Ville Saari

Implementing a modular architecture for virtual-world Directional Audio Coding

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 27.5.2013

Thesis supervisor:

Prof. Ville Pulkki

Thesis instructor:

M.Sc. (Tech.) Tapani Pihlajamäki

Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Ville Saari		
Title: Implementing a modular architecture for virtual-world Directional Audio Coding		
Date: 27.5.2013	Language: English	Number of pages:9+56
Department of Signal Processing and Acoustics		
Professorship: Acoustics and Audio Signal Processing		Code: S-89
Supervisor: Prof. Ville Pulkki		
Instructor: M.Sc. (Tech.) Tapani Pihlajamäki		

In the past few years, the increased knowledge of the human hearing and the increase in the computing power of computers has allowed creation of new three-dimensional audio coding methods, such as Directional Audio Coding (DirAC). It allows recreation of a spatial recording performed in a single location. The processing is done by analyzing the intensity and energy of different signals in frequency domain, estimating a few parameters from them and synthesizing the sound based on these.

The directional audio coding has been further developed and different extensions to it have been created. These allow, for example, playback with headphones and projection of a video with respective three-dimensional sound. Demonstrations for these extensions exists as separate applications and a single application to combine is needed to make the demonstration use more effective. In addition to these extensions, a virtual-world version of the directional audio coding has been developed. This provides functionalities that could be implemented to a game audio engine. To further enhance the performance of the virtual-world DirAC a new architecture has been proposed. A new implementation of the system is needed to prove that the new architecture works. The architecture also allows the creation of the other extensions within it so the old demonstrations can be unified in this system.

In this thesis a fully functional version of the new architecture was created. The system runs in real-time and it uses short-time Fourier transform based approach to transform the signal to the frequency domain and back. The system is capable of panning mono sources with varying sizes and shapes and rendering three-dimensional recordings with different loudspeaker setups. In addition to these rendering through headphones was made possible. The implementation has an easy-to-use user interface that gives the user possibility to create and control different soundscapes.

Keywords: Spatial sound, virtual-world, audio engine, sound synthesis, real-time, STFT

AALTO-YLIOPISTO
SÄHKÖTEKNIIKAN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Ville Saari

Työn nimi: Modulaarisen arkkitehtuurin toteuttaminen Directional Audio Coding-menetelmälle

Päivämäärä: 27.5.2013 Kieli: Englanti Sivumäärä:9+56

Signaalinkäsittelyn ja akustiikan laitos

Professuuri: Akustiikka ja äänenkäsittelytekniikka Koodi: S-89

Valvoja: Prof. Ville Pulkki

Ohjaaja: DI Tapani Pihlajamäki

Kasvanut tieto ihmisen kuulon toiminnasta ja entistä tehokkaammat tietokoneet ovat mahdollistaneet uusien tilaäänimenetelmien kehityksen. Yksi tällainen tilaäänimenetelmä on Directional Audio Coding (DirAC), joka mahdollistaa yhdestä paikasta tehdyn tilaääninauhoituksen uudelleenluomisen erilaisilla kaiutinjärjestelmillä.

DirAC:ia on kehitetty edelleen ja siitä on olemassa muun muassa versiot kuulokekuunteluun ja videon ja siihen liittyvän kolmiulotteisen äänen projisointiin jollekin pinnalle. Eri versioista on olemassa erillisiä demoja joilla niitä voidaan esitellä, mutta yhtenäinen järjestelmä helpottaisi demojen esittämistä. Näiden lisäksi Dirac:ista on myös virtuaalimaailmojen luomiseen sopivaksi muunneltu versio, jonka toiminnalisuudet muistuttavat pelikäyttöön soveltuvaa audiomoottoria. Tälle järjestelmälle on vastikään esitelty uusi arkkitehtuuri, joka parantaa järjestelmän toimintaa muutamissa ääritapauksissa. Tästä järjestelmästä tarvitaan uusi toteutus, jolla sen ominaisuuksia pystytään esittelemään. Tämä uusi arkkitehtuuri mahdollistaa myös muiden versioiden yhdistämisen yhtenäiseen järjestelmään, mikä helpottaa demojen esittelemistä, kuten aikaisemmin mainittiin.

Tässä työssä luotiin toimiva versio esitellystä uudesta järjestelmäarkkitehtuurista. Järjestelmä on reaaliaikainen ja se käyttää Fourier-muunnokseen perustuvia menetelmiä aika- ja taajuustasojen välisiin muunnoksiin. Järjestemällä on mahdollista luoda laajoja lähteitä mono-signaaleista ja toistaa kolmiulotteisia äänitteitä erilaisillä kaiutinjärjestelmillä. Lisäksi toistoon voidaan käyttää myös kuulokkeita. Toteutuksessa luotiin helppokäyttöinen käyttöliittymä, jonka avulla käyttäjä voi luoda ja hallita erilaisia äänimaisemia, jotka koostuvat aiemmin mainituista lähdetyypeistä.

Avainsanat: tilaääni, virtuaalimaailma, audiomoottori, reaaliaikainen, synteesi

Acknowledgements

I want to thank my instructor Tapani Pihlajamäki for all the help with the programming and writing of the thesis. Additionally, thanks to professor Ville Pulkki for making this work possible and hiring me at the first place. And for the existence of the foosball table. Also thanks to the great bunch of people working at the acoustics lab, especially to all the other people at the spatial sound group that have been really helpful during the process and answered my questions whenever there have been some. Last but not least, big thanks to my family and friends for all the support I have gotten during the writing process.

Otaniemi, 27.5.2013

Ville Saari

Contents

Abstract ii
Abstract (in Finnish) iii
Acknowledgements iv
Contents v
Abbreviations vii
List of Figures viii

1 Introduction 1

2 Physics of sound 2
2.1 Sound propagation 2
2.2 Sound pressure level 3
2.3 Sound energy and intensity 4
2.4 Sound in closed spaces 5
2.4.1 Reflections and absorption 5
2.4.2 Reverberation 6

3 Hearing 7
3.1 Structure of the ear 7
3.1.1 Outer ear 7
3.1.2 Middle ear 8
3.1.3 Inner ear 8
3.2 Psychoacoustics 9
3.2.1 Directional hearing 9
3.2.2 Distance perception 12
3.2.3 Pitch perception 12
3.2.4 Precedence effect 12
3.2.5 Critical bands 13
3.2.6 Masking 14

4 Signal Processing 16
4.1 Signal representations 16
4.2 Sampling and quantization 17
4.3 Discrete Fourier transform 19
4.4 Real-time processing 20
4.5 Short-time Fourier transform 20
4.6 Overlap-add processing 21
4.7 Filtering 21

5 Object transformations 25

6 Directional Audio Coding 28
6.1 Assumptions about spatial hearing . . . 28
6.2 B-format . . . 29
6.3 Analysis . . . 30
6.3.1 Time-frequency analysis . . . 30
6.3.2 Directional analysis . . . 30
6.3.3 Diffuseness analysis . . . 31
6.4 Synthesis . . . 31
6.4.1 Virtual microphones . . . 31
6.4.2 Non-diffuse synthesis . . . 32
6.4.3 Diffuse synthesis . . . 33
6.5 Binaural processing . . . 33
6.5.1 Head tracking . . . 34

7 Virtual-world Directional Audio Coding 36
7.1 Frontends . . . 36
7.1.1 Extended panning . . . 36
7.1.2 B-format reproduction . . . 37
7.1.3 Reverb . . . 37
7.1.4 Projection . . . 39
7.2 Backend . . . 39

8 Implementation 41
8.1 Programming . . . 41
8.2 Ring buffers . . . 41
8.3 Creating the world and the objects . . . 42
8.4 Transformation to the frequency domain . . . 43
8.5 Extended panning unit . . . 44
8.6 B-format reproduction unit . . . 45
8.7 Reverberation . . . 47
8.8 Binaural implementation . . . 47
8.9 User interface . . . 48

9 Discussion 50

10 Conclusion 53

Bibliography 54

Abbreviations

DirAC	Directional Audio Coding
HRTF	Head-Related Transfer Function
DoA	Direction of Arrival
ILD	Interaural Level Difference
ITD	Interaural Time Difference
DFT	Discrete Fourier Transform
STFT	Short Time Fourier Transform
ERB	Equivalent Rectangular Bandwidth
VBAP	Vector Base Amplitude Panning

List of Figures

2.1 A sound wave meeting a surface. 5
2.2 Conceptual picture of an impulse response. 6

3.1 The anatomy of the human ear. Adopted from Wikipedia (2013a). 7
3.2 Insides of the cochlea. Adapted from Karjalainen (2009). 8
3.3 The amplitude of a traveling wave along the basilar membrane. Adapted from Karjalainen (2009). 9
3.4 Interaural level differences for sounds with different frequencies. Adopted from Moore (2008). 10
3.5 ITD curves for different elevation values. The ITDs are measured from test person. Adopted from Karjalainen (2009). 11
3.6 Measuring an ERB with two masking noises on both sides of the tonal test sound. Adapted from Karjalainen (2009). 13
3.7 Hearing threshold levels caused by narrow-band noise. The center frequencies for the noises are 250Hz, 1kHz and 4kHz and the level is 60dB. Adapted from Karjalainen (2009). 14
3.8 A masking envelope in time caused by a masking tone. The shadowed lines present the start and the end of the masking tone. Adapted from Karjalainen (2009). 15

4.1 A conceptual picture showing the effects of too low a sampling rate. On both of the colums the coordinate systems from top to bottom represent analog signal's Fourier transform, the Fourier transform of a impulse train, convolution between these two(time domain multiplication), a low-pass filter to obtain the original signal, and the result of filtering. In (a) the sampling frequency just meets the Nyquist criterion and there is no aliasing. In (b) the sampling rate is too low and the output is not like the original. Note that increasing the sampling rate in time domain moves the peaks further away from each other in frequency domain. 18
4.2 A segment of a sine wave first sampled and then quantized. 19
4.3 STFT block diagram. 20
4.4 The overlap-add processing clarified with an example. On top there is the original signal which is split to blocks. These are then processed and the results are added together with some overlap. Adopted from Wikipedia (2013b). 22
4.5 Magnitude responses for four different classes of filters. In the case of the all-pass filter there are changes in the phase response, so the filter is a way to alter the phase without touching the magnitudes. 23

5.1 Coordinate system with rotations. 26

6.1 Block diagram for DirAC processing. Adopted from Pihlajamaki et al. (2013). 28
6.2 B-format microphone directional patterns. 29
6.3 Effect of the randomized phase. Adapted from (Kendall, 1995). 34
6.4 Effect of head tracking. 35

7.1 Full virtual world architecture. Adapted from Pihlajamaki et al. (2013). . 37
7.2 Extended panning unit. Adopted from Pihlajamaki et al. (2013). 38
7.3 B-Format reproduction. Adopted from Pihlajamaki et al. (2013). 38
7.4 Reverberation unit. Adopted from Pihlajamaki et al. (2013). 38
7.5 B-Format projection unit. Adopted from Pihlajamaki et al. (2013). 39
7.6 Backend. 39

8.1 On the right, points spread to a rectangle using halton sequence with base two for x-coordinate and a halton sequence with base three for y-coordinate. On the left, points spread to a rectangle using uniformly distributed pseudorandom sequence created by Matlab's rand() function. Points with the same color present sets of 20 consecutive points. It is noteworthy that halton spreads the sets of consecutive points evenly to the whole square. 42
8.2 Multi-resolution processing. 44
8.3 Transforming the object from the model space to the the listener space. . . 44
8.4 Extended panning processing for the non-diffuse stream explained. All the numerical values do not correspond to the actual system. The initial situation where the listener is listening to 5.0 loudspeaker setup is in (a). There is a sound file that the user wants to be panned to the direction 30°. In (b) the sound file is depicted in time domain and then transformed to the frequency domain with a four point DFT that divides the frequencies to four bands A, B, C, and D. In (c) there are the distributed angles and the loudspeaker gains that would be in the real system computed by VBAP. In (d) the final loudspeaker signals are created. 46
8.5 User interface. 49

Chapter 1

Introduction

Sound reproduction from recordings has been around for a long time. First, there were the single channel mono recordings. The next step after these was to add a second channel and create stereo recordings that could be played back with two loudspeakers. With two channels it is possible to create different perceived locations for the sound between the two loudspeakers used. After this multichannel recordings and multichannel playback were the natural next step. One example of multichannel playback methods is Ambisonics. Multichannel playback requires that the listener has as many loudspeakers in his listening space as there are channels. Nowadays, it is more and more common to have multichannel loudspeaker setups in living rooms and multichannel playback and recording are thus more important. A new technology to perform multichannel rendering has been created by Pulkki (2007). In the past years this technology has been further developed and a version suitable for virtual-world creation has been introduced.

In order to prove that the technique really works and it can be used in desired situations, demonstrations are needed. Some demonstrations already exist, but there is a need for an uniform system that takes use of all the capabilities and implements the new architecture(Pihlajamaki et al., 2013) that has been introduced lately. The aim for the thesis is to create a system that has all of the different versions of DirAC in it and allows an easy creation of demonstrations. The easy creation of new demonstrations will allow new ideas to be implemented with a little effort. The demonstrations are important for making the work made at the research group known.

The outline of this thesis is as follows. First, the physics of sound relevant to the techniques explained later are discussed. After this, the physiology of the human ear and how the human hearing works are explained. Next, some relevant signal processing topics are explained to provide understanding for the rest of the thesis. Following these preliminary chapters, there is a detailed explanation about the Directional Audio Coding and its virtual world version. Next, the implementation and the choices made along the way are explained. The thesis ends with discussion about the system, further development ideas, and a conclusion.

Chapter 2

Physics of sound

In this chapter, the basic physics behind the sound phenomenon are explained. Focus is on topics that are relevant to the Directional Audio Coding which is introduced later. First, the concept of sound pressure is derived from the movement of air molecules. After this, sound waves and their propagation, energy, and intensity are discussed. The sound pressure level is introduced in order to make talking about the sound pressures easier. Finally, reflections, absorption, and the concept of reverberation are introduced. All the derivations for equations, which are presented in chapter, are based on the book of Fahy (2001).

2.1 Sound propagation

The air consists of molecules which are in constant motion. The speed of the motion depends on the temperature and the direction of the motion is random, i.e., each direction has an equal probability. A molecule has a momentum which is defined in the classical mechanics as

$$\vec{p} = m\vec{v}, \tag{2.1}$$

where m is the mass of the molecule and $\vec{v}$ is the velocity. When moving, the molecules collide with each other, and in an event of collision, their directions of movement change. For practical reasons, the air can be thought as a continuous medium. This assumption is generally acceptable and causes no significant errors in engineering applications (Fahy, 2001). Instead of discrete molecules, there is a voidless medium in which properties at a point are defined as an average of the properties of the molecules in a certain area around the point. One of these properties is the pressure which is a force applied to a certain area

$$P = \frac{|\vec{F}|}{A}. \tag{2.2}$$

According to Newton's second law the force is equal to the rate of change of its momentum, that is,

$$\vec{F} = \frac{d\vec{p}}{dt}, \tag{2.3}$$

where t is the time. In air, the pressure p is the mean rate of change of momentum in a unit area. Pressure is a scalar quantity, which means that it has no direction.

Sound is defined as changes in the sound pressure. Based on equations 2.2 and 2.3, the rate of change of the momentum has to change to create sound. To do this, an external force, that makes the molecules to move faster to a certain direction, is required. The molecules affected by the force will collide with other molecules and set a chain reaction. A change in the sound pressure is transmitted through the air in this way as a wave.

A wave in three dimensions is a surface in which physical quantities are linked to its temporal and spatial location. In other words, it is possible to derive the properties of the wave knowing the time instant and the location.

A plane wave is a simple wave that has the same properties at any point on a surface normal to the direction of propagation. Equation for the pressure of a plane wave is

$$\frac{\delta^2 P}{\delta x^2} = \frac{\rho_0}{\gamma P_0}\frac{\delta^2 P}{\delta t^2}. \tag{2.4}$$

Here p is sound pressure, x is the observation point, ρ_0 is the mean density of air, P_0 is pressure in equilibrium and γ is the ratio of specific heats in a constant volume and a constant pressure. It has a value of 1.4 for air. The plane wave equation has a general solution

$$P(x,t) = f\left[\left(\frac{\gamma P_0}{\rho_0}\right)^{\frac{1}{2}} t - x\right] + g\left[\left(\frac{\gamma P_0}{\rho_0}\right)^{\frac{1}{2}} t + x\right], \tag{2.5}$$

where $f[\cdot]$ and $g[\cdot]$ are functions that depend on the dynamic conditions imposed upon the fluid at its boundaries, x is the observer location and t the time instant. Whatever the function f is, it is constant if the observation point x and the time t are related by $[(\frac{\gamma P_0}{\rho_0})^{\frac{1}{2}} t - x] =$ constant. This implies that the sound pressure is not changing if the observer is traveling in the x-direction at speed $(\frac{\gamma P_0}{\rho_0})^{\frac{1}{2}}$ (Fahy, 2001). This suggests that the speed of sound is

$$c = \sqrt{\left(\frac{\gamma P_0}{\rho_0}\right)} = \sqrt{\gamma R T_0}. \tag{2.6}$$

Because R and γ are constants for the air, the speed of sound in the air depends only on the temperature T. g from the equation 2.5 presents a wave traveling to the negative-x direction. The speed of sound in the room temperature (20°C) is $343.2\frac{\mathrm{m}}{\mathrm{s}}$.

2.2 Sound pressure level

The quietest still audible sounds have a sound pressure of 20 μPa when in comparison the loudest sounds that do not cause damage to the ear have a sound pressure of 20 Pa. Because the range of values is large, a logarithmic decibel scale is better for representing the pressure. The sound pressure presented in decibels is called the sound pressure level (SPL)

$$SPL = 20\log_{10}\left(\frac{P}{P_0}\right), \tag{2.7}$$

where P is the sound pressure in Pascals and P_0 is reference sound pressure 20 μPa (Rossing et al., 2002). With this equation, the decibel values for the previously mentioned sound pressures are 0dB and 120dB.

