AN INDEPENDENT DECLARATIVE 3D AUDIO FORMAT ON THE BASIS OF XML

Heiko Hoffmann, Raimund Dachselt, Klaus Meissner

Dresden University of Technology, Multimedia Technology Group, 01062 Dresden, Germany
`[hhoffmann,dachselt,kmeiss]@inf.tu-dresden.de`

ABSTRACT

This paper describes the development of an XML-based format called Audio3D for the declarative description of acoustic environments and sound sources for 3D auditory displays by an audio designer without the need of programming efforts. The format is platform and API independent and suitable for real-time and offline sound rendering. It can be used together with other XML-formats for 3D graphics such as X3D and is based on the concept of a hierarchical scene graph. Acoustic environments can be described in any level of detail using reflecting and absorbing surfaces or reverberation parameters for an abstract representation of multiple acoustic rooms.

1. INTRODUCTION

Nowadays 3D sound technologies can be found in almost all existing sound hardware. Various surround speaker systems that can be easily attached to computers are widely available. The main application area of 3D sound is still the field of computer games, but for many other use cases like product presentations, virtual environments, auditory displays and musical applications three-dimensional sound gains more and more importance.

The application programmers have to deal with different APIs of sound card and operating system producers, which usually implies platform dependency and a high programming effort. On the other side declarative 3D-graphics formats can be found that describe three-dimensional scenes using scene graphs, which can be interchanged between applications and platforms. The primary aim of the development of **Audio3D** was to create an XML-based declarative format for the description of three-dimensional sound for auditory displays. A hierarchical audio scene graph should be deployed in a high level way independent of specific sound APIs and sound reproduction systems. It can be used together with real-time 3D-graphics for interactive 3D-applications but also for high quality offline sound rendering. Parts of an Audio3D-document can be accounted as audio components of a 3D scene and reused in other documents and applications.

This article first introduces related work on sound APIs and 3D-graphics formats. It then describes the general structure of the Audio3D format and outlines its features in more detail. At the end examples for using the format are given and an outlook on future developments concludes this work.

2. RELATED WORK

Audio3D was developed in the context of common 3D-sound APIs like DirectSound 3D and the more platform independent OpenAL [1], which follow the guidelines for 3D-sound features that were specified by the Interactive Audio Special Interest Group (IASIG [2]). These guidelines were used as the basis for the definition of audio parameters in Audio3D.

To get a high level view of an acoustic scene a scene graph with geometrical descriptions is used, which is based on 3D-graphics formats such as VRML and its successor X3D [3]. These formats include features of 3D sound in terms of sound nodes that can be integrated into the scene graph. That way the sounds get a position in space and can be rendered for directional hearing with attenuation according to the distance to the listener's position. But a major part of the sound impression is based on reflection and diffraction of sonic waves in the environment hearable as a reverberation sound that decays with a time delay.

Until now only MPEG-4 [4] with its declarative format XMT also allows the description of complex acoustic environments with reverberation, occluded sounds, single reflections and sound filtering of the propagation medium. However its complicated structure and the lack of authoring tools and players especially for these features make it difficult to use. Other disadvantages are missing features for an efficient simplified description of acoustic scenes and audio parameters that are difficult to map on common 3D-sound APIs.

3. STRUCTURE AND FEATURES OF AUDIO3D

With Audio3D a variably usable, API-spanning declarative format was created that allows to define complex acoustic scenes with an audio scene graph. It may contain animated sound sources, obstacles and room walls that can reflect and absorb sonic waves as well as abstract rooms described with reverberation parameters and medium properties.

The format clearly divides the geometrical data for visual rendering (e.g. given in a common 3D-graphics format such as X3D) from the data for sound rendering. This makes it easier to implement the actual audio rendering in applications employing 3D-sound APIs separated from rendering graphics. It also allows a simplification of geometrical descriptions for the acoustic environment (for example geometrical primitives like boxes for areas with reverberation properties) that are sufficient for sound calculation but should not be visualized by the graphics renderer.

3.1. Audio3D Feature Levels

To make the format suitable for different use cases and demands and at the same time easy to use it is divided in three feature levels. Each level defines an extended set of XML-elements and attributes and indicates which features should be implemented in an application depending on its use cases. Audio3D documents can be validated against the separate XML-Schemas for each level. Table 1 provides an overview of the assignment of format features to the three levels and lists typical use cases.

