VenueExplorer, Object-Based Interactive Audio for Live Events

Matthew Paradis
BBC Research and Development
Centre House, 56 Wood Lane
London, W12 7SB
+44 30304 09889
matthew.paradis@bbc.co.uk

Rebecca Gregory-Clarke
BBC Research and Development
Centre House, 56 Wood Lane
London, W12 7SB
+44 30304 09643
rebecca.gregory-clarke@bbc.co.uk

Frank Melchior
BBC Research and Development
5th Floor, Dock House
MediaCityUK, Salford, M50 2LH
+44 30304 09546
frank.melchior@bbc.co.uk

ABSTRACT
VenueExplorer is a new approach to broadcasting live events which gives more control to the audience than traditional viewing methods. Users can navigate around an ultra-high resolution video, zooming into the areas of the event which interest them and accessing extra content. VenueExplorer aims to be platform independent and runs in the browser. In this paper we describe the development of object-based audio rendering to create a more engaging and personalised experience than that of video alone. We use the Web Audio API (WAAPI) to process audio based on the users viewport. We also describe a library that has been developed as part of this project for the handling of location based audio objects.

Categories and Subject Descriptors
H.5.1: [Multimedia Information Systems] Audio input/output

General Terms
Design, Experimentation, Human Factors.

Keywords
Object Based Audio, Web Audio API, Broadcasting.

1. VENUEEXPLORER
VenueExplorer is an application that allows the user to control the view that they are receiving of a live event. The scene can be panned and zoomed whilst maintaining the resolution of the video.

VenueExplorer was recently demonstrated as part of the BBC Research & Development Commonwealth Games showcase in Glasgow 2014 [2]. Athletics was identified as an ideal use-case for VenueExplorer due to its scale, atmosphere, and varied areas of interest. Typically there may be three or four sports occurring simultaneously, so there is a lot for a user to explore and choose from.

Developed after the completion of the EU Fascinate project [9] VenueExplorer was initially a video only experience but has since been developed to make use of the WAAPI to deliver multiple channels of audio to the browser which can then be mapped to specific events on screen.

The application runs in the web browser and uses MPEG DASH [11] to stream media to the client.

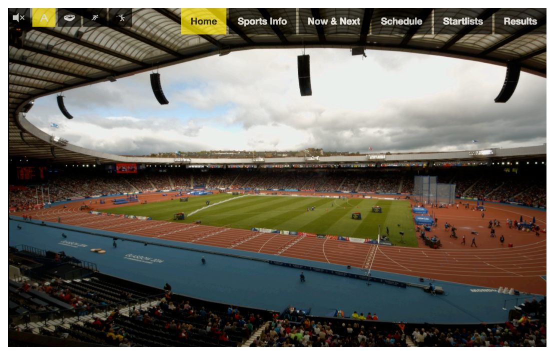

Figure 1: Screenshot of the VenueExplorer default view.

The user is initially presented with a wide-angle video view of the whole venue, from which they can zoom in and pan around to explore the event. The video is captured from a single locked off 4K camera which is encoded and delivered to the client. When the user zooms into the scene the video switches to a second stream containing one of nine overlapping video tiles (as shown in Figure 2) depending on which tile contains the area of the video defined by the viewport. The aim is to make this switch between the original down-sampled stream and the zoom area as seamless as possible.

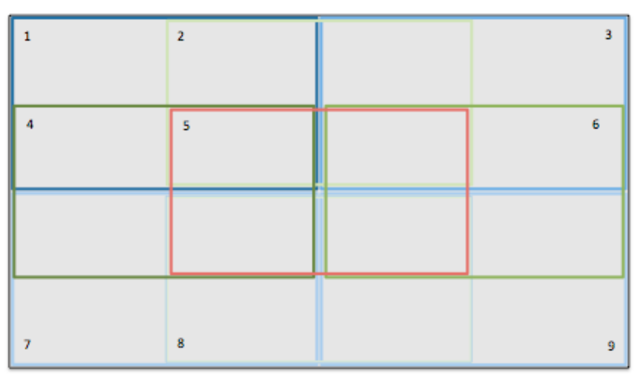

Figure 2: Video tiling.