2.3 Sound energy and intensity

Sound possesses both kinetic and potential energy. Kinetic energy is proportional to the square of the magnitude of the particle velocity and the potential energy is proportional to the square of the sound pressure. The sum of the potential and the kinetic energy stays the same as the sound propagates. The kinetic energy per unit volume is

$$T = \frac{1}{2}\rho_0 u^2, \tag{2.8}$$

where u is the speed of the fluid particle motion. The potential energy per unit volume is

$$U = \frac{1}{2}\frac{P^2}{\rho c^2} = \frac{\rho_0 P^2}{2Z_0}. \tag{2.9}$$

The total mechanical energy per unit volume is the sum of these two

$$e = T + U = \frac{1}{2}\rho_0 u^2 + \frac{\rho_0}{2Z_0}P^2. \tag{2.10}$$

According to the definition of the mechanical work, the rate of work done on one side of an imaginary surface by the fluid on the other side is given by the scalar product of the force vector acting on that surface and the particle velocity vector through the surface

$$\frac{dW}{dt} = \vec{F}\vec{u} = P\delta S\vec{n}\vec{u}, \tag{2.11}$$

where δS is the elemental area of the surface and $\vec{n}$ is an unit vector normal to the surface. Work rate per unit area can be written as

$$\frac{\frac{dW}{dt}}{\delta S} = P\vec{u}_n, \tag{2.12}$$

where $\vec{u}_n$ is the component of the particle velocity normal to the surface. The quantity in previous equation is defined as the instantaneous sound intensity. The equation for the intensity is

$$\vec{I}(t) = P\vec{u}_n. \tag{2.13}$$

2.4 Sound in closed spaces

2.4.1 Reflections and absorption

When a wave is traveling in a medium other than vacuum, its energy is transforming to other forms, mainly to heat. This loss of energy is called absorption. The sound pressure of a wave traveling in the air follows the inverse distance law

$$P \propto \frac{1}{r}, \tag{2.14}$$

where r is the distance from the sound source.

When a wave meets a surface at which the acoustic properties of the medium change, part of the wave is transmitted through the surface and the rest is reflected back. This is depicted in Fig. 2.1. How much of the energy is reflected and how much is transmitted is dependent of the transmission properties of the two media. Transmission properties of a medium can be described with the characteristic acoustical impedance

$$Z_0 = \rho_0 c_0, \tag{2.15}$$

where ρ_0 is the density of the medium and c_0 is the speed of sound. For the air in a room temperature the characteristic acoustic impedance has the value of 413.3 $\frac{\mathrm{Ns}}{\mathrm{m}^3}$. With the help of this, the intensities for the reflected and transmitted waves are defined as

$$\frac{I_{\text{transmitted}}}{I_{\text{incident}}} = \frac{4Z_1Z_2}{(Z_1+Z_2)^2}, \tag{2.16}$$

and

$$\frac{I_{\text{reflected}}}{I_{\text{incident}}} = \frac{(Z_1-Z_2)^2}{(Z_1+Z_2)^2}, \tag{2.17}$$

where Z_1 and Z_2 are the characteristic acoustical impedances for the first and second medium respectively.

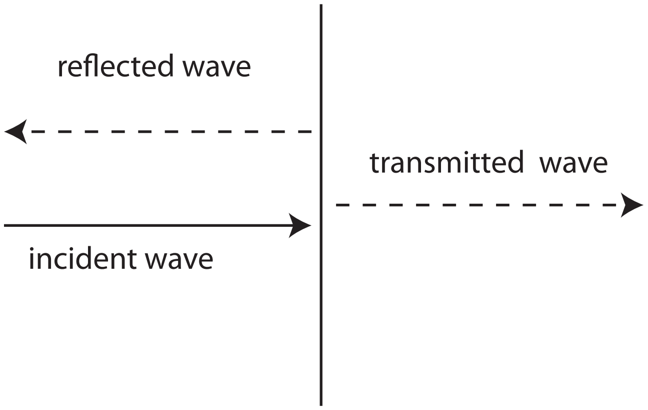


Figure 2.1: A sound wave meeting a surface.

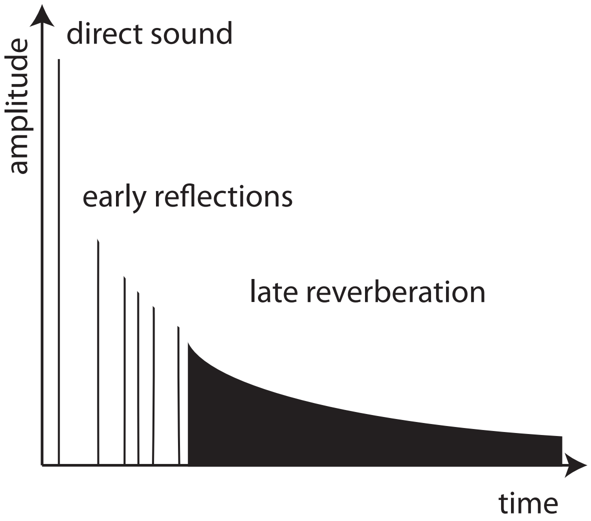


Figure 2.2: Conceptual picture of an impulse response.

2.4.2 Reverberation

Reverberation is the persistence of sound in a room or other closed space. In addition to the direct sound coming from the source, reflected copies of the sound, i.e. echoes, will reach the listener in a room. In figure 2.2, there is a made-up impulse response that shows an ordinary structure of an impulse response. Impulse response describes the sound that reaches the listener, when a short impulse-like sound is played in the hall. After the direct sound, a multitude of early reflections reach the listener, and after them a smooth late reverberation reaches the listener. In the late reverberation it is not possible to hear the individual echoes separately anymore. The reverberation builds up as the echoes and their echoes reach the listening position and it is decayed by the absorption from the air and the walls. Reverberation time is defined as the time that it takes for the SPL to drop below 60 dB of the level of the original sound. The reverberation time depends mainly on the material of the walls, the ceiling, and the floor of the space and the size of the space.

Chapter 3

Hearing

The human hearing system consists of two ears and the brain. The parts of a human ear can be roughly separated to the outer ear, the middle ear, and the inner ear. The ear extracts information about the sound that reaches the ear and transmits it to the brain which then interpret the message. In the following sections a closer look to the insides of the ear is taken and different topics of the human perception of the sound are explained. The knowledge about the perception of the sound provides the fundamental knowledge for the Directional Audio Coding which is discussed later.

3.1 Structure of the ear

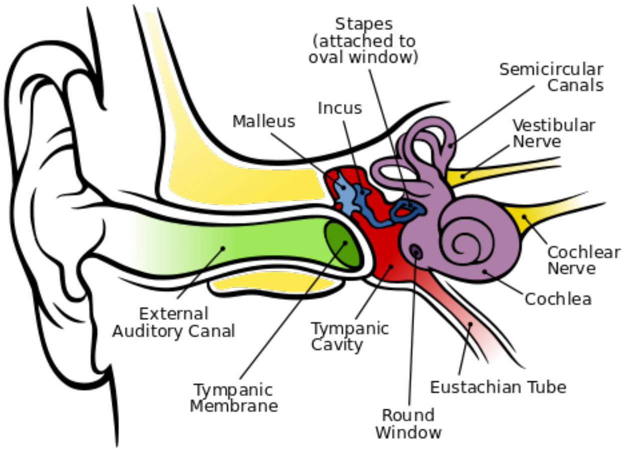


Figure 3.1: The anatomy of the human ear. Adopted from Wikipedia (2013a).

3.1.1 Outer ear

The structure of the human is depicted in Fig. 3.1. The outer ear consists of the pinna and the ear canal and it ends at the eardrum. In the hearing process the pinna has an effect on the directional hearing. In addition to the direct sound, delayed copies reflected from the pinna reach the eardrum. Summation of the sound and the delayed copies causes comb filtering, which attenuates some frequencies and emphasizes others. Depending on the direction of arrival of the sound, the attenuation is different.

3.1.2 Middle ear

The middle ear is the part of the ear that starts at the eardrum and ends at the oval window of the cochlea. When the sound reaches the eardrum it makes it vibrate. The eardrum is connected to the malleus, the first of the three ossicles located in the middle ear. This is the point where the vibrations in the air are transformed into mechanical form. The malleus is connected to the incus which is connected to the stapes. The chain of these three ossicles transforms the vibration to the oval window and to the inner ear. This chain reduces the magnitude of the vibrations significantly and this way performs impedance matching between the middle and the inner ear.

3.1.3 Inner ear

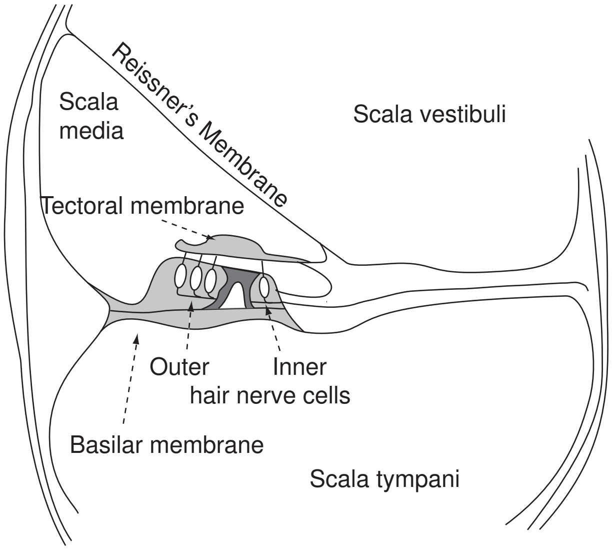


Figure 3.2: Insides of the cochlea. Adapted from Karjalainen (2009).

Whereas the middle ear is filled with air, the inner ear has liquid inside it. The part of the inner ear that is relevant to the hearing is called cochlea. In addition, there are organs that have to do with the sense of balance, but they have no effect on the hearing so they are not discussed here. The shape of the cochlea is a spiraling tube. The liquid inside the cochlea is split up into three parts by the basilar membrane and the Reissner's membrane. This is depicted in Fig 3.2. The liquid inside each part has different chemical properties. The ossicles send vibrations to the oval window, which causes a pressure difference between the two sides of the basilar membrane. The pressure difference travels as a wave along the basilar membrane. The amplitude of the wave grows first and then decays when traveling along the membrane. This is depicted in Fig. 3.3. A third membrane, the tectoral membrane, is located above the basilar membrane and between these two membranes there are the hair cells. The hair cells are divided to two groups by an organ called the tunnel of Corti. These two groups are called the inner hair cells and the outer hair cells and their functions are different. The inner hair cells transmit the most of the information to the brain and the outer hair cells have an effect on the response of the basilar membrane on sound. The wave traveling along the basilar membrane causes the membrane, and thus the hair cells to move. The hair cells touching the tectoral membrane are activated and send information about the sound to the upper levels of the hearing system.

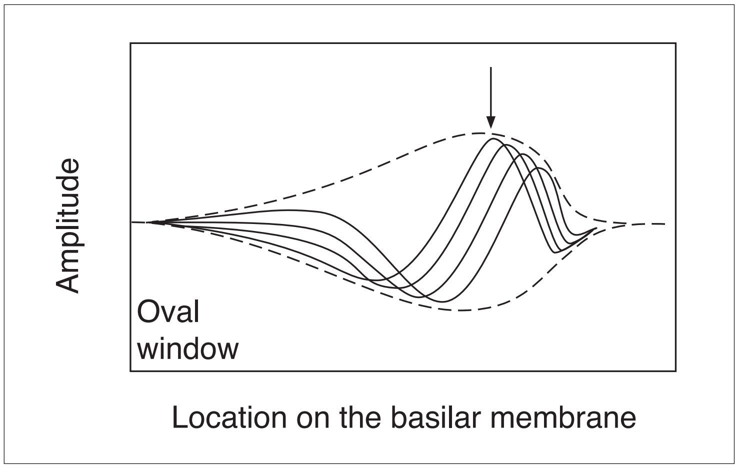


Figure 3.3: The amplitude of a traveling wave along the basilar membrane. Adapted from Karjalainen (2009).

3.2 Psychoacoustics

In this chapter, the perception of sound is discussed. The human ear picks up different cues from the sound that reaches it and sends this information to the brain which creates the perception of the sound. The localization of sounds, the perception of distance, and the pitch of sound are explained.

3.2.1 Directional hearing

The human hearing detects the direction of arrival of a sound event based on three different criteria (Grantham, 1995):

1. Interaural time difference (ITD)
2. Interaural level difference (ILD)
3. Monaural cues

Inspecting sinusoidal tones gives a basic understanding of how the location of a sound source is estimated from the cues mentioned. When a tone is played from a location that is not straight in front or straight behind the user, the sound will arrive first into the nearer ear and the sound will be more intense in this leading ear. These are the first two cues mentioned. In addition to these, the pinna, head and the torso cause reflections of the sound arrive to the ear later than the original sound. When a delayed sound arrives to the ear, it interferes with the direct sound, and depending on the phase difference, it will attenuate the direct signal or make it stronger. These reflections are dependent of the direction of arrival, and thus provide the human hearing system directional information.

The ILD and ITD cues are not equally useful for all frequencies. In the case of ILDs the wavelength, and thus the frequency of the sound is important for the cues. A tone with low frequency has a wavelength close to the diameter of the head or bigger. In this case the wave bends well around the head and there is very little attenuation whereas with high frequencies the wavelength is short and there is notable attenuation (Moore, 2008). The effect of the change in frequency is demonstrated in Fig. 3.4.

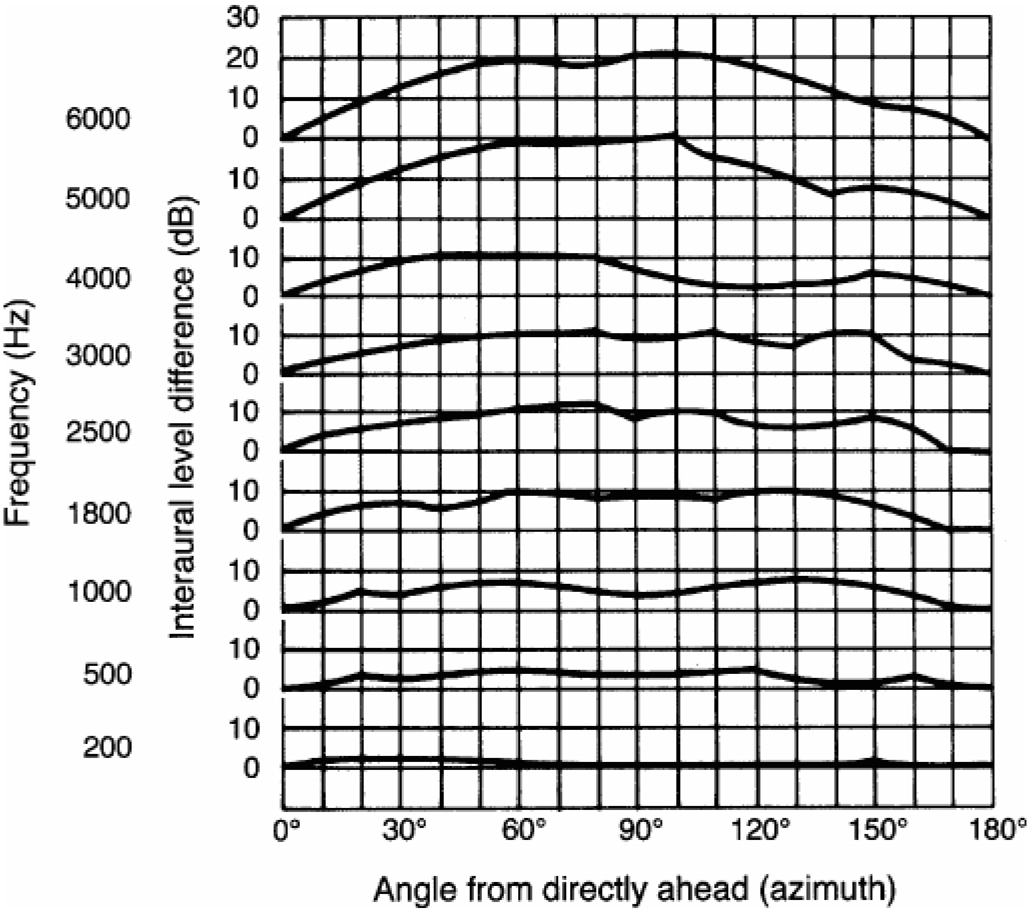


Figure 3.4: Interaural level differences for sounds with different frequencies. Adopted from Moore (2008).

The values of interaural time differences range from 0 to approximately 650 μs when the azimuth angle is changed from 0° to 90° (Moore, 2008). The ITD is roughly independent of the frequency of the sound but the usefulness of the ITD as a cue varies with the frequency. With the sinusoidal tones, the ITDs can be thought as phase differences between the two ears. Low frequency tones have a long period and the phase shift caused by the ITD is noticeably smaller than the period of the tone. The hearing is able to transform this cue to information about the direction of arrival. High frequency tones have a short period and the phase shift can be many times the period. In that kind of case the hearing is not able to tell the amount of cycles the sound has been delayed and the information becomes ambiguous.

Rayleigh (1907) created the so called duplex theory about sound source localization. It suggests that the ILDs are the main cue used in the high frequencies and the ITDs are the main cue used at the low frequencies. This seems to be true with sinusoidal sounds but with more complex sounds, the theory is not strictly true. The complex sounds in nature have onsets and offsets and they may change their spectral shape and intensity as a function of time. ITDs of these changes provide information for the localization and they replace the phase ambiguity cues provided for the steady sinusoids. In addition to these, for sounds that have a wide enough range of frequencies, the ITDs across all frequencies can be compared and if a common ITD through all the frequencies is found, it is the 'true' ITD (Grantham, 1995).

