KTH Computer Science
and Communication

Spatial Impulse Response Rendering of digital waveguide mesh room acoustic simulations

Stefan Enroth

Handledare: Damian Murphy
University of York, Department of Electronics

Godkänt den ..... Examinator: .....
(namn i klartext)

Examensarbete i Elektroakustik

Master of Science Thesis in Electro Acoustics

KTH - Skolan för Datavetenskap och kommunikation (CSC)
Avdelningen för Tal, musik och hörsel
100 44 Stockholm

KTH Computer Science and Communication

Examensarbete i Elektroakustik

Spatial Impulse Response Rendering of digital waveguide mesh room acoustic simulations

Stefan Enroth

Examinator: Sten Ternström

Handledare: Damian Murphy

Sammanfattning

Digital Waveguide Mesh (DWM) är en numerisk simuleringsteknik som kan användas för att modellera hur en akustisk våg fortplantar sig genom ett slutet system. Denna modellering resulterar i ett impulssvar för det modellerade utrymmet, och för de valda positionerna för ljudkälla och mottagare. *RoomWeaver* är ett PC-program som används för att modellera akustiken i utrymmen genom att använda DWM. Detta projekt rör hur man kan auralisera det resulterande impulssvaret.

Metoden som använts är Spatial Impulse Response Rendering (SIRR), som är en teknik för att leverera ett impulssvar via ett flerkanaligt högtalarsystem. Detta åstadkoms genom att först analysera riktningen och hur diffust ljudfältet är inom tids- och frekvensgränserna för vad människor kan uppfatta. Sedan används denna information för att syntetisera ett flerkanaligt impulssvar. Både implementationen av analysen och av syntesen utnyttjar Short Time Fourier Transform (STFT), vilket delar upp signalen i tids- och frekvenskomponenter.

KTH Computer Science and Communication

Master of Science Thesis in Electroacoustics

Spatial Impulse Response Rendering of digital waveguide mesh room acoustic simulations

Stefan Enroth

Examiner: Sten Ternström

Supervisor: Damian Murphy

Abstract

The Digital Waveguide Mesh (DWM) is a numerical simulation method that can be used to model acoustic wave propagation in an enclosed system. The output from a DWM model is the Room Impulse Response (RIR) of the modelled space for given source/receiver locations. *RoomWeaver* is a PC application used to model the acoustics of spaces using the DWM. This project deals with how to auralize the resulting RIR.

The method used is Spatial Impulse Response Rendering (SIRR), which is a technique for rendering a RIR to a multichannel speaker system. This is done by first analyzing the direction and diffuseness of the sound field within the time and frequency resolution of human perception, and then to use that information to synthesize a multichannel RIR. Both the analysis and synthesis is implemented using a Short Time Fourier Transform scheme, which splits the signal into time and frequency components.

Contents

List of figures

1. Introduction

Advancements in spatial audio have improved the possibilities of reproducing the experience of listening to a certain event. Spatial audio has evolved from simple stereo to techniques aiming to give a more enveloping experience. This opens up possibilities of coming closer to the goal of giving the listener the same experience as if he or she had been listening to the actual event live.

In a similar manner room acoustics models are becoming more advanced and are able to more accurately simulate the acoustics of a physical space. The simulations result in an impulse response that describes the acoustic properties of a room or structure. With this impulse response it is possible to apply the acoustic effect of the structure to any sound. This can be done with an anechoic sound to simulate what that sound would sound like in the modelled structure.

This project deals with the task of combining these two areas. Once the impulse response of the modelled structure has been synthesised with an appropriate modelling tool it has to be presented to the listener in a suitable way. The most basic output is an omni-directional impulse response which does give a sense of the reverberation properties of the modelled space, but lack any information about the direction of the sound. More advanced outputs from the modelling tools, as well as more advanced ways of reproducing the output are required.

This project will look at an acoustical research application called RoomWeaver, which has been developed at the University of York. It uses a technique called Digital Waveguide Mesh (DWM) in order to simulate the acoustics of a modelled 2-D or 3-D structure. For the purpose of surround sound playback it is possible to use a certain set of receivers in RoomWeaver to get a first order B-format signal as used in Ambisonics. This B-format signal will be used the project as the base for a technique called Spatial Impulse Response Rendering (SIRR). The technique manipulates an omni-directional impulse response in to a multichannel impulse response for a certain speaker setup. The process of doing this consists of first analysing the sound field of the original impulse response, and then using that information to synthesize a new multichannel response. SIRR has been developed to be used with real world recorded impulse responses, but in this project the technique will be applied to simulated impulse

responses created in RoomWeaver. In summary, this project is looking at one way of improving the quality of the reproduction of the simulated impulse response.

A working SIRR implementation in combination with the RoomWeaver application would allow a user to do the following: First create a 2-D or 3-D model of a chosen structure, and place source and receiver positions within that structure. The next step would be to run the DWM simulation in order to create a B-format impulse response. This impulse response would then be rendered in to a multichannel impulse response for a certain speaker setup using SIRR. Finally an anechoic sound could be convolved with the multichannel impulse response, and the user would be able to listen to what it would sound like in the simulated structure.

The report starts with a general presentation of the project in chapter 2. Following that some of the theories involved are presented in chapter 3, this includes basics about spatial hearing and spatial audio, the Digital Waveguide Mesh and Spatial Impulse Response Rendering. Chapter 4 describes how the SIRR analysis was implemented. In chapter 5 the different ways in which the analysis results were verified are discussed. Chapter 6 explains how a simple SIRR synthesis was implemented. In chapter 7 a few conclusions are drawn from the project and some suggestions for future work are made.

2. Overview of the project

This project deals with the problem of how to reproduce the acoustics of a simulated acoustic space. This can be useful in areas such as reverberation in musical applications or to enable an architect to listen to what a certain structure would sound like. There are several ways of simulating the acoustics of a certain space, and several ways to reproduce the result. The goal of this project is to explore the possibility of using Spatial Impulse Response Rendering to reproduce the output from a Digital Waveguide Mesh based PC application called RoomWeaver. The whole process can be described as follows:

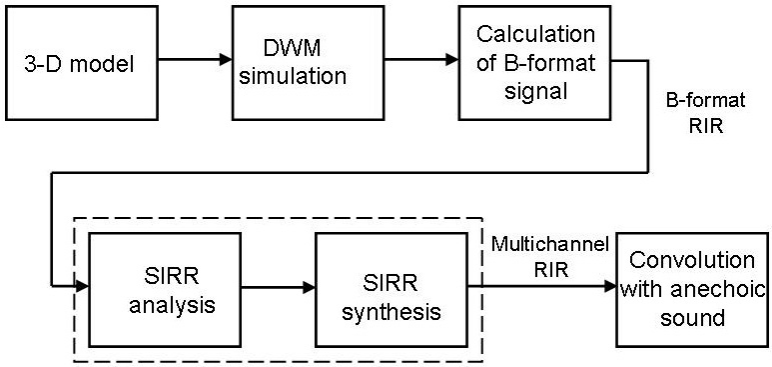

Figure 1: Block diagram of the whole process from a modelled structure to being able to listen to the acoustics. This project deals with the procedures within the dotted line.

