Sound in COLLADA

Shih-Han Chan, Cecile Le Prado, Stéphane Natkin,
Guillaume Tiger, and Alexandre Topol

CEDRIC, CNAM, 292 Rue Saint-Martin
75003 Paris, France
shihhan.chan@gmail.com, cecile.leprado@free.fr,
stephane.natkin@cnam.fr, tiger.guillaume@gmail.com,
alexandre.topol@cnam.fr

Abstract. Standard or normalized file formats exist since many years to write/read/exchange 3D scene descriptions. However, these descriptions are mainly for visual contents. Options given for sound compositions of 3D scenes are either lacking or poor. In this paper, we propose to include rich sound descriptions in the COLLADA standard, a commonly used scene language. Our work relies on a research project, the goal of which is to define and develop a sound engine for virtual cities. In this context we have implemented and experimented a first version of sound descriptions in COLLADA.

Keywords: COLLADA, Dynamic Audio, Audio Scene Description, Soundscape, Virtual Cities.

1 Introduction

The CEDRIC (Computer Science laboratory of the CNAM) is involved in TerraDynamica, a project funded by the French government. The purpose of this three-year project started in 2010 and developed by a dozen of academic and industrial partners is to bring life to TerraNumerica, a static virtual city. This virtual city (Paris in the experiment) will be used for numerous applications from safety simulation to art installation. Bringing life is basically putting animated characters in the city. Nevertheless, making pedestrian and car agents behave properly wouldn't be realistic if they were muted. Even though sounds are often simply noises in a city, they still carry important information for real or virtual citizens. The integration of sounds in TerraDynamica is one of the tasks assigned to the CNAM in this project.

Nowadays, spatial audio is considered as an essential element to deliver a more immersive and believable virtual reality experience. A sound is not merely an emission from a source. That is, sound waves can be absorbed (attenuated), reflected or refracted by the medium. Audio in soundscape theory, architectures and video games, is characterized by its strong relationship to space regarding both analytic and creative perspectives. This non-linear relationship exists through time and is developed within many cross-disciplinary works [1].

J. Anacleto et al. (Eds.): ICEC 2011, LNCS 6972, pp. 235–246, 2011.
© IFIP International Federation for Information Processing 2011

Numerical simulations of the propagation of the sonic wave in acoustic spaces are presented for architectural design. In some interactive media, such as video games, dynamic factors are then added-up in accordance with the gameplay design. In addition, game sound middleware provides a choice of high-level runtime capabilities, such as aural levels of details [2], interactive spatialization mixing, and runtime reverberation and filtering [3].

TerraNumerica, the static part of the city, composed by the terrain, roads, buildings or furnishings, is depicted in a database which is translated to a [1]COLLADA document in order to be utilizable in several applications. Moreover, to be consistent, the visual contents of dynamic agents are also described in a COLLADA file, whereas sound capabilities are not. Consequently, two solutions were then possible for us to add sounds to this virtual muted COLLADA description: First, to describe the whole sound scene independently in a convenient file format that would be generated from the database; Second, to add the sound description in the visual scene description.

Since computing an image viewed and computing a sound heard from a given position in a virtual environment both require a common set of data, it would be disk and memory consuming to duplicate the mutual information. Furthermore, the time required to generate, read and interpret both files would be greater then with a single file in which data is optimized. Hence we have chosen the second solution: to add the sound description in the COLLADA file that has already been used to illustrate the visual elements.

Using visual data for audio purpose is not always done. For example, OpenGL, the well known graphic library, uses transformations that do not affect sound sources and listener positions of OpenAL. On the other hand, some scene description standards have tried such a mix: Java3D on the API side or VRML V2 on the file format side, having sound sources included in the scene graphs and modified by transform nodes. MPEG4 BIFS, inspired by the VRML scene graph, also includes such capabilities, although sound parameters in those files or API are poor compared to what is possible to be done in sound environments nowadays through the use of DSPs [4].

From the earlier goal of adding sound capabilities to TerraDynamica, we ended by inserting sound nodes to the COLLADA file format. The proposition of these sound nodes is the scope of this paper. In the next section we describe the sound attributes and nodes to be embraced to the COLLADA files. Before concluding and presenting our future works, we present the implementations developed to practically validate our approach based on analysis of city soundscapes.