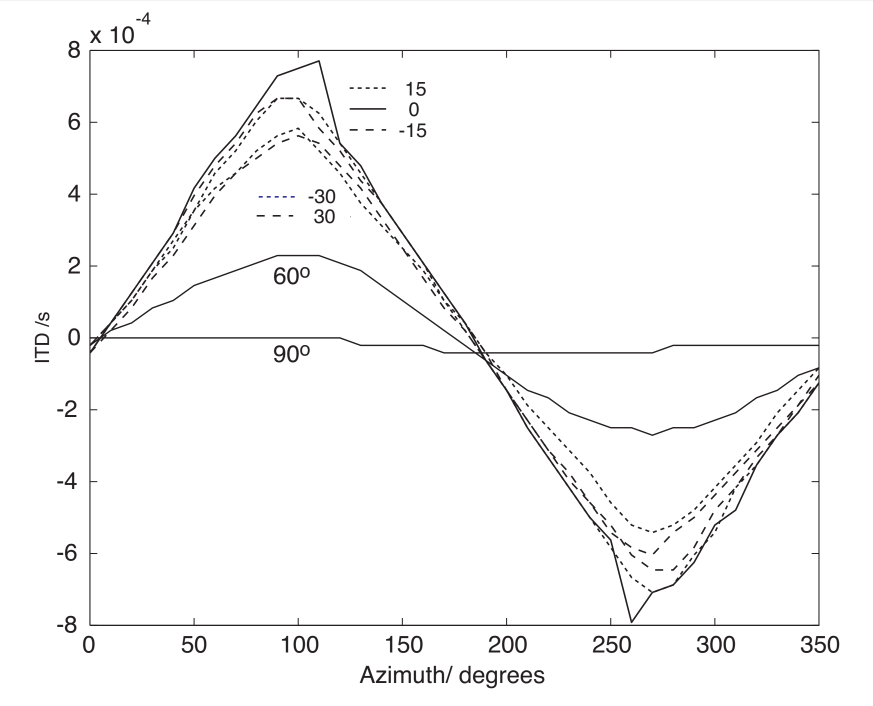


Figure 3.5: ITD curves for different elevation values. The ITDs are measured from test person. Adopted from Karjalainen (2009).

In the vertical plane the information about the angle comes mostly from the monaural cues provided by the pinnae. Because of the size of the pinnae, they reflect only high frequencies and when the high frequency content is removed from a signal, the cue gets weaker. In addition to these monaural cues, the human hearing is capable of interpreting the ILD and ITD changes caused by the different elevation value. Fig. 3.5 demonstrates the ITD measured with different elevation values. A minor cue that is used by the hearing system is the so called disparity cue which arises from the differences between two pinnae.

This provides information at frequencies from 10kHz to 12kHz (Moore, 2008).

3.2.2 Distance perception

For the distance perception, three main cues have been identified by Grantham (1995):

1. Sound pressure level
2. Direct-to-reverberant sound energy ratio
3. Amount of high-frequency content in the signal

The two first ones can be explained by the properties of the sound waves propagating in the air. The further away the source is, the weaker the signal is and this information can be used as a cue if the listener is familiar with the sound and has the knowledge about its loudness. The reverberant sound does not attenuate the same way as it does not arrive from the source but arrives as reflections from the walls, the ceiling, and the floor. This enables the listener to make conclusions about the ratio of the direct and reverberant energy. As the sound propagates in the air, high frequencies attenuate faster than the low frequencies. Thus, the further away the source is, the more high frequencies are missing from the sound.

3.2.3 Pitch perception

Pitch is an attribute of an sound in terms of which different sounds can be classified on a scale from low to high. It is associated with the frequency of a pure tone and the fundamental frequency of a complex sound. There are two traditional theories for the pitch perception:

1. The place theory
2. The temporal theory

The place theory suggests that the pitch is related to the excitation pattern of the basilar membrane, more specifically its maximum. In the time theory, the pitch is related to the time pattern of the neural impulses that are caused by the hair cells as described in Chap. 3.1.3. The intervals between the neural impulses approximate the integer multiples of the period of the waveform. The current view of the pitch perception is that the pitch is defined as a combination of these two theories. Depending on the frequency of the sound, the one of these theories is dominating on the pitch perception.

3.2.4 Precedence effect

In many listening situations the sound arrives to listeners ears via multiple paths. Some of the arriving sound is direct but much of the sound arrives through one or multiple reflections. When two separate sounds arrive to the listeners ears quickly one after other, they are heard as one sound. Depending on the type of the sounds the approximate interval at which the sounds are still heard as one event varies from 5ms to 50ms (Moore, 2008).

When separate sounds are heard as one, the perceived direction is the one of the first sound. This is phenomenon is called the precedence effect (Wallach et al., 1949).

3.2.5 Critical bands

The human hearing processes sounds in frequency bands. This means that if a sound has a certain frequency, it is processed together with other sounds with a close frequency. A frequency region at which sounds are processed together is called a critical band. When multiple sounds occur simultaneously in a critical band, only one sound event is perceived. The perceived loudness of two sounds on the same critical band is the same as the loudness of a single sound. The critical bands are not fixed but they are formed around the frequency of the perceived sound. This is explained by the structure and functionality of the basilar membrane explained in Chap. 3.1.3. The width of a critical band depends on its center frequency (Yost, 1994).

One way to define the width of a critical band is to play narrow-band noise on the center-frequency of the band and a widewr masking noise centered at the same frequency. When the bandwidth of the masking noise is increased there is no change in the perceived loudness until some certain width. This is the width of the critical band. Zwicker and Terhardt (1980) have presented an equation for a critical band measured this way

$$\Delta f'_{CB}025 + 75[1 + 1.4(1000 f_C)^2]^{0.69}. \tag{3.1}$$

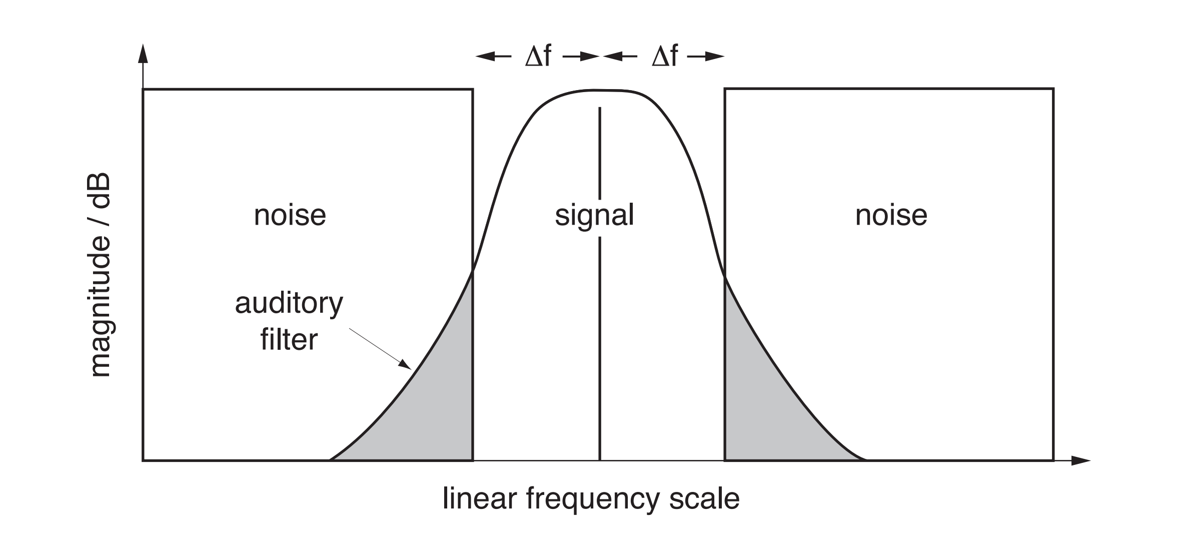


Figure 3.6: Measuring an ERB with two masking noises on both sides of the tonal test sound. Adapted from Karjalainen (2009).

Another way to measure the frequency resolution of the hearing is the use of Equivalent Rectangular Bandwidths (ERB). A tonal sound is played on the center frequency and it is masked by a noise mask from both sides as in Fig. 3.6. The width of the noiseless passband is increased until it is possible to hear the tonal sound on the center frequency. An equation for an ERB has been derived from listening tests (Glasberg and Moore, 1990)

$$ERB(f) = 24.7 + 0.108 f_{\mathrm{c}}, \tag{3.2}$$

where f is the center frequency and the bandwidth is expressed in Hz.

3.2.6 Masking

It is very seldom that a sound occurs in isolation. Usually, there are multiple sounds present in a listening situation and the sounds interfere with each other. This is called masking and usually one sound is referred as the signal and another as the masker. The masking happens in both the time domain and the frequency domain. In the frequency domain, masking happens inside critical bands, which were introduced in the previous section. Masking by narrow-band noise signals is depicted in Fig. 3.7. The masker causes masking on a certain range around its frequency and the effect is strongest in the middle of the range. In time domain, a masking signal causes a masking envelope that starts before the masking noise and lasts a while after it. This temporal masking is clarified in Fig. 3.8.

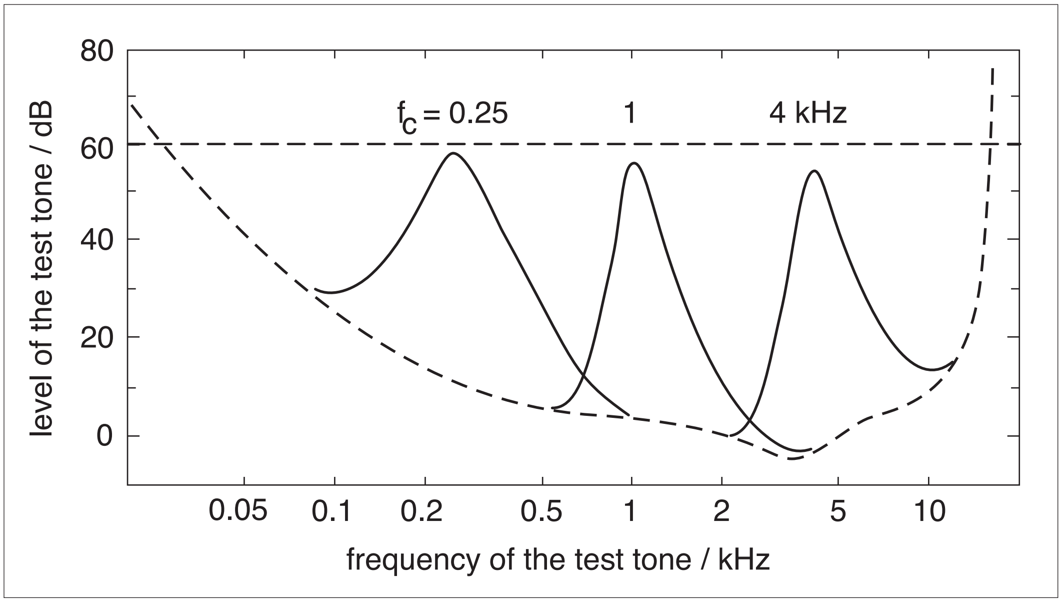


Figure 3.7: Hearing threshold levels caused by narrow-band noise. The center frequencies for the noises are 250Hz, 1kHz and 4kHz and the level is 60dB. Adapted from Karjalainen (2009).

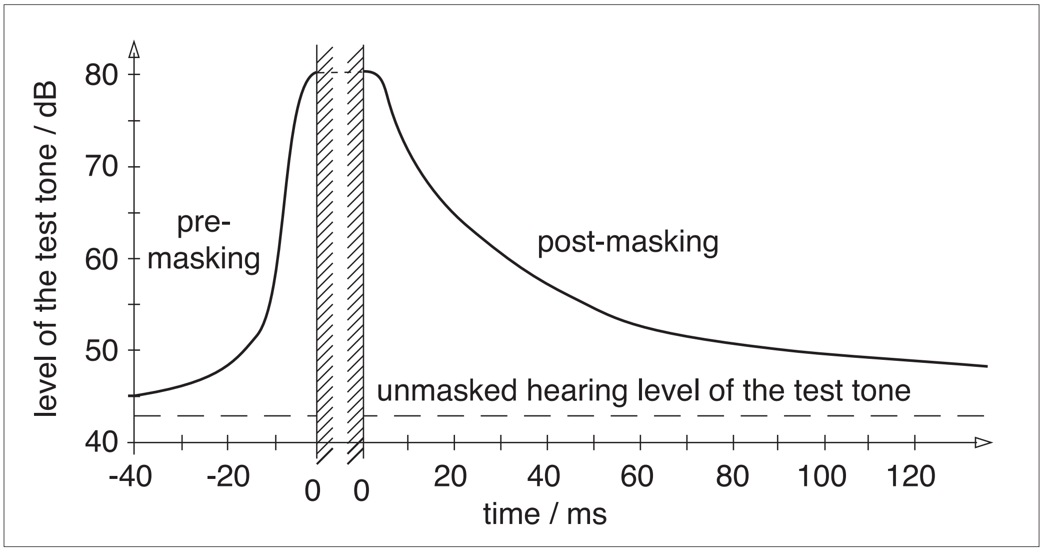


Figure 3.8: A masking envelope in time caused by a masking tone. The shadowed lines present the start and the end of the masking tone. Adapted from Karjalainen (2009).

Chapter 4

Signal Processing

A broad definition for a signal is that it is a function that represents information about some phenomena. In this thesis, signal is a function that represents information about a sound wave transformed to electrical form. In this chapter, processing that is required to transfer the signal between different domains and techniques required later are discussed. First, the two different domains in which the signal can be represented and the transformation between them are explained. After this, transforming the signal to digital representation and the discrete versions of the previous transformations are discussed. As the implemented system is running in real-time, the aspects of a real time system are discussed and the previous transformations are formulated to be suitable for real-time use. Finally, the concept of filtering and different filters are explained.

4.1 Signal representations

In the field of audio, there are two domains in which the signals are often presented. In the time domain, the attributes of the signal is known at different time instants. In frequency domain, the attributes of the signal is known at different frequencies. In this thesis, when a signal is in time domain, it is presented with a lowercase letter, and when it is in the frequency, domain it is presented with a capital letter.

A time domain signal $x(t)$ contains information about the amplitude of a signal at every time instant t. For example, a simple sine wave is represented with equation

$$x(t) = \sin(2\pi f t), \tag{4.1}$$

where f is the frequency of the wave and t is the time. The signal can be transformed to the frequency domain with the Fourier transformation which has the equation for a deterministic signal $x(t)$

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt, \tag{4.2}$$

where ω is the frequency in radians. According to Mulgrew et al. (1999) the Fourier transformation has a physical interpretation:"The complex Fourier transform is used to represent a signal as a 'sum' of cosine waves at all possible frequencies. All possible frequencies are required as the signal is not periodic and hence cannot have harmonics. The magnitude of the complex Fourier transform $|X(\omega)|d\omega(2\pi)$ is the amplitude of a sine wave with frequency ω rad/s. The angle of the complex Fourier transform $\angle X(\omega)$ is the phase shift associated with the cosine wave. The signal x(t) is said to have a component in a small frequency band ω to $\omega + d\omega$ rad/s which might be written approximately as:

$$\frac{|X(\omega)|d\omega}{2\pi}\cos(\omega t + \angle X(\omega))." \tag{4.3}$$

To perform the transformation from the frequency domain to the time domain the inverse Fourier transform,

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{(j\omega t)}dt, \tag{4.4}$$

is used.

4.2 Sampling and quantization

If the sound is going to be processed with digital signal processors, it has to be transformed to digital form. To do this, the sound wave traveling in the air is recorded with a microphone and stored as an electrical signal. The theory and practice of the microphones is discussed in the book by Eargle (2012). The continuous signal has the information about the sound in its amplitude. The signal is converted to digital form in two steps: sampling and quantization. In the sampling the value of the signal is stored every $\frac{1}{f_s}$ seconds, where f_s is the sampling rate expressed in Hz. Mathematically, the sampling process can be thought as multiplying the continuous function with an impulse train

$$\delta_T(t) = \sum_{n=-\infty}^{\infty} \delta(n - \Delta t), \tag{4.5}$$

where Δt is the sampling interval. The impulse function is defined as

$$\delta(t) = \begin{cases} 1 & ,t = 0 \\ 0 & ,t \neq 0 \end{cases} . \tag{4.6}$$

If the signal is sampled at a too low sampling rate, artifacts will appear (Mulgrew et al., 1999). To prevent this, the sampling rate must fulfill the Nyquist criterion: The sampling rate must be at least twice the highest frequency present. If it is not possible to know if the highest frequency present will be less than half of the sampling frequency, low-pass filtering can be used to make sure that the Nyquist criterion is met. In practice this filtering is always performed. The effect of the sampling in the frequency domain is depicted in Fig. 4.1.

