendering to headphones, however. We set out to develop such a standard and a real-time reference implementation, with the following requirements.

- It must have a high-quality output, suitable for both dynamic head-tracked rendering and static rendering for broadcast.
- It must support ADM-defined content, particularly distinctive features like object extent parameters, with rendering similar to that of the EAR.
- It must support real-time rendering of more than 100 channels on a single PC.

To support these requirements, we designed the BEAR (Binaural EBU ADM Renderer) around virtual loudspeaker rendering. Virtual loudspeaker gains are calculated using the EAR, and each virtual loudspeaker channel can be processed by a decorrelation filter, which when used with the width, height and depth ADM parameters can give the effect of an extended source. Head tracking is applied by modifying object positions prior to rendering.

The virtual loudspeaker signals are convolved with binaural room impulse responses (BRIRs) to generate a binaural headphone signal. The BRIRs used were recorded in the BBC listening room using the 22.2 loudspeaker layout with two additional rear floor

loudspeakers to improve localisation in that region. The BRIRs were truncated to a length of 50ms to balance timbral quality with the externalisation provided by the early reflections.

The broadband time-of-arrival is removed from the BRIRs and replaced by a per-object per-ear fractional delay, which reduces comb filter effects when multiple virtual loudspeakers are active. Delays are calculated from the virtual loudspeaker gains, by taking the average of the delays removed from the BRIRs, weighted by the gains. When decorrelation filters are used, separate per-ear and per-virtual loudspeaker delays are used (replicating the delays that were removed from the BRIR set). This is necessary to give the impression of an extended source.

The system is implemented using VISR by Franck and Fazi (14), a C++ framework for real-time audio processing, and libear[6], a C++ implementation of the EAR. A high-level API for rendering audio and metadata is provided for integration into applications.

ADM production tools

The EAR Production Suite (EPS)[7] is a set of open-source tools to support production of ADM-defined spatial audio content in a digital audio workstation (DAW). The EPS consists of a series of audio plug-ins and an extension for the REAPER[8] DAW, providing an example approach to ADM production in a typical DAW.

The EPS audio plug-ins allow for ADM audio objects to be defined within a DAW session, and for ADM audio programmes to be assembled from them. The plug-ins also provide real-time rendering facilities for monitoring purposes.

An audio object is created by adding an EPS input plug-in to a track in the DAW, which allows the ADM parameters to be configured specific to the audio object. Parameters can be time-varying, driven by existing automation features in the DAW. The input plug-ins feed a central plug-in which allows the user to build audio programmes from the audio objects and to define interactivity/personalisation options.

The monitoring plug-ins render the audio programme in real-time according to the ADM metadata. Multiple monitoring plug-ins are available to render to different loudspeaker layouts, using the libear library. There is also a binaural monitoring plug-in for headphones which uses the BEAR. This is particularly suited to XR audio production as it accepts listener orientation data using the Open Sound Control (OSC) protocol to support a range of head tracking devices, allowing the producer to monitor the end-user experience in XR applications in real-time during production.

The EPS extension for REAPER provides ADM import and export support within the DAW using standard Broadcast WAVE 64 (BW64) files (16). During import, the extension creates tracks, generates audio stems, instantiates and configures plug-ins, and constructs automation data to represent the media according to its ADM metadata. During export, the plug-in configuration and automation data provide the necessary information to produce the ADM metadata, and the audio stems are extracted via the control plug-in and written to file.

---

[6] https://github.com/ebu/libear
[7] https://ear-production-suite.ebu.io/
[8] https://www.reaper.fm/

Unity ADM plug-in

A plug-in has been developed for Unity to provide playback of ADM media within Unity-based applications. It provides a simple configuration interface, facilitating easy implementation without any code. Additionally, it provides an API for more advanced use cases, including where accurate synchronisation with other elements within a scene is necessary. The plug-in supports the use of BEAR for audio rendering, thus matching the output of the binaural monitoring plug-in from the EPS. This ensures that the end-users receive the experience as the producer intended. The audio is rendered in real-time, accounting for the orientation and position of the Unity camera to ensure audio-visual alignment. This inherently makes the plug-in compatible with XR applications built in Unity.