First the chosen structure is modelled in RoomWeaver. A crux of receivers is placed at the receiver position in the application, and with the appropriate calculations the result of the simulation is a first order B-format RIR. The B-format signal is then processed with the SIRR technique to produce a multichannel RIR for a certain speaker setup. This includes both the analysis of the B-format signal and then the rendering of a multichannel RIR from the omni-directional one available in the B-format signal. The multichannel RIR can then be convolved with an anechoic sound and played back through the chosen speaker setup. This allows the listener to hear what the sound would sound like in the modelled space.

The main focus of this project is the analysis part of SIRR technique. The analysis has been implemented and the results verified with different types of input. Additionally, a simple version of the synthesis has been implemented to verify that the analysis data can be used to synthesise a multichannel RIR. The project does not aim to complete the entire SIRR process, neither to evaluate the performance of it compared to other available techniques.

3. Background theory

3.1 Spatial hearing

Spatial hearing is based on how the auditory system analyses different spatial cues. These cues are Interaural Time Difference (ITD), Interaural Level Difference (ILD), spectral cues, interaural coherence and head movement [1], [2]. The two most important of these are ITD and ILD.

The ITD is the time difference in when a sound arrives at the ears. Since the ears are separated by a certain distance a sound will have to "travel" slightly longer to one ear than the other, unless the sound source is directly in front or behind the listener. This will cause a time difference between when the sound arrives at each ear. This delay will depend on the angle at which the sound arrives to the head, the largest time difference will be when the sound arrives directly from one side, and there will be no difference if the sound arrives from in front or behind the listener [2].

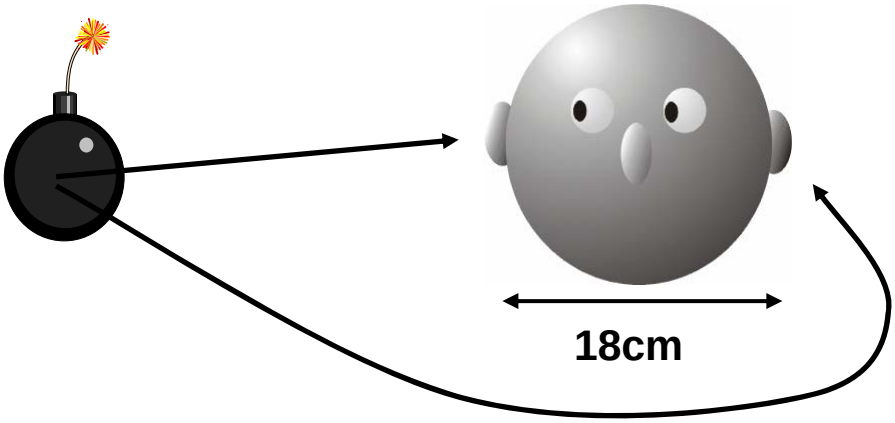

Figure 2: Interaural Time Difference. From [3].

The difference in time also results in a shift of phase which the auditory system can use to determine the timing difference between each ear. However if the phase shift is greater than 180 degrees there is an ambiguity about how the phase has shifted. For low frequency signals a phase shift that large corresponds to relatively large time difference, and can thus be useful for determining the direction. However for high frequency signals a smaller timing difference will correspond to a 180 degree phase shift. This means that this feature of a signal quickly

becomes less useful above a certain frequency; depending on the angle this will be around 700-1500 Hz [2].

Interaural Level Difference is the difference in loudness between the ears. This difference occurs when the sound is partly blocked by the head. Similar to ITD there will not be a difference in level when the sound source is positioned directly in front or behind the listener, and the difference will increase as the source moves to one side of the head with a maximum difference when the sound source is directly to one side of the listener. The difference also depends on the frequency of the signal. Low frequency sound waves will “bend” around the head due to diffraction and this will result in only a small part of the sound being blocked, and therefore a smaller difference in sound level. High frequencies on the other hand do not diffract as much around the head, and the level difference will be large, in some cases as much as 20 dB. This means that ILD is mostly used for high frequency sounds and ITD for low frequency sounds [2].

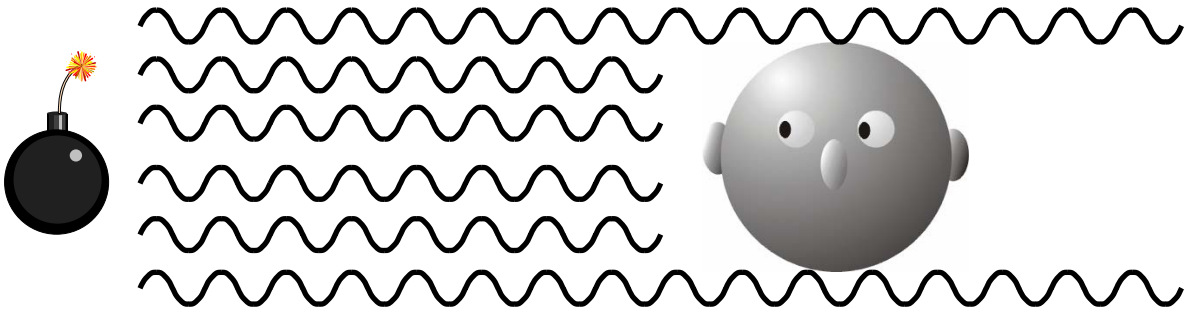

Figure 3: Interaural Level Difference. From [3].

The two cues presented above share some features. They both work based on the positioning of the ears, which has some effect on how useful these cues are for determining the direction of a sound source. For a given difference in timing or level there can be several different possible directions, this is called the cone of confusion [2]. This is the cone that forms such an angle to the ears that any source on the surface of the cone would result in the same ITD.

The third type of spatial hearing cue is spectral cues, which arise due to the shape of the outer ear, called the pinnae. When a sound arrives at the pinnae it will be reflected into the ear canal via all the ridges in the pinnae. Due to the complex and asymmetric look of the pinnae this will result in spectral changes to the sound that depends on the frequency and angle of incidence. The effect of this is that the pinnae, and in fact also the head and upper body, acts

as a direction-dependant filter for incoming sounds. In other words, for a given direction certain frequencies will be amplified while others will be attenuated. This fact can be used by the auditory system to determine from which direction a sound is arriving. Unlike ITD and ILD it is possible to distinguish between sources in front of and behind the listener, and it is also possible to perceive the vertical position of the source. Since it is the spectral changes of the sound that are used in this cue, it is most effective for localizing broadband sounds, and especially sounds containing frequencies above 6 kHz. This is because a broadband sound will be affected in different ways at different frequencies, and high frequencies have a wavelength that is short enough to be strongly affected by the pinnae. [2]

Interaural coherence is how similar the waveforms of the sound arriving at the ears are. It can be used to distinguish between the direct sound and reflections in a reverberant environment according to [5]. Being able to do this is important feature of the auditory system since it is the direction of the direct sound that is of interest when trying to locate a sound source.

