Game Sound Technology and Player Interaction:
Concepts and Developments

Mark Grimshaw
University of Bolton, UK

Information Science REFERENCE

INFORMATION SCIENCE REFERENCE
Hershey · New York

Director of Editorial Content:	Kristin Klinger
Director of Book Publications:	Julia Mosemann
Acquisitions Editor:	Lindsay Johnston
Development Editor:	Joel Gamon
Publishing Assistant:	Milan Vracarich Jr.
Typesetter:	Natalie Pronio
Production Editor:	Jamie Snavely
Cover Design:	Lisa Tosheff

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Game sound technology and player interaction : concepts and development / Mark Grimshaw, editor. p. cm.
Summary: "This book researches both how game sound affects a player psychologically, emotionally, and physiologically, and how this relationship itself impacts the design of computer game sound and the development of technology"-- Provided by publisher. Includes bibliographical references and index. ISBN 978-1-61692-828-5 (hardcover) -- ISBN 978-1-61692-830-8 (ebook) 1. Computer games--Design. 2. Sound--Psychological aspects. 3. Sound--Physiological effect. 4. Human-computer interaction. I. Grimshaw, Mark, 1963-
QA76.76.C672G366 2011
794.8'1536--dc22

2010035721

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The views expressed in this book are those of the authors, but not necessarily of the publisher.

Chapter 14
Spatial Sound for Computer Games and Virtual Reality

David Murphy
University College Cork, Ireland

Flaithrí Neff
Limerick Institute of Technology, Ireland

ABSTRACT

In this chapter, we discuss spatial sound within the context of Virtual Reality and other synthetic environments such as computer games. We review current audio technologies, sound constraints within immersive multi-modal spaces, and future trends. The review process takes into consideration the wide-varying levels of audio sophistication in the gaming and VR industries, ranging from standard stereo output to Head Related Transfer Function implementation. The level of sophistication is determined mostly by hardware/system constraints (such as mobile devices or network limitations), however audio practitioners are developing novel and diverse methods to overcome many of these challenges. No matter what approach is employed, the primary objectives are very similar—the enhancement of the virtual scene and the enrichment of the user experience. We discuss how successful various audio technologies are in achieving these objectives, how they fall short, and how they are aligned to overcome these shortfalls in future implementations.

INTRODUCTION

In the past, sound has often been a secondary consideration in visually intensive environments, such as Virtual Reality (VR) systems and computer games. However, hearing and several other perceptual modalities are now considered equally relevant to the user-experience within artificial and simulated domains. Linear soundscape composition, especially within computer games, has been facilitated with advancements in computer hardware and storage capacities. The sonic contribution of linear music to the virtual scene is extremely important, especially during gameplay, as it adds atmosphere, drama, emotion, and sometimes fantasy to the overall scene. However, interactive sounds and environmental acoustics are also important in enhancing the user-

DOI: 10.4018/978-1-61692-828-5.ch014

experience by immersing the user in the gameplay or VR scene. These types of sounds at present are still used mostly as effects rather than as authentic references to the virtual landscape. Accurate spatialization and real-time interactive sonic elements are essential if the user-experience is to be brought to the next level in future developments.

Many of the recently developed audio tools are based on well-established theory, but remain limited in their implementation of true spatial sound by hardware constraints. Some of the theory that has been successfully implemented to varying degrees are techniques such as Interaural Time Difference (ITD), Interaural Intensity Difference (IID), the Doppler Shift, and Distance Attenuation. However, many more spatial attributes remain difficult to render in real-time, such as high fidelity simulation of ear geometrics and head/shoulder shadow.

As mentioned earlier, some of the basic principles and techniques are now readily available to developers, but the underlying theory in this field indicates that for true spatial sound to be delivered to the listener, individualization[1] of the listening experience is key to its success. Despite the advances in hardware in recent times, all of the current spatialization techniques used within gaming and VR environments remain focused on a generalized listening experience, and, as of yet, no commercially viable method has been successfully implemented that achieves true individualized spatial sound. The generation of individualized Head Related Transfer Function (HRTFs) for commercial dissemination is one of the remaining milestones to be affected by hardware limitations. Many in the industry argue that generic solutions are sufficient in achieving an accurate sense of immersion in virtual environments for most users. This argument may well indeed hold true, except for the fact that it cannot truly be tested until we can compare it to individualized spatial sound on a commercial scale.

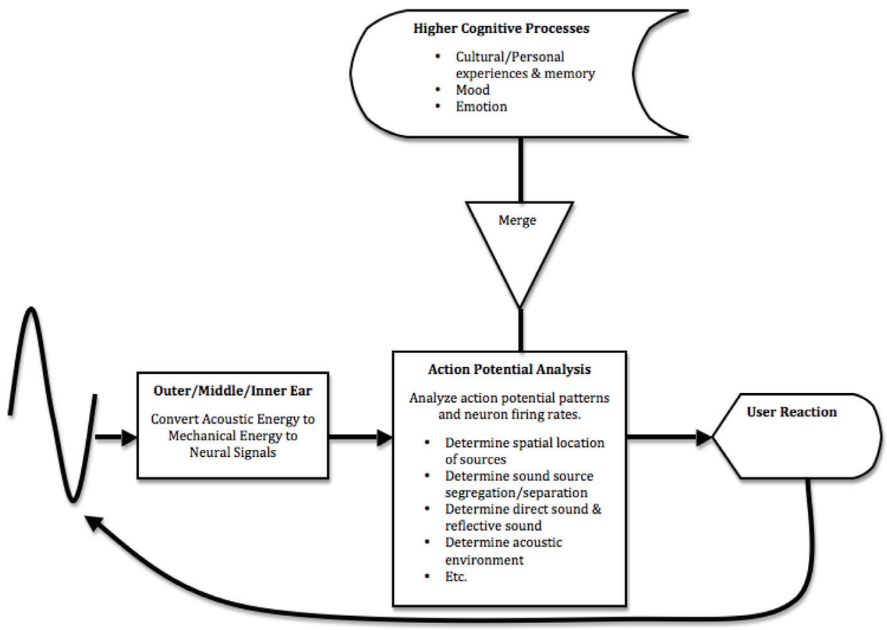
In addition to the limitations of implementing individualized spatial listening, there still persists the problem of rendering accurate room and outdoor acoustics. This is, again, down to the constraints of available hardware resources. Rendering what may be considered a simple scene in the visual domain could easily entail several very complex models of various acoustically dependent elements. For example, an accurate rendition of the listener closing a room door would need to model the room itself, the door's material and structure, and the change in acoustic space during the act of closing the door (from a coupled space to a singular space). In addition, other very important factors such as the material on the floor, walls and ceiling, and reflective and absorbing objects within the space also need to be modelled. All this, of course, in real-time!

In spite of the current limitations to implementing commercial solutions for individualized spatialization, the industry is employing very interesting and creative workarounds. It has not only tackled the distribution of sound in virtual space intuitively, but it has also efficiently tackled problems relating to large audio data file sizes and bandwidth. Therefore, in this chapter, not only do we review sound spatialization techniques, but in tandem, we also discuss audio compression technology and how this theme goes hand-in-hand with spatialized sound for VR and computer games.

PERCEPTUAL PROCESSING OF SOUND

The cognitive mechanisms involved in the aural perception of space are highly evolved and complex, and can be categorized into two distinct groups–direct analyses of physical/sensory information, and higher cognitive influences (see Figure 1). Both groups play a crucial role in our everyday hearing processes. Even in cases of perceived silence, background noise stimulates auditory spatial awareness by communicating spatial information about the surrounding environment to the listener based on both acute sensory

Figure 1. A simplified outline of the human auditory processing system. Higher cognitive processes influence what we hear and how we hear the external world

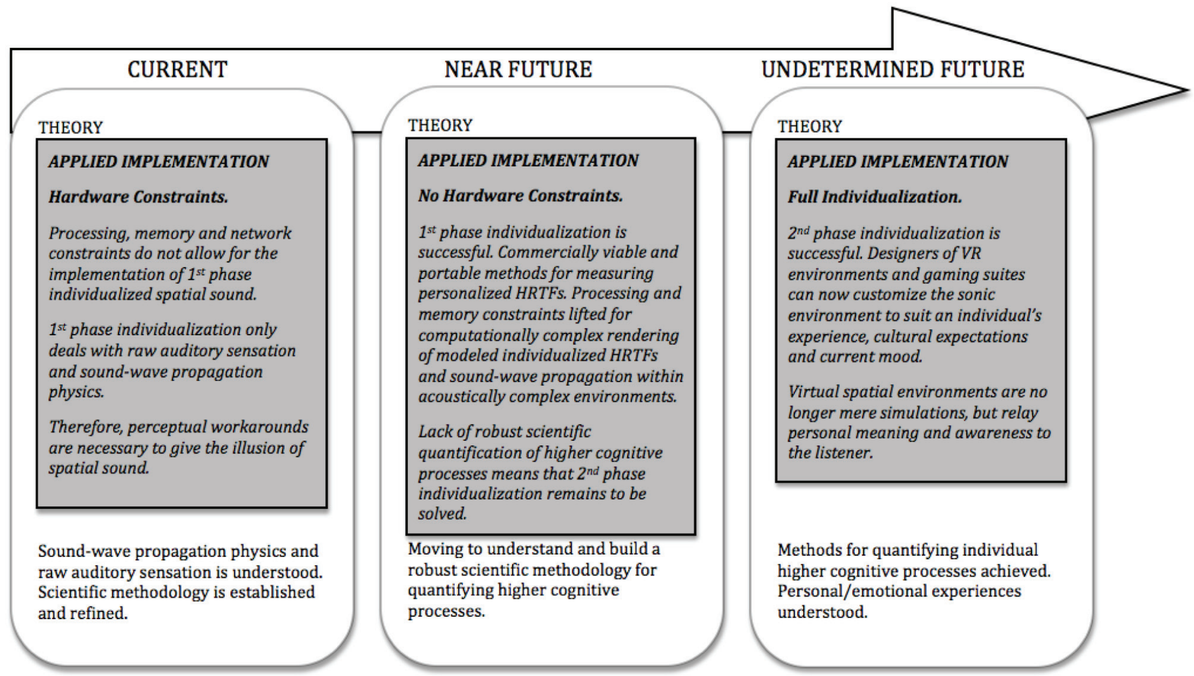


detection and environmental experience (Ashmed & Wall, 1999).