WAC'15, January 26-27, 2015, Paris, France.

In addition to the video view, additional information about the event can be presented to the user in the form of graphical overlays. This information could be static or else dynamically updated from a suitable real-time data source. In the case of athletics this might include a schedule of the day's events, location markers signifying the location of each of the sports, lists of competitors and results.

To complement the video and complete the user experience audio has been added to VenueExplorer. The approach taken was to remix commentary and ambiance feeds from the host broadcaster to deliver relevant audio to the user based on their current viewport. In addition we positioned a Double M/S [10] microphone array at the camera position to capture the overall stadium ambiance. In this experiment the Double M/S signal was decoded to stereo before being sent to the client using the Schoeps Double MS Tool BF VST plugin [12]. In future iterations of this project we propose to decode the Double M/S signal in the browser.

This approach allows the user to explore the stadium freely both visually and aurally. In an auto-mix mode the audio will change based on the sport or location that the user is currently viewing and a manual switched mode allows the user to specifically select from a range of feeds. This mode allows the user to explore the venue while following a certain event by listening to its broadcast feed.

The audio feeds are not tied to any one of the 9 video tiles provided to the user there may be multiple "hotspots" within a single tile or hotspots may cross the borders between tiles. The solution to this problem was to broadcast the audio as defined "objects" to allow them to be associated with the viewport of the user rather than the individual video streams resulting in audio feeds that are linked to the sports or areas of interest rather than the videos themselves.

2. OBJECT BASED AUDIO

The object-based approach to audio [4] allows multiple audio "objects" to be delivered to the client device, in this case the browser on a tablet or touch screen computer. The client is then responsible for rendering the objects according to the metadata that has been supplied. An audio object consists of 2 elements, the audio data itself and metadata describing the properties of the audio and how it should be rendered. The metadata can be embedded in the media itself [7][8], supplied as a concurrent data stream, or in the case of VenueExplorer, downloaded as part of the web application which renders the media to the device.

Object-based broadcasting has an advantage over traditional broadcast as the assets are kept separate, rather than being rendered into a final mix which then has to be adapted to the target device. In addition there can be an element of user control over the way that the media is played back. For example the emphasis could be put on commentary rather than background ambiance for a sporting event, or background music levels could be adjusted as demonstrated in the BBC 5 Live Football experiment [5]. This experiment streamed 3 objects to the client, commentary and crowd noise from each end of the stadium allowing the user to select the level of crowd vs. commentary and also position themselves at either end of the stadium.

The approach taken for VenueExplorer's audio processing was to define the metadata relating to the location, size and source mixing of events (discreet sports happening within the stadium) as an included json file within the application. This provides a central point at which the rendering rules could be changed and delivered to the clients.

Audio was delivered as two 5.1 streams to the browser. 5.1 was chosen as it was a format that was compatible with the entire workflow from the encoders to the HTML5 audio element. Although 5.1 streams were used the audio information does not relate to a 5.1 mix in any way. We are using it as a transport mechanism to deliver 12 channels of audio to the client , for example. (*Stream 1***:** Channel 1 Stadium Atmosphere (L); Channel 2 Stadium Atmosphere (R); Channel 3 Track Commentary (mono); Channel 4 Silent (LFE); Channel 5 Throws Atmosphere (L); Channel 6 Throws Atmosphere (R)**,** *Stream 2***:** Channel 1 Track Atmosphere (L); Channel 2 Track Atmosphere (R); Channel 3 Throws Commentary (mono); Channel 4 Silent (LFE); Channel 5 Jumps Commentary (mono); Channel 6 Jumps Atmosphere (mono)). There are issues in using the 5.1 format in this way. Firstly we are transmitting more data than is necessary on the network as we discard the LFE channel, this also causes the encoding of an unused channel. Importantly there is the possibility of accidently using audio on the LFE channel and it being interpreted by the encoder or rendering device in the wrong way, for example low pass filtering could be added. It is hoped that future formats such as MPEG.H [7] will provide a more efficient solution to this problem.