After the sampling, the signal is quantized. Depending on the number representation the system uses, there is a limited number of number values available. The sampled values are

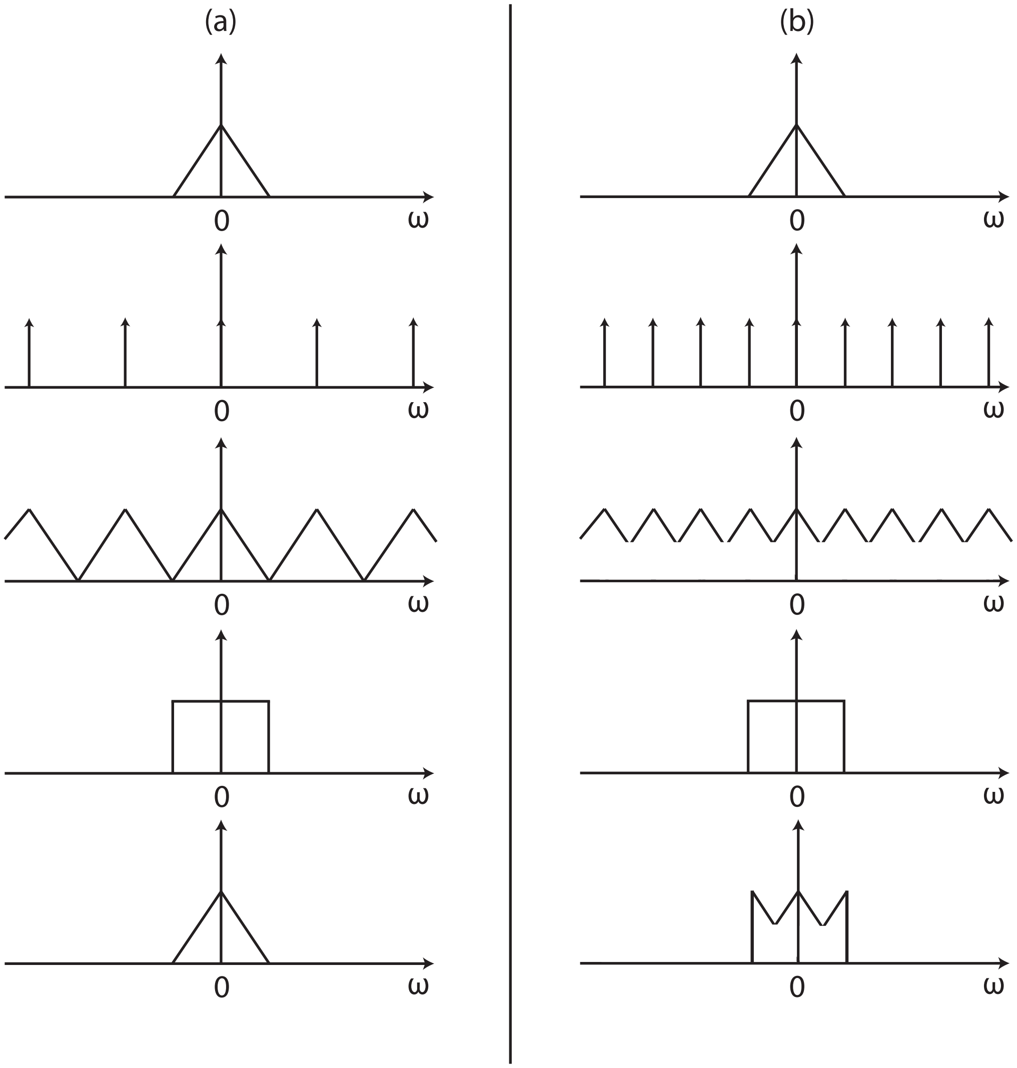


Figure 4.1: A conceptual picture showing the effects of too low a sampling rate. On both of the colums the coordinate systems from top to bottom represent analog signal's Fourier transform, the Fourier transform of a impulse train, convolution between these two(time domain multiplication), a low-pass filter to obtain the original signal, and the result of filtering. In (a) the sampling frequency just meets the Nyquist criterion and there is no aliasing. In (b) the sampling rate is too low and the output is not like the original. Note that increasing the sampling rate in time domain moves the peaks further away from each other in frequency domain.

rounded to the closest value available. Depending on the number of bits N_B used, there are $2^N_B - 1$ values available. Sampling and quantization are clarified in Fig. 4.2. Because of the quantization there will be error in the signal (Zador, 1982).

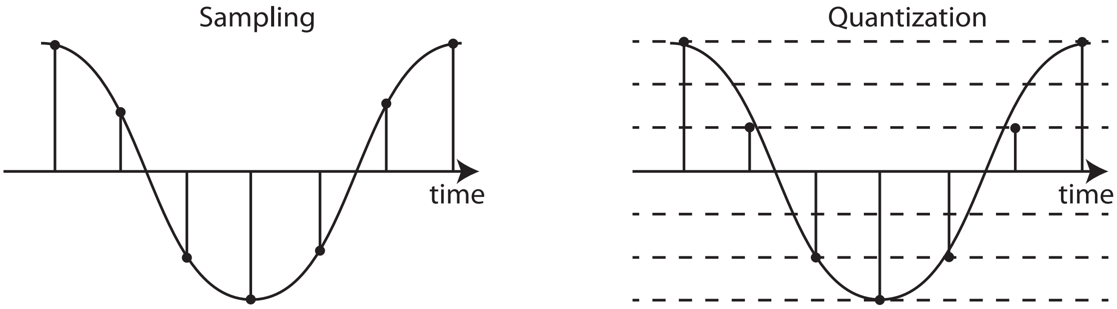


Figure 4.2: A segment of a sine wave first sampled and then quantized.

4.3 Discrete Fourier transform

If the signal has been sampled, it is not possible to perform the integration presented in Eq. . In the following, the discrete Fourier transform (DFT) is derived as it is done in Mulgrew et al. (1999). The sampled signal is represented with equation

$$x_c(t) = \sum_{n=-\infty}^{\infty} x(n\Delta t)\delta(t - n\Delta t). \tag{4.7}$$

The Fourier transform of the sampled signal is

$$X_c(\omega) = \int_{-\infty}^{\infty} x_c(t)e^{-j\omega t}dt. \tag{4.8}$$

By substituting equation 4.7 to 4.8 the following equation is obtained

$$X_c(\omega) = \int_{-\infty}^{\infty} \left[\sum_{n=-\infty}^{\infty} x(n\Delta t)\delta(t - n\Delta t)\right] e^{-j\omega t}dt. \tag{4.9}$$

The order of integration and summation is changed

$$X_c(\omega) = \sum_{n=-\infty}^{\infty} x(n\Delta t) \left[\int_{-\infty}^{\infty} \delta(t - n\Delta t)e^{-j\omega t}\right] dt. \tag{4.10}$$

Because of the properties of the impulse the integration is simplified into

$$X_c(\omega) = \sum_{n=-\infty}^{\infty} x(n\Delta t)e^{-j\omega t}dt. \tag{4.11}$$

This is the equation for the discrete Fourier transform. Similar to the analogue version, there is an inverse discrete Fourier transform

$$x(n\Delta t) = \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} X_c(\omega)e^{j\omega t}dt. \tag{4.12}$$

4.4 Real-time processing

In the system that is going to be introduced later in this thesis, the processing is done in real time. This means that a certain number of consecutive samples are sent to the processing and the same number of samples are output before taking in the next set of samples. The system is running in real-time, which means that it meets the following two criteria:

1. for every input there is a corresponding output
2. the delay between input and output is short enough for the current application.

4.5 Short-time Fourier transform

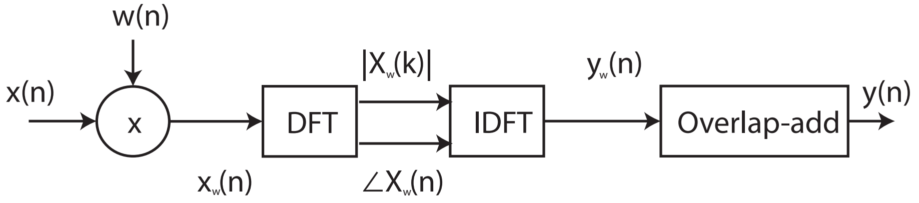


Figure 4.3: STFT block diagram.

In a real time system the transformations to the frequency domain and back need to be performed to the input signal block by block because it is not possible to wait for the whole signal and perform a DFT to it. Short-time Fourier transform or STFT is a tool that is used to analyze the frequency and phase content of a signal that changes over time. The derivation of the STFT is presented based on the material by Smith (2011). Equation for the STFT is

$$X_m(\omega) = \sum_{n=-\infty}^{\infty} x(n)w(n-mR)e^{-j\omega n}, \tag{4.13}$$

where $x(n)$ is the input signal at time t, $w(n)$ is the analysis window of length M, $W_m(\omega)$ is the DFT of windowed data centered about time mR, and R is the time index hop size. The analysis window has the Constant Overlap-Add (COLA) property at hop size R if

$$\sum_{m=-\infty}^{\infty} w(n-mR) = 1, \forall n \in \mathbb{Z}. \tag{4.14}$$

This means that the sum of all the consecutive windows has the value of one. This way the sum of all the successive DFTs performed is

$$\sum_{m=-\infty}^{\infty} X_m(\omega) \triangleq \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} x(n)w(n-mR)e^{-j\omega n} = \sum_{n=-\infty}^{\infty} x(n)e^{j\omega n} \sum_{m=-\infty}^{\infty} w(n-mR). \tag{4.15}$$

Applying the COLA property to the equation the last sum goes to one and the equation takes form

$$\sum_{m=-\infty}^{\infty} X_m(\omega) \triangleq \sum_{n=-\infty}^{\infty} x(n)e^{j\omega n} \triangleq X(\omega). \tag{4.16}$$

This proves that the sum of the successive DFTs of the signal is the DFT of the whole signal and the original signal can be reconstructed from the DFTs by adding them together with the same amount of overlap that the windows had. If the sum of the successive DFTs is the DFT of the whole signal then the inverse STFT is performed by taking the inverse DFT of the sum

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \sum_{m=-\infty}^{\infty} X_m(\omega)e^{j\omega n}d\omega = \sum_{m=-\infty}^{\infty} \frac{1}{2\pi}\int_{-\pi}^{\pi} X_m(\omega)e^{j\omega n}d\omega = \sum_{m=-\infty}^{\infty} x_m(n). \tag{4.17}$$

In the inverse STFT, each frame is transformed back to the time domain separately using IDFT

$$x_l(n) = \frac{1}{N}\sum_{n=0}^{N} X_l(k)e^{j\omega_k n}, \tag{4.18}$$

where X_l is the spectrum of the current frame, $e^{-j\omega_k n}$ is a complex sinusoid with frequency ω_k expressed in radians and N is the FFT size. The original signal is constructed from the separate blocks with overlap-add processing.

4.6 Overlap-add processing

Overlap-add is a method that is used to perform block-wise processing to a signal. Two examples of applications are block-wise convolution and the STFT processing discussed in the last section. In overlap-add processing the signal is divided to blocks and these blocks are processed separately. The processed signal is sent to the output and if the signal is longer than the original one, the values which exceed the original length are added to the start of the next block. The processing is clarified in Fig. 4.4.

4.7 Filtering

A filter is a way to select a subset of a larger set using some properties as a criteria. In the scope of this thesis a filter is a signal processing technique that alters a part of a signal in a desired manner. Four classes of filters are of interest in this thesis:

1. Low-pass filters
2. High-pass filters

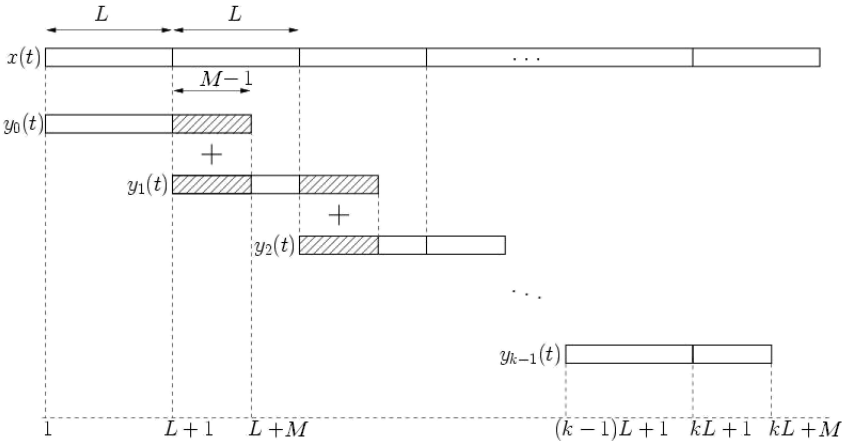


Figure 4.4: The overlap-add processing clarified with an example. On top there is the original signal which is split to blocks. These are then processed and the results are added together with some overlap. Adopted from Wikipedia (2013b).

3. Band-pass filters
4. All-pass filters.

Behavior of these filters is depicted in Fig. 4.5. The frequency where filter's behavior changes is called the cutoff frequency. A low-pass filter leaves the signal below the cutoff frequency as it is and attenuates the frequencies above it. A high-pass filter does the opposite. An all-pass filter has no effect on the amplitude of the signal, but it alters the phase response of the signal. There are many ways to represent a filter and its behavior. In this thesis, discrete difference equations are used. For example, a filter that always outputs the average of the current and the previous sample is presented with an equation

$$y[n] = \frac{1}{2}[x(n) - x(n-1)]. \tag{4.19}$$

Here y[n] refers to the current output and x[n] to the current input. Earlier values of the signal can be accessed by decreasing the index n.

Digital filters can be divided to two main categories:

1. Finite impulse response (FIR)
2. Infinite impulse response (IIR)

FIR filters have an finite response to impulse or any finite length input. In digital systems impulse response is the response of a system when the input is a single sample with value one. The output of an FIR filter is a weighted sum of the current sample and a finite number of previous samples. An FIR filter can be expressed with an equation

$$y[n] = \sum_{i=0}^{N} b_i x[n-i], \tag{4.20}$$

where $x[n]$ is the input signal, $y[n]$ is the output signal, b_i are the filter coefficients, and N is the order of the filter.

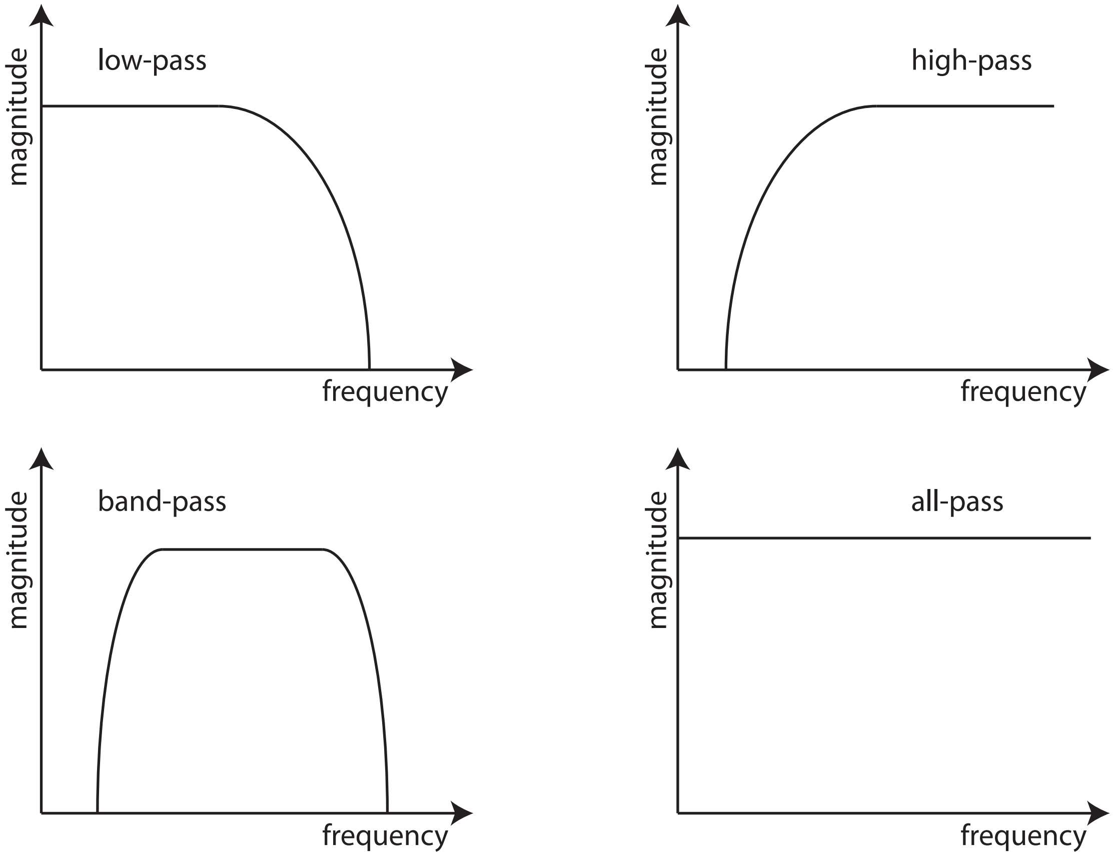


Figure 4.5: Magnitude responses for four different classes of filters. In the case of the all-pass filter there are changes in the phase response, so the filter is a way to alter the phase without touching the magnitudes.

IIR filters have an impulse response that is non-zero for an infinite time. Output consists of delayed output values and possibly delayed input values. An equation for an IIR filter is

$$y[n] = \frac{1}{a_0}\left(\sum_{i=0}^{P} b_i x[n-i] - \sum_{j=1}^{Q} a_j y[n-j]\right), \tag{4.21}$$

where P is the feedforward filter order, Q is the feedback filter order, b_i are the feedforward filter coefficients, and a_i are the feedback filter coefficients.

Both of the filter types have their advantages and disadvantages. The FIR filters are always stable whereas in designing the IIR filters the coefficients have to be chosen in a way that the stability is ensured. The IIR in general require a lower order to implement a certain filter. The FIR filters are simpler to implement than the IIR filters.

In practice, filtering with an FIR filter can be done for example using convolution. Convolution is a mathematical operation on two functions f and g, producing a third function. This third function can be thought to be the result of filtering f with g. Discrete convolution is defined as

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m], \tag{4.22}$$

where f and g are two functions mentioned earlier. One property of the Fourier transform is that time domain convolution corresponds to frequency domain multiplication. Thus the filtering in frequency domain is done by simply multiplying the signal with the filter response.

Chapter 5

Object transformations

The number one use for object transformations are the computer graphics. They are used, for example, to move, scale, and rotate graphical objects. In addition to the graphics use, object transformations can be applied to sound objects. In this thesis, an object is defined as a set of points. Shape of the object defines boundaries inside which all the points belonging to the object lie. Each object has its own coordinate system in which the definition is made. In this work, object transformations are used to move, scale and rotate sound objects. This enables the creation of a virtual world where the listener and the objects can move around without breaking the immersion.

