RENDERING MPEG-4 AABIFS CONTENT THROUGH A LOW-LEVEL CROSS-PLATFORM 3D AUDIO API

Jean-Michel Trivi, Jean-Marc Jot

Creative Advanced Technology Center, Scotts Valley, California, U.S.A.

ABSTRACT

The Advanced Audio Binary Interface For Scene Description of the MPEG-4 standard version 2 offers two acoustical models, one perceptual and the other physical, for the description and rendering of virtual audio environments. In order to facilitate the implementation of an MPEG-4 AABIFS renderer, we describe translation strategies to a cross-platform lower-level rendering model, EAX 4.0. This environmental audio API is here used to extend OpenAL, a cross-platform 3D Audio API through which the audio is streamed. A rendering architecture for the control and playback of the AABIFS stream is presented.

1. INTRODUCTION

In an MPEG-4 terminal supporting AABIFS content, positional and environmental rendering is the final process applied to audio data resulting from the audio composition graph of the decoded audio streams. In the context of the development of such a terminal, being able to take advantage of existing APIs, potentially offering hardware acceleration, reduces the development effort and enhances the performance of the resulting software. In order to facilitate such an approach, we review the 3D audio concepts and environmental models found in MPEG-4 Audio BIFS. We then review two existing APIs, OpenAL and EAX, the latter extending the first, and we describe how these APIs can be used to implement AABIFS spatialization.

2. MPEG-4 AUDIO BIFS VERSIONS 1 AND 2

2.1. Spatial audio presentation with MPEG-4 Audio BIFS

Borrowing the concept of *scene graph* from VRML97, MPEG-4 allows the hierarchical description of interactive spatialized sound scenes through its binary format, BIFS (Binary Format for Scene description). MPEG-4 AudioBIFS version 1 defines a number of nodes aimed at the creation of a scene graph describing how the audio content of the MPEG-4 scene is composed and rendered [1]. Of particular relevance to this article is the *Sound* node: it is used to attach 3D rendering properties to an audio subgraph that is topped by a *Sound* node. These properties define: the absolute position of the sound in the scene, its intensity (attenuation or amplification factor) and two ellipsoids defined by the sound's direction and front/back distances which characterize how the playback volume of the sound changes according to the relative position of the listener. Audio content associated with this node is to be spatialized by the audio renderer of the MPEG-4 player. This content originates from the composition graph defined by the `source` field of the *Sound* node. When the number of channels resulting from this composition graph is greater than one, those channels are down-mixed to a monophonic signal before entering the spatialization stage of the player (as is done in similar 3D sound APIs such as DirectSound3D, OpenAL, and the Java3D audio engine).

2.2. Advanced AudioBIFS directive sound sources

Version 2 of MPEG-4 [2] enhanced the palette of available audio tools for 3D sound scene description by offering in the Advanced AudioBIFS a refined model including environmental rendering, in order to improve the realism of MPEG-4 sound scenes.

2.2.1. Directive sound sources

AABIFS defines a new node, *DirectiveSound*, similar to the *Sound* node in its role to present monophonic data to be spatialized to the 3D renderer.

The main difference lies in the possibility of simulating realistic directivity characteristics by either specifying parametric filtering associated to radiation angles or triplets of angle, frequency and attenuation. The *DirectiveSound* node offers means for adjusting its own speed of sound, as well as the distance at which a 60dB attenuation is applied, and outside of which it is not heard. Also, it is possible to specify for this node whether the distance dependent low pass filtering used to simulate air absorption is applied.

2.2.2. Environmental model: physical approach

In a typical room impulse response, the room effect can be characterized by the properties of early reflections and of the late reverberation. The physical approach in AABIFS introduces two new nodes from which the room effect properties can be derived for a specific sound source de-

0-7803-7304-9/02/$17.00 ©2002 IEEE

scribed by a *DirectiveSound* node, whenever this latter is in the same defined region as the listener.

The *AcousticScene* node defines such a region as a bounding box of rectangular section. It also contains the specification of the reverberation time for a list of frequencies specified in the node. A reverberation level and delay (i.e. how long between the direct sound and reverberation) are available as well. *AcousticScene* is also a scene graph grouping node and can be used like the *Group* node for that purpose.

