Parametrization, Auralization, and Authoring of Room Acoustics for Virtual Reality Applications

Riitta Väänänen

9th May 2003

HELSINKI UNIVERSITY OF TECHNOLOGY P.O. BOX 1000, FIN-02015 HUT http://www.hut.fi	ABSTRACT OF DOCTORAL DISSERTATION

Author Riitta Väänänen

Name of the dissertation

Parametrization, Auralization, and Authoring of Room Acoustics for Virtual Reality Applications

Date of manuscript May 10, 2003	Date of the dissertation
☐ Monograph	☑ Article dissertation (summary + original articles)

Department	Electrical and Communications Engineering
Laboratory	Laboratory of Acoustics and Audio Signal Processing
Field of research	Virtual Room Acoustics
Opponent(s)	Dr. Jean-Marc Jot
Supervisor	Prof. Matti Karjalainen
(Instructor)	Dr. Jyri Huopaniemi

Abstract

The primary goal of this work has been to develop means to represent acoustic properties of an environment with a set of spatial sound related parameters. These parameters are used for creating virtual environments, where the sounds are expected to be perceived by the user as if they were listened to in a corresponding real space. The virtual world may consist of both visual and audio components. Ideally in such an application, the sound and the visual parts of the virtual scene are in coherence with each other, which should improve the user immersion in the virtual environment.

The second aim was to verify the feasibility of the created sound environment parameter set in practice. A virtual acoustic modeling system was implemented, where any spatial sound scene, defined by using the developed parameters, can be rendered audible in real time. In other words the user can listen to the auralized sound according to the defined sound scene parameters.

Thirdly, the authoring of creating such parametric sound scene representations was addressed. In this authoring framework, sound scenes and an associated visual scene can be created to be further encoded and transmitted in real time to a remotely located renderer. The visual scene counterpart was created as a part of the multimedia scene acting simultaneously as a user interface for renderer-side interaction.

Keywords virtual acoustics, room acoustic modeling, 3D sound, sound scene description, MPEG-4, authoring

UDC 534.84:004.032.6	Number of pages 167
ISBN (printed) 951-22-6543-5	ISBN (pdf) 951-22-6544-3
ISBN (others)	ISSN 1456-6303

Publisher Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing

Print distribution Report 70 / HUT, Laboratory of Acoustics and Audio Signal Processing, Espoo, Finland

☑ The dissertation can be read at http://lib.hut.fi/Diss/

Preface

This work was carried out in the Laboratory of Acoustics and Audio Signal Processing, Helsinki University of Technology (HUT), Finland during the period between 1997 to 2002, and at Institut de Recherche et Coordination Acoustique/Musique (IRCAM), Paris, France, during my one-year visit from February 2001 until February 2002, and going on from September 2002. During 1999-2002 the work has been done as a member of the Pythagoras Graduate school.

I would like to thank my supervisor Professor Matti Karjalainen for his supervision and support throughout the years that it took to bring this thesis work to completion. I am also grateful to Dr. Jyri Huopaniemi, who has acted as my practical instructor during the whole thesis work. Jyri gave the first inspiration to this work, and we have co-operated in the MPEG-4 standardization work, and written several publications together. Also the other co-authors of my thesis publications, Professors Matti Karjalainen and Vesa Välimäki, and Doctors Eric Scheirer and Ville Pulkki, deserve thanks for helping in making this thesis. I would also like to express my gratitude to the DIVA (Digital Interactive Virtual Acoustics) group, including Multimedia lab people, Professors Tapio Takala, and Lauri Savioja, and Dr. Tapio Lokki, for guiding me to the world of virtual acoustics, and for co-work and co-authoring in several publications. I am also thankful to IRCAM, in particular the Room Acoustics team and Dr. Olivier Warusfel for supervision and co-operation, Mr. Olivier Delerue for letting me mess up with ListenSpace (even helping me in that!), and the Carrouso project partners, especially Dr. Giorgio Zoia for concretical co-work in MPEG-4 related questions. Also without the co-operation with the MPEG-4 community, especially the Systems and Audio subgroups, and the reference software implementation (IM1) working group, this work would not have reached its current form. Special thanks belong to the pre-examiners of this thesis, Dr. Veronique Larcher and Professor Moncef Gabbouj, and to Dr. Nick Zacharov for their help in improving the quality of this thesis.

I also would like to acknowledge the financial supporters of this thesis work: Academy of Finland (through the Pythagoras Graduate school), Nokia Research Center, Nokia Research Foundation, HPYn tutkimussäätiö, Tekniikan edistämissäätiö, Jenny ja Antti Wihurin rahasto, Eemil Aaltosen säätiö, and IRCAM.

I've carried out most of the thesis research as one of the founding members of the Ladiesroom in the Acoustics lab. Therefore, many thanks for a pleasant working environment also belong to other Ladiesroom members: Ville Pukki, who has helped not only

in understanding the importance of VBAP, but also in maintaining the good humor even when the thesis work got deep and sideways. Ex-member Mairas has been great for his bad humor, and last but not least I would like to say thanks to Dr. Hanna Järveläinen (Hannasta ja Riitasta) for friendship, good travel company, and sharing the good and the bad while making our theses pretty much in phase in the Pythagoras Graduate School. Special thanks belong also to a hangaround member Henkka for not only hanging around (which is important) but also for friendship and encouragement. I'm grateful to other Akulab people, including Mara "kievinkana" Rahkila and Hynde for help in computer-related problems, and Lea Söderman for support and help in any practical issues. I would also like to thank Doctors Tero Tolonen, and Cumhur Erkut for their friendship and support during many years.

There are many people also out of the working circles to whom I would like to give special acknowledgements: My parents Raija and Eero Väänänen, and my sisters Kaisa and Marja, as well as my friends Johanna Blomqvist, Marika Saarinen, Anna-Maija Autere, and Gianluca "pomminpurkaja" Di Cagno have supported and encouraged me in this work, and helped me in remembering that there are more important things in life than a PhD. Finally, I'm grateful about the support and presence that I've got from Mr. Olivier Delerue in all different areas of life.

Riitta Väänänen Paris, France, 7.5.2003

Table of Contents

List of Publications

This thesis summarizes the following publications, referred to as [P1]-[P7]:

[P1] R. Väänänen, J. Huopaniemi, V. Välimaki, and J. Karjalainen. Efficient and parametric reverberator for room acoustics modeling. In *Proceedings of the International Computer Music Conference (ICMC '97)*, pages 200–203, Thessaloniki, Greece, September 25–30 1997.

[P2] E. Scheirer, R. Väänänen, and J. Huopaniemi. AudioBIFS: Describing audio scenes in MPEG-4 multimedia standard. *IEEE Transactions on Multimedia*, 1(3):237–250, 1999.

[P3] R. Väänänen, and J. Huopaniemi. Virtual acoustics rendering in MPEG-4 multimedia standard. In *Proceedings of the International Computer Music Conference (ICMC 1999)*, pages 585–588, Beijing, China, October 1999.

[P4] R. Väänänen and J. Huopaniemi. Spatial processing of sounds in MPEG-4 virtual worlds. In *Proceedings of EUSIPCO 2000 conference* , Vol. 4, pages 2209-2212, Tampere, Finland, September 2000.

[P5] R. Väänänen, J. Huopaniemi, and V. Pulkki. Comparison of sound spatialization techniques in MPEG-4 scene description. In *Proceedings of the International Computer Music Conference (ICMC 2000)*, pages 288-291, Berlin, Germany, September 2000.

[P6] R. Väänänen, and J. Huopaniemi. Advanced AudioBIFS: Virtual acoustics modeling in MPEG-4 scene description. *Accepted for publication in IEEE Transactions on Multimedia.*

[P7] R. Väänänen. User interaction and authoring of 3D sound scenes in the Carrouso EU project. *Preprint No. 5764 of the 114th Convention of the Audio Engineering Society (AES). Amsterdam, The Netherlands, March 2003.*

List of Symbols

A	amplitude
d	distance
f	frequency
dB	decibel
DS, **R1**, **R2**, **R3**	direct sound, directional early reflections, diffuse early reflections, and late reverberation of a simulated room impulse response
g	gain applied to sound signal
Hz	Hertz
L	listener position
M	magnitude in a magnitude frequency response
$\boldsymbol{RT}_{60}$	reverberation time
s	second
S	sound source position
t	time
t_0, t_1, t_2	delays of direct sound, directional early reflections, and diffuse reverberation of a room impulse response
α_n	angle (direction of radiation of a sound source)

List of Abbreviations

2D	two-dimensional
3D	three-dimensional
3DWG	3D Working Group
AABIFS	Advanced AudioBIFS
API	Application Programming Interface
BEM	Boundary Element Method
BIFS	BInary Format for Scenes
CASA	Computational Auditory Scene Analysis
CELP	Code Excited Linear Prediction
DIVA	Digital Interactive Virtual Acoustics
DML	Distributed Mode Loudspeaker
DSP	Digital Signal Processing
EAX	Environmental Audio Extension
FDN	Feedback Delay Network
FDTD	Finite Difference Time Domain
FEM	Finite Element Method
FIR	Finite Impulse Response
GA	General Audio
HRTF	Head-Related Transfer Function
HRIR	Head-Related Impulse Response
HVXC	Harmonic Vector eXctiation Coding
I3DL1	Interactive 3D Audio Rendering Guidelines Level 1.0
I3DL2	Interactive 3D Audio Rendering Guidelines Level 2.0
IIR	Infinite Impulse Response
ILD	Interaural Level Difference
IR	Impulse Response
ISO	International Standardization Organization
ITD	Interaural Time Difference
IASIG	Interactive Audio Special Interest Group
MPEG	Moving Picture Expert Group
OOP	Object-Oriented Programming

RIR	Room Impulse Response
SNHC	Synthetic/Natural Hybrid Coding
VAD	Virtual Auditory Display
VAE	Virtual Audio Environment
VBAP	Vector Base Amplitude Panning
VR	Virtual Reality
VRML	Virtual Reality Modeling Language
WFS	Wave Field Synthesis
XMT	Extensible MPEG-4 Textual format
X3D	Extensible 3D (Graphics) specification

1. Introduction

This thesis deals with parametric representation of acoustic environments and sound sources in the context of interactive virtual reality systems. During the work a software implementation was also performed to realize sound scenes that correspond to the parametric descriptions developed in this thesis. Finally, authoring of such scenes was addressed in the context of real-time communication. In order to create a virtual acoustic model in such a framework, the 3D sound scene data is first recorded, and the parametric scene is obtained with a help of the authoring tool. The scene is transmitted to a receiving terminal, where it is rendered audible so that a similar acoustic impression is obtained as in the recording situation.

1.1 Background for Sound in Virtual Realities

In the context of this thesis, the term virtual reality (VR) refers to a simulated real or imaginary environment. Typically it is presented to the user in three dimensions, using a computer screen (or more advanced display technology) to show the graphical part of the virtual world, and loudspeakers or headphones for playing the sound. Often the virtual reality environment is also expected to include a possibility for interaction. This interaction can be simple user navigation inside of the three-dimensional (3D) world without the possibility to affect the content of the world or its events. The interaction can also include advanced mechanisms for modifying the virtual world with the help of different input devices [1]. An important criterion for a successful virtual reality application is usually its immersiveness, meaning the feeling of realism experienced by the user about being present in the virtual environment. Factors that affecting this include, for example, the realistic quality of the graphics and sound, interactivity, and audiovisual synchronization (consistency between the audio and visual counterparts of the virtual scene). Also the rendering technology, i.e., the output devices used for displaying the graphics and reproducing the sound, affect the immersiveness and overall quality of the application.

Traditionally the emphasis in developing virtual reality systems has been on the graphics side, giving the virtual sound environment modeling lower priority. A natural explanation for this is that the vision is the most dominating human sense, as we obtain most information about the environment through our eyes. As a consequence, in the field of computer science, the development of computer graphics for providing visual output and feedback to user actions has been necessary, whereas sound output has not been con-

sidered to have much importance. The sound has become relevant more recently, first in the digital music industry, and thereafter in computers in context of multimedia and virtual reality (VR) applications [2]. Recent development of Application Programming Interfaces (API) has facilitated the programming of spatial sound content into audiovisual applications. Thus sound is slowly becoming a natural and accepted (even expected) part of multimedia applications.

When sound is included in virtual worlds, the level of detail to which the sound environment should be modeled depends on the application, and the computational resources reserved for processing the sound. In the simplest case sounds are played during the virtual world rendering without any spatial cues. An advancement to this is to define virtual sound sources with positions in the virtual space. These positions should be taken into account during the processing and playback of the sound so that the they are convincingly perceived by the user of the application. More detailed modeling of sound transmission in a space can be designed using the knowledge from the research carried out in the area of computer-based room acoustic modeling. Formerly this knowledge has been utilized for example in room acoustic prediction programs for architectural design, or in music production for creating spatial sound effects. However, with an increasing computational power of computers, we may wish to integrate more of those details of sound propagation even in generic multimedia applications and entertainment software (such as computer games). At the same time, the development of different communication networks (such as computer and mobile phone networks), encourage developing ways to transmit as large amount of data and applications as possible in a restricted bandwidth. A good example of a scheme where modern applications and communication channels are taken into account is the MPEG-4 standard, specifically developed for creating interactive multimedia applications [3]. One of the main goals of this thesis was to introduce means to flexibly and efficiently define spatial sound scenes within such a multimedia framework. With the defined sound scene parametrization tools it is possible to introduce interactive spatial sound scenes in both audio-only and audiovisual (e.g., virtual reality) applications.

1.2 Aim of the Thesis

In this thesis the primary aim was to create a framework for parametric representation of acoustic properties of sound sources and their environment. This representation can be used to store and transmit virtual acoustic spaces to be rendered on a computer that has the capabilities of interpreting and implementing the described sound scene. The advantage of the parametrization of a sound environment is, that it enables an efficient description of the space, i.e., transmitting it over a data channel does not reserve much bandwidth, nor storing it on a computer disk consumes a lot of disk space. Another advantage is that different factors that affect the final audible result can be modified independently and interactively. For example, the listener of the rendered scene can be given the possibility to move sound sources or change the reverberation properties of a virtual room. The sound scene parameters can be derived, for example, from geometrical properties of existing spaces (such as a room or a concert hall), from computer models of rooms, or they can be

designed by the content author of the virtual scene without a correspondence to the physical reality. At the *auralization* stage, where the parametrically defined scene is rendered audible [4], these parameters are used to create a computer simulation of the space and the sound sources. This means that when the scene is rendered, the sound emitted by a virtual sound source is processed so that the user has an impression of being in a corresponding real space that the defined parameters describe.

The second aim in this thesis was to implement a system that realizes the auralization of the acoustic scenes that are defined with the above-described parameter set. This was done as a part of a larger software framework that implements the rendering of dynamic and interactive scenes that may include various kinds of interactive visual and sound content (more precisely, the MPEG-4 reference software). This program takes a parametric scene description as input and outputs a rendered scene (its visual part on a computer screen and the sound scene through loudspeakers or headphones).

Thirdly, the authoring of spatial sound scenes was addressed at the final stage of the thesis. This involved developing and verifying a tool (again, as a part of a larger sound scene authoring tool framework) that through a graphical interface enables to produce and modify a sound scene. For such an experiment, the authoring may be performed as an off-line or a real-time process. The real-time scheme involves a complete communication chain from authoring and encoding of an audiovisual (or audio) scene, transmitting it over a network, to decoding and rendering it to the user. This part of the work proved that the proposed sound scene parametrization framework, and its real-time transmission and rendering, are mature technologies for adoption.

1.3 Related Research Areas

The topics of this thesis are closely related to several other research areas in the fields of sound environment modeling, software, and multimedia. Although its main emphasis is on the parametrization and control of 3D sound environments (as presented in publications [P2]-[P7]), the thesis also addresses the modeling methods (the digital audio signal processing algorithms) for auralization. This part has been carried out in parallel and partly in co-operation with researchers working in the areas of room acoustic and listener modeling (see, publication [P1], [5], and theses [6], [7], [8]). The topic of this thesis is also related to the modeling of the sound content production (sound synthesis), which is increasingly relevant in modern and future multimedia applications [9], [10], [11]. Furthermore, computational auditory scene analysis (CASA) technologies provide means for sound sources to be separated, and individually encoded according to object-based sound coding principles [12]. This topic is beyond the scope of this thesis, although it is useful for creating sound scenes with the tools developed in this work. Furthermore, psychoacoustic evaluation has been carried out in similar contexts concerning the perception of different components of virtual acoustic modeling. The results of those studies are useful especially in optimizing the data included in sound scenes, and the corresponding renderer implementation [13], [7], [14]. Finally, the room acoustic parametrization that was developed in the framework of this thesis, is closely connected to object-oriented scene

description languages that are traditionally used for creating 3D graphical applications. This parametrization allows extending such graphical scenes with immersive acoustic properties with little additional data and programming effort.

1.4 Outline of the Thesis

This introductory part is organized in the following way: The next two chapters provide an overview on the virtual acoustic research area, and the relation between this work and earlier research. Chapter 2 explains the virtual acoustic modeling. It presents the commonly accepted model of *source – medium – receiver* in the sound environment modeling context, and the connection between that model and its parametric definition. The different stages of auralization are explained. An overview is made of room acoustic modeling and sound reproduction methods, as the knowledge of those technologies is useful for generic sound environment modeling. Chapter 3 presents in more detail the background for object-based and parametric definition of computer-modeled sound environments. It offers an overview of application programming interfaces that can be used for adding sound to applications. Also the MPEG-4 framework, a new technology for creating virtual scenes (with enhanced spatial sound API) and applications containing various different types of media, is briefly explained. Finally Chapter 4 presents a summary of the publications included in this thesis, and Chapter 5 concludes the thesis and discusses future work.

2. 3D Sound Environment Modeling

This chapter describes concepts related to definition and rendering of virtual 3D sound environments. It acts as background information to how such environments can be parametrically encoded and rendered, which are the main topics of this thesis, and dealt with in publications [P1]-[P7]. Section 2.1 first overviews common virtual acoustic modeling concepts, including the components that are subjects to modeling (i.e., the sound source, the transmitting medium, and the receiver), and the stages of auralizing sound environments. In Section 2.2 aspects are explained that need to be taken into account in forming definitions of virtual sound environments. Thereafter, Section 2.3 discusses different ways of parametrizing and computer-based modeling of spatial sound environments. Finally, Section 2.4 overviews reproduction technologies used for the playback of virtual acoustic environments.