The physical interaction between moving sound-waves and static/moving objects in space are well understood (HRTF, ITD, ILD, Aural Occlusion, Doppler Shift and so on), but the higher cognitive mechanisms involved in audition have yet to be fully explored and explained. Many of these procedures are abstract concepts, such as the influence of cultural and personal experience on our sonic perception of space. This aspect of spatial sound design is vast and scientific methodology has yet to be fully developed for many of the issues. An understanding of the metamorphosis from raw auditory sensation to a spatial awareness that has meaning is perhaps the final chapter in creating true spatial sonic interfaces for VR and computer games. However, from a commercial standpoint, these issues are for future consideration given that convincing solutions for the "simpler" elements of spatial sound (that is, the physics of sound-wave propagation in the external environment) remain to be solved within reasonable cost and performance (Ashmed & Wall, 1999). Music is perhaps the closest, and one of the oldest, methods of interacting with higher cognitive processes. Figure 2 illustrates a forecast for the evolution of spatial sound implementation in VR and game technology.

In essence, raw auditory sensation of sound involves the physical transportation of both the attributes of the sonic event/source and the properties of an acoustic space (whether virtual or real-world) to the listener. Not only does the human auditory system detect and analyze the acoustic attributes of the sound source itself and localize it within a complex sonic soundscape, but it also determines the acoustic make-up of the space based on the difference between the direct sound and the physical effects the space has imposed on the sound as it travels through. An unfinished list of these alterations includes absorption, reflection, refraction, dispersion, and occlusion. But raw auditory sensation doesn't end

Figure 2. Evolution of spatial sound implementation in VR and game technology



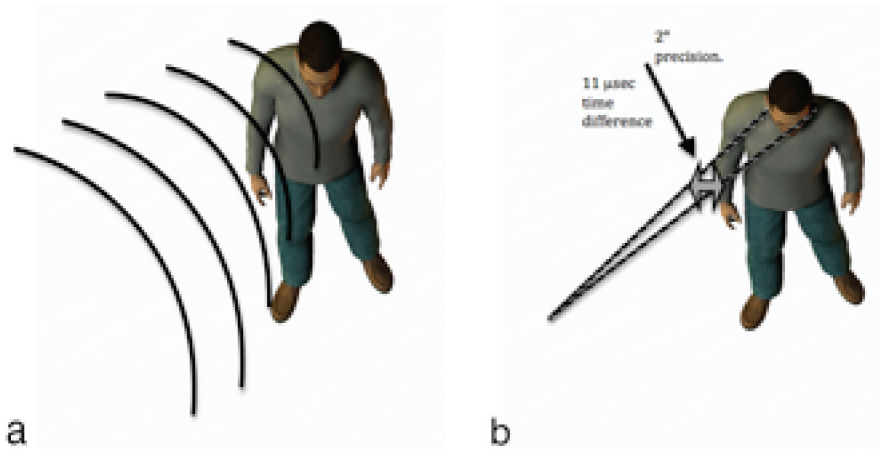
there. Before it enters the middle ear, the sound is filtered by the listener's unique ear and head shape/size. These processes of filtering, along with internal higher cognitive processes, combine to infuse the processed sound with a spatial signature unique to the listener.

Human Auditory System–Sound Localization Mechanisms

The basic (physiological-based) understanding of sound localization in humans is one comprising two distinct categories–the first dealing with the horizontal plane (left, right, front, back) and the other with the vertical plane (above and below the head position). With regard to sound localization on the horizontal plane, a number of factors come into play. One of the most obvious mechanisms is ITD where a sound will reach one ear before it reaches the other (the speed of sound remains constant, however, apparent differences in arrival time results from phase differences between the two signals). This mechanism is most useful for frequencies between 20 Hz to 2 kHz. In ITD, sound coming directly from the right will reach the right ear 0.6 msec before reaching the contralateral left (Begault, 1993). The accuracy of the human auditory system to locate sounds based on ITD is very impressive, with studies (Begault, 1993) showing a discrimination of an angle as small as 1º or 2º (which translates to a time difference of about 11 µsec!), depending on the position. Refer to Figure 3.

However, what happens when sounds are continuous and we do not have the onset information such as in sudden, brief sounds? A slight variant of ITD is used instead by the human auditory system. In such cases, the phase discrepancy between the right and left ears is analyzed and the location of a sound source is determined. In other words, the peak of a wave-cycle reaches the right ear before it reaches the left. From this method ensues another problem however. For high frequencies (in humans this would be around

Figure 3. a) Sound-wave arriving from the right. b) Sound-wave information reaches the right ear 11µsec before reaching the left. This corresponds to a sound-source detection precision of as little as 2º



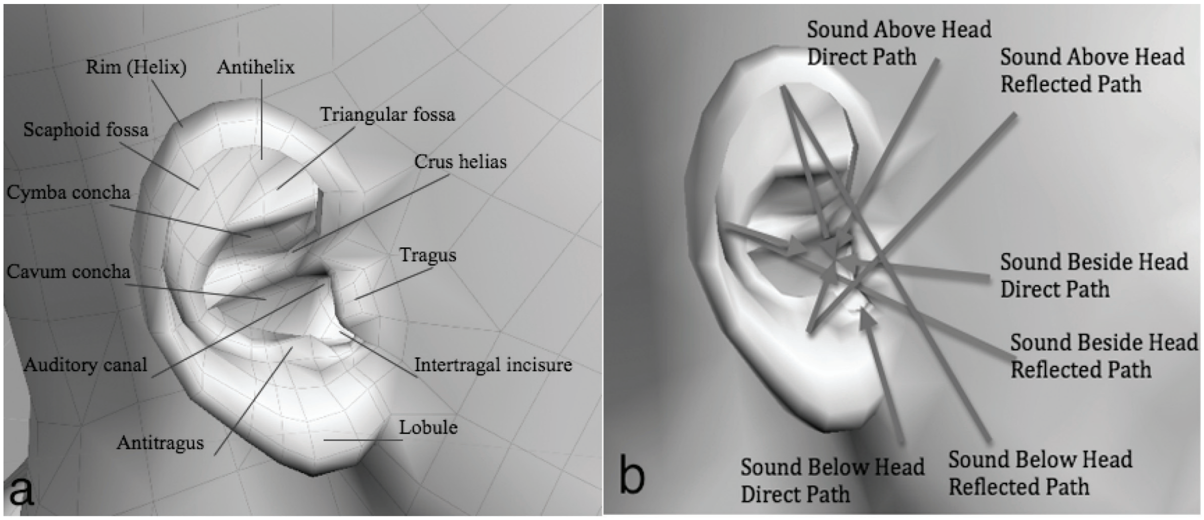
2 kHz to 20 kHz), the techniques involving ITD and its variant are not applicable. This is due to the nature of high frequencies, where the periods of each cycle are very short, meaning that many cycles have occurred within the distance between both ears. Therefore, when high continuous sounds are presented to the listener, phase discrepancy becomes unreliable. A very different analysis procedure is required to determine the location of continuous high-frequency sound source. To this end, IID is employed.

IID is a technique employed by the auditory system that describes the difference in intensity levels between sound signals arriving at both ears. In effect, this procedure takes into account the interaction between the external sound-wave and the listener's head and shoulders. As a solid body, the head and shoulders will reflect and absorb energy from a sound-wave as it travels past. In essence, this means that a sound traveling from the right will reach the right ear with a particular intensity level, but reaches the left ear with a lower intensity because of the head and shoulder interference.

The combination of ITD and IID (known as the Duplex Theory of sound localization) means that the human hearing system is very efficient at localizing sound on the horizontal plane. A very different approach is believed to take place where localization on the vertical plane is concerned.

The comparison between the inputs of a signal (time, phase or intensity) reaching both ears is not effective when localizing sound on the vertical plane. It is easy to comprehend why: A signal coming from above or below the listener is likely to reach both ears approximately at the same time and the head and body shadows the input of both ears almost equally. An alternative, albeit technically more complicated, analysis is used. The process entails the filtering of the signal before entering the auditory canal due to the geometric features of the pinnae. (Refer to Figure 4 for a detailed view of a pinna). The folds of the pinnae reflect certain frequencies of an incoming signal and, as a sound source moves vertically, the combined direct sound and reflected sound changes dynamically. See Figure 4a showing the structure of a pinna and 4b illustrating the combination of direct and reflected signal paths before entering the auditory canal.

Figure 4. a) The human pinna structure, b) a sound impinging and interacting with the pinna



SOUND PROPAGATION IN REAL SPACE

Synthetic environments in VR and computer games are very varied and sometimes very complex. Visual simulation of real-world scenes has come a long way in terms of player immersion and photo-realism. The more complex and detailed the visual representations become, the more elaborate and intricate the sonic attributes need to be in order to match user expectations. This can pose many problems for sound designers in terms of realistic acoustic simulation and sound source emission. Again, as with the theoretical understanding of psychoacoustics, environmental/room acoustics are well understood but the problem of implementing the theory in VR and computer games lies mainly with resource allocation and hardware constraints.

