Sound Level of Detail in Interactive Audiographic 3D Scenes

Diemo Schwarz, Roland Cahen, François Brument, Hui Ding, Christian Jacquemin

Proceedings of the International Computer Music Conference (ICMC) July 31–August 5, 2011, Huddersfield, UK

SOUND LEVEL OF DETAIL IN INTERACTIVE AUDIOGRAPHIC 3D SCENES

Diemo Schwarz — UMR STMS Ircam–CNRS–UPMC

Roland Cahen, François Brument — ENSCI–Les Ateliers Paris

Hui Ding, Christian Jacquemin — LIMSI CNRS and University Paris 11 Orsay

ABSTRACT

This paper shows methods for defining and rendering Sound Level of Detail (SLOD) for audiographic scenes using corpus-based granular synthesis. We introduce three levels of detail for sound (individual events, statistical texture, background din), to define a proximity profile around the listener. The smooth transition between the levels is assured alternatively by statistical modeling or audio impostors. The activation of the three levels is controlled by invisible and editable profile objects mapped to presets of audio process parameters. These also serve to balance the CPU load between the different audio processes. We have tested this method on various virtual scenes such as crowds, rain, foliage, or traffic.

1. INTRODUCTION

Topophonies are virtual navigable sound spaces, composed of audiographic objects. Graphic and sounding objects are audiographic when visual and audio modalities are synchronized in time and space and when they share a common process. Our goal is to navigate in large numbers of audiographic objects that are non-homogeneous in space and in time. Such clusters of objects can be used to describe a crowd, a flow of traffic, foliage, or rain.

We define a generic model of data for the efficient and fine definition of audiographic clusters of objects, as well as the smooth and interactive rendering of moving objects with credible sound behaviours, with reduced computational cost.

Most audiovisual real-time applications separate graphics and audio data, and exchange commands, in order to trigger sounds from a graphic event or to produce a visual event according to a sound effect. Our approach is to try to bring together as close as possible sound and image, in the idea of an audiographic object, as a single entity, and make it respond visually and sonically to stimuli, used as a common cause for its sound and visual effects.

The purpose of a bi-modal representation framework is to optimize rendering by restricting the data flow between the audio and the graphic engine, and by sharing as much of the computation as possible between the two modalities. We also believe that a proper scene design can only be performed in a model that encompasses both audio and visual data. The scene description should contain a complete definition of the audiographic affordance.

LOD is nowadays common in real-time Computer Graphics and its dynamic GPU-based computation has been facilitated by the introduction of Geometry Shaders. It is used in many applications such as urban or natural landscape rendering, crowd simulation, or scientific visualization, but usage of Sound Level of Detail (SLOD) is still rare. Yet it would allow to improve sound design and rendering. SLOD can partly derivate from the graphic models of LOD [5].

The issues for defining and implementing SLOD are threefold: First, to capture and smoothly continue behaviour, when the sound-generating process is no longer available. Second, to reduce computational load of sound processing, and third, to reduce communication bandwidth between the graphic and audio part of the model.

SLOD means varying sound resolution according to the required perceived precision. For example we do not need to render the sound of each character of a crowd when we are far away and cannot distinguish individual events happening among the crowd. Therefore we may replace each character sound by a global ambient sound called *impostor*.

However, sound has many particularities that image does not have: Salient sound streams and events do not follow the same rules as image and visual events' saliency and are not only related to proximity [2]. These differences are due in a large part to different perceptual properties between hearing and vision, resulting in different simplification schemes such as masking, culling, or perceptibility. This last point is not developed in this paper. Hereafter, we will only consider proximity to the listener as criterion for SLOD.