2.1 Virtual Acoustic Definitions

Virtual acoustic modeling in this work refers to a process, where the behavior of sound in a room is simulated so that, according to some criteria, the simulation reproduces the behavior that the sound would have in a real space corresponding to the simulated model. The virtual acoustic modeling has different aims in different applications. It is used, e.g., for room acoustic simulation when designing the architecture of a concert hall or a room from an acoustical point of view before it is built, or for evaluating the acoustics of existing rooms. Room acoustic modeling can also be used in communications applications for telepresence, i.e., to provide a listening experience without the need for the listener to be in the real space that the model is based on. Thus it seems natural that methods and tools developed for room acoustic modeling can also be used for creating virtual sound environments that are not derived from existing spaces, and also for creating (artificial) spatial sound effects [4], [15]. The latter approaches are particularly relevant in virtual reality and multimedia applications, of which the sound forms an integral part. The toolset that was created in the framework of this thesis is meant for fulfilling the needs of such applications.

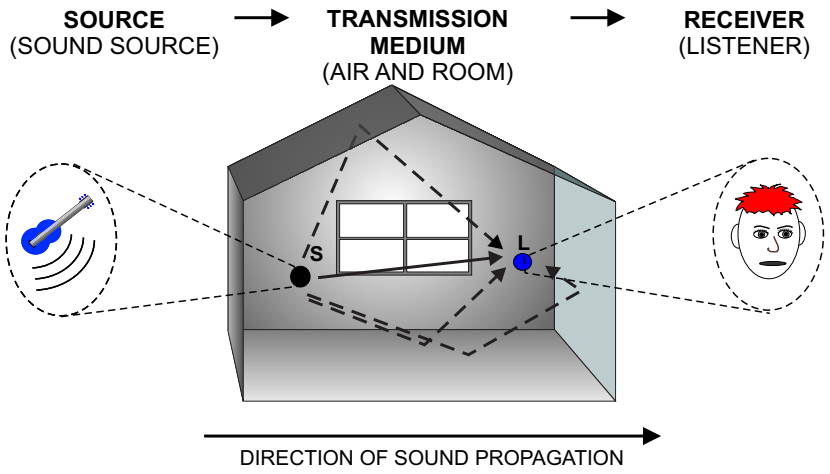

Figure 2.1: Source – medium – receiver model in acoustic communication.

2.1.1 Source – Medium – Receiver Model in Virtual Acoustics

A model that is applied in the sound propagation simulation usually follows the division of the acoustic communication system to three principal components: the sound source, the sound transmitting medium, and the receiver (e.g., the listener) [2]. Figure 2.1 illustrates this division. The following properties can be associated with the three components:

1. *Source modeling* consists of recreating the spatial characteristics of a sound source, i.e., its spatial location usually in 3D coordinates, and the directivity, i.e., the direction-dependent radiation pattern caused by the physical properties of the source. Directivity of musical instruments has been addressed, e.g., in [16], [17], [18], and that of a human head in [19], [20].

   Often the sound production mechanism is beyond the source model definition, as the actual sound content (emitted by the source) typically is obtained by streaming sound samples from a file (or from some other source, such as real-time synthesis or recording of sound). However, it may be well justified to include the sound as a part of the source model when it is synthetically produced by, e.g., music synthesis algorithm or speech synthesizer. Regardless of the means by which this sound content is obtained, it is ideally dry and monophonic to avoid overlapping of the spatial sound effects at the auralization stage.

2. *Medium modeling* consists of producing an effect of the simulated environment, which is applied to the sound originating from the source. Thus, this stage includes modeling of the sound propagation in the medium (typically air), and modeling of the effects caused by sound-interfering objects. The effect of the medium depends on the distance between the source and the receiver, often causing sound attenuation, and increased lowpass filtering effect, both proportional to that dis-

tance. The finite speed at which the sound propagates in the medium also causes a delay that increases as a function of distance (causing a Doppler effect when the distance changes rapidly with time). The sound interfering objects (such as walls), on the other hand, may cause sound reflections, or occlusion and obstruction when the source and the listener are on opposite sides of the object. Additionally, when modeling enclosed spaces that are formed of reflective surfaces, room reverberation effect may be added to simulate the result of multiple reflected sounds. These phenomena are taken into account in most room acoustic simulation methods, which will be overviewed in section 2.3.

3. *Receiver modeling* finally consists of taking into account the position and possibly the directive properties of the receiver. For example, in an application that is used for computation of monophonic room acoustic parameters, the receiver can be modeled as an omnidirectional microphone. In auralization and virtual reality applications, on the other hand, where the aim is often to listen to the sound as if the listener was in the virtual space, the system may introduce a sophisticated model of human binaural hearing. In the receiver modeling, the sound reproduction system (e.g., the loudspeaker or headphone setup) also affects the detailed processing algorithms applied to the sound before its playback (see, section 2.4 for the reproduction methods).

2.1.2 Auralization of Sound Environments

In auralization of sound environments the aim is usually to produce a perceptually satisfying audible result to a listener at one position at a time in the simulated, virtual sound environment. As proposed in [13] [5], the steps that are needed for performing such auralization can be presented as follows (see, Figure 2.2):

1. *Model definition* comprises of providing the data and control parameters for describing the properties and dynamic events of the modeled environment (consisting of the source, medium, and the receiver models). The model definition is a preliminary stage to the actual auralization process that consists of the two following steps.

2. *Modeling* comprises of the actual (nowadays mostly computational) acoustic simulation of the environment, which is carried out according to the source – medium – receiver model. The simulation may contain dynamic (time-varying) events and user interaction with the sound environment, through interactive modification of the defined control parameters. These control mechanisms can be defined as fixed features of the auralization program, in which case the interaction is always similar (e.g., listener movement in the defined environment) [5]. Another option is to make the interaction a part of the model definition in which case it can more flexibly be adjusted to the contents and the purpose of the application. The latter option is discussed in [P7] of this thesis in the context of authoring of sound environments and simultaneously defining customized interaction mechanisms for them.

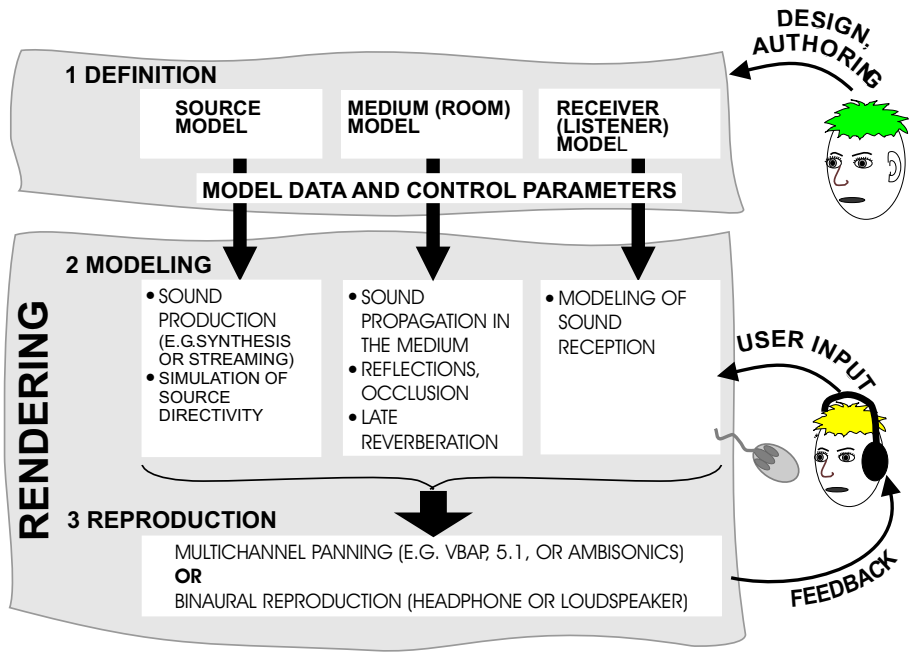

Figure 2.2: The three stages of auralization. At model definition, a parametric representation of the sound environment is created (by the author of the virtual environment).These parameters are used at the modeling stage for producing the appropriate spatial effect to sound. Rendering of the sound environment includes both the modeling and reproduction that are needed to reproduce the modeled sound field to the listener. This stage may involve user interaction that leads to control parameter changes.

3. *Reproduction* consists of rendering the digitally simulated sound field to an acoustic, audible sound field, where the user can hear the virtual sound sources at their defined positions, and the effect of the defined room acoustic model.

The required stages of auralization are applied to the three components of a virtual sound environment described previously (the source, medium, and the receiver). The two latter stages of auralization (modeling and reproduction) together form the rendering of a sound environment. In the following sections, the above three stages of auralization are explained in more detail.

2.2 Model Definition of Spatial Sound Environment

When a computer model of an acoustic environment is created, the first step is to define a model, i.e., a parametric representation of the environment, the sound sources, and the control parameters. The data definitions are needed for setting up the actual auralization process, i.e., the required digital signal processing (DSP) network. These data parame-

ters include, for example, a geometric description of a room with associated reflectivity properties of its walls, and sources characterized by their positions and directivities. The control parameters, on the other hand, are time-dependent variables of the system that are mapped to changes in DSP parameters that implements the auralization. User interaction in practice means routing of user input (e.g., through a keyboard, mouse, or head-tracker) to these control parameters.

To construct the model and control data in a format that can be fed to the auralization system, user intervention is usually required. An authoring tool is a program that helps in this process, by providing an interface that facilitates the sound environment definition. Publication [P7] of this thesis deals with this task by discussing a tool where the basic sound environment setup is graphically created, and the detailed parametric data is provided as a textual input. This data is converted to a format that can be interpreted by the rendering system, so that a real-time auralization of the defined sound environment is possible.

Most room acoustic modeling systems have their own parametrization to describe the virtual sound environment, and therefore different systems are not directly compatible with each other. In other words a sound environment defined to be used in one modeling software can not necessarily be directly used in another. As an improvement to this, among the main aims of this thesis concerning the definition of virtual sound environments were to:

1. Define a standardized set of parameters that can be used to create a 3D sound environment, which can be rendered on different platforms and with different reproduction systems. This also includes defining conformance rules, which ensure that in different rendering systems the audible results are consistent. This means that the exact algorithms and possible reproduction methods for rendering are not defined, but that the rendering result of a standard-compliant renderer has to meet certain requirements.

2. Ensure that this parameter set is generic enough for producing computer models of sound environments for a large number of different applications in a flexible manner, and that the created models or parts of them can easily be re-used. The applications may range from spatial sound effects processing (e.g., in music reproduction or movie track generation) to detailed room acoustic modeling for architectural design and evaluation of concert hall acoustics (for which commercial applications exist, as will be shown in the next section).

3. Ensure also that this set of parameters is extendable, so that in the future new features can easily be added to enable more accurate modeling of sound environments. Extensions may be needed, e.g., when progress is made in the spatial sound research, and when the computational power of rendering devices (e.g., computers) increases.

The described scheme was realized during this work as a part of a large object-oriented multimedia coding framework, where a versatile set of parameterized objects are available

for simulating interactive audio and visual environments (or combinations thereof). The defining of spatial sound environments has been dealt with in publications [P2], [P3], and [P6] of this thesis.

2.3 Room Acoustic Modeling Overview

This section explains different approaches to modeling of room acoustics. The motivation for this was, that the knowledge contained in the computational room acoustic modeling methods was useful also for creating generic virtual sound environments. Section 2.3.1 presents some commonly applied practices in (particularly real-time) room acoustic modeling implementations, and section 2.3.2 explains the concepts of physical and perceptual room acoustic modeling approaches. Finally, existing methods for room acoustic modeling (according to given geometry) are overviewed in Section 2.3.3.

2.3.1 Computer-based Generation of a Room Response

In auralization a binaural room impulse response is synthesized at a given listening position in the modeled space, and convolved with the source sound material (e.g., with anechoic or synthesized sound). Thus the sound signal that is reproduced to the listener of the system, contains the direct sound, and the effects of the reflected and reverberated sound. The audible result is usually dependent on the positions of the sound source and the listener in the virtual space. Thus in a dynamic (time-varying) auralization where the user can navigate in the virtual environment (i.e., continuously change the viewing position in it), the perception of sound varies accordingly. In the case of most sound reproduction systems, the rendering is carried out for one listening point at a time. In such a situation, input data to the rendering system is required about the moving listening position. This input is used to adjust the parameters, which control the rendering of the virtual listener position. An exceptional loudspeaker reproduction method is the Wave Field Synthesis (WFS), where the virtual space can be auralized for many users simultaneously, independent on their positions (see, 2.4).

2.3.1.1 Division and Parametrization of the Room Impulse Response

In real-time computer implementations of virtual room acoustics, the room impulse response (RIR) is usually divided to three time-domain parts, which are the direct sound, early reflections, and late reverberation [21], [22], illustrated in Figure 2.3. Three motivations for this can be pointed out:

- From the *physical* point of view the early part of the response can roughly be seen as separate sound events. This means that the direct sound (**DS**) in Figure 2.3 and each early reflection (**r1** to **rn** in Figure 2.3) can be characterized by their directions, delays, attenuation and modification of their spectral contents (caused by the air absorption, and the absorption of reflecting surfaces). The detailed shape of the time-domain representation of the early part of a room response depends strongly

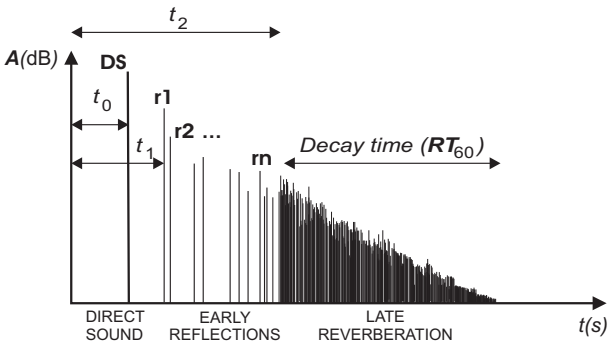

Figure 2.3: Division of room impulse response to direct sound (denoted with **DS**), early reflections (**r1** to **rn**) and late reverberation (characterized by its reverberation time RT_{60}). The direct sound is delayed by time t_0, which depends on the distance between the source and the listener. The early reflections and late reverberation (delayed by t_1 and t_2, respectively) are delayed more because of the longer propagation path of sound in these parts of the room impulse response (RIR).

on the geometry of the acoustic space, and the positions of the source and the listener in it. The late reverberation, on the other hand, in a typical reverberating space resembles a random process. It can often be considered an exponentially decaying random noise sequence (characterized by the reverberation time, RT_{60} in Figure2.3). Also the sound field is considered as (nearly) diffuse, i.e., the late response is independent on the source and listener positions, and the reflections arrive from all directions with the same probability.

- From the *perceptual* point of view the early part of the room impulse response affects the impression of the room geometry [23], while the reflections during the late reverberation can not be perceived separately. From this it may be concluded that in room acoustic simulation it is not necessary to carry out detailed modeling of the late reverberation, but that it is sufficient to produce an artificial reverberation effect to achieve the same perceptual result as the real room.

- From the *implementational* point of view it is more efficient to model the early part of the response in a detailed manner (e.g., each reflection separately), and the late reverberation with a DSP filter structure that implements the above-mentioned late reverberation properties. In an ideal case the late reverberation can be characterized and controlled with a small number of parameters, which do not need to be changed during the simulation of a single room.

Thus especially in real-time auralization, the computer-simulated room impulse response is parameterized so that these three parts can be modeled and controlled separately [24], [5]. The following section presents late reverberation modeling methods, that follow the same principles used in most real-time room acoustic modeling systems. After that section 2.3.2 discusses the physical and perceptual room acoustic modeling approaches.

2.3.1.2 Late Reverberation Modeling

For the above reasons, the problem of late reverberation simulation is reduced to DSP filter design where the impulse response of the reverberator contains the required properties. Often infinite impulse response (IIR) filters are used because long reverberation times can be obtained while keeping the computational complexity low. The first artificial digital reverberators were constructed of parallel comb filters and series-connected allpass filters [25], [26]. The comb filters are ideal for this purpose in a sense that they have an exponentially decaying impulse response, their decay time (that can directly be mapped to the reverberation time) of which can completely be controlled by the gain in the feedback loop. And by replacing this gain by a lowpass filter, the reverberation time can be made frequency-dependent, which allows the simulation of the natural phenomenon of decreasing reverberation time as a function of frequency (caused by the absorption of air and reflecting surfaces) [27]. The number of delay lines, and their total length, define the density of reflections and modal density, respectively. The allpass filters, on the other hand have been typically used to increase the reflection density and to create a diffuse effect for individual reflections. However, the drawback of using comb filters is the lack of modal density in the response (causing a metallic sounding effect) and the fact that they do not produce an increasing reflection density as a function of time, which is a property of natural reverberating spaces. Feedback delay networks (FDN) provide a solution to that: they provide less coloration and better reflection density than the comb filters while maintaining the reverberation time control, as explained in [27], [28], [29], [30].

Other reported reverberator structures include several allpass filters that are connected in series and nested with each other are proposed in [31]. In [32] the relation between FDNs and waveguide meshes for producing reverberation has been studied, and in [33] a fast convolution method is proposed to convolve sound with sampled room impulse response. In [34] an overview is given about reverberator DSP design.

In publication [P1] a reverberator DSP structure is proposed that combines the idea of comb filters, the FDN, and the allpass filters; Comb filters are connected in parallel, and with each delay line a lowpass filter is added to control the frequency dependent reverberation time. The outputs of the comb filters are summed and fed back to the reverberator input with a defined feedback factor. This corresponds to a special case of an FDN, but is implementationally more efficient than using a feedback matrix. In addition, with each delay line an allpass filter is added, which helps to rapidly increase the reflection density in the reverberator output. Also, the same property as in FDNs in general, namely the incoherent output from different delay lines, can be used to simulate a diffuse sound field by feeding them to different reproduction channels. This structure is also used in the DIVA system described in [5]. A further development for this reverberator is reported in [35].

2.3.2 Physical and Perceptual Characterizations of Room Acoustics

Often when discussing the acoustic modeling of rooms and other sound environments, two modeling approaches are brought up, namely, the physical and the perceptual approaches. The former is based on modeling of sound propagation in an environment given its physical configuration. The latter, on the other hand, is based on creating an

effect that imitates the perceived audible impression of a space rather than precisely simulates its room impulse response [24], [13]. The backgrounds of these two approaches are mostly in different directions, although they are increasingly starting to overlap and co-exist in modern computer-based applications. Traditionally the physical approach has been used in detailed room acoustic modeling for the purposes of architectural acoustic design and evaluation. The perceptual approach, on the other hand, has mostly been used in producing spatial effects to sound in music business.