Level	Core	Common	Full
Meaning	skeletal structure, essential audio scene graph properties	extension with room acoustic properties	desirable properties for complex audio scenes
Features	scene graph with transformations, distance attenuation, Doppler-effect, variable listener's position, point sound sources, animations	occlusion and obstruction by obstacles, rooms with reverberation, openings between rooms, levels of detail, priorities	single reflections, medium properties, size of sound sources, sound sources for special purposes
Use Cases	sounds for interaction feedbacks and user information, musical applications without room simulation	telecommunication in virtual rooms, entertainment applications	Virtual Reality, sound simulations
Player implementation	most features provided by sound-APIs, low programming effort	partly supported by sound-APIs, higher programming effort	scarcely supported by sound-APIs, very high programming effort

Table 1. *Features and use cases of the Audio3D levels*

3.2. Hierarchical Structure and Transformations

Similar to X3D a hierarchical, acyclic scene graph is used in Audio3D to organize nodes in groups and subgroups. Each group node separates a new branch in the scene graph and can include **Transformation**-nodes that transform the coordinates and sizes of all following child nodes within the group. The group nodes are called **AudioGroup** and can contain any number of transformations, subgroups, descriptions of sound sources, rooms with reverberation and objects which obstruct and reflect sound waves. These descriptions can also be part of a special group node called **LOD** (Levels of Detail). It consists of multiple AudioGroup-nodes containing different details. The player chooses one of these nodes depending on the current listener distance and rendering latency to save calculation time in the real-time rendering process.

The main AudioGroup-node, the listener and environment descriptions and parameters for the interpretation of the scene content are put together in a scene's root node called **AudioScene**. The following code fragment serves as an example of an Audio3D document and illustrates some typical elements and attributes. It describes two sound sources with transformations and a reverberant room. The empty spaces (..) in the code stand for additional parameters and sub-nodes.

```
<Audio3D xmlns=".." ..>
  <Header title=".." author="..” ../>
  <AudioScene coordinateSystem="righthanded">
    <Environment rollOffFactor="0.5" ../>
    <Listener position="0 0 1.8" ../>
    <AudioGroup>
      <Transformation center=".." rotation=".."/>
      <AudioGroup>
        <Transformation scale=".." ../>
        <Room priority="1">
          <Reverb decayTime="2" ../>
          <Walls attenuationTransHF=".." ../>
          <Transformation translation=".."/>
          <Box size="10 20 3" DEF="OurBox"/>
        </Room>
        <Source position=".." startTime="..">
          <Sound format="WAV" location=".."/>
        </Source>
      </AudioGroup>
      <Source ..> ..
      </Source>
    </AudioGroup>
  </AudioScene>
</Audio3D>
```

All Nodes in Audio3D can be identified with a name using the attribute **DEF** and referenced by name inside the attribute **USE**. If an AudioGroup-node is referenced all child nodes are reused too. To reuse only a transformation in another group with different content the Transformation-node can be referenced by its identifier within the other group.

For audio properties that are repeatedly used in different documents Audio3D offers the definition of presets. The presets are elements like in the audio scene graph but are stored in a separate file that can be included in an Audio3D-document. The predefined elements can then be referenced in the document.

3.3. Description of Sound sources

Each sound source in the scene is represented by a **Source** node. Depending on the used format level different parameters are possible to use. In the Core-level these are the source position, its radiation direction and angle, start and stop time for playing and ranges for distance attenuation. If a sound source shall be played always in front of the listener without a special position (for instance a background music clip) the **FrontSource**-node can be used. With this source type both mono and stereo sounds can be played while normal source nodes play all sounds in mono.

In the Common-level a priority attribute can be used to specify which sources are more or less important for the sound rendering. The Full-level then adds attributes to define a size for a volumetric source that radiates sound not only from one point but from a larger area. The noise from a long train for example can be considered as a wide-stretched sound source. To keep it simple the source volume can only be described in form of a cuboid. The Full-level also introduces a set of source nodes for special purposes that are useful in virtual reality applications:

- **MovementSoundSource** describes a sound that is dependent from its movement velocity like engines or rolling noises. Special attributes of this node control the sound's volume and pitch dependent from the velocity of the source.
- **MediumSoundSource** creates a sound that represents the resistance of the propagation medium against the movement of an object such as a car. This node provides control of the

sound's volume and pitch depending on the relative speed of the source to the medium.

- **AmbientSource** defines a background sound that is heard uniformly from all directions, for instance the noise of leaves in a forest.

3.4. Description of the Acoustic Environment

The simulation of a reverberant environment with multichannel sound rendering can be accomplished by calculating reflection paths for specular early reflections and mixing a diffuse reverberation sound to all audio channels. The simulation of single reflections requires a geometrical description of reflecting surfaces and the rendering of virtual sound sources for each reflection that reaches the listener's position (as depicted in figure 1). Since this needs much computation time it can be done in real-time applications only for a limited amount of reflections and should only be used for special situations such as big stonewalls or far distant echoes. An optimization of this technique that is suitable for real-time rendering at the expense of quality is called Beam Tracing and is described in [5]. The mixing of a reverberation sound to the direct signals of the sound sources is more efficient for diffuse late reverberation and less important early reflections because no directional information is needed in these cases and so a path tracing has not to be done.

