Geometrical Room Acoustics

Ray Based Simulations for Room Acoustics

Jaume Durany Vendrell

TESI DOCTORAL UPF / ANY 2016

DIRECTOR DE LA TESI
Dr. Adan Garriga Torres
Departament de Tecnologies de la Informació i les Comunicacions

upf. Universitat Pompeu Fabra Barcelona

Als meus pares i la meva germana... per ser-hi
sempre.

A tu, Feli... per inspirar-me a créixer
i ser millor persona.

Acknowledgements

The people that surround us during the journeys we choose are an essential part of our life and define the way we enjoy each success and face every challenge.

In 2004, 12 years ago, I met Dr. Adan Garriga when he started teaching the Modelling and Simulation course at Universitat Pompeu Fabra. His energy and positivism inspired me since the very beginning and that was one of the main reasons why I started this research with him. He has been one of the most significant persons in my professional development, always happy to support me in everything. Thanks, Adan, for your invaluable guidance and mentorship throughout all this years we have been working together, it has been a deep pleasure and honor to share all this with you.

I would also like to express profound gratitude to Dr. Toni Mateos for his support and supervision during the most arduous part of our research, he led us through what is probably the most significant contribution of this thesis. His contagious addiction to experimenting and innovating is something I have always enjoyed. Thanks Toni!

And this thesis wouldn't have been the same without Dr. Carlos Spa. We started our research together and his support and presence was key to enjoy and learn from every step we made, thanks Charlie!

I am also truly thankful to Jordi Arqués, Daniel Arteaga, Pau Arumí, Giulio Cengarle, David García, Natanael Olaiz and Ferran Orriols for their help and discussions, they all guided me and contributed in some way to the direction of my thesis.

I am as ever, especially grateful to my parents, Jaume and Maria Rosa, for their love and support throughout my life. I also wish to thank my sister, Mireia, together with Fran, for their encouragement and understanding during my study. Without them any of this wouldn't have been possible.

Finally, my sincere thanks go to Felicita. You keep reminding me that life can be anything we want it to be if we pursue our passions and dreams.

Abstract

Room acoustics is the science devoted to study sound propagation in enclosures where the sound conduction medium is bounded on all sides by walls, ceiling and floor. The acoustic information of any room, the so-called impulse response, is expressed in terms of the acoustic field as a function of space and time. The analytical formulation of the sound variables distribution is, in general, extremely hard to obtain and there only exist solutions of very simple and unrealistic scenarios. Therefore the use of computers for solving this type of problems has emerged as a proper alternative to calculate impulse responses.

In this Thesis we focus on the use of the ray-based methods to compute impulse responses. More precisely, we present the design and implementation of a sound ray tracing engine that computes the impulse response in any given environment not only for the pressure but also for the velocity vector of the acoustic field. With this extra information we have all the necessary data to model the propagation of sound and we can then naturally spatialize the sound to any speakers layout.

This research contributes to the main aspects in the computation of impulse responses using a ray-based approach. The presented ray tracing engine includes a method developed to apply the analytical solution for the *Acoustic Bidirectional Reflectance Distribution Function (A-BRDF)* in the *Vector Based Scattering Model (VBS)*, which reduces dramatically the computational cost.

Resum

L'acústica de sales és la ciència encarregada d'estudiar la propagació del so en entorns tancats. La informació acústica de qualsevol entorn, coneguda com la resposta impulsional, s'expressa en termes del camp acústic com una funció de l'espai i el temps. La formulació analítica de la distribució de les variables del so és, en general, extremadament complexa d'obtenir i només existeixen solucions d'escenaris molt simples i irreals. Per tant, l'ús d'ordinadors per solucionar aquest tipus de problemes ha emergit com una alternativa adequada per calular funcions de resposta.

En aquesta Tesi ens hem centrat en l'ús de mètodes basats en rajos per calcular funcions de resposta. Més concretament, presentem el disseny i la implementació d'un motor de traçat de rajos que calcula funcions de resposta en cualsevol entorn virtual, obtenint no només la funció de resposta per la presió sinó també pel vector de velocitats del camp acústic. Amb aquesta informació extra tenim totes les dades necessàries per modelar la propagació del so i podem de forma natural espacialitzar un so per qualsevol configuració d'altaveus.

Aquesta recerca contribueix als aspectes principals del càlcul de funcions de resposta utilitzant mètodes basats en rajos. El motor de traçat de rajos que presentem inclou un métode desenvolupat per aplicar la solució analítica de la *Funció de Distribució Acústica de Reflectància Bidireccional (A-BRDF)* al *Model de Dispersió Basat en Vectors (VBS)*, fet que redueix molt notablement el cost computacional.

Preface

In general, the sound distribution produced by the propagation of the acoustic field within a room is strongly related to the geometry of the enclosure and the absorbing properties of the materials. In fact, one can perceive the loudness and the echo of a specific room, leading to a particular distribution of the acoustic variables which depend on position and time.

The acoustic phenomena in rooms are very complex and in general it is extremely hard to find analytical expressions for the complete characterization of the acoustic field. For this reason, the use of computers for predicting the acoustic field in rooms emerges as an important contribution in the design of room acoustics.

Nowadays, the room acoustics simulations of real and virtual environments play an important role in different fields. As an example, for giving to architects a tool capable to predict with high accuracy the acoustic features of any room before it is constructed. Similarly, these applications can be used for improving the acoustics of built rooms, such as theaters, auditoriums or concert halls.

In any multimedia production the final exhibited sound is acquired from an initial recorded or synthesized sound that is processed with a set of very diverse filters in order to reduce its imperfections and/or enhance some of its qualities. One of the major objectives in this process is to obtain a sound in agreement with the visual information that the audience is receiving simultaneously. For example, if the audience is visualizing a palace in a movie, they expect that all the audio in that scene sound as if it was recorded in that place, and what is actually done is to filter the sound to add the necessary echoes to obtain a convincing result.

The objective is to try to automatize the process in such a way that the processing of the sound becomes easier, and to obtain a resulting sound not only satisfactory but also accurate. The way to obtain it is to create a tool capable of building a filter with the concrete features of a given environment to then process the sound with it.

Thus, computer room acoustics simulations are an important tool for predicting and improving the acoustics quality of built or unbuilt rooms. Moreover, the computer game and entertainment industries together with the training simulators based on virtual reality technologies need, day after day, to find more realistic auditive sensations in order to create a highly immersive environment with spatial attributes. Therefore, in this document the improvement in room acoustics simulations is treated according to the necessities of the industry of these days.

Contents

List of Figures

Introduction

Motivation and Scope

Room acoustics is the science devoted to study sound propagation in enclosures where the sound conduction medium is bounded on all sides by walls, ceiling and floor.

These room boundaries usually reflect a certain fraction of the sound energy inciding on them. Another fraction of the energy is absorbed, either by conversion into heat or by being transmitted to the outside by the walls.

The direct sound that comes straight from a number of sources is combined with the numerous reflected and diffracted components in the enclosure. This resulting combination is known as the acoustics of a room and also for the complexity of the sound field in a room [Kuttruff (1999)].

The analytical formulation of the sound variables distribution is, in general, extremely hard to obtain and there only exist solutions of very simple and unrealistic scenarios. Therefore, it is necessary to use computers for predicting proper solutions of the sound distribution [Schroeder (1973)].

Usually, two main groups of numerical methods are considered to classify computer room acoustic simulations [Savioja et al. (1999)]: wave-based and geometrical methods. The first group embraces a set of several techniques to numerically solve the partial differential equations that govern the sound propagation in rooms. Frequency-based techniques such as Finite Element Methods [Wright (1995); Savioja et al. (1996a)] and Boundary Element Methods are part of this first group, together with time-based alternatives such as Finite Difference in Time Domain (FDTD) [Botteldoren (1994); Savioja et al. (1996b)] and Digital Waveguide Mesh methods (DWM) [Murphy et al. (2007)]. Although all of them are computationally demanding they have gained interest during the last years because of the growing capabilities of standard computers. Wave-based methods are the best option to find a solution with a high degree of accuracy.

In the second group, geometrical methods such as ray tracing [Krokstad et al. (1968)], image source methods [Allen and Berkley (1979); Borish (1984)], beam tracing methods [Funkhouser et al. (2004, 1998); Laine et al. (2009)] and radiosity [Nosal et al. (2004a); Siltanen et al. (2009)] are the most often used modeling techniques which are based on the geometrical theory of acoustics [Morse and Ingard (1986)]. This approach assumes that sound wavelenghts are significantly smaller than the size of obstacles found in the room thus motivating the application of the advances already achieved in graphics rendering during the last decades to render audio.

Nowadays, the use of geometrical methods in computational acoustics modeling is a common practice in many different disciplines: the acoustic design of buildings such as auditoria or concert halls [Rindel (2000)], outdoor acoustics [Salomons (2001)], audio post production in Digital Cinema [IP-Racine (2006)] or audio for video games and other interactive applications [Funkhouser (2002); Savioja (1999)].

Some of the mentioned geometrical methods (sometimes in combination) are the basis of many room acoustics commercial softwares such as *Odeon* [Odeon (2015)], *Catt-Acoustics* [Catt (2015)] or *Ramsete* [Ramsete (2015)].

It is self-evident that geometrical methods can reflect only a partial aspect of the acoustical phenomena occurring in a room, neglecting phenomena such as difraction. However, the fact that no existing method is exact or 100% accurate positions geometrical methods as an interesting option in terms of computational cost and effective results in industries focused on real time interactions such as the gaming industry.

The final acoustic model of a room obtained using a ray based simulation will depend on how sound rays are shot from the sound source and followed through the enclosure. The definition of the final ray tracing algorithm will involve the definition of the way to model sound phenomena such as absorption, diffusion and interference.

The main scope of this thesis is:

To analyse the generation of acoustic impulse responses of rooms using ray based simulations, and to contribute to sound reflection modelling by presenting a method developed to apply the analytical solution for the Acoustic Bidirectional Reflectance Distribution Function [He et al. (1991); Siltanen et al. (2007)] in the Vector Based Scattering Model [Christensen and Rindel (2005)] (VBS), which is one of the existing ways to efficiently include diffusion in a ray tracing algorithm by means of random vectors. These investigations will be implemented in a ray tracing engine and results will be validated through comparison to their analytical expressions.

Outline

This Thesis is organized as follows:

- Chapter 1: This chapter presents a brief summary of the basic mathematical and physical fundamentals involved in sound propagation in enclosures, focusing on the problems that appear when it is approximated to rays. These notions are necessary for proper comprehension of this dissertation; however, a reader with some experience in this field, could just overview this chapter.

- Chapter 2: Once defined the mathematical and physical foundations of sound propagation in a room, we present and study various approaches used in room acoustics. We review the state of the art in geometrical techniques and mention the most remarkable references about wave-based methods to offer an ide of the evolution of both approaches during the last decades.

- Chapter 3: This chapter reviews the Room Acoustic Rendering Equation and the application of BRDFs to acoustics. It also introduces the general methodology to compute analytically the A-BRDF and presents its computation for the Vector Based Scattering model. The chapter also shows the comparison with the corresponding Monte Carlo results and the process to include this contribution to the ray tracing engine.

- Chapter 4: In this chapter we describe in detail the way ray tracing works for audio rendering. We also present the ray tracing engine that we have designed and implemented to compute the impulse responses for the pressure and the velocity in any given 3D environment, and discuss some of our results.

- Chapter 5: Finally, conclusions obtained in this dissertation are presented, including some guidelines for future research.

Chapter 1

Background

1.1 Basic Concepts in Sound Propagation

Waves are a physic phenomenon that is defined as the propagation of energy of a periodic perturbation without any transport of matter [Serway and Jewett (2002)]. It is easy to observe this kind of behaviour daily: waves on the sea, vibrations on a string, sound waves, TV and radio signals, X-rays, light, etc. There exist two different types of waves depending on their nature: electromagnetic waves and mechanical waves.

- Electromagnetic waves exist as a consequence of two effects: a variable magnetic field that yields an electric field or a variable electric field that yields a magnetic field. Electromagnetic waves do not need a medium to propagate and can propagate in vacuum. These waves propagate in a perpendicular direction of both fields, electric and magnetic and travel through vacuum with a velocity equal to the velocity of light.

- Mechanical waves need a medium for their propagation (they can not propagate in vacuum). In this case, the propagation of a perturbation is driven by the particles of the medium (solid, liquid or gas). There are two types of mechanical waves: transverse waves and longitudinal waves. Transverse waves are produced when the particles of the medium move in a perpendicular direction of the traveller waves. In the right hand of fig. 1.1 we can observe a spring with a periodic perturbation, and the movement of the particles of the spring is perpendicular to the energy transport.

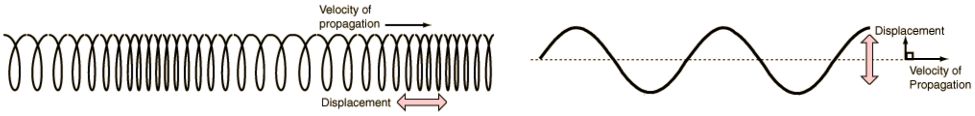


Figure 1.1: A longitudinal wave in a slinky on the left side and a transverse wave in a string on the right.

On the other hand, longitudinal waves are those for which the movement of the particles in the medium is produced in the same direction of the traveller waves. In left side of fig. 1.1 we can observe a slinky with a periodic perturbation moving in the direction of wave propagation. Note that the movement of particles generate compressions and rarefactions in the same direction of the energy transport.

Sound waves travelling through air are the paradigmatic example of longitudinal waves. Air particles vibrate producing changes in pressure and density alternating compressions and rarefactions all over the direction of the movement of the wave. Sound waves can propagate in any fluid with a velocity that depends on its mechanical properties. In fact, the velocity of all mechanical waves, c, depends, generally, on the elastic and inertial properties of the medium:

$$c = \sqrt{\frac{B}{\rho}} \quad , \tag{1.1}$$

where B is the volume elastic property and ρ is the density of the medium. The velocity of propagation also depends on the temperature of the medium. For sound waves moving through the air, the relation between velocity and temperature is given by the simple expression:

$$c = 331\sqrt{1+\frac{T}{273}} \quad , \tag{1.2}$$

where 331 m/s is the velocity of sound through air at $0\,^{\circ}\mathrm{C}$, and T is the temperature of the fluid in Celsius. Sound waves are divided into three categories that cover a wide range of frequencies. Audible waves (with frequency, f, ranging from 20 Hz to 20000 Hz), infrasonic waves (f $<$ 20 Hz) and ultrasonic waves (f $>$ 20000 Hz). Note that the audible range includes three orders of magnitude of frequencies.

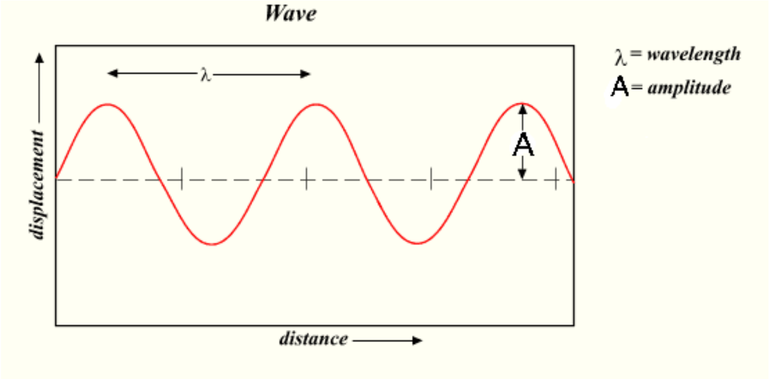


Figure 1.2: A sinusoidal wave.

From the mathematical point of view the most fundamental wave is the so called sinusoidal wave which is described by the equation $x(t) = A \cdot sin(2\pi f t + \theta))$, where A is the amplitude of the wave (i.e., a measure of the maximum disturbance in the medium during one wave cycle that determines the loudness of the sound), f is the frequency of the wave, which determines the pitch, t is the elapsed time and θ is the phase of the oscillation[1]. In fig. 1.2, the amplitude A is the maximum vertical distance between the baseline and the wave. The units depend on the type of wave: waves on a string have an amplitude expressed as a distance (meters), sound waves as pressure (pascals) and electromagnetic waves as the amplitude of the electric field (volts/meter).

The wavelength (denoted as λ) is the distance between two sequential crests. This generally has the unit of meters.

[1] The phase of an oscillation is determined by the initial displacement at time t=0

1.2 Sound Propagation

In any sound wave, the particles of the medium undergo vibrations about their mean positions. Therefore, a wave can be described completely by indicating the instantaneous displacements of these particles [2].

The vibrations in a sound wave do not take place at all points with the same phase. We can, in fact, find points in a sound field where the particles vibrate in opposite phase. Therefore, under the influence of a sound wave, variations of gas density and pressure occur, both of which are functions of time and position. The difference between the instantaneous pressure and the equilibrium pressure is called the *s*ound pressure.

The various acoustical quantities are connected by a number of basic laws which enable us to set up a general differential equation governing sound propagation:

- *C*onservation of momentum: the conservation of momentum is expressed by the relation:

  $$\vec{\nabla} p = -\rho_0 \frac{\partial \vec{v}}{\partial t} \quad , \tag{1.3}$$

  where p denotes the sound pressure, $\vec{v}$ the vector particle velocity, t the time and ρ_0 the static value of the gas density [Morse and Ingard (1986)].

- *C*onservation of mass: the conservation of mass leads to:

  $$\vec{\nabla} \cdot \vec{v} = -\frac{\partial \rho}{\partial t} \quad , \tag{1.4}$$

  being ρ the total density.

Under the further assumption that we are dealing with an ideal gas, the following relations hold between the sound pressure, the density variations and the temperature changes $\delta\Theta$:

[2] Usually the velocity of particle displacement is considered as a basic acoustical quantity rather than the displacement itself

$$\frac{p}{p_0} = \frac{k}{k-1}\frac{\delta\Theta}{\Theta+273} \quad . \tag{1.5}$$

Here k is the adiabatic exponent (for air $k = 1.4$). From the equations (1.3) and (1.4) we can derive the so called *w*ave equation:

$$c^2\Delta p = \frac{\partial^2 p}{\partial t^2} \quad , \tag{1.6}$$

where

$$c^2 = k\frac{p_0}{\rho_0} \quad , \tag{1.7}$$

being c the velocity of propagation.

This differential equation governs the propagation of sound waves in any lossless fluid. If one was able to find the exact solution of the wave equation subject to arbitrary boundary conditions, i.e. to solve it in an arbitrary scene, the problems of sound rendering would be immediately solved.

Moreover, it is worth mentioning here that equations 1.3 and 1.4 (which are called the linear Euler equations) are very similar to Maxwell's equations for the electromagnetic field. Therefore, all the methodologies applied to electromagnetism are also applicable to solve Euler equations of acoustics.

1.3 Numerical Methods

Exact solutions for the Euler equations presented in the previous section are only known for extremely simple geometries. Thus in virtually all applications, one needs to solve the equations numerically, and resort to some sort of approximations. It is precisely in these approximations that the art of rendering resides, and where the differences between light and sound begin to arise.

The sound propagation through an environment can be determined using two major approaches:

- the computation of the exact solution to the *w*ave equation, which is the equation that describes the propagation of sound through an environment

- the approximation of the exact solution based on geometric propagation paths.

There exist several techniques, usually known as *w*ave-based methods, to numerically solve the wave equation such as the Finite Element Method (FEM), Boundary Element Method (BEM) and Finite-Difference Time Domain (FDTD). Although all of them are computationally more demanding, they have gained interest during the last years because of the growing capabilities of standard computers.

In the second approach, *r*ay-based methods such as ray tracing and image source methods are the most often used modeling techniques. They are based on the fact that sound and light propagation share a number of similarities, thus motivating the application of the advances already achieved in graphics rendering during the last decades to render audio. Chapter 2 offers an overview of the state of the art in those two approaches.

It is worth reminding here that the *r*ay approximation is only valid when the wavelength of the field λ is much smaller than the typical size of the objects present in the scene L_{obj},

$$\text{ray approximation:} \qquad \lambda \ll L_{obj}\,. \tag{1.8}$$

Given that the visible light has a wavelength ranging from 390 *nm* (violet) to about 780 *nm* (red), it is clear that the ray tracing requirement (1.8) is satisfied in almost all imaginable scenes.

The situation for sound is quite different. The wavelength of air vibrations that can be heard by humans oscillates between 17 *mm* and 17 *m*, corresponding to frequencies of 20 KHz and 20 Hz, respectively. On the one hand, this range is much wider than that of light, extending over three orders of magnitude; on the other hand, sound wavelengths are up to eight orders of magnitude larger, arriving at wavelengths comparable to the objects present in most relevant scenes. Therefore, the requirement (1.8) is only satisfied for high frequency sounds. At low frequencies, the sound field reveals its wave behavior by diffracting off objects (fig. 1.3), and interfering with itself: these phenomena are missed in the ray approximation.

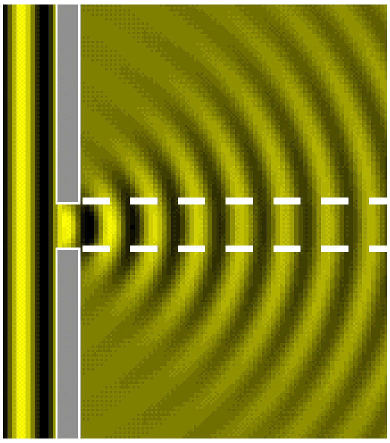


Figure 1.3: Wave diffraction. The small opening acts as a new sound source.

Let us give an example to illustrate the problem. Imagine a speaker trying to communicate with a listener in a room where both are separated by a wall with an opening (an open window or door) as in figure 1.4. The direct sound perceived by the listener is the part of the acoustic field diffracted in the opening, instead of the complicated path full of reflections that the ray-tracing approximation would provide. Besides *c*oloring the speech in a non-trivial way, this diffraction allows the listener to perceive the sound as coming from the right direction. If, instead of speech, only high frequency sounds were emitted, diffraction would be minimized, and the ray tracing computation would get closer to the true result.

The ray tracing algorithms are being extended in an attempt to overcome this problem [Funkhouser et al. (2002)], typically by incorporating diffraction filters for single propagation paths based on the geometric theory of diffraction. Though the results are acceptable in a wider range of situations, there is still a lot of research to be done in this topic.

Finally, it is worth mentioning one advantage of the large wavelengths of sound as compared to those of light: sound propagation is not affected by fine geometric details of the scene, thus giving good results in simple approximate 3D models.

All the research presented in the document is oriented towards

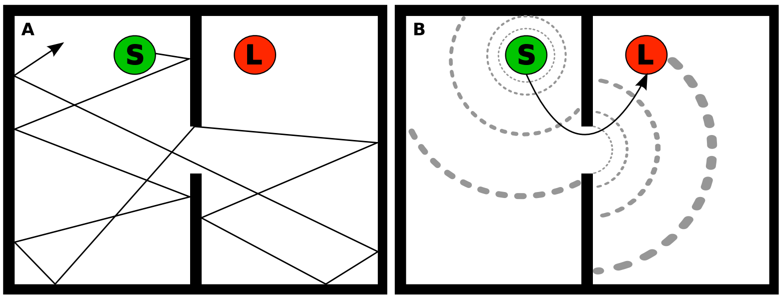


Figure 1.4: In A it is shown that within the ray approximation hardly any ray arrives to the listener. However, in the more realistic situation shown in B, diffracted sound is the first audio signal getting to the listener.

the use of geometric methods such as ray tracing to compute the acoustic field of any given environment.