2 Audio Scene Description in COLLADA

Every object or piece of world material in a game or film has a sound it would make if you interacted with it [1]. A sound world is built upon interactivities. As a consequence, we are seeking a scene descriptor in order to plot out the sound behaviors characterized by the dynamics.

[1] COLLADA documents are XML files, usually identified with a .dae (**d**igital **a**sset **e**xchange) filename extension.

2.1 Audio Nodes in Scene Language

A scene language is essentially a formal and hierarchical illustration of a virtual space itself and the objects located in this space. The language can be interpreted by a rendering program that will generate, either in real time or in batch, a perceptive (animated image and sound) representation of the scene. Each node of the hierarchy may contain some properties of an object (location, geometry, physical properties…), the procedural descriptions of the object behaviors which can be triggered by events, and the instructions to be followed by the renderer. The following figure represents a scene described in MPEG4 and VRML.

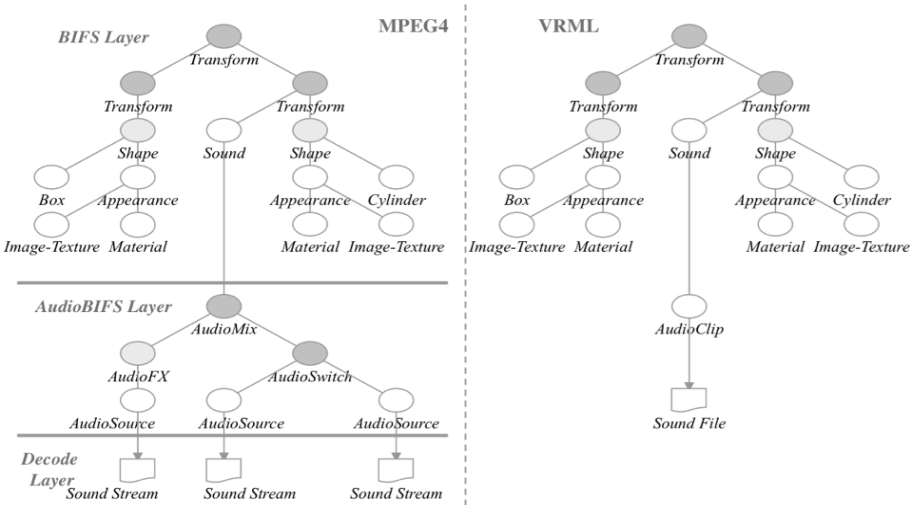

Fig. 1. MPEG4 and VRML scene graphs with sound nodes

MPEG4 [5, 6] provides probably the most advanced scene language considering the sound aspect. As one can see in Fig. 1, the BInary Format for Scenes (BIFS), which is the scene graph of MPEG4 is almost equivalent to the VRML2 scene graph. The main difference between the two scene graphs is that a BIFS can be streamed and can evolve over time if the MPEG4 server sends modifications to the client. For offline interactive or automated contents, MPEG4 has exactly the same paradigm as VRML2: the use of ROUTEs to chain behaviors for nodes to others.

In terms of sound nodes, both VRML2 and MPEG4 may describe the sound behavior of an object of the scene. VRML2 can only refer to an AudioClip which is a sample sound file given through an URL. In contrast, MPEG4 supports sounds that can be a mix of sample sound files (AudioMix), symbolic sound files (MIDI), real time sampled or symbolic streams of sounds. At each step of the mix, some effects (DSP) can be applied to the corresponding stream (through AudioFX nodes), depending on local or global parameters, properties of the scene, and real time events. For example, a sound node could be a mix of a synthetic wind with a sampled music, the level of which would be according to the current level of dramaturgy (real time computed parameter). Depending on the stream that a sound node is connected to, the

room ambiance can be computed either by a physical or a perceptive algorithm. No parameter is given explicitly in the BIFS for one method or the other. It is stream-dependant; hence a communication is a request between the stream generator and the MPEG4 client and server. It requires a heavy on-the-fly calculation to transfer stream based on the user's actions, the scene's evolution and the required renderer.

