SIXTH FRAMEWORK PROGRAMME
PRIORITY IST-2002-2.3.1.8
Networked Audiovisual Systems

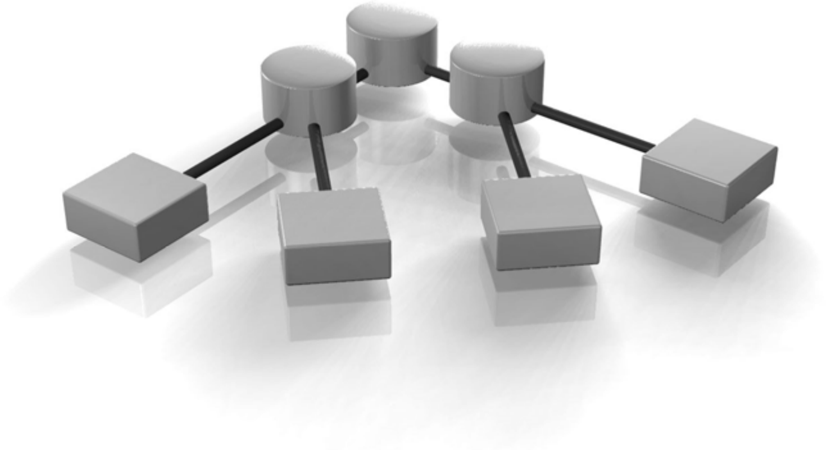
Uni-Verse Project

Deliverable D 7.5

Perceptual Modelling Module

April 29th 2007

Distribution: Public

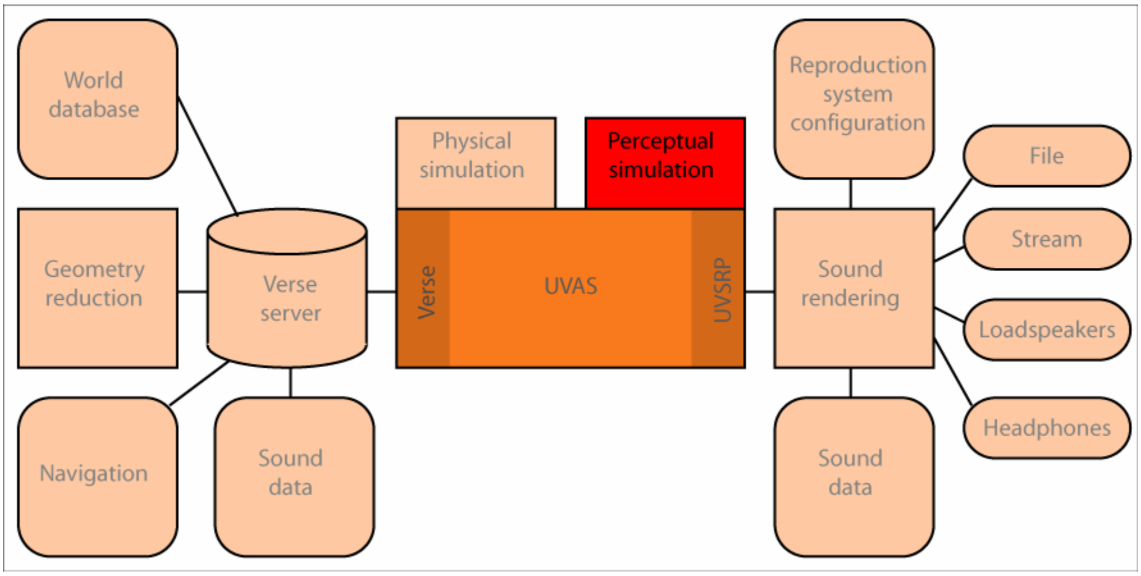


STREP project
Project acronym: Uni-Verse
Project full title: A Distributed Interactive Audio-Visual Virtual Reality System
Proposal/Contract no.: 002228