Parametric room acoustic modelling

In interactive XR with listener tracking, a flexible representation of room acoustics is needed, so the listener can experience the programme as if they and the audio sources were in the space from which the room impulse response was captured. The Room Acoustic Object (RAO) is an encoding of a spatial room impulse response (SRIR). We implemented a processing pipeline, summarised in Figure 7, which has the following stages:

1. Encode SRIR(s) as RAO(s)
2. Inject the RAO(s) into the metadata of a pre-existing ADM-defined audio programme
3. Render the modified ADM file binaurally, applying spatial reverb defined by the RAO(s) to individual spatial audio objects in the programme.

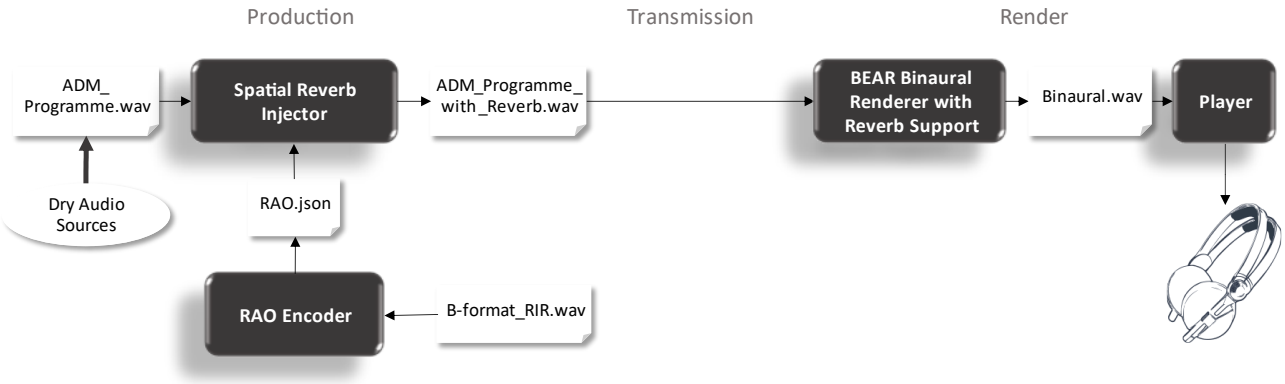


Figure 7: Room Acoustic Object processing pipeline

RAO encoder

The RAO Encoder implements the approach described in Coleman et al. (17). It takes as input a B-Format microphone recording of a SRIR and outputs a JSON file containing the RAO. The RAO comprises the Direct Path object, Early Reflection objects and a Late Reverb object. Each early reflection has direction, delay and gain relative to the direct path and a colouration described by a set of filter coefficients. The late reverb object has delay and attack times relative to the direct path and a gain and decay constant in each of several frequency bands.

Spatial reverb injector

The purpose of the Spatial Reverb Injector is to augment the metadata of an ADM-formatted audio programme file with reverb metadata. To do this, it extracts the XML-formatted metadata payload of an ADM-formatted BW64 file and modifies it. Firstly, it appends the

supplied RAO(s) in XML format. Secondly, it adds fields to each audio source to specify which RAO is to be applied to that source and at what gain. It then re-assembles the file with the augmented metadata. These augmentations of the metadata are outside of the current ADM standards. Note that the audio tracks contained in the file have not been modified.

Binaural renderer with reverb support

We extended the BEAR to render the files from the previous stage, now carrying the reverb metadata. A functional overview is shown in Figure 8. Each reflection defined in the RAO(s) is represented as a point source with appropriate delay, direction, gain and colouration. The late reverb is represented as a diffuse source with appropriate delay, attack time and band-specific gain and decay. The renderer uses the RAO metadata in the file to spawn these additional objects for each of the original audio objects and then renders them using processing paths that are added for the purpose.