Once encoded and delivered to the browser the 5.1 streams can be separated into discreet mono feeds using a chain of the following Web Audio API nodes: AudioElement - MediaElementSource - ChannelSplitter(6).

For each 5.1 source this now provides the required 6 discreet audio signals.

The audio object metadata is described using the following format this allows the channels to be grouped into objects, mixed, panned and have their location and operation mode set. Figure 3 shows a simple example of the metadata used for VenueExplorer along with brief descriptions of each parameter,

JSON	**Parameters**
`{`	
`"dynamicObjects":`	Object type
`[`	
`{"id": 0,`	Object unique id
`"x": 675,`	Centre X (px)
`"y": 387,`	Centre Y (px)
`"width": 578,`	Width (px)
`"height": 82,`	Height (px)
`"scalemax": 0.35,`	Maximum scale at which it can sound = 0.35 (VE specific value, where 0 = fully zoomed in and 1 = fully zoomed out.
`"scalemin":0,`	Minimum scale at which it can sound = 0.
`"fade": 1,`	Fade duration (secs)

```
        "gain": 0.3,                Object global gain
        "tag": "throws atmos",      Text identifier
        "chaninfo":
            [
                {"id": 0,           Id unique to channels
                "stream": 0,        Source stream id
                "channel": 1,       Stream channel id
                "gain": 1,          Source gain
                "pan": 0},          Source pan
                {"id": 1,           Id unique to channels
                "stream": 1,        Source stream id
                "channel": 0,       Stream channel id
                "gain": 1,          Source gain
                "pan": 1}           Source pan
            ]
        }
    ],
    "platform": "desktop"           Platform identifier
}
```

Figure 3: Object Metadata.

In this instance we are combining 2 source signals into a single audio object. We index the sources by numbering the input 5.1 streams (first stream = 0, second stream = 1) and then selecting channels 0-5 from those streams.

3. BBCAUDIOOBJECT.JS

In order to read and apply the metadata described in Figure 3 in a flexible way bbcaudioobject.js has been developed. This abstracts the workings of the WAAPI allowing the developer to create audio objects which are set up using the metadata as their creation arguments. An update or render function can then be called with the users current screen position and scale factor which will allow the objects to be evaluated.

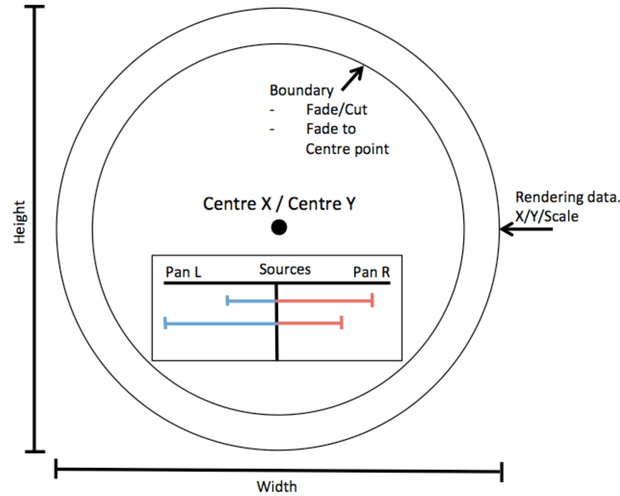

Figure 4: Illustration of a circular audio object.