In this work, homogenous coordinates are used to perform the transformations. Homogeneous coordinates have several advantages in comparison to the cartesian coordinates. For example they can represent points in the infinity with finite coordinates and the formulas involving homogeneous coordinates are simpler than their cartesian counterparts. The homogeneous coordinates are obtained from the cartesian coordinates by adding an extra component of one to a vector presenting a point, that is

$$\vec{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \tag{5.1}$$

A transformation is any operation that maps a certain point x to another point $A(x)$ within the coordinate system. Three types of transformations required in this thesis are translation, rotation, and scaling. Last two of these fall within the category of linear transformations which follow the two conditions:

$$1.\ A(\alpha x) = \alpha A(x), \quad \alpha \in \mathbb{R} \tag{5.2}$$

$$2.\ A(x + y + z) = A(x) + A(y) + A(z), \quad x, y, z \in \mathbb{R}^3 \tag{5.3}$$

Translation is a transformation which changes the position of the object for a fixed amount but has no effect on the shape or the orientation of the object. Mathematically a transformation is a translation if there exists a fixed $u \in \mathbb{R}^3$ such that $A(x) = x + u$ for all $x \in \mathbb{R}^3$. A transformation is called affine if it can be written as a combination of linear transformation and translation.

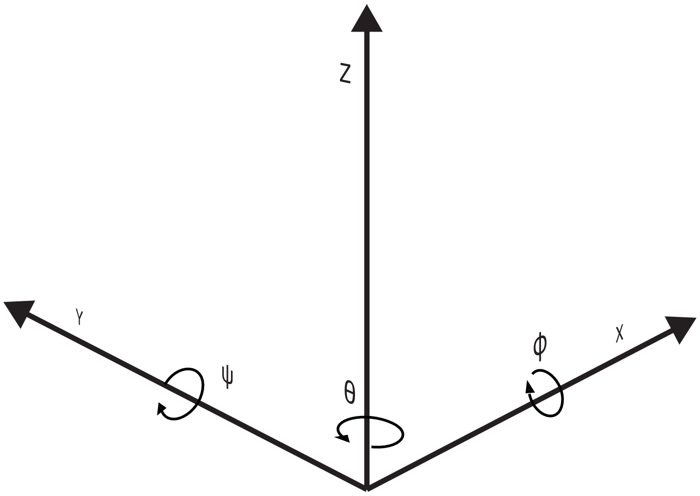


Figure 5.1: Coordinate system with rotations.

One way to perform an affine transformation is to use an affine transformation matrix. It can be constructed from separate matrixes for the rotations, scaling, and translations. The rotations around the three axes of the coordinate system in Fig. 5.1 are done with matrixes

$$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.4a}$$

$$R_y = \begin{pmatrix} \cos\psi & 0 & \sin\psi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\psi & 0 & \cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.4b}$$

$$R_z = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{5.4c}$$

The translation can be represented with a matrix

$$T = \begin{pmatrix} 0 & 0 & 0 & dx \\ 0 & 0 & 0 & dy \\ 0 & 0 & 0 & dz \\ 0 & 0 & 0 & 0 \end{pmatrix}, \tag{5.5}$$

where dx is the translation in relation to x-axis, dy is the translation in relation to y-axis, and dz is the translation in relation to z-axis. The scaling of the object is done with the matrix

$$S = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{5.6}$$

where s_x is the scaling coefficient for the x-coordinate, s_y is the scaling coefficient for the y-coordinate, and s_z is the scaling coefficient for the z-coordinate.

The rotations can be combined into a single affine transformation matrix by multiplying the three separate rotation matrices with each other, which results to

$$R_{tot} = \begin{pmatrix} \cos\theta\cos\psi & \cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \sin\phi\sin\psi - \cos\phi\sin\theta\cos\psi & 0 \\ -\cos\theta\sin\psi & \cos\phi\cos\psi - \sin\phi\sin\theta\sin\psi & \sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & 0 \\ \sin\theta & -\sin\phi\cos\theta & \cos\phi\cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.7}$$

The final transformation matrix is obtained by combining the rotation, the translation and the scaling

$$A_{tot} = R_{tot}S + T \tag{5.8}$$

To obtain the new location vectors for the shape, the original location vector of each point in the shape is individually multiplied with the matrix. This multiplication is represented by equation

$$\vec{p}_{\text{new}} = \vec{p}A, \tag{5.9}$$

where $\vec{p}$ is the location vector of a single point of the object and $\vec{p}_{\text{new}}$ is the new location vector.

Chapter 6

Directional Audio Coding

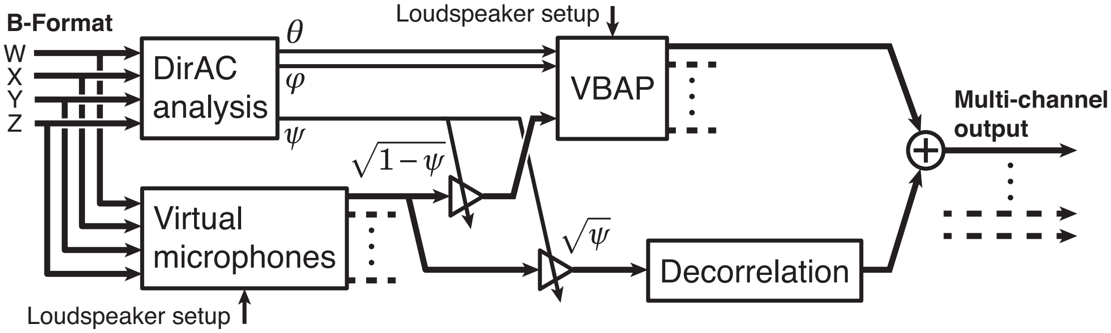


Figure 6.1: Block diagram for DirAC processing. Adopted from Pihlajamaki et al. (2013).

Directional audio coding (DirAC) is a method of audio coding that uses the psychoacoustical knowledge of the human hearing to code and reproduce a three dimensional sound field (Pulkki, 2007). The sound field is assumed to be captured from a single point. This reduces the cost of the method in comparison to the other systems that use recordings captured with multiple microphones. DirAC processing is split up into synthesis and analysis parts. In the analysis, information about the direction of arrival and the diffuseness of the sound is extracted, and in the synthesis, the sound is reproduced using this information. It is possible to alter the data obtained before the actual synthesis. DirAC can be applied, for example, to music reproduction, teleconferencing, or virtual-world applications which are discussed in this thesis.

6.1 Assumptions about spatial hearing

The following assumptions about the human spatial hearing are made in Pulkki (2007):

1. The direction of arrival of the sound transforms into interaural time difference (ITD), interaural level difference (ILD), and monaural localization cues.
2. The diffuseness of the sound transforms into interaural coherence cues

3. Timbre depends on the monaural spectrum together with the ILD, the ITD, and the interaural coherence of the sound.
4. The direction of arrival, diffuseness, and spectrum measured in one position with the temporal and spectral resolution of the human hearing defines the auditory spatial image the listener perceives.

6.2 B-format

In the original high quality DirAC, B-format signals are the standard input. They are discussed in more detail in (Benjamin and Chen, 2005) . The B-format signal used in this thesis consists of four signals that are called w(t), x(t), y(t), and z(t). In a recording situation, each signal is captured with a separate microphone. The microphones are placed in the same spatial location. w(t) is captured with an omnidirectional microphone that treats sound arriving from every direction equally. x(t), y(t), and z(t) are captured with figure-of-eight microphones that are pointing towards the respective axes in the coordinate system that is used. These three signals contain the directional information that is used in the reproduction of the sound. The directional patterns with the coordinate system used are depicted in Fig. 6.2. In most of the available B-format recordings, such gains are applied to the signals that $w(t)$ is multiplied with $\frac{1}{\sqrt{2}}$. This has to be taken into account in the processing.

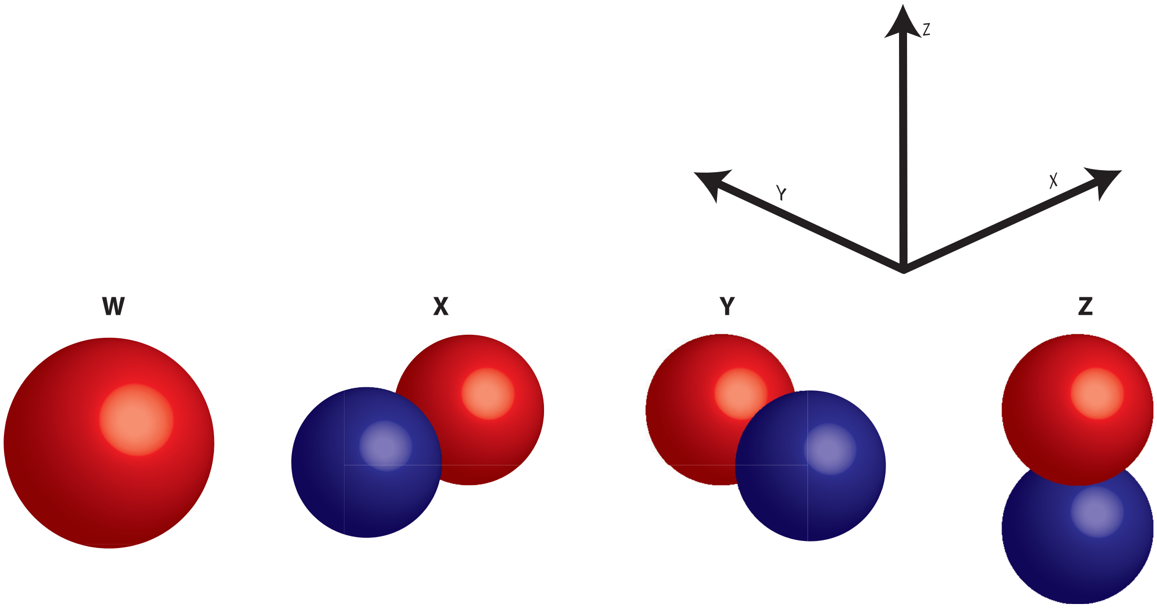


Figure 6.2: B-format microphone directional patterns.

6.3 Analysis

6.3.1 Time-frequency analysis

The processing is done separately for different frequency bands which must somehow be obtained from the original signal. Two ways that have been used to do this are the filter banks or the STFT. STFT based approach has a fixed resolution, meaning that all the obtained frequency bands are of the same size. However, the human hearing splits the audible frequencies to bands with varying size as described in Chap. 3.2.5. With the filter banks it is possible to create a transformation that is closer to the one performed by the human hearing than the one with STFT.

6.3.2 Directional analysis

The analysis is based on calculating the energy and the intensity at each frequency band and using these to estimate the direction of arrival and the diffuseness. The sound pressure at each frequency band is estimated to be the value of the same band from W(k)

$$P(k) = W(k). \tag{6.1}$$

The particle velocity $\vec{u}$ is estimated from the three remaining signals using the equation

$$\vec{U}(k) = \frac{1}{\sqrt{2}Z_0}(X(\vec{k})e_x + Y(\vec{k})e_y + Z(k)\vec{e_z}), \tag{6.2}$$

where $\vec{e_x}$, $\vec{e_y}$ and $\vec{e_z}$ are the cartesian unit vectors. By combining assumptions from equations 6.1 and 6.2 and equation 2.13 for intensity, a way to estimate the intensity from a B-format signal is obtained

$$\vec{I}(k) = \mathfrak{F}[P(t)\vec{u}(t)] = \frac{1}{\sqrt{2}Z_0}\mathfrak{R}\left[W(k)^*(X(k)\vec{e}_x + Y(k)\vec{e}_y + Z(k)\vec{e}_z)\right]. \tag{6.3}$$

Intensity is a vector with three components. Their magnitudes are marked as I_x, I_y, and I_z. Applying the assumptions about sound pressure and particle velocity to equation 2.10, we get the equation for energy

$$E(k) = \frac{\rho_0}{2Z_0^2}(|W(k)|^2 + \frac{1}{2}(|X(k)|^2 + |Y(k)|^2 + |Z(k)|^2)). \tag{6.4}$$

To estimate the direction of arrival, a vector that is opposite to the intensity vector is calculated. The equations for the azimuth and elevation are

$$\theta(k) = \arctan\left(\frac{-I_y(k)}{-I_x(k)}\right) \tag{6.5}$$

and

$$\varphi(k) = \arctan\left(\frac{I_z(k)}{\sqrt{I_x(k)^2 + I_y(k)^2}}\right). \tag{6.6}$$

6.3.3 Diffuseness analysis

The diffuseness of the signal is estimated using the length of the intensity vector and energy. Length of the intensity vector is

$$|\vec{I}(k)| = \sqrt{I_x^2 + I_y^2 + I_z^2} \tag{6.7}$$

The estimate of the diffuseness is

$$\psi(k) = \frac{\|\langle|\vec{I}(k)|\rangle\|}{c_0\langle E(k)\rangle}, \tag{6.8}$$

where $\langle\rangle$ denotes time averaging and $\|\cdot\|$ denotes the l^2 norm.

6.4 Synthesis

In the synthesis, a separate signal for each loudspeaker is created using virtual microphones. A copy of every loudspeaker signal is sent to both diffuse and non-diffuse processing. In the non-diffuse processing, directional data is added to the sound in order to reproduce it in a way that it has a perceivable direction. The diffuse stream is made to lack prominent direction, i.e., it sounds like the sound is coming from all directions. This is done by using decorrelation. In the following sections virtual microphones, the non-diffuse processing, and the diffuse processing are explained in more detail.

6.4.1 Virtual microphones

The simplest way to form the loudspeaker signals would be sending the omni-directional signal $w(t)$ to all channels. Using a linear combination of the B-format microphone channels allows the creation of multiple directional patterns pointing towards any direction. Informal listening tests have proven that using directional patterns pointing towards the loudspeakers improve the reproduction quality (Vilkamo et al., 2009). This is because the separation of different directions is enhanced by the virtual microphones. The coefficients for every loudspeaker channel are obtained with the equations (Vilkamo, 2008)

$$g_w(k) = 1 - \kappa, \tag{6.9}$$

$$g_x(k) = \frac{\kappa}{\sqrt{2}}\cos(\theta)\cos(\varphi), \tag{6.10}$$

$$g_y(k) = \frac{\kappa}{\sqrt{2}}\sin(\theta)\cos(\varphi), \tag{6.11}$$

$$g_z(k) = \frac{\kappa}{\sqrt{2}}\sin(\varphi). \tag{6.12}$$

Here θ is the azimuth and φ is the elevation of the current loudspeaker. κ is the coefficient that defines the virtual microphone pattern. Its values range from 0 (omni-directional) to 1 (figure-of-eight).

Using the virtual microphones introduces a difference in the gains for the diffuse and the non-diffuse sound. For this reason correction coefficients have to be applied both to the diffuse and non-diffuse streams (Vilkamo, 2008). The correction coefficients are

$$C_{\mathrm{nd}} = \sqrt{1 - \psi + M\psi} \tag{6.13}$$

and

$$C_{\mathrm{d}} = \sqrt{1 - \psi + M\psi}\sqrt{N} \tag{6.14}$$

where C_{nd} is the coefficient for the non-diffuse stream, C_{d} is the correction coefficient for the diffuse stream, ψ is the diffuseness value, N is the number of loudspeakers and M is an assisting variable

$$M = 1 - \kappa + \frac{\kappa^2}{3}. \tag{6.15}$$

After the creation of the loudspeaker signals, they are split up into diffuse and non-diffuse streams by multiplying the signal with $\sqrt{\psi}$ and $\sqrt{1-\psi}$ respectively.

6.4.2 Non-diffuse synthesis

The part of the signal that is multiplied with $\sqrt{1-\psi}$ is processed in the non-diffuse synthesis. At every time instant, a gain for each loudspeaker at each frequency band is calculated and the input signal is filtered with these gains. The gains are computed using vector-base amplitude panning (VBAP).

VBAP is a method for three-dimensional panning (Pulkki, 1997). It uses sets of three loudspeakers to position a monophonic sound source. The sound source can be positioned within a triangle that is defined by these three loudspeakers. The location of the source within the triangle is defined by the gains of the three loudspeakers.

VBAP finds the three loudspeakers that are closest to the desired location of the sound source and uses them to play the sound. $\vec{l_n}$ is a three-dimensional unit vector pointing towards the loudspeaker n. $\vec{p}$ is a three-dimensional vector pointing towards the sound source. $\vec{p}$ can be defined as a combination of the loudspeaker vectors

$$\vec{p} = g_1\vec{l_1} + g_2\vec{l_2} + g_3\vec{l_3}, \tag{6.16}$$

where g_1, g_2 and g_3 are the loudspeaker gains. If the gains are expressed as a vector $\vec{g} = [g_1 \quad g_2 \quad g_3]$, and the loudspeaker vectors as a matrix $\mathbf{L} = [\vec{l_1} \quad \vec{l_2} \quad \vec{l_3}]^T$ the gains can be solved with the equation

$$\vec{g} = \vec{p}\mathbf{L}^{-1} \tag{6.17}$$

The gains given by this equation can be scaled according to the following equation

$$\vec{g} = \frac{\sqrt{C}\vec{g}}{\sqrt{g_1^2 + g_2^2 + g_3^2}}, \tag{6.18}$$

where C is the desired sound power level and can be expressed as

$$C = g_1^2 + g_2^2 + g_3^2. \tag{6.19}$$

6.4.3 Diffuse synthesis

The part of the signal multiplied with $\sqrt{\psi}$ is processed in the diffuse synthesis. Diffuse sound is the part of the sound that is lacking a prominent direction. In the DirAC, the diffuse sound is created by decorrelating the signal separately for each loudspeaker channel. With audio signals, decorrelation is defined as a process that transforms one signal into a multitude of signals, which all sound like the original but have a different waveform and have as little correlation between them as possible (Bouéri and Kyriakakis, 2004). Correlation between two signals y_1 and y_2 is defined as

$$\Omega(m) = \sum_{n=-\infty}^{\infty} y_1(n) y_2(n+m), \tag{6.20}$$

where m is the temporal difference between $y1$ and $y2$ in samples. It is simpler to express the correlation as one number which is chosen to be the maximum of the absolute value of the correlation function. This gets values from 0 to 1, 1 meaning that the signals are the same and the closer to 0 the value gets the more different the signals are from each other.

