Web3D – 16th International Conference on 3D Web Technology, Paris, France, June 20–22, 2011

Spatial Audio–Graphic Modeling for X3D

Hui Ding* LIMSI–CNRS–UPS
Diemo Schwarz† IRCAM–CNRS–UPMC
Christian Jacquemin‡ LIMSI–CNRS–UPS
Roland Cahen§ ENSCI–Les Ateliers

Abstract

In audio–graphic scenes, visual and audio modalities are synchronized in time and space, and their behaviour is determined by a common process. We present here a novel way of modeling audio–graphic content for interactive 3D scenes with the concept of sound processes and their activation through 2D or 3D profiles. Many 3D applications today support both graphical and audio effects to provide a more realistic user experience; however a common model and interchange format for interactive audio–graphic scenes is still lacking. X3D is one of the most promising formats for 3D scene representation. It is extensible and supports simple spatial audio representation and almost all basic and advanced 3D computer graphics techniques. We therefore propose an extension of the X3D standard to represent the sound process and activation profile model for providing a rich audio–graphic description in X3D.

Keywords: X3D, audio–graphic modeling, sound process

1 Introduction

As a very promising format for representing 3D computer graphics, X3D supports almost all the common 2D/3D graphic techniques. Besides, X3D is a functionally comprehensive format and also supports spatialized audio and video, specifically mapping the audio-visual sources onto geometry in the scene. However, the spatialized sound in X3D is in general not refined and exhausted enough to represent an interactive audio-graphic scene. For example, it is not capable of describing the variable sound of leaves through wind due to sophisticated sound activation. Therefore, we extend X3D by introducing a novel method of audio-graphic modeling.

Most prior 3D scene description languages have mainly focused on visualization and auralization separately. [Funkhouser et al. 1999] and [Tsingos et al. 2001] have computed the sound propagation paths by simulating them as wave phenomena. [O'Brien et al. 2002] and [Bonneel et al. 2008] have focused on sound synthesis from different physical motions. Meanwhile, [Tsingos et al. 2004] and [Moeck et al. 2007] have investigated the computational limit problem by using auditory culling and clustering technologies to reduce the complexity of audio rendering. However, these works did not address the issue of the description and activation of sound process in 3D graphical environments. By considering together audio and graphic cues to 3D virtual environments, we present new concepts for characterizing the principle of representing such a sound process.

To sum up, we incorporate the concepts of sound processes and their activation for modeling audio–graphic scenes into X3D by extending the conventional schema in the private-extension schema. We think that our audio-graphic modeling method is suitable to be expressed in X3D, and with X3D support, our method could become more standardized.

The rest of the paper is organized as follows. Section 2 presents the principle of 3D audio-graphic scene description, the definition of the sound process and its specification in X3D. Section 3 describes how to place the sound source in the 3D scene with respect to X3D. Section 4 presents the activation profile and its specification in X3D. Finally, Section 5 concludes the paper.

2 Audio–Graphic Modeling

Graphic and sounding objects are audio–graphic when visual and audio modalities are synchronized in time and space and when they share a common process. In an audio–graphic scene, the graphics information are often visualized to viewer, and the audio information are relatively invisible and perceived only when activated. Different from work on acoustics [Tsingos et al. 2001] or sound synthesis [O'Brien et al. 2002], our work focus on how to place the sound sources in a 3D scene and their mapping to a sound process which can be activated by corresponding sound activation profiles according to different audio–graphic content.

Our principle of sound representation is that each sound *effector* is situated at a single point of a geometry object in the 3D scene (see section 3.1). The produced sound might occupy a larger space, but this *hotpoint* is where the sound processes activation level is measured. Each sound point is linked to a *sound process* (section 2.1). The *activation* of a sound process and other parameters are determined by *profiles* or *maps* that move in the scene (section 4).

For the example of sound made by leaves in the wind, we assume that every leaf is a sound point which is linked to a sound process, and the wind can be depicted as a kind of profile. When the wind profile moves through the leaves, the sound processes will be activated. In the following sections, we specify sound process, sound mapping, activation profile and their representation in X3D. We have demonstrated our method through this audio-graphic example scene in Unity3D[1].