1.4 Room Acoustics

*R*oom acoustics is concerned with sound propagation in enclosures where the sound conduction medium is bounded on all sides by walls, ceiling and floor. These room boundaries usually reflect a certain fraction of the sound energy inciding on them. Another fraction of the energy is absorbed, either by conversion into heat or by being transmitted to the outside by the walls. It is just this combination of the numerous reflected components which is responsible for what is known as *t*he acoustics of a room and also for the complexity of the sound field in a room [Kuttruff (1999)].

We talk about *G*eometrical Room Acoustics when we use the ray approximation to obtain the description of the sound field. As far as we have seen, it is self-evident that geometrical room acoustics can reflect only a partial aspect of the acoustical phenomena occurring in a room, neglecting phenomena such as diffraction. This aspect is, however, of practical importance and therefore we must deal with it in some detail.

1.4.1 Main features

If a sound ray strikes a solid surface it is usually reflected from it. This process takes place according to the reflection law well known in optics. It states that the ray during a perfectly specular reflection remains in the plane defined by the incident ray and the normal to the surface, and that the angle between the incident ray and the reflected ray is halved by the normal to the wall, as shown in figure 1.5.

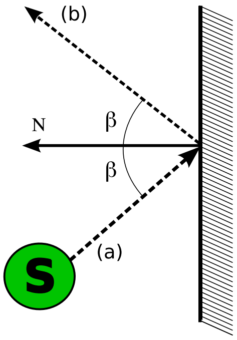


Figure 1.5: Specular reflection of a sound ray, showing the incident ray (a) and the reflected ray (b) forming the same angle β with the normal of the plane N.

There exist two important phenomena that determine the acoustic of the environment: the absorption and the diffusion of the walls:

- *A*bsorption: Each wall in the environment can be characterized by its *s*pecific acoustic impedance Z, which is the ratio of sound pressure p (in N/m^2 or Pa) to particle velocity v (in m/s) at a single frequency:

  $$Z = \frac{p}{v}. \tag{1.9}$$

  This parameter is defined taking into account the specific features of each material of the environment, and from this parameter it is possible to derive the absorption and the change of phase of the wave.

Usually not all the energy striking a wall is reflected from it; part of the energy is absorbed by the wall (or it is transmitted to the other side, which amounts to the same thing as far as the reflected fraction is concerned). The fraction of sound intensity which is not reflected is characterised by the *a*bsorption coefficient α. This coefficient can be defined as the ratio of the non-reflected to the incident intensity. It depends generally on the angle of incidence and on the frequencies which are contained in the incident sound. These information can be taken into account by reducing the intensity of the reflected ray by a fraction $1-\alpha$ of the primary intensity.

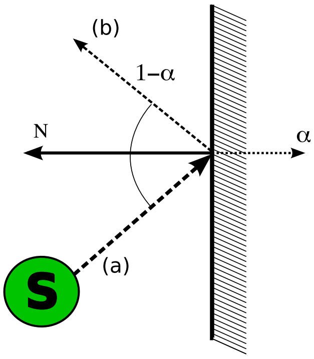


Figure 1.6: Situation in which part of the intensity of the incident ray (a) is absorbed by the wall affecting the reflected ray (b).

- *D*iffusion: We shall now take into consideration a more realistic approach in which the reflected sound ray usually has a direction that differs from the specular one. Actually, there is another factor that is necessary to be taken into account to completely describe the acoustics of an enclosed environment, namely the *d*iffusion of a surface. A surface is considered diffusive if it uniformly distributes sound pressure throughout the space, as shown in figure 1.7. Therefore the diffusiveness of a surface decreases the directionality and the coherence of the sound wave[3]. In analogy with this, a room

[3] In physics, two wave sources are perfectly coherent if they have a constant phase difference and the same frequency.

is said to be diffusive at some point if the reflected sound waves arrive uniformly from all directions, and non-diffusive if they all come from one single direction.

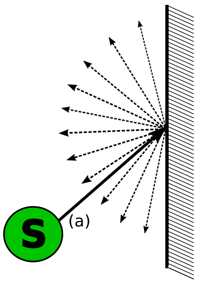


Figure 1.7: A diffusive surface that distributes uniformly the incident ray (a) in the environment.

Once we have already revised the major properties present in any computation of an acoustic field, then we have to define in detail the final objective we are interested in.

1.4.2 Impulse Response

The final objective is, given a sound source, to compute the sound perceived in any position of an enclosed environment, i.e. simulating what a microphone at that position would record.

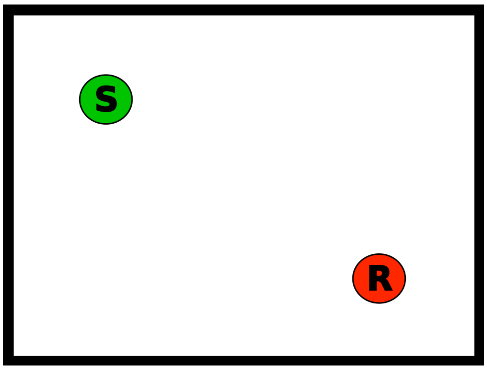


Figure 1.8: Schema of a simple rectangular room with a sound source (S) and a listener (L).

If we considerate the room configuration described in 1.8 with one single source of sound, then what we have to achieve is compute the way in which sound emitted by it arrives to the listener position, taking into consideration the absorption and the diffusion of the walls introduced before.

We will be also interested in acquiring directional information of the acoustic field at the listening point, in order to later be able to acoustically locate the sound sources during the exhibition stage.

As already explained, the application of geometrical methods to compute the sound field is supposed to give accurate results for the high frequency part of the audio stream.

If we create a pressure pulse at the source position by generating a number of rays with random direction and the same initial relative sound pressure, when all the ray reflections are followed through the environment, the sound signal received at a given point is then obtained as the superposition of the contributions of all the rays arriving from the source. Because of the different traveling distances, the waves (or rays) originating from the source arrive at the listener position with different delays and strengths. To obtain the signal at the receiving point one only has to add the sound pressures of all contributions. The strength of a particular contribution must also include the absorptivity and diffusiveness of the walls which have affected it.

If we mark the arrival times of the various rays by perpendicular dashes over a horizontal time axis and choose the heights of the dashes proportional to the relative sound pressure, we obtain what is frequently called the *I*mpulse Response of the scene. It contains all significant information on the temporal structure of the sound field at a certain point of the environment. Figure 1.9 shows a typical room impulse response.

To completely characterize the acoustic field at a certain point of the environment we will be interested in computing a set of four different impulse responses, one to acquire the information of the acoustic field at the listening point and the other three to acquire the directional information. This directional information is used to acoustically locate the sound sources at the exhibition stage.

It is important to understand a few qualitative properties com-

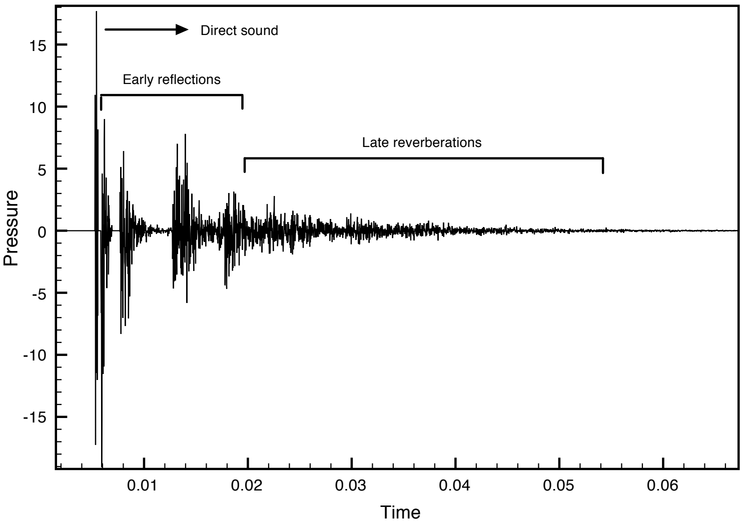


Figure 1.9: A typical impulse response.

mon to all impulse responses. During the first few milliseconds no sound is received at the listener's position, and the the first nonzero values show a loud peak corresponding to the *d*irect sound that has arrived to the listener position with no reflections (fig. 1.10a).

After the direct sound, some strong reflections occur then sporadically (fig. 1.10b), and the last part of the impulse response, or *l*ate reverberation tail, contains rays that have reflected off many surfaces in the environment (fig. 1.10c).

The role of the first isolated reflections with respect to our subjective hearing impression is quite different from that of the very numerous weak reflections arriving at later times. Psychoacoustic studies reveal that human hearing mostly uses the direct sound to capture the direction where the speaker is located [Blauert (1999)]; early reflections to capture the shape and size of the scene; and late reverberations to perceive acoustic features such as absorption and diffusion of the present materials.

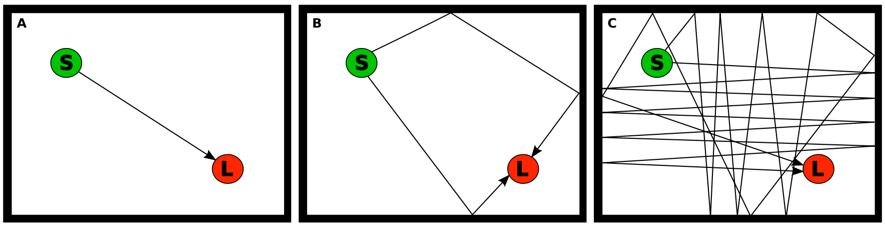


Figure 1.10: Direct sound from the sound source S to the listener L (A),early reflections (B) and late reverberations (C).

1.5 The final rendered audio

Once we have explained what is an impulse response the next step is to define:

- how do we obtain the set of 4 acoustic impulse responses that completely characterize the acoustic field given an environment, a sound source and a listener.

- how do we use it to confer all the features of the environment to any input audio

To obtain the 4 impulse responses we emit a gunshot from the sound source position (i.e. a Dirac delta function like the one shown in the left-bottom corner of fig. 1.11) and we measure the pressure g_p and the three components of the velocity $\vec{g}_v$ at the listener position.

All this procedure is for further using the well-known Green's Theorem, which states that the sound heard at a point L produced by *a*ny audio source at point S can be easily obtained if one knows what an acoustic pulse[4] produced at S would sound like at L.

The equation is simple: let $g_p(L,S)$ be the acoustic pressure at L produced by such a pulse, then any other audio $a(S)$ located at S would produce an acoustic pressure $p(L)$ at L given by

$$p(L) = g_p(L,S) * a(S)\,,$$

[4] The more detailed definition of acoustic pulse is a very loud and short intensity sound. More mathematically, a sound containing all frequencies with exactly equal intensity. Also known colloquially as *g*unshot signal.

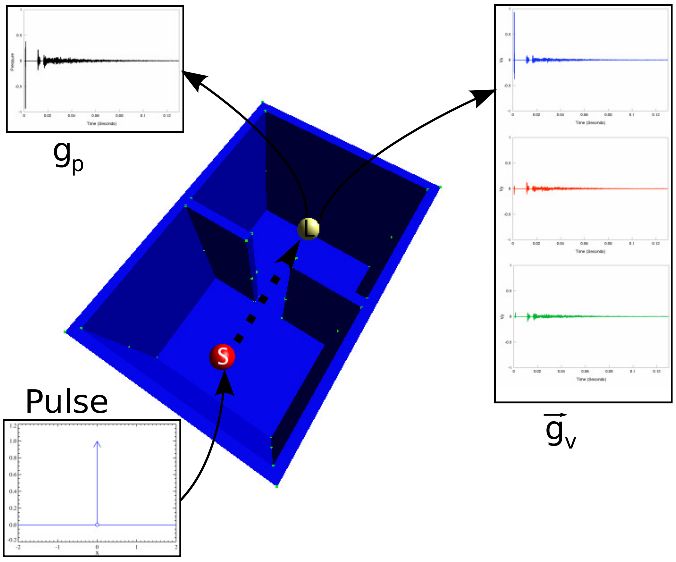


Figure 1.11: A dirac delta function is emitted from the source position (S) and the values for pressure (g_p) and velocity ($\vec{g}_v$) are measured in the listener position (L).

where $*$ denotes the convolution operation. In other words, if we want any input sound $a(S)$ to be rendered as it was heard in the position L of the environment we just need to convolve it with the acoustic pressure at the same position (see fig. 1.12).

As for the velocity vector mentioned above, all one needs to do is to compute the velocity of the acoustic field at the listening point. Green's Theorem applies here too: velocity $\vec{v}(L)$ at L is obtained via

$$\vec{v}(L) = \vec{g}_v(L,S) * a(S)\,,$$

where $\vec{g}_v(L,S)$ is the acoustic velocity vector at L produced by a pulse.

1.6 Audio Rendering

First of all, it is essential to understand what is audio rendering about. It can be easily deduced that the term *r*endering comes from the graphics literature, where it is commonly used to talk

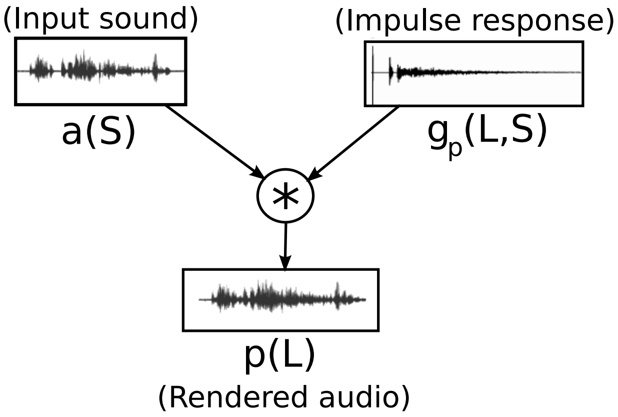


Figure 1.12: Schema of the process to obtain the final rendered audio.

about the process of obtaining a 2D image from a 3D world from an specific viewpoint with all the features that can be captured from that position (i.e. lighting, materials, geometry, etc.). In the case of audio, when we talk about *a*udio rendering we are actually referring to the analogous process of capturing an audio from a given virtual environment containing all its specific acoustic features.

Typically, when one wants to obtain a sound containing the features of an existing environment the standard solution is to play the sound in that environment using for instance a speaker (see figure 1.13), and then record what is heard in a specific position through a microphone. Obviously the problem appears when the environment does not exist (e.g., an imaginary futuristic star ship) and other methodologies have to be considered to obtain the same final audio.

Audio rendering focuses on that problem and aims to compute how a given audio played in a concrete position in a virtual environment is heard from another position in the same environment.

Moreover, it is not only a good solution to the previous problem but also has many applications in other areas, too. For example, It is very useful in architectural acoustics to pre-evaluate the quality of an opera house or a concert hall before building them, and it is also a good methodology to improve the noise suppression in the design of a new building.

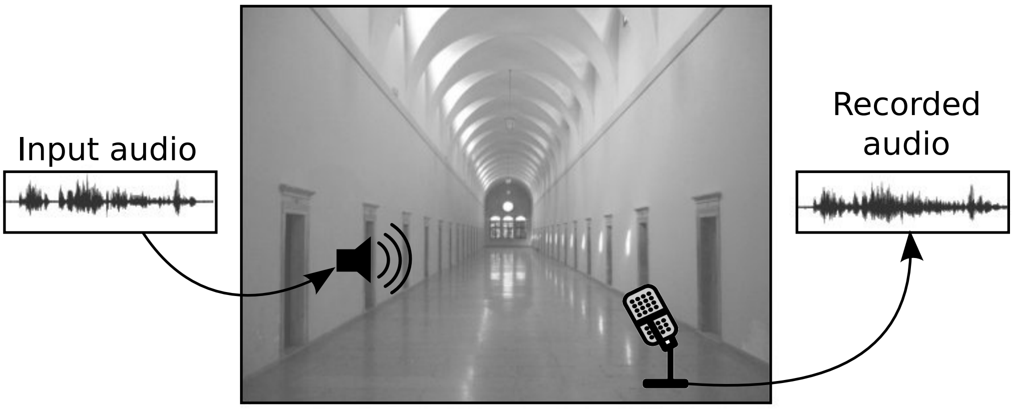


Figure 1.13: An schema of a typical audio recording process.

To render any audio we have to know first how can we compute all the features of the virtual environment in which is played to then apply them to the input audio. In fact, most of the research done nowadays in audio rendering is focused on computing the acoustic field of an environment to obtain all its particular features. As we will se later, once we acquire this information it is easy to use it as a filter to add all the features of the environment to the input audio in order to obtain the final rendered audio.

Now we are going to specifically concentrate on the needed procedure to obtain the features of an environment given a sound that is emitted from a sound source and received in a listener position. The main actors in this process are clear: one or more audio sources, a listener and an environment.

Just to simplify it, imagine that we put the audio source and the listener inside an infinitely large room, which is the same that considering a non-limited environment. A scheme of the example is shown in figure 1.14. In this case, if we want to know how the sound is arriving from the source to the listener, we only have to concentrate on the *d*irect sound going from one position to the other, and the rest is lost. Therefore the computation does not include any consideration about the environment and takes into account other involved parameters related to sound propagation.

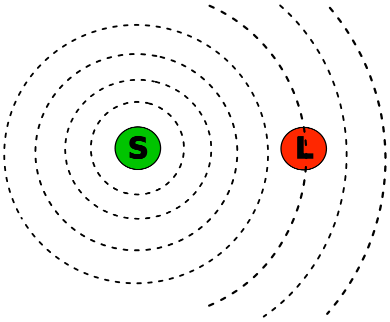


Figure 1.14: A sound source (S) and a listener (L) in a non-limited environment.

The way in which a sound propagates through air is a well defined topic in physics and it is possible to exactly compute the way in which the sound arrives from the source to the listener if parameters such as the temperature and the humidity of air, which affect to the velocity of propagation, and the distance traveled are known. Section 1.2 offers a short introduction to sound propagation and its main features.

Now let us observe the changes when we introduce a simple environment like the one illustrated in figure 1.15 into the previous case and we try to evaluate how it affects to the sound propagation.

In practice the difference when we add a bounded environment and play a sound from the source position is that the listener hears not only the same direct sound that received before ($t1$ in fig. 1.15) but also the same emitted sound coming from the reflections with the walls of the environment ($t2$ and $t3$ in fig. 1.15) arriving at different times. Those reflected sounds are affected by the specific features of the wall in which have collided and therefore introduce information about the environment.

Consequently, if one wants to evaluate how the sound is received in the listener position, two main aspects have to be taken

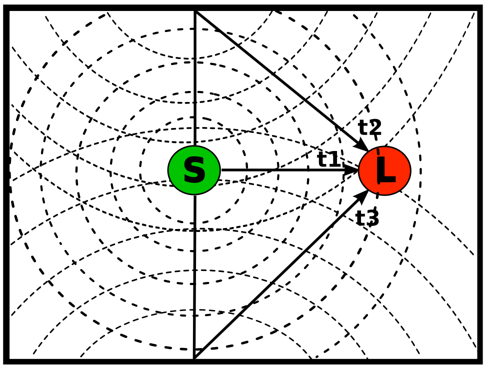


Figure 1.15: A sound source (S) and a listener (L) inside a simple rectangular environment. The listener receives the three first different sound waves containing the same emitted sound at three different times t1, t2 and t3. The wave collisions with left and right walls are omitted.

into account:

- The way in which sound propagates through air (figure 1.14)
- The way in which sound is affected when it collides with any wall or any object in the scene (figure 1.15).

As the reader will probably imagine, the second consideration implies the parametrization of any material of a wall, floor or ceiling present in the virtual environment.

Depending on how the environment is affecting the sound waves, the listener will receive all the arriving sound waves at different moments and with different pressures. A short introduction to the most important aspects of *r*oom acoustics is presented in Chapter 2.

1.7 The role of audio rendering in multimedia productions

Producing the audio for a movie, for an audiovisual or for a TV show is usually a well defined process with a set of differentiated steps. The same happens with the production of audio effects for an interactive experience or for a video game. Although there are many differences between the production of a soundtrack for a super-production and the compilation of a music CD, there are some generalized invariant stages that have to be covered if one wants to obtain the final desired audio, as shown in figure 1.16.

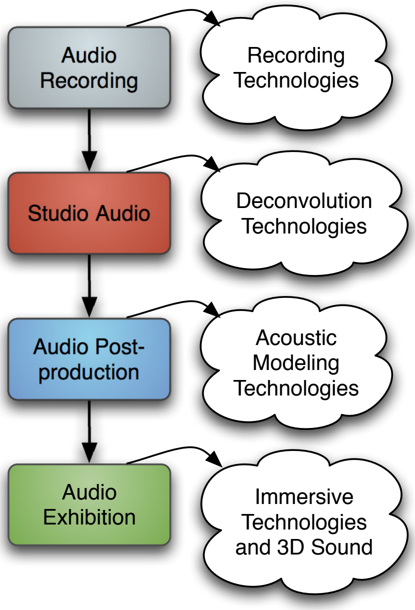

Audio Recording
Recording Technologies
Studio Audio
Deconvolution Technologies
Audio Post-production
Acoustic Modeling Technologies
Audio Exhibition
Immersive Technologies and 3D Sound

Figure 1.16: Standard audio chain present in most audiovisual or standalone audio productions.

- *A*udio Recording: In most cases the first step would be the recording of all the necessary audio. In the case of a movie, one would record all the dialogs of the scenes, and part of the soundtrack played by an orchestra or by a music group in a Studio. The existing alternative to recording the audio is synthesizing it, but in practice the result is the same: going from nothing to the first raw version of our audio compilation.

  Imagine for instance the production of an action movie full of special effects, in which a big number of scenes are recorded in the studio with a chroma background, and the rest are recorded in real controlled environments. Suppose that in the climax scene the main actress is entering to the *S*agrada Familia[5] from the top, descending silently by a rope, and the director wants a subjective camera in the position of the actress in such a way that the audience see exactly what she sees. The first objective in terms of audio would be to obtain all the data implied: in this case suppose that they want to increase the tension of the scene with the noises of her breathing and the friction of her hands and feet with the rope as she descents, and complete it with the voices of two guards walking far from the focus, in the floor of the monument.

- *S*tudio Audio: A second standard process is to *c*lean the acquired sound in order to have an audio ready to be processed. In other words, if we obtain a noisy recording of a dialog, it is not useful for our objective, and thus we have to look for other ways to acquire a better option. In this case one choice can be to re-record the audio in a controlled environment i.e. a Studio, but there is also some research oriented to the implementation of a suitable deconvolutioning engine that allows the inverse process of acquiring a clean sound from a noisy one. This is a very hard problem and nowadays any of the attempts have lead to acceptable results.

  In the previous proposed example, if the recording of the scene takes place in the real environment, the results will usually contain noises and other artifacts, and the audience will not be satisfied by an spectacular scene in which the main actress is entering to the Sagrada Familia if the corresponding audio is not in harmony with it. Moreover, it is clear here that the record of the breathing and the friction of the hands and feet with the rope is almost an impossible task to accomplish. Typically, these sounds would be previously recorded in the studio. Otherwise, the only existing option as

[5] The *Sagrada Família* is a Roman Catholic church in Barcelona (Catalonia), designed by Catalan architect Antoni Gaudí.

we previously observed is to try to *d*elete the imperfections by some hard handmade audio filtering.

- *A*udio Post-production: Subsequently, once all the audio is ready-to-use the next stage is to post-produce it, which means processing it by adding some special effects: enlarging the reverberation time, changing the pitch, etc. with the objective of reaching the final desired audio product. As one can deduce, most of the changes done here involve a huge percent of handmade work, and there are many different ways to change an audio signal thus becoming a tedious step.