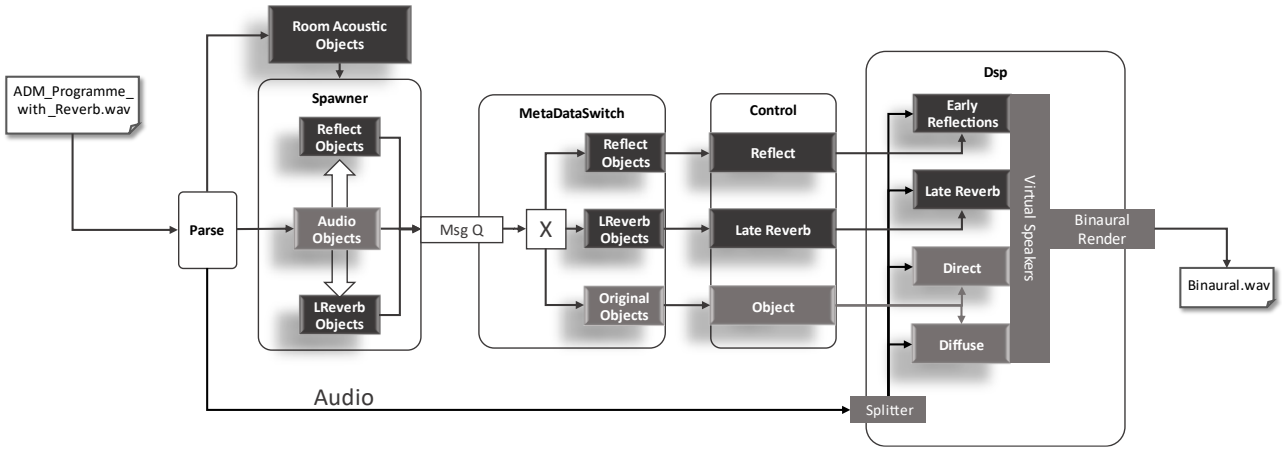


Figure 8: Extended BEAR Renderer with Spatial Reverb Support

Future audio developments

The tools presented in this section demonstrate a pipeline for high-quality standards-based production of spatial audio for 6-DoF XR experiences. Future developments will involve the evaluation of the tools with professional producers as well as the evaluation of end-user experiences created with the tools. This will drive plans for enhancements and new features. The BEAR headphone renderer and the RAO metadata extension could be considered for standardisation in future. The production tools could be extended to support real-time streaming between the DAW and the game engine during production. This would allow monitoring the complete end-user experience while editing the mix.

3 EVALUATION

In this section, we present the results of a subjective evaluation of the light field backgrounds from Section 2.1[9], as well as a quantitative analysis of the audio-visual tracker from 2.3.

3.1 Subjective User Test

To evaluate the subjective performance of our light field backgrounds, we conducted user testing on an early version of our renderer, with 23 participants selected from a pool of 18–

[9] The authors would like to acknowledge contributions by Ajethan Navaratnasingam to the subjective user test.

35-year-olds with normal eyesight and little to no prior experience of virtual reality. The implementation used in these experiments lacked a scene proxy, and the datasets were captured with a GoPro HERO4 Black[10] with limited vertical field of view. The results presented here are therefore not representative of the system as described in Section 2.1, but they should nonetheless give an indication of the benefits of light field representations for immersive environments.

The research question was whether the view-dependent effects of our light field representation (horizontal motion parallax, occlusions, reflections) would lead to a subjectively more natural and enjoyable viewing experience than an omni-directional stereo (ODS) representation of the same scene (cf. Anderson et al. (18)).

The test setup consisted of the eight scenarios listed in Table 1. These were presented in random order, each for two minutes, to participants seated on a swivel chair wearing the original HTC Vive[11] virtual reality headset. The subjects were instructed to actively look around but to remain seated throughout the experiment. To encourage them to explore the virtual environments, they were given three simple tasks per scenario asking them to locate, describe or count objects in the scene. No other instructions were given.