Content

1 Introduction

The subject of this report is the work of WP 7.5 Perceptual Simulation module which is an integrated part of the *Uni-Verse Acoustic Simulation* (UVAS) system (see Figur 1) and is implemented as module in the UVAS framework which is described in deliverable D 7.3. The purpose of the perceptual based acoustic simulation module is to complement the physical based simulation in UVAS. In game applications for instance the main interest of the sound designer is not to mimic the physical reality but rather to give the player an experience of the atmosphere of the acoustic space surrounding him which may not have anything in common with the geometry of that surrounding. In such case the designer would prefer a simulation based on a parametric description rather than on physical simulation. The designer will then be able to specify the acoustics of the surrounding using only a small set of parameters rather then the more cumbersome process of building a complete acoustic model of the space and in many cases it might not even be possible to build such model that will produce the result that the designer have in mind.



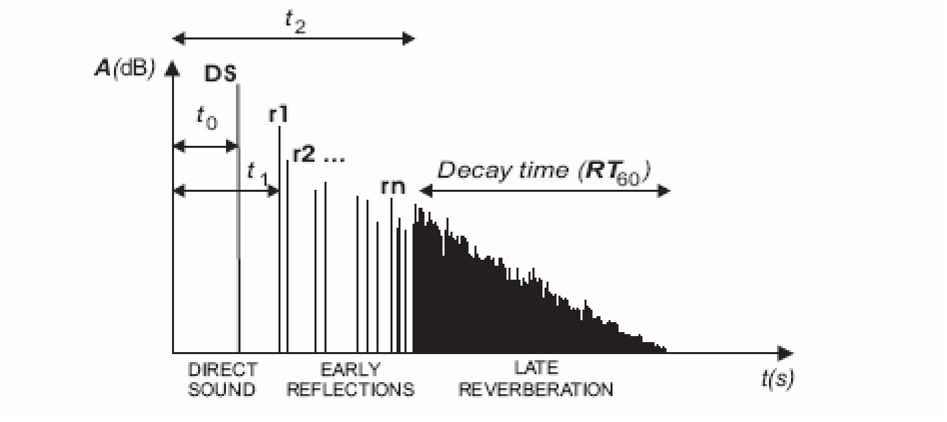
Figur 1, Uni-Verse Acoustic Simulation System

This module is based on a virtual room where the acoustic properties of the room can be specified using a small set of parameters. A virtual room can be placed anywhere in the simulated world and is very loosely coupled to the geometry of the simulated world. There can be many virtual rooms in the simulated world defined at different locations. The geometry of the world is only used to find which virtual room is associated with the position of the listener and to generate default parameter settings. The most important parameters of a virtual room is it volume, the mean acoustic absorption of the surrounding surfaces and the shape of the room.

When the simulation is started a model of the simulated world is loaded from a Verse server. This model contains the geometry of the world, a number of virtual room definitions, description of the sound sources and the listener. The module is then searching the loaded geometry to find the virtual room that is associated with the position of the listener. When a room is found its parameters are used to compute a new set of parameters for the Sound Rendering Module which is described in D 7.4. The calculated parameters together with information about the listener and the sound sources involved are then sent to the Sound Rendering Module which then produces the audible result.

2 Acoustic Model

The acoustics of a room is defined by the room impulse response (RIR), (see Figur 2). In a real-time acoustic simulation system it is not plausible to directly use the RIR by for instance computing the direct convolution of the RIR and the recorded sound of the sound sources. Other methods that are more effective and flexible are needed. One common solution is to divide the RIR into three distinct time-domain sections (see Figur 2), the direct sound (DS), the early reflections (r1…rn) which is about the first 100 ms of the response and the late reverberation which is the rest of the RIR **[Savioja99]**. The reverberation time (RT_{60}) is a standardized measure of the decay rate of the late reverberation **[Kutruff91]**.