2.1 Sound Process Definition

The sound process, located at one or many points in a group, is defined by the following general parameters:

identifier
A string identifying this instance of a sound process.

model
A string refering to the class of the sound process running this instance.

activations
A vector of numbers between 0 and 1 giving the activation levels of the process (i.e. possibly weights for several presets determining the parameters of the process).

This could be interpreted as a generic parameter vector, also.

The concrete parameters are specific to individual sound process models, e.g. *impact material* for rain sound process models, *car speed* for traffic sound process models.

Note that the volume of the sound process is tuned by its individual sound placement in section 3.1.

---

*e-mail: hui.ding@limsi.fr
†e-mail: schwarz@ircam.fr
‡e-mail: christian.jacquemin@limsi.fr
§e-mail: cahen@ensci.fr

[1] Unity3D is an integrated development environment for creating interactive 3D content like 3D video games. http://unity3d.com/

2.2 X3D Representation of a Sound Process

In X3D, the AudioClip and the MovieTexture are so far the only possible sound sources (the former playing samples, MP3, or MIDI files).

We propose to extend the formalism by a new subclass of X3DSoundSourceNode that we call TPSoundProcess, that can have the attributes corresponding to the sound process parameters listed above.

The new object hierarchy is depicted in figure 1.

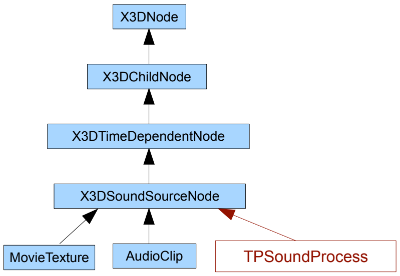

Figure 1: *The proposed element TPSoundProcess extending X3DSoundSourceNode.*

Its specification of the XML encoding is:

```
<TPSoundProcess
 DEF=""                        ID
 USE=""                        IDREF
 description=""                SFString  [inputOutput]
 loop="false"                  SFBool    [inputOutput]
 pauseTime="0"                 SFTime    [inputOutput]
 pitch="1.0"                   SFFloat   [inputOutput]
 resumeTime="0"                SFTime    [inputOutput]
 startTime="0"                 SFTime    [inputOutput]
 stopTime="0"                  SFTime    [inputOutput]
 url=""                        MFString  [inputOutput]
 model=""                      SFString  [inputOutput]
 activation=""                 SFString  [inputOutput]
 containerField="children"     NMTOKEN
/>
```

Compared to Sound, TPSoundProcess contains *model* and *activation* as new attributes. The attribute DEF can be used as the parameter "*identifier*".

3 Sound Source

One sound source should be located at a single point in a local coordinate system so as to usually attach itself to a 3D primitive or object. One or more sound sources should be linked to a sound process in order to emit sound when the sound process is activated by a certain profile. We present in the following subsections how to a map sound process to sources and its representation in X3D.

3.1 Sound Placement

We place sound processes (see section 2.1) at single points in the 3D scene, which correspond to the *effector*, e.g. the tree leaf that is activated, or the impact point of a raindrop.

The produced sound might occupy a larger space, as determined by the directivity information (see figure 2), but this *hotpoint* is where the sound processes activation level is measured (see section 4). Each sound point is assigned to a *sound process* instance (see section 2.1) that runs a certain *sound process* model.

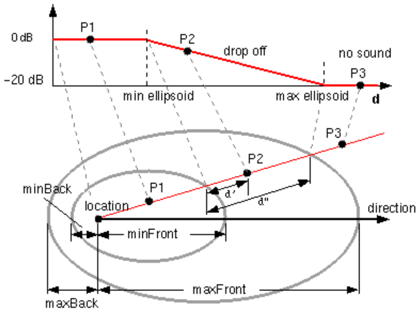

Figure 2: *X3D Sound Node Geometry (from [The Web3D Consortium 2008])*

3.2 X3D Representation of a Sound Source

In X3D, the Sound subclass of the abstract X3DSoundNode class represents the placement of a sound source at a point location and its directivity and distance-based attenuation. It was supposed to be the proper node which holds a link to our extension of X3DSoundSourceNode, which is the sound process (see section 2.1). However, since it contains only AudioClips or MovieTexture for sound playback[2], we need to create a new element named "TPSound" based on Sound (see figure 3) in order to contain our TPSoundProcess.