  Returning to the heroine entering to the *S*agrada Familia, the producers want to immerse the audience as much as possible in the movie, and thus it is necessary that the audio sounds like it was heard inside the building. As we have just mentioned before, this is a hard and very subjective process, and in many situations the final result is quite convincing but very different from the real one. Obviously, one can use the reverb of a standard cathedral to integrate the audio with the scene, and probably the result will be attractive, but very different from the original one, missing all the particularities and the differentiating features of an unique monument.

- *A*udio exhibition: Finally the last stage in an audio production is the exhibition, usually coming with the visual part that has also been obtained by a standard process. Depending of the final audience, the audio has to be available for being converted to a set of many different exhibition systems. In the case of a movie like the one described, sound is usually subdivided into six different tracks, while if the final experience is going to be exhibited in a stereo system, then only two tracks are needed.

As far as we have seen, there is a lot of research still to be done in most of the steps previously commented, but from now on our focus would be set on the audio post-production stage, which is the one that can take advantage of audio rendering. As we previously said, there exist a large amount of handmade work in post-production, and one of our main objectives would be to look for the automatization of the processes involved.

Although the focus is set on the post-production stage, we will be also interested in improving the immersion of the exhibition systems.

All the following sections would turn around the same question: *h*ow can we accurately apply changes to the original audio in such a way that it sounds as it was recorded in a specific environment?

Chapter 2

Room Acoustics Simulation Techniques

2.1 Introduction

It is clear from the the previous section that whereas high frequency sounds can be modeled within the ray approximation, other approaches must be more desirable options for low frequencies. Most of the existing techniques for sound rendering can be classified into two main categories: *r*ay-based methods and *w*ave-based methods, as illustrated in figure 2.1.

Computers have been used for over thirty years to model room acoustics, and nowadays computational modeling has turned to be relatively common practice in the design of buildings that share specific auditive necessities such as concert halls. In the following revision of the two major approaches we will also try to summarize their evolution during the last decades, remarking most important influences. We end up the chapter with a revision of the four most important Audio modeling softwares.

2.2 Wave-based methods: low frequencies

At low frequencies one is forced to numerically solve the wave equation in order to obtain accurate results for sound rendering. All of the existing methods are based on the discretization of time

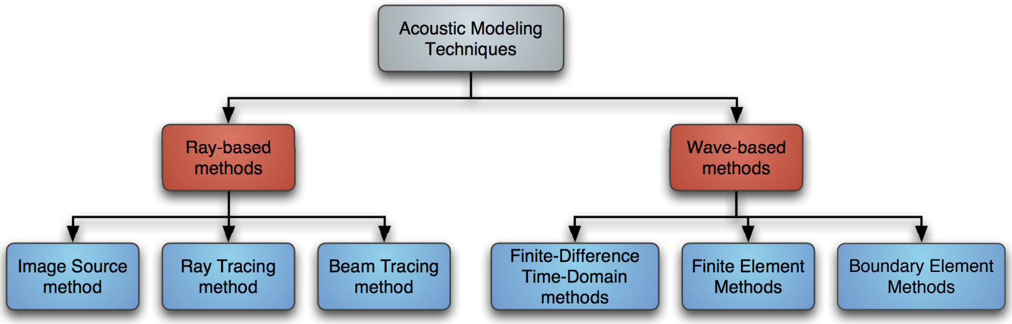


Figure 2.1: A schema of the major acoustic modeling techniques.

and/or space (by considering small intervals Δx, Δt) in order to turn the differential equation into an algebraic system. The main advantage of discretization is the fact that the numerical solution converges to the exact one as $\Delta x,\ \Delta t \to 0$, turning accuracy into a computational cost issue.

The most popular wave-based algorithms are: finite-differences time-domain methods (FDTD), finite element methods (FEM) and boundary element methods (BEM). These methods have been mostly applied to computational electromagnetics and computational fluid dynamics [Chung (2002)], including sound rendering simulations.

2.2.1 Finite-Difference Time-Domain Methods

The main principle in Finite-difference time-domain (FDTD) techniques is that derivatives in the wave equation are replaced by corresponding finite differences. It is the most common numerical method to solve Euler equations. There are many fields in which FDTD have been successfully applied: Computer Graphics, Electromagnetism, Seismology, Solid State Physics (Elasticity Theory) and also in Meteorologic predictions. In room acoustics, there exist some papers about FDTD [Botteldoren (1994)], [Botteldoren (1995)], [Garriga et al. (2005)], [López et al. (2007)]. Our belief is that for new multimedia applications, FDTD will play an important role, specially in the low frequency domain.

The enclosure is subdivided in a mesh of points, and the differential operators are approached using Taylor expansions [Chung (2002)]. This method is always used for homogeneous meshes[1], but more recently they have been also applied to multi-grid meshes.

Using Finite Differences Methods we obtain a discrete solution for the acoustic field: for every grid point we obtain the values for the acoustic pressure and the three components of the acoustic velocity.

The FDTD methods produce impulse responses better suited to auralization than BEM and FEM, which typically calculate frequency domain responses.

2.2.2 Finite Element Methods

Finite Element Methods are suitable only for small enclosures and low frequencies due to heavy computational requirements. In Finite Element Methods, the complete space has to be discretized with elements. In practice this means that matrices used by a FEM solver are larger but sparsely filled.

Finite element methods solve the wave equation (and associated boundary conditions), subdividing space into *e*lements. The wave equation is then expressed as a discrete set of linear equations of these elements. The boundary integral form of the wave equation (i.e., Green's or Helmoltz-Kirchoff's equation) can be solved by subdividing only the boundaries of the environment and assuming the pressure (or particle velocity) is a linear combination of a finite number of bases functions on the elements. One can either impose that the wave equation is satisfied at a set of discrete points (collocation method) or ensure a global convergence criteria (Galerkin method). In the limit, finite element techniques provide an accurate solution to the wave equation. However, they are mainly used at low frequencies and for simple environments since the compute time and storage space increase dramatically with frequency. As we observed in the review of FDTD methods there is no much literature of FEM applied to interactive room acoustics [Babuska et al. (2004)][Ihlenburg (1998)].

[1] In an homogeneous mesh all the volume elements have the same shape

Finite element techniques are also used to model energy transfer between surfaces. Such techniques, which are referred to as *r*adiosity methods have already been applied in acoustics, as well as other fields and provide an efficient way of modeling diffuse global energy exchanges (i.e., where surfaces are lambertian reflectors) [Nosal et al. (2004a)]. While they are well suited for computing energy decay characteristics in a given environment, energy exchange techniques do not allow direct reconstruction of an impulse response. Instead, they require the use of an underlying stadistical model and a random phase assumption. Moreover, most surfaces act primarily as specular glossy reflectors of sound. Although extensions to non-diffuse environments have been proposed in computer graphics, they are often time and memory consuming and not well suited to interactive applications.

2.2.3 Boundary Element Methods

The boundary element method (BEM) is obtained by putting together two components: the formulation of the problem as a boundary integral equation and the finite element method (FEM) of numerical solution [Ciskowski and Brebbia (1991)] [Seybert et al. (1990)]. Put another way, the BEM consists of applying a finite element method discretisation to a boundary integral equation formulation of a problem.

The idea of the method is to approximate the solution of the integral equation, that is the pressure, p, on the boundary ∂D, by a simple function, usually a polynomial, at each element.

This method depends crucially on our ability first of all to obtain solutions, by analytical means, to certain simple specific acoustic problems. Most crucially we need to be able to write down explicitly the acoustic field due to a point source of sound in free space.

In principle (with powerful enough computers which have enough memory) we can solve large classes of complex acoustical problems with this method. Of course, in practice, computational limitations will become an issue and, therefore, more efficient algorithms become desirable.

The BEM has the attraction compared to the FEM that only the surface of the domain is discretised. This means that the number of degrees of freedom (the size of the linear system to be solved) is hugely more for the FEM compared with a BEM discretisation of the same problem. But, on the other hand, the matrix in the linear system for BEM is a full matrix, i.e. each element of the matrix is non-zero. By contrast the FEM matrix is very sparse (almost each element is non-zero) which means it can be stored and solved efficiently. It is impossible to give a general rule as to which of BEM and FEM are better. It all depends on the particular discretisation and solution scheme and their implementation, but in practice BEM has a tendency to be superior for exterior problems and for problems in which the boundary is complex but smooth.

2.3 Ray-based methods: high frequencies

Ray-based methods are derived from the geometrical theory of acoustics (see, e.g., [Kuttruff (1999)]) which is valid within the ray approximation regime (1.8). The general approach of all these methods is to find significant ray paths along which sound can travel from a source to a receiver through the scene. The most important ray-based methods that have been applied to acoustics from the most prior systems are image source method, ray tracing and beam tracing. The basic distinction between these methods is the way the reflection paths are typically calculated.

To model an ideal impulse response all the possible sound reflection paths should be discovered. The image source method finds all the paths, but the computational requeriments are such that in practice only a set of early reflections is computed. Ray tracing applies the Monte Carlo simulation technique to sample these reflection paths and thus it gives a statistical result. By this technique higher order reflections can be searched for, though there are no guarantees that all the paths will be found.

The use of both image source methods and ray tracing methods to compute the sound field of an environment was first introduced around the 70's [Allen and Berkley (1979)] [Krokstad et al. (1968)]. At the same time, the use of ray tracing was growing in the image rendering literature, and in 1989 A. S. Glassner presented one of the most important books about ray tracing, containing not only all the necessary background to build a ray tracer but also some acceleration techniques to improve the computation time [Glassner (1991)]. The revolution that supposed the emerging use of this ray-based method generated an increasing interest in the acoustics research from the beginning of the 90's, generating new proposals such as pyramid tracing presented by A. Farina [Farina (1995)] or the subsequently beam tracing proposed by T. Funkhouser [Funkhouser et al. (1998)]. All these reseach originated the creation of many applications to compute the sound field of a room [Rindel (2000)], and there were also some emerging efforts oriented to the application of the ray tracing method to interactive applications [Savioja et al. (1999)].

2.3.1 Image Source

The image source method is one of the most common ray-based modeling methods. The concept of image sources has been applied to various field problems in electromagnetic and acoustic wave propagation. It has been extended to arbitrary geometries with plane walls [Nironen (2004)].

The concept is based on the principle that a specular reflection can be represented as a sound source (outside the physical boundary) that radiates in free space. It computes specular reflections paths by considering virtual sources generated by mirroring the location of the audio source, S, over each polygonal surface of the environment, as illustrated in figure 2.2. For each virtual source, S_i, a specular reflection path can be constructed by iterative intersection of a line segment from the source position to the receiver position, R, with the reflecting surface planes. Specular reflection paths are computed up to any order by recursive generation of virtual sources.

The primary advantage of image source method is their robustness. They guarantee that all specular paths up to a given order of reverberation time are found. However, image source methods model only specular reflection, and their expected computational complexity grows exponentially. In general, $O(n^r)$ virtual

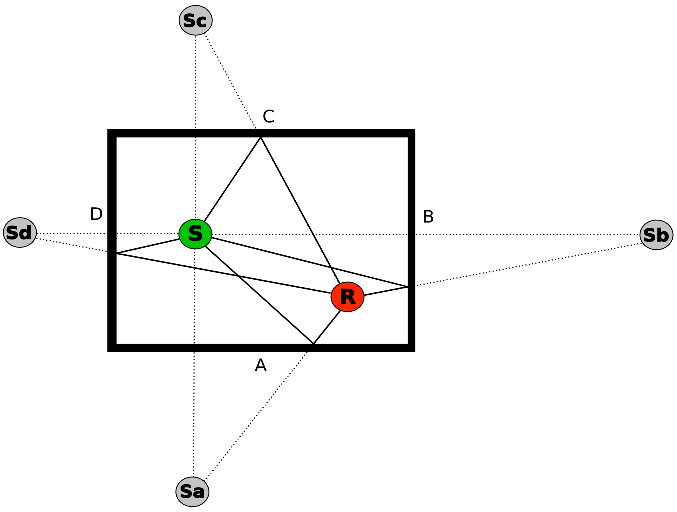


Figure 2.2: Image source method.

sources must be generated for r reflections in environments with n surface planes. Moreover, in all but the simples environments (e.g., a box), complex validity/visibility checks must be performed for each of the $O(n^r)$ virtual sources since not all of the virtual sources represent physically realizable specular reflection paths. For instance, a virtual source generated by reflection is blocked by another surface in the environment or intersects a point on a surface's plane which is outside the surface's boundary is "invisible". During recursive generation of virtual sources, descendents of invalid virtual sources can be ignored. However, descendents of invisible virtual sources must still be considered, as higher-order reflections may generate visible virtual sources. Due to the computational demands of $O(n^r)$ visibility checks, image source methods are practical for modeling only a few specular reflections in simple environments.

From a computational point of view the image source method is also a ray-based method. There are also hybrid models, in which ray tracing and image source method are applied together. Typically early reflections are calculated with image sources due to its accuracy in finding reflections paths, and later reflections are handled with ray tracing.

2.3.2 Ray Tracing

Ray tracing methods find propagation paths between a source and receiver by generating rays emanating from the source position and following them through the environment until a set of rays has been found that reach the receiver, as illustrated in figure 2.3 [Krokstad et al. (1968)].

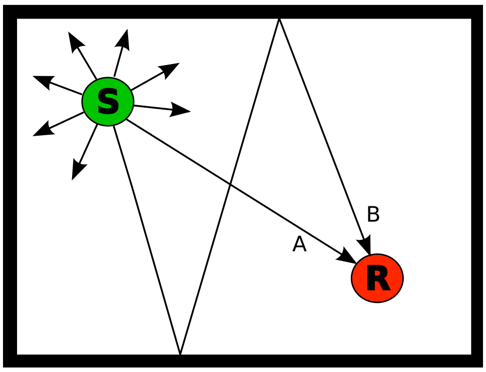


Figure 2.3: Ray tracing method, in which a number of rays are traced and followed until they reach the listener position.

The specular reflection rule is most common, in which the incident angle of an incoming ray is the same as the incident angle of the outgoing ray. More advanced rules which include for example some diffusion algorithm have also been applied.

The listeners are typically modeled as volumetric objects, like spheres or cubes, but the listeners may also be planar.

The primary advantage of these methods is their simplicity. They depend only on ray-surface intersection calculations, which are relatively easy to implement and have computational complexity that grows sublinearly with the number of surfaces in the model. Another advantage is generality. As each ray surface intersection is found, paths of specular reflection, diffuse reflection and refraction can be sampled, thereby modeling arbitrary types of propagation, even for models with curved surfaces.

The primary disadvantages of ray tracing method stem for their discrete sampling of rays, which may lead to undersampling errors in predicted room responses. For instance, the receiver position and diffraction edges are often approximated by volumes of space (in order to enable intersections with infinitely thin rays), which can lead to false hits and paths counted multiple times. Moreover, important propagation paths may be missed by all samples. In order to minimize the likelihood of large errors, ray tracing systems often generate a large number of samples, which requires a large amount of computation.

2.3.3 Beam Tracing

Beam tracing methods classify propagation paths from a source by recursively tracing pyramidal beams (i.e., sets of rays) through the environment [Funkhouser (2002)]. Briefly, for each beam, polygons in the environment are considered for intersection with the beam in front-to-back visibility order (i.e., such that no polygon is considered until all others that at least partially occlude it have already been considered). As intersecting polygons are detected, the original beam is clipped to remove the shadow region, a transmission beam is constructed matching the shadow region, and a reflection beam is constructed by mirroring the transmission beam over the polygon's plane, as shown in figure 2.4. This method has been used in a variety of applications, including acoustic modeling, illumination, visibility determination and radio propagation prediction.

The primary advantage of beam tracing is that it leverages geometric coherence, since each beam represents an infinite number of potential ray paths emanating from the source location. It does not suffer from the sampling artifacts of ray tracing, nor the overlap problems of cone tracing, since the entire space of directions leaving the source can be covered by beams exactly. The disadvantage is that the geometric operations required to trace beams through a 3D model (i.e., intersection and clipping) are relatively complex, as each beam may be reflected and/or obstructed by several surfaces.

Some systems avoid the geometric complexity of beam tracing by approximating each beam by its medial axis ray for intersection and mirror operations, possibly splitting rays as they diverge with distance. In this case, the beam representation is useful only for

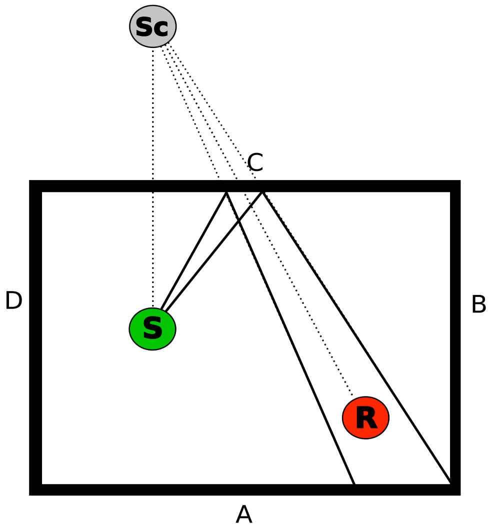


Figure 2.4: Beam tracing method.

modeling the distribution of rays/energy with distance and for avoiding large tolerances in ray-receiver intersection calculations. If beams are not clipped or split when they intersect more than one surface, significant propagation paths can be missed, and the computed acoustical field can be grossly approximated.

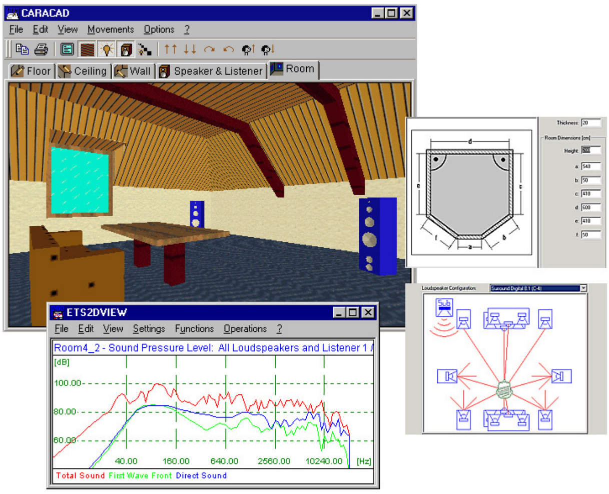
2.4 Existing Software Solutions

There exist many applications implementing all the methods presented above, but the most important are all focused on ray-based methods.

- **C**ARA [Cara (2015)]. CARA (Computer Aided Room Acoustics) is a very advanced computer program for computing and optimizing Room Acoustics of arbitrary rooms. CARA is based on the sound source imaging method in combination with a back tracing procedure.

  CARA analyzes and improves room acoustic influenced sound coloration in a two step procedure:

- determination of the basic acoustic properties of the room (Acoustic Ambience) in combination with suggestions regarding the furnishings (wall materials, furniture) in order to linearize the reverberation time spectrum,
- automatic Positional Optimization for the loudspeakers and the listener to minimize sound wave interferences (standing waves) in the listening area.


CARACAD
File Edit View Movements Options ?
Floor Ceiling Wall Speaker & Listener Room
ETS2DVIEW
File Edit View Settings Functions Operations ?
Room4_2 - Sound Pressure Level: All Loudspeakers and Listener 1 /
[dB]
100.00
80.00
60.00
40.00
160.00
640.00
2560.00
10240.00
[Hz]
Total Sound First Wave Front Direct Sound
d
e
c
f
a
b
Sub

Thickness: 20
Room Dimensions [cm]
Height: [illegible]
a: 540
b: 50
c: 410
d: 600
e: 410
f: 50

Loudspeaker Configuration: Surround Digital 0.1 (C-4)

Figure 2.5: Some CARA screenshots.

For a detailed analysis CARA calculates the total sound field data at 1,000-3,000 equally distributed grid points in the room. The evaluation of the sound field data concerns the modal structure (steady state sound pressure distribution), the room response of a Dirac-Pulse excitation in the time domain, the distribution of sound coloration, the sound imaging (stereophonic sound localization), and the speech intelligibility.

CARA provides the expert with a great number of room acoustic reference numbers (frequency dependent reverberation times, sound coloration, speech intelligibility, lateral sound level, or stereophonic sound localization) derived from sound pressure frequency responses or transient room responses/reverberation diagrams.

The special function Auralization renders a listening test in the (virtual) room to evaluate for example sound differences due to different loudspeaker positionings. Any piece of music (two channels Stereo) may be used for this. Some screenshots of this application are shown in figure 2.5.

- **C**ATT-Acoustic [Catt (2015)]. CATT-Acoustic is an application for windows OS based on several modules that make use of ray tracing methods to compute the sound field of a given environment.

  CATT is an acronym for Computer Aided Theater Technique since theater lighting and decor CAD programs were the first CATT products in 1986. Since 1988, however, CATT has concentrated on software for acoustics prediction/auralization (CATT-Acoustic) and more recently FIR reverberation tools (The FIReverb Suite). It also offers a 3D-viewer module based on OpenGL to present the environments.

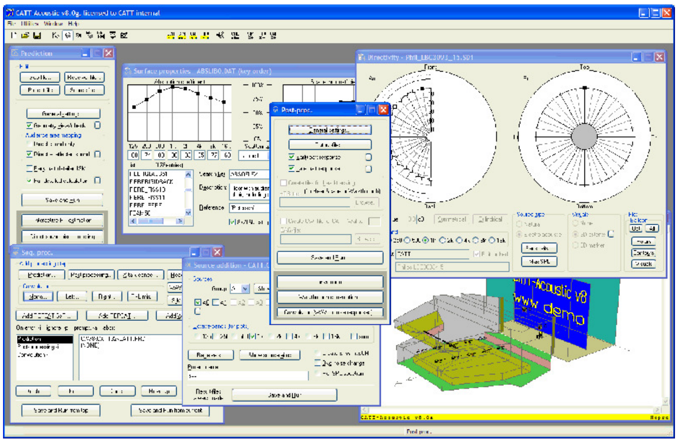

CATT Acoustic v8.0g, licensed to CATT internal
Prediction
Surface properties ABSLIB0.DAT (key order)
Directivity - Phil_LBC3093_15.SD1
Post-proc.
Seq. proc.
Source addition - CATT
CATT-Acoustic v8
www demo
Post-proc.

Figure 2.6: CATT-Acoustic screenshot.

The word "Catacoustics" was used in the 17th century to describe reflected sound (direct sound was called "Acoustics", and refracted sound was called "Diacoustics"). This makes the name CATT-Acoustic have some meaningful acoustical background since, in deed, reflected sound is exactly what it is about. This is of course pure coincidence since the software was named in 1988.

Version 9.0c Build 3 is the last significant version released for 32 Bits Windows on Feb 2015. A screenshot of this application is shown in figure 2.6

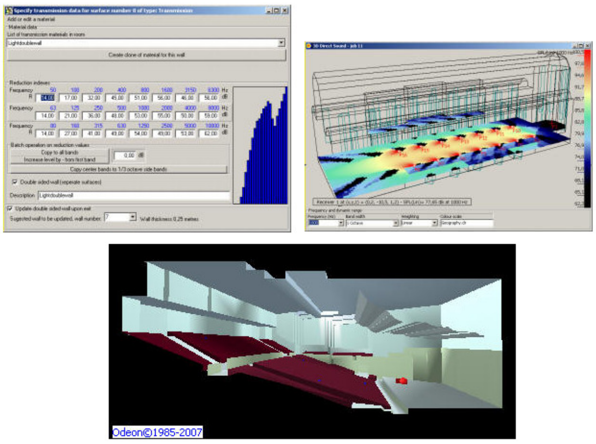
- **O**DEON [Odeon (2015)]. Brüel & Kjaer is the sole worldwide distributor of ODEON, a reliable, easy-to-use, modelling software tool for indoor acoustics, developed at the Technical University of Denmark.