Scenario	Scene description (incl. preview)	Representation	Resolution (W×H×frames)
Indoor-LF-HQ	Indoor scene in a BBC Children's studio: 	Light field	360×640×2108
Indoor-LF-LQ			360×640×1054
Indoor-ODS-HQ		Omni-directional stereo	2×640×2108
Indoor-ODS-LQ			2×640×1054
Outdoor-LF-HQ	Outdoor scene on a patio overlooking a garden: 	Light field	512×512×3854
Outdoor-LF-LQ			512×512×1927
Outdoor-ODS-HQ		Omni-directional stereo	2×512×3854
Outdoor-ODS-LQ			2×512×1927

Table 1: Scenarios compared in subjective evaluation of background light fields

Question	Scale
Q1) Could you please rate the clarity of the image you just viewed?	1 (very unclear) – 5 (very clear)
Q4) How realistic was your sense of movement within the virtual environment?	1 (very unrealistic) – 5 (very realistic)
Q5) How would you rate the perception of depth in the scene?	1 (unperceivable) – 5 (very perceivable)
Q7) How much would you say you enjoyed looking around the scene?	1 (very unpleasant) – 5 (very enjoyable)

Table 2: Questions used in subjective evaluation of background light fields

After presentation of each scenario, the participants were asked to complete a questionnaire about image quality, sense of presence, enjoyability and comfort. All answers were given on a 5-point scale. For brevity, we only present the results for the questions listed in Table 2.

The results are presented in Figure 9. The graphs show the Mean Opinion Score (MOS) with 95% confidence interval for each question and every scenario. As expected, the perceived image clarity (Q1) decreases when the resolution is reduced (from HQ to LQ), for

[10] https://web.archive.org/web/20141231101757if_/http://shop.gopro.com/hero4/hero4-black/CHDHX-401.html#/tab2
[11] https://web.archive.org/web/20181121202020/https://www.vive.com/uk/product/

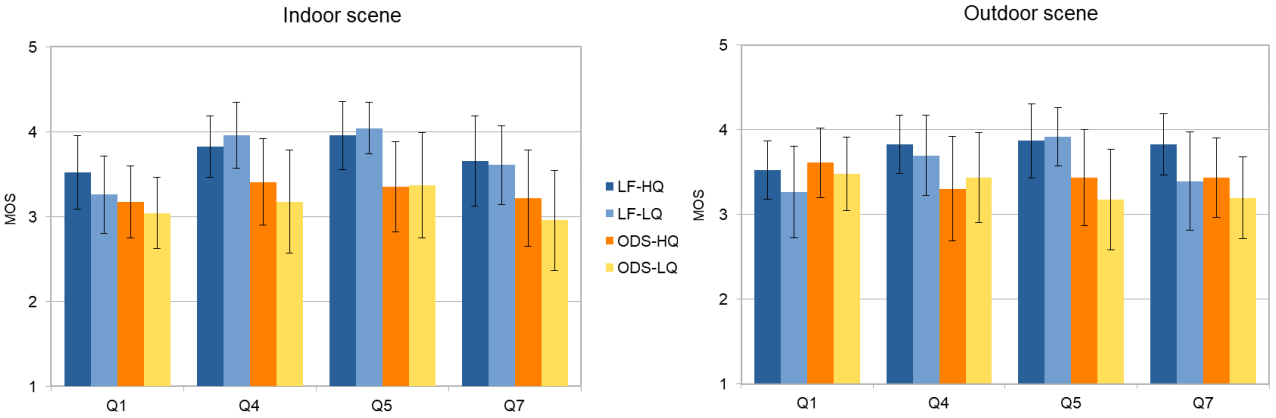


Figure 9: Mean Opinion Score results for the questions and scenarios from the tables above

both representations, and in both scenes. Furthermore, subjective realism (Q4), depth perception (Q5) and enjoyability (Q7) tend to be higher for our light fields than for their omni-directional stereo counterparts.

Comparing light fields and omni-directional stereo in their HQ versions, we performed statistical hypothesis testing[12] with the null hypothesis that LFs outperform ODS. This revealed a statistically significant performance difference in favour of our light fields, consistent across scenes, for subjective realism (Q4) and enjoyability (Q7). For the indoor scene, subjective depth perception (Q5) was also significantly higher for LFs. For the outdoor scene, however, we had to reject the null hypothesis for Q5 (p-value 9.4%). We speculate that depth perception was more pronounced in the indoor scene than in the outdoor scene because the former had more distinctive details at different depths that occluded each other. The slightly smaller vertical field of view in the outdoor scene might also have had an effect.