To our knowledge, the ability to interpret the whole capabilities of MPEG4 scene description has never been implemented. Nonetheless, MPEG4 defines the main structures that must be considered in the design of sound nodes in scene languages even if some recent research capabilities, like sound Level of Detail, have to be added.

In our project, COLLADA [7, 8, 9] is selected as it defines an open standard XML schema from which digital contents of assets can be easily retrieved. COLLADA is an intermediate format for transporting data among various graphic tools and is widely used in applications from Google Earth to most of the games on PS3. The current status of COLLADA contains "visual" and "physics" scene descriptions. The visual part is the classical one, with some extension like the ability to include rendering programs (shaders node). The physics part defines physical properties (mass, elasticity…) of object used. The two descriptions are distinct but there is a mapping between the corresponding objects. The sound description has not been considered yet. The next section describes some basic sound functions that we propose to be specified and implemented in a renderer of COLLADA.

2.2 Sound Nodes in COLLADA

As in the MPEG4 scheme, we shall be able to attach sound streams generated by different methods. Sampled files, symbolic sound files (SAOL), sampled streams or symbolic streams are available and the choice among which depends on the type of objects. A pedestrian, for instance, may have his voice in a sample file whereas the footstep is a synthesized stream. Whatever the origin of the sound is, it is specified by an URL leading to a sound asset or a script, and will be spatialized by algorithms chosen by the COLLADA player. To leave such an open possibility for rendering the sounds spatialization, some parameters must be given in the description of the scene.

As opposed to MPEG4 which relies heavily on client-server architectures, we intend to minimize communications between stream processors and the sound renderers. With MPEG4, to render a spatialized sound with occlusions and reverberation, the parameters needed must be given in the stream processor. The parameters are not attached to/described in the MPEG4 scene directly. Based on the sound renderer selected, these parameters must be computed from some data enclosed in the scene graph to match the visual scene. Hence, a sound stream that does not communicate with either the MPEG4 client or server, holding the BIFS structure, will not be reusable in a different scene. For this main reason, we suggest appending a set of parameters to the COLLADA scene specification so that both physical and psycho-acoustic renderers can produce correct sounds.

Introducing sounds in scene induce an important question about the notion of graph scenes. Historically a graph scene was defined to specify the static part of the scene, in other words, is a typed language but not a procedural one. Yet by nature, sound is dynamic. Of course the types of dynamics object can be specified, except if we

consider that the main goal of such a language is a realtime rendering, the specification of the time and interactive behaviors becomes more and more important. Interactions were introduced in VRML and MPEG4 with simple signals like “mouse click” and the ability to route a signal from a node to another. More complex synchronization scheme was introduced in MPEG4 based on SMIL (XMT-0) [11]. In COLLADA, complex dynamic behaviors are either embedded in a block of source code where you can program features or point to an external procedure. We think that some more general and explicit control scheme should be provided. In the following we make suggestions, although the implementation and the core of this problem is out of the scope of this paper.

We suggest that a sound extension of COLLADA should meet the following goals:

1. Be compatible with the spirit and the syntax of COLLADA.
2. Avoid as much as possible specifying data that are already in the scene description. For example, the room geometry should provide some of the data needed to render the acoustic with a physical method. Physical parameters of objects should be used in physical sound synthesis. In COLLADA, the visual scene and the physical scene are defined as separate hierarchies. To keep the same principle and as a consequence of the two preceding items we suggest adding an Audio Scene. Subsequently, a COLLADA model will be defined by a visual scene, an audio scene and optionally a physical scene. An object of the visual scene can be associated with an object of the audio scene, sharing parameters like the coordinate systems, the location and orientation or the same geometry. However, this mapping is not mandatory. The auditory scene can rely on perceptive and sound design principles not directly related to the location of the objects in the visual scene. It seems to us that this idea is important for fictional applications like games or art installations.
3. Have all the capabilities of MPEG4 and support advanced developments in sounds that were not included in MPEG4. It can be, for example, clustering of sound sources [2] and [10] or realtime synthesis of Foley effects. As a consequence, the following atoms, constructors, libraries and parameters should be either a part of the standard or may be considered as allowed extensions:
    a. Parameters of an object that specifies, either from a physical point of view (acoustic color) or a perceptual one (room ambiance), the acoustic of the object. This should be applied as an acoustic element equivalent to “material” in the visual scene.
    b. Sound sources can be generated by various methods. So we suggest a “sound” node with, for example, some parameters like directionality and volume. This node is the equivalent of “light” in the visual scene. We suggest that a sound source may be qualified according to some semantic and staging point of view. In the next section we propose, as an example, to classify sound sources according to three levels of a soundscape. This is an instruction given to the render to cluster and localize sources in the spatialization step of the sound pipeline. The file format of a sound node must be specified. For instance, the background of a scene can be a loop on a single surround sound file in Ambisonic B format.