Figur 2, Room Impulse Response (RIR)

Reflections in general are delayed and attenuated version of the original sound. The delay is caused by the distance the sound has to travel from the source to the listener and the attenuation is caused by absorption in the atmosphere and in the reflecting surfaces. In the early reflection part single reflections can be separated by the human perception and can from a perceptual point of view be viewed as separate sound events. The early reflections can be simulated using discrete delays and attenuation. The direct sound is simulated in a similar manner. In the late reverberation part the reflections are too dense so the human perception can not separate single reflections and they are perceived as a diffuse sound field. The late reverberation is simulated using statistical methods for instance the Schroeder reverb algorithm **[Schroeder62] [Moorer79]**.

2.1 Perceptual Parameterization

We need to find a parameterization of the RIR such as a small set of parameters describes the acoustics of a room in a form suitable for real-time auralization where the parameters are closely related to the perceived quality of a room acoustic response. Jot and Warusfel has developed a perceptual based acoustic simulation system **[Jot95]**. This system is based on studies of parametric representation of room acoustics by Jullien **[Jullien 95].** The model is closely related to the MPEG-4 version 2: Advanced AudioBIFS standard **[Väänänen04].** The representation is based on a set of 9 independent parameters:

1. Late reverberance is the decay time of the late reverberation, it corresponds to the standardized reverberation time RT_{60}.

2. Heaviness is a factor of the low-frequency (below 250 Hz) reverberation time.

3. Liveness is a factor of the high-frequency (above 4 kHz) reverberation time.

4. Room presence is the total energy level of the room response.

5. Envelopment is the ratio between early reflections and the direct sound.

6. Running reverberance is the perceived reverberation time while the sound is continuously played.

7. Source presence is the perception of the proximity of the sound source.

8. Source warmth affects the level of the low-frequencies of the direct sound

9. Source brilliance affects the level of the high-frequencies of the direct sound

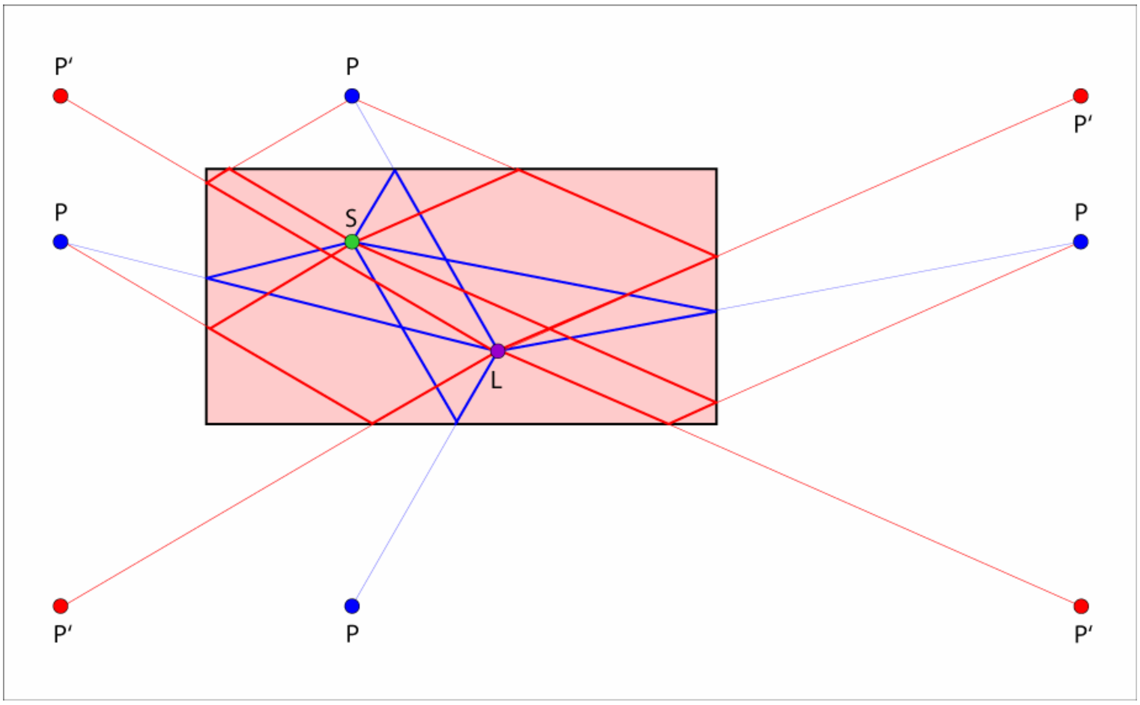
The situation in our application is different from the premises of work of Jot and Warusfel. In their model there is no geometry involved no geometric relation of source and listener nor any geometric description of the room such as size or shape. In our case we have geometric relation between source and listener given by the geometry of the simulated world, we like to keep this relation intact through the auralization process. To get a realistic impression of the direction to a given source the directions of early reflections must have a realistic relation to the source and the virtual room. Our approach is to actually simulate a virtual room and to employ a simple image source method **[Allen79]** to find the early reflections. One consequence is that some of the parameters from the Jot and Warusfel model will be given by the simulation of the room rather than by the user. New parameters are needed to describe properties of the virtual room. This will shift the focus from an abstract view of the room acoustic to a more concrete description of the room it self and its acoustic properties. We believe a description where the geometry of the simulated room is involved is more intuitive and easier to understand for people without a deeper understanding of room acoustics theory. Our parameter set is based on the description of a virtual room; its dimensions and acoustic properties and a few of the parameters borrowed from the Jot and Warusfel model.