  ODEON is PC software for simulating the interior acoustics of buildings where, from the geometry and properties of surfaces, acoustics can be calculated, illustrated and listened to. ODEON's prediction algorithms (image-source method combined with ray tracing) allow reliable predictions in modest calculation times. ODEON is ideal for the prediction of acoustics in large rooms such as concert halls, opera halls, auditoria, foyers, underground stations, airport terminals, and industrial workrooms. For noise prediction of large machinery in industrial environments, a special ray tracing algorithm has been developed allowing the modelling of surface and line sources.

  The last version released, 13, has been on 2015, including improved user interface, a material optimization tool, enhancements in the measuring system and some other features that can be found at their website. Some screenshots are shown in figure 2.7.

- **R**AMSETE [Ramsete (2015)]. Ramsete has been the first publicly available software for acoustical simulations based on the Pyramid Tracing computing scheme, which is quite different from traditional Ray Tracing or Cone Tracing schemes.

  Briefly, the Pyramid Tracing algorithm is capable to solve the sound propagation problems in large enclosures or outdoors, following the Geometrical Acoustics assumptions. It takes into account specular reflections over sound absorbing



Specify transmission data for surface number 8 of type: Transmission

Add or edit a material

Material data

List of transmission materials in room: Lightdoublewall

Create clone of material for this wall

Reduction indexes

Frequency	50	100	200	400	800	1600	3150	6300	Hz
R	14.00	17.00	32.00	45.00	51.00	56.00	46.00	56.00	dB
Frequency	63	125	250	500	1000	2000	4000	8000	Hz
	14.00	21.00	36.00	48.00	53.00	55.00	50.00	59.00	dB
Frequency	80	160	315	630	1250	2500	5000	10000	Hz
R	14.00	27.00	41.00	49.00	54.00	49.00	53.00	62.00	dB

Batch operation on reduction values

Copy to all bands

Increase level by - from first band

0.00 dB

Copy center bands to 1/3 octave side bands

☑ Double sided wall (seperate surfaces)

Description: Lightdoublewall

☑ Update double sided wall upon exit

Suggested wall to be updated, wall number: 7

Wall thickness 0.25 metres

3D Direct Sound - job 11

Receiver 1 at (x,y,z) = (0.2, -10.5, 1.2) - SPL(A)= 77.65 dB at 1000 Hz

Frequency and dynamic range

Frequency (Hz): 1000

Band width: 1 Octave

Weighting: Linear

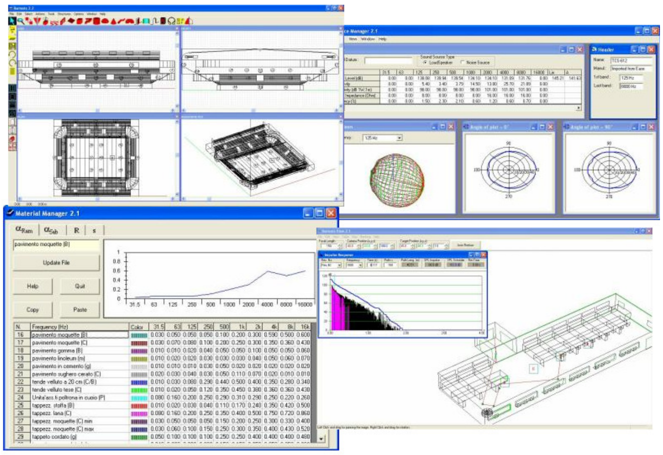
Colour scale: Geography.clr

Figure 2.7: ODEON screenshots.

surfaces, being the absorption coefficient defined over 10 octave bands. Furthermore, it takes into account diffraction over free edges of screens, energy passing through the panels (sound insulation), holes in large surfaces, windows or doors made of a material different from the surrounding surface.

The last version of this application, 2.7b, released on March 2012, is composed of several modules: Ramsete CAD (3D CAD system that can import and export AutoCAD files), Source Manager (manages all sound source data), Ramsete Tracer (computation engine of the application), Material Manager (manages acoustic properties of your materials), Ramsete view (allows to display the results of the simulation process) and some others. Some screenshots of these modules are shown in figure 2.8.

- **E**ASE [AFMG (2016)]. Ease is one of the current acoustical simulation software references that provides system designers and consultants with a set of tools for all aspects of professional practice, from detailed, realistic modeling and simulation of venue acoustics and sound system performance to informative and engaging client presentations, as well as data assessment and verification.



N.	Frequency (Hz)	Color	31.5	63	125	250	500	1k	2k	4k	8k	16k
16	pavimento moquette (B)	🟦	0.030	0.050	0.050	0.050	0.100	0.200	0.300	0.590	0.500	0.600
17	pavimento moquette (C)	🟥	0.030	0.070	0.080	0.100	0.200	0.250	0.300	0.350	0.360	0.430
18	pavimento gomma (B)	🟪	0.010	0.010	0.020	0.040	0.050	0.050	0.100	0.050	0.050	0.060
19	pavimento linoleum (m)	🟨	0.010	0.020	0.020	0.030	0.030	0.030	0.040	0.050	0.060	0.070
20	pavimento in cemento (g)	⬜	0.010	0.010	0.010	0.030	0.050	0.020	0.020	0.020	0.020	0.020
21	pavimento sughero cerato (C)	⬜	0.020	0.030	0.040	0.030	0.050	0.110	0.070	0.020	0.010	0.010
22	tende velluto a 20 cm (C/B)	🟦	0.010	0.030	0.080	0.290	0.440	0.500	0.400	0.350	0.280	0.340
23	tende velluto tese (C)	🟩	0.010	0.020	0.050	0.120	0.350	0.450	0.380	0.360	0.360	0.430
24	Unita'ass.ti poltrona in cuoio (P)	🟦	0.080	0.160	0.200	0.250	0.290	0.310	0.290	0.250	0.220	0.260
25	tappezz. stoffa (B)	🟥	0.010	0.020	0.030	0.040	0.110	0.170	0.240	0.350	0.420	0.500
26	tappezz. lana (C)	🟪	0.080	0.160	0.200	0.250	0.350	0.400	0.500	0.750	0.720	0.860
27	tappezz. moquette (C) min	🟪	0.030	0.050	0.050	0.050	0.150	0.200	0.250	0.300	0.330	0.400
28	tappezz. moquette (C) max	🟦	0.030	0.060	0.100	0.150	0.250	0.300	0.350	0.400	0.430	0.520
29	tappeto cordato (g)	🟩	0.050	0.100	0.100	0.100	0.250	0.250	0.400	0.400	0.400	0.480

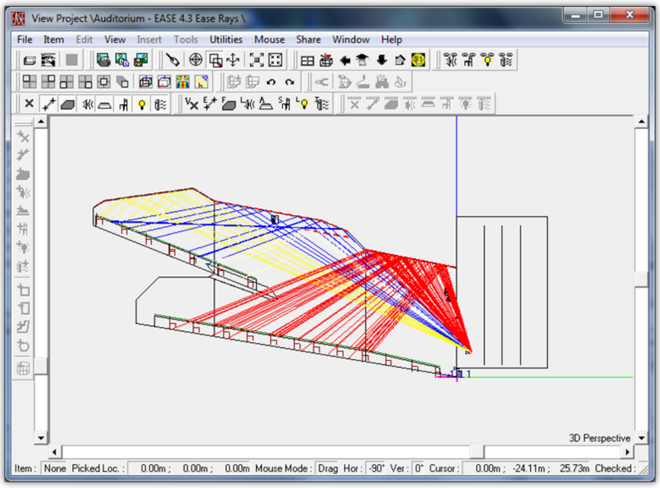
Figure 2.8: RAMSETE screenshots.

EASE, with more than 30 years of research and development, completes the main application with a row of additional modules that over the years have expanded the capabilities as well as the acoustical simulations as a whole. Among the extensions we find tools such a built-in drawing module for construction of a 3D acoustical model and a module for reflection pattern studies with ray tracing.

EASE offers different geometrical methods, ranging from image source method to simple ray tracing as well as a hybrid method comprising both. In all the options the focus is to visualise and study the propagation of sound throuhout the room taking into account the energy loss at each ray reflection. It uses a relatively small number of rays, e.g. 1000, to display them and analyse details such as bounce location, traveled path, distance, delay, etc.

In another module focused on calculations for in-depth reflection analysis, EASE offers a stochastic approach to model the propagation of sound in the room, giving the user the opportunity of performing a comprehensive analysis of reflections that are detected at specific spots – i.e. at Listener Seat locations. This approach uses a Monte Carlo simulation method, and in contrast with the simple Ray Tracing technique, here the number of particles (rays) is a fundamental factor, determining simulation accuracy and the repeatability of the results. In many cases 100,000 particles or more are used. This utility registers information for those rays that pass within a meter of a chosen point, e.g. a Listener Seat, and for each point individually.

AFMG offers two different versions of EASE and different modules for users to choose, EASE JR and EASE Standard, being the last one the full version that includes all ray tracing related features. In addition to this, AURA is the main extension that offers hybrid ray tracing routines and consideration of scattering coefficients. A screenshot of one of the modules is shown in figure 2.9.



Item : None Picked Loc. : 0.00m ; 0.00m ; 0.00m Mouse Mode : Drag Hor : -90° Ver : 0° Cursor : 0.00m ; -24.11m ; 25.73m Checked :

Figure 2.9: EASE screenshot: Built-in drawing module for construction of a 3D acoustical model.

2.5 Conclusions

In this chapter, we presented an overview of the most common numerical techniques applied to room acoustics. Basically, the numerical methods are divided in two big groups deppending on the considerations used to approximate the the sound phenomenon. On one hand, wave methods give accurate solutions in the steady-state of the acoustic pressure in the frequency space. The main disadvantage of this methods is that it is needed to carry out one different simulation by each frequency and this involves a high computational cost.

On the other hand, the second group of room acoustics simulations are the the geometrical methods that treat the sound propagation as rays or beams travelling through the space. This assumpition allows to build efficient algorithms, since the computational resources required for these algorithms make possible to create real time applications so important in computer graphics. Nevertheless, one important drawback is that phenomena such as diffraction, commonly observed in room acoustics, is not obtained, and this group of simulations are usually extended in an attempt to overcome this problem, typically by incorporating diffraction filters for single propagation paths based on the geometric theory of diffraction. Though the results are acceptable in a wider range of situations, there is still a lot of research to be done in this topic.

In Chapter 3, we introduce the *Room Acoustic Rendering Equation* which is a model for acoustic energy propagation and the *Acoustic Bidirectional Reflectance Distribution Function* (A-BRDF) which is the term of the equation that models sound reflections. Moreover, we present the method we have developed to compute the analytical solution for the A-BRDF in cases where sound reflections are diffuse, and diffusion is parametrized by one (or more) random variables. Secondly, we introduce in Chapter 4 the ray tracing engine that we have designed and implemented to compute impulse responses for the pressure and the velocity in any given 3D environment making use of the analytic solution of A-BRDFs to obtain all the needed outputs to create compatible sound for any of the most common 3D exhibition systems.

Chapter 3

Acoustic BRDF

3.1 Introduction

Computers have been used for over forty years to model room acoustics. Nowadays, computational acoustics modeling has become a common practice in many different disciplines: the acoustic design of buildings such as auditoria or concert halls [Kuttruff (1999); Rindel (2000)], outdoor acoustics [Salomons (2001)], audio post-production in Digital Cinema [IP-Racine (2006)] or audio for video-games and other interactive applications [Funkhouser (2002); Savioja et al. (1999); Savioja (1999)].

The most used techniques in computational room acoustics are the so-called geometrical methods which are based on the geometrical theory of acoustics [Morse and Ingard (1986)]. The general approach of the geometrical methods is to find significant paths along which sound can travel from a source to a receiver through the environment. The most important geometrical methods that have been applied to acoustics are: image source method [Allen and Berkley (1979); Borish (1984)], ray tracing [Krokstad et al. (1968)], beam tracing [Funkhouser et al. (2002, 1998); Laine et al. (2009)] and radiosity [Nosal et al. (2004b,a); Siltanen et al. (2009)]. Some of these (sometimes in combination) are the basis of many room acoustics commercial softwares such as *EASE* [AFMG (2016)], *Odeon* [Odeon (2015)], *Catt-Acoustics* [Catt (2015)] or *Ramsete* [Ramsete (2015)].

Recently, Siltanen et al. introduced the *Room Acoustic Rendering Equation* [Siltanen et al. (2007)] which is a model for acoustic energy propagation. This approach, borrowed from computer graphics, integrates several geometrical methods within the same theoretical framework. One key concept in the equation is the *Acoustic Bidirectional Reflectance Distribution Function* [He et al. (1991)] (A-BRDF) which is the term that models sound reflections.

In the present chapter we develop a method to compute the analytical solution for the A-BRDF in cases where sound reflections are diffuse, and diffusion is parametrized by one (or more) random variables. The method makes use of various properties of continuous probability functions, and exploits the relation between two and three dimensional probability densities.

The method is applied to the Vector Based Scattering Model [Christensen and Rindel (2005)] (VBS), which is one of the existing ways to efficiently include diffusion in a ray tracing algorithm by means of random vectors. Analytical results are obtained and shown to coincide with the large number of rays limit of the corresponding Monte Carlo simulation.

The chapter is organized as follows: section 3.2 briefly reviews the Room Acoustic Rendering Equation and the application of BRDFs to acoustics. Section 3.3 introduces the general methodology to compute analytically the A-BRDF. Section 3.4 presents the analytical computation of the A-BRDF for the Vector Based Scattering model, and the comparison with the corresponding Monte Carlo results. Finally, future work and conclusions are discussed in Section 3.5.

3.2 Room Acoustic Rendering Equation

Let us briefly review the Room Acoustic Rendering Equation (RARE) introduced in [Siltanen et al. (2007)], which models the propagation of acoustic energy through the environment both in diffusive and general non-diffusive conditions. Following the notation in [Siltanen et al. (2007)], the RARE reads:

$$l(x',\Omega) = l_0(x',\Omega) + \int_G R(x,x',\Omega) l\left(x, \frac{x'-x}{|x'-x|}\right) dx, \qquad (3.1)$$

where $G \subset \mathbb{R}^3$ is the set of all surface points in the enclosure, $l(x',\Omega)$ is the outgoing time-dependent radiance from point x' in the direction Ω, $l_0(x',\Omega)$ is the radiance emitted by the surface from x' in the direction Ω, and $R(x,x',\Omega)$ is the reflection kernel where x is the location of the previous reflection. The reflection kernel is a product of three terms:

$$R(x,x',\Omega) = \nu(x,x')\rho\left(\frac{x-x'}{|x-x'|},\Omega;x'\right) g(x,x'), \qquad (3.2)$$

where $\nu(x,x')$ is the visibility term, $g(x,x')$ is the geometry term, $\rho\left(\frac{x-x'}{|x-x'|},\Omega;x'\right)$ is the A-BRDF, and Ω stands for the direction of the rays. Henceforth, directions will also be specified either using unit-norm vectors or using a specific coordinate system on the sphere, (θ,ϕ), the zenital and azimuthal angles, respectively, as shown in Fig. 3.1.

All the physics of the problem is contained in the A-BRDF, which can be expressed as the following ratio (see Fig. 3.1),

$$\rho\left(\frac{x-x'}{|x-x'|},\Omega;x'\right) = \rho(\Omega_i;\Omega_o;x') = \frac{dL_o(\Omega_o;x')}{dE_i(\Omega_i;x')} \quad , \qquad (3.3)$$

where L_o is the radiance at a point x', exiting along the outgoing direction Ω_o; and E_i is the irradiance at the same point incident along the direction Ω_i. Again, x represents the location of the previous reflection of the sound ray.

The A-BRDF can be interpreted as the probability that sound energy incident from a direction Ω_i is reflected in the direction Ω_o.

As such, the A-BRDF must also be *normalized* in order to avoid an artificial increase of the acoustic energy in the system. For the sake of simplicity of notation, in the rest of this paper we will not consider explicit dependence of the A-BRDF on the reflection position x', and hence write $\rho(\Omega_i;\Omega_o)$.

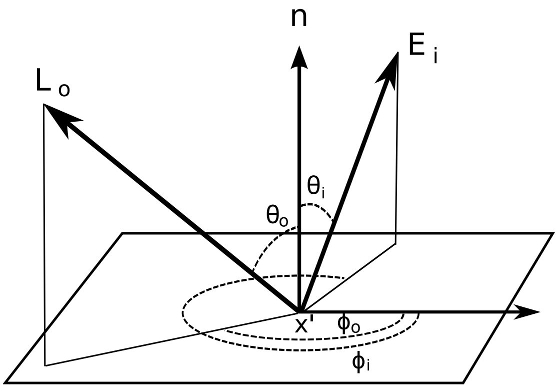


Figure 3.1: The A-BRDF returns the ratio of reflected radiance, L_0, to the incident irradiance, E_i, at point x_0.

3.3 General methodology for computing the A-BRDF

In this section we present a method to compute analytically the A-BRDF in cases where reflections suffer diffusion parameterized by a random variable, *i*.e. in cases where the vector $\vec{o}$ along the outgoing direction, Ω_o, is determined using a random vector, $\vec{r}$,

$$\vec{o} = \vec{o}(\vec{r}, \hat{\imath}), \tag{3.4}$$

governed by a probability density $p_3(\vec{r})$. The subindex denotes that p_3 is a 3-dimensional probability density, *i*.e. $p_3(\vec{r})d^3r$ yields the differential probability that the vector lies in the interval $(\vec{r}, \vec{r} + d\vec{r})$. We have chosen a random vector in $\mathbb{R}^3$ rather than a single random variable, to cover more general cases, such as the VBS model discussed in the next section. We have also included dependence on the incident direction of the sound ray, represented by the unitary vector $\hat{\imath}$ with direction Ω_i. [1]

To compute the A-BRDF in such cases, we will use Eq. (3.4), which relates the outgoing direction and the random vector $\vec{r}$, to obtain a relation between the *p*robability density of the outgoing direction, and $p_3(\vec{r})$. In other words, we will use Eq. (3.4) to

[1] Note that Lambert diffusion is an extreme case of (3.4) where the outgoing direction does not depend on the incoming one.

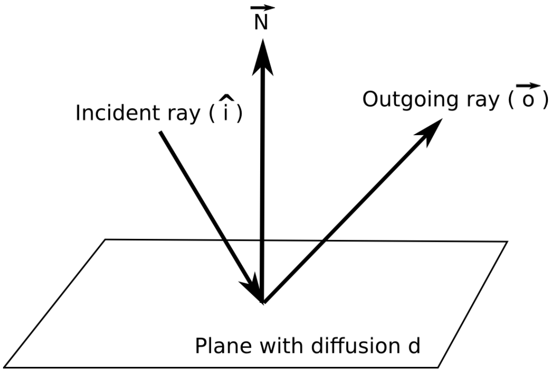


Figure 3.2: The outgoing ray $\vec{o}$ will be function of the incident ray $\hat{i}$ and other parameters.

obtain the probability that the outgoing ray is in a small neighborhood $(\Omega_o, \Omega_o + d\Omega_o)$, which is precisely the information provided by $\rho(\Omega_i;\Omega_o)$.

The relation between probability densities can be obtained recalling the way differential forms transform under a change of coordinates,

$$p_3(\vec{r})d^3r = \left|\frac{\partial\vec{r}}{\partial\vec{o}(\vec{r},\hat{\imath})}\right| d^3o p_3(\vec{o},\hat{\imath})\,, \tag{3.5}$$

which yields the following relation:

$$p_3(\vec{o},\hat{\imath}) = \left|\frac{\partial\vec{o}(\vec{r},\hat{\imath})}{\partial\vec{r}}\right|^{-1} p_3(\vec{r}) \quad . \tag{3.6}$$

Here, $\left|\frac{\partial\vec{r}}{\partial\vec{o}(\vec{r},\hat{\imath})}\right|$ is the Jacobian of the transformation defined by Eq. (3.4).

One last step is needed to obtain the A-BRDF. Note that $p_3(\vec{o},\hat{\imath})$ contains information about the probability that the outgoing vector $\vec{o}$ has modulus $o = |\vec{o}|$ between $(o, o+do)$, besides the probability that it has direction between $(\Omega_o, \Omega_o + d\Omega_o)$. Thus, to obtain $\rho(\Omega_o, \Omega_i)$, all that is left to do is an integration over all possible moduli:

$$\rho(\Omega_o,\Omega_i) = \int_0^\infty p_3(o,\Omega_o,\Omega_i)o^2do \ . \tag{3.7}$$

Note that the need for this last integration arises from the fact that relations of the form (3.4) are often written in a manner where the outgoing vector is not guaranteed to have unit norm, as will be shown in the next section.

The general method presented here can also apply to cases where the random vector is unitary, which is the case of VBS. In such cases, the probability density of the random unit vector $p_2(\hat{r})$ is actually two-dimensional, and it can be seen as the restriction to the unit sphere of the three-dimensional probability density $p_3(\vec{r})$,

$$p_3(\vec{r}) = p_2(\hat{r})\delta(r-1)\,, \tag{3.8}$$

where $r = |\vec{r}|$. This relation guarantees that probabilities can be indistinctly computed integrating $p_2(\hat{r})$ or $p_3(\vec{r})$ over two or three-dimensional intervals, respectively,

$$\int_\Sigma d\Omega\; p_2(\hat{r}) = \int_\mathbb{R} drr^2 \int_\Sigma d\Omega\; p_3(\vec{r}) \;\; . \tag{3.9}$$

3.4 A-BRDF for the Vector Based Scattering Model

Although several ray tracing simulations propagate rays using only specular reflections, one of the existing choices in room acoustic simulations is to include a diffusion model to determine the direction of the outgoing rays. The *Vector based scattering* (VBS) [**?**] is a method for room acoustics ray-tracing computations that makes use of a specific model to parameterize diffusion of sound waves by obstacles. Within VBS, the vector along the outgoing ray, $\vec{o}$, is calculated via a linear combination of two vectors

$$\vec{o} = d\,\hat{r} + (1-d)\hat{s}\;, \tag{3.10}$$

where $\hat{s}$ is a unit vector along the specular direction, $\hat{r}$ is a random unit vector, and $d \in [0,1]$ is the diffusion coefficient of the material (see Fig. 3.3). Clearly, $\hat{s}$ is entirely determined by the incident vector $\hat{\imath}$, and thus, Eq. (3.10) is an explicit example of the general Eq. (3.4).

In this case, the random vector $\hat{r}$ is uniformly distributed along the hemisphere whose equator is the plane of collision, and which contains the incoming and outgoing rays. The problem of determining the outgoing direction can be thought of that of adding a random vector of length d to the outgoing unit vector, after rescaling by $1-d$ (see Fig. 3.3).

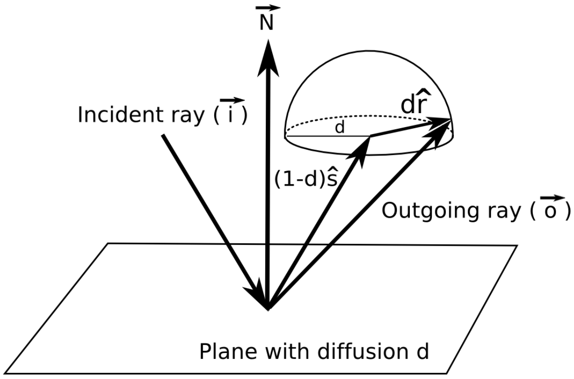


Figure 3.3: Scheme of the VBS, with $\hat{r}$ randomly generated within the upper hemisphere.