We have chosen to work with corpus-based concatenative synthesis (CBCS) [6], which can be seen as a content-based extension of granular synthesis based on audio descriptor analysis. Granular synthesis is rarely used in real-time 3D, because it is more difficult to control than sampling and has a higher processing cost, but its gives the possibility to use recorded sounds and to control in real-time many transformations, such as pitch, duration, smoothing, and timbre in order to produce variations. We think that this method can be applied to produce good statistical sound textures [7] as well as traditional sampling; moreover, audio descriptors can be used as high level parameters to control the sound character. Note that the realistic spatial rendering of sound is not the topic of this article.

Our model is implemented in UNITY3D and MAX/MSP for a certain number of example scenes.

2. PREVIOUS AND RELATED WORK

In 3D graphics, LOD encompasses various optimization techniques for 3D object rendering that are used to de-

crease its complexity and increase its efficiency. The trade-off between quality and performance determines that a good LOD technique should increase the rendering speed by representing less object details without the user noticing the resulting changes. LODs are generally parameterized by controlling the ratio by which the complexity of the object is decreased. The most usual control is distance: the complexity of a 3D object is decreased as it moves away from the viewer and becomes smaller in the image space. Other metrics such as priority, size, hysteresis, or velocity can also be taken into consideration for LOD control. The generation of different detail of a 3D object generally consists in mesh simplification techniques. Hoppe proposed the edge-collapse operator for mesh simplification [4] and progressive meshes for continuous LOD [3]. One of the simplest LOD techniques is called *impostor*, and is used to replace a mesh-based rendering by a simple mapping of an image onto an flat polygon. Whatever LOD technique is chosen, human perception has to be taken into consideration since the user is the ultimate judge for the quality of the resulting image. User-based evaluation is a necessary companion activity in LOD algorithm design so that the increase in performance does not come at the cost of image quality loss.

Against this large body of work of LOD in graphics stand only few works in audio [5], mainly concerned with balancing the computational load in model-based synthesis for a restricted number of interactions (impact, friction, rolling) [1], notably reducing the number of resonance modes in modal synthesis of impact sounds. Our interest in SLOD is the dynamic structuring of sound sources in interactive 3D scenes. A virtuous side-effect of bi-modal LOD studies should be to orient the research on graphical LODs towards rapidly changing scenes. Currently, graphical LODs are mostly used for large data sets (whether meshes or point sets) that do not vary much over time. By focusing on sound that is by nature temporal and highly influenced by direct or indirect interactions such as wind on foliage, we also wish to highlight the interest of deeper studies on graphical LODs for rapidly evolving graphical objects such as a moving crowd, foliage, or rain.

Descriptor-based interactive sound texture synthesis using corpus-based concatenative synthesis techniques have been introduced in previous work by the authors [7].

3. SOUND LEVEL OF DETAIL (SLOD)

We have chosen to use a simple design with three SLODs:

SLOD 1—foreground: individually driven sound events and sound behaviours: When we are very near to an audiographic cluster, for example rain drops on tree leaves, each drop collision should be heard and seen individually.

SLOD 2—middle ground: group-driven sound event, statistical behaviours: Above a certain density of events, when they can hardly be isolated anymore, they play stochastically, according to a sound behaviour preset. This limit is passed when sources are farther than a certain distance from the listener.

SLOD 3—background: sound impostors: Even further away, sources can be simply rendered by continuous audio impostors such as audio files, or take advantage of the scene depth partition or spatial clustering knowledge to dynamically mix groups of procedural impostors according to the view point and the evolution of the scenario.

3.1. Passing from SLOD1 to SLOD2

In order to change between levels 1 and 2, we have the two alternatives of statistical modeling and audio impostors, elaborated in the sections below.

Using statistical modeling, we do not lower the CPU resources of the sound-generating process, but we reduce the bandwidth of the communication from the graphics engine to the sound engine. The advantage of statistical modeling is that the model's parameters can be manipulated or interpolated, and that the storage requirements are minimal.

This is not true for audio impostors: their sound and density are frozen once recorded, and they take up more memory, however, they reduce CPU usage to a minimum of playback of longer grains with density close to one.

