Audio Engineering Society

Convention e-Brief 477

Presented at the 145th Convention
2018 October 17 – 20, New York, NY, USA

Audio Application Programming Interface for Mixed Reality

Rémi Audfray[1], Jean-Marc Jot[1], and Sam Dicker[1]

[1]*Magic Leap, Inc.*

Correspondence should be addressed to Rémi Audfray (`raudfray@magicleap.com`)

ABSTRACT

In mixed reality (MR) applications, digital audio objects are rendered via an acoustically transparent playback system to blend with the physical surroundings of the listener. This requires a binaural simulation process that perceptually matches the reverberation properties of the local environment, so that virtual sounds are not distinguishable from real sounds emitted around the listener. In this paper, we propose an acoustic scene programming model that allows pre-authoring the behaviours and trajectories of a set of sound sources in a MR audio experience, while deferring to rendering time the specification of the reverberation properties of the enclosing room.

1 Introduction

Computer-generated audio rendering for MR can leverage two decades of signal processing technology developments in game and virtual reality audio rendering systems and application programming interfaces, building upon and extending from prior developments in the fields of computer music and architectural acoustics (including binaural techniques, artificial reverberation, physical room acoustic modeling and auralization). The aim of this engineering brief is to extend well established standards to create a rendering description model that could serve as an interface between MR applications and rendering systems, or audio Application Programming Interface (API) [1]. The rendering system would then be driven from a higher level control system, based for instance on a combination of physics and psychoacoustics principles.

2 Audio API

A very common way to describe an acoustic system for simulation and interactive applications is to decompose it into a set of sound sources, a receiver (or listener), and an acoustic environment. This audio API proposal follows that approach, leveraging and expanding on [2], [3] and [4]. The parameters controlled through the audio API can be grouped into 4 categories: global properties, source properties, listener properties, and room properties.

2.1 Global Properties

First, a set of control frequencies are defined. These control points will then be used throughout the rest of the audio API. An advantage of sharing these control frequencies across the entire API is when equalizers are cascaded, their combined effects can be described in the same terms, by adding the dB gains at the control frequencies. This property is valid for instance when using Proportional Parametric Equalizers [5], and allows efficient implementation by lumping several effects into a single digital filter unit. Table 1 shows a list of these global properties in the audio API, which include:

- *control_lf*, the reference frequency for all ‘lf’ controls.
- *control_mf*, the reference frequency for all ‘mf’ controls.
- *control_hf*, the reference frequency for all ‘hf’ controls.

2.2 Sound Source

A sound source can be modeled by describing its location in space, orientation, and radiation pattern, including level, directivity, and magnitude response. Table 2 shows a list of the source properties in the Audio API. These properties can be sorted in the following categories: geometry properties, distance properties, radiation properties, and per-source offsets.

Geometry Properties These properties describe the placement and orientation of the sound source in space:

- *position_r, position_u, and position_f*, the coordinates of the source relative to the listener, in cartesian form as seen as ’up’, ’right’, and ’front’ respectively from the listener’s point of view.
- *direction_r, direction_u, and direction_f*, the direction towards which the source is aiming in cartesian form as seen as ’up’, ’right’, and ’front’ respectively from the source’s point of view, ’front’ being the direction from the source to the listener.

Distance Properties There are also a set of properties that define the calculation of a distance-based gain offset (attenuation) that is automatically applied to the source’s direct sound component, as a function of source-listener distance:

- *min_distance*, the minimum distance for the sound source. When a source is closer than min_distance, the gain offset is the same as at Min Distance. By default, in our audio API proposal, min_distance is set to 1 meter.
- *max_distance*, the maximum Distance for the sound source. When a source is farther away than Max Distance, the gain offset is the same as at Max Distance.
- *rolloff_factor*, the rolloff factor for the sound source. This determines how steeply the gain rolls off from min_distance to max_distance. The roll-off follows the “Inverse Distance Clamped” model:
  - If distance < min_distance, the gain is unaffected.
  - If min_distance $\leq$ distance $\leq$ max_distance, the attenuation is specified by the formula: $Gain = MinD/(MinD + Rolloff * (distance - MinD))$.
  - If rolloff_factor is set to 0.0, the gain is unaffected.

This distance attenuation behavior is equivalent to the distance model defined in [4], and the ’inverse distance clamped model’ defined in [2].

Radiation Properties These parameters specify the directivity of the sound source, so that the direct sound component will be automatically attenuated and filtered when the sound source is not pointing to the listener. This effect will be automatically updated according to the sound source orientation and the listener position. Additionally, making the source more directive will produce a natural attenuation of the reverberation for this source.