While the *AcousticScene* node might be used solely to describe a room response for lower computational resources, a second node, *AcousticMaterial*, can be used as a sibling or child to define a more accurate simulation of the room acoustics. It is based on the geometry and the acoustical properties of room boundaries. These properties are the reflectivity and the transmission characteristics of the material associated with each one of the polygons that define the geometry.

The physical approach is based on the DIVA system developed at the Helsinki University of Technology [3].

2.2.3. Environmental model: perceptual approach

Unlike the previous approach, where the room effect is tied to a specific region of the virtual presentation space, the perceptual model associates a room effect with the *DirectiveSound* node itself. The *PerceptualParameters* node is referenced by the *DirectiveSound* node as a field and defines a set of environmental parameters that describe how the listener will perceive the interaction between the room and the sound source.

As defined in the MPEG-4 AABIFS specification, these high-level perceptual parameters are used to alter a generic reverberation response model, which is divided into four sections: the direct sound (R0), the directional early reflections (R1), the diffuse early reflections (R2), and the diffuse late reverberation (R3). Each of those is bound temporally by 4 time limits (l_0, l_1, l_2, l_3) and further divided into three spectral regions (low, mid and high) by two reference frequencies (f_{low}, f_{high}) as shown on figure 1.

Among the nine perceptual parameters, three determine the room's late reverberation decay time and its relative variations for low and high frequencies.

The other six parameters are typically particular to each source and may vary dynamically according to its position in a room: the energy of the late reverberation, the early decay time, the ratio between the early reflections and the direct sound, the combined energy of the direct sound and directional early reflections, and its relative variations for low and high frequencies.

The *PerceptualParameters* node also offers the possibility to define two sets of gain coefficients for the low/mid/high frequency bands. One affects the direct sound and the other the reverberation.

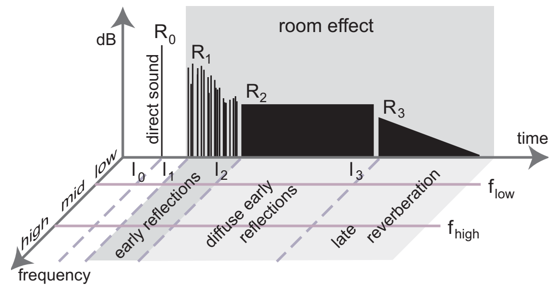

Figure 1: Generic reverberation response model

The perceptual approach was originally developed by IRCAM and France Telecom for the generic simulation of performance room acoustics [4].

3. LOW LEVEL 3D AUDIO RENDERING API

3.1. OpenAL

The Open Audio Library [5] is a software programming interface to the real-time generation of 3D audio based on an object oriented paradigm. The API uses a style similar to OpenGL's. For the rendering of an audio context, implicitly associated with an OpenAL process, the API defines three fundamental objects: the Listener, the Buffer and the Source. The Listener object is unique to each context. Upon creation, Buffers and Sources are assigned a name (an index) that is used to designate them, until they are released using the built-in deletion mechanism.

OpenAL offers a model for logarithmic attenuation relative to the source-listener distance, parametrized by a roll-off factor and a reference distance (equivalent to the model used in IASIG I3DL2 [6] and DirectSound3D). The amount of Doppler effect can be adjusted for each source.

The listener and source objects can be positioned in space, oriented, and assigned a velocity vector used for Doppler effect calculation[1]. The source model is refined by parameters that determine the computation of the frequency independent attenuation based on distance and orientation: minimum and maximum gains, reference distance and maximum distances, roll-off factor, inner and outer directivity cone angles, and attenuation in the outer cone.

A sound source described by a Source object is emitting data originating from its associated playing buffer. Buffer objects can be filled with the appropriate data to be used by the application and shared by several sources simultaneously. Each source contains a queue of buffers of arbitrary length, which are processed (used for playback) sequentially. This mechanism allows for the streaming of audio content for an associated source by monitoring the state of the queue to fill the buffers that have been processed with newly available data.

[1] The velocity field is not automatically computed by the OpenAL implementation according to the position updates. The application needs to take care of the update in order to produce Doppler effects.

Calls to manage playback can be addressed individually to each source, but a vector variant is also specified to allow for the synchronization of multiple sources. Such a feature can for instance be used to play simultaneously a sound source and several image sources modeling delayed reflections.