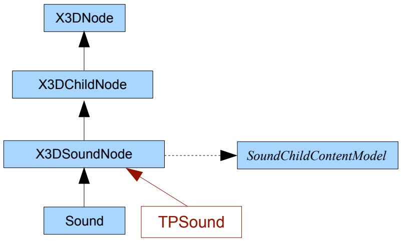

Figure 3: *The proposed element TPSound extending X3DSoundNode.*

Based on Sound, TPSound could be specified for the XML encoding as follows:

```
<TPSound
 DEF=""                        ID
 USE=""                        IDREF
 direction="0 0 1"             SFVec3f   [inputOutput]
 intensity="1"                 SFFloat   [inputOutput]
 location="0 0 0"              SFVec3f   [inputOutput]
 maxBack="10"                  SFFloat   [inputOutput]
 maxFront="10"                 SFFloat   [inputOutput]
 minBack="1"                   SFFloat   [inputOutput]
 minFront="1"                  SFFloat   [inputOutput]
 priority="0"                  SFFloat   [inputOutput]
 spatialize="true"             SFBool    [initializeOnly]
 containerField="children"     NMTOKEN
 >
 <TPSoundProcess />
</TPSound>
```

Grouping of sound points to be serviced by a single sound process is achieved by assigning the same sound process instance to several sound points.

4 Activation Profile Representation

The activation and in fact any parameter of the sound process are either determined by a global setting (a scene parameter, determined

---

[2] See X3D encoding documentation http://www.web3d.org/x3d/specifications/ISO-IEC-19776-1.2-X3DEncodings-XML

by the user or dependent on some script), or they are given by an *activation map* or *profile* [Schwarz et al. 2011]. A profile is a 2D or 3D scalar field the value of which can be looked up by position. This profile determines the activation value at each sound source point location. The profile is attached to a (possibly invisible) 2D or 3D object that can move through the scene in order to control the sound processes, e.g. wind in trees, a hand in leaves, or excitation of a crowd.

One possibility for defining the profiles detailed in the following is that of *parametric profiles* (section 4.1), where the distance to a reference point is passed through a mapping function. An alternative, not treated in this article, are *mesh-based profiles* [Freed et al. 2010], where each vertex in a point set carries an activation value, and we interpolate inside the triangular mesh.

4.1 Parametric Profiles

A parametric profile is given by a function that can be evaluated at a point in the scene. For instance, a gaussian-shaped profile has as parameters the centre, width, and curve (standard deviation, kurtosis).The function, that defines the shape of the profile, is then extended to 2D or 3D by different means detailed in section 4.1.3: revolution, inclusion in a geometric support object which defines the extent of the profile, or extrusion.

This architecture allows to compose a great variety of profile shapes from only a few primitives, and by specifying only a hand full of parameters.

4.1.1 Profile Functions

The different 1D parametric profile functions depicted in figure 4 are defined by distance to a reference point, and carry 1–4 parameters. They all have as common parameter an inversion flag. The list of functions and parameters is:

linear mindist, maxdist
A linear segment rising from zero to one between mindist, maxdist. Beyond these, the linear function is constant at zero and one, respectively.

delta mindist, maxdist, middle, width
trapezoidal, or cut cone shape, for fade-in/fade-out of activation

exscale mindist, maxdist, base
Exponential curve between mindist, maxdist. Here, the *base* parameter determines the curvature of the function.

$$y = \frac{e^{\left(\log base \frac{x-mindist}{maxdist-mindist}\right)-1}}{base-1} \tag{1}$$

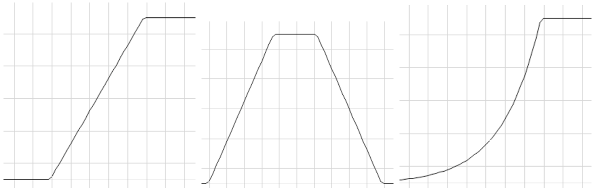

Figure 4: *the linear, delta, and exscale parametric profile function shapes.*

4.1.2 x3d representation of profile functions

we introduce a new complextype tpprofilefunction that can be inherited by x3dnode, and this tpprofilefunction will derive different concrete elements such as tpprofilefunctionlinear, tpprofilefunctiondelta, and tpprofilefunctionexscale which define the profile function. (see figure 5)