A method for producing decorrelated signals was introduced by Kendall (1995). In this method, the input signal is filtered with an all-pass filter that has a random phase. To obtain multiple signals that are decorrelated, a set of all-pass filters is used. The filters are created in the frequency domain. The amplitudes of the filter are all set to unity and the phases are chosen to be random number sequences that are orthogonal to each other. A problem with this method is that the frequency response is set to unity only on certain points. Between these points, the value differs from unity because of the phase. This is demonstrated in Fig. 6.3. A better method was introduced by Bouéri and Kyriakakis (2004). In this method a set of linear-phase band-pass filters is used. Each of the filters corresponds to one critical band of the human hearing introduced in section 3.2.6. These filters guarantee a smooth phase change between different bands. A random time shift varying from -20ms to 20ms is applied to each frequency band. This way the precedence effect is not affecting the perception of the signal as it would be if the whole signal would be delayed by a fixed time. The time shift has been made frequency dependent to prevent too high delays on high frequencies because these would create audible artifacts (Pihlajamäki, 2008).

6.5 Binaural processing

The original way to reproduce the signal in DirAC was to use multiple loudspeakers. Using headphones instead of loudspeakers is another way to render the output. In this case, the directional information that comes naturally from the placement of the loudspeakers has to be produced artificially. As described in Chap. 3.2, the directional information

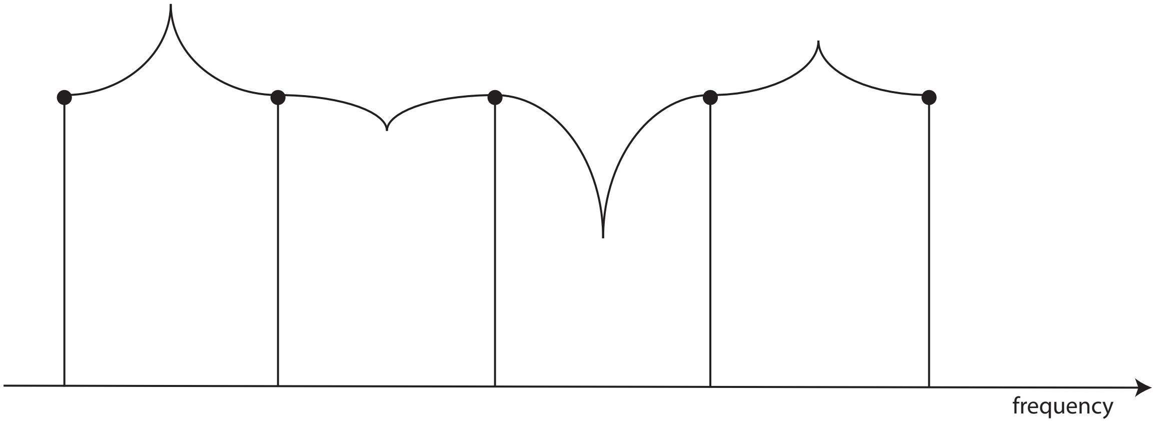


Figure 6.3: Effect of the randomized phase. Adapted from (Kendall, 1995).

is mainly in the difference of the sound in the two ears, which is caused by the torso, head, pinnae, and the distance between the ears. This information can be captured by putting a microphone to the entrance of both ear canals and measuring test tones played from different directions (Laitinen, 2008). From the measurement, a function to describe the alteration in the arriving sound is obtained. This function is called the head-related transfer function (HRTF). For every direction there is a separate HRTF for the right ear and for the left ear. Creating a certain direction through headphones is done by sending a monophonic signal filtered with the right HRTF to the right ear, and the monophonic signal filtered with the left HRTF to the left ear.

6.5.1 Head tracking

When using headphone rendering, it is possible to enhance the immersion by using head tracking (Laitinen and Pulkki, 2009). The human hearing uses unconscious head movements to enhance the localization of sound sources and the use of head tracking allows this. Other advantage of the head tracking is, that the sound scene can be reproduced in a way that the sound objects do not move when the user moves his head. For example, in a situation when the user hears a sound event in front and turns the head 90 degrees to the right the event will be still perceived in front without head tracking even though it should be perceived 90 degrees to the left. This is demonstrated in figure 6.4.

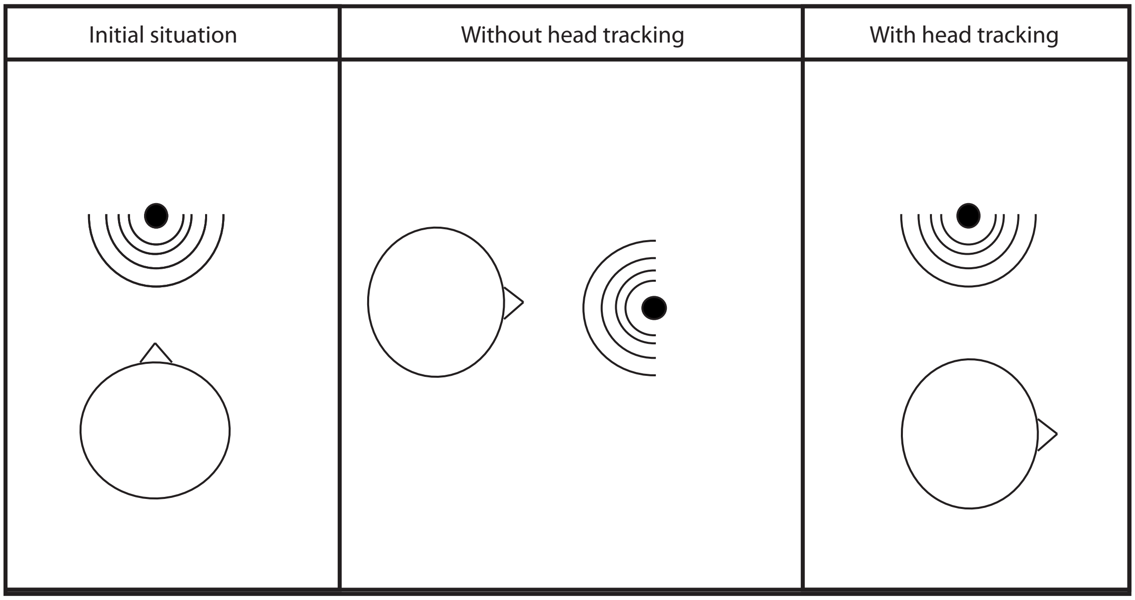


Figure 6.4: Effect of head tracking.

Chapter 7

Virtual-world Directional Audio Coding

In this thesis, a virtual world is seen as a space in which the listener and an user defined number of sound objects exist. These sound objects can be mono sources, B-format recordings, or other multichannel recordings. Their spatial attributes should be presented to the listener in a controlled way. The spatial attributes include perceived distances, directions, and spatial extent of the objects. A system doing this based on the DirAC has been designed (Pulkki et al., 2009; Laitinen et al., 2012). This first system worked well but had some undesired properties in extreme situations. In a situation where there were multiple anechoic sources the perceived direction was fluctuating and the system added some undesired feel of spaciousness to anechoic recordings. To overcome these, a new modular architecture has been presented by Pihlajamaki et al. (2013). In the modular architecture, the processing is split up into multiple frontends and a single backend as depicted in Fig. 7.1. This way the analysis and the synthesis of the different sources is separated and the previously mentioned problems are solved. In order to maintain a low enough computational complexity the heaviest tasks are performed only once, in the backend. In the following sections the functionalities of different frontends and the backend are described.

7.1 Frontends

The frontends perform the processing of different kinds of inputs. As a result, each frontend outputs two signals: A signal for the diffuse stream and a signal for the non-diffuse stream. In the following sections different frontends and their functionalities are discussed.

7.1.1 Extended panning

The block diagram of the extended panning unit is presented in Fig. 7.2. This unit provides panning for point-like sources and spatially extended sources. The input for this frontend is a mono audio file that is to be rendered. As an information for the rendering, the unit receives five different inputs:

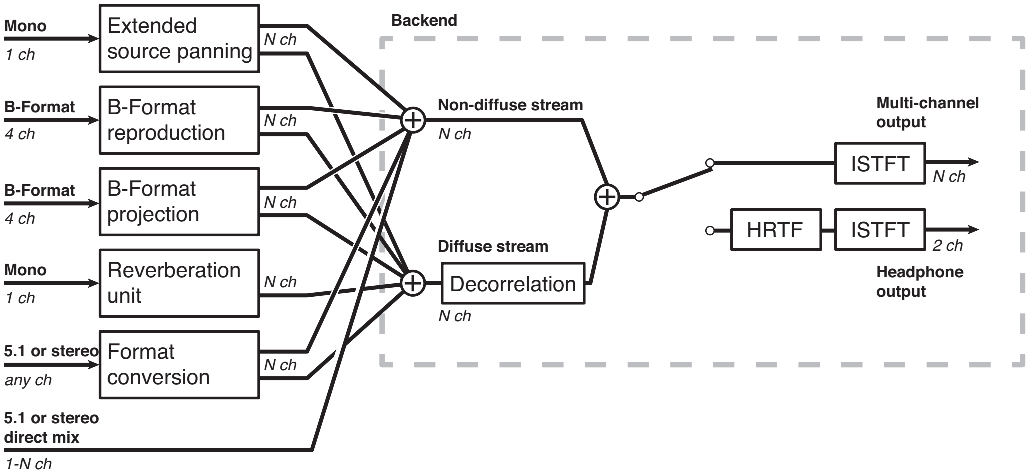


Figure 7.1: Full virtual world architecture. Adapted from Pihlajamaki et al. (2013).

- listener location
- source location and geometry
- mono input signal
- diffuseness value for the object
- gain for the object.

Before any other processing, delay, a distance attenuation factor, and an user defined gain are applied to the mono signal. The delay is calculated based on the distance between the sound object and the listener. The sound object is presented as a group of points, where each point corresponds to a certain frequency band. Different sizes for the object are created by increasing and decreasing the volume inside which the points are distributed. From the location of each point the relative direction to the listener is calculated with the help of object transformation introduced in Chap. 5. The azimuth and elevation angles are then fed to the VBAP block which calculates loudspeaker gains separately for each frequency band. These gains are applied to each loudspeaker signals and the signals are passed to the backend. The diffuse stream is passed to the backend as it is because the decorrelation is performed once for all the signals.

7.1.2 B-format reproduction

The B-format reproduction unit is identical with the normal analysis and synthesis of the DirAC when combined with the unified backend. These functionalities are described in Chap. 6. The unit can be used for example to produce background ambience to the virtual world from a B-format recording. The block diagram for this unit is in figure 7.3.

7.1.3 Reverb

The reverberation generation unit is depicted in Fig. 7.4. Its input is the mono signal for the mono source that requires added reverberation. A copy of the signal is passed to three

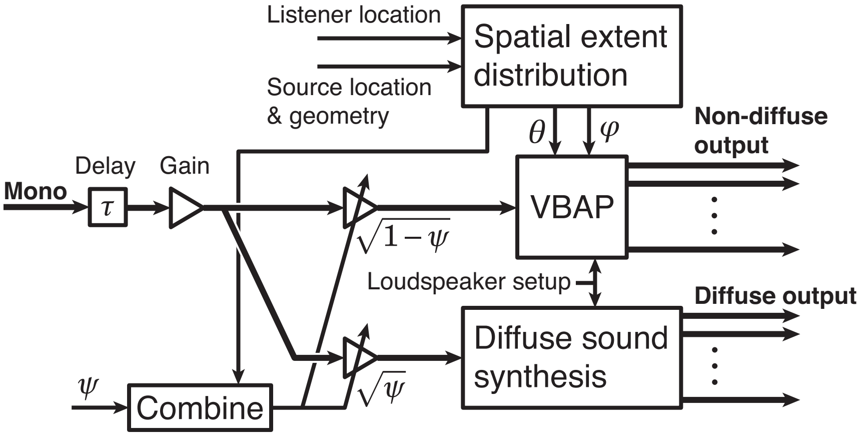


Figure 7.2: Extended panning unit. Adopted from Pihlajamaki et al. (2013).

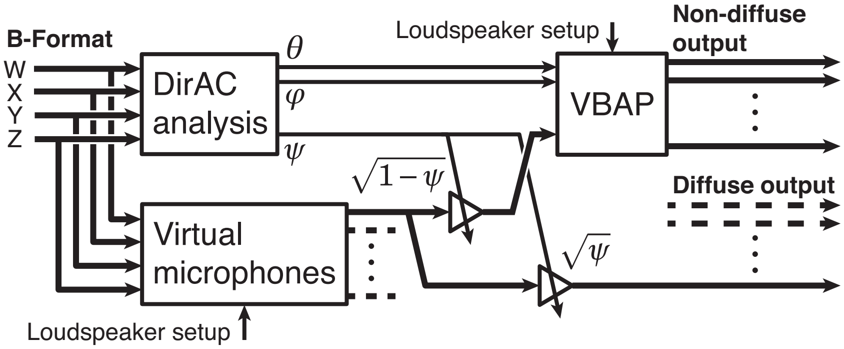


Figure 7.3: B-Format reproduction. Adopted from Pihlajamaki et al. (2013).

separate reverberators. These three reverberators correspond to reverberation coming from three different directions, more accurately the three axes of the coordinate system used which is depicted in Fig. 5.1. The reverberators are designed in a way that they produce signals orthogonal to each other. A virtual microphone technique introduced in Chap. 6.4.1 is applied to these three signals in order to create a separate signal for each loudspeaker. The result signals are mixed to the diffuse output of the frontends and they are processed in the backend.

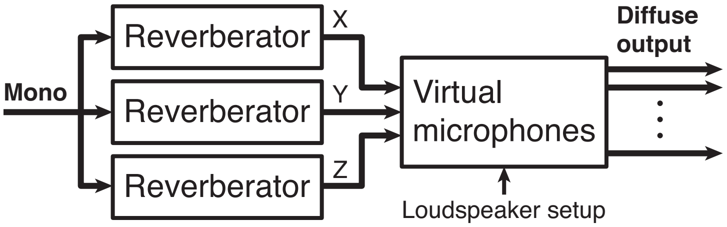


Figure 7.4: Reverberation unit. Adopted from Pihlajamaki et al. (2013).

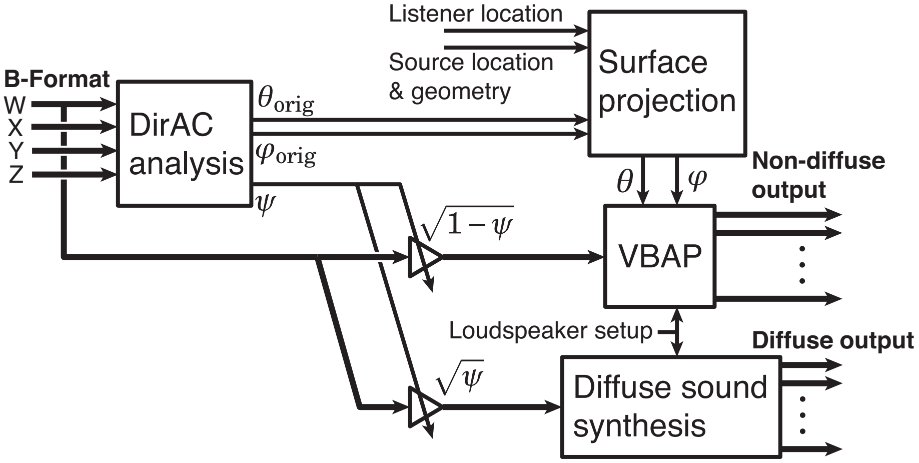


Figure 7.5: B-Format projection unit. Adopted from Pihlajamaki et al. (2013).

7.1.4 Projection

The projection frontend is based on research by Pihlajamäki and Pulkki (2012). This is a combination of the B-format reproduction and extended panning units which allows projecting a B-format recording on a surface and listener movement in relation to the recording. This is done by first performing the normal DirAC analysis for the input B-format signal. The directional information that is obtained from the analysis is transformed into vectors that point from the locations of the sounds to the recording position. If the listener moves away from the center position, the vectors are updated with an affine transformation. Transformations were described in Chap. 5. This transformation leads to new directions and in addition to that, attenuation is calculated based on the $\frac{1}{r}$ law introduced in Chap. 2.4.1. With this information the sound is reproduced in a similar manner as in the extended panning frontend using the w(t) component of the B-format signal as the mono source.

7.2 Backend

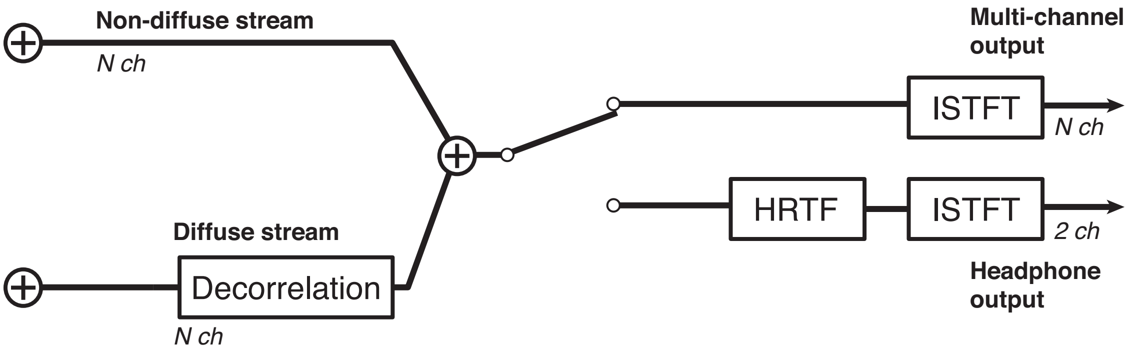


Figure 7.6: Backend.