Software implementations of OpenAL are available for Microsoft Windows, MacOS and Linux. An implementation taking advantage of hardware acceleration is also available for Windows PCs with Creative SoundBlaster cards.

3.2. EAX

EAX is an environmental rendering API that can be accessed as an extension to 3D audio APIs such as OpenAL and DirectSound. It defines several objects as sets of properties associated to environments or sound emitters.

3.2.1. Low-level parameters

The reverberation model used in EAX is based on a decomposition of a typical room response as a group of early reflections followed by an exponentially decaying reverberation. It is characterized by a set of low-level reverberation parameters that are compatible with the IA-SIG I3DL2 guideline: level and initial delay of the reflections and of the reverberation, high-frequency attenuation, decay time at medium and high frequencies, modal density and diffusion (or echo density) [5, 6].

Version 3.0 of EAX includes advanced reverberation parameters: low-frequency level and decay time, directional panning of the reflections and of the reverberation, as well as periodic echo and a periodic pitch modulation (for special effects), both having adjustable salience and period.

Access to all the parameters of the reverberation engine allows for smooth audio transition from one environment to another by interpolating the parameters.

Low-level per-source parameters are provided to control the level of the direct path and of the reflected sound at low and high frequencies.

3.2.2. High-level features

While the low-level parameters allow a sound designer to create the acoustical signature of an environment to be simulated, access to higher-level parameters provide a simpler way to control various aspects of the audio scene. Some of these features are based on a statistical model of reverberation in rooms [7].

EAX offers three per-source muffling effects to facilitate the simulation of obstacles, partitions or openings standing between a source and the listener: occlusion, obstruction and exclusion (see figure 2). These effects combine in a single parameter the attenuation and low-pass filtering of the direct-path sound, the reflected sound or both.

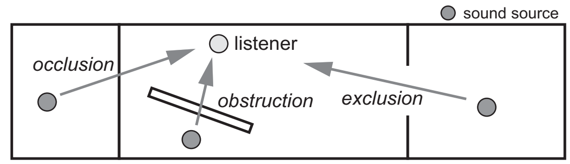

Figure 2: Muffling effects

EAX also enhances the distance and directivity models of the DirectSound3D and OpenAL APIs by providing means for tuning the characteristics of air absorption, for making the directivity of a sound source frequency-dependent and for automatically adjusting the attenuation and spectral content of the reverberation according to these parameters.

The set of reverberation parameters also includes an `EnvironmentSize` property, designed to simulate the scaling of room dimensions by adjusting all the relevant low-level parameters accordingly.

3.2.3. EAX 4.0 features

Version 4 of EAX exposes three classes of interfaces (also named property sets) to the programmer: Source, Environment and Listener. EAX 4.0 allows for the instantiation of up to four reverberators operating simultaneously. The main role of the Listener interface is to designate the environment in which the listener is located.

In order to simulate a sound feeding multiple environments, as it is the case in adjacent rooms for instance, the Source property set defines the energy that the sound source sends to each Environment (at low and high frequencies).

4. TRANSLATION FROM THE MPEG-4 MODELS TO THE LOW-LEVEL RENDERING MODEL

4.1. Advanced AudioBIFS Rendering Levels

The MPEG-4 standard defines four Levels that specify the resources available to the terminal for AABIFS content rendering. The combination of OpenAL and EAX compare to the four requirement levels as follows:

Number of sources: OpenAL does not specify the number of simultaneous sound sources, this is up to the specific implementation and is not a limitation of the API.

Number of temporal sections individually controlled for each source: EAX offers two temporal sections, early reflections and late reverberation, as specified in Level 3. However, directional early reflections (R_1) can be rendered by use of individual discrete sources to meet Level 4 requirements.

Number of reverberators: EAX provides control to four reverberation processes, as required by Level 4.

Number of control frequencies: the EAX API exposes three frequency bands for reverberation parameters (matching Levels 3 and 4) and two for source directivity filters (as in Level 2).

4.2. Source parameters

Position and orientation can directly be mapped to the OpenAL source parameters. EAX and OpenAL offer two angles and frequencies to specify source directivity. If more are used, values that best approximate the filters should be applied. Turning air absorption on and off is achieved by modifying the EAX source `AirAbsorptionFactor` respectively to 1 or 0. The `SpeedOfSound` parameter, which controls the amount of Doppler effect and defines an initial delay before a sound is heard, can be simulated via the OpenAL `DopplerFactor` and by queuing a "silence" buffer of the appropriate length in the source.