The rest of the section is organized as follows: in subsection 3.4.1 we focus on a slight variation of the VBS model whereby the random vector $\hat{r}$ is uniformly distributed along a whole unit sphere, rather than a hemisphere (Fig. 3.4). This variation simplifies the presentation of the computations and allows for a deeper analysis of the results. Subsection 3.4.2 presents the extension to the common case of the upper hemisphere that will be used in the implementation of our ray tracing engine in chapter 4. Analysis and numerical validations of the results are presented in both subsections.

3.4.1 Uniform distribution of the random vector $\hat{r}$ along a whole unit sphere

To obtain the A-BRDF we need to use Eq. (3.7), which in turn depends on the relation between probability densities (3.6). The VBS is an example where the random vector has unit norm, and therefore requires Eq. (3.8),

$$p_3(\vec{r}) = p_2(\hat{r})\delta(r-1) = \frac{1}{4\pi}\delta(r-1)\,, \tag{3.11}$$

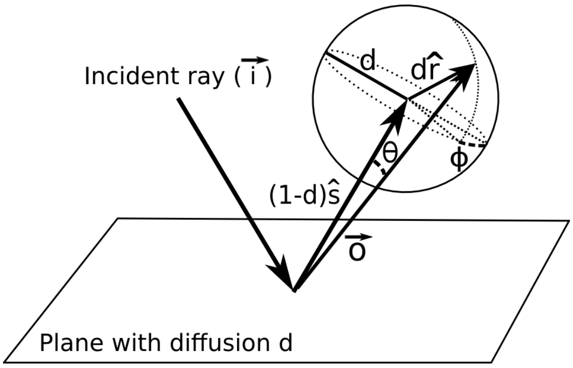


Figure 3.4: Scheme of the VBS, with $\hat{r}$ randomly generated within the whole sphere.

where $r = |\vec{r}|$, the Dirac δ-term ensures that the random vector is unitary (see Eq. (3.8)), and the choice $p_2(\hat{r}) = (4\pi)^{-1}$ ensures that the probability is normalized to one:

$$\int_{\mathbb{R}} dr r^2 \int_{S^2} d\Omega \; p_3(\vec{r}) = 1 \,. \qquad (3.12)$$

To complete the computation in Eq. (3.6), we need the Jacobian of the transformation (3.10), which turns out to be constant,

$$p_3(\vec{o}, \hat{\imath}) = \left| \frac{\partial \vec{o}(\vec{r}, \hat{\imath})}{\partial \vec{r}} \right|^{-1} p_3(\vec{r}) \;=\; \frac{1}{4\pi d^3}\, \delta(r-1) \,. \qquad (3.13)$$

To continue, we need to invert Eq. (3.10) and express the modulus $|\vec{r}|$ as a function of the outgoing and incident rays:

$$\begin{aligned} r^2 &= \vec{r}\cdot\vec{r} = \left(\frac{\vec{o} - (1-d)\hat{s}}{d} \right)^2 \\ &= \frac{1}{d^2}\left(o^2 - 2(1-d)o\gamma + (1-d)^2 \right) \,, \end{aligned} \qquad (3.14)$$

where $\gamma \equiv \hat{o}\cdot\hat{s}$ is the cosine between the specular and the outgoing directions. Note that expressing $\vec{r}$ as a function of $\vec{o}$ does not guarantee that $\vec{r}$ is unitary; this fact is only ensured by the delta-function in Eq. (3.13). The expression for the probability density of the outgoing vector is then given by:[2]

$$\begin{aligned} p_3(\vec{o},\hat{\imath}) &= \frac{1}{2\pi d^3}\delta\left(f(o,\gamma)\right), \\ f(o,\gamma) &\equiv \frac{1}{d^2}\left(o^2-2(1-d)o\gamma+(1-d)^2\right)-1\,. \end{aligned} \tag{3.15}$$

The last step to obtain the A-BRDF consists on inserting Eq. (3.15) in Eq. (3.7) and integrating over the radial coordinate. From (3.15), it is clear that, in this case, the A-BRDF depends on the incident and outgoing directions only through the combination γ. We shall thus write $\rho(\gamma)$ in the remainder of the document. From (3.7):

$$\rho(\gamma)=\int_0^\infty p_3(o,\hat{\imath})o^2 do=\int_0^\infty do\frac{o^2}{2\pi d^3}\delta\left(f(o,\gamma)\right)\,. \tag{3.16}$$

This integral is straightforward using a general formula for integrals containing Dirac's delta functions:

$$\int_a^b dx\, y(x)\delta(f(x))=\sum_{f(x_i^*)=0}\frac{y(x_i^*)}{|f'(x_i^*)|}\,, \tag{3.17}$$

where the sum is over all roots x_i^* of the function $f(x)$ in the interval (a,b). In our case, $f(o,\gamma)$ has at most two roots, $o_\pm$:

$$o_\pm=(1-d)\gamma \;\;\pm\;\; \sqrt{(1-d)^2\gamma^2-(1-2d)}\,. \tag{3.18}$$

Each of these roots contributes to the integral in Eq. (3.16) only when they are real and positive, the latter restriction due to the fact that the integral range is $(0,\infty)$. Thus, given that $o_+>o_-$, depending on the value of γ,d we obtain one of the following results

$$\begin{aligned} \rho(\gamma) &= 0, \\ \rho(\gamma) &= \frac{1}{2\pi d^3}\frac{o_+^2}{|f'(o_+)|}, \\ \rho(\gamma) &= \frac{1}{2\pi d^3}\left[\frac{o_+^2}{|f'(o_+)|}+\frac{o_-^2}{|f'(o_-)|}\right]. \end{aligned} \tag{3.19}$$

[2] We have used the equality $\delta(r-1)=2\delta(r^2-1)$ to avoid the use of square roots inside delta-functions.

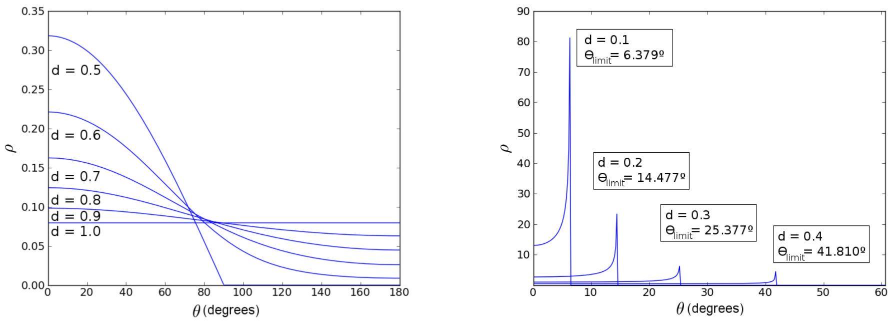


Figure 3.5: Analytical results for diffusion values larger (left) and lower (right) than $1/2$.

The condition that both solutions be real is

$$\gamma^2 \geq \frac{1-2d}{(1-d)^2} \quad , \tag{3.20}$$

which basically states that unless diffusion is high enough, there are outgoing angles that are unattainable for some incident directions. Indeed, it can be seen that Eq. (3.20) is always satisfied if $d \geq 1/2$ (all directions attainable), and not satisfied for some outgoing directions if $d < 1/2$ (presence of unattainable directions).

Similarly, only o_+ is positive if $d \geq 1/2$, which yields

$$\rho(\gamma) = \frac{2(1-d)^2\gamma^2 + 2d - 1}{4\pi d\sqrt{(1-d)^2\gamma^2 + 2d - 1}} + \frac{(1-d)\gamma}{2\pi d} \quad , \tag{3.21}$$

whereas both o_+ and o_- are positive if $d < 1/2$, which yields

$$\rho(\gamma) = \frac{2(1-d)^2\gamma^2 + 2d - 1}{2\pi d\sqrt{(1-d)^2\gamma^2 + 2d - 1}} \quad . \tag{3.22}$$

Fig. 3.5 shows various plots of the analytical results for the A-BRDF, Eqs. (3.21) and (3.22). The two extreme limits work as expected: when $d \to 1$, all outgoing directions are equally probable (Lambert diffusion limit); whereas when $d \to 0$, only outgoing directions close to specularity ($\theta = 0$) are attainable.

The results for $1/2 < d < 1$ are also intuitive: the probability density of the outgoing ray is maximum at specularity, and decreases progressively away from it. The maximum is less pronounced as we increase the diffusion coefficient attaining a uniform distribution for $d = 1$.

The results for $d < 1/2$ might look anti-intuitive at first sight. Besides the existence of a limit angle θ_L explained above, it is worth remarking that the maximum of the probability density does not lie at specularity, but very close to the limit angle. Indeed, the probability density diverges at θ_L. As shown in the next section, this divergence is actually integrable, and therefore leads to finite and meaningful measurable probabilities. To understand physical origin of this divergence note that, when $d < 1/2$, the outgoing vector $\vec{o}$ lies on the sphere centered at $(1-d)\hat{s}$, as shown in Fig. 3.3. The limit angle corresponds to a line with origin at the reflection point, and which is *t*angent to the sphere. The probability density that $\vec{o}$ lies in the interval $(\theta, \theta + \delta\theta)$ is basically the fraction of area of the sphere that lies between those two angles. At strict tangency, no matter how small $\delta\theta$, there is always a finite area lying in the range. [3]

To validate Eqs. (3.21) and (3.22) we will show that Monte Carlo experiments reproduce the same results in the limit of large number of rays. To avoid subtleties with the divergences explained above, we will compare probability distributions rather than probability densities; the former contains the same information as the latter, but it is necessarily absent of divergences.

The probability of having an outgoing ray contained in a finite solid angle Σ can be obtained from the probability density via

$$P(\Omega_o \in \Sigma, \Omega_i) = \int_\Sigma d\Omega_o \rho(\Omega_o, \Omega_i) \ . \tag{3.23}$$

Given that, as described above, $\rho(\Omega_o, \Omega_i)$ depends only on the cosine angle $\gamma = \cos\theta$, we will compare with Monte Carlo experiments the probability $F(\theta_0)$ that the outgoing ray has an angle θ with respect to the specular direction less than θ_0, irrespective of

[3] A simpler analogous problem is that of finding the fraction of length of the unit circle $x^2 + y^2 = 1$ that lies in a small interval $(x, x + dx)$. The answer diverges at $x = \pm 1$ despite the fact the the total length is, obviously, finite. At those points, $x = const.$ lines intersect the circle tangentially.

the rotational angle ϕ,

$$F(\theta_0) = P(0 \leq \theta \leq \theta_0, \forall\phi) \ . \tag{3.24}$$

Our analytical prediction for $F(\theta_0)$ follows from (3.21) and (3.22):

$$\begin{aligned} F(\theta_0) &= \int_0^{2\pi} d\phi \int_0^{\theta} \sin\theta \, d\theta \rho(\cos\theta) = 2\pi \int_{\gamma_0}^{1} d\gamma\rho(\gamma) \\ &= 2\pi \left[G(1) - G(\gamma_0)\right] \end{aligned} \tag{3.25}$$

where $\gamma_0 = cos(\theta_0)$, and $G(\gamma)$ is the indefinite integral of $\rho(\gamma)$, which can be computed analytically. For $d \geq 1/2$,

$$G(\gamma) = \frac{\gamma^2(1-d)}{4\pi d} + \frac{\gamma\sqrt{(1-d)^2\gamma^2 + 2d - 1}}{4\pi d} ; \tag{3.26}$$

whereas for $d < 1/2$,

$$G(\gamma) = \frac{\gamma\sqrt{(1-d)^2\gamma^2 + 2d - 1}}{2\pi d} \ . \tag{3.27}$$

We will now show that Eqs. (3.26) and (3.27) provide same result as Monte Carlo experiments in the limit of large number of rays. In order to do so, consider a single plane characterized by a diffusion coefficient d. To simplify the experiment, let us set the direction of the incident sound ray to be orthogonal to the plane, which leads to a specular vector, $\hat{s}$, that coincides with the normal to the plane. For a given value of the diffusion coefficient, N random 3-vectors $\vec{r}_j$, $\{j = 1...N\}$ are produced and added to the specular vector following the VBS equation (3.10), thus obtaining N outgoing rays $\vec{o}_j$.

Fig. 3.6 shows the histogram (red points) corresponding to the outgoing directions of the vectors $\vec{o}_j$ and its comparison to the analytical results (blue continuous lines). A large number of random rays $N = 10^5$ was used. The agreement is excellent.

3.4.2 Uniform distribution of the random vector $\hat{r}$ along the upper hemisphere

Having the angle θ (the angle between the specular direction $\vec{e}$ and the direction to the sound source $\vec{t}$, see fig.3.4), and the intersection

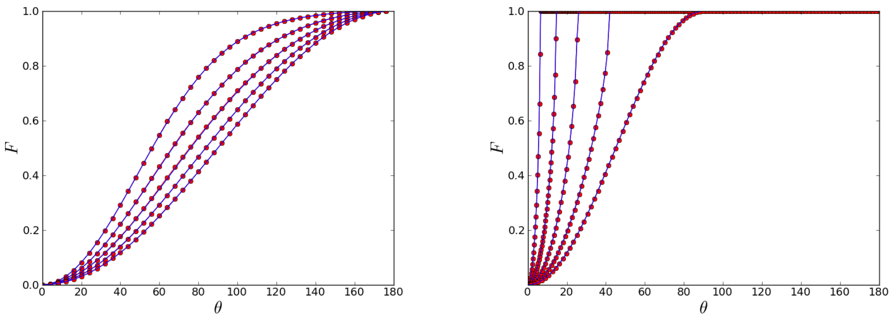


Figure 3.6: Analytic curves (blue lines) for the distribution function compared to Monte Carlo results (red dots) based on $N = 10^5$ random rays, for $d \geq 1/2$ (left), and $d < 1/2$ (right).

of $\vec{t}$ with the hemisphere of radius equals to the diffusion, what we want is to define an area around the intersection point (let us name it i) with the same size that the projected solid angle of the sound source to the hemisphere.

Knowing that the solid angle Ω can be defined as

$$\Omega = \frac{\pi SourceRadius^2}{DistanceToSource^2}, \tag{3.28}$$

we want to define the amount of area in terms of θ and ϕ. We can then define $\Delta\theta$ as

$$\Delta\theta = \frac{aSourceRadius}{DistanceToSource}, \tag{3.29}$$

Where a is a function that goes from 2.0 to $sqrt(\pi)$ depending on the angle θ:

$$a = 2.0 + \frac{\theta}{\theta_{limit}}(-2.0 + \sqrt{\pi}) \tag{3.30}$$

Knowing that

$$\Omega = \Delta\phi \int \sin\theta d\theta = \Delta\phi \left[\cos\left(\theta - \frac{\Delta\theta}{2}\right) - \cos\left(\theta + \frac{\Delta\theta}{2}\right)\right], \tag{3.31}$$

we can define $\Delta\phi$ as

$$\begin{aligned}\Delta\phi &= \frac{\Omega}{\left[\cos\left(\theta-\frac{\Delta\theta}{2}\right)-\cos\left(\theta+\frac{\Delta\theta}{2}\right)\right]} \\ &= \frac{\pi SourceRadius^2}{DistanceToSource^2\left[\cos\left(\theta-\frac{\Delta\theta}{2}\right)-\cos\left(\theta+\frac{\Delta\theta}{2}\right)\right]}.\end{aligned} \tag{3.32}$$

We want to compute the probability distribution P keeping a constant solid angle of the same size that the exact one. We can define P as follows:

$$P = \Delta\phi\int_{\gamma_1}^{\gamma_2} d\gamma p(\gamma) = \Delta\phi(F(\gamma_2)-F(\gamma_1))\alpha \tag{3.33}$$

where $\gamma_1 = \cos\left(\theta+\frac{\Delta\theta}{2}\right)$, $\gamma_2 = \cos\left(\theta-\frac{\Delta\theta}{2}\right)$ and $\gamma = \cos(\theta)$.

There is an extra factor α in the definition of P that scales the result depending on the amount of solid angle that is projected over the upper hemisphere, we will explain how to compute this term in the following sections. It is also important to remember here that there can exist two intersections of $\vec{t}$ with the upper hemisphere, and therefore we must compute P for all the existing solutions and obtain the probability distribution by adding them.

We have at most two possible solutions for the vector-hemisphere intersection problem; vector $\vec{t}$ may not intersect the hemisphere (0 solutions), may intersect the hemisphere at one point (one solution) or may intersect the hemisphere at two points (2 solutions). Let us call p_+ to the formula that gives us the probability density function for the nearest intersection to the sound source position and p_- to the other one.

Therefore, depending on the solution evaluated we are going to deal with two different expressions:

$$p_+ \longrightarrow \int d\gamma p_+(\gamma) = \frac{\gamma\sqrt{(1-d)^2\gamma^2+2d-1}}{2\pi d} + \frac{(1-d)\gamma^2}{2\pi d} \tag{3.34}$$

$$p_- \longrightarrow \int d\gamma p_-(\gamma) = \frac{\gamma\sqrt{(1-d)^2\gamma^2+2d-1}}{2\pi d} - \frac{(1-d)\gamma^2}{2\pi d} \tag{3.35}$$

where d is the diffusion.

The last step in order to obtain P is to compute the amount of solid angle that is projected over the hemisphere. As one can easily see, depending on the position of the sound source there will be some cases in which only part of the solid angle will be laying on the upper hemisphere, and we have to take it into account. The computation of α also depends on the solution studied (t_+ or t_-).

Having the intersection point of the source vector with the hemisphere we can use $\Delta\theta$ and $\Delta\phi$ to obtain the 4 points forming the approximated area, A$(-\Delta\theta/2, -\Delta\phi/2)$, B$(-\Delta\theta/2, +\Delta\phi/2)$, C$(+\Delta\theta/2, -\Delta\phi/2)$ and D$(+\Delta\theta/2, +\Delta\phi/2)$.

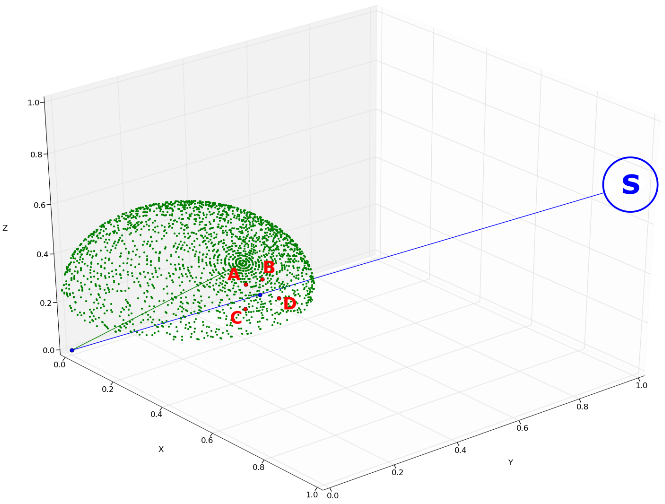


Figure 3.7: Finding the four points that delimit the solid angle approximation.

Once we have these points over the hemisphere we evaluate if they are lying over the upper hemisphere by comparing the angle ϕ of each point with the angle ϕ that intersects the equator of the sphere given the same angle θ of the evaluated point.

The cosine of the angle ϕ that intersects the equator of the sphere also depends on the solution studied (t_+ or t_-). For instance, we can compute it for the positive solution as follows:

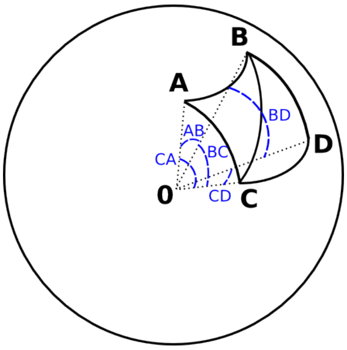


Figure 3.8: Scheme of the angles needed to compute the area.

$$cos(\phi)_{t_+} = \frac{specularDir_z t_+ \cos(\theta) - (1.0 - diffusion) specularDir_z}{specularDir_x t_+ \sin(\theta)}. \tag{3.36}$$

The next step is to compute the total area delimited by the four points. To do this we divide it in two triangles,ABC and BCD, and we apply the formula to compute the areas of both spherical triangles, given the angles AB,BC,CA and BD,CD,BC respectively (see fig. 3.8).

$$SphericalTriangleArea = d^2(angleA + angleB + angleC - \pi), \tag{3.37}$$

where *angleA*,*angleB* and *angleC* are the spherical angles computed as follows:

$$angleA = \arccos\left(\frac{\cos(BC) - (\cos(CA)\cos(AB))}{\sin(CA)\sin(AB)}\right) \tag{3.38}$$

$$angleB = \arccos\left(\frac{\cos(CA) - (\cos(AB)\cos(BC))}{\sin(AB)\sin(BC)}\right) \tag{3.39}$$

$$angleC = \arccos\left(\frac{\cos(AB) - (\cos(BC)\cos(CA))}{\sin(BC)\sin(CA)}\right). \tag{3.40}$$

Once we have the total area, if all four points lay over the upper hemisphere we just return $\alpha = 1.0$, but in case that one or more points do not lay over the hemisphere then we have to find the intersection points with the equator and define one or two new spherical triangles (depending on the number of points under the hemisphere) to compute the amount of area that we have to substract from the total, as shown in fig.3.9

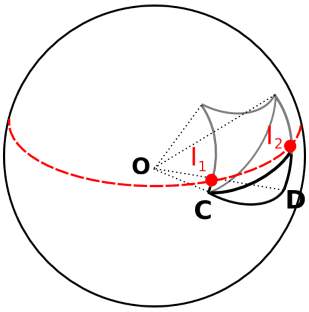


Figure 3.9: Scheme of the new spherical triangles defined using the intersections with the equator to compute the area to substract.

Once we compute the area cut following the same process explained then we finish returning

$$\alpha = \frac{totalArea - cutArea}{totalArea}. \tag{3.41}$$

To validate Eqs. (3.34) and (3.35) we will follow the same Monte Carlo experiments we presented in subsection 3.4.1 to reproduce the same results in the limit of large number of rays.

In order to do so, we will consider again a single plane characterized by a diffusion coefficient d. To simplify the experiment, let us set the direction of the incident sound ray to be orthogonal to the plane, which leads to a specular vector, $\hat{s}$, that coincides with the normal to the plane.

For a given value of the diffusion coefficient, N random 3-vectors $\vec{r}_j$, $\{j = 1...N\}$ are produced and added to the specular vector following the VBS equation (3.10), thus obtaining N outgoing rays $\vec{o}_j$.

Fig. 3.10 shows the histogram (red points) corresponding to the outgoing directions of the vectors $\vec{o}_j$ and its comparison to the analytical results (blue continuous lines). A large number of

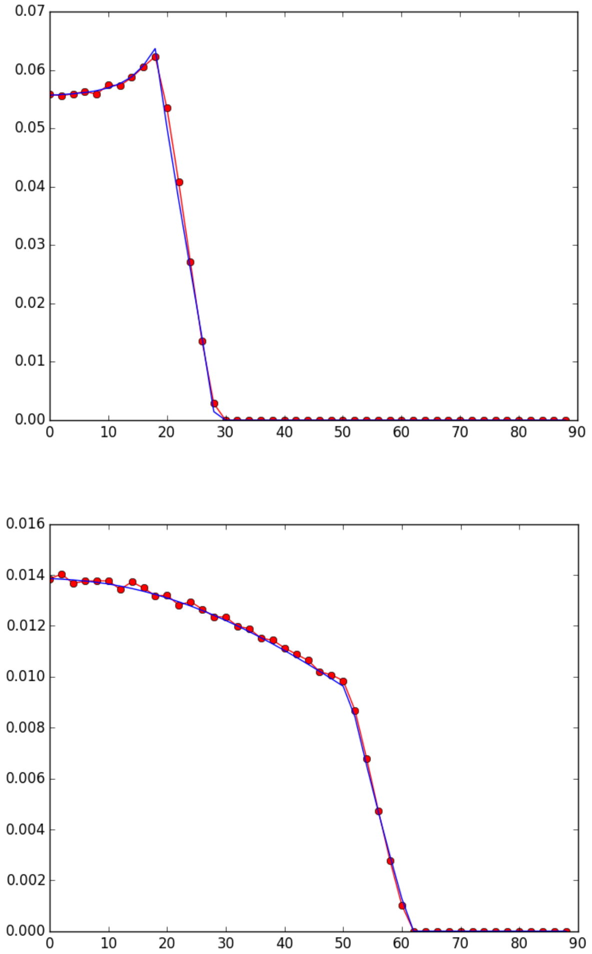


Figure 3.10: Analytic curves (blue lines) compared to Monte Carlo results (red dots) based on $N = 10^6$ random rays, for $d = 0.3$ (top), and $d = 0.6$ (bottom).

random rays $N = 10^5$ was used. The agreement is as excellent as it was in the simplified case of the full sphere validated previously.