c. Microphone, which is the sound counterpart of "camera". A Microphone may be mono, stereo or multi-channel (spatialized sound) and has a given directivity.
d. Basic DSP functions that transform a sampled sound file or stream into another one. This is the counterpart of the nodes "effect" and "shaders" in the visual scene.
e. Constructors of complex DSP - the language used to construct a complex sound node:
    i. Mixers: to transform m sound steams into n sound streams.
    ii. Synchronizers: clocks and timers, play, stop, pause, sequential, parallel playback.
f. Synthesizers that transform a virtual sound file or stream into a sampled one; analyzers that transforms a sampled sound file or s stream into an event, a parameter or a virtual sound stream; and time to spectral transformations.
g. Instructions for the sound renderer: synthesis, DSP algorithm and parameters, mixing instructions depending on the real acoustic (binaural, transaural, multichannel 5.1…).

3 A First Implementation

3.1 Goal of the Experiment

To have a first demonstration of the capabilities of our proposal, we have developed a parser for some of the language elements presented in the previous section and used it in a TerraDyamica demonstration. The image renderer used was Unity (http://unity3d.com/unity/) the sound renderer was Fmod (http://www.fmod.org/). Our implementation choices have been inspired by the analysis of soundscapes [12, 13, 14, 15], and in particular the video game Prototype [3].

3.2 Dynamic Urban Soundscape

Sound in the urban environment can be considered from both spatial and descriptive point of views.

The well known soundscape theory proposed by Murray Schaffer [12] is a useful tool to describe the spatial content of a sound environment, including cities [16]. The theory divides what is audible into three layers of space: background, midground and foreground. Furthermore, soundscape is a source of information about the current state of the environment [17]. Its "acoustic coloration" varies along with weather. Soundscape brings clues regarding the physical nature of the environment and an appreciation of the space surrounding the listener.

According to different listening points of interest, we can apply semantic filters to spread dynamically individual sources within the soundscape perspective. Sound descriptions dedicated to the urban environment and categorization based on everyday users' experience of the city have emerged [18]. These specific studies allow for understanding how urban soundscape is heard from a qualitative and functional point of view.

The techniques used to simulate the urban soundscape range from layering of prerecorded sounds to various methods of synthesis [19, 20, 21, 22].

Prototype, a Radical Entertainment game, is indented to create a dynamic ambience of New York City [3]. Not separating the city zone-by-zone with detailed sounds, the desire was alternatively to base the ambience on the objects inside the game world and their relative densities, populations, and emotional states. To do so, they broke the ambience down into three "tiers:" background, midground, and foreground.

- The background ambiences were a quadraphonic track of the more distant perspective recordings of Manhattan: a Central Park background and a rooftop background that would fade based on the position of the listener.
- The midground layer was composed of a collection of grouped components in the game world, including grouped pedestrians, vehicles and infected crowds.
- Lastly, foreground ambiences were a composition of sounds from single objects in the environment, played based on the changes of state of these objects.

A smooth transition from foreground through midground to background provides a sense of aural depth of the field while limiting the number of individual foreground sound components thanks to the support of the midground ambiences.

Besides Prototype's audio levels of detail approach, several source clustering strategies have been described on the basis of geometry and resynthesis [10].