2.3.2.1 Physical Approach

In the physical approach, a model of the studied (typically enclosed) space is defined in terms of its geometry, and usually a single sound source and a receiver are given positions in the defined enclosure. Acoustic properties are associated with these components, such as the reflectivity of the surfaces, and the directivity of the sound source. According to all this data, an artificial impulse response is generated by simulating the propagation of a sound in the modelled space. If the geometrical configuration of the room, or the position of the source or the receiver is changed, the whole room response is re-calculated. This approach can be used to produce an impulse response to be used, e.g., for computing statistical room acoustic parameters of the modeled space. Such modeling can also be used for auralization of the space by giving (ideally monophonic, anechoic) sound as input and adding the processing needed to simulate the directional hearing of the listener, before the sound is reproduced.

Recent research and development of room acoustic modeling methods have applied the physical approach also to dynamic applications, where changes of the acoustic parameters (such as movement of the listener) may happen in real time. This inevitably calls for simplifications in the modeling of sound propagation, and careful design to implement the parameter changes at a signal processing level is required to avoid audible artefacts. In real-time physical modeling the sound reflections are modelled up to a certain time, number or order, and the late reverberation is replaced by a statistically controlled late reverberator, as explained in Section 2.3.1. This approach in the context of real-time auralization is extensively explained in [5]. It is also applied in the system, the definition and implementation of which was done in this thesis, and in the MPEG-4 standard framework, discussed in publications [P3] and [P4].

2.3.2.2 Perceptual Approach

The perceptual approach relies on a parametric description of the perceived quality of a room acoustic response. Traditionally this approach is used in most applications and devices where a room acoustic effect is needed instead of a detailed modeling of sound propagation. This is the case for example in sound effects processors that are used in music performances and for music production. In such devices (for example commercial reverberators) many levels of parametrization can be found. For example, the characterization of reverberation may simply be given as the type or the size of a hall (e.g., ”bathroom”, or ”opera hall”) that the resulting response simulates. Such description is a

preset that is then somehow mapped to several parameters controlling the DSP implementation of the impulse response. For more professional users, the effects processors may also contain parameters that are closer to the DSP implementation of the room effect simulator (such as, early reflections delay, or reverberation time). Thus the former example is in a sense a very high level characterization and gives an idea of the type of room effect although providing no details about the structure of the impulse response. The latter, on the other hand, is a low-level description that to an experienced user gives an idea about the audible result of adjusting such parameters. The main idea however in the perceptual approach is, that the room effect is modified by user-controlled parameters, and not by the physical environment (even if also in the perceptual approach the room effect may contain features derived from the physical properties of halls).

In [36], [37] a system is presented where a room acoustic response is controlled via a set of perceptual parameters. They are a result of psychoacoustic experiments where the perceptual influence of the variations of the energy contents of a room impulse response was studied with respect to the different properties of the acoustic space and the sound source in it. The result is a set of 9 mutually orthogonal (independent) parameters, and able to generally describe any reverberant halls (at least, those typically used for music performances). These parameters are:

1. Source presence, describing the energy of the direct sound, and the perception of the proximity of the sound source. The source can be given a distance at which this parameter is valid; moving the source from this distance (e.g., in a virtual reality context), automatically affects the perceived source presence value (reducing it with increasing distance and vice versa) [38].

2. Source warmth that corresponds to the emphasis of the low frequencies in the direct sound,

3. Source brilliance, corresponding to the high frequency emphasis of the direct sound,

4. Room presence indicating the energy of the whole room effect

5. Envelopment describing the ratio of the early reflections and the direct sound,

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

7. Late reverberance indicating the reverberation time during pauses in the sound (and that corresponds to the standardized reverberation time $\boldsymbol{RT}_{60}$),

8. Heavyness that is a factor for the low-frequency reverberation time and

9. Liveness, a factor for the high-frequency reverberation time.

Additionally definition of three DSP filters is added, and they are applied to the direct sound, the diffuse room effect (including the diffuse early reflections and reverberation), or the whole sound (i.e., direct sound *and* room effect). These filters are called the *direct sound filter*, the *omni directivity filter*, and the *input filter*, respectively. The direct filter

simulates an attenuating effect to direct sound when the source is behind an obstacle in a reverberating space (i.e., *occlusion*). The omni directivity filter, on the other hand contains the data about the source directivity averaged over all the angles, and is applied to the diffuse part of the response. Finally, the input filter can be used to simulate an obstructive effect of walls when the source is in another room than the listener.

These perceptual parameters are referred to as ”high-level” parameters, and they are mapped to ”low-level” parameters that define the energies (in three frequency band) of four time-domain sections of an impulse response. The delays of these time-domain sections may be varied (e.g., to simulate rooms of different sizes).

The described perceptual parameter set is used in the Spat$\sim$ program [36], [37], [39], and also as one of the approaches for sound environment modeling in the MPEG-4 scene description language [40]. The relationship between the described perceptual parameterization and the MPEG-4 standard is explained in [P6], and [41], although it is not among the contributions of this thesis. This parametrization is originally made mainly for musicians to allow them to add spatial effects to their compositions. However it is also well suited to virtual reality applications, since the source presence parameter (as described above), enables the simulation of the relative movement of the source with respect to the listener.

2.3.2.3 Example DSP Implementation of Real-Time Auralization

Figures 2.4 and 2.5 illustrate a DSP filter structures that can be used in real-time auralization of an acoustic space defined according to the physical and the perceptual approaches. On the top of both pictures (”SOUND ENVIRONMENT DEFINITION”), the parametrically defined properties are listed, which define room acoustics (that are parameterized differently in the two approaches), and the properties that are common to the two approaches (the source, listener, and propagation medium properties). The physical approach relies on a geometrical room description that includes definitions of the reflecting surface geometry, and the reflectivity and obstruction that each surface causes to the sound. Late reverberation is characterized in terms of reverberation time $\boldsymbol{RT}_{60}$, delay t_2, and reverberation level. The perceptual parametrization in the example in Figure 2.5 follows the description given in the previous section (section 2.3.2.2). The perceptual parameters are converted to frequency-dependent energy contents of the modeled room impulse response through a mapping, presented for example in the MPEG-4 standard [40].

In both figures, the ”DSP IMPLEMENTATION” illustrates filter structures that are typically used in real-time room modeling. In these DSP implementations the early part of the response is formed of separate modeling of the direct sound and early reflections, and the diffuse part of the response is obtained with an IIR filter structure that implements the defined late reverberation properties.

In both approaches the direct sound is obtained by introducing a delay that depends on the distance between the source and the listener, and a given speed of sound in the medium. It is modified by the direct sound filter (H_{Direct}) that implements the effects of the air absorption, distance-dependent attenuation, source directivity (that depends on

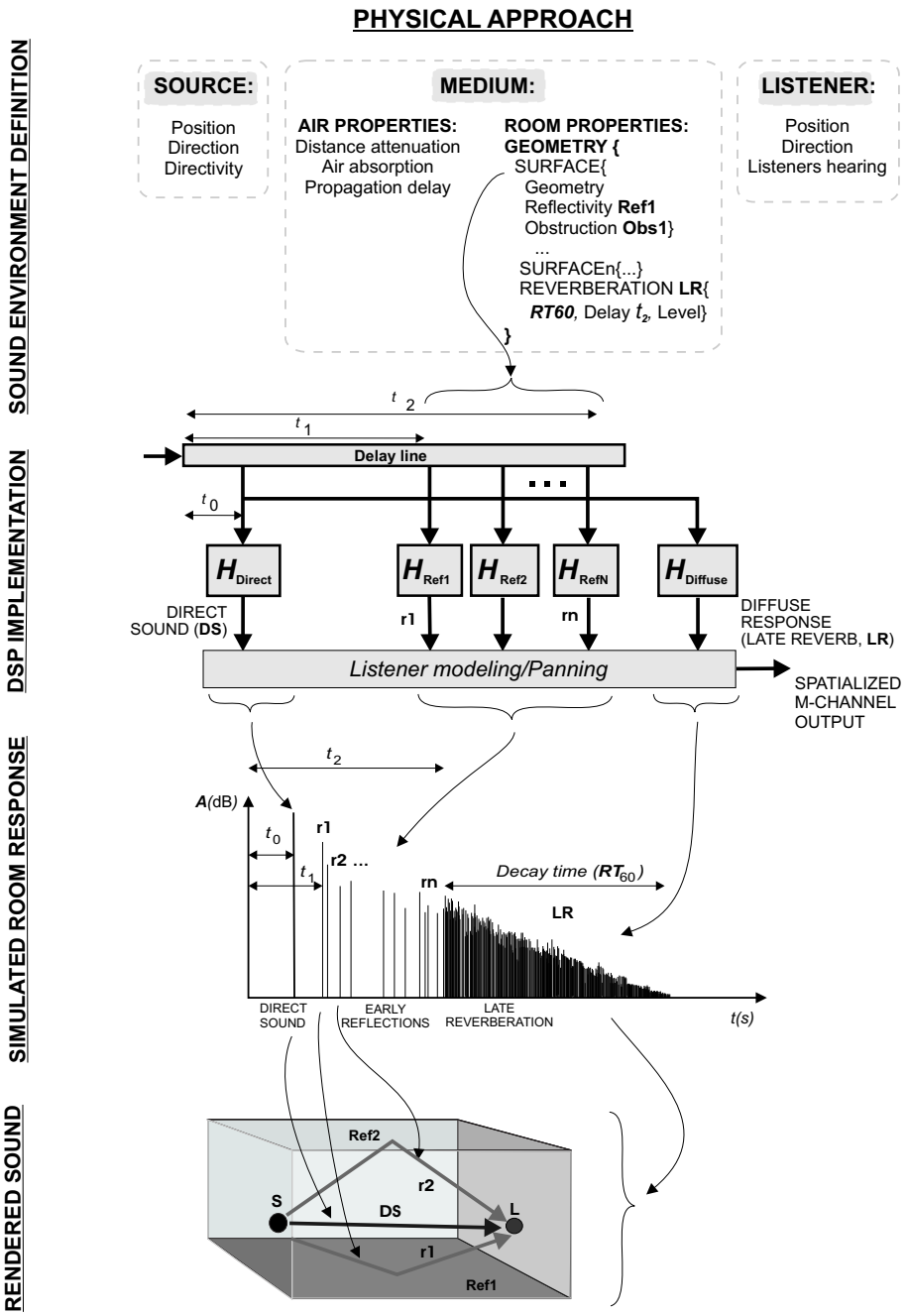

Figure 2.4: An example DSP implementation of RIR in real-time auralization when the physical approach is applied.

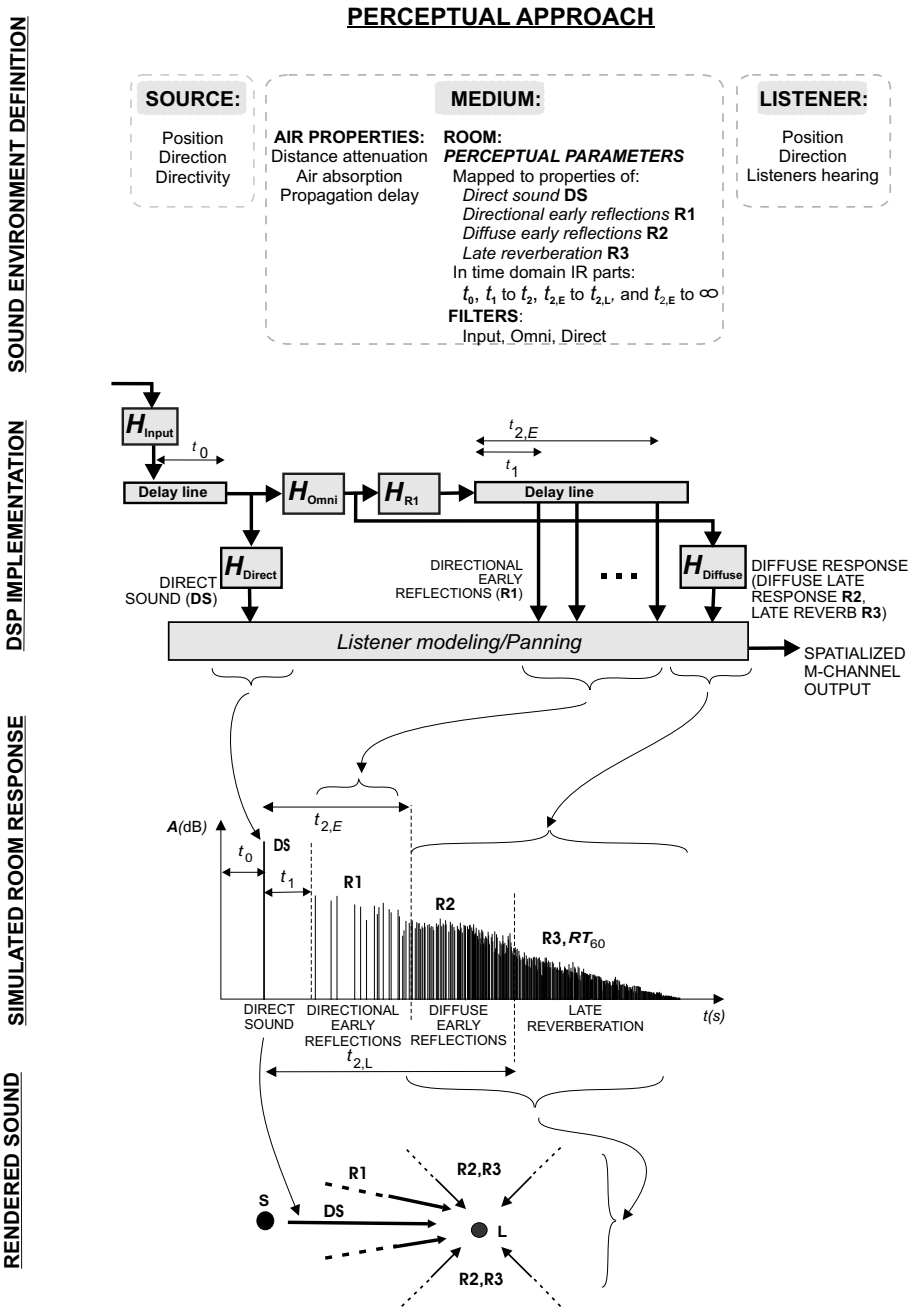

Figure 2.5: An example DSP implementation of RIR in real-time auralization when the perceptual approach is applied.

the current orientation of the source with respect to the listener). It also includes the obstruction effect which in the physical approach is caused by a wall or other surface appearing on the direct sound path, and in the perceptual approach it is defined by the *direct filter*, explained in the previous section.

In the physical approach the delays and reflectivity filters are individually calculated for each early reflection. Each early reflection filter $\boldsymbol{H}_{Ref1}$-$\boldsymbol{H}_{Refn}$ includes a similar distance effect as the direct sound, i.e., the air absorption and distance dependent attenuation effects, and additionally the filtering effect caused by the surface that causes the reflection. The directions of the direct sound and each reflection in this approach are rendered in the panning module, the structure of which depends on the reproduction method used for the playback of the sound. The late reverberation is obtained by feeding the direct sound to the IIR DSP structure that can be controlled by the given statistical parameters ($RT60$, Level, and $t3$).

In the perceptual approach the room effect part is obtained by calculating the energy and frequency contents of the four RIR parts denoted by **DS** (for direct sound), **R1** (directional early reflections), **R2** (diffuse early reflections), and **R3** (late reverberation), from the given perceptual parameters. Thus the filters and delays of the early reflections are not necessary to be calculated separately. Instead, the filters $\boldsymbol{H}_{Ref1}$-$\boldsymbol{H}_{RefN}$ in the physical approach can be replaced with one filter (represented by $\boldsymbol{H}_{R1}$), and the delays can be given a constant, ideally random distribution to avoid coloration of sound. The diffuse part is divided to two parts, where the *diffuse early reflections* (denoted by **R2**) makes the transition smoother from the early reflection part to the diffuse, exponentially decaying late reverberation. The spatial distribution of the *directional early reflections* is close to that of the direct sound, coming symmetrically from both sides of the sound source (simulating the early reflections of concert halls), while the diffuse parts **R2** and **R3** should be evenly distributed in space.

The part ”SIMULATED ROOM RESPONSE” in figures 2.4 and 2.5 shows the respective impulse responses that the presented filter structures produce. Finally at the ”RENDERED SOUND” part of the figures the perceived spatial sound is illustrated.

From the above explanations it can be seen that there are many similarities in the physical and perceptual approaches (when they are used in a virtual reality context), both on the definition and the rendering sides [P4]. In the definition the shared properties include the source and listener related ones (their positions, orientations, and the source directivity), and the medium characteristics (that cause the distance-dependent effects). The major difference in the rendering is the modeling of the (directional) early reflections, and the control of the room impulse response parameters; In the physical approach the filters and delays for each early reflection must be re-computed whenever the listener or the source moves. In a similar situation in the perceptual approach, principally only the gains of the different time-domain parts of the response change (depending only on the changing distance between the source and the listener). Also in the perceptual approach, the user can easily be given the control over adjusting the intuitive perceptual parameters [42], [5].

Of the described implementations, the physical approach has been implemented as a part of this thesis work. Both approaches are included in the MPEG-4 standard, where

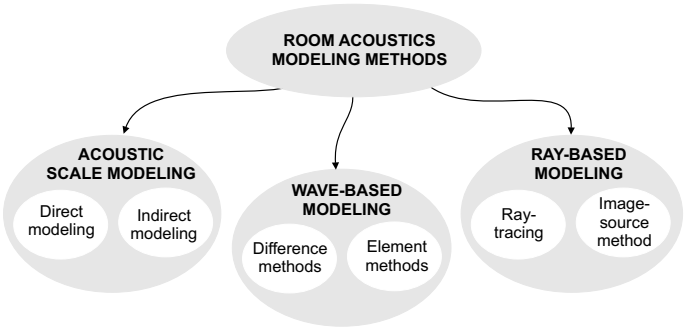

Figure 2.6: Different approaches to room acoustic modeling.

they can be used for creating sound scenes, where some of the sources are rendered according to the physical approach and others according to the perceptual approach. In such a situation it is important that the DSP algorithms are designed and optimized so, that the auralization implementation for each source adds as little processing overhead as possible.