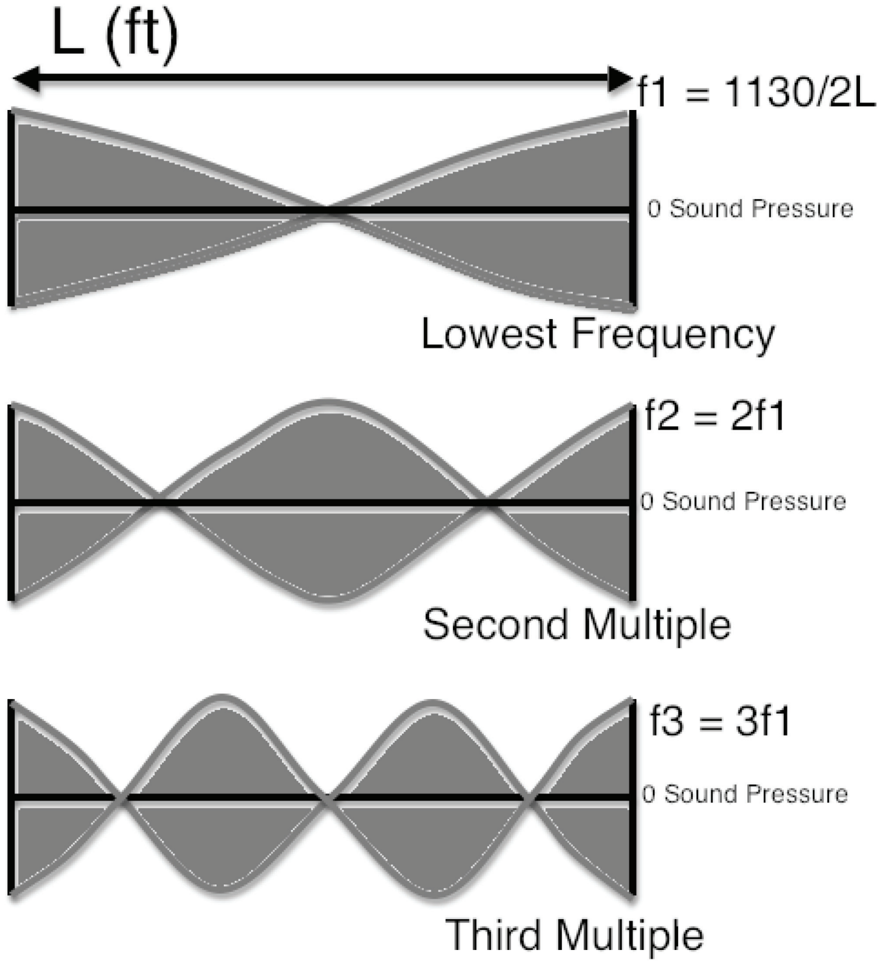
The propagation of sound can vary dramatically from scene to scene or from level to level during an instance of gameplay. The player can be within a small room enclosure in one scene and change suddenly to a wide, open space in another. Some scenes take place in unusual environments, such as under water or in outer-space, where the acoustics are very different from the typical air-based medium. Not only does the shape, size, and context of the space influence acoustics, but so too do static and moving objects within that space as well as materials of large surfaces such as tiled walls.

Indoor Acoustics

In the most basic terms, sound in an empty room is both absorbed by and reflected off surfaces. The energy that is reradiated is dispersed around the room and the listener hears both direct and reflected sound as a result. Between each opposing wall, a standing frequency and its associated multiples resonate . The standing waves that are produced express the room's resonant characteristic: there are multiple resonant frequencies in any one room. The acoustic result for the listener is a reinforcement of those resonant frequencies (by way of emphasized energy) when present in the soundscape. (Refer to Figure 5).

Another basic feature of indoor acoustics is reflection, which is often applied broadly in VR and computer game environments using general delay and reverberation units. In Figure 6a, the

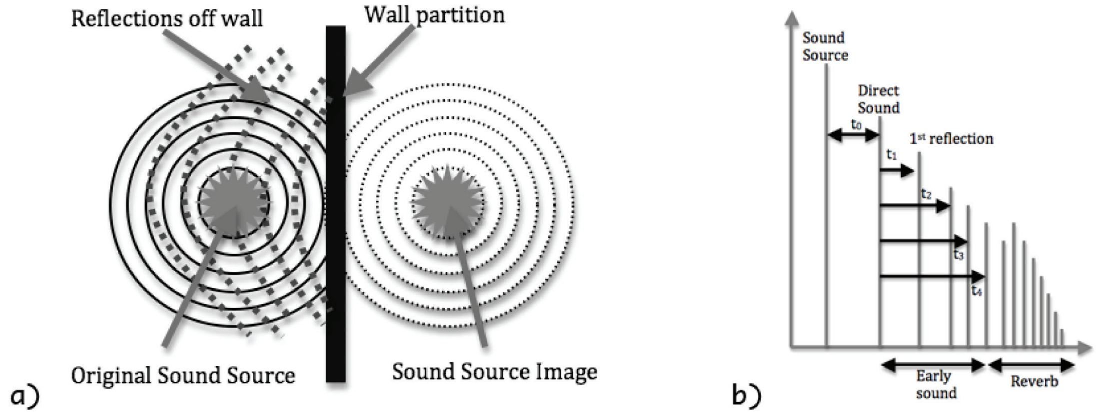
Figure 5. Standing waves between two parallel walls



most basic scenario is illustrated when a sound source is emitted next to a single-walled surface in free space. Sound is reflected off the surface and these reflections act as if they emanate from an exact copy of the original source but, instead, it is on the opposite side of the wall and the same distance from the wall as the original. Reflected sound is generally categorized according to the time it takes to reach the listener after the direct sound. Reflected sound reaching the listener approximately 50-80 ms after the direct sound is often referred to as *early sound* and can be indistinguishable from direct sound. To the listener, early sounds increase the perceived presence of the direct sound and have a slight intensity drop compared to the original direct sound. Reflections arriving after early sounds are usually more numerous, have shorter time gaps between each instance and contribute to the creation of *reverberation* (see Figure 6b).

Figure 6b illustrates a simple typical indoor reflected sound. However, imagine a listener positioned within a simple rectangular room. The sound source is now reflected off six surfaces (not

Figure 6. a) A sound is reflected from a surface in free space. The reflected sound-waves act like a copy of the original, but at the opposite side and the same distance from the surface. Based on image from Everest (2001), b) reverberation, comprised of direct sound, early reflections, and reverberation



to mention the listener). The concoction of direct sound and reflected sound, as well as taking the listener's position in the room into account, quickly becomes more complex to simulate. Couple this with the fact that the process of surface reflection is a function of the sound's frequency (wavelength), then a 100% accurate simulation would need to consider any frequency within the human hearing range (circa 16 Hz to 20000 Hz, or wavelengths from 21.5 m to 0.017 m). To deal with this large frequency range, sound designers and audio developers conform to modes whereby frequencies are arbitrarily grouped in relation to geometric acoustic data. In addition to room modes, reflections from static or moving objects also need to be considered, as well as the absorption coefficients of the various materials. All of these factors make spatial sound simulation of indoor environments resource-intensive and a complex component in overall computer game and VR design.

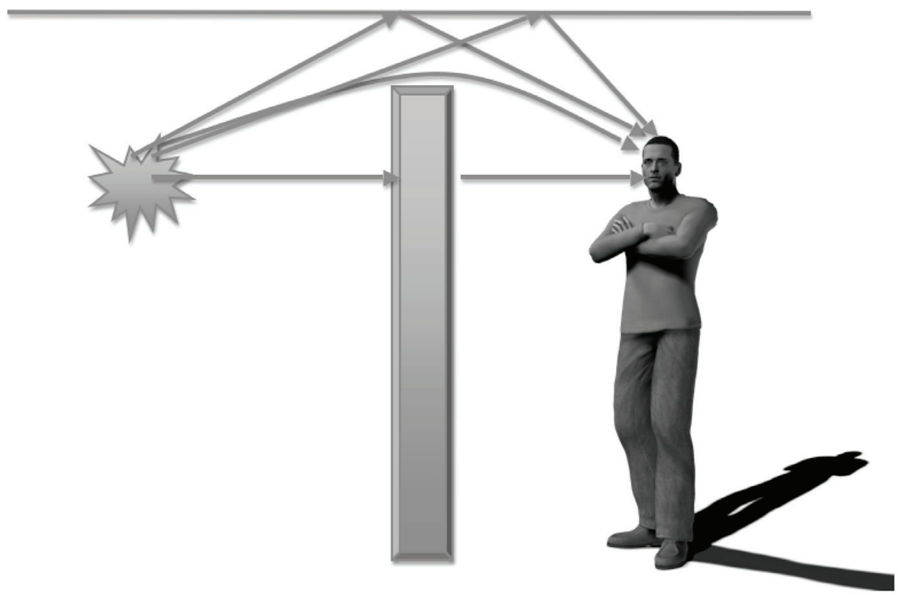
At points where there is semi-occlusion, such as glass-free windows or a ¾ wall partition (see Figure 7), there is at least some transmission of sound through the barrier, some diffraction around the barrier, and some reflection around the barrier. These indirect paths to the listener on the other side, as well as filtering based on the type of material the barrier consists of, impact on the final sound heard by the player/listener.

Outdoor Acoustics

In an open environment, away from reflective buildings or nearby surfaces, the outdoor scene could be considered a free field in the most simplistic terms. In such circumstances simple rules can apply, such as the inverse square law, when dealing with sound pressure and intensity levels between sound source and listener. However, increasingly rich visual representations of outdoor environments within VR and computer games leads players to expect increasingly accurate acoustic simulation. Bearing this in mind, future implementations may need to consider factors such as atmospheric absorption, refraction, turbulence, diffraction, humidity, temperature, ground material, and the listener's proximity to the ground (Fletcher, 2004).