The source presence parameter is unnecessary in our model as this effect will be given implicitly by the distance of the source and the listener in our model. The two other source parameters, source warmth and source brilliance has no relation to the room and will therefore not be included in our model. Room presence and envelopment are implicitly included in the description of the virtual room. We do not include running reverberance as we think this is a parameter of minor importance and we like to keep our model as simple and intuitive as possible.

2.2 The Virtual Room

The virtual room has a shoebox like shape that is described by a size parameter and two proportion parameters. The acoustic properties of the room are determined by the mean absorption coefficients, heaviness and liveness parameters. The simulation has two distinct components, the early reflections and the late reverberation.

The early reflections are calculated using a very simple image source algorithm. The algorithm can be simplified by using the fact that the geometry of the room is guarantied to be convex so there is no need to search for any hidden surfaces only raw calculation of mirroring the source need to be performed. A simplified example is showed in Figur 3. The example is done in 2D space to makes it easier to understand but in the actual application all computations are based on a 3D model. Blue dots marked P are mirrored images of the source S. The path of a reflection can easily be constructed using the phantom image. To find the point of incidence of the reflection a thin blue line is drawn from the phantom image to the listener L. The point of incidence is where this line is crossing a surface, from this point we draw two thicker blue lines representing the path, one to the source and one to the listener. Red dots marked P’ are second order phantom sources, the associated paths can be constructed in a similar way but now there are two points of incidence in the path. Not all possible second order phantom sources are included in the example only the first four ones, there are 12 in total in the 2D case (all may not be audible). The number of reflections the total system can handle is limited by the audio rendering module and the power of the machine it is running on.



Figur 3, Early Reflections

The late reverberation is simulated by a modified Schroeder reverb algorithm in the current implementation as this is the method implemented in the audio renderer however the reverberation parameters are general and can be used by any reverberation algorithm for instance and algorithm based on feedback delay networks (FDN) **[Rocchesso97]**. The parameters for the reverb are reverberation time (RT_{60}) in 10 frequency bands, the level and the delay, where the level and the delay are local to each sound source. The reverberation time is calculated from the volume of the room, the area of the enclosing surfaces and the mean absorption coefficients using the Sabine equation. This result is then adjusted to include the heaviness and liveness parameters.

The Sabine equation (see Equation 1) **[Kutruff91]** defines the relationship between the volume V of the room, the mean absorption coefficient a of the enclosing surfaces of area S and the reverberation time RT_{60}. In this application we are calculating the reverberation time in 10 frequency bands, n is the number of the frequency band.

$$RT_{60,n} = \frac{0.161\,V}{a_n\,S}$$

Equation 1, Sabine Equation

2.3 Room Parameters

The size parameter t of the room is the cubic root of the volume of the room (given in meters). The proportions are given as a height factor *h* and a width factor *w* in relation to the depth of the room.