2.3.3 Physical Room Acoustic Modeling Methods

This section overviews methods that exist for room acoustic modeling from geometrical room description. First a brief overview is provided on acoustic scale modeling (although it is not practical for virtual reality applications). Due to the increasing area of applications requiring sound environment modeling, and in many cases low cost, most room acoustic modeling is nowadays performed with the aid of a computer and digital signal processing (DSP). Geometrical room acoustic modeling methods include the *image-source method* and the *ray-tracing*. Both methods deal with sound as rays that represent the sound wavefront propagating in the air, and reflecting off polygon surfaces with defined reflectivity characteristics. Mathematical acoustic modeling methods are based on solving the wave equation in a room, and they include *element methods* and *difference methods*, relying on detailed modeling of sound wave propagation. Figure 2.6 summarizes the different room acoustic modeling approaches.

2.3.3.1 Acoustic Scale Modeling

In the acoustic scale modeling, a down-scaled version of a natural-sized room is built. Room impulse response (RIR) measurements or sound recordings in that space are carried out using ultrasonic signals as sound material, with the wavelength reduced in the same proportion as the dimensions of the room compared to the real one. Correspondingly, in an ideal case all the materials in the down-scaled room should have absorptive and reflective properties that are up-scaled in the frequency-domain in the same proportion as the reduction of the room size. To compensate for the lack of air absorption in the down-scaled models, using dry air or nitrogen is proposed as a fluid to fill up the scale-models

[4]. In the recording situation customized sound source and microphone setups are used for emitting and receiving sounds, and for a listening experience the ultrasonic recording has to be up-scaled to the natural wavelengths.

When acoustic scale modeling is used for the purpose of auralization of sound, a division is made to direct and indirect scale modeling [4]. The former means an approach where ultrasound material is input to the system, recorded, and scaled to natural wavelengths. In the latter approach, a binaural room impulse response (BRIR) is first recorded in the space, up-scaled, and convolved with a monophonic, anechoic sound material. If the recording is carried out binaurally, reduced-sized artificial heads are used in binaural recordings [43], [44], [45]. A system for room acoustic measurements called MIDAS is proposed in [46], and the corresponding impulse response processing explained in [47].

The advantage of acoustic scale modeling is that some acoustic phenomena that in real-time computational modeling often have to be neglected (such as diffusion and edge diffraction), are naturally produced in a scale model. On the other hand, it is not a straightforward task to find materials whose absorptive characteristics at the scaled wavelengths correspond to those at the natural frequencies, or to design down-scaled emitters and receivers.

2.3.3.2 Image-Source Method

In the image-source method, sound reflections are computed by forming specular images of the primary source with respect to reflecting surfaces. Those image-sources are considered as secondary sound sources from which sound rays are emitted with an attenuation that depends on the reflectivity of the surface, and a delay that depends on the distance between the image source and the receiver (listening) point [48], [49], [50], [51]. Also the visibility of each image source is studied to consider whether it is visible from an observed listening point and thus if its needs to be calculated [50]. When this method is used for auralizing sound environments, the direction of each reflection is rendered according to the sound reproduction method used (see next section for different reproduction methods for spatial sound).

The image-source method is practical for virtual reality applications where a room impulse response has to be rendered in real-time and taking into account dynamic changes in the response (caused by the listener or sound source movements). It is used as a basis for computing the early reflections part of a room response, for example, in the DIVA system [52], [5], [6], and in the MPEG-4 reference software implementation [P3].

A drawback of the image-source modeling is that it can not accurately model the wave phenomena because the sound wavefronts in those models are represented by rays. Thus, ignoring phenomena such as diffraction, diffuse reflections, and angle-dependent reflectivity of surfaces, is a simplification that often is made in real-time, dynamic auralization. This may naturally cause audible errors in the modeled room impulse response [15]. The authenticity of room acoustics modeled dynamically with this method has been studied, e.g., in [53], [7]. Improvements for image-source based room acoustic modeling by taking edge diffraction into account are proposed in [54], [55], [56].

2.3.3.3 Ray-tracing

Ray-tracing is another geometrical room acoustic modeling method, where sound rays are emitted from a point source in different directions, and their paths are traced in a reverberating enclosure up to a defined time. The receiving point is represented by a finite volume with a given location, and the sound rays hitting it are registered as reflections characterized by a delay, energy, and direction [57, 58]. In this approach the response is calculated typically in octave bands, and diffusion is taken into account by giving reflective surfaces frequency-dependent sound scattering properties that cause sound rays to be reflected to different directions when they hit the wall.

In [59] the ray-tracing is proposed to be used in combination with the image-source method. Commercial software based on ray tracing exist for architectural acoustic design such as ODEON [60], [61] and CATT [62]. The idea in these programs is usually to calculate frequency-dependent room impulse responses, visualize the ray propagation in 3D scenes, and to calculate room acoustic parameters from the computed responses.

2.3.3.4 Element Methods and Difference Methods

The *Finite Element Method* (FEM) and the *Boundary Element Method* (BEM) can be used to model sound wave propagation in an enclosure with a defined geometry. In FEM, the volume of the room, and in BEM, its boundaries are divided into elements at which the sound pressures are calculated. In [63], two-dimensional FEM is used for obtaining room impulse response, and in [64], 3D FEM is used for studying resonances of a room. The complexity of these methods is proportional to the size of the studied enclosure, and inversely proportional to the size of the (volume or boundary) elements that define the highest possible frequency that can successfully be modeled by these methods.

Difference methods are based on finite-difference approximation of the time and space derivatives of the wave equation [65], [66]. Digital waveguide meshes are examples of computational implementations of difference methods. In room acoustic modeling 2D or 3D meshes have been used to simulate low-frequency sound propagation, see e.g., [67], [68], [6], [69].

2.4 Reproduction of Spatial Sound

In virtual reality context the aim of auralization is typically to model the *binaural hearing* of a human listener within the virtual world environment. In practice this means reproducing signals to both ears of the listener so that the perceptual impression of the sound field is similar as it would be in a natural space that the model describes. To reach this aim, the sound is processed according to the definition of the room geometry (or other parametric representation of the room acoustics), but also according to the positions and orientations of the virtual listener and the sound sources in the virtual room. Thus in real-time and dynamic auralization, the reproduced sound ideally includes the spatial and temporal changes of the sound field of the modeled environment, and the directions of

the incoming sounds at the listener. In practice the early part of a room impulse response is usually modeled by rendering the directions of the direct sound and early reflections in detail, whereas the late part of the response is modeled with less detail, assuming diffuse distribution of the directions of the sound reflections (as explained in the previous section). For both the early and the late part of the response, the chosen reproduction method is one of the factors affecting the signal processing done to the sound (the part that implements the direction of the arrival of the sound and reflections). In the following, different headphone and loudspeaker setups for the reproduction part of auralization are overviewed, briefly explaining how they can be used for rendering the position of sound with respect to the listener.

A division of spatial sound reproduction methods can be done to binaural techniques, i.e., those that model the directional hearing of a listener with the aid of Head-Related Transfer Functions (HRTF), to the ones aiming at producing virtual sound sources with multiple loudspeakers and amplitude panning techniques, and the Wave Field Synthesis (WFS). In the HRTF techniques the aim is to produce an artificial binaural room impulse response (by which the sound is convolved) directly at the ears of the listener. In the amplitude-panning techniques virtual sound sources are formed, which the listener ideally perceives correctly at their given positions. Both are based on rendering of the sound to one listening position at a time (called the sweet-spot in the case of loudspeaker reproduction) [15], [70]. An exception is the WFS where no assumption is made about the listener position, but instead the whole wavefield is rendered inside of a limited listening area with the help of loudspeaker arrays [71].

2.4.1 Head-Related Transfer Functions in Binaural 3D Sound Reproduction

Head-Related Transfer Functions (HRTF) are used in binaural headphone listening and in binaural crosstalk-cancelled loudspeaker reproduction. In both, the aim is to reproduce the (perceptually) same signal at the entrances of the ear canals of the listener, as would be produced by a real sound source placed in a defined position. HRTFs are frequency-domain representations for impulse responses of sound coming from different directions to the ear canal (Head-Related Impulse Response, HRIR). These impulse responses are affected by the reflections off the human pinnae, head, and the torso, and their temporal and spectral structure vary with the direction of the incoming sound [72], [73], [74]. Studies on HRTF filter design have been carried out, e.g., in [75], [8], [76], [77].

2.4.1.1 Binaural Headphone Reproduction

In binaural headphone reproduction, the left and the right channels are filtered with HRTFs that correspond to the current direction of the virtual sound source with respect to the listener orientation. Individual listening experience is produced to the user, and the accuracy and realism of the perceived sound direction depends, for example, on the design method of the DSP filters used in the HRTF modeling and whether the HRTF's are individualized for the particular listener. The realism can be increased, for example, by a head-tracking

system that takes into account the user movements by adjusting the filtering always to the current position and orientation of the listener.

2.4.1.2 Cross-Talk Cancelled Binaural Loudspeaker Reproduction

When using HRTFs in loudspeaker reproduction, usually with two loudspeakers, the sound is correctly rendered only for a small limited area called the sweet-spot. Cross-talk cancelling DSP filters are required to compensate the signals from right speaker to the left ear and vice versa, and similarly as in binaural headphone reproduction, tracking is ideally needed in order to maintain stable virtual source positions. Early studies involving cross-talk cancelled stereophonic loudspeaker reproduction are presented in [78, 79]. Later it has been given the name transaural processing in [80]. In [75] filter design methods for cross-talk cancelled binaural reproduction are presented, in [8] taking also into consideration the use of warped filters [81]. In [76] an extensive overview is made on cross-talk cancelled techniques, and listener tracking is dealt with in that context.

2.4.2 Panning Techniques

Panning techniques in the virtual sound reproduction context are usually methods where virtual sound sources are formed by feeding several loudspeakers with the same monophonic sound signal but with different amplitudes and/or with relative time shifts. The former approach is called *amplitude panning* and the latter *time panning*. In the following, amplitude panning techniques are described for sound source positioning in 2D and 3D loudspeaker configurations. Time panning is based on the fact that sound is perceived shifting towards the loudspeaker from which is emitted first [73] (with a maximum of 1ms delay of the sound from the second speaker). It is not commonly used for positioning of virtual sources (but more for spatial sound effects) since their localization varies as a function of frequency [82], [83].

2.4.2.1 2D Amplitude Panning

In 2D panning, loudspeakers are positioned in a horizontal plane around the listener. The perception of virtual source locations formed by pair-wise panning in a 2D setup has been studied in [84], where it was concluded that stable virtual sources can be formed when the angle between each loudspeaker pair is maximally 60°. The simplest case of 2D amplitude panning is stereophony where two loudspeakers are positioned in 60° aperture with respect to the listener position, and a virtual source can be formed on the axis between these two speakers using the sine or the tangent law, see for example [85], [86].

Recommendations for two 2D multichannel loudspeaker setups are given in [87], namely, the quadraphonic and the 5.1 setups. The quadraphonic setup is a four-speaker layout with uniformly positioned loudspeakers. In the 5.1 setup, on the other hand, the directions are 0, ± 30, and ± 110 degrees with respect to the listening point to place the loudspeakers. They can be considered an extension to the 2-speaker stereophonic setup where the sound source can be positioned between any pair of adjacent speakers by adjusting their relative amplitude gain (although they do not satisfy the previously mentioned

requirement about the maximum of 60° between speakers when creating virtual sound sources, except for the 3 front channels of the 5.1 layout). The use of the 5.1 loudspeaker configuration is nowadays a default setup for home theater systems (e.g., used in Dolby AC-3, also known as Dolby Digital, and DTS Surround sound formats), and its suitability in virtual reality reproduction has been discussed, e.g., in [88]. A recent new step is the 6.1 format, adding one additional rear channel to the 5.1 setup (used in Dolby EX and DTS formats).

2.4.2.2 3D panning

The 3D panning techniques are setups where the speakers are not on a single horizontal plane, but provide possibility also for sound source elevation. Vector Base Amplitude Panning (VBAP) is a 3D panning method where virtual source positions can be created by triplet-wise amplitude panning inside arbitrarily defined triangles of loudspeakers. Thus, phantom sources are formed by three real sound signals at a time, whose gains define the position of the phantom source [89], [83].

2.4.2.3 Ambisonics

Ambisonics is a method for encoding and reproducing spatial sound fields as a combination of their spherical harmonic components. The B-format ambisonics contains three channels in the case of horizontal (pantophonic) sound field encoding and four channels for 3D sound field (periphonic) encoding. In the decoding stage at least four channels for pantophonic system and eight for periphonic system are needed [90]. Ambisonics has been also adopted to reproduce 3D directions of sound sources in virtual reality context [70] and room acoustic modeling [91].

2.4.3 Wave Field Synthesis

In Wave Field Synthesis (WFS) the aim is to reproduce a sound field over a large listening area with the aid of loudspeaker arrays and the principles of acoustic holophony [92]. Therefore, unlike in the methods described above, in WFS the accuracy of rendering is not dependent on the position or orientation of the listener within the specified rendering area. Thus in this method, the problem of the sweet-spot is overcome, allowing multiple listeners to move in the rendering space without any extra computational complexity to the reproduction of the sound scene.

In the holophonic approach to sound scene rendering, spherical and plane waves can be reproduced and controlled in the rendering area. Spherical waves caused by virtual point sources can be imitated by reproducing similar wavefronts with arrays of loudspeakers that act as secondary sources. Following this principle, sources can be placed either on opposite sides of the loudspeaker array (with respect to the listener), or in front of the array (by focusing wavefronts). Each loudspeaker is excited with a source signal that has a delay and an amplitude that will lead to producing a similar wavefront as would be emitted from the virtual source. In this manner the source position can be perceived

correctly anywhere in the rendering area, which is restricted by the size and the shape of the loudspeaker arrays.

In current practical implementations, only a horizontal reproduction of a wave field is realized with linear arrays of loudspeakers. Also, a limited frequency range for correctly rendered sound field is achieved with a small inter-speaker distance (e.g., 12 cm, which leads to approximately 1400 Hz as the highest correctly reproduced frequency) [93]. Distributed Mode Loudspeaker (DML) panels provide a new way of reproducing sound by flat panels that are driven by multiple exciters. They are practical for example in audiovisual applications (virtual reality or movies) as the picture can be projected on the surface of the panels. This further improves and facilitates associating the sound with the visual content. DMLs have recently been studied to be used for WFS reproduction [94], [95].

When WFS is used for virtual acoustic reproduction, different parts of a room impulse response (direct sound, reflections, and late reverberation) can be produced as circular or plane waves. Circular waves represent point sources that are close to the listener, whereas the plane waves are used to represent distant sources and reverberation. For example, in the context of the Carrouso (EU, IST) project [96], a proposition is made to produce the direct sound with circular wave front (representing a point source), and plane waves for reflections and the late reverberation [97]. In [98], an approach to produce virtual sound scenes with Wave Field Decomposition (WFD) of the wave field is discussed. In this approach the whole wave field (produced by a sound source in a room) is measured by recording impulse responses along microphone arrays, and a representation of the wave field is formed with a finite number of plane waves arriving from a finite number of directions and at a given grid of positions.

3. Object-Based Presentation of Sound Scenes

This chapter gives an overview of tools that have been developed for creating sound scenes. In Section 3.1 generic concepts and terms related to sound scene description and sound objects are presented. Section 3.2 explains the sound *Application Programming Interfaces* (API), which are programming tools used to include sound to applications. A distinction is made between low and high-level APIs, and examples of such APIs are given. Finally, in Section 3.3 the MPEG-4 standard is overviewed as an example of a toolset that can be used to form rich multimedia scenes containing various different types of media, with the aid of a scene description language, and audio and visual coding tools.

3.1 Object-Oriented Sound Scene Description Concepts

Object-oriented programming (OOP) is intended for creating applications with a modular and hierarchical structure. Commonly mentioned advantages of OOP as opposed to procedural programming are, that it is easier because of better correspondence between the program components and real-world objects (i.e., that it is closer to human thinking), and that it is faster because of the ability to re-use and evolve existing code. A *class* in OOP provides a template for an object that is an individual instance of a class. In other words, a class defines the implementation of an object. Each class is defined by its data (variables, or fields in some contexts) and the methods (functions) that operate on the data. To use an object, the programmer only needs to know how to use their *public methods* that define their *interface*, i.e., the access to modify the object, which makes them easier and simpler for the programmer to use. Inter-relationships between objects can be created by passing messages from one object to another. Furthermore, the re-use of code is guaranteed by *inheritance*, where subclasses are derived from their superclasses, without the need to re-write the code, but being able to define individual characteristics for the derived classes [99]. Among the most popular object-oriented programming languages are C++ [100] and Java [101], [102].

A *scene* in this context means a data presentation on a computer, including possibly various audio and visual elements. Usually these application elements are considered as *data objects*. The advantages of object-based presentation of audiovisual data are the same as in object-oriented programming in general; Fast development of applications,

where modifications can be carried out on individual properties of the audio and visual data, either by other application components or interactively by the user. This interactive modification of a scene may include changing the data variables of an object, moving it within the coordinate system of the rendering device (e.g., a computer screen), and interactive playing (starting or stopping) of sound and video. Creating a scene out of such data objects may also contain grouping or mixing of several objects to form more complex, higher-level graphical or sound objects, or their combinations. Furthermore, a *scene description* defines a scene. In other words, it is the code or textual description that contains the information about data compositing, i.e., which data is present in the scene, and how it is organized in the application presented to the user. This organization includes, for example, the positioning of data objects in the rendering coordinate system, their time-dependent behavior, and user interactivity.

According to the definition provided in [103], an application programming interface is a specification of how a programmer accesses the behavior and state of classes and objects. Several APIs exist for creating graphical and/or sound scenes. They are often library extensions of programming languages that enable object-oriented creation of audiovisual scenes. Furthermore, in the present work, *scene description language* is a term used for a high-level API that offers easy-to-use tools for highly hierarchical and object-oriented composition of scenes. In such an API, the scene description is defined as a *scene graph*, where the objects (usually called nodes) can be linked together to form a hierarchical (e.g., tree-like) structure. In the scene graph, the lowest-level objects are typically the ones rendered to the user (e.g., sound, video, or graphical elements). They inherit properties from their *parent* objects that can be used to group, position, and resize their children. Usually the scene description languages also support including interactive and other dynamic events in scenes. See, Figure 3.1 for an example of a scene graph.