4.3. Environmental effects

4.3.1. Physical approach

Similar to a first-person 3D game engine, a renderer managing a scene described by the physical approach takes into account the positions of the sources and of the listener relatively to the scene geometry (room boundaries, obstacles). When the position requirements are met, the properties of the *AcousticalScene* node can be directly mapped to the late reverberation section of an EAX reverberation. Strategies discussed in [3] allow for the computation of the characteristics of the source-specific early reflections with image sources, while the diffuse early reflections, adjusted by the relative position of the listener to the walls, are mapped onto the EAX reverberation parameters.

4.3.2. Perceptual approach

Rendering the perceptual parameters consists in the simulation of R_1, R_2 and R_3. Because EAX doesn't support R_1, two simulation strategies can be chosen:

a) discrete sources are used to simulate R_1, or
b) R_2 is omitted and R_1 is rendered through the EAX Reflections and their directional adjustment (pan vector).

This second approach offers the advantage of using fewer 3D sources while still conveying the main room effect characteristics.

Because the perceptual model requires a reverberator for each source, the following strategy can be used if more than four sources have associated perceptual parameters: (1) identify the four "prominent" sources, and (2) feed the remaining (less prominent) sources into their "best-match" reverberator (i.e. the one with the closest relative energy ratios and time limits for R_1, R_2 and R_3).

5. RENDERING ARCHITECTURE

The interpretation of the perceptual and physical nodes of an AABIFS MPEG-4 scene is provided to the renderer of an MPEG-4 terminal by the audio scene graph management layer. The latter extracts, for each sound node, the audio composition graph which determines both the life-cycle of the node and its parameter updates, as well as the way the associated data is to be computed. The audio data stream is demultiplexed and decoded before being composed. In the case of directive sound nodes, this data can then be streamed through the OpenAL buffer queuing mechanism, while energy parameters are translated into EAX parameters.

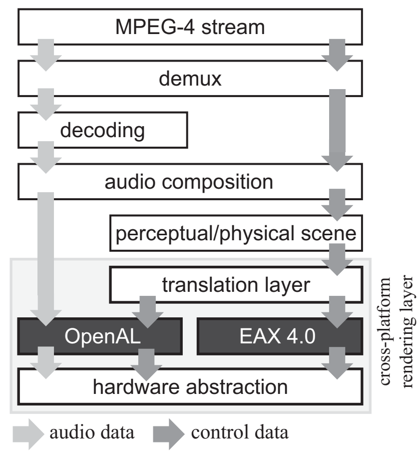

Figure 3: Rendering architecture

6. CONCLUSION

An architecture was described which allows for the rendering of MPEG-4 AABIFS scenes via the cross-platform OpenAL/EAX API. The deployment of MPEG-4 AABIFS spatialization may be facilitated by taking advantage of an existing rendering platform. The AABIFS environmental spatialization models (physical or perceptual, from Level 1 to Level 4) translate well to the EAX 4.0 model. The limitations that remain to achieve full compliance to the Level 4 requirements may be addressed in future extensions of the EAX API.

7. REFERENCES

[1] Eric d. Scheirer, Riitta Väänänen, Jyri Huopaniemi, AudioBIFS: Describing Audio Scenes with the MPEG-4 Multimedia Standard. *IEE Transactions on Multimedia*, Vol. 1, No 3, pp. 237-250, September 1999.

[2] ISO/IEC 14496, MPEG-4 Standard, Amendment 1. "Information Technology – Coding of audiovisual objects. 2000.

[3] Savioja & al. *J. Audio Eng. Soc.*, vol. 47, no. 9, pp. 675-705, 1999.

[4] Jot. *Multimedia Systems*, Vol. 7, no. 1, 1999.

[5] Creative Labs (2001). SDK Downloads (EAX, OpenAL, EAGLE). http://developer.creative.com/

[6] IA-SIG (1999). 3D Audio Rendering Guidelines, Level 2 (I3DL2). http://www.iasig.org/

[7] Jot, J.-M., Cerveau, L., Warusfel., O., Analysis and synthesis of room reverberation based on a statistical time-frequency model. *Proc. 103rd Conv. of the Audio Eng. Soc* (preprint no. 4629), 1997