In the Common-level of Audio3D the acoustic environment can be described with several reverberation areas that are defined geometrically by graphical primitives like boxes, cylinders, cones, spheres and face sets which follow the X3D standard. Each area is specified with the element **Room** that contains both the graphical primitives and reverberation settings, which follow a general reverberation model consisting of some early reflections and many diffuse and decaying late reflections. Such a model and its typical parameters are described in the "Interactive 3D Audio Rendering Guidelines" by IASIG [2]. If the listener's position lies inside such an area, as shown for example in figure 1, a reverberation sound is generated for all sound sources depending on the reverberation levels that can be defined separately for each sound source. If the defined area shall be considered as a closed room the child node **Walls** can be used inside the Room-definition. This node contains parameters to set the attenuation rate for low and high frequencies from sound sources that are outside the room while the listener's position lies inside. In this case the attenuation affects both the direct sound level from the outside source and the reverberation sound originated from the room reflections. If the sound source is situated inside the room but behind an obstacle only the direct sound is muffled but not the reverberation effect. For this case the Audio3D format offers the node **Object**, which contains the same attenuation parameters like **Walls**.

For other areas that have openings such as doors and windows the player generates virtual sound sources that play all sounds from inside of these areas at the positions of the openings. Two example virtual sound sources are depicted in figure 1. Such openings can be specified with the child node **Opening** inside the Room-definition. The position and shape of an opening is described by polygons with the X3D-Node **IndexedFaceSet**. Through openings the reverberation sound from inside of a room can also be heard outside. It depends however on the features of the player and the audio system, whether localized reverberation sound can be generated or not.

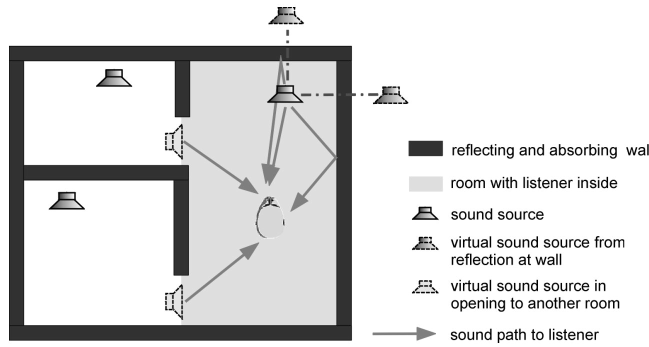

Figure 1. *Reverberant rooms with walls and openings*

For overlapping areas a parameter called **priority** tells the player which settings shall be used if the listener is standing inside the overlapping area. If the priorities are equal a smooth transition between the reverberation settings is used by linear interpolation of the parameter values dependent on the listener's position inside the area.

It is important to consider also the **diffraction** effect for sound sources moving around obstacles. The sound level of a source that vanishes behind an obstacle fades away smoothly instead of dropping abruptly. Higher frequencies get more attenuated than lower frequencies. Since the calculation of diffraction is very difficult (see [6] for details), Audio3D offers a simplification. With special parameters for the environment definition the attenuation of sounds behind an obstacle can be made dependent on the angle between the lines originating in the listener's position and leading to the sound source and the edge of an obstacle.

The Full-level extends the Walls- und Object-nodes with parameters to define the reflection behavior of obstacles for low and high frequencies. With these attributes single reflections can be calculated. For each sound source the author can also specify up to which distance single reflections should be generated. Since an exact reflection calculation is very time-consuming this is important for efficient real-time sound generation. Single reflections should be used with care and only for important sound sources and obstacles in the scene. If single reflections are used together with reverberation they should be considered in the reverb simulation by reducing the amount of simulated early reflections or the overall reverb level.

Another extension of the Full-level is the specification of the element **Medium** that defines acoustic parameters for the propagation medium and can be associated with room definitions and the whole environment. Possible parameters are the sonic speed, a velocity vector for a moving medium and the absorption rate relative to the distance of high frequencies. The medium can be simulated by the player with a low pass filter and a delay effect for single reflections.