The mean absorption coefficients are given in 10 frequency band (octaves) according to the ISO 266:1975 standard **[ISO95]**. There are default values for the absorption coefficients based on measurements of the acoustics of a concert hall. The default values are modelled after a concert hall to give a good reverberation when the room is scaled to different sizes.

The delay and the level of the late reverberation should be chosen so there is a smooth transition between the early reflections and the late reverberation. The delay and level of the reverb are calculated for each source and is depending on the early reflections of that source. The delay is calculated from the mean value of the distance of the five latest early reflections. The level is the inverse of the distance of the five latest reflections, the value is clamped to no be greater than 1 (this happens when the distance is less than 1 meter).

parameter	**description**	**range**	**default**	**comments**
size (t)	size of the room	1 - 100	24	$V = t^3$
width (w)	width factor of the room	0.1 - 10	0.7	$V = u^3 w h$ $u = t / (\sqrt[3]{w} \sqrt[3]{h})$
height (h)	height factor of the room	0.1 - 10	0.5	$V = u^3 w h$ $u = t / (\sqrt[3]{w} \sqrt[3]{h})$
surface fractal (f)	fractal factor enclosing surfaces of the room	1-2	1	$S' = S \cdot f$ where S is the area of the surfaces of the original shoebox
liveness (L)	affects reverberation time in freq. above 4 kHz	0.5-2	1	$RT_{8-10}' = RT_{8-10} \cdot L$
heaviness (H)	affects reverberation time in freq. below 250 Hz	0.5-2	1	$RT_{1-4}' = RT_{1-4} \cdot H$
abs_1 – abs _10	absorption coefficients in 10 freq. bands (ISO 266:1975)	0-1		default is based on a concert hall
rt_1 –rt_10	reverberation time (RT60) in 10 freq. bands (ISO 266:1975)	0.1-10		default values are calculated using Sabine equ. based on the room parameters and the absorption coefficients
early_order	highest order of reflection to use	1-4	2	
early_n	Maximum number of reflections to use	10-30	30	disregarding the order

2.4 Default Values

Default values of the parameter can be computed using the Sabine equation based on the geometry of the surrounding and the acoustical properties (absorption coefficients) of that surrounding. However it is not a well defined problem how to compute an estimate of the volume and the area of the surfaces of the surrounding space, some kind of heuristic algorithm is needed. There is no information given in the geometry of what could be considered as the acoustic surrounding. What is needed is a heuristic system that can reason about the topology of the space and make a conclusion about which parts of the geometry the surrounding consists of. It is very demanding to develop such system.

An alternative solution which in our case seems very attractive is to use the early reflections from the physical based simulation. The early reflections are like a sonar image of the surroundings, an estimate of the volume can be computed based on this information. We have developed a new algorithm to compute an estimate of the volume from a set of early reflections. First we compute the points of incidence, that is were the reflection is hitting a surface of the surrounding, for each of the reflection. The result is a set of points of incidence, it is per definition true that these points all lie on the shell of the surrounding space. Next step is to compute an estimate of the volume based on this cloud of points. It is very computational demanding to compute an estimate directly from the cloud of points.

To overcome this problem and to speed up the computation an oriented bounding box (OBB) **[Got00]** is computed and the estimate we are searching for is the computed volume and the area of the faces of that box. The OBB is a rectangular parallelepiped shape, the same as the virtual room of which we seek its parameters. The idea of a bounding box is to find the smallest box that includes all of the points in the given cloud of points (usually vertexes). OBB are commonly used in computer graphics problems, particularly in collision detection and ray tracing. There are numerous of established algorithms to compute OBB from a set of points. We have chosen a method based on principal component analysis (PCA) **[Dimitrov06]** to fit the OBB to the cloud of points. The method to compute the OBB is a heuristic method and it has been shown in empiric studies that the algorithm usually gives a good fit of the bounding box to the cloud of points in practical cases **[Dimitrov06] [Dimitrov07]**.