3.1.1. Statistical Modeling

In order to capture their statistical behaviour, the level 1 audio processes are modeled by recording the descriptors of the grains they produce, and the inter-event time, for a short duration (3 seconds proved sufficient in our test scenes). This can either happen in the scene editor on saving the scene, or in the player on loading the scene.

We use the log of the inter-event times in milliseconds, in order to achieve a finer time resolution for dense textures, sacrificing precision for sparser textures, where the perception of the fine differences in timing becomes less acute.

The recorded data is binned into histograms. Figure 1 shows histograms of inter-event times and two descriptors, sampled from a rain simulation driven by a particle generator in UNITY3D.

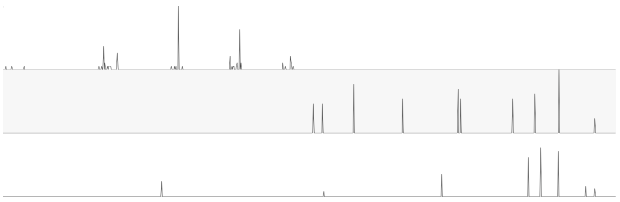

Figure 1. Histograms of log inter-event time (top), spectral centroid (middle) and loudness (bottom).

In order to reinterpret the models for resynthesis, we use the method known as *inverse transform sampling* [7], where these histograms H_i with bins B_i are interpreted as probability density functions (PDF) P_i, from which we calculate the cumulative sum to obtain the discrete cumulative density function (CDF) as:

$$C_i(n) = \sum_{k=1}^{n} H_i(k) \tag{1}$$

We then draw random bin indices accordingly by accessing the CDF by a uniformly distributed random value

$x_1 \in [0,1[$, and draw descriptor and timing values within the bin of width B_i^w using another uniformly distributed random value $x_2 \in [0,1[$:

$$\hat{p} = B_i(\lceil C_i(x_1) \rceil) + \frac{x_2}{B_i^w} \quad (2)$$

This process generates a stream of event times and target descriptor values that obeys the same distribution as the sampled sound process, in the limits of the discretisation of the histogram.

The resulting distributions can also be easily interpolated to generate a smooth evolution from one texture or state of the scene to the next.

The drawn target descriptors then serve to control a CBCS engine with a corpus of source sounds, as explained in section 5. We call this method *prodedural impostors*.

3.1.2. Audio Impostors

The second method of audio impostors simply records the output of the level 1 process, and splits it into average sized grains, so that grains can be shuffled around for playback, in order to avoid a repetitive loop.

3.2. Passing from SLOD2 to SLOD3

SLOD 3 combines the SLOD 2 sounds of many processes, clustered by angular proximity. Here, we record audio impostors mixed from the SLOD 2 output of the clustered sources.

An alternative is to include pre-recorded ambient textures as SLOD 3 source, in order to achieve highest realism for the background din. Here, we can make use of sound descriptors to match one of several possible background textures to the SLOD 2 sound.

4. CONTROL OF SLOD IN AN INTERACTIVE 3D SCENE BY PROXIMITY PROFILES

In order to control our static audio LOD, we propose a system of *proximity profiles* (see figure 2), or 2D *maps*, that control the passage from one LOD to another, by determining the "presence" of the process generating the audio for each LOD.

This "presence" is for the moment simply rendered by volume, resulting in a cross-fade between the sound generating processes per LOD.