- *inner_angle*, the inner cone angle. Defined symmetrically around the forward vector of the source.
- *outer_angle*, the outer cone angle. Defined symmetrically around the forward vector of the source.
- *outer_gain*, the gain outside the outer cone.
- *outer_gain_lf*, the low frequency gain offset outside the outer cone.
- *outer_gain_mf*, the mid frequency gain offset outside the outer cone.
- *outer_gain_hf*, the high frequency gain offset outside the outer cone.

Property Name	Type	Units	Min	Max	Default
control_lf	float	Hz	125.0	500.0	200.0
control_mf	float	Hz	control_lf * 4	control_hf / 4	1000.0
control_hf	float	Hz	2000.0	8000.0	5000.0

Table 1: Global Porperties

Per-Source Offsets The set of parameters listed in this section enable fine tuning the direct-path and the room reflections and reverberation response at min_distance for each sound source.

- *gain*, the overall gain for the source. This gain affects all paths: Direct, Reflections, and Reverb.
- *direct_send_gain*, the direct-path gain offset.
- *direct_send_gain_lf*, the direct-path relative gain offset for low frequencies.
- *direct_send_gain_mf*, the direct-path relative gain offset for mid frequencies.
- *direct_send_gain_hf*, the direct-path relative gain offset for high frequencies.
- *room_send_gain*, the room-path gain offset (affects reflections and Reverb).
- *room_send_gain_lf*, the room-path relative gain offset for low frequencies.
- *room_send_gain_mf*, the room-path relative gain offset for mid frequencies.
- *room_send_gain_hf*, the room-path relative gain offset for high frequencies.

2.3 Listener

Similarly, the listener can be modeled by describing its location in space, orientation, and its direction and distance dependent sensitivity, or head-related transfer functions. However, since we defined the sound source's position and orientation relative to the listener, all that is needed to characterize the listener is the set of HRTFs, which are not currently exposed in this API proposal.

2.4 Acoustic Environment

In a lot of applications, the acoustical environment can be described as a room. The properties of a room are commonly represented with a room impulse response (RIR). The proposed model takes the conventional approach of decomposing the RIR into a parametric representation, separating direct energy, early reflections, and late reverberation, as illustrated on Figure 1.

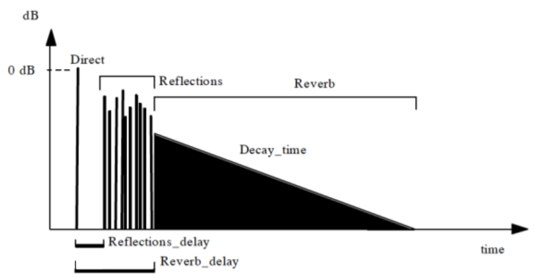

Fig. 1: Room Impulse Response

The direct energy is captured in the source and listener properties, and is independent of the acoustic environment. Early reflections can be described in terms of level and timing. The reverberation tail can be described using the room fingerprint concept, which captures reverberation time and level as a function of frequency [6], as well as a time delay.

- *room_gain*, the global room gain (which affects both Reverb and Reflections as a global offset).
- *reflections_delay*, the delay time of the first reflection relative to the direct-path sound arrival (see room response graph on Figure 1).
- *reflections_gain*, the early reflections gain adjustment. A reflections_gain of 1.0 means that their combined energy is the same as the input sound energy when the source is within min_distance.
- *reverb_delay*, the delay time of the reverb relative to the direct-path sound arrival.

Property Name	Type	Units	Min	Max
position_r	float	meters	-<MaxFloat>	<MaxFloat>
position_u	float	meters	-<MaxFloat>	<MaxFloat>
position_f	float	meters	-<MaxFloat>	<MaxFloat>
direction_r	float	meters	-<MaxFloat>	<MaxFloat>
direction_u	float	meters	-<MaxFloat>	<MaxFloat>
direction_f	float	meters	-<MaxFloat>	<MaxFloat>
min_distance	float	meters	> 0.0	<MaxFloat>
max_distance	float	meters	> min_distance	<MaxFloat>
rolloff_factor	float	none	0.0	<MaxFloat>
inner_angle	float	degrees	0.0	outer_angle
outer_angle	float	degrees	inner_angle	360.0
outer_gain	float	linear	0.0	1.0
outer_gain_lf	float	linear	0.0	1.0
outer_gain_mf	float	linear	0.0	1.0
outer_gain_hf	float	linear	0.0	1.0
gain	float	linear	0.0	8.0
direct_send_gain	float	linear	0.0	1.0
direct_send_gain_lf	float	linear	0.0	1.0
direct_send_gain_mf	float	linear	0.0	1.0
direct_send_gain_hf	float	linear	0.0	1.0
room_send_gain	float	linear	0.0	1.0
room_send_gain_lf	float	linear	0.0	1.0
room_send_gain_mf	float	linear	0.0	1.0
room_send_gain_hf	float	linear	0.0	1.0

Table 2: Source Porperties

- *reverb_gain*, the total energy of the reverb when the listener and the source are collocated.
- *reverb_decay_time*, the reverberation decay time.
- *reverb_decay_lf_ratio*, the relative decay time multiplying factor for low frequencies.
- *reverb_decay_hf_ratio*, the relative decay time multiplying factor for high frequencies.