3.5 Conclusions

A method to derive analytical solutions for the Acoustic Bidirectional Reflectance Distribution Function (A-BRDF) in cases of diffuse reflections parametrized by random variables has been presented and discussed. The method works for generic relations between outgoing, incoming and random vectors of the form (3.4), and makes use of various properties of continuous probability functions, exploiting the relation between two and three dimensional probability densities.

The method has been applied to the well-known Vector Based Scattering Model, for which exact analytical A-BRDF have been obtained, Eqs. (3.21) and (3.22). The results provide, by means of only one analytical formula evaluation, the same results as the corresponding Monte Carlo simulation in the limit of large number of rays.

The computation of analytical solutions for the A-BRDF can be added to a standard acoustic ray tracing engine by introducing an extra step in the algorithm to compute the contribution of every ray collision with the environment to the final Impulse Response. That is, instead of only using the information coming from the rays that reach the listener position to compute the Impulse Response, valuable data can be obtained from each collision that contributes to the computation of the Impulse Response thus reducing significantly the computational cost, as shown in Fig. 3.11. The use of this methodology could imply a reduction of about 10^3-10^4 rays in any of the existing comercial acoustic ray tracing engines. Further work will focus on the application of the method discussed here to other ray tracing diffusion models for acoustics and on the computation of real time Impulse Responses for simple environments through the method introduced in this section.

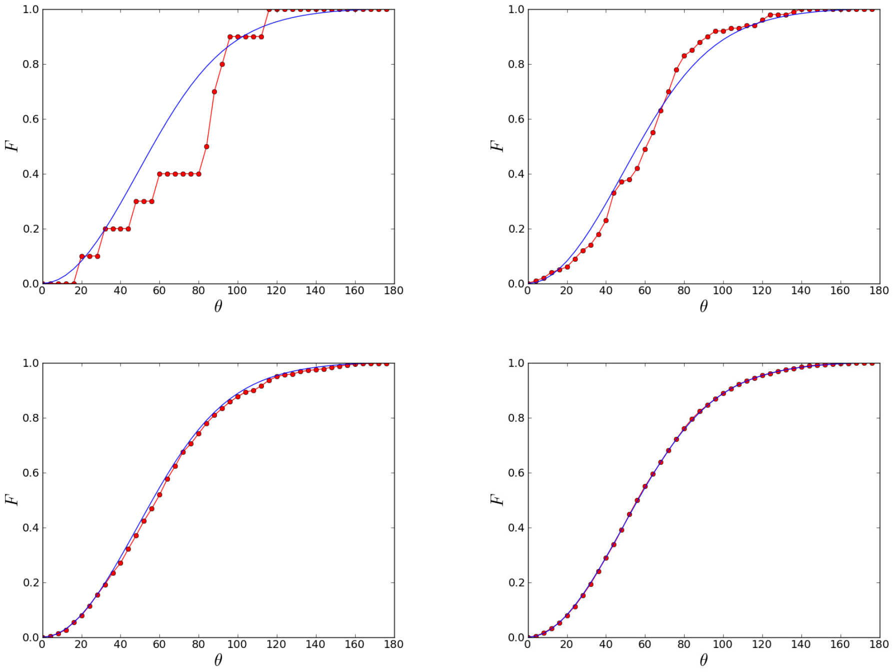


Figure 3.11: Analytic curves (blue lines) compared to Monte Carlo results (red dots), for $d = 0.6$ and different number of random rays: $N = 10^1, 10^2, 10^3, 10^4$, ordered from left to right and top to bottom.

Chapter 4

Ray tracing for 3D sound rendering

4.1 Introduction

We have designed and implemented a ray tracing engine that computes the impulse responses for the pressure and the velocity in any given 3D environment. Once the application is started, the computation is supported by a 3D graphical application that allows the user to interact with the different actors involved in the computation.

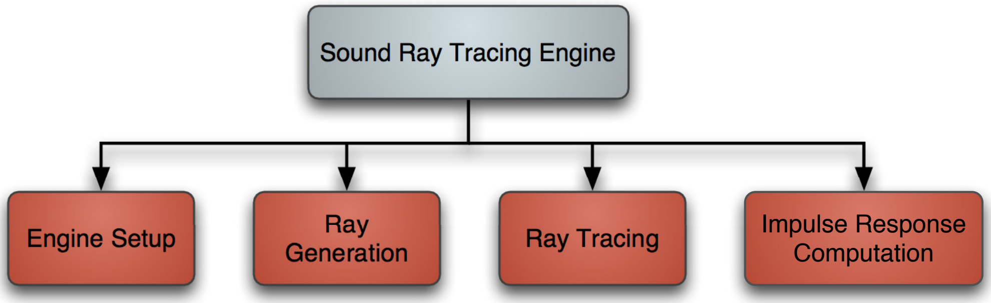


Figure 4.1: Main modules of the sound ray tracing engine developed at the Audio Group at Barcelona Media.

The whole engine is programmed in C++ [Stroustrup (2001)] and makes use of OpenGL libraries [Neider et al. (1994)] to generate the graphic output. The idea was to implement a ray tracing engine so as to have a basic application to test all the subsequently research done. It was not thought to be neither fast nor optimized ray tracing engine, but a research tool based on the *b*rute force method (i.e., a method that proofs all the cases).

The structure of this initial version can be subdivided in four main modules regarding the steps that are needed to perform the computation of the impulse response, as illustrated in fig. 4.1: The setup, the ray generation, the ray tracing and the impulse response generation. In the following sections we will go through them with more detail.

After the presentation of the main modules we will introduce a case study of the use of our ray tracing engine to build a gallery of virtual environments for the IP-RACINE European Integrated Project [Racine (2009)]. The main objective of IP-RACINE was to extend the state of the art and enhance European competitiveness in the cinema industry by creating technologies to deliver enhanced cinematic entertainment that is transferable cross-platform.

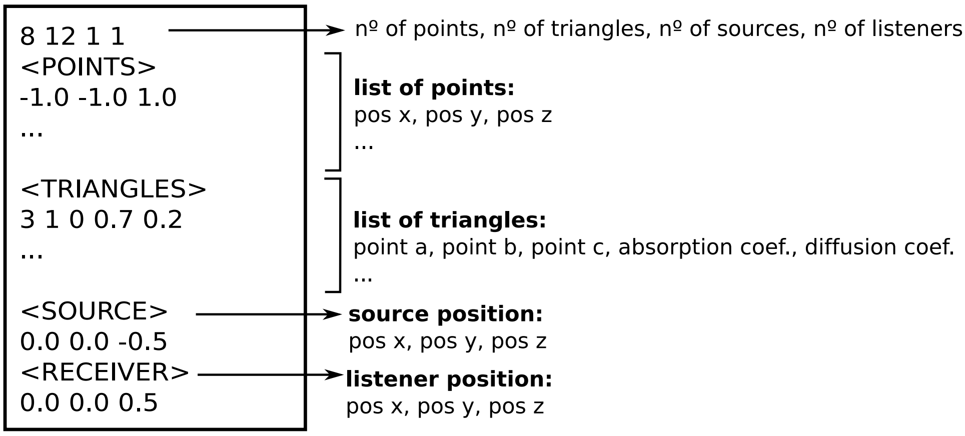
We will conclude this chapter reviewing the use of the impulse responses for the pressure and the velocity that we generate with our engine to render 3D sound.

4.2 Ray Tracing engine setup

The inputs we have included in our ray tracing engine to setup the computation of the impulse responses are the following:

- *3*D scene: One of the inputs needed is the specification of the 3D scene. More precisely, what our application requires is the geometry of the environment (i.e., all the polygons in the scene) and the two parameters that are used to define the features of any wall of the scene: the absorption coefficient and the diffusion coefficient (see section 4.4).

  To obtain this information we have implemented a simple plugin for *3*D Studio Max [3dsmax (2007)] (a 3D modeling software) with its own scripting language, *M*axScript that exports all the triangles contained in the scene and the two acoustical parameters associated to each triangle to a file having the structure shown in fig. 4.2.



```
8 12 1 1
<POINTS>
-1.0 -1.0 1.0
...

<TRIANGLES>
3 1 0 0.7 0.2
...

<SOURCE>
0.0 0.0 -0.5
<RECEIVER>
0.0 0.0 0.5
```

Figure 4.2: Structure of the file that we obtain when we export all the information using the plugin implemented for 3D Studio Max.

Then the engine is able to read all this information and store it. One possible improvement to be considered in the near future is to add the feature of reading some standard format such as *3*ds and thus be able to work with other 3D modeling software alternatives like blender [Blender (2007)]. Moreover, that would gives us the option to work with the whole information of the 3D scene (e.g. lighting, textures, ...).

- *s*ound source and listener positions: Both positions are also specified using the 3D modeling software, and they are exported through the same implemented plugin, as shown in fig. 4.2.

  Although most of the experiments have been done using one sound source and one listener position, the engine has been implemented with the possibility of working with any number of sound sources.

- *n*umber of rays and number of reflections per ray: The number of sound rays that the sound source is going to throw in the environment, and also the number of reflections that are going to be computed for each ray.

The natural way of finishing the computation of any sound ray would be to consider a threshold intensity of the ray, but by now we do not define any cutoff because we prefer obtaining as much information as possible to know exactly what it is done.

In the current version of the engine all the needed information to compute impulse responses is defined at execution time and to manage all the application parameters we make use of Boost C++ Library [Boost (2015)]. We have merged all the predefined parameters in one single configuration file in order to facilitate any change. In that file there are not only the parameters already mentioned such as the number of rays and the number of reflections but also some other parameters related to the graphics generation or some physical quantities needed such as the velocity of sound and the density of air.

4.3 Ray Generation

The acoustic pulse is simulated by standard Monte Carlo techniques. A large number of rays are thrown in random directions at the same time from the source's position using two random angles, ϕ and θ, as shown in figure 4.3.

We define any sound source with the following set of parameters:

- *p*osition: As we already commented, the sound source position is acquired from the 3D model of the environment, exported with the plugin we have implemented (see fig. 4.2).

- *n*umber of rays: The number of rays that are going to be used for the computation of the impulse response of the environment is defined in the configuration file that we mentioned before.

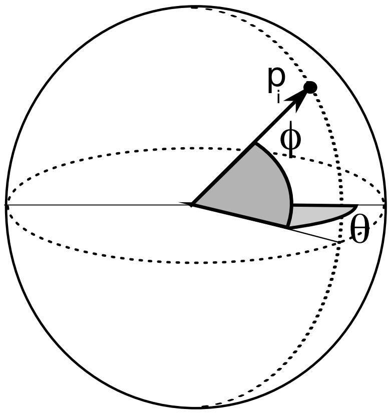


Figure 4.3: Standard Monte Carlo technique to sample uniformly all the points p_i of the sphere.

- *r*adius: The sound source is considered an omnidirectional and punctual emitter (i.e., all the rays thrown have the same origin). Anyway, we need the radius of the source for the graphical representation; it does not affect to the simulation.

We have implemented an array of rays to store at each position the actual state of all the rays. It has a size equal to the *n*umber of rays parameter already commented. All the rays that we generate from the sound source contain this information:

- *d*irection and origin: these two vectors contain the origin positon, which is the same as the sound source position, and the actual direction of the ray. The initial direction is chosen randomly.

- *n*umber of reflections: this variable will contain the actual number of reflections that the ray has experienced.

- *d*istance: This variable stores the total distance traveled by the ray. It will be essential to later be able to know exactly the time in which it gets to the listener [1].

[1] We will only use the distance traveled in case that the ray arrives to the listener position at some point of the simulation.

- *p*ressure: The pressure of all the rays is the same in the start time (i.e., we assign a pressure $p = 1$ to all the generated rays). As the ray travels through the environment and collides with it, this pressure decreases.

- *f*requency: Unlike most ray tracers for light rendering, it is crucial to assign a frequency to each ray[2]. This way, we also get information about the *p*hase of the ray, and have into account phenomena like interference between out-of-phase and in-phase waves. Thus each thrown ray is assigned a random frequency within the audible range (20 Hz-20 KHz)[3].

Again, we have defined an array of reflections that contains a sorted list of all the points in which the ray has collided with any of the walls of the environment. This information is not relevant for the simulation but for the graphical representation of the way followed by any of the rays.

4.4 Ray Tracing

Each generated ray is assigned an initial pressure, and then propagates in the scene according to a set of rules that share many similarities with those of light rendering. In the absence of collisions, a ray propagates in straight lines, its pressure phase varies according to a oscillatory function of its frequency, and its pressure modulus decays inversely proportional to the distance traveled.

In practice, the implemented ray tracing engine stores the total traveled distance for each ray and when a ray arrives to the listener then uses the traveled distance to decrease its pressure inversely proportional to this value.

When a collision with a surface takes place, the *s*urface impedance function determines the pressure attenuation and its change in phase according to standard physics laws. In particular, the absorption coefficient is easily determined as a function of the surface

[2] The reason being again the much larger wavelength of audible sound as compared to light.

[3] The range of frequencies can be easily changed in the configuration file.

impedance and the incidence angle. Finally, and extra *s*urface diffusion coefficient determines the angle of the reflected ray; essentially, the lower the diffusion coefficient, the closer to specular reflection.

As we previously introduced in section 3.4, we make use of the *Vector based scattering* (VBS) [Christensen and Rindel (2005)] to parameterize diffusion of sound waves by obstacles. Within VBS, the vector along the outgoing ray is calculated via a linear combination of two vectors (see Eq. 3.10).

Consequently, in our sound ray tracing engine we have defined two different coefficients for each wall, one for the absorption of the material and another for the diffusion, both in the range [0,1].

- *a*bsorption: When a ray collides with a wall, the absorption coefficient is applied by a simple product of the arriving pressure p_0 with the absorption coefficient α of the wall i, obtaining the pressure of the reflected ray p_{final}:

$$p_{final} = p_0\ \alpha_i \quad . \tag{4.1}$$

- *d*iffusion: Then we also apply the diffusion coefficient β of the wall i to the incident ray by considering the influence of the specular direction $\vec{d}_{spec}$ and a random direction $\vec{d}_{rand}$, previously computed, to obtain the final direction $\vec{d}_{final}$:

$$\vec{d}_{final} = \vec{d}_{spec}(1-\beta_i) + \vec{d}_{rand}\beta_i \quad . \tag{4.2}$$

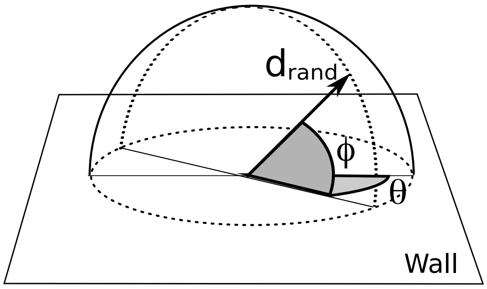


Figure 4.4: We define two angles ϕ and θ to obtain the random direction needed to compute the diffusion coefficient of a given wall.

This random direction $\vec{d}_{rand}$ mentioned above is computed using the 3x3 rotation matrix of the wall R_i and two new random angles $\phi \in [0, \frac{\pi}{2}]$ and $\theta \in [0, 2\pi]$ regarding the wall (see fig. 4.4):

$$\begin{aligned}\vec{d}_{rand} = \quad (\quad & (R_i[0][0]d^x_{rand}) + (R_i[0][1]d^y_{rand}) + (R_i[0][2]d^z_{rand}), \\ & (R_i[1][0]d^x_{rand}) + (R_i[1][1]d^y_{rand}) + (R_i[1][2]d^z_{rand}), \\ & (R_i[2][0]d^x_{rand}) + (R_i[2][1]d^y_{rand}) + (R_i[2][2]d^z_{rand})) \quad ,\end{aligned}$$

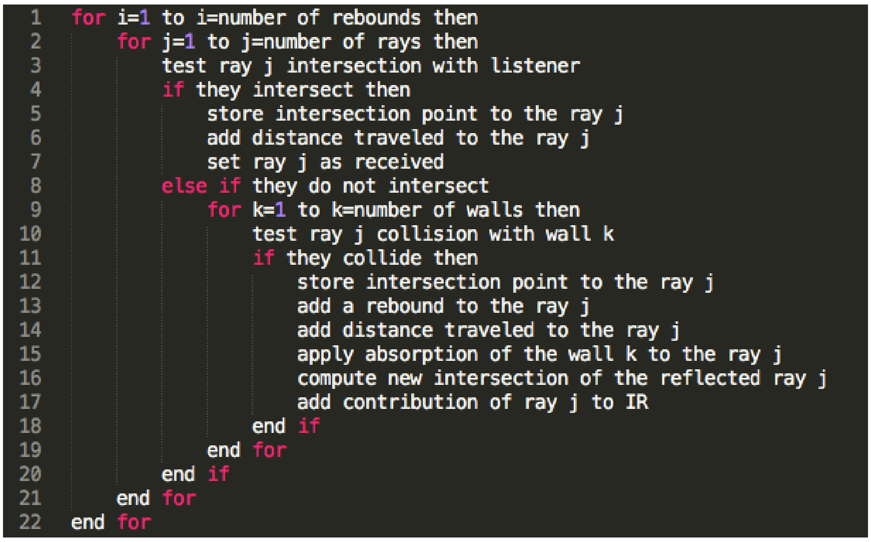
where:

$$\begin{aligned} d^x_{rand} &= \sin\theta\sin\phi, \\ d^y_{rand} &= \cos\theta, \\ d^z_{rand} &= \sin\theta\cos\phi \quad . \end{aligned}$$

Now that we have defined the way we process the collisions of the rays with the environment and the effect of the distance traveled in the final pressure carried by the sound ray, then we can introduce the main algorithm of the engine, which is actually the core function of any ray tracing (see fig. 4.5). As it can be observed, the algorithm first test if a ray has reached the listener, and if not it looks for the intersection with the environment[4]. Once it finds the wall in which the ray collides then it store the intersection point and modifies the direction and the pressure of the reflected ray regarding the parameters of the wall.

The main contribution of our research is the *Add contribution to IR* step shown in the main algorithm (see fig. 4.5). Besides continuing each ray through the environment in the same way any standard ray tracing engine would perform, we have introduced a step in our algorithm that computes the contribution of every ray collision with the environment to the final Impulse Response. That is, instead of using only the information coming from the rays

[4] This algorithm has been simplified but in reality it contains some more steps to choose the correct collision from all the detected possibilities



```
for i=1 to i=number of rebounds then
    for j=1 to j=number of rays then
        test ray j intersection with listener
        if they intersect then
            store intersection point to the ray j
            add distance traveled to the ray j
            set ray j as received
        else if they do not intersect
            for k=1 to k=number of walls then
                test ray j collision with wall k
                if they collide then
                    store intersection point to the ray j
                    add a rebound to the ray j
                    add distance traveled to the ray j
                    apply absorption of the wall k to the ray j
                    compute new intersection of the reflected ray j
                    add contribution of ray j to IR
                end if
            end for
        end if
    end for
end for
```

Figure 4.5: Schema of the main ray tracing algorithm.

that have reached the listener position to compute the impulse response, we get data from each collision that contributes to the computation of the impulse response thus reducing dramatically the computational cost.

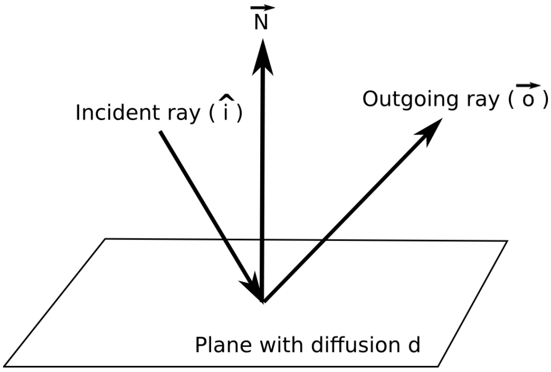


Figure 4.6: The outgoing ray $\vec{o}$ will be function of the incident ray $\hat{i}$ and other parameters.

As we have seen in chapter 3, we have developed a method to compute the analytical solution for the A-BRDF in cases where sound reflections are diffuse. The method is applied to the Vector

Based Scattering Model [Christensen and Rindel (2005)] (VBS), which is one of the existing ways to efficiently include diffusion in a ray tracing algorithm by means of random vectors.

We include this analytical solution for the A-BRDF in our ray tracing engine at every ray collision. As we can observe in fig. 4.6, what we want to compute is the probability that sound energy incident from direction Ω_i is reflected in the direction Ω_o, being Ω_o the direction to the listener position. We use this information as a coefficient that is applied by a simple product of the arriving pressure p_0 with the probability that this pressure is reflected towards the listener position (see eqs. 3.34 and 3.35), obtaining the final pressure p_{final} that arrives to the listener position and is used to compute the final impulse responses for the pressure and the velocities.

$$p_{final} = p_0\, \rho(\Omega_o, \Omega_i) \quad . \tag{4.3}$$

Once all the computation is finished the application shows the 3D environment containing the sound source and the listener (represented by a red sphere and a white sphere, respectively), as illustrated in fig. 4.7a. It is also possible to show all the rays that have arrived from the sound source to the listener position (fig. 4.7b) and all the rays traced during the computation (fig. 4.7c).

4.5 Impulse Response Generation

Everytime that a ray collides with the listener, or when we compute the contribution of each ray-environment collision to the final impulse response as presented in the previous section, the associated pressure at the arrival time is stored, together with its incoming direction. The last step is to discretize the time axis in bins of length equal to inverse of the sampling rate [5], and add the pressure and velocities of all rays in a given bin.

We have defined the listener with the following parameters:

[5] For example, for CD quality, the sampling rate is 44100 Hz, corresponding to a bin size of 0.022 *ms*.

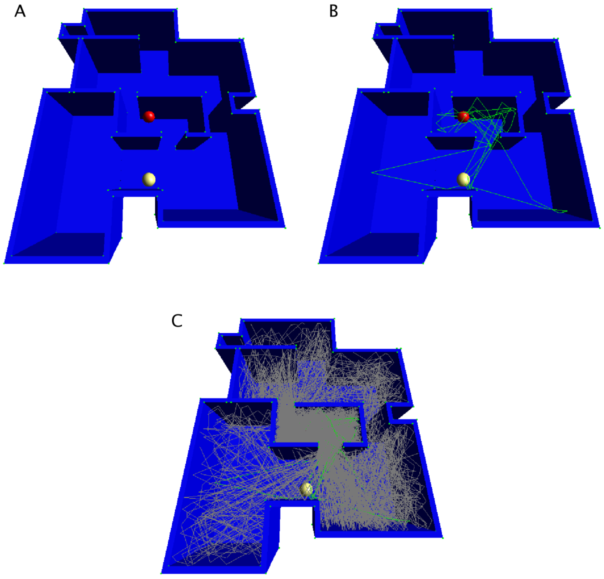


Figure 4.7: Screenshots of the graphic output after the computation. The environment with the sound source in white and the listener in red (A), the rays that arrive from the source to the listener (B) and all the rays traced during the computation (C).