Sound attenuation outdoors is frequency dependent, with high frequencies loosing much of their energy due to the elements described above. An example in nature is lightening and thunder. Lightening occurring close to the listener results

Figure 7. Sound is transmitted to the listener on the other side of an occluding barrier via through and around the barrier, as well as reflected from surfaces such as a ceiling



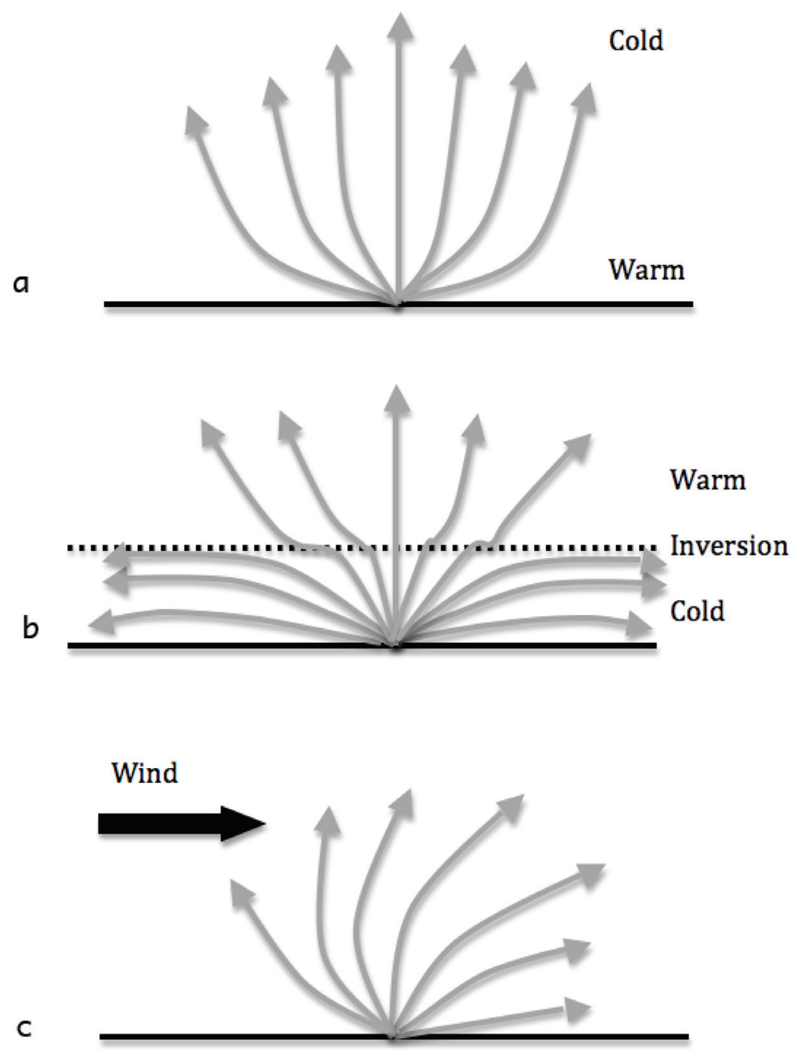
in thunder that is quite rich in both high and low frequencies. However, if the listener were much further from the source, only the low frequencies would survive the distance.

Most gameplay occurs on or close to the virtual ground where sound is reflected back to the listener at varying rates depending on the surface type. Some other factors that can influence outdoor sound propagation at ground level are things such as low-lying mist or fog. These conditions alter the sonic properties of the outdoor environment, with the effect of increasing the apparent loudness of distant sounds. This phenomenon is caused by a temperature irregularity, where cold air is closer to the ground than warm. This forces some sound-waves to bend back towards the ground at the point of temperature inversion (from cold to warm), instead of propagating upward and diminishing. See Figure 8a and 8b. However, an opposing factor in such an environment may also come into play where some sound may be attenuated or muffled due to humidity levels in the low-lying fog.

Furthermore, ground level gameplay may also incorporate wind elements that also have an effect on sound-wave direction. Sound may be carried toward the listener if they are positioned downwind or they may have difficulty hearing the sound in cases where they are upwind (see Figure 8c). Slopes and valleys also have an effect, as do ground surfaces themselves. For example, grass surfaces do not tend to affect frequencies below 100 Hz, but can seriously attenuate higher frequencies up to 40dB/km at 1 kHz (Fletcher, 2004). Similarly, trees and foliage scatter mid to high-end frequencies also, whilst their effect on low frequencies is minimal.

In relation to super-hero characters capable of flying well above ground level, similar effects in terms of sound propagation would also occur. These effects are predominantly due to temperature change (getting colder with increased height) and as the character flies further upward, colder temperatures slow the speed of sound (the difference being as much as 10% between ground level and 10000 meters) (Fletcher, 2004). Other

Figure 8. a) Typical dispersion of sound from a ground-based source showing the effects of warmer air close to the ground a) and colder air close to the ground b), also, wind can affect sound energy and dispersion c). Based on an image from Everest (1997)



factors affecting the sonic environment of our flying super-hero may also need to be considered, such as turbulence, which itself can generate low frequency sound in addition to scattering high frequencies.

VIRTUAL SPATIAL SOUND IMPLEMENTATION—INTRODUCTION

The level of fidelity in the implementation of sound localization varies considerably from system to system. Most of the time, constraints such as network bandwidth, processing speeds, storage restrictions, and memory considerations limit the flexibility required by sound designers. Such restrictions have led scientists to find imaginative alternatives of rendering spatial audio in synthetic environments–using strategies such as compact file sizes, low bit rates, client-based synthetic sound rendering and so forth, without impacting on perceived sound quality. In this section, we will review examples of some of the popular approaches to rendering spatial sound in games and VR.

In general, left-right source positioning is relatively easy to achieve on both headphone-based and speaker-based implementations. However, front-back sound positioning is often less successful, especially when reproduced on headphones. The primary reason for this is due to the inherent requirement of the listener to perform head movement when determining the location of sound in front or back scenarios. A surround-speaker setup in this respect does not have the same level of difficulty due to the arrangement of discrete channels placed in the physical world. In more sophisticated headphone scenarios, some VR systems incorporate head-tracking to allow for the integration of head orientation and movement in the virtual scene. Typically, for headphone output, however, most implementations for computer games and/or mobile devices use generic filtering processes. These are especially vulnerable to difficulties when attempting to externalize the sound in front-back scenarios. The use of the term "externalize" in this context relates to the impression the listener has of the sound being some distance out from (either in front of, behind, above, or below) their own listening position. Front-back spatial sound through headphones usually results in sound sources being heard *inside* the head rather than being virtually projected out–the impression of depth would be realized were the sound sources virtually projected out.

The result arising from this situation is front-back confusion, something that can negatively impact on a listener's experience during gameplay or VR navigation, where characters or objects are heard momentarily in a different location from their visual origin.

In addition to front-back confusion, vertical sound localization remains a significant challenge in VR and computer games. Until a commercially viable method for obtaining individualized HRTF measurements and accurate real-time processing of head-tracking is achieved, these issues will continue to task developers who will have to rely on simpler approaches. The current method for obtaining an individual's HRTF is by measuring the right and left Head-Related Impulse Response (HRIR), which can then be convolved with any mono signal source. Essentially, the HRTF is the Fourier Transform of the HRIR. The HRTF measurements are usually undertaken in an anechoic chamber with specialized equipment. Most HRTF implementations in computer games and VR systems are derived from generic HRTF databases developed from a specialized human head manikin or derived from an average set of HRTFs taken from a particular population. The pinnae and head dimensions of the manikin head devices are procured from statistical data of average human biometrics. Of course, the disadvantage to this approach is that the player's ear and head shape may be very different from that of the manikin, which results not only in the lack of a true, individualized spatial experience, but in the distortion of spatial listening cues. However, recent research into novel ways of acquiring individualized characteristics of the human hearing system are being explored, paving the way forward to exciting developments in spatial audio for VR and computer games (Satoshi & Suzuki, 2008) (Otani & Ise, 2003).

Ambisonics

Ambisonics is a spatial audio system that was developed in the 1970s by Michael Gerzon and often touted as being superior in terms of spatial reproduction when compared to commercial domestic multichannel formats. It is a system that includes audio capturing techniques, the representation or encoding of the signal as a soundfield (referred to as 'B-format'), and the decoding of the signal during reproduction. Particular microphone patterns and positions, or a specially built microphone with four specifically arranged capsules (Soundfield Microphone), are required to capture the signal in a compatible way. The result is 4 signals conventionally referred to as X, Z, Y, W in first order Ambisonics:

- X = Front minus Back
- Z = Up minus Down
- Y = Left minus Right
- W = A non-directional reference signal. Front + Up + Left + Back + Down + Right.

Although considered one of the most advanced and realistic spatial reproduction systems available, the Ambisonics format has suffered from commercial setbacks. These can be attributed to bad timing in entering the marketplace, misleading associations with ill-fated quadraphonic techniques, and the lack of uptake by key music industry players during its development. However, recently the computer game industry and virtual reality researchers have stimulated renewed interest in Ambisonics.

Real-Time Processing

The real-time simulation of physical environments in computer games and VR systems requires a significant degree of real-time processing. Because of the complexity of this task, a certain amount of latency can be expected but limits must be placed on the extent of the latency so as not to thwart the user-experience. With current hardware, latency issues remain a concern, especially when attempting to reproduce a real-world scenario that requires rapid and consistent refresh rates of visual, aural, haptic, and motional events. Even within a sound-only game, the continuous motion of a sound source around the listener requires a significant amount of computation if its trajectory is to be smooth and uninterrupted. A balance needs to be struck and compromises are necessary. It is broadly accepted that update rates of 60 Hz and a total delay time of up to 50 msec are acceptable for acoustic virtual sound (Vorländer, 2008).