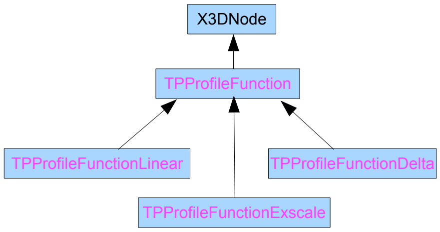

Figure 5: *profile function class hierarchy.*

4.1.3 profiles in 2d

any of the 1d parametric profile functions are extended to define an activation profile in 2d (that is usually parallel to the ground) in several ways:

revolution: by revolution of the profile around the vertical y-axis at a reference origin point o, either on

infinite support: purely based on the distance to o, or on

finite support: i.e. the distance to o is rescaled to the distance r to the intersection point of the ray from o through the query point with the boundary of the support. then, maxdist defines the relative boundary, i.e. $scale = r/maxdist$. the finite support can be one of:

- circular support, possibly transformed (ellipse)
- polygonal support, either given by a list of points, or by the convex hull of a point set

extrusion: linear extension of one 1D function along a line, resulting in a rectangular or parallelogram-shaped 2D profile

interpolation: Two or more 1D functions are placed in parallel in a rectangle and extruded while interpolating between neighbouring functions.

For easier authoring of 3D scenes, we can visualise the 2D profiles in 2D either by colour or transparency, or we can generate a 3D visualisation object. Here, each point of the mesh of the visualising object is projected to the ground plane and then samples the profile, affecting the profile value times a max height to the y coordinate. Figure 6 shows such a visualisation for three revolution profiles.

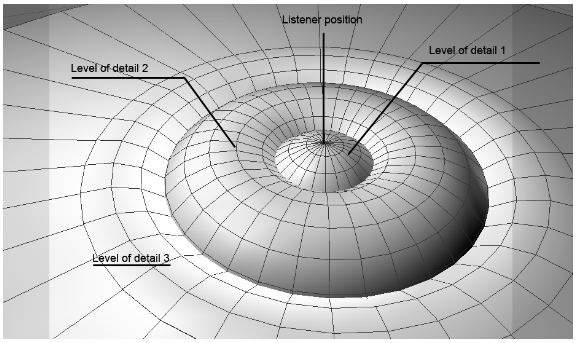

Figure 6: *3D visualisation of three revolution profiles, here placed around the listener position for control of level of detail.*

4.1.4 Profiles in 3D

The extension of the 1D parametric profile functions to 3D is similar to the 2D extension:

- **revolution:** spherical revolution of the function around an origin *o* in order to create a (distance based) spherical profile, again either on
  - **infinite support:** resulting in a transformed sphere (ovoid), or on
  - **finite support:** deformed by a (convex) mesh, similar to the inscription into a 2D boundary in section 4.1.3.
- **extrusion:** The 1D profile function is extruded along a plane to create a parallelepiped.
- **interpolation:** Two or more planes of 2D profiles are interpolated along a line.

4.1.5 X3D Representation of Parametric Profiles in 2D or 3D

The representation needs to link a profile node to a profile function, and a support geometric object (ellipse, polygon, cylinder, sphere, mesh), and possibly a visualisation geometric object (a mesh sampling the profile, a 2D or 3D colour texture).

A profile is attached to a geometrical object which stands for a container of the profile. This container could be 2D or 3D, and can move around in our 3D scene. In our extension X3D schema (figure 7), we add a TPGeometryProfileContainer element that will contain two child nodes corresponding respectively to X3DGeometryNode and TPProfileFunction. The group ShapeChildContentModel allows the container to be any existing 2D or 3D geometry node or to create a new geometry instance by IndexedFaceSet for example.