The creation and triggering of an object is straightforward using bbcaudioobjects.js. This paper does not cover the process of reading the metadata or calculating the viewport location as this is not relevant to the WAAPI.

```
//create an audio context
audioContext = new AudioContext();

//create a new audio object passing the audio context
audioObject = new BBCAudioobject(audioContext);

//connect a source (a channelSplitter output) defining
//gain and pan
audioObject.sourceConnect(<audioSrc>,0,1);

//define the location and behaviour of the object
audioObject.objectDefine({
    x:675,
    y:387,
    width: 578,
    height: 82,
    scaleMax: 0.35,
    scaleMin: 0,
    fadeTime: 2,
    name: "throws atmos",
    gain: 0.3
});

//Now periodically call the render function
audioObject.inputPosition(<X>, <Y>, <scale>);
```

On receiving an x/y/scale combination which is "in range" for the object, the object will either fade in all of its panned sources over the defined fade time (2 seconds in this case) whilst maintaining the gain levels set for individual sources. If the object is already sounding it will do nothing, and if the location parameters fall out of range it will fade out over the defined fade time.

Using this framework we can define a number of different behaviors currently for square, rectangular or circular objects.

- Fade on boundary
- Switch on boundary
- Fade to center

By assigning "bbcaudioobjects" to arrays and iterating over the metadata we can produce a flexible framework for handling multiple audio "hotspots" which can be configured through the metadata configuration alone. This makes changes and balancing of the object very straightforward. Objects can be added, removed or repositioned without changing the application code.

4. RESULTS OF INITIAL TRIAL

In carrying out this project we have made a number of observations on our approach, and also documented some of the current limitations that we encountered using the WAAPI.

Camera Angle – not related to the WAAPI, but fundamental to the effectiveness of this approach. Figure 1 shows the view point of the 4K camera used for the Commonwealth Games trials [2].

From an audio, and to an extent, video perspective the distribution of events is very uneven as shown in the annotated Figure 5.

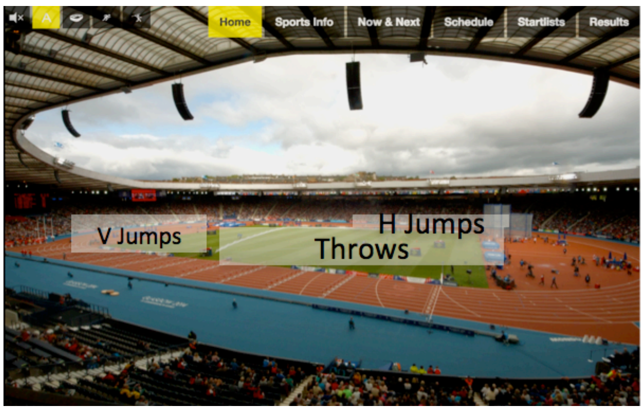

Figure 5: Screenshot of the Venue Explorer with graphical overlays.

It is very difficult to achieve good separation between some of the events, in Figure 5 the horizontal jumps event and throws are very close together and as the track covers the majority of the field of play it also encroaches on several other events.

In order to solve this the camera should be placed at an angle which achieves the maximum separation between events. This also provides more appropriate separation for the graphical overlays and provides more of a "birds eye" view of the event.

Compatibility – we discovered discrepancies in the implementations between WAAPI nodes in the desktop and tablet implementations for browsers. One example of this was the CreateMediaElementSource method. At the time of writing this was not fully implemented in Android although there is a placeholder which prevents errors from being thrown. This means that VenueExplorer, with audio, is only available on desktop browsers until the functionality is implemented for Android.

Authoring – in order to determine the location and dimensions of audio objects the pixel values are required as inputs to the library. This is quite time consuming and there is a high margin for error when doing this manually. An item of future work is the development of an authoring tool to allow objects to be defined graphically and then exported as valid metadata.