In many instances, a perceptual evaluation referred to as the Just-Noticeable-Difference (JND) is a useful instrument. This psychoacoustic evaluation procedure can be used for pitch differentiation or temporal differentiation, to name just a few. In terms of spatial sound, it is useful to know the accuracy of the human hearing system in differentiating between the same sound-source at different degrees in space. With this information, previous calculations that were not perceptually detected by the listener can now be disregarded. Some useful JND values in this regard allows for the reduction of redundant data, and resources can be used for other processes such as sound propagation. The performance of human listeners in point-to-point localization on the azimuth is most accurate in the frontal direction at 1º. On the left/right axis, the performance diminishes significantly to 10º. The rear direction is 5º and the JND on the vertical plane is the least accurate at 20º (Vorländer, 2008). Therefore, a computer game or VR system can effectively render spatial sound with diminished precision at certain locations around the listener.

Real-time binaural synthesis (the generation of spatial sound for headphone reproduction), is also a very resource demanding process, and interesting techniques are used to address the problem. One of the methods used is to preprocess binaural sound for reproduction when the listener reaches particular coordinates and orientation in the virtual room. A best-matched filter is applied as the listener's position changes. The key to this approach is to make the changeover from filter to filter as inaudible as possible. Fast convolution is required no matter which real-time binaural synthesis approach is taken. With many channels being processed in parallel, and with the continuous updating of listener position, convolution must be rapid and dynamic with no perceived artifacts during the many transitions. Multi-processing systems have aided in achieving more realistic rendering, as have techniques such as optimized fading between impulse response updates.

Head-tracking technology also introduces an amount of latency into a real-time processing system. Usually, the technologies employed are optical, inertial, mechanical, ultrasonic, and electromagnetic. However, new developments in terms of eye tracking and image recognition are being explored to reduce the amount of hardware encumbrance placed on the user. These techniques are also finding interesting applications in the computer game industry by taking advantage of the integrated webcam facilities built-in to most modern consumer computers.

Developing Spatial Sound Environments

So far in this chapter we have explored some of the key concepts in spatial sound and how they might apply to computer games and VR environments. The next section examines a number of key implementations of spatial sound. Where possible the emphasis is upon standardized implementations, such as MPEG-4, Java 3D, and OpenSL-ES, which are very stable, unlikely to change in the near-term, and have also informed the development of other implementations. There are also other implementations that are introduced by virtue of their prevalence within the industry.

Java 3D Sound API

Although the Java 3D API specification was originally intended for 3D graphics it has proved to be a suitable vehicle for the rendering of three-dimensional sound. It makes sense from a developer's point of view to keep all of the three-dimensional functionality within the same API set.

The implementation of spatial sound in the Java 3D specification employs a hierarchy of nodes that comprise:

- Sound node
- PointSound node
- ConeSound node
- BackgroundSound node

There are also two Java classes for defining the aural attributes of an environment. These are the Soundscape Node and the AuralAttributes Object. Each node is defined in a SceneGraph. The SceneGraph is a collection of nodes that constitute the three-dimensional environment. The application reads the nodes and their associated parameters from the SceneGraph and constructs the three-dimensional world with that information.

The BackgroundSound node is not a spatial sound rendering node. Its purpose is to facilitate the use of ambient background sounds within the Java application. The audio input to this node is normally a mono or stereo audio file.

Spatial Sound in Java 3D

The Sound Node itself does not address the spatial rendering of the sound source: this is accomplished in one of two ways. Firstly, by explicitly constructing the spatial attributes of the sound using either the PointSound Node or the ConeSound Node, or secondly, by configuring the acoustical characteristic of an environment using the Soundscape Node.

The first technique, constructing the spatial attributes, is dependent upon the type of sound source that is being used. If the sound source is a uniformly radiating sound (positional sound) then the PointSound node should be used, otherwise the developer should use the ConeSound node (directional sound).

Distance attenuation, as implemented in the Java 3D specification, employs distance attenuation arrays, which modify the amplitude of positional and directional sound sources, and also applies angular attenuation modifications to the amplitude of directional sound sources. When a sound object is created it has to be assigned an initialGain value: if this field is empty then the value defaults to 1.0 (where 1.0 is the maximum gain and 0.0 is the equivalent of a gain value of –60dB). In relation to the generic Sound Node, no distance attenuation is applied. This would seem to be a shortcoming in the specification as distance attenuation is one of the strongest cues in establishing depth perception for sound and should be accessible from the generic object. If the developer did not want the sound to have distance attenuation then he could simply leave the distance field blank.

The SoundScape node (refer to Figure 9) configures the acoustical properties of the listener's environment. An unlimited number of SoundScape nodes can be contained within a scene. The defined SoundScape node region determines which sets of acoustical properties are to be used. As a result of being able to specify several SoundScape node regions, one can generate a number of aural environments within the scene. For instance, within the one scene there could be three rooms, each with a different acoustical signature. Alternatively, with more detailed scene description, one could set up a number of acoustical regions within a single room using a number of SoundScape nodes.

The acoustical properties, that is, reverberation and atmospheric attributes of the SoundScape node, are specified in the AuralAttributes Object. The AuralAttributes Object is a component object of the SoundScape Node. It specifies the following aural properties: reverberation, Doppler effect, distance frequency filtering, and atmospheric rolloff. Table 1 contains a list of the parameters and their default values for when an AuralAttribute Object is first constructed.

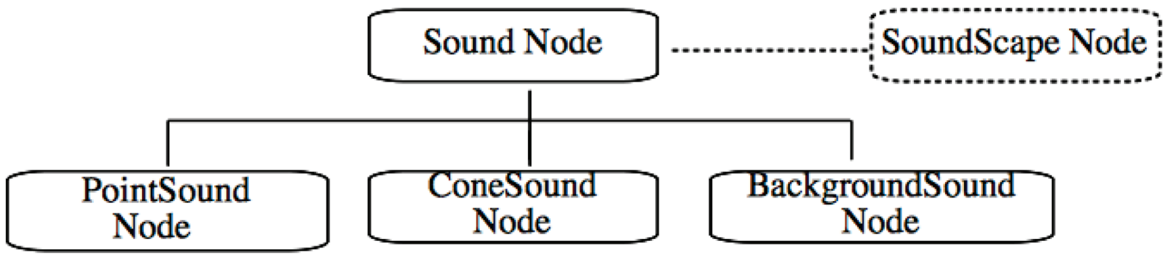
The AuralAttributes node describes reverberation with three components: delay time, reflection coefficient, and feedback loop. Delay time is used to calculate the amount of time taken for the sound to reach the listener having undergone one reflection. This component is either set explicitly or implied by the bounding regions of the volume. Note that the bounding region is not necessarily the same as the region specified for the Soundscape Node. Delay time is measured in milliseconds.

The reflection coefficient is used to determine the attenuation factor for the sound. The reflection coefficient(s) represent the reflective or absorption properties of the environment. A value of 1.0 represents an un-attenuated sound and a value of 0.0 represents a sound that has been fully absorbed. These coefficients are applied as a uniform attenuation across the spectrum. This is not a very refined scheme as most reflective/absorptive materials alter the spectrum of the sound in a non-uniform manner. Using the specification's present implementation of reflection/absorption, there would be very little timbral difference between a sound that has been reflected by plaster and one that has been reflected by a metallic object.

The final component, feedback loop, specifies the number of times a sound is reflected or the order of reflection. If the feedback loop has a value of 0.0 no reverberation is performed, if it is set to one then the listener hears an echo and, if it is set to –1.0 the reverberation will continue until the amplitude of the signal dies to –60dB. This is known as *effective zero* in the Java 3D specification. Effective zero relies upon a –6dB drop in gain for every doubling of distance (inverse square law). Values between 0.0 and 1.0 refer to the number of iterations of the loop.

The parameter attributeGain is used to alter the speed of sound in order to mimic the effects of atmospheric change. The default value is 0.344 meters per millisecond and this refers to the speed of sound at room temperature. This value is then altered by the gain scale value specified by the developer. A value greater than 1.0 will increase the speed of sound and conversely a value less than 1.0 will decrease the speed of sound.

Figure 9. Sound node hierarchy in the Java 3D specification



The Doppler effect is achieved by taking the value of the speed of sound, multiplying it by the velocity scale factor (which is the change of speed relative to the listener's position) and then proportionally applying the frequency scale values. If the velocity scale factor is 0.0, no Doppler effect is processed.

For further information on the Java 3D API the reader is referred to Murphy (1999a), Murphy(1999b), Murphy and Rumsey (2001), and Murphy and Pitt (2001).

XNA/XACT Implementation of Surround Sound

XNA/XACT is a collection of game development technologies produced by Microsoft for the various Windows platforms (including the Xbox console) . In relation to the current mass-market, 5.1 channel sound is the standard deployment for multichannel sound output. This consists of a left, center, right, left surround, right surround (5) and Low-Frequency Enhancement (.1). 7.1 sound is also popular and 10.2 (see figure 10) is the emerging next generation of surround speaker setup. Work is also being done on 22.2-surround sound. Games generally follow either a 2.0 (stereo) or 5.1-channel standard.