3.3 Basic Sound Node

To construct a basic sound node in COLLADA, we first list a few essential audio properties that must be taken into account, such as the playback frequency and the volume level. Allowing for future maintenance and extension, it is our intention to study and follow COLLADA's present schema of graphic elements and to keep the overall structure coherent.

The first element we would like to add into is <sound>, which will specify a sound source along with a set of its properties. In COLLADA, the nodes named <library_*s> are the libraries containing pieces of elements <*>s which can be later referred to and used in other elements. Here we have the node <library_sounds> that declares a module of <sound> elements.

Table 1. Attributes of the <sound> element

id	xs:ID	A text string containing the unique identifier of this element. This value must be unique within the instance document. Optional.
sid	sid_type	A text string value containing the scoped identifier of this element. This value must be unique within the scope of the parent element. Optional.
name	xs:token	The text string name of this element. Optional.
frequency	float	A float value that indicates the playback frequency for the sound in Hz. Optional.
volume	Float	A float value that indicates the volume for the sound. From 0.0 (silent) to 1.0 (full volume, default).

Here is an example of a <sound> element that refers to an external WAV asset.

```
<library_sounds>
  <sound id="CarEngine_1-sound" volume="0.8">
    <init_from>
      <ref>./sounds/CarEngine_1.wav</ref>
    </init_from>
  </sound>
</library_sounds>
```

3.4 DSP

A number of basic DSP effects, such as parametric equalizer, high-pass and low-pass filters, and reverberation, are being applied to help make sound compelling in virtual worlds. According to the regulation of organizing data sets that we introduced in advance, customized <dsp> elements will be stored in the container <library_dsps>. Categorized DSP effects with their specific parameters are then scripted as the child elements of a <dsp> node.The following is an example of a <dsp> element with its child elements. <library_dsps> provides a library in which to place <dsp> elements.

```
<library_dsps>
  <dsp id="default-dsp">
    <parametric_eq count="6">
      <enabled>true true false true false true</enabled>
      <q>80.0 160.0 500.0 2000.0 5000.0 12000.0</q>
    </parametric_eq>
    <lowpass_filter>
      …
    </lowpass_filter>
    …
  </dsp>
</library_dsps>
```

3.5 Streaming Sounds

A <sound> is embedded or external referenced file data. It declares the storage for the acoustical representation of an object. A <stream>, on the other hand, is a generalization of <sound> that can link multiple <sound> resources into a single object. It states the positioning of the sound sources and their perspectives in the auditory space.

The following is a <stream> node that holds a single car engine sound. A car dsp with pre-defined parameters is employed.

```
<stream id="car_1-stream" type="3D" pov="FOREGROUND">
  <loop>0</loop>
  <instance_sound url="CarEngine_1-sound"/>
  <instance_dsp url="#car-dsp"/>
</stream>
```

One can also create a playlist of sounds in a <stream> element with random or sequence playback. The following shows an example of a <stream> node where three sounds are randomly chosen from ten car engine sounds. The three sounds output a stream in 3d and will loop forever.

```
<stream id="car2-stream" type="3D" pov="FOREGROUND">
  <loop>-1</loop>
  <create_list count="3" shuffle="true">
    <instance_sound url="CarEngine_0-sound/>
    …
    <instance_sound url="CarEngine_9-sound/>
  </create_list>
  <instance_dsp url="#car-dsp"/>
  <group target="individual_vehicles-group"/>
</stream>
```