3.2 Audiovisual Tracking Performance

The performance of the tracker is evaluated with one of our Romeo and Juliet sequences shown in Figure 10. In this sequence, both actors were actively moving within the field of view of the centre camera, and only one actor was active (talking) at a time. The duration of the sequence was 15.8s with a total of 475 video frames. An example of the tracking result from the 3D AV-IPHD tracker is shown in Figure 10. The white bounding box represents the estimated position from visual processing, the green asterisk represents the estimated position from audio processing, and the red bounding boxes represents the estimated positions from the 3D AV-IPHD tracker. It can be observed that there is a missed detection from visual processing when Juliet turned her face

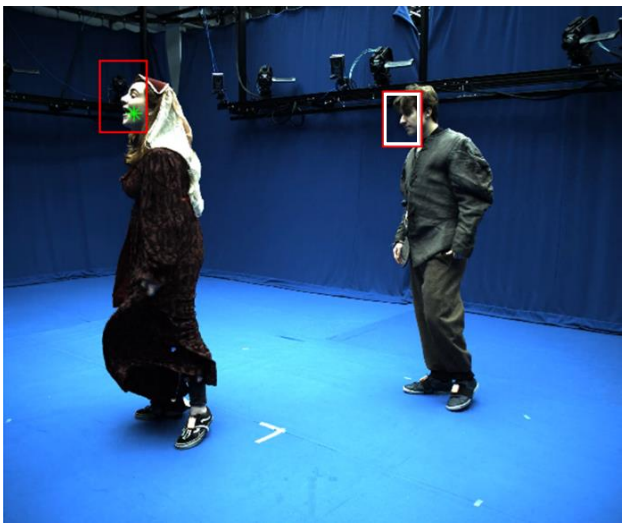


Figure 10: Tracking result for frame #110 of the Romeo & Juliet sequence

[12] Results are from a paired, one-tailed Student's t-test with a significance level of 5%.

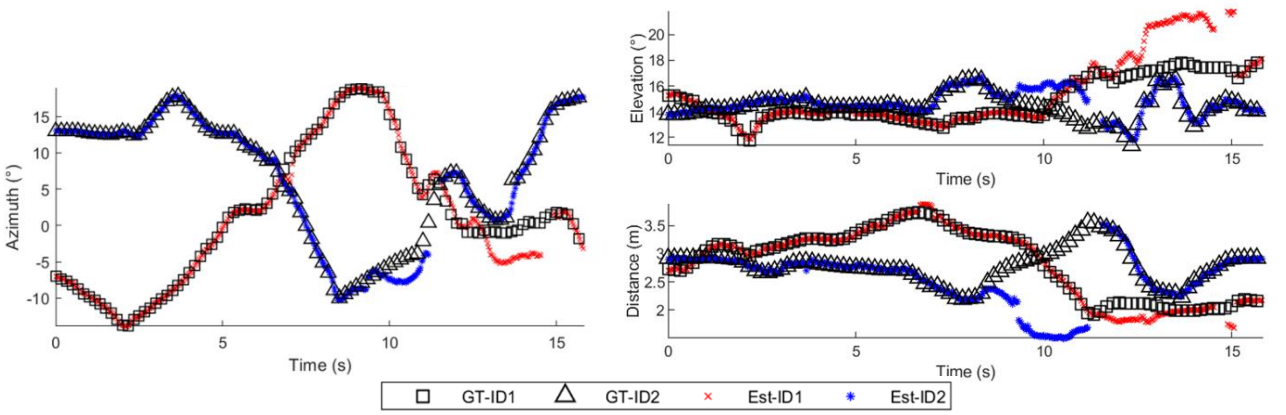


Figure 11: Ground truth (GT) and 3D positions estimated with the 3DAV-IPHD tracker (Est) for the two individuals from Figure 10

away from the camera. However, our 3D AV-IPHD tracker can detect and provide the 3D position estimate of Juliet's mouth.

The estimated 3D positions using the proposed 3D AV-IPHD tracker against the ground truth for Romeo (ID1) and Juliet (ID2) are depicted in Figure 11. A detection is considered to be valid if the estimated position is within the tolerance value of ±5º in azimuth, ±10º in elevation and ±0.7m in distance from the ground truth. A recall score of 0.92 is achieved by using the 3D AV-IPHD tracker, outperforming tracking using audio processing (recall score of 0.21), using visual processing (recall score of 0.76) and using the visual-only 3D V-IPHD tracker (recall score of 0.88). By only considering the ground truth when there is a voice activity, a perfect recall score of 1 can be achieved using the 3D AV-IPHD tracker.