5.1-surround sound is effective to a point. However, it is essentially a coarse representation of the spatial sound field. Stepping between the speakers, as sound moves dynamically from frontal speakers to surrounds, can be quite audible. A natural sound (that emanates directly from an acoustic sound source and not via speakers) retains its timbral characteristics despite colouration from

Table 1. AuralAttribute Object Properties

Parameter	Default Value
attributeGain	1.0
rolloff	1.0
reflectionCoeff	0.0
reverbDelay	0.0
reverbBounds	null
reverbOrder	0
distanceFilter	null
frequencyScaleFactor	1.0
velocityScaleFactor	1

room acoustics and HRTFs. As it dynamically moves from front to surround positions around the listener's head, the sound is filtered in a number of ways. However, the listener can distinctly recognize that the sound is from the same source as it retains its fundamental timbral characteristic.

With exact matching speakers, therefore, one would expect the same would hold true with 5.1-surround sound, since their discrete position in space is simply like a natural sound source traveling around the listener's head. Unfortunately, this is not the case as 5.1-surround is a system that represents too few steps in the sound field. Therefore, sound designers need to compensate for timbral instabilities by equalizing the signal as it reaches the surround speakers. This is a difficult task, as sound designers cannot predict the type of room the listener will have their gaming system in.

MPEG-4 and Spatial Sound

MPEG (Motion Picture Experts Group) is a working group of an ISO/IEC subcommittee that generates multimedia standards. In particular, MPEG defines the syntax of low-bitrate video and audio bit streams, and the operation of codecs. MPEG has been working for a number of years on the design of a complete multimedia toolkit, which can generate platform-independent, dynamic, interactive media representations. This has become the MPEG-4 standard.

In this standard, the various media are encoded separately allowing for better compression, the inclusion of behavioral characteristics, and user-level interaction. Instead of creating a new Scene Description Language (SDL) the MPEG organization decided to incorporate Virtual Reality Modeling Language (VRML). VRML's scene description capabilities are not very sophisticated so MPEG extended the functionality of the existing VRML nodes and incorporated new nodes with advanced features. Support for advanced sound within the scene graph was one of the areas developed further by MPEG. The Sound Node of MPEG-4 is quite similar to that of the VRML/Java 3D Sound Node. However, MPEG-4 contains a sound spatialization paradigm called Environmental Spatialisation of Audio (ESA). ESA can be divided into a Physical Model and a Perceptual Model.

Physical Model (see Table 2): This enables the rendering of source directivity, detailed room acoustics and acoustic properties for geometrical objects (walls, furniture, and so on.). *Auralization*, another term for realisation of the physical model, has been defined as: "creating a virtual auditory environment that models an existent or non-existent space" (Väänänen, 1998).

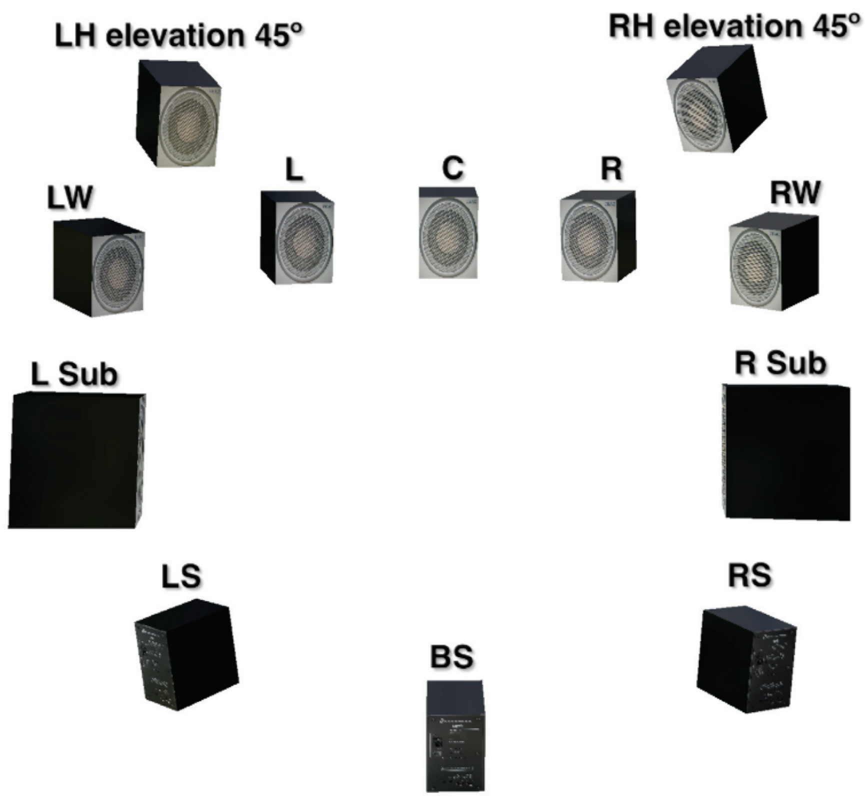
Three Nodes have been devised to facilitate the physical modelling approach. These are AcousticScene, AcousticMaterial and DirectiveSound.

Briefly, DirectiveSound is a replacement for the simpler Sound Node. It defines a directional sound source whose attenuation can be described in terms of distance and air absorption. The direction of the source is not limited to a directional vector or a particular geometrical shape.

The velocity of the sound can be controlled via the speedOfSound field: this can be used, for example, to create an instance of the Doppler effect. Attenuation over the distance field can now drop to -60dB and can be frequency-dependent if the useAirabs field is set to TRUE. The spatialize field behaves the same as its counterpart in the Sound Node but with the addition that any reflections associated with this source are also spatially rendered. The roomEffect field controls the enabling of ESA and, if set to TRUE, the source is spatialized according to the environment's acoustic parameters.

AcousticScene is a node for generating the acoustic properties of an environment. It simply establishes the volume and size of the environment and assigns it a reverberation time. The auralization of the environment involves the processing of information from the AcousticScene and the

Figure 10. 10.2 enhanced surround sound. The ITU-R BS 775-1 standard for 5.1-surround is L, C, R, LS, RS and one Sub. 10.2-surround expands this by adding an extra Sub, left and right elevated speakers, back surround, wide left and wide right speakers. L = Left. C = Center. R = Right. LW = Left Wide. RW = Right Wide. LH = Left Height. RH = Right Height. L Sub = Subwoofer Left Side. R Sub = Right Subwoofer Right Side. LS = Left Surround. RS = Right Surround. BS = Back Surround



acoustic properties of surfaces as declared in AcousticMaterial.

Perceptual Model (see Table 3): Version 1 of the MPEG-4 standard only rendered spatial sound based upon physical attributes, that is, geometric properties. However, virtual and synthetic worlds are not constrained by physical laws and properties: it became necessary to introduce a perceptual equivalence of the physical model. To this end, two new nodes were added in version 2 of MPEG-4: PerceptualScene and PerceptualSound. Rault, Emerit, Warusfel, and Jot (1998) highlighted the merits of the perceptual approach in a document to the MPEG group:

A first advantage we see in this concept is that both the design and the control of MPEG4 Scenes is more intuitive compared to the physical approach, and manipulating these parameters does not require any particular skills in Acoustics. A second advantage is that one can easily attribute individual acoustical properties for each sound present in a given virtual scene.

The principle elements of the perceptual model are drawn from research undertaken by IRCAM's *Spatialisateur* project, and additional features are derived from Creative Lab's Environmental Audio Extensions (EAX) and Microsoft's DirectSound

Table 2. MPEG-4, V2. Advanced Audio Nodes–Physical Nodes

Node	Field
AcousticScene	params
	3DVolumeCenter
	3DVolumeSize
	reverbtime
AcousticMaterial	reffunc
	transfunc
	ambientIntensity
	diffuseColor
	emissiveColor
	shininess
	specularColor
	transparency
DirectiveSound	direction
	intensity
	directivity
	speedOfSound
	distance
	location
	source
	useAirabs
	spatialize
	roomEffect

API (Burgess, 1992). Using the perceptual model, each sound source's spatial attributes can be manipulated individually, or an acoustic-preset can be designed for the environment.

Fields such as Presence, Brilliance, and Heavyness are used to configure the room/object's acoustic characteristics. In all, there are 9 fields used to describe, in non-technical terms, the spatial characteristics of a room or a sound object. These fields have been derived from psycho-acoustic experiments carried out at IRCAM (the *Spatialisateur* project). Of the 9 subjective fields, 6 describe perceptual attributes of the environment and 3 are perceived characteristics of the source. Table 1 lists the parameters for both Environment and Source.

It can also be seen from Table 4 that the last 3 fields of the Environment section and all of the Source fields are dependent upon the position, orientation, and directivity of the source. The validity of this approach could be questioned in terms of its subjectivity, for example, the choice of words such as Warmth and Brilliance. However, the use of subjective terms as acoustic parameters, in this context, is to facilitate the non-specialist to compose a soundscape with convincing acoustic properties. This effectively opens up the complex world of acoustics to the non-specialist. For further information on MPEG spatial sound the reader is referred to Murphy (1999a), Murphy (1999b), Murphy and Rumsey (2001), and Murphy and Pitt (2001).

Table 3. MPEG-4 version 2. Advanced Audio Nodes–PERCEPTUAL NODES

Node	Field
PerceptualScene	AddChildren
	RemoveChildren
	BboxCenter
	UseAirabs
	UseAttenuation
	RefDistance
	Latereverberance
	Heavyness
	Liveness
	RoomPresence
	RunningReverberance
	RoomEnvelopment
	Presence
	Warmth
	Brilliance
	Fmin
	Fmax
PerceptualSound	direction
	intensity
	directivity
	omniDirectivity
	speedOfSound
	distance
	location
	relParams
	directFilter
	inputFilter
	useAirabs
	useAttenuation
	spatialize
	roomEffect
	source

Some Considerations

Head Tracking is an important tool in a dynamic virtual environment. Apart from the obvious advantages it brings to the visual presentation it is also important in the spatial rendering of sound. According to Burgess: “The lack of these [head-related] cues can make spatial sound difficult to use. We tend to move our heads to get a better sense of a sound’s direction. This ‘closed-loop’ cue can be added to a spatial sound system through the use of a head-tracking device” (Burgess, 1992).