4.1. The 4P Mapping Model

The interpretation of the above profiles is not direct (by a mapping from activation level to volume), but passes through a mapping stage for more flexibility and better integration with the other (timbral) parameters controlling the audio processes:

The *profiles* are mapped to *presets* that control the *parameters* of the audio *processes*, hence the name *4P model*. Figure 4 gives an overview of the model. In detail: A *profile* is a scalar field situated in the 3D scene. The scalar activation values are in the range $[0,1]$. A *preset* t_i is a list of m parameter values v_{ij} and mix weights w_{ij}

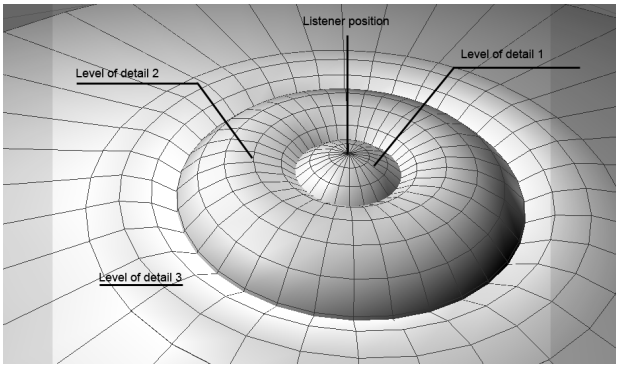

Figure 2. 3D rendering of proximity profiles.

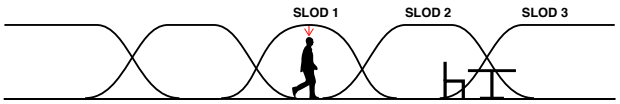

Figure 3. Cross-cut through proximity profiles.

for each parameter. A *preset mix* x is then a linear combination of the m parameter values of $i = 1..n$ presets t_i, gouverned by two sets of weights, the parameter weights w_{ij}, *given individually for each preset and each parameter*, and a mix factor per preset f_i, given by the profile's activation value. Each of the m resulting parameter values x_j is then given by

$$x_j = \sum_i f_i * w'_{ij} * v_{ij} \quad (3)$$

where w'_{ij} are the weights of parameter j normalised to sum to one, i.e. $\sum_i w_{ij} = 1$, except if the sum is 0. The mix factors f_i are usually given directly by the profile's activation value, and not normalised internally.

In the current application, the w_{ij} are either 0 or 1, allowing to exclude certain parameters of certain presets from the mix, but of course it could be generalised to any number between 0 and 1.

The 4P model in its most general form links a profile to any number of presets, possibly in several audio processes, specifying a type of behaviour for each mapping. The behaviours can be either *continuous*, where the parameters are updated continuously, or *trigger*, where a

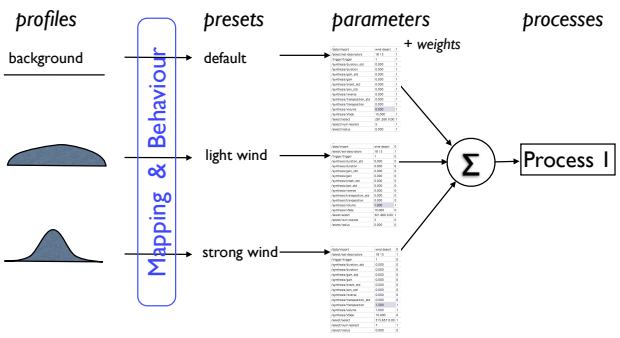

Figure 4. The 4P model: profiles controlling presets mapped to parameters of the sound processes .

sound event is generated synchronously with an activity change, e.g. from impacts of rain drops.

This model needs a "background" preset with all parameters set to default or neutral values and their weights non-null, and then mixes in only certain parameters from other presets.

The advantages of the 4P model are that it generalises mappings from profiles to 1 parameter (and thus subsumes the direct mapping case), to n parameters, or m sound characters (when different profiles are mixed).

The advantage of using 3D scene objects for the SLOD control, over simple calculation of activation by distance, is that we can then use the facilities of the 3D engine to detect collisions with the sounding objects in the scene and thus only need to evaluate the profiles when necessary.

5. PROTOTYPE IMPLEMENTATION

The prototype implementation is a combination of the scene in UNITY3D with custom scripts and classes generating and evaluating the activation profiles, communication via the OSC protocol with a sound engine running in MAX/MSP based on the CATART system[1] for corpus-based concatenative synthesis, making it possible to navigate through a two- or more-dimensional projection of the descriptor space of a sound corpus in real-time, effectively extending granular synthesis by content-based direct access to specific sound characteristics.