We will only give a brief description of the method here. A more detailed description is found in the papers by Dimitrov et al **[Dimitrov06] [Dimitrov07]**. PCA is an old method also known as the Karhunen-Loève transform, it is used in statistics and is traditionally used to reduce dimensionality in multivariate problems. Mathematically PCA is defined as an orthogonal linear transformation which transforms a data set to a new coordinate system such as in the projection of the data onto the new coordinate system the dimension with the greatest variance comes first. In our case this means that the first dimension will be the longest axis of the OBB, the next will be the second longest. The transform can be found using the covariance of the data set. In our case the data set is the cloud of points. The dimensionality of our problem is 3 and we also like the result to have 3 dimensions, so we are not using the PCA algorithm to reduce the dimensionality but to find the axis of our bounding box.

The data is represented as matrix where each row is representing one point. Consequently the matrix has 3 columns, one for each dimension. The empirical mean of the data set is subtracted before computing the covariance matrix. The eigenvalues and eigenvectors are computed from the covariance matrix. The rows of the eigenvector matrix is reordered in such a way that the eigenvector associated with the largest eigenvalue comes first, the one associated with the second largest comes second a.s.o. The sorted matrix of the eigenvectors is the transform we are searching for. The points are now projected onto the axis of the new coordinate system. The maximum and minimum values on each axis define the dimensions of the OBB. Once we have its dimensions it is trivial to compute its volume and the area of the enclosing surfaces.

2.5 Storing of Parameters

The notion of room has no counterpart in the geometry of the simulated world, the question is where the parameters of the virtual rooms should be stored. We have decided to store the parameters in a data structure that is associated with the floor, which is defined as the first surface that is hit by a ray sent from the centre of the listeners head and downwards. This will cover all situations in normal buildings and worlds.

To find witch virtual room that is associated to the position of the listener the geometry of the simulated world has to be searched. This could be computationally demanding if the world is big as potentially every surface in the world has to checked. To speed up this process the geometry is divided using the binary space partition tree (BSP) that is already implemented in UVAS. To find the virtual room only surfaces in the cell in the BSP-tree that contains the listener has to be search. Implementation

The Perceptual Simulation module is implemented and integrated in the UVAS framework which is described in deliverable D 7.3. Some of the modules of UVAS have been affected; the changes were often small and would not be mention further in this document. However there where parts of UVAS that were more profoundly affected which will be discussed below. The Sound Rendering module which is external to the UVAS framework described in deliverable D7.4 has also been affected: We have tried to affect the sound rendering as little as possible by reusing resources developed for the audio culling module. We have managed to meet this goal; only minor additions to the communication protocol were needed to be implemented in the sound rendering for the perceptual module.

3 Implementation

The UVSRP protocol was extended to handle the reverberation parameters. New function implemented in the API are; `set_reverb_time`, `set_reverb_level` and `set_reverb_delay`.

The main part of the Perceptual Simulation module is implemented as a subclass of the SimulationManager class.

The BSP algorithm implemented as part of the Beam Tracing module is also needed in the Perceptual Simulation module. The BSP algorithm has been factored out from the Beam Tracing module and is now implemented as its own independent manager.

A new class, Room, subclassed from DomainObject was added to represent the virtual room. Instances of the new class can be added to the Scene Graph.

New functionality in the Verse FrontEndManager has been implemented to handle the Verse representation of the room parameters.

New functionality in the UvsrpSoundRenderingManager to handle the reverberation parameters was implemented.

3.1 Storing room parameters in Verse

A room is represented by a text node in verse. Parameter values are stored as tags in tag group uvas in the text node.