All of the above cues are affected by head movements, and they will be affected in different ways depending on where the sound source is located. This can help resolve ambiguities about where the source is located. If for example the head is rotated towards the perceived sound source the ITD and ILD should be reduced until the head is pointed straight at the sound source. If the listener first thinks the source is located in the front arc, but the changes in the cues when rotating the head are unexpected, it could mean that the source is instead located behind the listener.

3.2 Spatial audio

The most basic format for audio reproduction is mono; a single track of audio. This format can convey some spatial information such as reverberation, but it is not possible to reproduce a sense of direction. To convey directional information at least two channels of audio is needed, and this format is called stereo. Stereo reproduction can use two different ways of reproducing a sound source from a certain direction. The first one is timing difference between the two channels, and the second one is level difference. These differences will create the ITD and ILD cues for the listener, which will create a sense of direction of the sound played. It is possible to use only one of these cues to give the sense of direction, but it will require a larger difference. For example if only level difference is used the level

difference has to be larger than it would need to be if both cues were used to give the same sense of direction [4].

Surround sound in the form of 5.1 is becoming more and more common. The name 5.1 means five speaker channels placed around the listener, and a single (.1) low frequency channel. This has also been expanded to 6.1 and 7.1 with six and seven channels of audio respectively. The principle for all of them is the same as with stereo; timing and level differences between channels that are adjacent in the speaker setup create spatial cues for the listener.

One form of spatial audio that does not follow the same principle is Ambisonics [6], which has a couple of major differences in comparison to the previously presented forms of spatial audio. Ambisonics is split into an encoding and a decoding stage, which are independent of each other. The encoding stage involves decomposing the sound field into the n^{th} order spherical harmonics. A first order signal consists of an omni-directional pressure component, called the W channel, and three orthogonal bi-polar velocity components called X, Y and Z. Together these are called a first order B-format signal, and describe the sound field at a certain point in space. At this point the B-format signal describes both the pressure and velocity of the sound field in any direction. Once a B-format signal has been encoded it can be decoded to any arbitrary speaker setup using certain decoding equations. These equations are altered according to the speaker positions of the setup being used. Generally all speakers will be active when playing a certain sound.

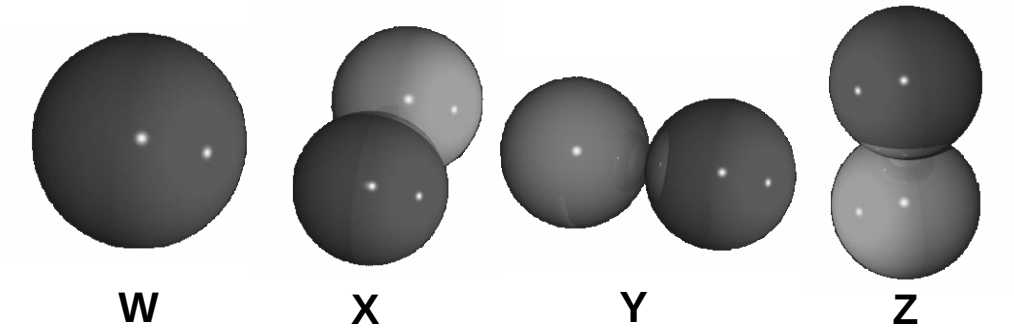

Figure 4: The spherical harmonics of a first order B-format signal.

3.3 Room acoustics modelling

The goal of room acoustics modelling is to describe the acoustic effect of a certain room or structure. One way of describing this effect is with a Room Impulse Response (RIR). A RIR is what one would hear at a certain point in a room if a perfect impulse were played at another point in the room. The impulse response describes at what point in time, and at what amplitude, the direct sound and the following reflections arrive. This will normally be first the direct sound, followed by reflections with decreasing amplitude. This RIR will be unique for the room and the positions of the source and listener.

One way of calculating this RIR is to use geometric acoustic models. There are different types of geometric models but they all work under the assumption that the sound travels in a straight line. This means that the sound can be dealt with in the same manner as rays of light. A reflected sound wave will therefore have the same angle of reflection as the angle of incidence, in the same way as a ray of light reflecting in a mirror. The absorption that occurs when the sound is reflected can be modelled by subtracting a percentage of the ray's energy at each reflection. The main drawback with this technique is that the assumption that sound travels in a straight line holds only when the wavelength is short compared to the surfaces involved. This means that it is generally only valid for higher frequencies and large spaces. Additionally, wave phenomena such as diffraction and interference cannot be modelled using these methods.

Another technique for acoustic modelling is wave based acoustic models. As with geometric models there are different methods, but what they all have in common is that they are based on the solution of the wave equation. This means that the model will treat the sound as a wave, and will be able to model effects such as diffraction. One way of doing wave based acoustic simulations is to use the Digital Waveguide Mesh that will be explained in the next part of this chapter.

3.4 Digital Waveguide Mesh

Before explaining what the Digital Waveguide Mesh is, it is necessary to explain the digital waveguide. This is a numerical simulation of wave propagation through one dimension. It is based on the d'Alembert solution to the one-dimensional wave equation [7]. The solution

describes the wave propagation through a system as two waves travelling in opposite directions. If this solution is made discrete in time and space, it can be implemented as two digital delay lines representing the left- and right-going wave components respectively.

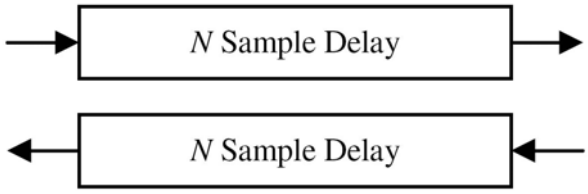

Figure 5: The two digital delay lines that form the basis for all digital waveguide models. From [9].

The implementation of digital delay lines assumes that the system is linear and commutative, which allows all the losses that occur as the wave travels through the system to be applied as one big loss at the end. A simple one-dimensional system like this can be used to model several different things. Among the things that have been modelled there is the human vocal tract as well as string and wind instruments [7].

Several pairs of delay lines can be connected to each other with scattering junctions. This means that a signal will propagate through the connected delay line elements. Connecting several delay lines together with scattering junctions in different ways results in a digital waveguide network [7]. As an example, six different string models could be connected to each other, simulating the strings of a guitar. Plucking one string will produce small vibrations in the other strings as the vibration propagates through the bridge of the guitar.