More on reverb_gain Since the definition of reverb_gain is departing from existing literature, it is worth expending a bit on the concept. Figure 2 illustrates the general calculation of the Reverb Energy (RE), which measures, in terms of signal power, the amplification of an input signal by the reverberation processing system. RE is equal to the area under the Reverb RMS power envelope, integrated from the reverb onset time. In an interactive audio engine for mixed reality, the reverb onset time is at least equal to the propagation delay for a given virtual sound source. Therefore, the calculation of RE for a given virtual sound source depends on the position of the virtual sound source relative to the listener.

Figure 3 illustrates the calculation of the RE for a virtual sound source collocated with the listener, assuming that the reverberation onset time is equal to the time of sound emission. In this case, the RE represents the total energy in the reverberator impulse response when the reverberation onset time is assumed to be equal to the time of emission of a unit impulse by the sound source. This total energy can be computed as the area under the RMS power curve, integrated over time from the time of emission $t = 0$.

3 Conclusion

The model proposed in this paper provides a characterization of all that is needed to fully describe an MR

Property Name	Type	Units	Min	Max
room_gain	float	linear	0.0	8.0
reflections_delay	float	seconds	0.005	reverb_delay
reflections_gain	float	linear	0.0	8.0
reverb_delay	float	second	0.02	0.1
reverb_gain	float	linear	0.0	8.0
reverb_decay_time	float	seconds	0.1	20.0
reverb_decay_time_lf_ratio	float	linear	0.1	2.0
reverb_decay_time_hf_ratio	float	linear	0.1	2.0

Table 3: Room Porperties

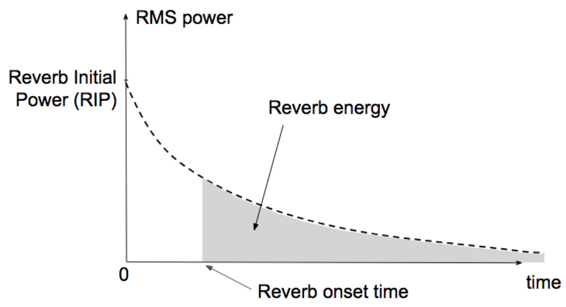

Fig. 2: Reverb Energy

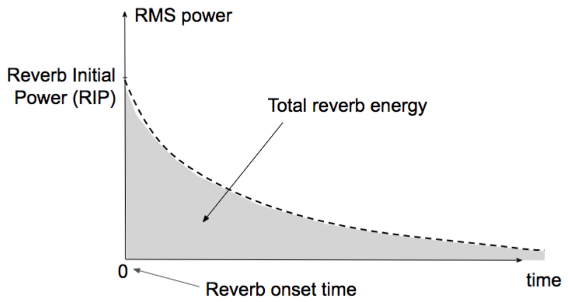

Fig. 3: Reverb Energy - Source-Listener Co-located

experience for the purpose of audio rendering. Having this universal language for describing MR audio rendering enables compatibility across systems and applications, while preserving the opportunity for each rendering tool to differentiate from the competition by balancing complexity and quality trade-offs, and advancing the state of the art in terms of rendering algorithms.

References

[1] Magic Leap, Inc., "Soundfield Audio Component Reference," SDK v.0.15.0 Documentation, 2018.

[2] Hiebert, G., "OpenAL 1.1 Specification and Reference," Standard, Creative Labs, Singapore, 2005.

[3] Peacock, D., Harrison, P., D'Orta, A., Carpentier, V., and Cooper, E., "OpenAL - Effect Extension Guide," Standard, Creative Labs, Singapore, 2000.

[4] Interactive Audio Special Interest Group, "Interactive 3D Audio Rendering Guidelines - Level 2.0," Standard, MIDI Manufacturers Association, Los Angeles, CA, 1999.

[5] Audfray, R., Jot, J.-M., and Dicker, S., "Practical Realization of Dual-Shelving Filter Using Proportional Parametric Equalizers," in *Audio Engineering Society 145th Convention*, 2018.

[6] Jot, J.-M. and Lee, K. S., "Augmented Reality Headphone Environment Rendering," in *AES International Conference on Audio for Virtual and Augmented Reality*, 2016.