5. INITIAL USER FEEDBACK

Qualitative user feedback about VenueExplorer was gathered from tests conducted with members of the public at BBC R&D's user testing laboratory. Overall the response to the proposition was overwhelmingly positive, and the audio capabilities were particularly well received. Generally the users all preferred the experience with the audio features included rather than as a video-only application.

Users liked the immersive aspect of hearing the crowd noise, as this was considered very important to the feeling of "being there".

> *"Straight away it's more like I'm at the event, it's really good"* (male, 27)

The commentary option was also felt to be very important as a way of understanding what's going on in the scene.

The 'automix' capability, whereby the commentary is mixed in automatically as the user zooms into an event, was seen to have a clear 'wow factor'. This feature works well for users who are exploring the scene with general interest, rather than a specific goal in mind.

> *"being able to go over to the discus and hear it, that feature's good" (female, 47)*

> *"being able to go from event to event almost immediately and listen in on the events – it feels different to TV commentary, it feels like you're more there experiencing it…more submerged in the event itself" (male, 22)*

The manual method of switching between audio options was also very much appreciated, and works better for those who have a more specific goal in mind, for example, they may have decided they are interested in the pole vault, and find it simpler and faster to select the commentary by clicking a button rather than zooming in and waiting for it to fade in.

> *"Perhaps it's automatically mixed when it comes on but then you can choose" (female, 54)*

When asked if they would like more control over the relative levels of crowd noise and commentary, several people expressed an interest, however their main reservation was that this option should not overcomplicate the interface, as the simplicity of the UI was considered one of the best factors.

6. FUTURE DEVELOPMENT

The VenueExplorer trial with audio has led to a number of discussions over the applications of this technology in the future. One key area of interest is music festivals. Like athletics events a music festival is made up of a number of discreet events, indoor and outdoor stages, ad hoc performances and the festival atmosphere. We are keen to investigate a similar approach to allow users to experience the atmosphere of navigating around a festival, while giving them a new way of discovering content.

In addition we are investigating the use of VenueExplorer to provide a new listener perspective. By multi-micing orchestras and providing surround capture we can process this audio in the browser in real time allowing the listener to focusing on specific instruments, or even move around the stage itself.

7. ACKNOWLEDGMENTS

The authors would like to thank David Evans & Stephen Perrot for their support in media encoding, distribution and video rendering, Paul Debenham for project management and support of the Commonwealth Games showcase. and Stuart Memo for reviewing bbcaudioobject.js .

8. REFERENCE

[1] http://www.bbc.co.uk/rd/projects/venue-explorer

[2] http://www.bbc.co.uk/rd/blog/2014/07/how-the-commonwealth-games-is-helping-define-the-future-of-broadcasting

[3] http://webaudio.github.io/web-audio-api/

[4] M. Armstrong, M. Brooks, A. Churnside, M. Evans, F. Melchior, M. Shotton *Object-based broadcasting – curation, responsiveness and user experience*, Proceedings of IBC2014, Amsterdam, September 2014.

[5] M. Mann, A. Churnside, A. Bonney, F. Melchior,, *Object-based audio applied to football*. Proceedings of the ACM International workshop on Immersive media experience, 2013.

[6] B. Shirley, R. Oldfield, F. Melchior, J.-M. Batke,, *Platform Independent Audio*. in Media Production, Delivery and Interaction for Platform Independent Systems, Wiley 2014.

[7] J. Herre, J. Hilpert, A. Kuntz, J. Plogsties, *MPEG-H Audio - The Upcoming Standard for Universal Spatial / 3D Audio Coding*. Proceedings of ICSA 2014.

[8] EBU Tech 3364, Audio Definition Model. Geneva, January 2014 https://tech.ebu.ch/docs/tech/tech3364.pdf

[9] http://www.fascinate-project.eu

[10] http://www.schoeps.de/documents/SCHOEPS_ Double_MS_paper _E_2010.pdf

[11] http://dashif.org/mpeg-dash/

[12] www.schoeps.de/en/products/dms_plugin_bf