An example of a library extension API for developing audiovisual scenes is the DirectX for C++ [104], where 2D and 3D graphics and sound objects can be used to create multimedia applications. Another example is the OpenGL, a cross-platform C and C++ API for 3D graphics [105]. Java3D [106] is also a cross-platform API and an extension to the Java programming language. It offers support for highly object-oriented programming of interactive 3D graphics applications. Examples of scene description languages, on the other hand, are the Virtual Reality Modeling Language (VRML97) [107], [101] (and its successor X3D [108]), and the MPEG-4 Binary Format For Scenes (BIFS) [109]. Both enable fast development of platform-independent audiovisual applications and virtual worlds with little expertise knowledge about programming. More details about high and low-level APIs are presented in the next section.

The scenes can be audiovisual (including both graphics and sound elements), visual-only, or audio-only. Another classification of scenes can be made to two-dimensional (2D) and three-dimensional (3D) scenes. An example of a 2D scene is a multimedia presentation (including, e.g., pictures, video, and associated sound), or interactive TV application with downloadable programs. 3D applications, on the other hand, often contain a virtual room or other virtual environment, with 3D visual and sound objects. A virtual viewing point is often added that can also be used for navigating in a 3D virtual world. 3D scenes can also be used for telepresence (e.g., in teleconferencing) where users in remote

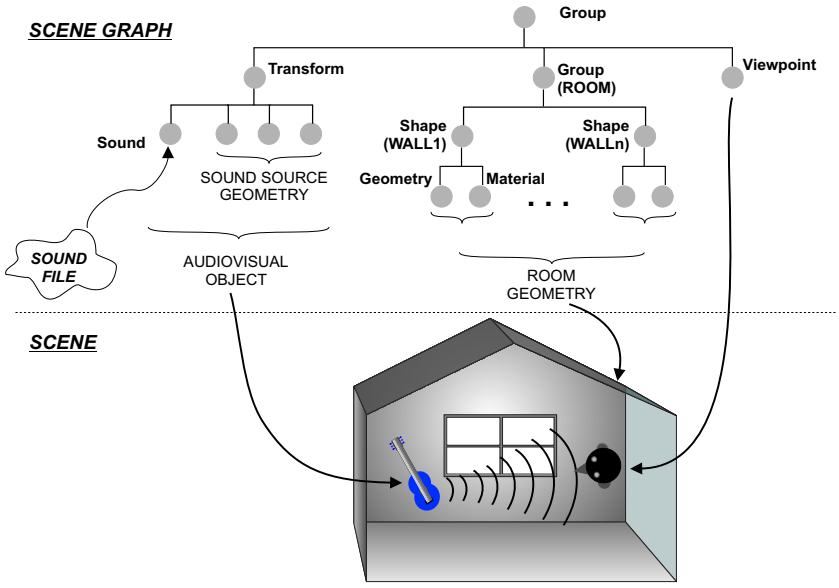

Figure 3.1: A scene graph that defines a 3D scene composited of graphics and sound nodes. The syntax in this example resembles the VRML scene description. The topmost Group node binds together the rest of the hierarchical scene, which includes both graphical and sound objects. The sound file in this example provides the actual audio content, which is attached to the scene with the help of the Sound node. The Transform node groups and positions the Sound node and a set of graphical nodes to form an audiovisual sound object. The Viewpoint node defines the position from which the virtual world is viewed and heard.

locations are displayed and auralized to each other inside a virtual room, where all the participants have their own positions. In such an application it may be particularly important to be able to associate the sound of each speaker spatially to a correct position (maybe with natural-sounding room effect added). 3D audio-only scenes also form an important set of applications, which provide tools for spatial sound compositing for musicians and sound engineers.

Finally, the *sound scene description* defines the compositing of sound into a sound scene, and can be approached from two different viewpoints, namely, the *virtual reality compositing* and *content compositing*. The former refers to the process used for spatial presentation of sound. In this approach the scene is created using objects that define the spatial properties of the sound source, the propagation medium, and the receiver (explained in Section 2.1). The content compositing, on the other hand, involves operations for adding effects and filtering to individual sound streams, or for forming mixed sound tracks. Thus the scene description objects that define and perform sound content compositing, may include different mixing and sound filtering functionalities. These two sound compositing frameworks can be applied both in audio-only and in audiovisual applications. A good example of a versatile and flexible sound scene description toolset is

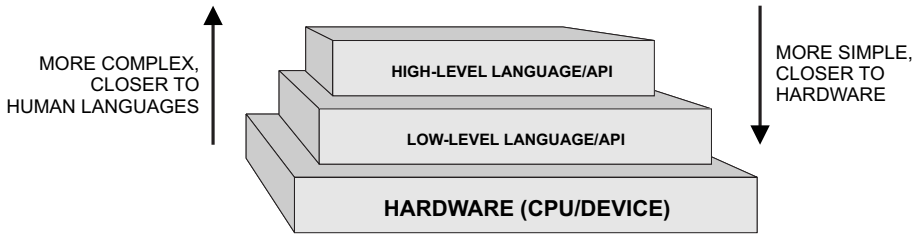

Figure 3.2: The characterization of low and high-level programming languages and APIs.

included in the MPEG-4 standard, explained in publications [P2] and [P6] of this thesis, and in the MPEG-4 standard editions [3] and [40].

3.2 Sound Scene Description Application Programming Interfaces

This section discusses the APIs that are used to aid a programmer or a content producer to create sound scenes, or add sound to (principally) graphical applications. First the properties of low and high-level sound scene APIs are explained, and after that existing sound APIs and specifications are briefly overviewed.

3.2.1 Low Level and High-Level APIs

Figure 3.2 illustrates the differences between the low-level and high-level programming languages. The low-level programming languages operate closer to the hardware (e.g., a CPU of the computer) than the high-level ones. At the very lowest level are machine languages (above which are the assembly languages) that directly communicates with hardware. The high-level programming languages, on the other hand, contain more complex functionalities and are closer to human languages (and therefore easier to use). They are often platform-independent (such a C, C++, or Java), but naturally require a mapping to the low-level control of the hardware.

In the context of audio or graphics APIs, a similar classification can be made; The low-level APIs operate more directly with the hardware (such as the DirectX API with the sound and the video cards), whereas the high-level and platform-independent ones (such as Java3D or VRML) require a lower-level language or an API for mapping it to the hardware control. The advantage of a low-level API is usually a smaller latency compared to high-level ones. Traditionally in low-level APIs only simple audio I/O functionalities are provided, in which case the sound samples are read from a file or a sound stream, and played in the application without further processing. In this case the sounds are often considered rather as ambient (i.e., not having any spatial character when they are reproduced), than as spatial sound source objects (with a position and associated acoustic effect). Recent specification developments bring improvements to the lack of spatial sound processing support in low-level APIs, as will be shown in Section 3.2.2. Without

these improvements, when spatial effects are needed and none exist in the API definition itself they have to be implemented by the programmer at an audio DSP level.

The two main purposes of high-level APIs are to provide platform-independent and easy (usually object-oriented) programming of structured sound objects, where the spatial characteristics are ideally built-in properties of the API definition. Preferably, in this approach, the creation of such spatial sound objects can be controlled via intuitive parameters, the values of which define the individual characteristics of each sound source. Thus the programmer does not need detailed understanding about the DSP implementation of the acoustic effects, since according to the object-oriented programming principles their implementations are hidden inside the objects. In this approach the application (a sound or multimedia scene) is often written as a hierarchical scene graph, explained in the previous section. The way the objects are organized in this scene graph, as well as the linking and the interaction among them and from the user, define the spatial configuration and dynamic behavior of the scene.

Until recently, in most programming languages and scene description API's, the sound functionalities have been included only to support the visual part of an application, and no sophisticated tools for sound compositing (as presented in 3.1) were available. Even when such features were present, they only enabled adding simple effects such as coarse modeling of spatial sound (e.g., positional sound source with simple directivity and distance attenuation characteristics, included for example in the earlier versions of DirectSound3D and VRML97 specifications). Typically the sound has been obtained from pre-recorded audio files, without the possibility to real-time sound communication. Improvements are provided by several recent specification developments. This is due to increasing computational capacity of sound cards and computers, and the spreading knowledge of spatial sound modeling. Among the advanced APIs are the Java3D API of the Java programming language, OpenAL library development that is done in co-operation between several industrial partners, the latest version of DirectSound (in DirectX 9.0), the EAX extensions to DirectSound, and the MPEG-4 scene description language.

Table 3.1 summarizes the features that in this work are considered as characteristic features of low-level and high-level sound APIs. It is obvious that this division is not strict in the case of each existing sound API. However it gives an idea of what is meant by this classification in the present work, where a high-level API was developed for creating spatial sound scenes. In the next subsection existing APIs for spatial sound programming are discussed.

FEATURE	LOW-LEVEL SOUND API	HIGH-LEVEL SOUND API
Difficulty of programming	Requires expertise on a programming language, (possibly platform-dependent) API, and sometimes on DSP implementation of needed algorithm.	Easy, intuitive, highly parametric and object-oriented.
Platform and hardware dependency	Can be high, except in cases where sound I/O APIs are written for each platform separately	Same applications run on many platforms (which is transparent to the programmer)
Structure of program	Low-level hierarchy	Highly hierarchical (structured, object-oriented)
Amount of code (needed for a specific application or a spatial effect)	High	Low because of highly parameterized objects
Control of scene rendering details	High when the programmer implements the spatial sound algorithms.	Low, as the API defines only the interface, i.e., does not give access to the DSP algorithms.
Standardized functionalities	Often not, as the low-level APIs are usually developed for a specific platform or sound card.	Often yes, as it is needed for predicting the rendering result. This gives usually flexibility to the exact implementation, but defines acceptable margins for the rendering result of each effect.
Versatility of spatial sound features	Usually low	Usually high
Where is the 3D sound programming knowledge needed?	The application programmer must have knowledge on implementing the spatial sound algorithms.	Effort is required mostly at the stage of the specification development, and in implementing the rendering software.
Examples	DirectSound in DirectX	OpenAL, MPEG-4 BIFS, Java3D

Table 3.1: Summary of properties that characterize the low and high-level sound APIs.

3.2.2 3D Sound APIs

The following section provides a brief explanation on the status of existing 3D sound APIs. First the *Interactive 3D Audio Rendering Guidelines*, produced by the 3D working group of the *Interactive Audio Special Interest Group* (IASIG), are presented. These guidelines give recommendations for minimum requirements regarding 3D sound APIs. After that the different APIs that include at least some spatial sound functionalities (summarized in Table 3.2), are overviewed.

API	SPATIAL SOUND CHARACTERISTICS
DirectX 9.0	Library extension for C++ programming language by Microsoft, for creating multimedia applications. The sound part enables positional sound sources with cone-like directivity, distance-dependent attenuation, Doppler effect, and I3DL2 compatible reverberation.
EAX 4.0	Sound API from Creative Technology. EAX extensions exist for DirectX and OpenAL.
OpenAL	Open source audio library. Developed in cooperation between different industrial partners.
VRML97	ISO Standard. Simple positional, directive sound source model. Frequency independent directivity and distance attenuation with elliptical attenuation regions.
Java3D	3D extension to Java. Positional, directive sources with frequency-dependent directivity and distance attenuation, Doppler effect, and reverberation. Forthcoming version (currently beta version 1.3) includes the set of I3DL2 spatial sound functionalities (obstruction, occlusion, reflections, reverberation).
MPEG-4	BInary Format for Scenes (BIFS). Positional sound source model with frequency dependent distance attenuation and directivity, and propagation delay. Includes both physical and perceptual characterization of room acoustics.

Table 3.2: Summary of spatial sound characteristics in different sound application programming interfaces.

3.2.2.1 Interactive Audio Special Interest Group (IASIG)

Interactive Audio Special Interest Group (IASIG) has been established to share ideas about interactive audio [110]. IASIG is organized in several working groups dealing with different aspects of sound related tasks in the areas of, for example, internet audio, synthetic audio, audio composition, audio effects processing, and spatial sound. The 3D Working Group (3DWG) of IASIG has produced a set of guidelines to assist and improve the development of 3D sound capabilities (programming and rendering) in computers. Two documents have been published: the Level 1 and Level 2 3D Audio Evaluation Guidelines (denoted by I3DL1 and I3DL2, respectively). The Level 1 guidelines provided minimum acceptable requirements for 3D sound rendering, and a lexicon for understanding the terms used to describe 3D sound related concepts in different contexts [111]. Concerning the spatial sound, it includes the source position rendering relative to a given listening point. The Level 2 guidelines, on the other hand, define a more advanced set of 3D audio features that are needed to create interactive 3D sound environments and spatial sound effects (including room acoustic modeling) [112]. Following gives a short description of the recommended spatial sound properties that are included in an API conforming to the Level 2 (I3D2L) guidelines:

Reverberation model in the IASIG Level 2 guidelines follows the commonly accepted division of a room impulse response to the direct sound, early reflections and late reverberation (as explained in the previous chapter). The energies and relative delays of the different parts of the response are controlled independently. Filtering can

be added to decrease the energy of the high frequencies of the direct sound and the room effect. The late reverberation is controlled by a reverberation time parameter, which defines the time of the 60 dB attenuation of exponentially decaying response. In addition, the late reverberation is characterized by a diffusion factor and a modal density (both relative quantities), which control the density of reflections in the time domain, and the modal density in the frequency domain, respectively.

Source-receiver distance rendering is characterized by a roll-off factor that defines the speed of attenuation as a function of distance from a source. Automatic attenuation of the reflected sound as a function of distance should also be rendered.

Occlusion and obstruction definitions are intended for producing an effect that causes the listener to perceive the source as if it is behind a sound-obstructing obstacle or in another room, respectively. Typically these phenomena cause a frequency dependent (lowpass) filtering to sound. Both effects are characterized by lowpass filters that are controlled in terms of global attenuation, and relative high-frequency attenuation (with a given frequency limit for the high frequencies).

3.2.2.2 DirectSound in Microsoft DirectX

Microsoft DirectX is a C++ API for creating multimedia applications on Windows platforms [104]. DirectSound is a part of DirectX, which enables adding spatial sound features to applications. The 3D sound rendering model of DirectSound is illustrated in Figure 3.3. The sound source can be defined by its position, orientation, and directivity. The directivity is characterized by two cones (defined by their angles), and different sound volumes inside of the inner cone and outside of the outer cone. When the listener is inside of the inner cone (e.g., Listener$_i$ in Figure 3.3), the sound level is equal to a relative value of 0 dB. Outside of the outer cone (for Listener$_o$) the sound level is given a negative value (Vol$_o$ as dB), indicating a decrease of the volume with respect to that inside of the inner cone. In the transition region (for Listener$_t$) between the cones the sound level decreases to Vol$_o$ with an increasing angle.

Also the listener in this model has a position and an orientation. Thus the perceived spatial impression of the sound source depends on the relative position and orientation of the source with respect to the listener. In addition to the attenuation caused by the directivity definitions (explained above), the source – receiver distance defines additional attenuation. The attenuation curve as a function of this distance is defined by the minimum and maximum distances (decided by the programmer), between which the sound by default decreases 6 dB when the distance is doubled (starting from the minimum distance), and outside of the maximum distance the sound level is either constant or inaudible. In addition to the described spatial sound rendering model, the latest version of DirectX (v. 9.0) includes a reverberation model that follows the Interactive 3D Audio Rendering Level 2.0 Guidelines (I3DL2) [113].

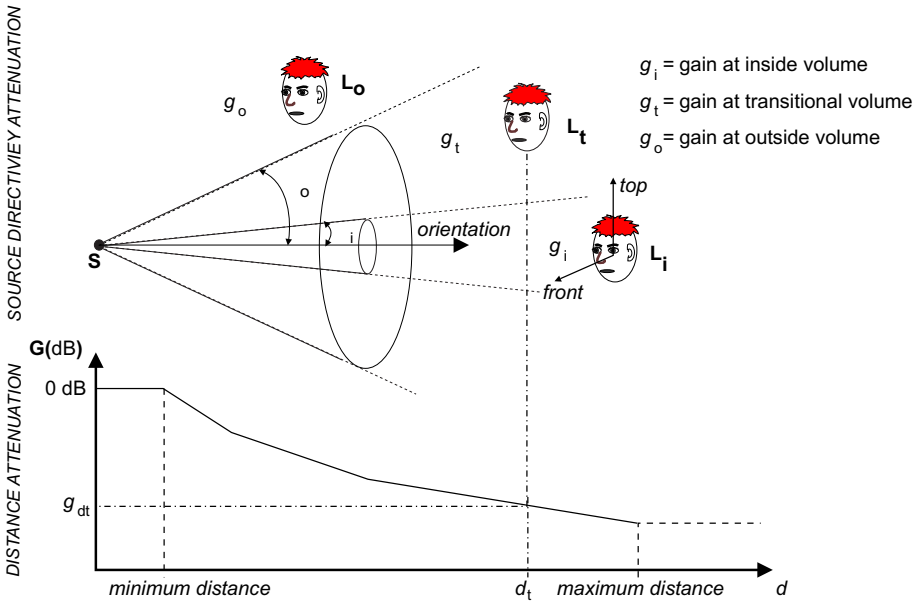

Figure 3.3: The 3D sound and listener models in DirectSound.

3.2.2.3 Environmental Audio Extensions: EAX

The EAX (currently version 4.0) API from Creative Technology that can be used as an extension to the DirectSound and OpenAL API, and it conforms to the complete set of 3D audio rendering guidelines given in I3DL2 [114]. Thus applications can be created including the modeling of positional sound sources, distance rendering, reverberation, occlusion, and obstruction. Additionally, in sound reflection clusters and individual reflections can be added as environmental sound effects. Also continuous transitions between acoustic spaces (rooms) can be carried out.

3.2.2.4 OpenAL

OpenAL is a cross-platform 3D sound API developed for enabling more realism in virtual environments (for example, in computer games) by adding positional, dynamic sound sources [115]. The OpenAL specification defines a set of minimum requirements for a conforming API implementation. The requirements include a minimum number of rendered sources, distance rendering (attenuation), and Doppler effect rendering. It contains an extensible, open-source implementation of the API, and test programs to evaluate specific implementations. In OpenAL, the following spatial properties can be defined for the sound source and the listener:

Sound source definition includes a description about the position, directivity (similarly as in the DirectX), direction, and the velocity of the source.

Listener definition contains the position, orientation, and the velocity of the listener. The effect of each source in the environment is computed with respect to this listening position.

The actual sound content emitted by the spatial sound sources is provided through sound buffers that can be used for associating streamed sound, or sound read from a file, with the source. They have a specified size and contain a pointer to the buffered sound data. The buffers also hold information about the audio format including the sampling frequency, bit depth, and number of channels. The distance dependent attenuation and doppler effect rendering follow the one given in I3DL2 guidelines, but no other environmental effects of I3DL2 (reverberation, occlusion, obstruction) are currently required.