When a digital waveguide network consists of a two- or three-dimensional grid structure it is called a *digital waveguide mesh* (DWM) [7]. A digital waveguide mesh can be used to simulate an acoustic space with a grid of scattering junctions connected by delay lines in a shape that is analogous to the actual modelled space [8]. There is a difference between the digital waveguide mesh and other digital waveguide networks, which is that the mesh resembles the modelled object in its structure. The grids can have different layouts, called topologies, for example rectilinear or triangular [8]. The topology of the mesh will affect the performance of the simulation. Every scattering junction in the grid is connected to its neighbours with bi-directional delay lines, in the same way as the one-dimensional case. So if the system is excited at one point the signal will propagate to the neighbours of that scattering junction, and after that to their neighbours and so on.

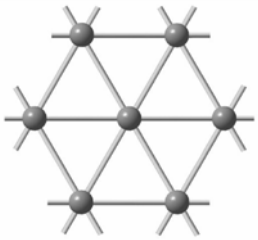

Figure 6: Example of a 2-D triangular DWM mesh. From [8].

RoomWeaver [8] is an acoustics modelling application, developed at the University of York, which uses a DWM to simulate a space as described above. RoomWeaver provides the user with a graphical user interface to create a virtual space. The space is created as a 2D or 3D model within the application. The surfaces in the model can be defined with different properties, such as absorption and diffusion. Within the space both sources and receivers can be placed.

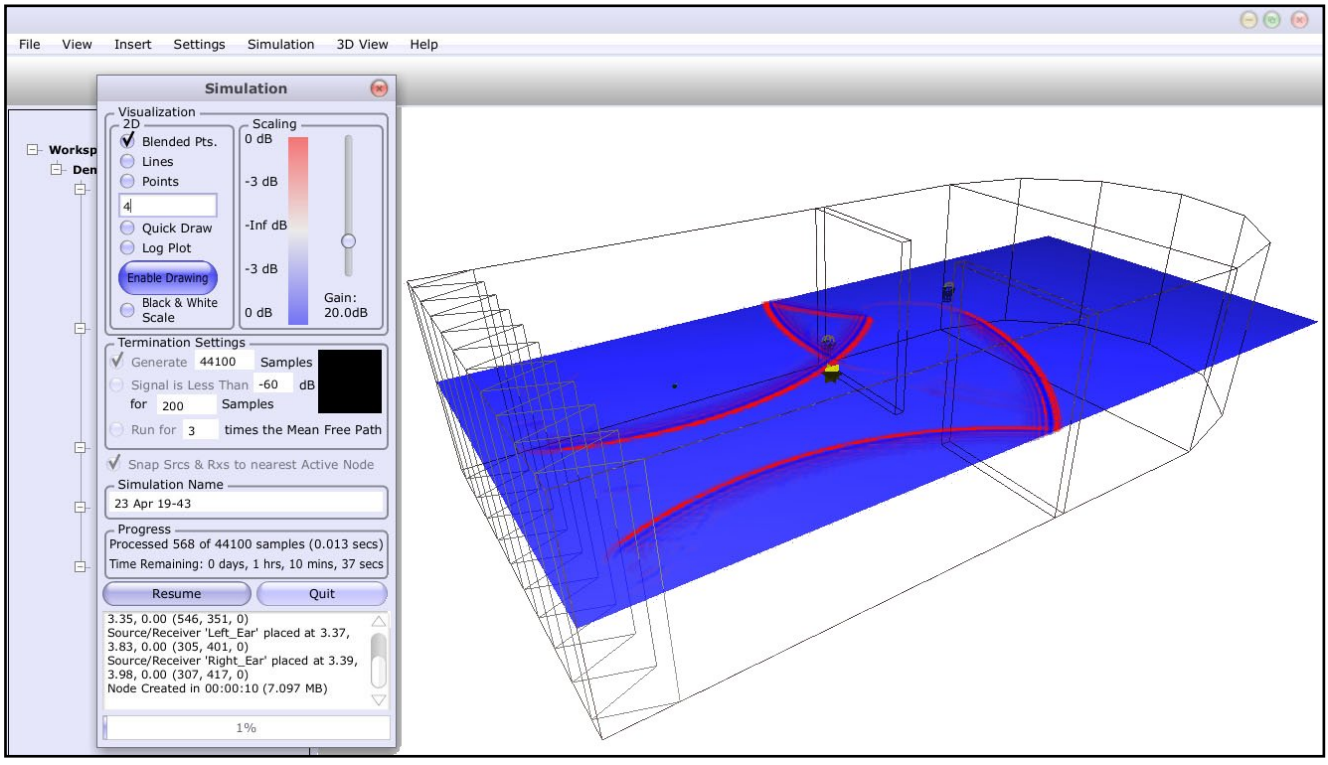

Figure 7: Screenshot of the RoomWeaver application. From [9].

3.5 Spatial encoding from the DWM

From the RoomWeaver application it is possible to extract a B-format signal, as used in Ambisonics, by using a set of seven receivers [9]. These receivers are arranged in a formation with one in the middle and the six others placed a small distance *d* away along each Cartesian axis. The distance *d* determines what frequencies might be captured successfully; higher frequencies require smaller spacing. The middle receiver can be used directly as the W channel, and the X, Y and Z channels can be calculated from the three receivers along each axis. The three receivers are used as three different pairs; the two furthest receivers with spacing 2*d* and the middle one with each axial receiver with spacing *d*. Along one axis in figure 7 this means that the three pairs are: p3 and p2, p3 and p1, and finally p1 and p2. The pair with the greater distance between them is used to capture the lower frequencies and the two other pairs are averaged to get the higher frequencies. [9]

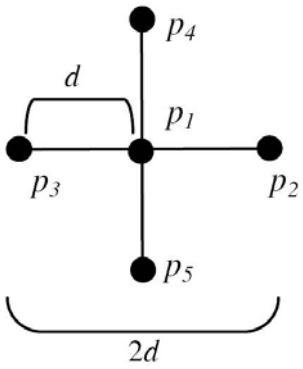

Figure 8: 2-D view of the receivers used to obtain the first order B-format signal. From [9].

3.6 Spatial Impulse Response Rendering

Spatial Impulse Response Rendering (SIRR) [1] is a technique used for reproducing room acoustics over a multichannel loudspeaker system. SIRR analyses the direction of arrival and diffuseness of a RIR within small time and frequency bands. The length of the time windows and the width of the frequency bands are perceptually motivated. Using the analysis data, an omni-directional impulse response is manipulated into a multichannel response.

3.6.1 Psychoacoustical background

SIRR does not aim to recreate the sound field of the recording space perfectly from a physical point of view; instead it tries to recreate the same perceptual cues that the listener would get from the recorded sound field. According to psychoacoustical research presented in section 2.1, human sound localization relies on five perceptual cues. These are interaural time difference (ITD), interaural level difference (ILD), monaural spectral cues, interaural coherence (IC) and the effect of head rotation on these.