3.5. Predefined Behavior with Animations

Another feature of the Audio3D-format is an easy description of playtimes and animations of sound sources. This way it can also be used as an interchange format for musical applications and presentations where audio objects are associated with a predefined behavior. To realize behaviors the Audio3D-elements can contain an **Animation** child node that defines keyframe values for the attributes of its parent node. The keyframes describe attribute values for certain time stamps which are compared with the actual playing time. The values between keyframes are interpolated linearly. Elements whose attributes can be animated are sound sources, the listener, transformation nodes, objects, room walls, openings and the medium.

The Doppler-effect plays an important role for a realistic impression of moving sound sources. However, for special purposes, e.g. spatial music, the effect can be undesirable. That is why Audio3D allows defining the strength of the Doppler-effect simulation.

4. USAGE OF THE AUDIO3D FORMAT IN APPLICATIONS

The usage of Audio3D is possible with different types of applications. Depending of their aims the applications can work with real-time or offline sound rendering. In the case of auditory displays a real-time sound generation is necessary to allow the application to react on user interactions. Audio3D is especially suited for this application case by means of its features for a simplified description of the acoustic scene and its orientation towards common 3D-sound APIs.

4.1. Application Types

Since the format is platform independent different kinds of applications are possible that use Audio3D:

- Standalone applications can use the format to store its data and also to interchange it with other applications.
- A Web browser plug-in can process Audio3D documents that are loaded from the Internet, for example in the context of product presentations. Another possibility is to use a Java3D applet that processes the document.
- A server side application can generate audio streams from the information of an Audio3D document and send them to one or more clients. In the other direction client applications can control the generation process by sending Audio3D descriptions to the server. This can take place for example in a teleconferencing system, where Audio3D describes a virtual conference room.
- With a stylesheet and an XSLT-processor an Audio3D document can be transformed into another 3D format such as X3D or MPEG-4. This might cause a loss of information because the transformation is limited to the functionality of the target format.

4.2. A Simple Player-Application for Audio3D Documents

For the demonstration of the Audio3D features and its reproduction with common sound APIs a player application was developed that uses DirectSound 3D and EAX for real-time sound generation. It runs under Windows. Audio3D XML input files are read in with a SAX parser that sends events for all elements of the document to the player. The specified sound samples are loaded as streams in DirectSound 3D buffers, which are updated in intervals during playing. EAX is used to simulate environments with reverberation as described declaratively. The player also calculates the attenuation of sound sources behind simple objects, which were described with graphical primitives. It thereby considers the diffraction effect. In the current version the Core-level and parts of the Common-level are well supported. To illustrate the easy connection and relation to 3D graphics scenes the player also visualizes the simulated environment and sound sources with 2D or 3D rendering.

5. CONCLUSION AND FUTURE WORK

The novel XML-based format Audio3D allows the declarative description of complex acoustic scenes independently from the visual scene description. This allows using much simpler geometrical models for the real-time sound rendering process and more detailed graphical models for the 3D visualization. The format offers novel features such as the description of overlapping or connected rooms, specialized sound sources and a simplified diffraction model. The features are still experimental and have to be tested and discussed together with acousticians and application developers. New features can be introduced easily without the need of changes in 3D-formats such as X3D or MPEG-4. The mapping of Audio3D-documents to various sound APIs is possible. To show the capabilities of the format a player application was developed, which renders 3D-sound with common APIs in real-time and visualizes the sound sources and acoustic environments. In the near future the player application will be extended with more format features. As soon as available new functionality of 3D sound APIs will be integrated, too.

Audio3D was developed in the context of a component-based 3D-application framework called CONTIGRA [7]. Further details about Audio3D can be found on the project Web pages. Further developments will include an editor application for the graphical manipulation of sound parameters within the CONTIGRA framework. After having implemented all format features Audio3D is considered to become a special profile for extending the 3D graphics standard X3D.

6. REFERENCES

[1] Open Audio Library, http://www.openal.org/

[2] Interactive 3D Audio Rendering Guidelines Level 2, Interactive Audio SIG, http://www.iasig.org

[3] X3D-Standard by the Web3D-Consortium, http://www.web3d.org/x3d/

[4] Overview of MPEG-4 standard by the Motion Picture Expert Group, http://mpeg.telecomitalialab.com/

[5] T. Funkhouser, P. Min, I. Carlbom: Real-Time Acoustic Modeling for Distributed Virtual Environments, *Proceedings of ACM SIGGRAPH 1999*, Princeton University, Bell Laboratories, USA, 1999

[6] N. Tsingos, T. Funkhouser, A. Ngan, I. Carlbom: Modeling Acoustics in Virtual Environments Using the Uniform Theory of Diffraction, *Proceedings of ACM SIGGRAPH 2001*, ACM Press, New York, 2001, pp. 545-552

[7] Project CONTIGRA, Chair for Multimedia Technology, Dresden University of Technology http://www.contigra.com