3.2.2.5 Virtual Reality Modeling Language (VRML)

As the first standardized specification for creating interactive virtual reality applications, the VRML97 (also known as VRML 2.0) introduced a simple model for adding spatial sound sources to virtual worlds [107], [101]. A VRML scene is defined in a form of a scene graph (explained in Section 3.1), where nodes are the functional objects that are hierarchically linked and that the scene is composed of. The sound functionalities in the VRML standard allow spatial positioning of a source (defined by a node called Sound) by giving it a location and a direction. A distance dependent attenuation region is defined by two ellipsoids illustrated in Figure 3.4. The values of the minFront and minBack fields define an ellipse inside of which the sound volume is uniform. Between this inner ellipse and the outer ellipse (defined by the maxFront and maxBack ellipses) the sound attenuates to -20 dB, and outside of the outer ellipse the sound is not heard. Thus this parameterization enables forming of spatially positioned sources with a simple directivity pattern.

The VRML enables associating of a downloadable sound file with a spatial sound source object. Thus it is not suitable for real-time transmission or streaming of sounds. The forthcoming X3D (Extensible 3D) standard is a successor of VRML. Although it contains several extensions to the VRML specification (e.g., multi-user worlds, and body animation), it does not bring any new spatial sound rendering functionalities.

3.2.2.6 Java3D

Java3D API is an extension to the Java programming language to create animated, interactive 3D applications [106]. In a similar way as a VRML scene, a Java3D application is programmed in a form of a scene graph. However, its spatial sound properties are more advanced enabling reverberation, air absorption, and frequency dependent directivity definitions. The forthcoming (currently beta) version, Java3D 1.3, is conformant to the I3DL2 guidelines (explained in Section 3.2.2.1).

The directivity of a sound source in Java3D can be defined in quite a similar fashion as in DirectX: the source can be given a position and a direction, cones with different opening angles are defined together with their associated gains. However, in Java3D, any number of cones can be defined for a single source, and the directivity gains can be

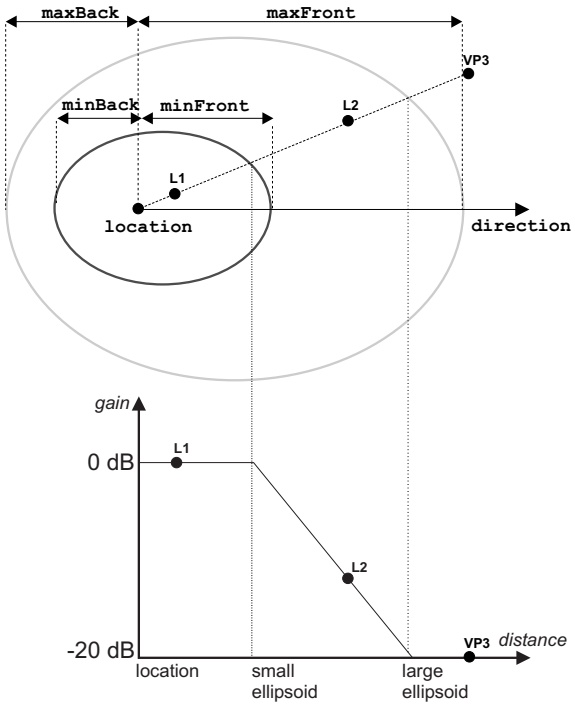

Figure 3.4: The audibility and distance attenuation of the VRML sound source.

made frequency dependent (by giving a cutoff frequency that defines a lowpass filter). In addition to this directivity definition, elliptical distance attenuation regions can be set that resemble those of the VRML Sound node (see, Figure 3.5). The distance dependent attenuation can also be made frequency dependent, for simulating the increasing lowpass filtering as a function of distance [116].

3.2.2.7 MPEG-4 BInary Format for Scenes

The BIFS scene description language in the MPEG-4 standard (see, Section 3.3) is used for data compositing and for creating virtual worlds. A superset of the VRML nodes is defined in BIFS, and its most important extensions concerning sound are the audio nodes. In the first edition of the standard [3], a set of nodes were included for advanced mixing and effects processing of sound streams. These nodes are referred to as the AudioBIFS, explained thoroughly in Publication [P2] and in [117]. The second edition of MPEG-4 [40] contains an extended set of audio nodes for more advanced spatial sound presentation (referred to as the Advanced AudioBIFS, presented in Publication [P6]). Advanced AudioBIFS enables sound environment modeling according to both physical and perceptual approaches (explained in Section 2.3.2).

The following explains the supported sound scene properties in the physical modeling

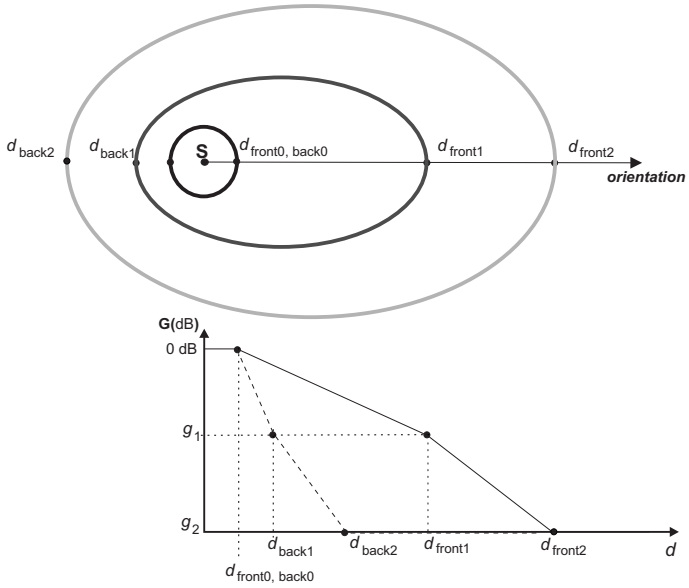

Figure 3.5: The sound attenuation model in Java3D. In addition, sound cones resembling those in MPEG-4 and DirectX can be used for directivity definition.

context. In the perceptual approach, the set of parameters described in Section 2.3.2.2 are available for defining a room acoustic effect. When a source is associated with perceptual parameters (with the aid of a node called PerceptualParameters, which is linked to a DirectiveSound node that is an object used for representing 3D sound sources in MPEG-4 scenes) the physical acoustic characteristics of the environment are ignored for that source.

Source is defined by a BIFS node called DirectiveSound. The spatial source properties are the position, direction, and directivity. The directivity of an MPEG-4 sound source is illustrated in Figure 3.6. For each given direction (with respect to the main direction of the source), a frequency dependent gain can be given that at the rendering stage is implemented as a digital filter. The gain can be given either in terms of filter coefficients defining a transfer function $H(z)$ of an IIR digital filter, or as a piecewise linear approximation of a magnitude response $|H(z)|$, which is given as a discrete set of frequency – gain pairs.

Medium properties include the effects caused by the sound propagation in the air, and room-related effects. The former include the distance-dependent attenuation, air absorption, and the speed of sound in the medium. They are associated for each source object (DirectiveSound node) separately, and their rendering can also be disabled for the sources for which they are not needed. The speed of sound as well as the distance dependent

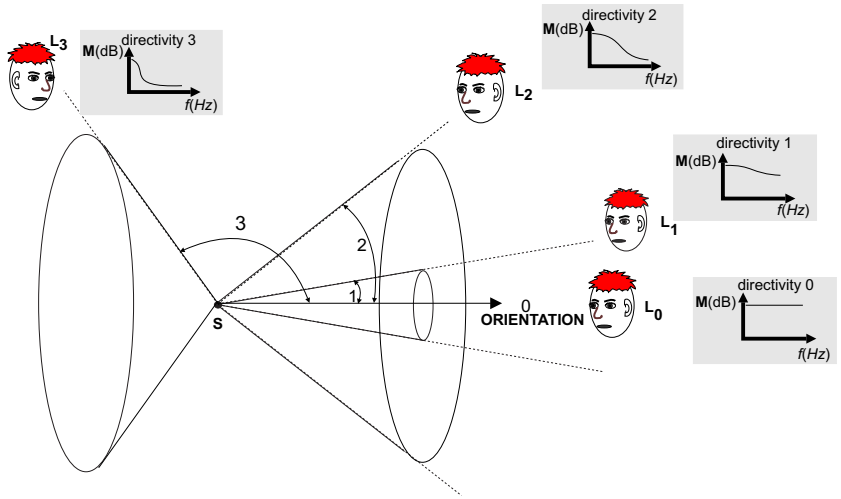

Figure 3.6: The 3D sound directivity in the MPEG-4 BIFS.

attenuation are adjustable quantities in the scene (individually for each source). Their default values correspond to those that the sound would naturally have in the air and in free-field conditions.

The room-related properties are the reflectivity and transmission of surfaces, which can be used as building elements for rooms, or for creating discrete reflections or echoes. The reflectivity expresses the portion of the sound that is reflected off the surface, and the transmission function defines how much of the sound passes through it. The reflectivity and transmission of a surface are expressed similarly as the directivity filtering of a source. They can be pure gains (indicating frequency-independent reflectivity), filter coefficients, or alternatively magnitude response approximations of IIR digital filters. They are given in material properties of a geometry object that defines a polygon (a flat surface with arbitrary number of vertices). The BIFS nodes needed for such surface definitions are called IndexedFaceSet and AcousticMaterial, the former defining the geometry, and the latter the visual properties, such as the color or the texture of the surface, and simultaneously the explained acoustic properties (see, Figure 3.7).

Another room related property is the late reverberation, which can be added to enhance the room acoustic effect by associating it with spaces where early reflections are caused by reflective surfaces. It can also be used to create a simple reverberation effect, without the presence of early reflections. The late reverberation is characterized by a frequency-dependent reverberation time, delay of the reverberation (with respect to the direct sound), and the level of the reverberation. Both the reverberation, and the effects caused by obstructing and reflecting surfaces, are applied to a source when it is in a defined 3D region in a scene, at the same time with the virtual listening point. This region is defined with the help of the AcousticScene node, which is also used for grouping surfaces under a single auralization process. Figure 3.7 illustrates this grouping. A room geometry

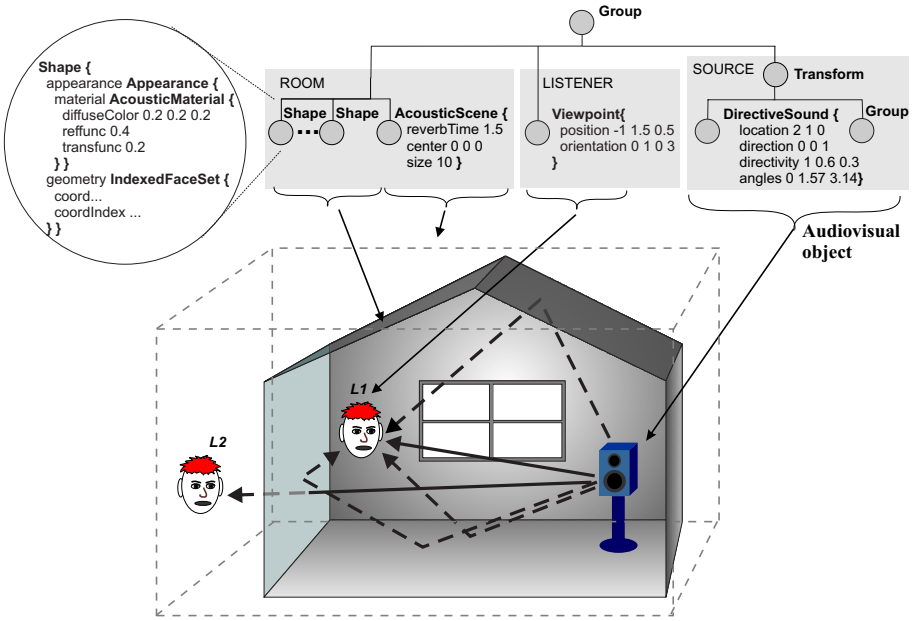

Figure 3.7: An example of an Advanced AudioBIFS scene when applying the physical modeling approach. The visual room is composited of IndexedFaceSet nodes, that each define the geometry of a polygon representing a surface. Visual and acoustic properties for each surface are given by an AcousticMaterial node that besides visual properties, also defines the reflectivity of the surface, and the transmission of sound through it. AcousticScene node defines an audibility region, inside of which both the source and the listener must be in order to hear the sound. It can also be used to define common late reverberation properties for the same region. Thus when the listener is in the room (denoted by **L1**), the direct sound, reflections (filtered with the *reffunc* fields of the corresponding reflective surfaces), and the reverberation are heard. Outside of the room (at the listening point **L2**, at a position which is still in the rendering region defined by the *size* and *center* fields of the AcousticScene), the sound is filtered by the gain defined in the *transfunc* field of the AcousticMaterial.

is given under the same Group node as the AcousticScene node. AcousticScene sets the boundaries (dotted line box in Figure 3.7) inside of which the sound is heard, and the reverberation, reflectivity, and transmission of surfaces applied. Several such AcousticScene nodes can be defined in a single BIFS scene, each of which may contain a different room geometry. The 3D regions can be spatially separated to avoid rendering of sources from different rooms simultaneously (see, publications [P3], [P6]).

The Listener model in MPEG-4 BIFS is included in nodes called Viewpoint and ListeningPoint. The former is meant for audiovisual scene viewing and navigation, and the latter for audio-only scenes, or for adding a virtual listening point that is different from the visual viewpoint in an audiovisual scene. These nodes contain the information about

the position and orientation of the listener. The rendering of the directions of sounds (and the directional reflections) is not a normative part of the MPEG-4 standard, therefore no detailed information about the spatial hearing of the listener, such as HRTF filter data, is included in the scene. All the sound rendering is performed with respect to the listening point. Thus the rendered response depends on the positions and orientations of the listener, the sound sources, and the reflective and obstructive surfaces.

Advantages of the Proposed Framework The described parametrization of the physical sound environment modeling is one of the main results of this thesis. Among its advantages is the efficient and object-oriented programmability of sound scenes, meaning that versatile sound environments can be created with a small but descriptive set of parameters. The similarities between VRML and MPEG-4 allow adaptation of VRML scenes to MPEG-4 coding scheme with little modifications, and VRML content (where the emphasis is typically on the visual side) can be enhanced by adding acoustic properties to the scenes. The VRML resemblance also brings about the advantage, that the technology included in existing VRML authoring tools could be used for creating 3D audiovisual MPEG-4 scenes with advanced acoustic properties. Also in some room acoustic modeling software, the geometrical definition of acoustic spaces contains similar type of descriptions about the surface geometries and their acoustic properties. Conversion tools for creating MPEG-4 scenes from these spaces would be simple to make. The authoring of MPEG-4 sound scenes is addressed in [P7]. In that context, only the authoring of sound scenes according to the perceptual approach is discussed. However, the MPEG-4 sound scene authoring explained in [P7] is a part of a more generic 3D sound scene authoring scheme where there are ongoing developments for extending it also to the physical approach [118].

3.3 MPEG-4 Framework

MPEG-4 is a standardized specification of the Moving Pictures Experts Group, which is a subcommittee of the International Standardization Organization (ISO) [3]. It is an example of a modern specification, which includes various new technologies for object-based encoding and presentation of digital media [119], [120], [121]. The MPEG-4 standard contains coding (compression) methods for digital video and audio (similarly as the previous MPEG standards MPEG-1 [122] and MPEG-2 [123]), but also coding methods for synthetic data, meaning audio or visual content that is generated by a computer from parametric descriptions. It has also various mechanisms for enabling user interactivity.

The core components of MPEG-4 for making interactive audiovisual scenes, are the scene description tools (in BIFS, [124], [109] defined in the MPEG-4 systems part [125], [126]), and the audio and the visual coding tools (defined in the audio [127], [128], [41], [129] and visual [130], [131], [132] parts of MPEG-4, respectively). The use of these parts for creating an MPEG-4 application is illustrated in Figure 3.8. The main idea is, that the actual content (the coded audio and/or visual data) is kept separately from the scene description that defines the compositing of that data (the spatio-temporal organization of

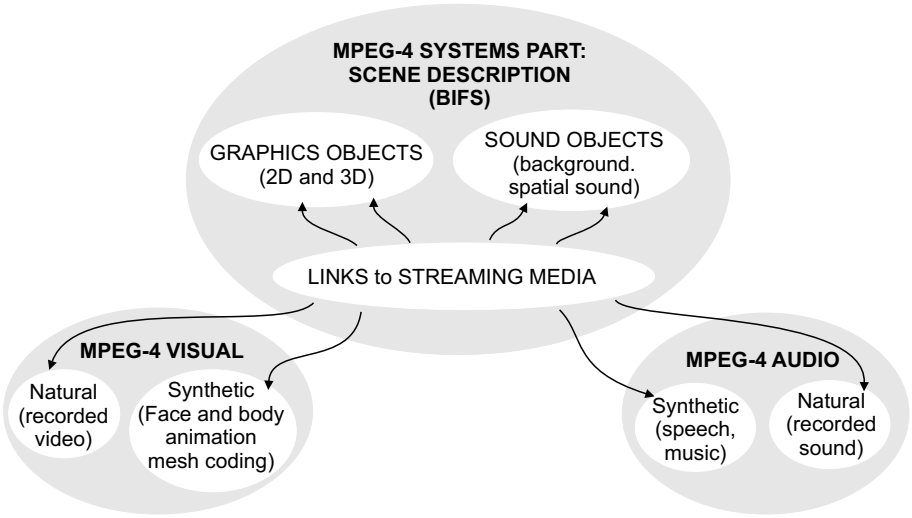

Figure 3.8: Principal components involved in MPEG-4 scenes. MPEG-4 Systems contains the descriptions of the graphical and spatial sound objects, and the actual audio and visual streams may be linked to the scene, which is the actual application that is presented to the user with the needed output devices.

the audio and visual streams). Furthermore the scene description enables a large number of interaction functionalities, that enable including user interactivity as a part of the scene description. This user interaction aspect is dealt with in publication [P7].

All the data included in an MPEG-4 application are considered as media objects. In this scheme, each sound stream (a recorded sound source, for example), or recorded video, is a streaming object. They may be handled individually at the decoding process by placing each of them in a 2D or a 3D coordinate system, and modifying their spatial properties (such as a size or position) dynamically and interactively with the help of the BIFS scene description tools of the MPEG-4 Systems. Thus two main tasks of BIFS can be pointed out: one is to composite different audio and visual streams, and the other is to create audiovisual scenes and virtual worlds. These two approaches can naturally be combined in a single application. Figure 3.9 illustrates the linking of a sound stream to the MPEG-4 scene description.