The backend is an unified unit that performs its processing to signals that have been combined from different frontends. The block diagram for the backend is in Fig. 7.6. The inputs to the backend are the non-diffuse and diffuse streams, both of which contain separate signals for each loudspeakers. Before adding the two streams together, the diffuse stream is decorrelated. The process of decorrelation was introduced in Chap. 6.4.3. If loudspeakers are used as the output method, the signals from the addition of the two streams is transformed back to the time domain using ISTFT. If the headphone output is used, the signals are filtered with the according HRTFs, added together to the left ear signal and the right ear signal and then brought back to the time domain with the ISTFT.

Chapter 8

Implementation

In this chapter, the implementation and the practical choices that were made during the work are discussed. First, the programming languages used and the structure of the program are discussed. After that, the different parts of the virtual-world system described in the previous chapter and their implementation issues are discussed. In the end, there are sections about the binaural implementation and the user interface that was designed and programmed.

8.1 Programming

The system described in the previous chapters was implemented as an external for Max6 programming language developed by Cycling74 (2013). Max6 allows a relatively easy creation of interactive 3-D graphics with its variant Jitter. This made the language well suitable for the requirements of the system. It was possible to write the externals using C as the language which is also good because of the efficiency of the language in computationally intensive tasks. The external which implements the virtual-world DirAC was written with C, the graphical implementation of the virtual world was written with Jitter, and the communications between these two with Max6 and Javascript.

8.2 Ring buffers

A ring buffer is a fixed size sampled data buffer that is used in a way if it were connected from end to end. When the end of the buffer is reached the next sample of data that is accessed is the first one. The implemented buffer consists of a data chunk, a write pointer and a user defined number of read pointers. The data chunk is used to hold sampled data. The write pointer points to the sample where the next input is to be written and is moved one step forward always when a sample is written. The read pointers point to locations where the data is to be read and they are also moved forward when samples are read. In the implemented system, ring buffers were used to store the signal on many occasions.

8.3 Creating the world and the objects

When the application is launched, only some files are loaded to the program memory. These include the filters used for the decorrelation and the head-related transfer functions. In addition to this, basic parameters, like the sampling rate and frame size are loaded. The world is also created but at this point it only contains the listener object and nothing else. When the application is running the user can create sound objects. To create one, the user must choose a mono file or a B-format file that is going to be the new object. The contents of the file are loaded to the program memory but the file is not transformed to the frequency domain. This would make the processing less computationally intensive but it is not possible to easily add a delay to a frequency domain signal. The mono objects are all initially positioned to the origin and the user can move them to other position if that is necessary.

Upon the creation of the object, a distribution for the frequency bands of the object is created. These distributions are needed for the spatial extent distribution block of the extended panning unit as was described in Chap. 7.1.1. When the signal is going to be in the frequency domain, the different frequency bins that each represent a certain frequency band are going to be distributed inside the boundaries of the object. The boundaries are defined by the shape of the object. If the user changes the shape of the object, the points have to be distributed again. For a cube lying one corner in the origin the x, y, and z coordinates of each point are chosen to be values between 0 and 1.

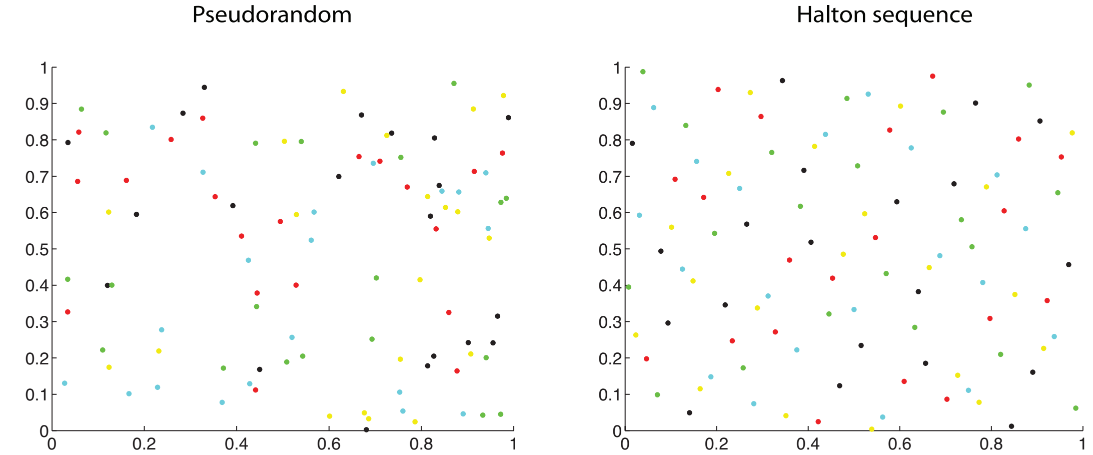


Figure 8.1: On the right, points spread to a rectangle using halton sequence with base two for x-coordinate and a halton sequence with base three for y-coordinate. On the left, points spread to a rectangle using uniformly distributed pseudo-random sequence created by Matlab's rand() function. Points with the same color present sets of 20 consecutive points. It is noteworthy that halton spreads the sets of consecutive points evenly to the whole square.

There are two ways that have been used to distribute the frequency bins inside the boundaries of the object. One way is to use a random number generator to create a pseudo-random numbers from uniform distribution for the x, y, and z components of the location vector of each point. The other way is to use a uniformly distributed sequence.

With a uniformly distributed sequence, the points are more evenly spread. The difference between a pseudo-random and a uniformly distributed sequence is depicted in Fig. 8.1. Halton series with bases 2,3, and 5 for the x, y, and z components of the location vector of each point was used in the implementation. Algorithm that produces a single number from the series is:

```
double haltonNumber(int index, int base){
    double result = 0;
    double d = 1/(double)base;
    int i  = index;
    while (i>0) {
        result = result + d*(i%base);
        i = floor(i/base);
        d = d/base;
    }
    return result;
}
```

Index refers to the number that is created and base must be a prime number. Each different base value creates a different series. A halton sequence with base two starts this way:

$$\frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{5}{8}, \frac{3}{8}, \frac{7}{8}, \frac{1}{16} \cdots \tag{8.1}$$

8.4 Transformation to the frequency domain

In the implementation, the transformation to the frequency domain was done with the short-time Fourier transform introduced in Chap. 4.5. The STFT has a fixed resolution, which is dependent of the length of the windowing function. Depending on the length, either the time or frequency resolution is good but not both of them. This is because of the properties of the Fourier transform (Mulgrew et al., 1999). For audio signals, a good frequency resolution is required for the low frequencies and a good time resolution is required for the high frequencies because the human hearing has a similar resolution.

To overcome the previously mentioned resolution problem, a multi-resolution approach to STFT has been developed by Laitinen et al. (2011). In the multi-resolution approach, the signal is split to two or more frequency bands with filtering and the bands are processed separately. A smaller window size if used for the high frequencies to provide better temporal resolution and make the synthesis of transient sound events possible. For the low frequencies a longer window is used to obtain better frequency resolution and sound quality. To perform the separation to the two frequency bands, filtering the signal with both low-pass and high-pass filters with the same cutoff frequency, could be performed. However, in this implementation the signal was only low-pass filtered and the resulting signal was subtracted from the original to obtain the high-pass signal. This method has two advantages:

1. No information about the original signal is lost, so filtering and adding the two results together provides perfect reconstruction.

2. Filtering and subtraction is less computationally intensive than filtering twice.

The disadvantage of this kind of filtering is, that the behavior of the signal close to the cutoff frequency is not certainly known. The DFT delays the signal and the delay is proportional to the size of the DFT. In the case of two different resolutions, one of the two signals is going to be delayed more than the other. Effects of this additional delay have been studied by Pihlajamaki and Pulkki (2011). The way to overcome this problem, was to add an additional delay to the signal path with smaller resolution and smaller delay. A block diagram of the multi-resolution system with the additional delay is in Fig. 8.2.

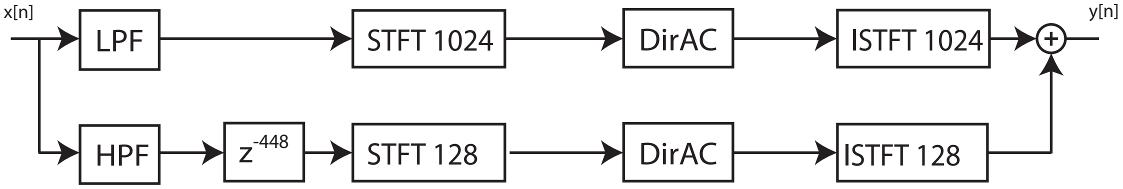


Figure 8.2: Multi-resolution processing.

The transformation back to the time domain is done with the inverse STFT. The original signal is recovered from the transform using Overlap-Add method which was explained in Chap. 4.6.

8.5 Extended panning unit

The extended panning unit handles the panning of the mono signals in the virtual world and in its basic functionality was described in Chap. 7.1.1. If there are multiple sound objects that are to be panned, this processing is done separately for each object.

In the beginning of the processing, a frame from the sound file is read from a ring buffer holding it. The signal is delayed according to the time it takes for the sound to travel from the object to the listener. Knowing the location of an object, this time in samples can be calculated with the equation

$$t_s = \frac{d}{c_0} f_s, \tag{8.2}$$

where d is the distance in meters, f_c is the sampling rate, and c_0 is the speed of sound. The delay is applied to the signal by reading the frame from a position that is t_s samples behind the actual read pointer. After this, the transformation to frequency domain is done as described in the previous section.

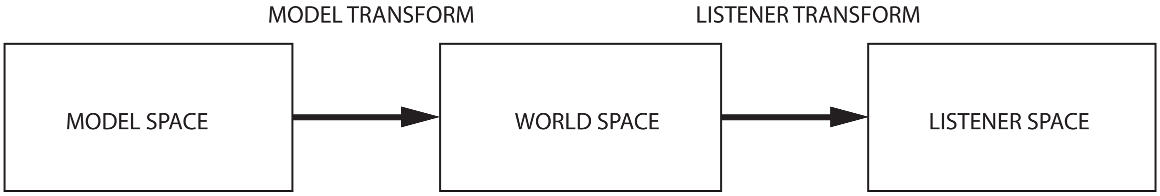


Figure 8.3: Transforming the object from the model space to the the listener space.

When the signal is in the frequency domain, all the frequency bins have their own location as was described earlier. The location points exist in the model space and need to be transformed into the listener space so that their location relative to the listener is known. This is done in two steps: first, the object is transformed from the model space to the world space with the model transform, and then from the world space to the listener space with the listener transform. The transformations are presented in Fig. 8.3. Fundamentals of the transformations were presented in Chap. 5. The first transformation is done using the location of the object, the orientation of the object, and the size of the object as the parameters. In the second transform, the location of the listener and the orientation of the listener are used as parameters and the inverses of the rotation and affine transformation matrix are applied to each point.

Knowing the x, y, and z coordinates of a point in listener space, its azimuth and elevation in relation to the listener are calculated with equations

$$\theta(k) = \arctan\left(\frac{y}{x}\right) \tag{8.3}$$

and

$$\varphi(k) = \arctan\left(\frac{z}{\sqrt{x^2+y^2}}\right). \tag{8.4}$$

These angles are fed to the VBAP which calculates the loudspeaker gains for each frequency bin. The output signals are created based on these gains and the diffuseness value set by the user. The mono signal is split to the diffuse part and the non-diffuse part by multiplying it with the diffuseness value ψ and $1-\psi$ respectively. The diffuse part is sent to the output as it is and the non-diffuse part is multiplied with the loudspeaker gains to create a different signal for each loudspeaker. The concept of the extended panning unit processing for the non-diffuse stream is presented in Fig. 8.4.

8.6 B-format reproduction unit

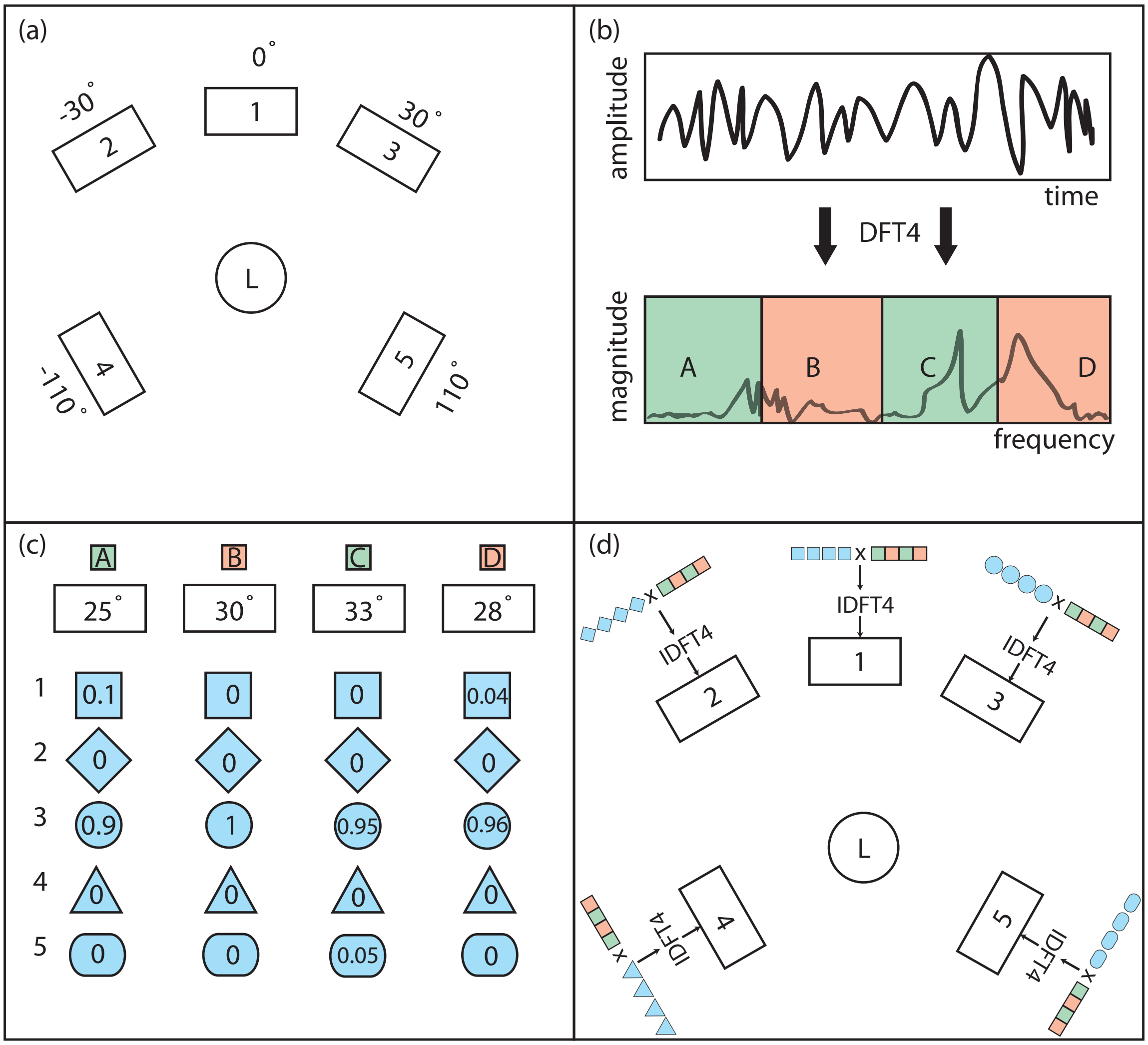
In the B-Format reproduction frontend, a user-defined gain is applied to the signal before transforming it to the frequency domain. After that the energy, the intensity, the diffuseness, and the directions are calculated as described in Chap. 6.3. Fast changes in the direction may cause audible artifacts in the reproduction (Pulkki, 2007). This is not desired, and thus smoothing to slow down the changes in the directions is applied to both the diffuseness and the direction. For the diffuseness analysis both the intensity and the energy are averaged with a first order IIR one-pole filter

$$y[n] = (\alpha-1)x[n] - \alpha y[n-1], \tag{8.5}$$

where α is a coefficient describing the filter. The value of α is made frequency dependent

$$\alpha_k = \frac{N}{\tau f_s}\left(\frac{kf_s}{N}\right)^{\gamma}. \tag{8.6}$$

Here k is the number of the frequency bin, N is the total number of frequency bins, f_s is the sampling rate, and τ and γ are user defined constants. To average the direction of



	A	B	C	D
	25°	30°	33°	28°
1	0.1	0	0	0.04
2	0	0	0	0
3	0.9	1	0.95	0.96
4	0	0	0	0
5	0	0	0.05	0

Figure 8.4: Extended panning processing for the non-diffuse stream explained. All the numerical values do not correspond to the actual system. The initial situation where the listener is listening to 5.0 loudspeaker setup is in (a). There is a sound file that the user wants to be panned to the direction 30°. In (b) the sound file is depicted in time domain and then transformed to the frequency domain with a four point DFT that divides the frequencies to four bands A, B, C, and D. In (c) there are the distributed angles and the loudspeaker gains that would be in the real system computed by VBAP. In (d) the final loudspeaker signals are created.

arrival, the gains calculated in the non-diffuse synthesis are averaged with the filter from Eq. 8.5.

In the synthesis, the virtual microphone coefficients are applied to the original B-format signal and a separate signal for each loudspeaker is created. A copy of these signals is sent to both diffuse and non-diffuse processing as described in Chap. 6.4.1. In the diffuse processing, the virtual microphone correction is applied to the signal and the result is sent to the backend.

In the non-diffuse synthesis, the gains for different frequency bands are calculated with VBAP and they are applied to the different frequencies in a similar way as in the extended panning unit. In addition to the gains, the virtual microphone correction coefficients are applied to the signals.