These cues are analyzed in the auditory system with limited resolution in both time and frequency. The frequency resolution has been determined through the research of the equivalent rectangular bandwidth (ERB) of the auditory filter [2]. The time resolution, or the equivalent rectangular duration, is not as easily defined. The ability to track changes in ITD and ILD and perceive these as a moving sound source is limited to changes of about 2-3 Hz [2]. However much faster fluctuations in ITD can be detected although they are not perceived as a moving sound source, instead they are perceived as a larger or wider sound source. Additionally, the time resolution has to be high enough to account for the precedence effect, where one sound suppresses an immediately following sound. This means that a suitable time resolution is approximately 10 ms [1], [10].

3.6.2 Principle of SIRR

SIRR builds upon a number of assumptions [11]:

1. The direction of arrival of sound will transform into ITD, ILD, and monaural localization cues.
2. Diffuseness of sound will transform into interaural coherence cues.
3. Timbre depends on monaural (time-dependant) spectrum together with ITD, ILD, and IC.
4. When direction of arrival, diffuseness, and spectrum of sound are reproduced within the temporal and spectral resolution of human hearing, the perceptual quality of the spatial reproduction is good.
5. If the perceptual quality of the spatial reproduction of a room response is good, the perceptual quality of the reproduction of sound convolved with the response is also good.

So by recreating the direction and diffuseness of the sound field the localization cues will be created in interaction with the listener's head. This would result in the same spatial cues as the original sound field would have produced.

The SIRR technique has two main parts: first the analysis of the direction and diffuseness of the sound; then the synthesis, where an omni-directional impulse response is rendered to a multichannel speaker system based on the analysis data.

3.6.3 Analysis

The analysis uses a time-frequency processing scheme to create both time windows and within each window different frequency bands. This can be achieved by either using an auditory filter bank or by using the more efficient short-time Fourier transform (STFT). In the binaural cue coding (BCC) algorithm [12], which shares some features with SIRR, both methods have been tested; and the STFT implementation performed equally well as the auditory filter bank solution.

The analysis is based on the concept of sound intensity, which describes how energy is transferred in a sound field. Depending on the sound field, part of the energy is oscillating locally and another part constitutes the net flow of energy. In this case the direction of the instantaneous intensity will vary over time, and the time-averaged intensity will describe the net flow of energy, also called active intensity, in a certain direction. The direction of arrival of the sound is simply the opposite direction to this net flow. The proportion of energy that is oscillating locally can be used to estimate the diffuseness of the sound field, a larger proportion of oscillating energy means a more diffuse sound field. To calculate the proportion of oscillating energy the analysis uses the active intensity and the time averaged energy density of the sound field [1].

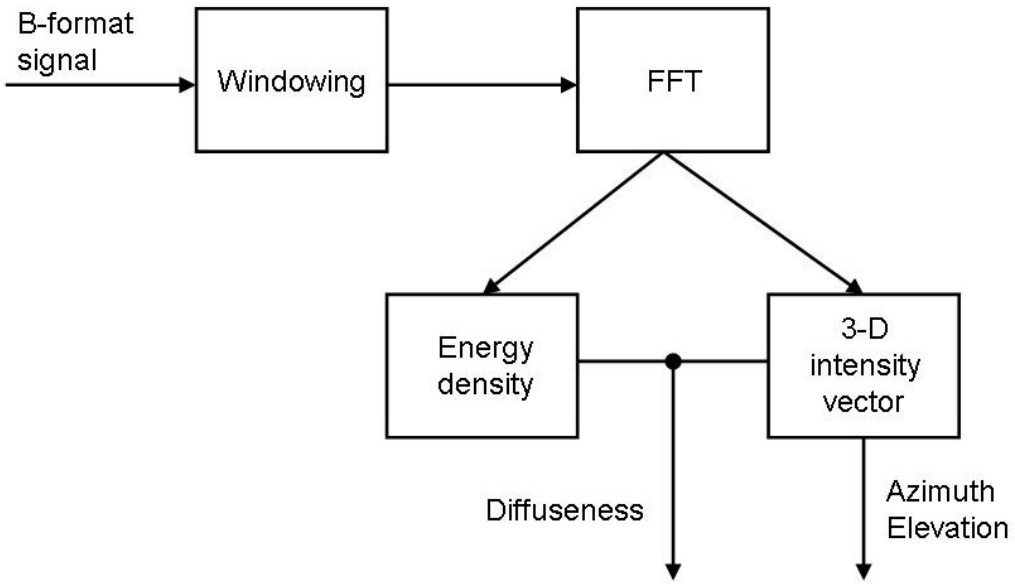

Figure 9: Block diagram of the SIRR analysis.

Here follows a summary of the equations used in the analysis [1]. There are three sets; the first (1-3) dealing with the general case, the second (4-5) with a STFT analysis and the final one (6-8) with a first order B-format input. The instantaneous sound intensity is defined as the product of sound pressure *p* and particle velocity vector ***u***

$$\mathbf{I}(t) = p(t)\mathbf{u}(t) \tag{1}$$

The instantaneous energy density of a sound field can be calculated as

$$E(t) = \frac{1}{2}\rho_0\left[Z_0^{-2}p^2(t) + \mathbf{u}^2(t)\right] \tag{2}$$

where ρ_0 is the mean density of air and Z_0 is the characteristic acoustic impedance of air defined as $Z_0 = \rho_0 c$, with *c* being the speed of sound. Using this, the proportion of oscillating energy can be written as

$$\Psi = 1 - \frac{\left\|\left\langle \mathbf{I}(t)/c \right\rangle\right\|}{\left\langle E(t)\right\rangle} = 1 - \frac{2Z_0\left\|\left\langle p(t)\mathbf{u}(t)\right\rangle\right\|}{\left\langle p^2(t)\right\rangle + Z_0^2\left\langle \mathbf{u}^2(t)\right\rangle} \tag{3}$$

where $\|\bullet\|$ denotes the norm of a vector and $\langle\bullet\rangle$ denotes time averaging. The reason for time averaging is that it is very hard to synthesise a sound field with the same instantaneous properties as the analysed one, but it is possible to recreate the time averaged properties. The value of Ψ varies between 0 and 1, with 1 being a perfectly diffuse sound field.