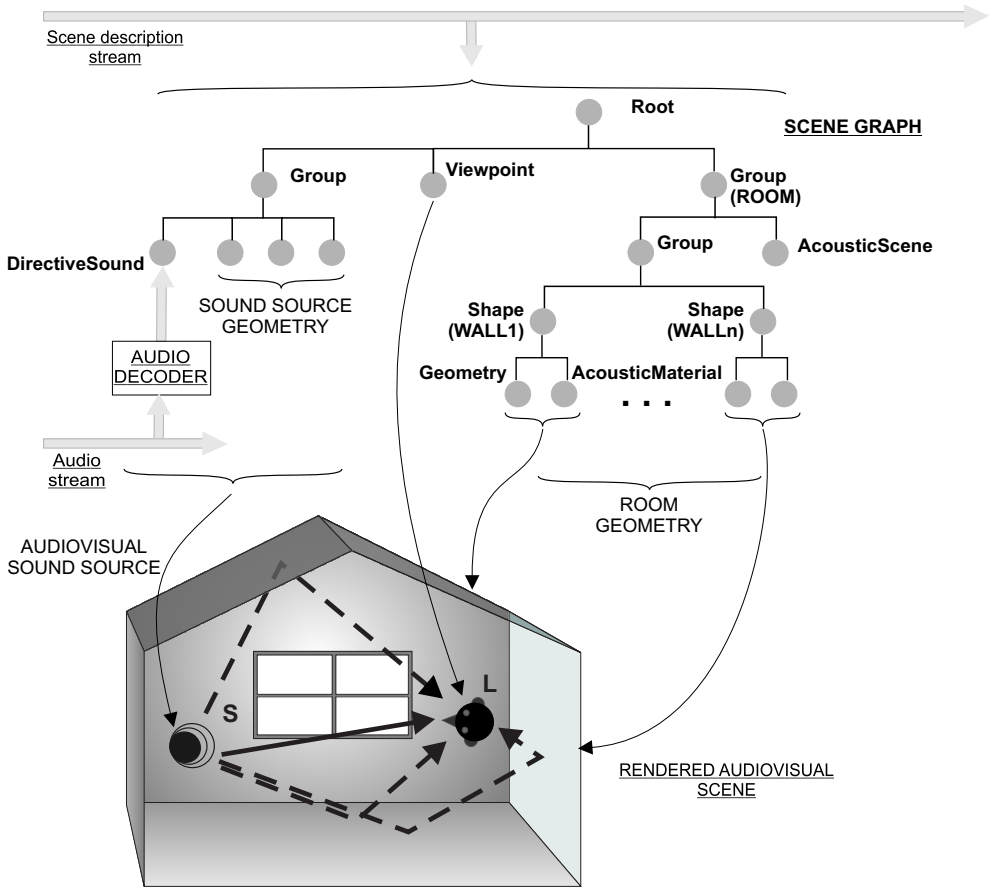

Figure 3.9: An example of an audiovisual MPEG-4 scene. Geometrical objects, viewing point, and spatial sound objects are defined in the scene description stream in a form of a scene graph that consists of BIFS nodes. Sound content is carried in a separate audio stream, which is linked with the sound object so that the sound appears coming from its defined position.

4. Summary of Publications and Author's Contribution

This chapter summarizes the publications, which form the main part of this thesis. The first ideas for this research came from Dr. Jyri Huopaniemi who was an instructor of the thesis, and a co-author in most of the publications. Other co-authors include Professors Matti Karjalainen, Vesa Välimäki, and Lauri Savioja, and Doctors Tapio Lokki, Ville Pulkki, and Eric Scheirer. From the co-operation with them the author obtained the background knowledge and scientific support for developing and applying the virtual acoustic modeling concepts in a new framework. Also the cooperation with the MPEG-4 community, especially the systems and the reference software implementation subgroups, have provided an important ground for the developments and implementations carried out in this thesis. Finally, the cooperation with IRCAM (in particular Dr. Olivier Warusfel and Mr. Olivier Delerue), and with the Carrouso EU project, have contributed to the contents of the publications.

The author's contribution in the first article ([P1] concern the modeling and parametrization of late reverberation, with a DSP structure later used in the DIVA (Digital Interactive Virtual Acoustics) system [5] where virtual acoustics is rendered in real time while the user navigates a virtual room. While participating the research of the DIVA group, the author got the idea of developing further the means to define and implement room acoustic properties in generic multimedia applications. These topics are dealt with in the publications [P2]-[P6]. Finally publication [P7] is a result of work carried out at IRCAM, and it deals with the authoring and user interaction of sound scenes.

[P1]

A digital signal processing structure is presented for producing a late reverberation effect to sounds. This reverberator consists of feedback delay lines, each of which is connected in series with a comb-allpass filter containing a considerably smaller delay than the feedback delay line. A lowpass filter is associated with each feedback delay line to enable a frequency-dependent control over the reverberation time. The outputs of each delay line are also summed and fed back to the input of the reverberator. This filter structure is in fact a special case of the Feedback Delay Network (FDN) reverberator. Its advantage is a rapidly-increasing reflection density in the impulse response (enabling a smaller number

of delay lines than with the basic FDN reverberator).

An experiment was presented for simulating measured room acoustics with this reverberator: The frequency-dependent reverberation time was parametrically adjusted to match that of a measured response.

The author has designed, implemented (In Matlab and C programming language), and evaluated the reverberator presented in this article. The paper has been mostly written by the author. The author carried out the room impulse measurements with the help of Dr. Jyri Huopaniemi, and analyzed them to obtain the reverberator parameters and the early reflections. The other authors helped mainly by proofreading the text.

[P2]

This article is the first journal publication about the audio part of the MPEG-4 scene description language (BInary Format for Scenes, or BIFS). The functionalities of the AudioBIFS used for hierarchical audio scene compositing in MPEG-4 are discussed in detail. Also the virtual reality compositing (auralization) of sound with the help of Advanced AudioBIFS in MPEG-4 is introduced. This article has its focus on the AudioBIFS (written mostly by Dr. Eric Scheirer) that is used for hierarchical compositing and filtering of audio streams. The 3-D sound model of the Virtual Reality Modeling Language (VRML97) is presented, as a starting point to further develop the spatial sound scene description in MPEG-4. The framework for enhanced 3-D sound scene description and rendering of virtual room acoustics in the MPEG-4 context is introduced.

The author's contribution in this article was to explain the definitions needed for parametric representation of 3-D sound scenes (most of section IV), and the comparison of it with the earlier, more simplified modeling of sound in standards used for creating audiovisual virtual worlds (in Section II.C).

[P3]

This article discusses the real-time rendering issues that can and need to be taken into account in designing MPEG-4 Advanced AudioBIFS scenes. It presents the digital filter structure applied in the reference software implementation of the MPEG-4 Systems standard, and shows how sound scenes with multiple rooms are defined and rendered in Advanced AudioBIFS.

The author's contribution to this article and the related work was to define the parametrization of the spatial sound scene description, and the related MPEG-4 definitions, in the context of physical sound scene modeling. Dr. Jyri Huopaniemi collaborated to this task, and in proofreading the text of this article. The described implementation of a spatial sound scene rendering (as a part of a 3D audiovisual rendering software), as well as simple example scenes to test the defined functionalities, were created by the author alone.

[P4]

This article discusses the differences and similarities between the physical and perceptual approaches used for room acoustic modeling. It presents this topic from the viewpoint of DSP implementation (in particular addressing its computational complexity) of the virtual room acoustics in MPEG-4 Advanced AudioBIFS scenes. Required filter structures for implementing different features defined in the scene description objects are described.

The author wrote most of this article, getting support from Dr. Jyri Huopaniemi mostly in terms of comments and proofreading the article.

[P5]

In this article different reproduction techniques for rendering of MPEG-4 3-D sound scenes are presented. It discusses the fact that the same parametric description of a sound scene can be auralized with the help of different sound reproduction systems, of which simple 2-speaker panning, binaural headphone reproduction, and the Vector Base Amplitude Panning (VBAP) were implemented. It addresses also the division of sound scene parameters to those that are dependent on the used reproduction system, and to those that are not affected by it.

The author has implemented the sound scene rendering of the three mentioned reproduction methods, and written most of the article (with the help from Dr Ville Pulkki and Dr. Jyri Huopaniemi mostly in giving comments on the produced text). Dr. Ville Pulkki helped in adapting the VBAP rendering within the rendering software, and gave the example code to implement VBAP.

[P6]

This article provides a detailed description of the finalized Advanced AudioBIFS definition in the MPEG-4 Systems standard. The BIFS nodes that are included in the Advanced AudioBIFS are thoroughly dealt with, aiming at showing, what type of 3-D sound scenes can be transmitted and rendered as a part of the MPEG-4 content. Concepts of 3D sound scene modeling are overviewed, and scene description functionalities that are useful for interactive sound scene creation are explained.

The author wrote most of the text in all the sections except in Sections II and part of section VIII.

[P7]

This article deals with authoring and user interaction issues in the context of creating spatial sound scenes (defined according to the perceptual room acoustic modeling approach). An existing authoring tool was taken as a basis to develop an extension which allowed for producing sound scenes from a graphical representation created with the tool. These

sound scenes are first written in a textual format and transmitted to an encoder that converts them to a standard MPEG-4 format. Scene modifications can also be initiated from the authoring tool by transmitting scene parameter updates to the MPEG-4 decoder over a network.

A user interface is also transmitted as a part of the scene description. The properties of this user interface depends on the sound scene configuration (e.g., how many sound sources there are and what are their locations in the scene), and also on which interaction capabilities the author wants to include in the user interface. Thus the authoring tool also produces automatically a 2D visual representation of the scene, and optionally a visual interaction tool that lets the renderer-side user to modify the scene locally.

The author's contribution was to design and program the needed extensions to a sound scene authoring tool. These extensions include objects that correspond to the MPEG-4 Advanced AudioBIFS nodes and their properties. Functionalities for making the conversion from the internal format of the authoring tool to a textual format accepted by an MPEG-4 encoder were implemented, and the BIFS format user interface was designed. In this framework also dynamic scene modifications can be sent from the authoring tool to the renderer.

5. Conclusions and Future Challenges

The aim of this thesis was to develop a parameter set for representing 3D sound scenes for various applications and data transmission contexts. One of the motivations for this was that previously existing systems, which enable describing or rendering of sound scenes (such as application programming interfaces, professional room acoustic modeling programs, or research projects), but do not have a common way of representing sound scenes. Thus a model that is created to be used in one system can not be used in another. A second motivation was, that existing application programming interfaces (such as DirectX or VRML97 standard) did not provide the possibility to add sophisticated spatial sound properties to applications.

The developed spatial sound scene parameter set is useful for audiovisual (2D or 3D) applications, but also for audio-only applications. In virtual world applications, for example, spatial sound may be needed for supporting the visual part of the scene and for increasing its immersiveness. In audio-only applications the sound may be put through spatial sound processors without the presence of a visual scene (e.g., for music post-processing purposes or for artistic sound installations). One of the main advantages of the developed parameter set is that it provides a flexible definition of the sound scene: Where a spatial processing is not needed it can be (or individual parts of it) disabled for selected sound sources.

The implementation of a 3D sound scene renderer, which was also performed as a part of this thesis, proves the feasibility and flexibility of the proposed sound scene description. The sound scene parametrization (and its implementation) has been included in an international standard (MPEG-4). It has also been chosen as a scene transmission format in a EU project called the Carrouso, which further proves its usefulness and that there is actually a need to transmit sound scene data in modern communication networks. In the latter framework (real-time communication of sound scenes), the third aim of this thesis, namely the authoring of sound scenes, was obtained and its results tested.

5.1 Future Challenges

Future challenges for the topics of this theses are:

- Scene description development, to introduce even more advanced acoustic properties in interactive spatial sound scenes. Such improvements may include, e.g.,

studying and taking into account phenomena caused by the wave-like nature of sound, and more complex sound reflectivity models. Another improvement would be to increase the consistency between the developed parameter set and other existing specifications (such as those recommended in I3DL2 guidelines).

- Accordingly to study how these features can be implemented in real-time sound scene rendering software. Another challenge concerning the rendering of scenes is, what are the relevant features to be rendered in each application. For example, the MPEG-4 conformance rules defined minimum requirements for rendering, but not which properties are highest in the priority when some have to be suppressed. For example, the minimum number of rendered sound sources and reflections is defined, but not what are the rules for choosing the audible reflections. This is partly a psychoacoustic issue, and partly dependent on the application. A flexible (and ”intelligent”) renderer is able to choose the most important features, or offer the user the possibility to affect the priority of the implemented effects at the rendering stage.

- Authoring tool development is an important task for facilitating the creation of sound scenes and for increasing the acceptance and use of the proposed technology. Such tools should take into account the variety of applications that can be created using the parametric characterization of spatial sound scenes. Existing programs for authoring of audiovisual scenes on one hand, and for room acoustic modeling on the other, provide useful directions and a starting point for the development of such tools.

Bibliography

[1] R. S. Kalawsky. *The Science of Virtual Reality*. Addison-Wesley. Wokingham, England, 1993.

[2] D. Begault. *3-D Sound for Virtual Reality and Multimedia*. Academic Press, Cambridge, Massachusetts, USA, 1994.

[3] ISO/IEC 14496. International Standard (IS) 14496:1999. Information Technology – Coding of audiovisual objects (MPEG-4). 1999.

[4] M. Kleiner, B.-I. Dalenbäck, and P. Svensson. Auralization - an overview. *Journal of the Audio Engineering Society*, 41(11):861–875, November 1993.

[5] L. Savioja, J. Huopaniemi, T. Lokki, and R. Väänänen. Creating interactive virtual acoustic environments. *Journal of the Audio Engineering Society*, 47(9):1148–1166, December 1999.

[6] L. Savioja. *Modeling Techniques for Virtual Acoustics*. PhD thesis, Helsinki University of Technology, Espoo, Finland, December 1999.

[7] T. Lokki. *Physically-based Auralization: Design, Implementation, and Evaluation*. PhD thesis, Helsinki Univeristy of Technology, Espoo, Finland, November 2002.

[8] J. Huopaniemi. *Virtual Acoustics and 3-D Sound in Multimedia Signal Processing*. PhD thesis, Helsinki University of Technology, Espoo, Finland, November 1999.

[9] T. Tolonen. *Object-Based Sound Source Modeling*. PhD thesis, Helsinki University of Technology, Espoo, Finland, October 2000.

[10] J. O. Smith. Physical modeling synthesis update. *Computer Music Journal*, 20:44–56, 1996.

[11] E. D. Scheirer and L. Ray. Algorithmic and wavetable synthesis in the MPEG-4 multimedia standard. In *Preprint No. 4811 of the 105th Convention of the Audio Engineering Society*, San Francisco, California, USA, September 1998.

[12] D. Ellis. *Prediction-driven Computational Scene Analysis*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachussets, USA, June 1996.

[13] J. Huopaniemi, L. Savioja, R. Väänänen, and T. Lokki. Virtual acoustics - applications and technology trends. In *Proceedings of the X European Signal Processing Conference (EUSIPCO). Vol 4.*, pages 2201–2208, Tampere, Finland, September 5-8 2000.

[14] H. Järveläinen. *Perception of Attributes in Real and Synthetic String Instrument Sounds*. PhD thesis, Helsinki University of Technology, Espoo, Finland, January 2003.

[15] B.-I. Dalenbäck, M. Kleiner, and U. P. Svensson. Auralization, virtually everywhere. In *Preprint NO. 4228 of the 100th Audio Engineering Society Convention*, Copenhagen, Denmark, May 1997.

[16] M. Karjalainen, J. Huopaniemi, and V. Välimäki. Direction-dependent physical modeling of musical instruments. In *Proceedings of the 15th International Congress on Acoustics*, pages 451–454, Trondheim, Norway, June 1995.

[17] J. Huopaniemi, M. Karjalainen, V. Välimäki, and T. Huotilainen. Virtual instruments in virtual rooms – a real-time binaural room simulation environment for physical models of musical instruments. In *Proceedings of the International Computer Music Conference (ICMC'94)*, pages 455–462, Aarhus, Denmark, September 1994.

[18] N. H. Fletcher and T. D. Rossing. *The Physiccs of Musical Instruments*. Springer, New York, 1991.

[19] J. Flanagan. Analog measurements of sound radiation from the mouth. *The Journal of the Acoustical Society of America*, 32:1613–1620, 1960.

[20] J. Huopaniemi, K. Kettunen, and J. Rahkonen. Measurement and modeling techniques for directional sound radiation from the mouth. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Mohonk Mountain House, New Paltz, New York, USA, October 1999.

[21] J. Stautner and M. Puckette. Designing multi-channel reverberators. *Computer Music Journal*, 6(1):569–579, 1982.

[22] F. R. Moore. A general model for spatial processing of sounds. *Computer Music Journal*, 7(3):559–568, 1983.

[23] M. Barron. The subjective effects of first reflections in concert halls - need for lateral reflections. *Journal of Sound and Vibration*, 15:211–232, 1971.

[24] J-M. Jot. Real-time spatial processing of sounds for music, multimedia and interactive human-computer interfaces. *Multimedia Systems*, 7:55–69, 1999.

[25] M. R. Schroeder. Natural sounding artificial reverberation. *Journal of the Audio Engineering Society*, 10(3):219–223, 1962.

[26] J. A. Moorer. About this reverberation business. *Computer Music Journal*, 3(2):13–28, 1979.

[27] J-M. Jot and A. Chaigne. Digital delay networks for designing artificial reverberators. In *An Audio Engineering Society preprint 3030 (E-2). Presented at the 90th AES Convention*, pages 1–14, Paris, 1991.

[28] J-M. Jot. An analysis/synthesis approach to real-time artificial reverberation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 1992*, pages II 221–II 224, San Francisco, California, USA, 1992.

[29] J-M. Jot. *Etude et realisation d'un spatialisateur de sons par modeles physique et perceptifs*. PhD thesis, l'Ecole Nationale Superieure des Telecommunications, Telecom Paris 92 E 019, September 1992.

[30] D. Rocchesso and J. O. III Smith. Circulant and elliptic feedback delay networks for artificial reverebration. *IEEE Transactions on Speech and Audio Processing*, 5(1):51–63, January 1997.

[31] W. Gardner. Virtual acoustic room. Master's thesis, Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, USA, 1992.

[32] J.O Smith and D. Rocchesso. Connections between feedback delay networks and waveguide networks for digital reverberation. In *Proceedings of the International Computer Music Conference*, pages 376–377, 1994.