8.7 Reverberation

The three mono-reverberators mentioned in chapter 7.1.3 were implemented as convolution reverberators. This means that each input frame is convolved with an impulse response measured from a real room. The convolution is explained in the section 4.7. In other words, the value of the signal at every time instant is thought as an impulse and it is smeared in time to be like the impulse response. Because the convolution makes the frame longer than it originally is, the output of the reverberator is saved to a ring buffer so the later samples of the result can be accessed in the next frame. The convolution reverb is computationally intensive and the required amount of processing depends on the length of the impulse response used. The computation of the convolution was made less intensive by performing it in the frequency domain. In the implementation, impulse responses measured from a real room were used.

Another way to implement the reverberation would be the use of so called feedback delay network that has been studied by Jot and Chaigne (1991). This provides a computationally less intensive implementation of the reverberation. In practice, finding the right parameters has proved to be difficult. An implementation of this was programmed as a part of the work but there was not enough time to find good parameters so this was not used in the final system.

8.8 Binaural implementation

Binaural processing is explained in chapter 6.5. It is not reasonable to measure HRTFs from so many directions that a complete soundscape could be rendered in a way that every direction would have its own function. A solution that was used is to have a limited number of HRTFs measured and use these as virtual loudspeakers. If the sound is coming from a direction that has its own HRTF, it is filtered with that and the sound is played back. If the direction has no own HRTF, a superimposition of different HRTFs is used. The azimuth, the elevation, and the information of the locations where the HRTFs have been measured are used together with the VBAP to decide which HRTFs are used and with what weights. A system based on the same idea has been introduced by Noistering et al. (2003).

As was described in Chap. 6.5, the quality of the binaural processing can be improved with the use of head tracking. Using head tracking causes changes to the extended panning unit and the B-format reproduction unit. In the extended panning unit, the orientation of the listener is changed according to the orientation of the head. The azimuth is changed by subtraction

$$\theta_2 = \theta_1 - \alpha, \tag{8.7}$$

where θ_1 is the orientation of the listener, θ_2 is the new orientation, and α is the head orientation. Changing the tilt and elevation has an effect on both azimuth and elevation. For the rest of the calculations, the coordinate system is changed from spherical to cartesian with equations

$$x = \cos\theta\cos\varphi \tag{8.8a}$$
$$y = \sin\theta\cos\varphi \tag{8.8b}$$
$$z = \sin\varphi, \tag{8.8c}$$

where θ is the azimuth and φ the elevation. Changes by the head elevation are computed with

$$x_2 = \cos\left(-\beta + \arctan\left(\frac{z}{x}\right)\right)\sqrt{1-y^2} \tag{8.9a}$$
$$y_2 = y \tag{8.9b}$$
$$z_2 = \cos\left(-\frac{\pi}{2} - \beta + \arctan\left(\frac{z}{x}\right)\right)\sqrt{1-y^2}, \tag{8.9c}$$

where β is the head elevation angle. Changes by the head tilt are calculated with

$$x_3 = x_2 \tag{8.10a}$$
$$y_3 = \cos\left(-\gamma + \arctan\left(\frac{z_2}{y_2}\right)\right)\sqrt{1-x_2^2} \tag{8.10b}$$
$$z_3 = \cos\left(-\frac{\pi}{2} - \gamma + \arctan\left(\frac{z_2}{y_2}\right)\right)\sqrt{1-x_2^2}, \tag{8.10c}$$

where γ is the head elevation angle. The cartesian coordinates are transformed back to spherical coordinates with

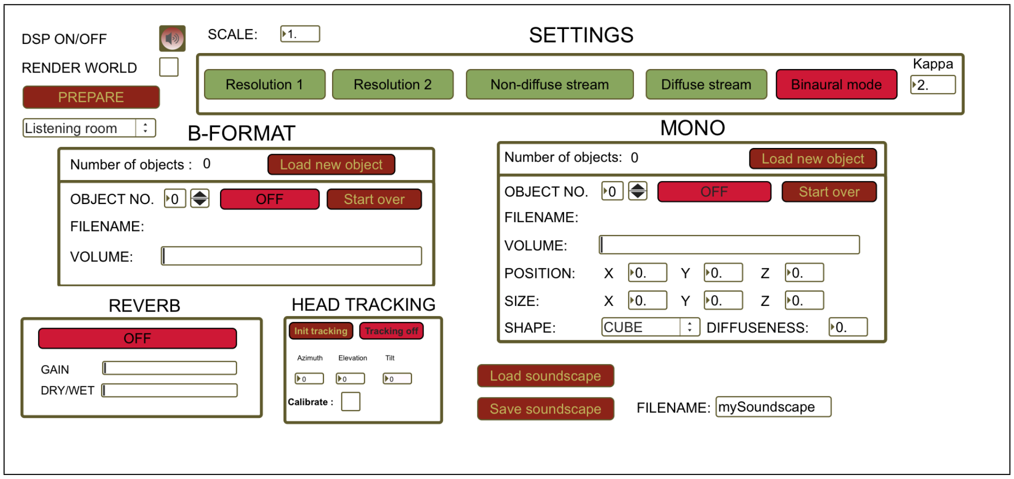
$$\theta = \arctan\left(\frac{y_3}{x_3}\right) \tag{8.11a}$$
$$\varphi = \arctan\left(\frac{z_3}{\sqrt{x_3^2 + y_3^2}}\right) \tag{8.11b}$$

In the B-format playback, the analyzed directions are changed according to the previous equations. In addition to this, the virtual loudspeakers are rotated so that the front speaker is still in front of the listener. This means that the virtual microphone signals are calculated again with new loudspeaker locations. The new location angles are derived from the old ones using the same equations as were used for the analyzed directions.

8.9 User interface

An user interface (UI) was created using MAX6 programming language. It is depicted in Fig. 8.5. In the top left corner, it has the controls for turning on and off the processing, rendering the graphics, and loading a new loudspeaker setup to the system. Other controls on the top of the UI are for turning on and off different resolutions, turning on and off the diffuse and the non-diffuse streams, for applying the binaural mode, and changing the directional pattern of the virtual microphones.

In the middle there are controls for mono and B-format objects. It is possible to browse through the objects to get control on a specific object. In the bottom there are the controls for the reverberation unit and the head tracking. The last thing in the bottom, are the controls that allow saving and loading soundscapes. This is useful when there are multiple sound objects present with different locations, volumes, and other parameters, and the same soundscape is going to be used again later on.



SCALE: 1.

Kappa: 2.

Listening room

B-FORMAT
Number of objects : 0
OBJECT NO. 0
FILENAME:
VOLUME:

MONO
Number of objects: 0
OBJECT NO. 0
FILENAME:
VOLUME:
POSITION: X 0. Y 0. Z 0.
SIZE: X 0. Y 0. Z 0.
SHAPE: CUBE DIFFUSENESS: 0.

REVERB
GAIN
DRY/WET

HEAD TRACKING
Azimuth 0 Elevation 0 Tilt 0
Calibrate : □

FILENAME: mySoundscape

Figure 8.5: User interface.

Chapter 9

Discussion

In the preceding chapters, the theory behind the Directional Audio Coding, the technique itself, its virtual-world version, and how the actual system was implemented have been explained. In this chapter, the potential uses of the system, small things that could be done to enhance the performance of the system, and further improvements are discussed.

The programming was carried out using C as the main programming language. As the programming advanced, the program grew rather big and the current version contains approximately 10000 lines of code. As the C language has no capabilities to perform object-oriented programming, the structure of the program has become difficult to present in a simple and easily understandable form. In retrospect, it would have been beneficial to do the programming using C++ or some other C-based objective language. This would have allowed the use of objects and clarified the structure while it would still have been possible to use pure C for the computationally intensive parts that benefit from the use of the language. On the other hand, doing this would have slowed down the process because the author's skills in C++ are not on the same level as in C.

The first use of the system is the demonstration of the capabilities of the DirAC and its further enhancements. The system has the previous separate implementations in one package. These previous implementations include the standard implementation, the binaural implementation, the virtual-world implementation, and the projection implementation. This is useful when running demonstrations at the university. There is no longer need to change between the different programs if different things are to be demonstrated. To make the demonstrations even better, there is still a need for readymade scenarios and applications that focus on the best aspects of the system.

The virtual-world version of DirAC has the intention to be like a real game audio engine. For this reason, the most suitable use for it would be in games. If the engine would be used in a game, the one of the bigger things missing is the Application Program Interface (API). The API needs to implement the communication between the audio engine and the rest of the game, including graphics and physics engines. As the program is now implemented as an external object for Max6, there is a lot of work to do to make it run as a standalone application and communicate with the audio peripherals of the computer. Additionally, in a real game, the resources allocated for the audio engine are very limited and the efficiency of the program would need to be improved.

In the current implementation, the performance has already been enhanced by using parallel programming on some parts and by using optimized vector libraries to perform most of the calculations that could be vectorized. As a further improvement parallelism could be used more. There are still functions that could be modified in a way that they could be run in parallel with others. In addition to this, the object transformations could be done using quaternions. Especially when multiple transformations are performed in a chain the quaternions become more efficient than their matrix counterparts. If this would significantly increase the performance of the system remains to be studied in the future. When the shapes of the objects remain quite simple, I think it would be efficient to transform only a few points that define the shape with a transform and express the rest of the points in relation to these. This way the rest of the transformation could be done as simple additions and subtractions instead of multiplications. For example, from a cube, only the corners are required, and from a sphere, only the center point. Verifying that this would work requires further research.

I think that another thing which has prevented a major breakthrough for the three-dimensional audio for games is the fact that multichannel loudspeaker setups are not found at every home. From a marketing perspective, it is not worth spending time on a feature that will not be available for everyone who buys a game. Ensuring that everyone has the capability to playback the three-dimensional sound scene would make spatial sound more appealing for the game studios. In my opinion it is highly unlikely that there will be a day when everybody will have more than two loudspeaker in their living room. On the other hand, even nowadays almost everybody has some kind of headphones at their home. And if not, they are not too expensive to get. This suggests that rendering the spatial audio scene through headphones is the way that is going to make itself present in the games and other media too. As was stated earlier in the thesis, HRTFs are required to do this. Problems arise from the fact that they are personal and everyone needs their own individual functions. Two possible ways that have been used to create HRTF pairs are measuring them in an anechoic chamber or modeling them mathematically. The mathematical models do not seem to be there yet and measuring requires special equipment and spaces that are not available or affordable for everyone. These methods and it is possible that in the near future there will be a breakthrough that will allow easy creation of HRTFs. Because there is not a method that would be good enough yet, it can be, that the right way to obtain the functions is neither of these.

Use of test tones and HRTFs calibrated with these has not been studied too much, but I personally think that it provides an interesting alternative for creating the functions. This way it would be possible to obtain HRTFs that are specific for the user, for the headphones used, and the room that the listener is in. As there has not been too much research related to this, it remains unclear if it would be possible to simplify the HRTFs ideally to a single parameter that could be changed in order to tune the filters and still get a quality that is acceptable. In my opinion, the use of a single HRTF that describes the difference between the two ears and modeling it with a delay and a filter would be one possible way to do this.

By playing a sine on a certain frequency and controlling the gain for that frequency could be used to find a value that makes sound appear to come from a certain direction. This procedure could then be extended to exponential sweeps to go through all the frequencies.

Chapter 10

Conclusion

The aim for this thesis was to implement a new architecture for the virtual-world Directional Audio Coding (DirAC). This architecture was supposed to further enhance the performance of the system and in addition the programmed system was to be implemented in a way that it would create an easy-to-use platform for creating demonstrations. Creating more demonstrations with relatively small effort would enable different ideas related to the system be tried out easily and make the system better for demonstration use. Ultimately this will lead to more interest in the system and this way to further development opportunities.

DirAC is based on the knowledge about the human hearing and different signal processing techniques. To make understanding of the system possible the basic physical phenomena behind the sound, how the human hearing works and the required signal processing methods were thoroughly explained in the first part of the thesis. After this the DirAC itself and the virtual world variant of it were described based on the published research articles.

In the last part of the thesis the implementation of the system was discussed in more detail. Topics of this section included the used programming languages and techniques and a review of the virtual-world system's implementation. After the implementation details the choices that were made, were discussed. In addition to this discussion things that would make the commercialization of this kind of system were talked about.

The immediate future work with this this system is the creation of different demonstrations that show how well the system works and what are the best attributes it has. Furthermore, there is still work to be done related to the performance of the system and rendering the sounds via headphones.

Bibliography

E. Benjamin and T. Chen. The Native B-Format Microphone. In *Audio Engineering Society Convention 119*, 10 2005.

M. Bouéri and C. Kyriakakis. Audio Signal Decorrelation Based on Critical Band Based Approach. In *Proceedings of the AES 117th Convention*, 2004.

Cycling74. Max/MSP, 4 2013. URL http://cycling74.com. Last checked: 22.05.2013.

J. Eargle. *The Microphone Book: From mono to stereo to surround-a guide to microphone design and application*. Focal Press, 2012.

F. Fahy. *Foundations of Engineering Acoustics*. Academic Press, 2001.

B. Glasberg and B. Moore. Derivation of auditory filter shapes from notched noise data. *Hearing Research*, 47(1-2):103–138, August 1990.

W. Grantham. Spatial Hearing and Related Phenomena. In B. C. J. Moore, editor, *Hearing*, pages 297–345. Academic Press, 2nd edition, 1995.

J.-M. Jot and A. Chaigne. Digital Delay Networks for Designing Artificial Reverberators. In *Audio Engineering Society Convention 90*, 2 1991.

M. Karjalainen. *Kommunikaatioakustiikka*, volume 2nd. Helsinki University of Technology, 2009.

G. S. Kendall. The Decorrelation of Audio Signals and Its Impact on Spatial Imagery. *Computer Music Journal*, 19(4):71–87, 1995.

M.-V. Laitinen. Binaural Reproduction for Directional Audio Coding. Master's thesis, Helsinki University of Technology, May 2008.

M.-V. Laitinen and V. Pulkki. Binaural Reproduction for Directional Audio Coding. In *Proceedings of the IEEE Workshop on Applications of Signal Process- ing to Audio and Acoustics*. IEEE, October 2009.

M.-V. Laitinen, F. Kuech, S. Disch, and V. Pulkki. Reproducing Applause Type Signals with Directional Audio Coding. *Journal of the Audio Engineering Society*, 59(1/2):29–43, January/February 2011.

M.-V. Laitinen, T. Pihlajamäki, C. Erkut, and V. Pulkki. Parametric time-frequency representation of spatial sound in virtual-worlds. *ACM transactions on applied perception*, 9(2), June 2012.

B. Moore. *Cochlear Hearing Loss : Physiological, Psychological and Technical Issues*. Wiley, 2nd edition, February 2008.

B. Mulgrew, P. Grant, and J. Thompson. *Digital Signal Processing: Concepts and Applications*. Macmillan press ltd, 1999.

M. Noistering, A. Sontacchi, T. Musil, and R. Holdrich. A 3D Ambisonic Based Binaural Sound Reproduction System. In *Audio Engineering Society Conference: 24th International Conference: Multichannel Audio, The New Reality*, 6 2003.

T. Pihlajamäki. Multi-resolution Short-time Fourier Transform Implementation of Directional Audio Coding. Master's thesis, Aalto University, 2008.

T. Pihlajamaki and V. Pulkki. Low Delay Directional Audio Coding for Real-Time Human-Computer Interaction. In *Proceedings of the AES 130th Interna- tional Convention*, May 2011.

T. Pihlajamäki and V. Pulkki. Projecting Simulated or Recorded Spatial Sound onto 3D-Surfaces. In *Proceedings of the AES 45th International Confer- ence*, March 2012.

T. Pihlajamaki, M.-V. Laitinen, and V. Pulkki. Modular Architechture for Virtual-World parametric Spatial Audio Synthesis. In *Proceedings of the AES 49th International Conference*, February 2013.

V. Pulkki. Virtual Sound Source Positioning Using Vector Base Amplitude Panning. *Journal of the Audio Engineering Society*, 45(6):456–466, June 1997.

V. Pulkki. Spatial Sound Reproduction with Directional Audio Coding. *Journal of the Audio Engineering Society*, 55(6):503–516, June 2007.

V. Pulkki, M.-V. Laitinen, and C. Erkut. Efficient Spatial Sound Synthesis for Virtual Worlds. In *Audio Engineering Society Conference: 35th International Conference: Audio for Games*, 2 2009.

Rayleigh. On our perception of sound direction. *Philosophical Magazine*, 13(74), 1907.

T. Rossing, R. Moore, and P. Wheeler. *The Science of Sound*, volume 3rd. Addison Wesley, 2002.

J. O. Smith. Short time fourier transform, 2011. URL `https://ccrma.stanford.edu/~jos/sasp/Short_Time_Fourier_Transform.html`. Last checked: 22.05.2013.

J. Vilkamo. Spatial Sound Reproduction with Frequency Band Processing of B-format Audio Signals. Master's thesis, Helsinki University of Technology, May 2008.

J. Vilkamo, T. Lokki, and V. Pulkki. Directional Audio Coding: Virtual Microphone-Baseed Synthesis and Subjective Evaluation. *Journal of the Audio Engineering Society*, 57(9):709–724, September 2009.

H. Wallach, E. Newman, and M. Rosenzweig. The precedence effect in sound localization. *American Journal of Psychology*, 62(3):315–336, July 1949.

Wikipedia. Anatomy of the human ear, 2013a. URL `http://commons.wikimedia.org/wiki/File:Anatomy_of_the_Human_Ear.svg`. Last checked: 22.05.2013.

Wikipedia. Depiction of overlap-add algorithm, 2013b. URL http://en.wikipedia.org/wiki/File:Depiction_of_overlap-add_algorithm.png. Last checked: 22.05.2013.

W. Yost. *Fundamentals of Hearing: An Introduction*. Academic Press, 1994.

P. Zador. Asymptotic quantization error of continuous signals and the quantization dimension. *Information Theory, IEEE Transactions on*, 28(2):139–149, 1982.

E. Zwicker and E. Terhardt. Analytical Expressions for critical-band rate and critical bandwidth as a function of frequency. *The Journal of Acoustical Society America*, 68(5): 1523–1525, May 1980.