4 CONCLUSION AND OUTLOOK

Spatial audio is being promoted by Apple, Sony and Amazon in their latest audio products. There will increasingly be interest in combining spatial audio production with spatial video. The tools described in this paper can enable this to happen. In the coming months, we plan to capture music venues as background light fields and record bands as light field video with spatial audio to produce immersive experiences that can by default be viewed in a VR head-set, where the user will be able to move within a 2-metre diameter and see realistic changes in the scene whilst hearing the audio in a highly realistic manner. Moreover, we have already started to use the tools to accelerate the production of high-quality scenes for mindfulness practice by combining background light fields and spatial audio to create highly realistic re-productions of virtual meditation spaces. We have captured birdsong with spatial audio—see the BBC Soundscape for Wellbeing series; and plan to combine this with appropriate backgrounds to bring the viewer into an immersive natural experience.

6 REFERENCES

1. Gibb A, Schweiger F. Artistic Style Transfer to Light Fields. In: Conference on Visual Media Production. London; 2018.
2. Hartley R, Zisserman A. Multiple view geometry in computer vision. 2nd ed. New York, NY, USA: Cambridge University Press; 2003.
3. Hirschmüller H. Stereo Processing by Semiglobal Matching and Mutual Information. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2008 Feb;30(2):328–41.
4. He K, Gkioxari G, Dollar P, Girshick R. Mask R-CNN. In: IEEE International Conference on Computer Vision. 2017. p. 2961–9.
5. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Süsstrunk S. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2012 Nov;34(11):2274–82.
6. Cao Z, Hidalgo G, Simon T, Wei S-E, Sheikh Y. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2021 Jan;43(1):172–86.
7. Malleson C, Collomosse J, Hilton A. Real-Time Multi-person Motion Capture from Multi-view Video and IMUs. Int J Comput Vis. 2020 Jun 1;128(6):1594–611.
8. DiBiase JH, Silverman HF, Brandstein MS. Robust Localization in Reverberant Rooms. In: Brandstein M, Ward D, editors. Microphone Arrays: Signal Processing Techniques and Applications. Berlin, Heidelberg: Springer; 2001. p. 157–80. (Digital Signal Processing).
9. Mahler RPS. Multitarget Bayes filtering via first-order multitarget moments. IEEE Transactions on Aerospace and Electronic Systems. 2003 Oct;39(4):1152–78.
10. Mahler R. Approximate multisensor CPHD and PHD filters. In: 13th International Conference on Information Fusion. 2010. p. 1–8.
11. Mohd Izhar MA, Volino M, Marston D, Hilton A, Jackson PJB. Tracking sound sources for object-based spatial audio in 3D audio-visual production. In: e-Forum Acusticum 2020. 2020. p. 2051–8.
12. Recommendation ITU-R BS.2076-2 – Audio Definition Model. International Telecommunication Union; 2019.
13. Pike C, Taylor R, Parnell T, Melchior F. Object-Based 3D Audio Production for Virtual Reality Using the Audio Definition Model. In: AES International Conference on Audio for Augmented and Virtual Reality. Audio Engineering Society; 2016.
14. Recommendation ITU-R BS.2127 – Audio Definition Model renderer for advanced sound systems. International Telecommunication Union; 2019.
15. Franck A, Fazi FM. VISR—A Versatile Open Software Framework for Audio Signal Processing. In: AES International Conference on Spatial Reproduction. Audio Engineering Society; 2018.
16. Recommendation ITU-R BS.2088 – Long-form file format for the international exchange of audio programme materials with metadata. International Telecommunication Union; 2019.
17. Coleman P, Franck A, Menzies D, Jackson PJB. Object-Based Reverberation Encoding from First-Order Ambisonic RIRs. In: AES Convention 142. Audio Engineering Society; 2017.
18. Anderson R, Gallup D, Barron JT, Kontkanen J, Snavely N, Hernández C, et al. Jump: virtual reality video. ACM Transactions on Graphics. 2016 Nov 11;35(6):1–13.