[33] W. Gardner. Efficient convolution without input-output delay. *Journal of the Audio Engineering Society*, 43(3):127–136, 1995.

[34] W. Gardner. *Applications of Digital Signal Processing to Audio and Acoustics*, chapter Reverberation Algorithms, pages 85–131. Kluwer Academic Publishers, Norwell, Massachusetts, USA, 1998.

[35] L. Dahl. A reverberator based on absorbent all-pass filters. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00)*, Verona, Italy, December 2000.

[36] J-M. Jot and O. Warusfel. A real-time spatial sound processor for music and virtual reality applications. In *Proceedings of the International Computer Music Conference*, pages 294–295, Banff, Canada, September 1995.

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

[38] J-M. Jot. Efficient models for reverberation and distance rendering in computer music and virtual audio reality. In *Proceedings of the International Computer Music Conference*, Thessaloniki, Greece, September 1997.

[39] J.-P. Jullien. Structured model for the representation and the control of room acoustical quality. In *Proceedings of the 15th International Congress on Acoustics*, Trondheim, Norway, 1995.

[40] ISO/IEC 14496. International Standard (IS) 14496:2000. Information Technology – Coding of audiovisual objects (MPEG-4). Second Edition. 2000.

[41] R. Väänänen and J. Huopaniemi. *The MPEG-4 Book*, chapter 12: SNHC Audio and Audio Composition, pages 545–581. Prentice Hall, 2002.

[42] J.-M. Jot, V. Larcher, and J.-M. Pernaux. A comparative study of 3-D audio encoding and rendering techniques. In *Proceedings of the AES 16th International Conference. (Spatial Sound Reproduction)*, pages 281–300, Rovaniemi, Finland, April 1999.

[43] N. Xiang and J. Blauert. A miniature dummy head for binaural evaluation of tenth-scale acoustic models. *Applied Acoustics*, 33:123–140, 1991.

[44] N. Xiang and J. Blauert. Binaural scale model for auralization and prediction of acoustics in auditoria. *Applied Acoustics*, 38:267–290, 1993.

[45] K. Oguchi and M. Nagata S. Ikeda and. Application of binaural hearing to scale-model testing. *Journal of the Audio Engineering Society*, 41(11):931–938, 1993.

[46] J.-D. Polack, X. Meynial, G. Godd, and A. H. Marshall. The MIDAS systen for all-scale room acoustic measurements. In *Proceedings of the AES 11th International Conference.*, pages 322–331, Portland, OR, 1992.

[47] J.-D. Polack, X. Meynial, and V. Grillon. Auralization in scale models: Processing of impulse response. *Journal of the Audio Engineering Society*, 41(11):939–945, 1993.

[48] B. M. Gibbs and D. K. Jones. A simple image method for calculating the distribution of sound pressure levels within an enclosure. *Acoustica*, 26:24–32, 1972.

[49] J. B. Allen and D. A. Berkley. Image method for efficiently simulating small-room acoustics. *Journal of the Acoustical Society of America*, 65(4):943–950, April 1979.

[50] J. Borish. An extension of the image model to arbitrary polyhedra. *Journal of the Acoustical Society of America*, 75:1827–1836, 1984.

[51] U. R. Kristiansen, A. Krokstad, and T. Follestad. Extending the image method to higher-order reflections. *Acta Acoustica*, 38(2-4, Special Issue on Computer Modelling and Auralization of Sound Fields in Rooms):195–206, 1993.

[52] L. Savioja, J. Huopaniemi, T. Lokki, and R. Väänänen. Virtual environment simulation – Advances in the DIVA project. In *International Conference on Auditory Display (ICAD'97)*, pages 43–46, November 1997.

[53] T. Lokki, J. Hiipakka, and L. Savioja. A Framework for evaluating virtual acoustic environments. In *Preprint No. 5317 of the 110th Audio Engineering Society Convention.*, Amsterdam, Netherlands, May 2001.

[54] U. P. Svensson, R. I. Fred, and J. Vanderkooy. Analytic scondary source model of edge diffraction impulse responses. *The Journal of the Acoustical Society of America*, 106(5):2331–2344, November 1999.

[55] R. R. Torres, U. P. Svensson, and M. Kleiner. Computation of edge diffraction for more accurate room acoustics auralization. *The Journal of the Acoustical Society of America*, 109(2):600–610, February 2001.

[56] V. Pulkki, T. Lokki, and L. Savioja. Implementation and visualization of edge diffraction with image-source method. In *Preprint No. 5603 of the AES 112th Convention*, Munich, Germany, May 2002.

[57] A. Krokstad, S. Strom, and S. Sorsdal. Calculating the acoustical room response by the use of a ray racing method. *Journal of Sound and Vibration*, 8(1):118–125, 1968.

[58] K. H. Kuttruff. Auralization of impulse responses modeled on the basis of ray-tracing results. *Journal of the Audio Engineering Society*, 41(11):876–880, 1993.

[59] B.-I. Dalenbäck, U. P. Svensson, and M. Kleiner. Prediction and auralization based on a combined image source/ray-model. In *Proceedings of the 14th International Congress on Acoustics (ICA'92), paper No. F2-7*, Bejing, China, July 1992.

[60] Graham Naylor. ODEON – Another hybrid room acoustical model. *Applied Acoustics*, 32(2-4, Special Issue on Computer Modeling and Auralization of Sound Fields in Rooms):131–143, 1993.

[61] C. Lynge. *ODEON Room Acoustics Software*. Brüel & Kjaer, http://www.dat.dtu.dk/ odeon/, October 2001.

[62] B.-I Dalenbäck. *CATT-Acoustic*. CATT. http://www.catt.se/, February 2002.

[63] S. Choi and H. Tachibana. Estimation of impulse response in a sound field by the finite difference method. In *Proceedings of the 13th International Congress on Acoustics (I.C.A.)*, volume 2, pages 129–132, Belgrad, July 1989.

[64] G. SenGupta. Finite element analysis of natural frequencies of acoustic enclosures with periodic properties. *Journal of Sound and Vibration*, 145(3):528–532, 1991.

[65] D. Botteldooren. Finite-difference time-domain simulation of low frequency room acoustic problems. *Journal of the Acoustical Society of America*, 98(6):3302–3308, December 1995.

[66] S. A. Van Duyne and J.O. Smith. The 3d tetrahedral digital waveguide mesh with musical applications. In *Proceedings of the International Computer Music Conference*, pages 9–16, 1996.

[67] L. Savioja, J. Backman, A. Järvinen, and T. Takala. Waveguide mesh method for low-frequency simulation of room acoustics. In *15th International Congress on Acoustics*, pages 637–640, Trondheim, Norway, June 1995.

[68] L. Savioja. Improving the three-dimensional waveguide mesh by interpolation. In *Proceedings of the Nordic Acousitcal Meeting (NAM '98)*, pages 265–268, September 1998.

[69] L. Savioja and V. Välimäki. Reducing the dispersion error in the interpolated digital waveguide mesh by frequency warping. pages 973–976.

[70] D.G. Malham and A. Myatt. 3-D sound spatialization using ambisonic techniques. *Computer Music Journal*, 19(4):58–70, 1995.

[71] M. M. Boone, E. N. G. Verheijen, and G. Jansen. Virtual reality by sound reproduction based on wave field synthesis. In *Preprint No. 4145 of the 100th AES Convention*, Copenhagen, Denmark, May 1996.

[72] F. L. Wightman and D. J. Kistler. Headphone simulation of free-field listening. I: stimulus synthesis. *Journal of the Acoustical Society of America*, 85(2):858–867, 1989.

[73] J. Blauert. *Spatial Hearing, the Psychophysics of Human Sound Localization*. MIT Press, Cambridge, Massachusetts, USA, 1997.

[74] H. Möller. Fundamentals of binaural tehcnology. *Applied Acoustics*, 36:171–214, 1992.

[75] J-M. Jot, V. Larcher, and O. Warusfel. Digital signal processing issues in the context of binaural and transaural stereophony. In *Presented at the 98th AES convention. An Audio Engineering Society preprint 3980 (I6)*, Paris, February 1995.

[76] W. Gardner. *3-D Audio Using Loudspeakers*. Kluwer Academic Publishers, Boston, 1998.

[77] V. Larcher. *Techniques de spatialisation de Son pour la ralit virtuelle*. PhD thesis, L'Universite de Paris VI, 2001.

[78] M. R. Schroeder and B. S. Atal. Computer simulation of sound transmissioon in rooms. In *IEEE International Convention Record (7). New York: IEEE Press*.

[79] P. Damaske. Head-related two-channel stereophony with loudspeaker reproduction. *The Journal of the Acoustical Society of America*, 50(4):1109–1115, 1971.

[80] D.H. Cooper and J. L. Bauck. Prospects for transaural recording. *Journal of the Audio Engineering Society*, 37(1/2), 1989.

[81] M. Karjalainen, A. Härmä, and J. Huopaniemi. Warped filters and their audio applications. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Mohonk Mountin House, New Paltz, New York, 1997.

[82] D.H. Cooper. Problems with shadowless stereo theory: Asymptotic spectral status. *Journal of the Audio Engineering Society*, 35, 1987.

[83] V. Pulkki. *Spatial Sound Generation and Perception by Vector Base Amplitude Panning*. PhD thesis, Helsinki University of Technology, Espoo, Finland, August 2001.

[84] G. Thiele and G. Plenge. Localization of lateral phantom sources. *Journal of the Audio Engineering Society*, 25(4):196–200, April 1977.

[85] D. M. Leakey. Some measurements on the effect of interaural intensity and time difference in two channel sound systems. *Journal of the Audio Engineering Society*, 31:977–986, July 1959.

[86] B. Bernfield. Attempts for better understanding of the directional stereophonic listening mechanism. Rotterdam, Netherlands, 1973.

[87] ITU-R BS.775-1. Multichannel stereophonic sound system with and without accompanying picture. Recommendation of International Telecommunications Union – Radiocommunication Standards. 1994.

[88] P. Flanagan, D. Dickins, and L. Layton. Real-time virtual acoustics for 5.1. In *Proceedings of the AES 16th International Conference. (Spatial Sound Reproduction)*, pages 136–140, Rovaniemi, Finland, April 1999.

[89] V. Pulkki. Virtual source positioning using vector base amplitude panning. *Journal of the Audio Engineering Society*, 45(6):456–466, June 1997.

[90] M. Gerzon. Periphony: Width-height sound reproduction. *Journal of the Audio Engineering Society*, 21(1/2):2–10, 1973.

[91] J. Pope, D. Creasey, and A. Chalmeres. Real-time room acousitcs using ambisonics. In *Proceedings of the AES 16th International Conference. (Spatial Sound Reproduction)*, pages 427–435, Rovaniemi, Finland, April 1999.

[92] A. J. Berkhout. A holographic approach to acoustic control. *Journal of the Audio Engineering Society*, 36(12):977–995, December 1988.

[93] M. M. Boone. Acoustic rendering with wave field synthesis. In *Proceedings of the ACM SIGGRAPH Campfire*, page ??, Snowbird, Utah, May 2001.

[94] E. Corteel, U. Horbach, and R. Pellegrini. Multi-channel inverse filtering of distributed-mode loudspeakers for wave field synthesis. In *Preprint NO. 5611 of the 112nd Audio Engineering Society Convention*, Munich, Germany, May 2002.

[95] M. M. Boone and P. J. de Bruijn. On the applicability of distributed mode loudspeaker panels for wave field synthesis based sound reproduction. In *Preprint No. 5165 of the 108th AES Convention*, Paris, France, February 2000.

[96] CARROUSO. Creating, Assessing and Rendering in Real time Of high quality aUdio-viSual envirOnments in MPEG-4 context. Carrouso homepage: http://emt.iis.fhg.de/projects/carrouso/, 2002.

[97] R. Väänänen, O. Warusfel, and M. Emerit. Encoding and rendering of perceptual sound scenes in the CARROUSO project. In *Proceedings of the AES 22nd International Conference (Virtual, Synthetic, and Entertainment Audio)*, pages 289–297, Espoo, Finland, June 2002.

[98] U. Horbach, E. Corteel, R. Pellegrini, and E. Hulsebos. Real-time rendering of dynamic scenes using wave field synthesis. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, Lausanne, Switzerland 2002.

[99] B. Stroustrup. What is object-oriented programming. In *Proceedings of the 1st European Software Festival*, Munich, Germany, February 1991.

[100] B. Stroustrup. *The C++ Programming Language*. Addison-Wesley, 3rd edition, 1997.

[101] R. Carey and G. Bell. *The Annotated Vrml 2.0 Reference Manual*. Addison-Wesley, Boston, MA, May 1997.

[102] B. (Editor) Joy, G. Steele, J. Gosling, and G. Bracha. *The Java Language Specification, Second Edition.* Addison Wesley, 2000. ISBN 0201310082, *url:*<http://java.sun.com/docs/books/jls/index.html.

[103] Sun Microsystems. The Java Tutorial. Published at http://java.sun.com/docs/books/tutorial/java/concepts/.

[104] Mircrosoft DirectX 9.0 Documentation. Available at: http://msdn.microsoft.com/library/.

[105] OpenGL web site: http://www.opengl.org/.

[106] A. E. Walsh and D. Gehringer. *Java 3D API Jump-Start*. Prentice Hall PTR, 1st edition edition, August 2001.

[107] ISO/IEC 14772-1. International Standard (IS) 14772-1. The Virtual Reality Modeling Language (VRML97) (Information technology – Computer graphics and image processing – The Virtual Reality Modeling Language (VRML) – Part 1: Functional specification and UTF-8 encoding.). April 1997. *url*: http://www.vrml.org/technicalinfo/specifications/ISO_IEC_14772-All/index.html.

[108] ISO/IEC X3D. International Standard (IS) 19775-1:2002 (Committee Draft), X3D. Information technology – Computer graphics and image processing – eXtensible 3D (X3D) – Part 1: Architecture and Base Components. 200x. URL (draft): http://www.web3d.org/TaskGroups/x3d/specification-2002february/main.html and http://www.web3d.org/x3d/.

[109] J. Signes, Y. Fisher, and A. Eleftheriadis. MPEG-4's binary format for scene description. *Signal Processing: Image Communication. Tutorial Issue on MPEG-4*, 15(4-5):321–345, 2000.

[110] IASIG. Interactive Audio Special Interest Group (IA-SIG) homepage: http://www.iasig.org/, 2000.

[111] IASIG. Interactive 3D Audio Rendering Guidlines, Level 1.0. (Available at the web site of Interactive Audio Special Interest Group (IA-SIG): http://www.iasig.org/wg/closed/3dwg/3dl1v1.pdf), June 1998.

[112] IASIG. Interactive 3D Audio Rendering Guidlines, Level 2.0. (Available at the web site of Interactive Audio Special Interest Group (IA-SIG): http://www.iasig.org/wg/closed/3dwg/3dl2v1a.pdf), September 1999.

[113] IASIG. DirectSound 3.0 Extension API. (Available at the web site of Interactive Audio Special Interest Group (IA-SIG): http://www.iasig.org/), 1997.

[114] Creative Technology Ltd. Environmental Audio Effects (EAX) API 3.0. Available at EAX web site http://eax.creative.com (Developer info at http://developer.creative.com).

[115] OpenAL. Open audio library (OpenAL) 1.0 Specification and Reference. Available at OpenGL web site: http://www.opengl.org/.

[116] W. Dale. A machine-independent 3D positional sound application programmer interface to spatial audio engines. In *Proceedings of the AES 16th International Conference. (Spatial Sound Reproduction)*, pages 160–171, Rovaniemi, Finland, April 1999.

[117] E. D. Scheirer, R. Väänänen, and J. Huopaniemi. AudioBIFS: The MPEG-4 Standard for Effects Processing. In *Proceedins of the First COST-G6 Workshop on Digital Audio Effects (DAFX98)*, Barcelona, Spain, November 1998.

[118] O. Delerue and O. Warusfel. Authoring of virtual sound scenes in the context of the Listen project. In *Proceedings of the AES 22nd International Conference (Virtual, Synthetic, and Entertainment Audio)*, pages 39–47, Espoo, Finland, June 2002.

[119] R. Koenen. Mpeg-4: Multimedia for our time. *IEEE Spectrum*, 36(2):26–33, February 1999.

[120] F. Pereira and T. Ebrahimi, editors. *The MPEG-4 Book*. Prentice Hall, 1st edition edition, December 2002.

[121] A. E Walsh and M. Bourges-Sevenir. *MPEG-4 Jump-Start*. Prentice Hall, 1st edition edition, December 2001.

[122] ISO/IEC 11172. International Standard (IS) 11172 Coding of moving pictures and associated audio for digital stroage media at up to about 1.5 Mbit/s (MPEG-1). 1992.

[123] ISO/IEC 13818. International Standard (IS) 13818-3. Information technology - generic coding of moving pictures and associated audio. (MPEG-2). 1994.

[124] J.-C. Dufourd. *The MPEG-4 Book*, chapter 4: BIFS Scene Description, pages 103–147. Prentice Hall, 2002.

[125] O. Avaro, R. Koenen, and F. Pereira. *The MPEG-4 Book*, chapter 2: The MPEG-4 Overview, pages 37–63. Prentice Hall, 2002.

[126] R. Väänänen. Synthetic audio tools in MPEG-4 standard. In *Preprint No. 5080 (B-2) of the 108th AES Convention*, Paris, February 2000.

[127] C. Herpel. *The MPEG-4 Book*, chapter 3: Object Description and Synchronization, pages 65–101. Prentice Hall, 2002.

[128] M. Nishiguchi and B. Edler. *The MPEG-4 Book*, chapter 10: Speech Coding, pages 451–485. Prentice Hall, 2002.

[129] E. D. Scheirer, Y. Lee, and J-W. Yang. Synthetic and SNHC audio in MPEG-4. *Signal Processing: Image Communication. Tutorial Issue on MPEG-4*, 15(4-5):445–461, 2000.

[130] M. Wollborn, I. Moccagatta, and Ulrich Benzler. *The MPEG-4 Book*, chapter 8: Natural Video Coding, pages 293–381. Prentice Hall, 2002.

[131] E. Jang, T. Capin, and J. Österman. *The MPEG-4 Book*, chapter 9: Visual SNHC Tools, pages 383–485. Prentice Hall, 2002.

[132] A. M. Tekalp and J. Östermann. Face and 2-d mesh animation in mpeg-4. *Signal Processing: Image Communication. Tutorial Issue on MPEG-4*, 15(4-5):387–421, 2000.