Table 4. Perceptual Fields for MPEG-4 Spatial Audio

Environment Fields	**Source Fields**
LateReverberance	Presence
Heavyness	Warmth
Liveness	Brilliance
RoomPresence	
RunningReverberance	
RoomEnvelopment	

Recent research has shown that the use of Head Tracking reduces front-back reversals by a ratio of 2:1 (Blauert, 1997) and there is evidence that it assists in the externalisation of sources that would otherwise be located 'inside-the-head'. Another area where Head Tracking is helpful is in the simulation and control of the Doppler Effect and to resolve source-listener movement ambiguities. Blauert (1997) terms this "persistence":

In connection with spatial hearing, the term 'persistence' refers to the fact that the position of the auditory event can only change with limited rapidity. Under appropriate conditions the position of the auditory event exhibits a time lag with respect to a change in position of the sound source. Persistence must always be taken into consideration when using sound sources that change position rapidly. (p. 47)

Virtual Spatial Audio On Mobile Systems

Although not widely known, there have been a number of 3D audio solutions available for mobile devices for a number of years. Despite this, manufacturers have been quite slow in implementing Operating System (OS) and hardware support for these audio APIs or only offer limited support on a select number of devices. As a consequence, third-party developers cannot rely on 3D audio effects for their mainstream applications as it is too unreliable in a programming environment, requiring compatibility with a wide variety of device models. Therefore, there seems to be an unusual scenario whereby 3D audio on mobile phones continues to be of extreme interest to interface researchers and API developers, but the practical implementation of the technology is stagnant in terms of mainstream consumerism. This is set to change, however, with the emergence of faster wireless networks, more powerful mobile operating systems and the establishment of digital media broadcast standards for handheld devices (such as DVB-H).

JSR-234

JSR-234, or the Advanced Multimedia Supplements (AMMS), is an API initiated by Nokia and developed under the Java Community Process. It allows for more control over multimedia elements, including the creation of 3D audio environments. It is an optional supplement to the Mobile Media API (MMAPI, JSR-135) designed for J2ME/CLDC mobile devices. Refer to Sun (2010), Li (2005) and Goyal (2006) for more information on CLDC and MIDP specifications.

MMAPI is itself an optional low-footprint API, implemented in MIDP 2.0, allowing developers to create Java applications to playback and capture audio and video in a variety of multimedia file formats, perform camera operations, stream radio over a network, generate musical tones and so forth. A large number of mobile phone devices support MMAPI, but this is reduced when it comes to fully implementing AMMS.

It is important to point out that these multimedia APIs are not part of the actual mobile phone OS: they do, however, enable developers to create third-party applications in the form of a MIDlet (.jar and .jad files). Therefore, whilst developers cannot create 3D audio menu items for the phone's OS, it provides an excellent platform for testing the psychophysical considerations of such an interface in preparation for future spatial audio implements in the core OS.

In JSR-234, a source-medium-receiver model (Paavola & Page, 2005) is employed (see Figure 11). This approach goes beyond the capabilities of MMAPI.

Using MMAPI alone, sourced audio data can be started, stopped, paused, and primitively controlled. A summary of the MMAPI process is as follows: the abstract class, DataSource, locates the audio content; a Manager class creates the appropriate Player interface; and the Player in turn incorporates control methods for rendering and primitively controlling the audio content (see Figure 12). Control methods include VolumeControl, ToneControl, PitchControl, StopTimeControl, RecordControl and RateControl. No spatial attributes are capable if using MMAPI alone and MMAPI's Manager class cannot be expanded to include this service.

In order to fulfil the requirements of spatial audio rendering, AMMS was created to extend MMAPI. Of interest to us in the AMMS package are GlobalManager, Spectator, and Module. The GlobalManager class is similar in action to the MMAPI Manager class, but is also very different in what interfaces it creates. Therefore, it does not extend or replace the Manager class and the MMAPI Manager is still required to create Players (see Figure 13). The GlobalManager handles the creation of Module interfaces and allows access to the Spectator class.

The Modules implemented via the GlobalManager are the EffectsModule and the SoundSource3D module. In contrast to MMAPI, AMMS allows several Players to be assigned to one audio effect (that is, mixing several Player instances). This allows common effects to be applied to all Players and this helps to optimize effects on limited resources. The types of control effects possible using the EffectsModule interface are equalization, panning, virtualization, reverb, and chorus/flanger (Paavola & Page, 2005).

Figure 11. An overview of JSR-234

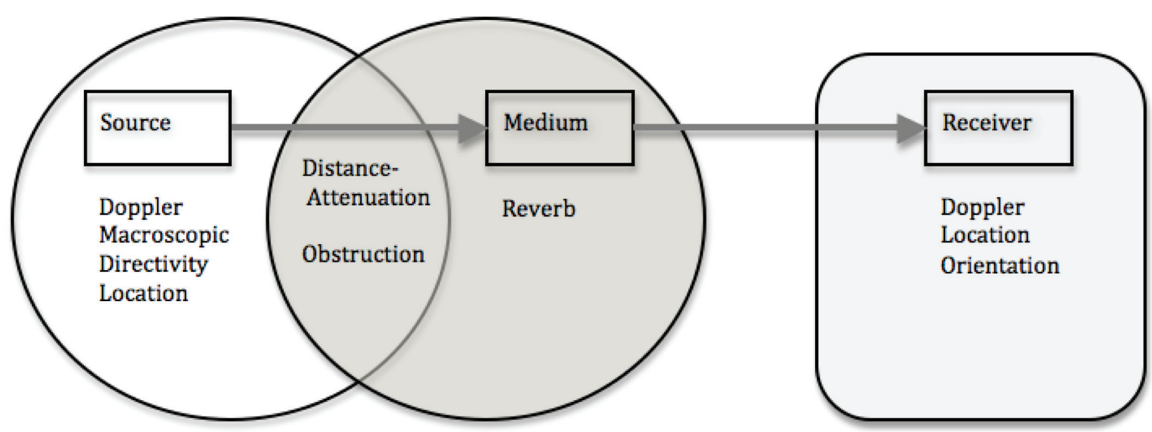
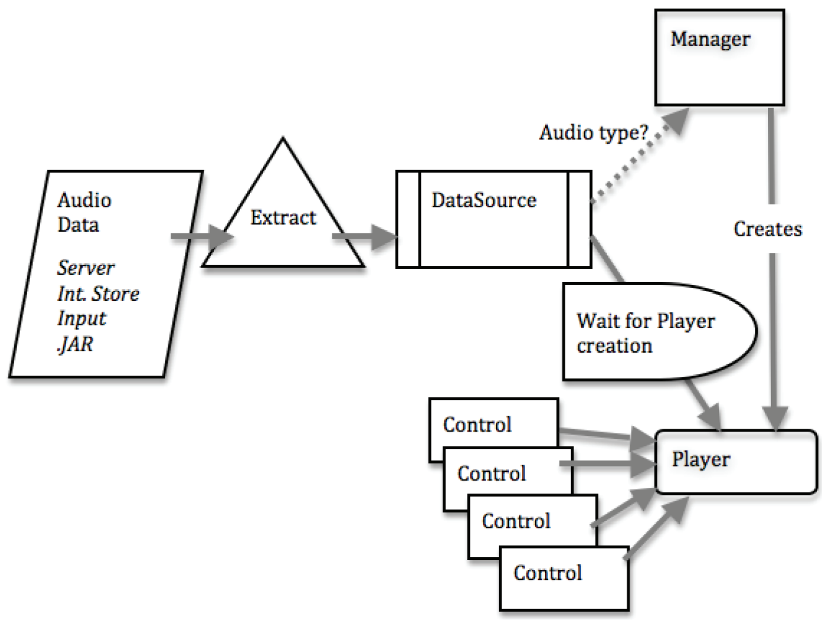


Figure 12. Basic MMAPI process. The Manager class bridges between the DataSource class and Player interface. Primitive controls are implemented via the Player's methods



The SoundSource3D module performs control effects specifically relating to positioning sources in the virtual space. These include sound source directivity (sound cones), distance attenuation, Doppler shift, location control, specifying the dimensions of an extended sound source (not a point source), obstruction of the sound traveling from source to listener, and reverberation effects (JSR-234 Group, 2005; Paavola & Page, 2005).

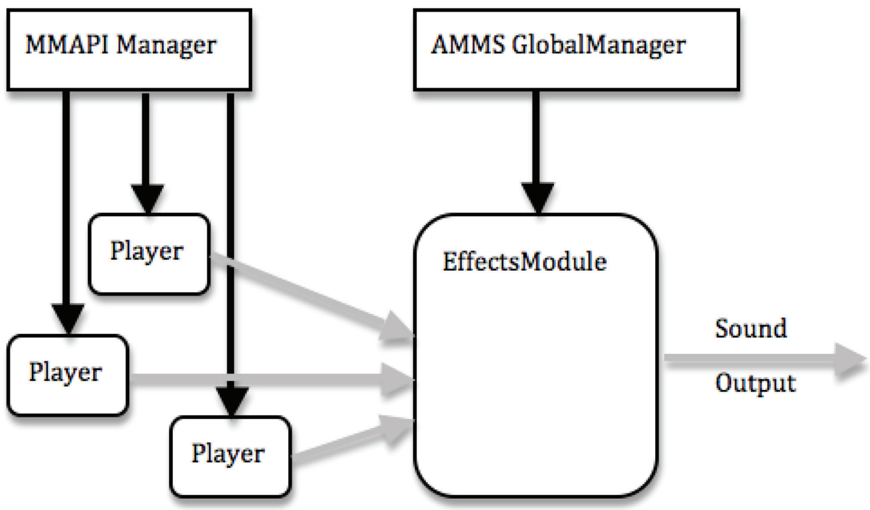
The Spectator class represents the listener in the virtual space (JSR-234 Group, 2005). As with other sound sources in the virtual space, the listener/spectator must also possess spatial cues. The controls associated with the Spectator class are location, Doppler, and orientation.