When the analysis is based on the STFT the single-sided frequency distribution of the active intensity in an analysis window is

$$\mathbf{I}_a(\omega) = 2\,\mathrm{Re}\{P^*(\omega)\mathbf{U}(\omega)\} \tag{4}$$

where $P(\omega)$ and $\mathbf{U}(\omega)$ are the Fourier transforms of sound pressure and particle velocity in the time window. In a similar way to (2) and (3) the single-sided frequency distribution of the diffuseness estimate is given by

$$\Psi = 1 - \frac{\left\|\left\langle \mathbf{I}_a(\omega)/c \right\rangle\right\|}{\left\langle E(\omega)\right\rangle} = 1 - \frac{2Z_0\left\|\mathrm{Re}\{P^*(\omega)\mathbf{U}(\omega)\}\right\|}{\left|P(\omega)\right|^2 + Z_0^2\left|\mathbf{U}(\omega)\right|^2}. \tag{5}$$

In a B-format signal the W signal is proportional to the pressure *p*, and the velocity vector depends on the X, Y and Z signals. The frequency distribution of the active intensity is

$$\mathbf{I}_a(\omega) = \frac{\sqrt{2}}{Z_0}\mathrm{Re}\{W^*(\omega)\mathbf{X}'(\omega)\} \tag{6}$$

where

$$\mathbf{X}'(t) = X(t)\mathbf{e}_x + Y(t)\mathbf{e}_y + Z(t)\mathbf{e}_z \tag{7}$$

with $\mathbf{e}_x$, $\mathbf{e}_y$ and $\mathbf{e}_z$ being the unit vectors along each Cartesian axis. The diffuseness estimate is

$$\Psi(\omega) = 1 - \frac{\sqrt{2}\left\|\mathrm{Re}\{W^*(\omega)\mathbf{X}'(\omega)\}\right\|}{|W(\omega)|^2 + |\mathbf{X}'(\omega)|^2 / 2}. \quad (8)$$

It should be noted that (1), (4) and (6) are similar as well as (3), (5) and (8). It is also important to notice that the requirement to be able to perform this analysis is the pressure and particle velocity of the sound field. This explains why a B-format signal is suitable to perform the analysis on; it contains both the pressure and particle velocity information at a single point in space.

3.6.4 Synthesis

The synthesis [1] starts with an omni-directional impulse response measured in the same position where the analysis was performed. For this purpose the W channel from a B-format signal, used in the analysis part, is suitable. The impulse response is processed with the same time and frequency resolution as the analysis, which results in the same time-frequency components that the analysis was performed on. Each component has corresponding azimuth, elevation and diffuseness values from the analysis. The energy of each component is then split into non-diffuse and diffuse parts based on the diffuseness estimate. The non-diffuse part is reproduced as accurately as possible from the analysed direction with a suitable technique, for example by amplitude panning between the loudspeakers. This can be achieved with Vector Base Amplitude Panning (VBAP) which will be described in the next section.

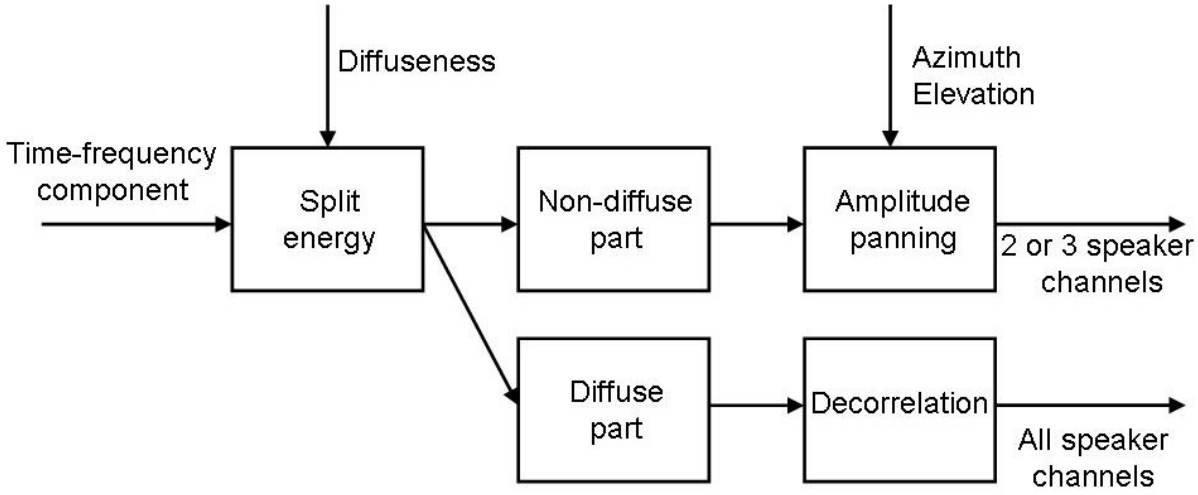

Figure 10: Block diagram of the SIRR synthesis.

The rendering of the diffuse sound is beyond the scope of this project, but for completeness a short summary follows. The diffuse part is reproduced by distributing the energy uniformly around the listener in a decorrelated form. This can be achieved in several different ways, described in [11]. The one finally used by Merimaa and Pulkki is a hybrid of amplitude panning for low frequencies and phase randomization for high frequencies. With amplitude panning the diffuse sound is not treated any different from the non-diffuse sound. However, due to the diffuseness of the sound, the directional data will vary stochastically as a function of time and frequency. This is then perceived as diffuse sound by the listener. Phase randomization creates random noise for each channel. The magnitude spectrum of this noise is then adjusted in each time-frequency window to match the diffuse signal, thereby replacing the diffuse sound with random noise with similar time-frequency envelope.

3.6.5 Vector Base Amplitude Panning

Vector Base Amplitude Panning [13] is a technique for positioning virtual sound sources around a listener with either a 2-D or 3-D speaker array. For any desired direction the two or three nearest speakers, depending on if it is a 2-D or 3-D speaker array, will be used to create the virtual sound source. This means that a 2-D speaker array will consist of several pairs of speakers, where each speaker will belong to two pairs, one with each neighbouring speaker. In other words any given direction around the listener will be between a single pair of speakers. Only these two speakers will be used to produce the sound, while the other will be inactive. The reason for doing this is that it is a robust and effective way of producing accurate virtual sound sources. The technique works in the same way with a 3-D array, except that instead of being panned between a pair of speakers it will be panned within a triangle of speakers.

When VBAP is used to pan a sound to a certain direction in 2-D the process starts with calculating vectors to all the speakers. The next step is to calculate a vector for the desired direction, and to select the correct pair of speakers. Finally a gain-factor is calculated for each speaker. This is done by projecting the direction vector onto each of the speaker vectors. The length of this projection is the base for the gain factor, which is then normalized to make sure that the sound is perceived equally loud no matter where between the two speakers it is positioned.

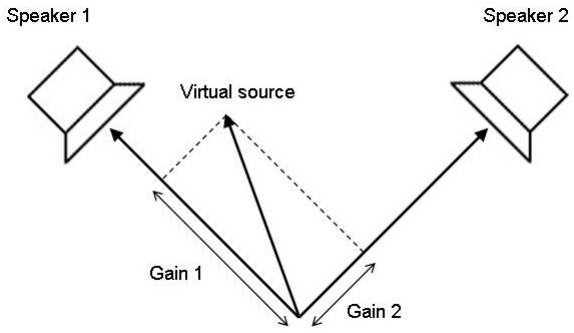

Figure 11: Relationship between the speaker directions and gain factors in VBAP.