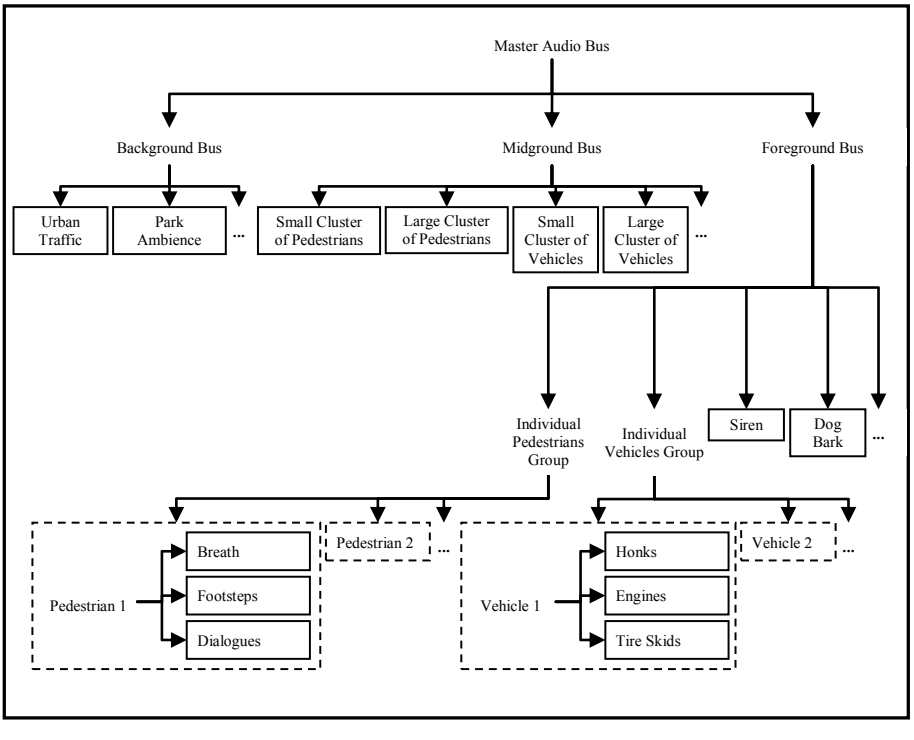

Fig. 2. An example of COLLADA audio scene hierarchy

3.6 Soundscape Hierarchy

In our audio scene hierarchy in COLLADA, sounds are divided into background, midground and foreground, based on Schafer's soundscape theory.

We have an ambient surround background that is not associated with a visible object and is not localized. A midground sound is stereo, linked to a clustered group of motion objects and situated in the barycenter of the group. Each foreground sound is mono and conventionally scripted within a single COLLADA dae file with the character or object it belongs to. In this way, the sound position will be simultaneously updated in real time.

Here is an example of how a stream can be put into an audio scene.

```
<library_visual_scenes>
  <visual_scene id="DefaultVisualScene">
    <node id="car2" name="car2" type="NODE"/>
  </visual_scene>
</library_visual_scenes>
<library_audio_scenes>
  <audio_scene id="DefaultAudioScene" name=”Vehicle”>
    <node id=”car2” name=”car2”>
      <instance_stream url=”#car2-stream” target=”#car2”/>
    </node>
  </audio_scene>
</library_audio_scenes>
<scene>
  <instance_audio_scene url=”#DefaultAudioScene”/>
  <instance_visual_scene url="#DefaultVisualScene"/>
</scene>
```

According to the principle of creating dynamically clustered objects in the visual part of COLLADA (dynamic LOD), we have to separate the description of the localization of midground sounds from the description of the visual cluster. Therefore, we create a COLLADA document which is purely constructed by acoustic ingredients and an overall sketch of the aural scene in the game environment. Another element type declared in this file is <group_streams>, which defines a group that shares the same properties (format, DSP to be applied…). By indicating the target attribute in its <group> child element, a <stream> becomes a part of a certain grouped stream. Notice that the derived streams of a <group_streams> must be in the same perspective as its mother group.

The following demonstrates how to assign the stream “car2-stream” to the group “vehicles-group:”

```
<stream id=”car2-stream” type=”3D” pov=”FOREGROUND”>
  …
  <group target=”./main_aduio.dae/vehicles-group”/>
</stream>
```

4 Conclusion and Future Works

This paper demonstrates achievement of enhancing COLLADA standard with sound representations. With the fundamental basis of audio scene structured in COLLADA, we begin to build an editor tool for sound designers' convenience to easily add aural properties into COLLADA documents, as well as a parser tool for program engineers to retrieve information of the sound scenery from dae files. Our future work follows three directions:

We will define a coherent and complete proposal of sound descriptions in COLLADA and propose it as a draft to the Khronos standardization group. We are continually working to understand the needs and to experiment with sound features in different uses of virtual cities. For instance, adaptive audio in virtual cities is not merely a continuously streaming content. It, cued by time or movements, shall be able to travel among different perspectives from background via midground to foreground and vice versa. Motion objects can be transferred into another state by an event trigger; so does sound [23]. At last, we are implementing these features of our COLLADA proposal under a variety of rendering engines (e.g. Unity, OGRE, Panda3D, etc) with a choice of audio middleware (e.g. Fmod, Wwsise, OpenAL, etc).