Rooms are associated with faces in the geometry using a scheme similar to multi materials. The object node which is a parent of the geometry is linked to rooms with links that has a valid and unique target. The geometry node linked to the object has a VN_G_LAYER_POLYGON_FACE_xxx layer. The value of the face in this layer determines which room it is associated to by matching the value with the target of the room links from the object node.

parameter	**tag name**
size	room_size
width	room_width
height	room_height

surface fractal	room_surface_fractal
liveness	room_liveness
heaviness	room_heaviness
abs_1 – abs _10	abs_1 – abs _10
rt_1 –rt_10	rt_1 –rt_10
early_order	early_order
early_n	early_n

4 Future work

4.1 Room size versus enclosing geometry

In the current implementation the position of the virtual room relative to the geometry is defined by the position of surface it is associated with. However this can produce strange results when the size of the virtual room is much smaller than the enclosing geometry and the listener or the sources are far out of the virtual room. A solution to this would be to adjust the position of the virtual room and the position of the sources according to the position of the listener. The sources should be moved such as the direction to the listener is not changed and the distance relation could be kept by adjusting the level of the source and its reverberation accordingly.

5 References

[Allen79] J. Allen and D. Berkeley, "*Image method for efficiently simulating small room acoustics*," Journal of the Acoustical Society of America, vol. 65, no. 4, pp. 943--950, April 1979.

[Dimitrov06] Darko Dimitrov, Christian Knauer, Klaus Kriegel, Günter Rote. ”On the Bounding Boxes Obtained by Principal Component Analysis”. In Proc. 22nd European Workshop on Computational Geometry, Pages 193–196, Delphi, Greece, 2006.

[Dimitrov07] Darko Dimitrov, Christian Knauer, Klaus Kriegel, Günter Rote. ”Upper and Lower Bounds on the Quality of the PCA Bounding Boxes”
In Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision - WSCG 2007, Pages 185–192, Plzen, Czech Republic, 2007.

[Got00] Gottschalk, S. “Collision Queries using Oriented Bounding Boxes”. PhD Thesis. University of North Carolina. Department of Computer Science. 2000.

[ISO95] ISO Standards Handbook: Acoustics Volume 1: General aspects of acoustics; in general; noise with respect, 1995, Ed. 2, 616 p., ISBN 92-67-10221-4

[Jot95] J.-M. Jot and O. Warusfel. *Spat»*: A spatial processor for musicians and sound engineers. In *Proceedings of Colloquium on Musical Informatics,* Ferrara, Italy, May 1995.

[Jot99] J.-M. Jot, “Real-time spatial processing of sounds for music, multimedia and interactive humancomputer interfaces.” Multimedia Systems, Special Issue on Audio and Multimedia, vol. 7, no. 1, pp. 55-69, 1999

[Jullien 95] Jullien Jean-Pascal, “Structured Model for the Representation and the Control of Room Acoustical Quality”. ICA: International Congress on Acoustics. Trondheim : Juin 1995, p. 517-520

[Kutruff91] Kutruff, K. H. (1991) *Room Acoustics,* Elsevier Science Publishers, Essex.

[Moorer79] Moorer, J. A., "About This Reverberation Business", *Computer Music Journal*., vol. 3, no. 2, pp. 13–28, 1979.

[Rocchesso97] Rocchesso, D., and Smith, J. O. Circulant and elliptic feedback delay networks for artificial reverberation. IEEE Trans. Speech and Audio Processing, vol.5, no.1, pp. 51-63, 1997.

[Savioja99] L. Savioja. Modeling Techniques for Virtual Acoustics. PhD thesis, Helsinki University of Technology, 1999.

[Schroeder62] M. R. Schroeder.”Natural-Sounding Artificial Reverberation”. J. Audio Eng. Soc., 10(3):219¬233, July 1962.

[Väänänen04] Väänänen, R.; Huopaniemi, J. ”Advanced AudioBIFS: virtual acoustics modeling in MPEG-4 scene description”, Multimedia, IEEE Transactions on Volume 6, Issue 5, Oct. 2004 Page(s): 661 – 675