The statistical modeling, interpolation, and generation of probability distributions is conveniently handled by the modules ftm.inter, mnm.hist, mnm.probsampler, mnm.pdf from the MnM library included in FTM&CO[2].

The 4P model is implemented in the JAMOMA framework[3], which stores the presets and mappings in XML.

6. APPLICATIONS

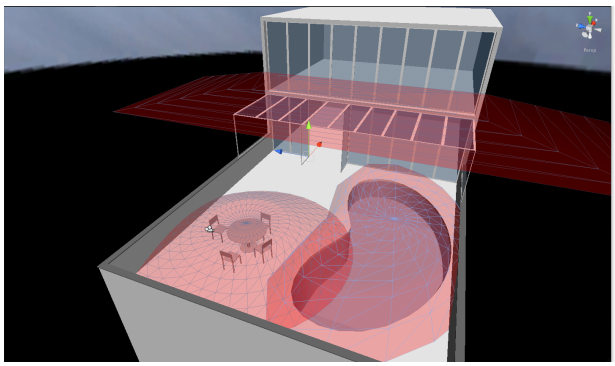

Figure 5. Screen shot of a rain scene with visualisation of activation profiles.

Figure 5 shows a scene modeled in UNITY3D, with activation profiles around the scene objects. These profiles give the activities of the level 1 processes of individual rain drop impacts, that fade out with distance to the centres. This sound is crossfaded with level 2 textures by inverse profiles not shown in the figure. Sound examples and videos can be found on the project web site `http://topophonie.fr`.

[1] `http://imtr.ircam.fr/index.php/CataRT`

[2] `http://ftm.ircam.fr`

[3] `http://jamoma.org/`

7. DISCUSSION AND FUTURE WORK

The combined advantages of our synthesis and SLOD model are twofold: First, using the same granular synthesis model for all 3 levels simplifies the consistent editing and control of the interactive scene audio content, assuring smooth transitions between levels of detail, and second, the 4P model for control of these transitions, as well as for other sound parameters, places the SLOD as objects in the scene that are easy to edit and visualise.

We are currently working on an extended saliency model based on geometrical and temporal density of the sources, user interaction and saliency weight. Evaluation of our model by subject tests is planned by comparing the generated L2 and L3 with environmental recordings.

Further connections with graphics LOD are to be studied, concerning sharing of calculation, e.g. of parameters, space subdivision or clustering, or masking.

8. ACKNOWLEDGEMENTS

The work presented here is partially funded by the *Agence Nationale de la Recherche* within the project *Topophonie*, ANR-09-CORD-022. We thank the project partners for the many fruitful discussions.

9. REFERENCES

[1] N. Bonneel, G. Drettakis, N. Tsingos, I. Viaud-Delmon, and D. James, "Fast modal sounds with scalable frequency-domain synthesis," *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3, p. 24, 2008.

[2] A. Bregman, *Auditory scene analysis: The perceptual organization of sound*. The MIT Press, 1994.

[3] H. Hoppe, "Progressive meshes," in *Computer Graphics and Interactive Techniques*. ACM, 1996.

[4] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Mesh optimization," in *Computer Graphics and Interactive Techniques*. ACM, 1993.

[5] T. Moeck *et al.*, "Progressive perceptual audio rendering of complex scenes," in *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. ACM SIGGRAPH, April 2007.

[6] D. Schwarz, "Corpus-based concatenative synthesis," *IEEE Signal Processing Magazine*, vol. 24, no. 2, pp. 92–104, Mar. 2007, special Section: Signal Processing for Sound Synthesis.

[7] D. Schwarz and N. Schnell, "Descriptor-based sound texture sampling," in *Sound and Music Computing (SMC)*, Barcelona, Spain, Juillet 2010, pp. 510–515.