- *p*osition: As we previously explained, the listener position is acquired together with the sound source position from the 3D model of the environment, exported with the plugin we have implemented (see fig. 4.2).

- *n*umber of received rays: This variable stores the total number of rays that have arrived from the sound source to the listener position.

- *r*adius: In this case the radius of the listener affects to the accuracy of the final result. The impulse response has to be calculated at one point of the environment, and the ideal case would be to consider the listener not as a sphere but as a point. However, in practice it is not possible if we want to obtain a big number of rays arriving to the listener position in a reasonable time of computation.

> The most important information is the impulse response computed from all the collisions. It is stored in a structure containing the discretization of the time axis and the pressure at all discrete times.

Once the engine has computed the impulse response of the environment given the sound source and the listener positions, then it is able to convolve it with a predefined input sound. The user can listen to the final rendered audio by just pushing one key. The convolution module of the engine has been implemented following step by step the method proposed in [Press et al. (1992)]. And it has also been necessary to implement a module for reading and writing *w*ave audio data files.

As we will see in the following section, we use the incoming directions stored for each ray to obtain three more impulse responses for the three components of the velocity. This information is key to render 3D sound.

4.6 3D Sound Rendering

Despite the advanced state of the art in 3D audio, the expression "surround sound" still means horizontal surround sound to most of the music and cinema consumers and professionals [Cengarle (2012)]. The most common formats beyond stereo are in fact 5.1 surround and its extension, the 7.1 surround. 5.1 is a format which uses a standard stereo Left-Right pair, a Center channel and two channels for Rear Left Surround and Rear Right Surround, plus a channel for Low Frequency Effects (LFE) via a dedicated subwoofer. The 7.1 for music adds two rear surround channels whose recommended position is between $\pm 135^o$ and $\pm 150^o$, while the two surround channels from the 5.1 are displaced between $\pm 90^o$ and $\pm 110^o$. For movie theaters, the 7.1 surround splits the existing left and right surround lines of speakers into four channels, to accommodate the Back Left Surround and Back Right Surround channels. 5.1 surround is a sort of extension of stereo on the horizontal plane oriented to movie content: the presence of the Center channel helps anchoring the dialogs to the screen; the L-R pair is used for front panning and stereo content, and to maintain compatibility with the previous standard, while the two rear channels are expected to provide ambience and enveloping sound, together with occasional sound effects.

In recent years, 3D sound has appeared in the cinema industry; currently, four companies offer solutions that include loudspeaker setups in the upper hemisphere, tools for doing the audio postproduction in 3D and equipment to playback the resulting soundtracks: immsound [Immsound (2015)], Auro3D [Auro3D (2015)], iosono [Iosono (2015)] and Dolby [Dolby (2015)].

Stereo sound is the oldest, most used and most known sound spatialization technique. In stereophony, speaker levels are adjusted in order to (virtually) position the sound source in a point between the speakers. The location perception is accomplished by the several psychoacoustic mechanisms of spatial sound perception [Gerzon (1974)]. The concept of sound level balancing or panning has been further extrapolated to multi-channel stereophony, commercially known as surround sound formats, such as 5.1 or 7.1, but this paradigm presents a major drawback: a mix produced for a specific speaker layout can only be successfully reproduced (according to the mixing intended purpose) by the exact same speaker setup. Less speakers will unfailingly provide an incomplete sonic representation, while more speakers will not add any extra information.

In opposition to this approach that considers specific number of channels and is dependent of the speaker setup, we can think of systems that overcome their limitations. For instance, it would be desirable to have a layout-independent technique, which might be extrapolated to any desired speaker configuration.

One of the most common spatialization techniques that are used nowadays and benefit from the sound scene abstraction approach is Ambisonics, a complete sonic theory developed by M. Gerzon and P. Fellgett in the 1970s. Ambisonics is a full-sphere surround sound technique: in addition to the horizontal plane, it covers sound sources above and below the listener [Gerzon (1973)]. Unlike other multichannel surround formats, its transmission channels do not carry speaker signals. Instead, they contain a speaker-independent representation of a sound field called B-format, which is then decoded to the listener's speaker setup. This extra step allows the producer to think in terms of source directions rather than loudspeaker positions, and offers the listener a considerable degree of flexibility as to the layout and number of speakers used for playback.

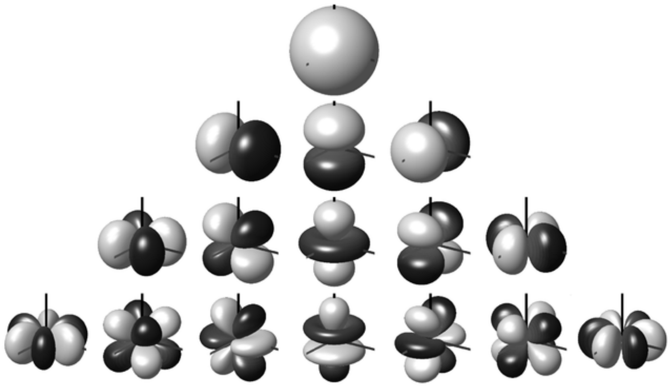


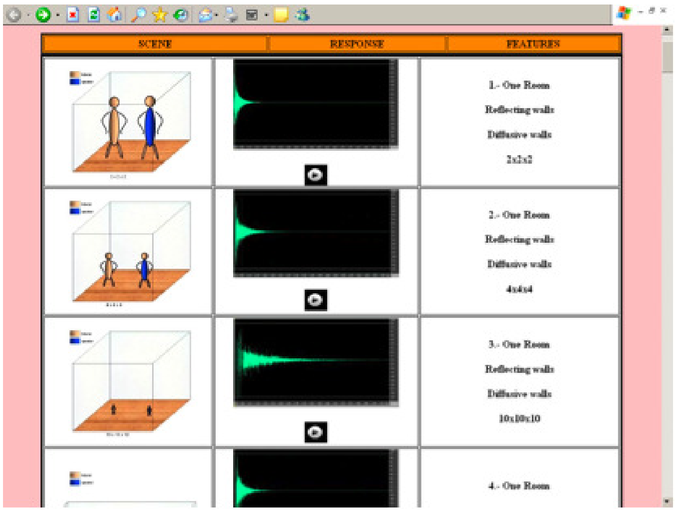
Figure 4.8: Visual representation of the Ambisonic B-format components up to third order.

B-format signals comprise a truncated spherical harmonic decomposition of the sound field. They correspond to the sound pressure W, and the three components of the pressure gradient XYZ at a point in space. Together, these approximate the sound field on a sphere around the microphone. W is the zero-order information, corresponding to a constant function on the sphere (the sphere at the top in Fig. 4.8), while XYZ are the first-order terms (the dipoles or figures-of-eight in Fig. 4.8).

The three components of the pressure gradient XYZ correspond to the front-minus-back sound pressure gradient (X), left-minus-right sound pressure gradient (Y) and up-minus-down sound pressure gradient (Z). Our ray tracing engine stores all the ray incoming directions to the listener position which means that we can not only compute the sound pressure W but also the three components of the pressure gradient XYZ thus allowing us to spatialise audio for any speakers layout [Olaiz et al. (2009)].

4.7 A Case Study: Racine project

Our ray tracing engine has been used to build a gallery of virtual environments for the IP-RACINE European Integrated Project [Racine (2009)]. The main objective of IP-RACINE is to extend the state of the art and enhance European competitiveness in the cinema industry by creating technologies to deliver enhanced cinematic entertainment that is transferable cross-platform. Within this project we have defined a set of predefined standard virtual environments and we have computed their impulse response for different absorption and diffusion values (see fig. 4.9). Usually in the post-production stage of a movie production large databases of filters are used to enhance the audio, but most of them are synthesized. This is a first step towards the introduction of more realistic filters based on the real environments recorded in the movie.



SCENE	RESPONSE	FEATURES
		1.- One Room Reflecting walls Diffusive walls 2x2x2
		2.- One Room Reflecting walls Diffusive walls 4x4x4
		3.- One Room Reflecting walls Diffusive walls 10x10x10
		4.- One Room

Figure 4.9: Gallery of virtual environments implemented for the IP-RACINE european project.

In the same IP-RACINE project we have also developed a module for our sound ray tracing engine that is able to define a path for the listener and then compute the final rendered audio containing all the acoustic changes produced by the movement of the listener through the path.

The process defined to compute this rendered audio with a moving listener is as follows:

1. *D*efinition of the path: Our engine has a module to import a path for the listener from a file. It contains all the points of the path and the velocity in which the listener moves through them (see fig. 4.10).

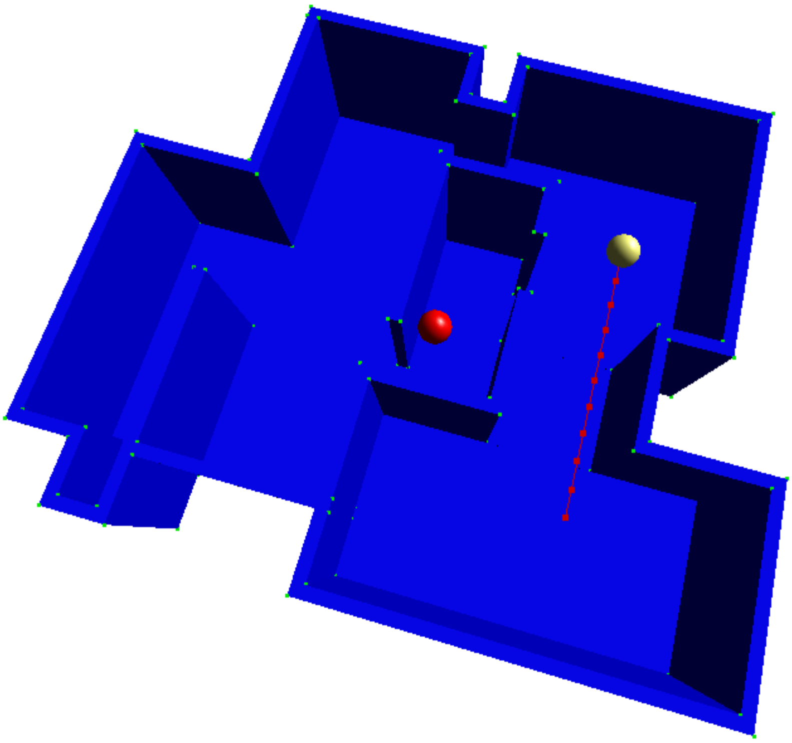


Figure 4.10: The engine is able to show the path of the listener.

2. *C*omputation of the impulse responses: Once we have the path we compute an impulse response at all the points of the path.

3. *C*onvolution and interpolation: The last step is to confer the information of the features of the environment to the input audio. The problem here is that, depending on the

time, the features that have to be considered change because the listener changes its position. In figure 4.11 we show an schema of the position of the listener L at time t in a path made of 4 positions p_i. At each position of the path we have pre-computed an impulse response IR_i. We choose the impulse response that we have to use at this time t regarding the actual position of the listener. For example, if the listener is between the two pre-computed impulse responses IR_2 and IR_3, then the impulse response used, IR_{final}, will be the result of interpolating these two impulse responses:

$$IR_{final} = lp3\, IR_3 + lp2\, IR_2 \quad . \tag{4.4}$$

At each movement of the listener the impulse response used will be different and thus the features conferred to the final audio will change.

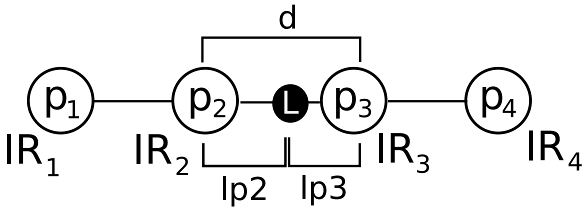


Figure 4.11: Schema of a path made of four nodes (p_1 to p_4) with a listener L moving through it.

This tool was used in the IP-RACINE to compute a rendered audio in a scene where the listener (a video camera) was moving through a scene where a song was played from a speaker. The point was that the scene was recorded with a chroma background and the speaker was put afterwards, during the post-production stage. The result was an audio that sounded as it was recorded from the camera position, and it was possible to notice the changes in the acoustic perception as the camera moved through the scene. Figure 4.12 illustrates the final post-produced scene with the superposed computation of the engine.

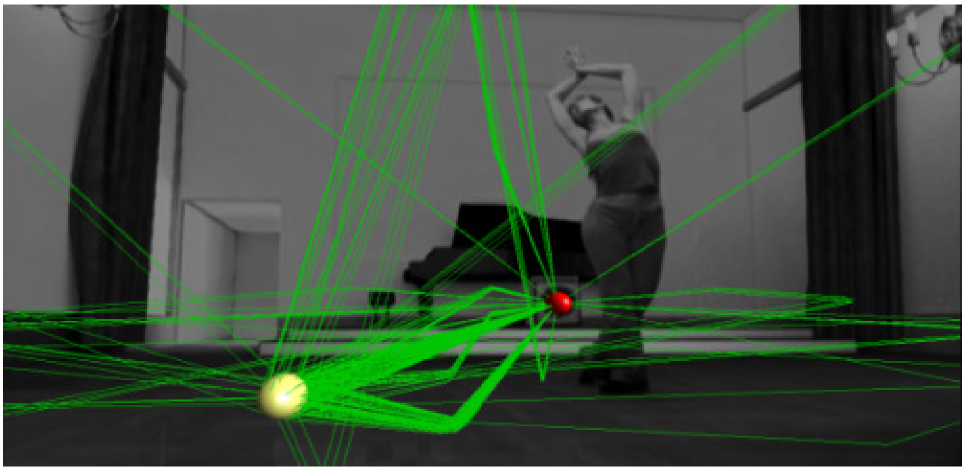


Figure 4.12: We computed the path of a camera (represented by the white sphere) recording a dancer in a chroma background, and we spatialized the audio coming from the speaker (red sphere) added in the post-production stage. We made it sound as it was recorded in the camera position.

4.8 Towards Interactive Sound Rendering

When we talk about *i*nteractive sound rendering we are referring to the process of computing in real time the rendered audio at the listener position as the listener, the sound source or both freely move through a virtual environment.

This is a hard problem if we want to compute the impulse response in real time using the ray tracing engine implemented and it is necessary to look for other methods that do not involve real time computation of the impulse response.

The power of Green's theorem, presented in section 1.5, is that one only needs to compute $g_p(x,y)$ once. Done this, one can locate any audio stream at y and hear it at x by means of the simple and easy to implement real time operation (1.10). This way, a scene is *a*coustically captured by computing g_p.

One of the consequences derived from the previous discussion is that one can perform most of the computations off-line. Imagine a 3D scene in which both the listener and the sound sources can freely move around. If a sound source moves along the set points y_i, and the listener, along the set points x_j, one can compute off-line the set of impulse responses $\{g_p(x_j, y_i), g_{\vec{v}}(x_j, y_i)\}$ for all possible combinations (x_j, y_i). Finally, once the input audio is given, the only on-line computations that remain are the simple convolutions (1.10), with the right choice of impulse response, according to the position of source and listener at each time.

By means of this method, moving sounds can be rendered real time, and a high degree of interactivity can be achieved.

Then, the method that we propose is the following:

1. *O*ff-line computation: Divide the environment in a mesh of points and compute the impulse response at each pair of points.

2. *O*n-line computation: Make a convolution of the input sound with the resulting impulse response from the interpolation of the 4 nearest points surrounding the actual listener position.

All the work done until now has been focused on the off-line computations, which is actually the part that has to be faced by the sound ray tracing engine.

In all the computations that we execute we are using very simple environments like squared rooms or two squared rooms linked with a door, as illustrated in figs. 4.13a and 4.13b, respectively.

For these kind of environments we have added an extra module to the sound ray tracing engine that subdivides them into a regular mesh of points given Δx, Δy and Δz. In fact, now we are only creating a 2D mesh covering a plane of the scene, as seen in fig. 4.14, but we can easily add the third component to mesh the whole environment.

The total number of impulse responses to be computed in 2D, n, is the result of computing the impulse response between each pair of points in the mesh without repetitions and without taking into account the order of the pair because independently of who is the sound source and who is the listener the final result is the same. Then, the number of impulse responses to be computed in 2D given a set of k points can be defined as follows:

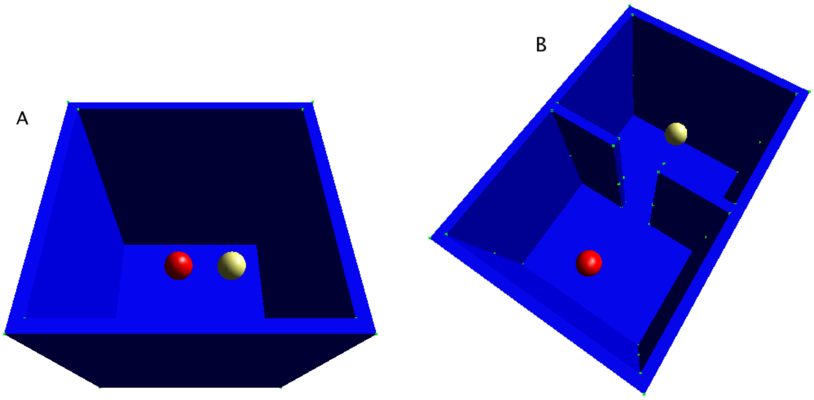


Figure 4.13: Simple squared room (A) and 2 linked squared rooms (B) used to compute impulse responses.

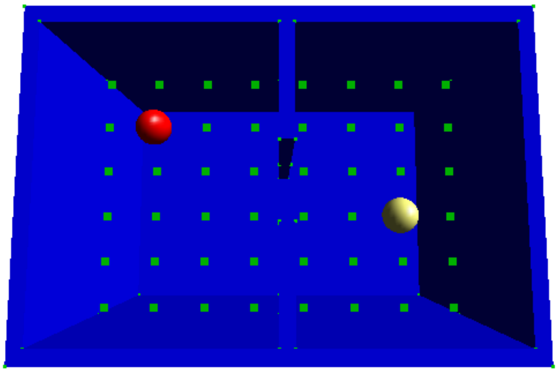


Figure 4.14: 2D mesh of points defined in the environment to compute the response function at each point.

$$n = \frac{k(k-1)}{2} \quad .$$

So, for instance, if we set a 2D mesh of 48 points like the one in fig. 4.14 then we will have to compute 1.128 impulse responses, which would mean a large computation time. The first studies determined that the mean time necessary to compute an impulse response, throwing between 15 and 20 millions of rays, is about 120 seconds. As far as we have seen in the firsts tests, when we throw more than 20 million rays the impulse response does not improve. Consequently, in the previous case the total amount of time needed would be around 135.000 seconds (more than 37 hours). In this computation the size of the listener clearly affects to the total amount of time needed, where a bigger size notably decreases the computation time but also decreases the accuracy of the results.

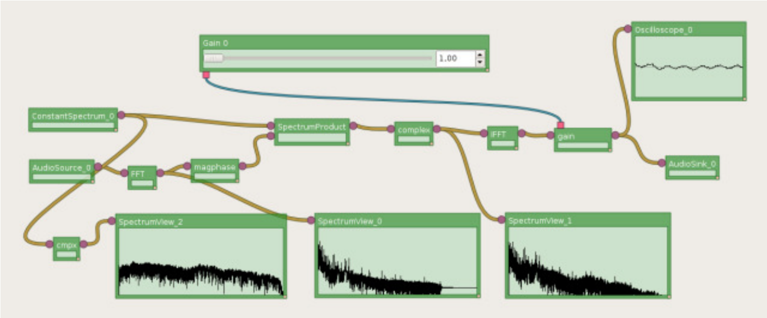


Figure 4.15: Snapshot of the clam network used to make real-time convolutions given a set of impulse responses.

All our tests to compute real-time convolutions given a mesh of impulse responses have been executed using Clam (see fig. 4.15), a C++ Library for Audio and Music [Clam (2007)], obtaining promising results [Olaiz et al. (2009)].

4.9 Conclusions

As we have seen our ray tracing engine is able to compute the acoustic field of an environment given sound source and listener positions (see fig. 4.16).

The main contribution of our sound ray tracing engine is in the information that we compute. The existing alternatives only compute the intensity of the sound field, whereas our engine computes not only the pressure but also the velocity vector of the sound field. With the addition of the velocity vector we have a complete characterization of the sound field from a physical point of view. In other words, if one only works with the intensity (or the pressure) then it is possible to confer the features of the environment to any input

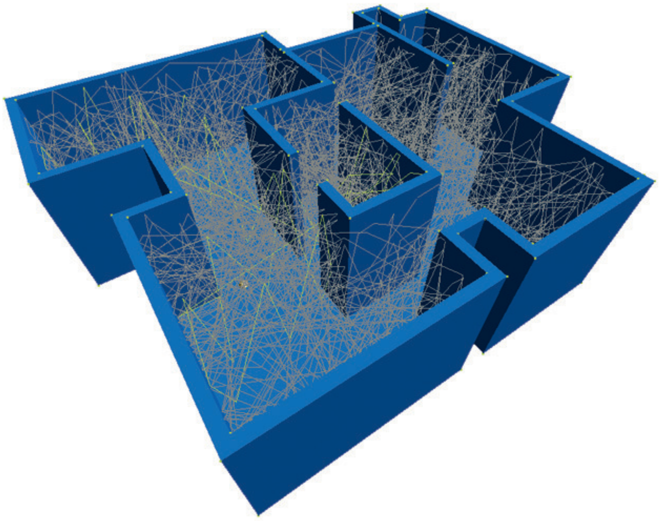


Figure 4.16: Snapshot of an imported 3D model containing all the rays traced.

sound, but without the velocity vector the direction from which the sound gets to the listener position is not known and therefore we have no information to correctly spatialize the audio stream.

Our philosophy is to always work with the complete physical model of the sound field, allowing us to be able to reproduce the whole audible experience.

At the moment we have focused our experiments in simple virtual environments, obtaining impulse responses for pressure and the three components of velocity. In fig. 4.17 we show the results for two different simulations in a simple scene (fig. 4.18). A gunshot is emitted from the source S and collected at the listener position L. Both source and listener are placed along the x axis. The pressure impulse response is plotted in black, while blue, red and green colors are used for the three components of the velocity impulse responses.