References

1. Farnell, A.J.: Sound synthesis for games. Sunderland University, Sounding Out 3 (2006)
2. Tsingos, N., Gallo, E., Drettakis, G.: Breaking the 64 spatialized sources barrier. Gamasutra.com (2003)
3. Morgan, S.: Dynamic game audio ambience: bringing Prototype's New York City to life. Gamasutra.com (2008)
4. Topol, A., Schaeffer, F.: Enhancing sound description in VRML. In: ICMA ICMC (2001)
5. Scheirer, E.D.: The MPEG-4 structured audio standard. In: Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing, pp. 3801–3804 (1998)
6. Scheirer, E.D., Vnnen, R., Houpaniemi, V.: AudioBIFS: Describing audio scenes with the MPEG-4 multimedia standard. IEEE Transactions on Multimedia 1(3), 237–250 (1999)
7. Arnaud, R., Barnes, M.C.: COLLADA: Sailing the Gulf of 3D Digital Content Creation. A K Peters Ed. (2006)
8. COLLADA – Digital Asset Schema Release 1.4.1 Specification, 2nd edn. (2008)
9. COLLADA – Digital Asset Schema Release 1.5.0 Specification (2008)
10. Tsingos, N., Gallo, E., Drettakis, G.: Perceptual sound rendering of complex virtual environments. ACM SIGGRAPH 2004 Papers 23(3), 249–258 (2004)
11. Shao, B., Velazquez, L.M., Scaringella, N., Singh, N., Mattavelli, M.: SMIL to MPEG-4 BIFS Conversion. In: AXMEDIS 2006, pp. 77–84 (2006)
12. Murray Schafer, R.: The Tuning of the World, The Soundscape (1977) republished as The Soundscape (1994) ISBN 978-0892814558
13. Weis, E., Belton, J.: Film Sound: theory and practice. Columbia University Press (1985)
14. Truax, B.: Soundscape Composition as Global Music: Electroacoustic Music as Soundscape. Organised Sound 13(2), 103–109 (2008)
15. Truax, B.: The Analysis of Electroacoustic Music as Soundscape. Electroacoustic Music Studies 2007. De Montfort University (2007)

16. Lavandier, C., Cance, C., Dubois, D.: L'évaluation de la qualité des environnements sonores: perspectives actuelles. In: Proceedings, 10ème Congrès Français D'acoustique (2010)
17. Truax, B.: Acoustic Communication. Ablex Publishing, New Jersey (1984)
18. Raimbault, M., Dubois, D.: Urban soundscapes: experiences and knowledge. Cities 22(5) (2005)
19. Nordahl, R.: Evaluating environmental from a presence perspective for virtual reality applications. EURASIP Journal on Audio, Speech and Music Processing 2010 (2010)
20. Valle, A., Lombardo, V., Schirosa, M.: Simulating the Soundscape through an Analysis/Resynthesis Methodology. In: Ystad, S., Aramaki, M., Kronland-Martinet, R., Jensen, K. (eds.) CMMR/ICAD 2009. LNCS, vol. 5954, pp. 330–357. Springer, Heidelberg (2010)
21. Menzies, D.: Physically motivated environmental sounds synthesis for virtual worlds. EURASIP Journal on Audio, Speech, and Music Processing 2010 (2010)
22. Verron, C., Aramaki, M., Kronland-Martinet, R., Pallone, G.: A 3D immersive synthesizer for environmental sounds. IEEE Transactions on Audio, Speech, and Language Processing 18(6), 1550–1561 (2010)
23. Collins, K.: Game Sound: An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design. MIT Press, Cambridge (2008)