The result is that the closer to one of the speakers the virtual source is positioned, the higher the gain factor for that speaker will be. If the source is placed in the same direction as the speaker, all the sound will be produced by that speaker.

4. Implementation of the analysis

The analysis was implemented as a function in MATLAB. It uses 4-channel wave-files (.wav) as input, where the 4 channels are the B-format channels W, X, Y and Z.

The input is windowed with 256 samples long Hann-windows, and each window is zero-padded with 128 samples before and after the window. The windows overlap with half the window length, 128 samples. Before the windowing process the entire input is zero-padded with 128 samples before and 256 samples after the signal. This is necessary because when the input is windowed the first window start at sample 1 and has a maximum at sample 128, and the next window will start at this maximum, so any information in the first 128 samples will be attenuated according to the "slope" of the first window. With the zero-padding the first sample of the real signal will be at the maximum of the first window, and all the following samples will be within two windows.

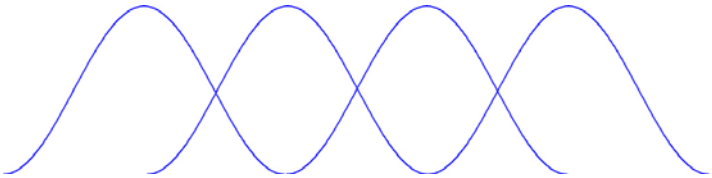

Figure 12: Overlapping Hann-windows.

The result of each window is a 512x4 matrix, where each column is a channel of the B-format signal. A Fast Fourier Transform (FFT) with the length 512 is then applied to each column, which results in four transforms, one for each channel. These are the transforms that are used for the calculations in the analysis.

The active intensity is calculated along each Cartesian coordinate, X, Y and Z. This is done with a slight variation to equation (6) which results in 3 active intensities $\mathbf{I}_x(\omega)$, $\mathbf{I}_y(\omega)$ and $\mathbf{I}_z(\omega)$. The equation used is

$$\mathbf{I}_\beta(\omega) = \frac{\sqrt{2}}{Z_0}\mathrm{Re}\{W^*(\omega)\mathrm{B}(\omega)\} \tag{9}$$

where β is *x*, *y* or *z*, and B is the FFT of the respective channel. The difference from equation (6) is that three active intensities are calculated, one along each Cartesian coordinate, instead of a single 3-D active intensity vector. The reason for this was that it was easier to implement as the next step in the calculations need these three vectors. If the 3-D active intensity vector is calculated it has to be projected on to a unit vector for each axis before the next step.

When the three active intensities have been calculated the azimuth and elevation are calculated with simple trigonometry [1].

$$azimuth(\omega) = \tan^{-1}\left[\frac{-I_y(\omega)}{-I_x(\omega)}\right] \tag{10}$$

$$elevation(\omega) = \tan^{-1}\left[\frac{-I_z(\omega)}{\sqrt{-I_x^2(\omega) + -I_y^2(\omega)}}\right] \tag{11}$$

For the diffuseness estimate the three active intensities are combined into a single active intensity vector, and then equation (8) is used to calculate the diffuseness estimate.

These calculations are performed in each time window. The calculations are only performed on the first 257 points of the FFT transform due to symmetry. This means that the result from the analysis will be three values: azimuth, elevation and diffuseness. Each of these values will be a 257xN matrix where N is the number of time windows in the input signal.

The final step of the analysis is to smooth the output values according to the Equivalent Rectangular Bandwidth (ERB) scale [2]. This approximates the resolution of the human auditory system with rectangular filters with a width that depends on the centre frequency. An equation that describes this is

$$ERB_N = 24{,}7(4{,}37F + 1) \tag{12}$$

where ERB_N is the width in Hz and *F* is the centre frequency in kHz. A vector of length 257 is created with frequencies that correspond to the components of the FFT. For each of these

centre frequencies a width is calculated, and then the value of the FFT components are averaged with a window of that width.

5. Verification of analysis results

The results from the analysis need to be verified to ensure that it is working as intended. To do this various inputs with known direction, and in one case diffuseness, were analysed. First impulse and sine wave inputs were used, then simple artificially created RIR's and finally a real world RIR.

5.1 Impulse and sine wave inputs

Three types of very simple signals were created in Matlab and encoded as B-format signals according to the following equations [14]:

$$\begin{aligned} W &= input \cdot 0.707 \\ X &= input \cdot \cos A \cdot \cos B \\ Y &= input \cdot \sin A \cdot \cos B \\ Z &= input \cdot \sin B \end{aligned} \tag{13}$$

where A is the desired azimuth and B the elevation. The first type of signal was a single impulse, the second type was two sine waves from different directions and the final type was a set of 10 impulses from different directions. For each type of input a Matlab script was created which would encode the signal in a B-format signal with certain parameters. The impulse would be encoded to a certain direction, and the two sine waves to a selected direction and frequency each. With this set of scripts several different examples were created with different parameters.

Since the direction is chosen when the B-format signal is encoded it is easy to compare the analysed result with the chosen direction to verify if the analysis is working. If there is an error in the analysis, it is also easy to see in what way it is not working, which was useful during the development process. This is especially true for the single impulse which is the most basic input; it was used to see that the direction was estimated correctly. Once it was established that the analysis was working with the impulse the sine waves were tested. These show both that the direction is estimated for the correct frequency component of the FFT, and that the estimates are correct for two different sounds from different directions arriving

simultaneously. The final test with a set of impulses confirms that the analysis is able to estimate different directions within different time windows.

The following figure shows the results of the analysis when the input is two sine waves of 1500 Hz and 4000 Hz, arriving from 45 and -45 degrees. The number of components in frequency has been reduced to give a better overview of the results. Arrows pointing to the right represent sound coming from in front of the listener.

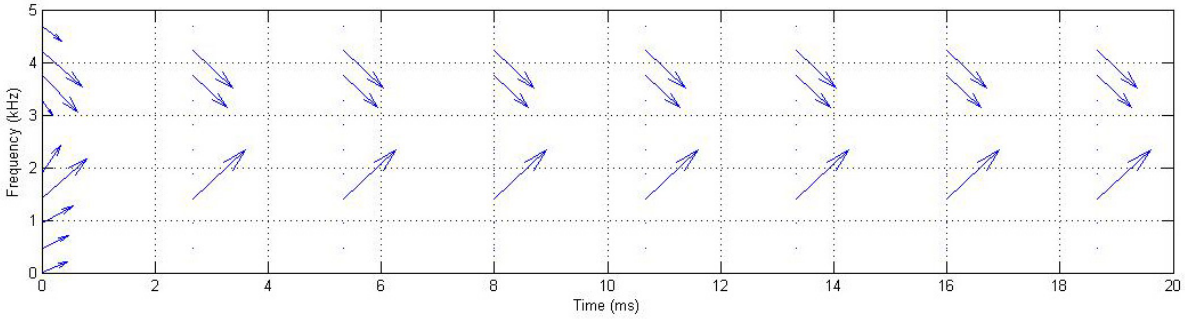

Figure 13: Analysed directions of two sine waves from different directions. Arrows pointing to the right represents sound coming from the front of the listener.