OpenSL ES

OpenSL ES is an open standard API for interactive spatial audio for embedded systems developed by the Khronos Group (2009). Target devices for OpenSL ES are basic mobile phones, smart phones, PDAs and mobile digital music players. It is a C-language audio API with some overlap with OpenMAX AL 1.0, a multimedia/recording API for embedded systems from the same group. Like the relationship between MMAPI and AMMS, OpenMAX AL has basic audio capabilities, with OpenSL ES providing advanced 3D audio and sonic effects, and both share many common methods. However, unlike MMAPI and AMMS, both OpenMAX AL and OpenSL ES are entirely independent and each can perform as a standalone API on target devices.

Three Profiles are present in the OpenSL ES implementation: **Phone**, **Music**, and **Game** (see Table 5). Different audio capabilities exist for different profiles but manufacturers are free to implement two or all three profiles on their devices. All features of a given profile have to be implemented by the manufacturer in order to ensure compatibility. Therefore, a manufacturer implementing the Phone profile, but wanting to

Figure 13. MMAPI Manager creates Players. AMMS GlobalManager creates Modules that Players hook into



incorporate elements of the Game profile, must fully implement the Game profile also.

CONCLUSIONS AND FUTURE DEVELOPMENTS

Spatial presentation of sound is a very important feature of VR and becoming more important in computer games. Without spatial sound, virtual environments would lack the complex qualities required for a convincing immersive experience. The development of synthetic sound spatialization techniques for immersive environments has lagged behind comparible visual technology. However, there now exists a number of options for developers who wish to incorporate spatial sound into computer games and VR.

Java 3D provides a collection of tools that enable developers to integrate spatial sound in a virtual environment. While the API set allows for the construction of reasonable spatial sound experiences, there are a number of shortcomings which hinder the advancement of Java 3D for spatial sound environments: most notable of these are the lack of support for HRTFs and issues around real-time processing in Java.

MPEG-4 version 2 is where the main breakthroughs in the integration of spatial sound in an international standard have been achieved. To cater for both ESA and absolute sound rendering, a dual approach has been developed. This dual approach of both physical and perceptual descriptions of spatial sound seems to encapsulate all of the necessary attributes for a cogent spatial experience.

XNA opens up spatial sound for game development. While not an open platform, unlike OpenAL, the development environment is accessible and fits neatly into the XNA architecture of computer game development.

The future holds some remarkable potential for spatial sound in computer games: the designer's imagination being the only limiting factor. Imagine being immersed completely in sound, where gameplay relies heavily on the sense of hearing. Walking down a dark corridor in a first-person shooter, hearing your footsteps below you, en-

Table 5. The table shows audio-only information for the profiles associated with OpenSL ES. MIDI specifications have been excluded from the table. Adapted from Khronos Group (2009)

API Feature	Phone	Music	Game
PLAYBACK/PROCESSING CONTROLS			
Play multiple sounds at same time	YES	YES	YES
Playback mono & stereo	YES	YES	YES
Basic playback controls	YES	YES	YES
End-end looping	YES	YES	YES
Partial looping	NO	NO	YES
Set playback position	YES	YES	YES
Position-related notifications	YES	YES	YES
Sound prioritization	YES	YES	YES
Audio to several concurrent outputs	YES	NO	NO
Volume Control	YES	YES	YES
Audio balance & pan control	NO	YES	YES
Metadata retrieval	NO	YES	YES
Modify playback rate & pitch	NO	NO	YES
Play sounds from secondary store	YES	YES	YES
Buffer/queues	NO	NO	YES
CAPABIILITY QUERIES			
Query capabilities of implementation	YES	YES	YES
Enumerate audio I/O devices	YES	YES	YES
Query audio I/O device capabilities	YES	YES	YES
EFFECTS			
Stereo widening	NO	YES	YES
Virtualization	NO	YES	YES
Reverberation	NO	YES	YES
Equalization	NO	YES	YES
Effect control	NO	YES	YES
3D AUDIO			
Positional 3D audio	NO	NO	YES
Sound cones	NO	NO	YES
Multiple distance models	NO	NO	YES
Source & listener velocity	NO	NO	YES
Source & listener orientation	NO	NO	YES
3D sound grouping	NO	NO	YES
Simultaneous render of multiple 3D controls	NO	NO	YES

vironmental sounds coming from air ducts and doorways, suddenly, you hear a noise behind and to your left, you turn to be confronted by a ghastly beast who wants you for lunch. You fire your weapon, the piercing impact of the firing mechanism on your ears, the sound reverberating and interacting with the room, shell casings tinkling on the floor, and the creature falls to the ground with a resonating thud. The application of spatial sound will advance and drive the narrative and drama in a computer game, and ultimately lead to an immersive user experience.

REFERENCES

Ashmed, D. H., & Wall, R. S. (1999). Auditory perception of walls via spectral variations in the ambient sound field. *Journal of Rehabilitation Research and Development*, *36*(4).

Begault, D. R., & Wenzel, E. M. (1993). Headphone localization of speech. *Human Factors*, *35*, 361–376.

Blauert, J. (1997). *Spatial hearing: The psychophysics of human sound localization* (rev. ed). Cambridge, MA: MIT Press.

Burgess, D. (1992). Techniques for low cost spatial audio. In *Proceedings of the 5th annual ACM symposium on User interface software and technology.*

Duda, R. O., Algazi, V. R., & Thompson, D. M. (2002). The use of head-and-torso models for improved spatial sound synthesis. In *Proceedings of the 113th Audio Engineering Society Convention.*

Everest, F. A. (1997). *Sound studio construction on a budget*. City, ST: McGraw-Hill.

Everest, F. A. (2001). *Master handbook of acoustics*. City, ST: McGraw-Hill.

Fletcher, T. D. R. N. H. (2004). *Principles of vibration and sound* (2nd ed.). New York.

Goyal, V. (2006). *Pro Java ME MMAPI: Mobile media API for Java Micro Edition*. City, CA: Apress Press Inc. JSR-234 Group. (2005). *Advanced multimedia supplements API for JavaTM2 Micro Edition*. Nokia Corporation.

Khronos Group. (2009). *OpenSL ES specification*. The Khronos Group.

Li, S., & Knudsen, J. (2005). *Beginning J2METM platform: From novice to professional* (3rd ed.). City, CA: Apress Press Inc.

Mark, F., Bear, B. W. C., & Paradiso, M. A. (2007). *Neuroscience —Exploring the brain* (3rd ed.). City, ST/Country: Lippincott Williams & Wilkins.

Murphy, D. (1999). *A review of spatial sound in the Java 3D API specification*. Institute of Sound Recording, University of Surrey.

Murphy, D. (1999). Spatial sound description in virtual environments. In *Proceedings of the Cambridge Music Processing Colloquium.*

Murphy, D., & Pitt, I. (2001). Spatial sound enhancing virtual story telling. *Springer Lecture Notes In Computer Science, 2197*.

Murphy, D., & Rumsey, F. (2001). A scalable spatial sound rendering system. In *Proceedings of the 110th AES Convention*.

Otani, M., & Ise, S. (2003). A fast calculation method of the head-related transfer functions for multiple source points based on the boundary element method. *Acoustical Science and Technology*, *24*(5), 259–266. doi:10.1250/ast.24.259

Paavola, M. K. E., & Page, J. (2005). 3D audio for mobile devices via Java. In *Proceedings of the AES 118th Convention*.

Rault, J. B., Emerit, M., Warusfel, O., & Jot, J. M. (1998). Audio rendering of virtual room acoustics and perceptual description of the auditory scene. *TCI/SC29/WG11*.

Satoshi Yairi, Y. I., & Suzuki, Y. (2008). Individualization of Head-Related Transfer Functions based on subjective evaluation. In *Proceedings of the 14th International Conference on Auditory Displays*.

Sun Microsystems. (2010). *Java ME API*. Retrieved February 4, 2010, from http://java.sun.com/javame/reference/apis.jsp.

Väänänen, R. (1998). Verification model of advanced BIFS (systems VM 4.0 subpart 2). *ISO/IEC JTCI/SC29/WG11*.

Vorländer, M. (2008). *Auralization—Fundamentals of acoustics, modelling, simulation, algorithms and acoustic virtual reality* (1st ed.). Berlin: Springer.

KEY TERMS AND DEFINITIONS

API (Application Programming Interface): A mechanism in software engineering to allow two separate pieces of software to integrate or interface, for instance a library of functionality being integrated into an application.

Doppler Effect: The apparent shift in pitch/frequency of a sound due to motion parallax.

Fast Convolution: Convolution is an operation applied to two signals, the most common operation being multiplication. A Fast Convolution is an optimised circular convolution, typically used in conjunction with a Fast Fourier Transform.

Free-Field: An open space/environment which does not interact with the sound source (as opposed to 'room interaction' in a closed space.

Head Tracking: The tracking of position and orientation of head movement by an external sensor in VR or computer games.

HRTF: Head Related Tranfer Function.

Occlusion: An obstacle that blocks the effective transmission of sound by absorbing energy, or reflecting sound waves.

ENDNOTE

1 Individualization refers to a listening experience that is tailored to and unique to the individual listener.