All figures on the left of fig. 4.17 correspond to a choice of very absorbing materials, whereas figures on the right column correspond to a choice of a highly reflecting materials. We can clearly see the three regions of the impulse response: direct sound,

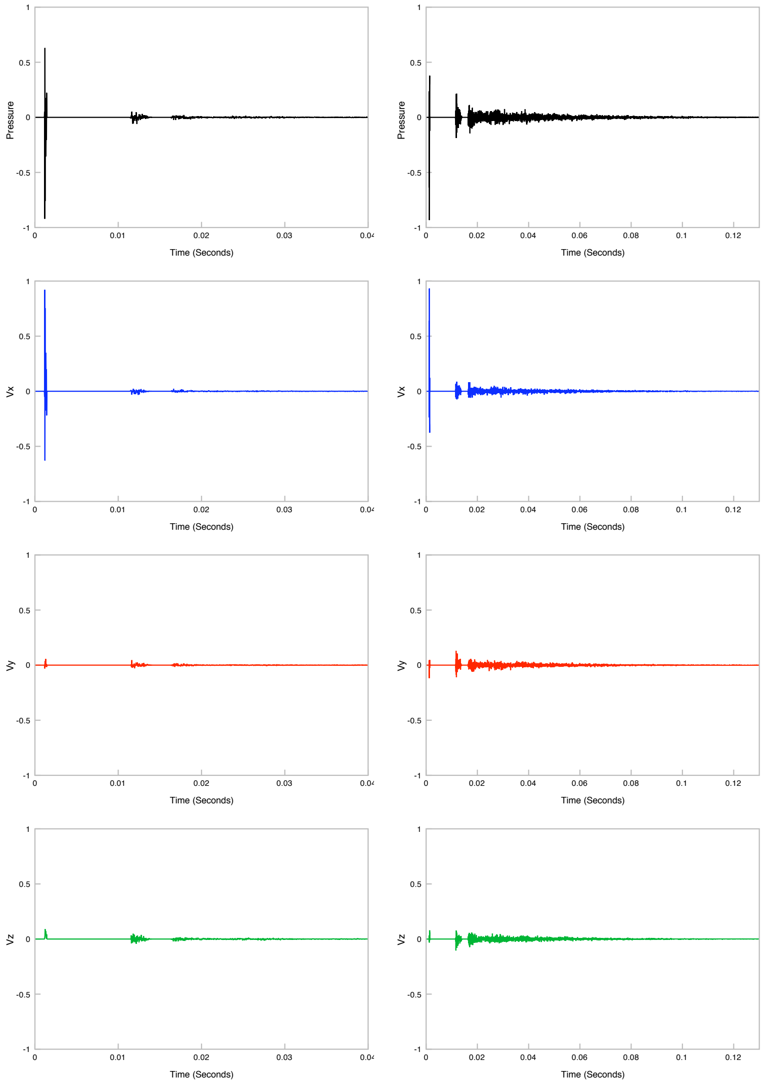


Figure 4.17: Results for two different simulations in the scene shown in figure 4.18. On the left column we plot the four impulse responses corresponding to a choice of very absorbing materials. On the right column we plot the four impulse responses corresponding to a choice of a highly reflecting materials.

early reflections and late reverberation. Both early reflections and late reverberations are notably attenuated in the absorbing room (left column) and reinforced in the reverberant one [6]. Another important aspect to highlight is the appearance of a direct sound peak in v_x as a consequence of the geometric alignment between source and listener shown in figure 4.18.

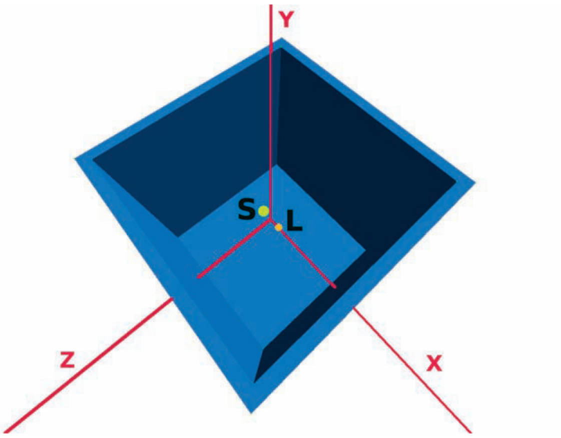


Figure 4.18: 4 x 4 x 4 m^3 virtual environment used to compute the results shown in Fig. 4.17. The distance from the sound source to the receiver is 0.5 meters.

In fig. 4.19 we show the same kind of results when we fix the absorption and change the diffusion coefficient of the walls. All figures on the right of fig. 4.19 correspond to a choice of very diffusive materials, whereas figures on the left column correspond to a choice of a highly specular materials.

In this case we can observe that when the environment is highly specular then the impulse response contains some well defined peaks representing the reflections with each of the walls. As the direction of the reflected ray is almost no affected then all rays have similar paths. Otherwise, when the environment is diffusive then the impulse response is much more uniform and there are no more differentiated peaks than the first one representing the direct sound. This is due to the almost random directions of the rays after each reflection.

[6] Note the different scale used in the x axis for the two rooms: the responses in the reverberant room lasted about three times more.

From a more technical perspective, all these results have been obtained by throwing 20 millions of rays from the sound source considering a maximum of 20 reflections. Only about 4 % of the rays hit the listener and contributed to the final impulse responses. All the simulations have been performed sampling the same range of frequencies (20 Hz-20 KHz).

The current version of our ray tracing engine has been designed in order to easily adjust the physics of the problem. Successful impulse responses have been already been produced and, after convolving them according to (1.10), we have obtained some convincing surround soundtracks.

We should mention that performance has not been optimized yet. However, this could be one of our future lines of work, adapting all optimizations coming from the computer graphics community with the objective of developing an interactive real time ray tracing for audio.

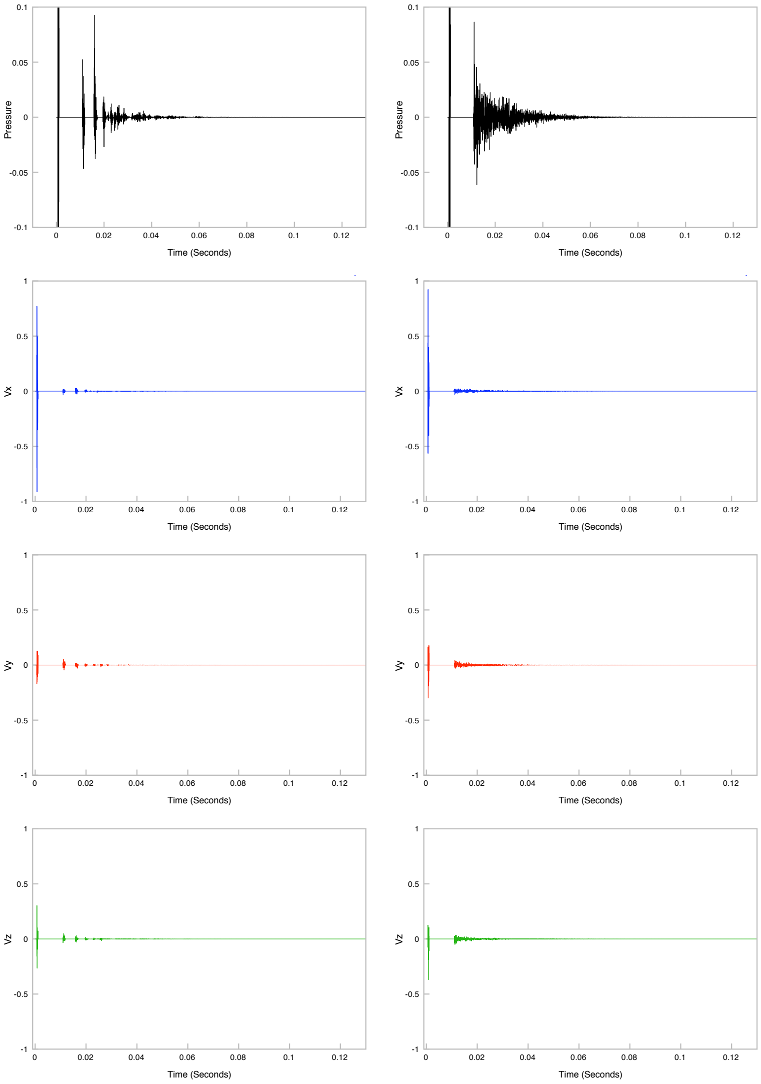


Figure 4.19: Results for two different simulations in the scene shown in figure 4.18. On the left column we plot the four impulse responses corresponding to a choice of very specular materials. On the right column we plot the four impulse responses corresponding to a choice of a highly diffusive materials.

Chapter 5

Conclusions and Future Work

5.1 Summary and Conclusions

Room acoustics is the science devoted to study the sound propagation in enclosures. Although the mathematical and physical problem is perfectly stated (see Chapter 1), the analytical solution, as a function of the space and time, of the sound propagation in closed environments is in general impossible to obtain. Therefore, the use of computers has emerged as a proper manner to get accurate solutions of the sound phenomena. These computer simulations have a considerable interest of engineers and architects, and also for the entertainment industry, as well as virtual reality applications.

In this Thesis we presented different approximations to obtain the acoustic information of an enclosure. Each computational approach gives different advantages and drawbacks that were commented in detail in Chapter 2. On one hand, wave methods give accurate solutions in the steady-state of the acoustic pressure in the frequency space. The main disadvantage of this methods is that it is needed to carry out one different simulation by each frequency and this involves a high computational cost.

On the other hand, geometrical methods treat the sound propagation as rays or beams travelling through the space. This assumpition allows to build efficient algorithms, since the computational resources required for these algorithms make possible to create real time applications so important in computer graphics. Nevertheless, one important drawback is that phenomena such as diffraction, commonly observed in room acoustics, is not obtained, and this group of simulations are usually extended in an attempt to overcome this problem, typically by incorporating diffraction filters for single propagation paths based on the geometric theory of diffraction.

Though the results are acceptable in a wider range of situations, there is still a lot of research to be done in this topic. We decided to study carefully the ray tracing method, which is a method that depends on ray-surface intersection calculations, which have computational complexity that grows sublinearly with the number of surfaces in the model. Another advantage is generality; as each ray surface intersection is found, paths of specular reflection, diffuse reflection and refraction can be sampled, thereby modeling arbitrary types of propagation, even for models with curved surfaces.

The primary disadvantages of ray tracing method stem for their discrete sampling of rays, which may lead to undersampling errors in predicted room responses. For instance, the receiver position and diffraction edges are often approximated by volumes of space (in order to enable intersections with infinitely thin rays), which can lead to false hits and paths counted multiple times. Moreover, important propagation paths may be missed by all samples. In order to minimize the likelihood of large errors, ray tracing systems often generate a large number of samples, which requires a large amount of computation.

Knowing that ray-surface intersection calculations and large number of propagation paths were key in this methodology we focused our research on the *Room Acoustic Rendering Equation* which is a model for acoustic energy propagation and the *Acoustic Bidirectional Reflectance Distribution Function* (A-BRDF) which is the term of the equation that models sound reflections (see Chapter 3).

As a result, we developed a method to compute the analytical solution for the A-BRDF in cases where sound reflections are diffuse, and diffusion is parametrized by one (or more) random variables. The method makes use of various properties of continuous probability functions, and exploits the relation between two and three dimensional probability densities. The method was applied to the Vector Based Scattering Model (VBS), which is one of the existing ways to efficiently include diffusion in a ray tracing algorithm by means of random vectors. The usefulness of the analytic results presented is more apparent when the convergence of the Monte Carlo results is considered, a large number of rays would be needed, if analytical results were not available, to fulfill each particular ray-tracing precision criteria.

Finally, this Thesis deals with the design and implementation of a ray tracing engine to compute the acoustic response in any given 3D environment (see Chapter 4). The engine we implemented computes not only the pressure but also the velocity vector of the sound field. With the addition of the velocity vector we have a complete characterization of the sound field from a physical point of view. In other words, if one only works with the intensity (or the pressure) then it is possible to confer the features of the environment to any input sound, but without the velocity vector the direction from which the sound gets to the listener position is not known and therefore we have no information to correctly spatialize the audio stream. In other words, our ray tracing engine stores all the ray incoming directions to the listener position which means that we can not only compute the sound pressure W but also the three components of the pressure gradient XYZ thus allowing us to spatialise audio for any speakers layout.

The implementation of this ray tracing engine has allowed us to test all the research done. It was not thought to be neither fast nor optimized ray tracing engine, but a research tool to proof all the cases. It makes use of the analytic solution of A-BRDFs developed within our research and is subdivided in four main modules regarding the steps that are needed to perform the computation of the impulse response: the setup, the ray generation, the ray tracing and the impulse response generation.

This Thesis reviews the use of the implemented ray tracing engine to build a gallery of virtual environments for the IP-RACINE European Integrated Project. The main objective of IP-RACINE was to extend the state of the art and enhance European competitiveness in the cinema industry by creating technologies to deliver enhanced cinematic entertainment that is transferable cross-platform.

We conclude the introduction of the implemented ray tracing engine by reviewing the process of computing in real time the rendered audio at the listener position as the listener, the sound source or both freely move through a virtual environment. We propose a method divided in two steps: an off-line computation where the environment is divided in a mesh of points and we compute the impulse response at each pair of points (source-listener positions), and an on-line real-time computation where we make the convolution of the input sound with the resulting impulse response from the interpolation of the 4 nearest points surrounding the actual listener position. This method can be used in any interactive system to spatialise audio in real-time.

5.2 Contributions

In this Thesis, the main contributions can be highlighted as follows:

- A novel methodology to compute the analytical solution for the A-BRDF in cases where sound reflections are diffuse, and diffusion is parametrized by one or more random variables (Chapter 3). The method is applied to the Vector Based Scattering Model (VBS), which is one of the existing ways to efficiently include diffusion in a ray tracing algorithm by means of random vectors (see Sec. 3.4).

- Design and implementation of a ray tracing engine to compute the acoustic response in any given 3D environment (see Chapter 4).

- The engine we implemented computes not only the pressure but also the velocity vector of the sound field thus allowing us to spatialise audio for any speakers layout (Sec. 4.6).

- Addition of the methodology to compute the analytical solution for the A-BRDF into our ray tracing engine (see Sec. 4.4). Its inclusion avoids the use of a large number of rays to fulfill each particular ray-tracing precision criteria (Sec. 3.5).

- Use of the implemented ray tracing engine to build a gallery of virtual environments for the IP-RACINE European Integrated Project (Sec. 4.7).
- A method to spatialise audio in real-time based in two steps, the first to compute all the acoustic information of the environment and the second to compute in real-time the final audio by means of convolutions of the input sound with the nearest pre-computed impulse reponses (Sec. 4.8).

Some parts of this thesis have been presented previously, these publications are listed as follows:

- J. Durany, T. Mateos and A. Garriga. "Ray tracing engine for 3D audio rendering". Submitted. 2016.
- J. Durany, T. Mateos and A. Garriga. "Analytical computation of acoustic bidirectional reflectance distribution functions". *Open Journal of Acoustics, 5, 207-217*. http://dx.doi.org/10.4236/oja.2015.54016. 2015.
- J. Durany, A. Garriga, P. Arumi and T. Mateos. "Sound spatialization using ray tracing". *Proc. of the EAA Symposium on Auralization.* Espoo, Finland, 15 – 17. 2009.
- P. Arumí', D. Garcia, T. Mateos, A. Garriga and J. Durany. "Real-time 3D audio for digital cinema". *J. Acoust. Soc. Am.*.123, 3937. 2008.
- J. Durany, A. Garriga and T. Mateos. "Towards a realistic ray tracing for room acoustics". *J. Acoust. Soc. Am.*.123, 3769. 2008.
- W. Bailer, P. Arumi, T. Mateos, A. Garriga, J. Durany and D. Garcia. "Estimating 3D camera motion for rendering audio in virtual scenes". *IET 5th European Conference on Visual Media Production (CVMP).* 19. 2008.
- J. Durany. "Geometrical Sound Rendering". Degree of Diploma of Advanced Studies, Doctorate in Computer Science and

Digital Communication. Advisors: Adan Amor Garriga Torres and Vicente Lopez Martinez. Department of Information and Communication Technologies, Universitat Pompeu Fabra. Barcelona, July 2007.

5.3 Future Research Lines

From the conclusions of this work, some new and challenging research lines could be proposed, being some of them already open. Future work may follow the lines listed here:

- To compare the computed impulse responses according to experimental measures, such as the Round-Robin room [Bork (2000); Vorlaender and Mommertz (2000)]. These simulations would give information of the accuracy of the ray tracing engine developed and presented in this Thesis.

- To work on the performance of the ray tracing engine. The current implementation was not thought to be neither fast nor optimized ray tracing engine, but a research tool to proof all the cases.

- To add support to import third-party CAD 3D formats. It would be useful to import any 3D models created in Autodesk AutoCAD, 3D Studio Max and SketchUp. We developed a plug-in to be able to export environments from 3D Studio Max together with the acoustic properties of their walls, but it would an improvement to add the import capability in our engine to work with the standard formats and be able to define the acoustic parameters of the environment using our graphic interface.

Bibliography

3dsmax (2007). http://www.autodesk.com/3dsmax.

AFMG (2016). http://ease.afmg.eu/.

Allen, J. and Berkley, D. (1979). Image method for efficiently simulating small room acoustics. *Journal of the Acoustic Society of America*, pages 912–915.

Auro3D (2015). auro3d.

Babuska, I., Banerjee, U., and Osborn, J. (2004). Generalized finite element methods - main ideas, results and perspective. *International Journal of Computational Methods*, 1:67–103.

Blauert, J. (1999). *Spatial Hearing*. Mit Press.

Blender (2007). http://www.blender.org.

Boost (2015). Boost c++ library.

Borish, J. (1984). Extension of the image model to arbitrary polyedra. *Journal of the Acoustic Society of America*.

Bork, I. (2000). A comparison of room simulation software - the 2nd round robin on room acoustical computer simulation. *Acta Acust. united Ac.*, 86:943–956.

Botteldoren, D. (1994). Acoustical finite-difference time-domain simulation in a quasi-cartesian grid. *Acoustical Society of America*, 95(5).

Botteldoren, D. (1995). Finite-difference time-domain simulation of low-frequency room acoustic problems. *Acoustical Society of America*, 98(6).

Cara (2015). http://www.cara.de.

Catt (2015). http://www.catt.se.

Cengarle, G. (2012). 3d audio technologies: applications to sound capture, post-production and listener perception. Master's thesis, Pompeu Fabra University.

Christensen, C. and Rindel, J. (2005). A new scattering method that combines roughness and diffraction effects. *J. Acoust. Soc. Am.*, 117(4):2499–2499.

Chung, T. J. (2002). *Computational Fluid Dynamics*. Cambridge University Press.

Ciskowski, R. D. and Brebbia, C. (1991). *Boundary Element Methods in Acoustics*. Springer.

Clam (2007). http://clam.iua.upf.edu/.

Dolby (2015). Atmos.

Farina, A. (1995). Pyramid tracing vs. ray tracing for the simulation of sound propagation in large rooms. *Proceedings of International Conference on Computer Acoustics and its Environmental Applications*.

Funkhouser, T. (2002). "sounds good to me!" computational sound for graphics, virtual reality, and interactive systems. *SIGGRAPH Course Notes*.

Funkhouser, T., Carlbom, I., Elko, G., Pingali, G., Sondhi, M., and West, J. (1998). A beam tracing approach to acoustic modeling for interactive virtual environments. *SIGGRAPH*, pages 21–32.

Funkhouser, T., Tsingos, N., Carlbom, I., Elko, G., Sondhi, M., and West, J. (2002). Modeling sound reflection and diffraction in architectural environments with beam tracing. *(invited paper) Forum Acusticum*.

Funkhouser, T., Tsingos, N., Carlbom, I., Elko, G., Sondhi, M., West, J., Pingali, G., Min, P., and Ngan, A. (2004). A beam tracing method for interactive architectural acoustics. *Journal of the Acoustic Society of America*, 115:739–756.

Garriga, A., Spa, C., and López, V. (2005). Computation of the complete acoustic field with finite-differences algorithms. *Invited paper in Forum Acusticum (Budapest).*

Gerzon, M. A. (1973). With-height sound reproduction. *Journal of the Audio Engineering Society*, 21:2–10.

Gerzon, M. A. (1974). Surround-sound psychoacoustics. *Wireless World*, 80:483–486.

Glassner, A. S. (1991). *An introduction to ray tracing.* Academic Press.

He, X., Torrance, K., Sillon, F., and Greenberg, D. (1991). A comprehensive physical model for light reflection. *ACM SIGGRAPH*, 25(4):175–186.

Ihlenburg, F. (1998). *Finite Element Analysis of Acoustic Scattering.* Springer-Verlag.

Immsound (2015). imm sound.

Iosono (2015). iosono.

IP-Racine (2006). *Digital Cinema Perspectives.* BCM Éditions'.

Krokstad, A., Strom, S., and Sorsdal, S. (1968). Calculating the acoustical room response by the use of a ray tracing technique. *Journal of Sound Vibration.*

Kuttruff, H. (1999). *Room Acoustics.* Spon Press.

Laine, S., Siltanen, S., Lokki, T., and Savioja, L. (2009). Accelerated beam tracing algorithm. *Applied Acoustics*, 70(1):172–181.

López, J. J., Escolano, J., and Pueo, B. (2007). On the implementation of a room acoustics modeling software using finite-difference time-domain methods. *AES 122nd Convention.*

Morse, P. and Ingard, K. (1986). *Theoretical Acoustics.* Princeton University Press.

Murphy, D., Kelloniemi, A., Mullen, J., and Shelley, S. (2007). Acoustic modeling using the digital waveguide mesh. *IEEE Signal Processing Magazine*, 24(2):55–66.

Neider, J., Davis, T., and Woo, M. (1994). *OpenGL programming guide.* Addison-Wesley.

Nironen, H. (2004). Diffuse reflections in room acoustics modelling. Master's thesis, Helsinki University of Technology.

Nosal, E.-M., Hodgson, M., and Ashdown, I. (2004a). Improved algorithms and methods for room sound-field prediction by acoustical radiosity in arbitrary polyedral rooms. *Journal of the Acoustic Society of America.*

Nosal, E.-M., Hodgson, M., and Ashdown, I. (2004b). Investigation of the validity of radiosity for sound-field prediction in cubic rooms. *Journal of the Acoustic Society of America.*

Odeon (2015). http://www.odeon.dk.

Olaiz, N., Arumí, P., Mateos, T., and Garcia, D. (2009). 3d-audio with clam and blender's game engine. *LAC 2009.*

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1992). *Numerical Recipes in C.* Cambridge University Press.

Racine (2009). http://www.ipracine.org.

Ramsete (2015). http://www.ramsete.com.

Rindel, J. (2000). The use of computer modeling in room acoustics. *Journal of Vibroengineering*, 4:219–224.

Salomons, E. (2001). *Computational atmospheric acoustics.* Kluwer Academic Publishers.

Savioja, L. (1999). Modeling techniques for virtual acoustics. Master's thesis, Helsinki University of Technology.

Savioja, L., Huopaniemi, J., Lokki, T., and Väänänen, R. (1999). Creating interactive virtual acoustic environments. *Journal of the Audio Engineering Society*.

Savioja, L., Järvinen, A., Melkas, K., and Saarinen, K. (1996a). Determination of the low frequency behaviour of an IEC listening room. In *Proc. of the Nordic Acoustical Meeting*, pages 55–58, Helsinki, Finland.

Savioja, L., Karjalainen, M., and Takala, T. (1996b). DSP formulation of a finite difference method for room acoustics simulation. In *Proc. IEEE Nordic Signal Processing Symp. (NORSIG'96)*, pages 455–458, Espoo, Finland.

Schroeder, M. (1973). Computer models for concert hall acoustics. *American Journal of Physics*, pages 461–471.

Serway, A. and Jewett, J. W. (2002). *Física para ciencias e ingenierías*. Thompson.

Seybert, A. F., Cheng, C. Y. R., and Wu, T. W. (1990). The solution of coupled interior/exterior acoustic problems using the boundary element method. *Journal of the Acoustic Society of America*, 88(3):1612–1618.

Siltanen, S., Lokki, T., Kiminki, S., and Savioja, L. (2007). The room acoustic rendering equation. *J. Acoust. Soc. Am.*, 122(3):1624–1635.

Siltanen, S., Lokki, T., and Savioja, L. (2009). Frequency domain acoustic radiance transfer for real-time auralization. *Acta Acustica united with Acustica*, 95(1):106–117.

Stroustrup, B. (2001). *El lenguaje de programación C++*. Addison-Wesley.

Vorlaender, M. and Mommertz, E. (2000). Definition and measurement of random-incidence scattering coefficients. *Appl. Acoust.*, 60:187–199.

Wright, J. (1995). An exact model of acoustic radiation in enclosed spaces. *J. Audio Eng. Soc.*, 43:813–820.