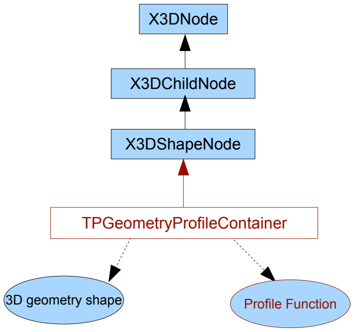

Figure 7: *TPGeometryProfileContainer class hierarchy*

The TPProfileFunctionModel in TPGeometryProfileContainer works as the SoundChildContentModel in Sound, so that TPProfileFunctionModel contains all the models of functions of profile which are classified by dimension (TPProfileFunctionModel1D, TPProfileFunctionModel2D, TPProfileFunctionModel3D). Every model contains different profile functions (for example, TPProfileFunctionModel1D contains TPProfileFunctionLinear, TPProfileFunctionDelta, TPProfileFunctionExscale).

The TPGeometryProfileContainer could be specified for the XML encoding as follows:

```
<TPGeometryProfileContainer
 DEF=""                       ID
 USE=""                       IDREF
 bboxCenter="0 0 0"           SFVec3f    [initializeOnly]
 bboxSize="-1 -1 -1"          SFVec3f    [initializeOnly]
 containerField="children"    NMTOKEN
 >
 <ShapeChildContentModel />
 <TPProfileFunctionModel />
</TPGeometryProfileContainer>
```

5 Discussion and Conclusion

Our work shows a model for audio–graphic scene description which uses the concept of sound process and its representation. To standadize our model and port it to different platforms, we would like to incorporate it into the X3D format. Compared to the traditional sound elements in X3D, our representation of sonic objects requires a much more complex information-exchange between audio and graphical processes at the rendering level. However, X3D does not directly support complex 3D computer graphics computations, thus these computations must be carried out by the scripting engine. This will be an obstacle for the X3D audio–graphic scene designer who is neither expert in computer graphics, nor in scripting languages.

In the near future, we will continue to explore further profile activation for different sound process in order to propose a complete extension schema of X3D.

Acknowledgements

The work presented here is funded by the *Agence Nationale de la Recherche* within the project *Topophonie*[3], ANR-09-CORD-022. We thank the project partners for the common work and discussions from which this paper is the fruit.

References

BONNEEL, N., DRETTAKIS, G., TSINGOS, N., VIAUD-DELMON, I., AND JAMES, D. 2008. Fast modal sounds with scalable frequency-domain synthesis. *ACM Transactions on Graphics (TOG) 27*, 3, 24.

FREED, A., MACCALLUM, J., SCHMEDER, A., AND WESSEL, D. 2010. Visualizations and Interaction Strategies for Hybridization Interfaces. In *Proceedings of the International Conference for New Instruments for Musical Expression NIME*, 343–347.

FUNKHOUSER, T. A., MIN, P., AND CARLBOM, I. 1999. Real-time acoustic modeling for distributed virtual environments. In *Proceedings of SIGGRAPH 99*, 365–374.

MOECK, T., BONNEEL, N., TSINGOS, N., DRETTAKIS, G., VIAUD-DELMON, I., AND ALOZA, D. 2007. Progressive perceptual audio rendering of complex scenes. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM SIGGRAPH.

O'BRIEN, J. F., SHEN, C., AND GATCHALIAN, C. M. 2002. Synthesizing sounds from rigid-body simulations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM, New York, NY, USA, SCA '02, 175–181.

SCHWARZ, D., CAHEN, R., HUI, D., AND JACQUEMIN, C. 2011. Sound level of detail in interactive audiographic 3D scenes. In *Proceedings of the International Computer Music Conference (ICMC)*.

THE WEB3D CONSORTIUM, 2008. X3D specification – part 1: Architecture and base components. http://www.web3d.org/x3d/specifications. Edition 2, ISO/IEC 19775-1.2:2008.

TSINGOS, N., FUNKHOUSER, T., NGAN, A., AND CARLBOM, I. 2001. Modeling acoustics in virtual environments using the uniform theory of diffraction. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH '01, 545–552.

TSINGOS, N., GALLO, E., AND DRETTAKIS, G. 2004. Perceptual audio rendering of complex virtual environments. In *ACM SIGGRAPH 2004 Papers*, ACM, 249–258.

---

[3] http://www.topophonie.fr