5.2 Simple simulated inputs

The next step was to verify the results with a more complex and more realistic input. This was done with two simulated RIR:s of a 2-D representation of a 4x4 meter room created in RoomWeaver with two different source and receiver positions. These RIR:s have been used when the technique to render a B-format signal from RoomWeaver was developed [14]. In that process a ray-tracer was used to predict the direction and time of the direct sound and first order reflections. This was compared to the result when decoding the B-format signal to 144 virtual speaker channels positioned uniformly around the listener. If for example the direct sound is supposed to arrive from 45 degrees, most of the energy should be in the virtual speaker channels in that direction. Plotting the amplitude of the signal as a function of time and direction results in a plot where it is possible to see from which direction the direct sound and different reflections are arriving. With the results from the ray-tracer in the same plot it is easy to see if the B-format encoding works as intended. These plots were compared to plots of the same kind of the SIRR analysis results to see if the analysis arrived at the same results. For a more detailed plot the time windows in the analysis overlap more than normal.

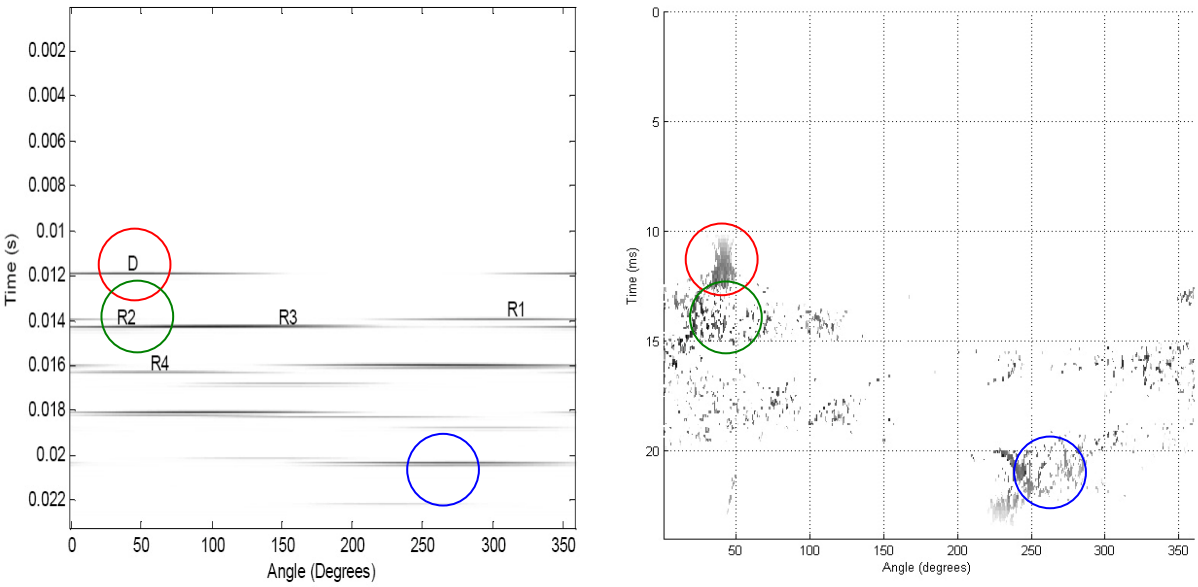

Figure 14: Left: Amplitude of the B-format signal plotted along time and direction predictions. Right: Energy of the W-channel plotted according to SIRR analysis results.

In the plots above some of the similarities between the two plots are circled. Generally they match fairly well. Some things can be noted; first, the analysis provides a more precise direction, which can be seen in the first circle (red), the direct sound. Second, when several reflections arrive roughly at the same time, as with reflections R1, R2 and R3, they will fall within the same time window in the SIRR analysis. Due to the nature of the analysis only one direction estimate will be produced for each frequency component in that time window. This means that unless the reflections cover different frequency bands the resulting estimate will be a single direction. It can be seen in the plot that the estimated direction when the three reflections arrive roughly at the same time varies more than the estimate of the direct sound.

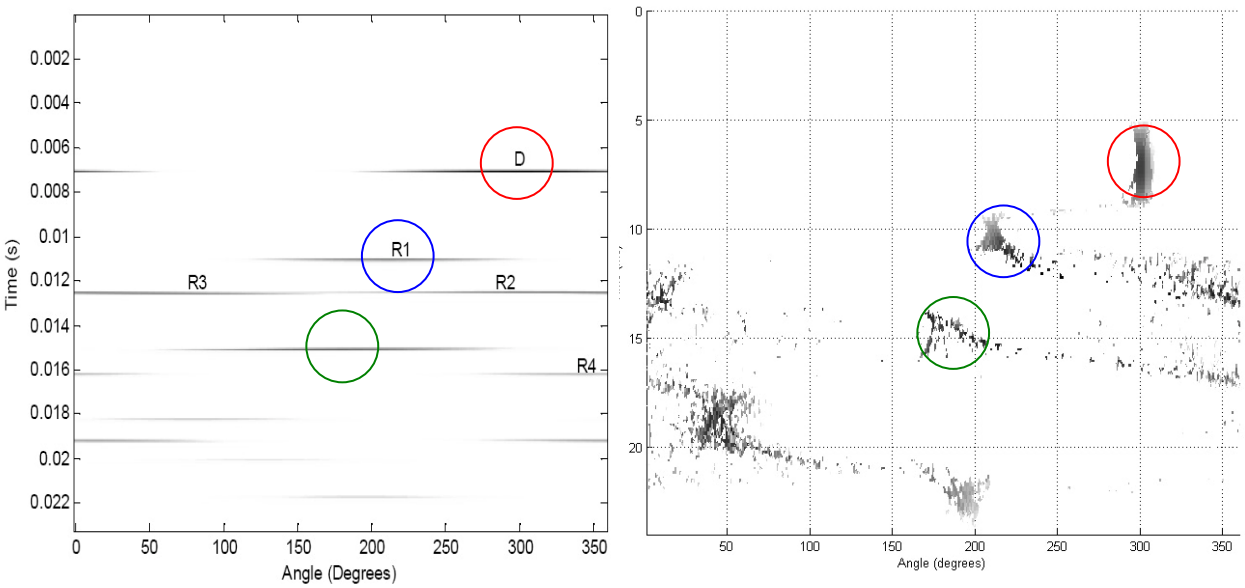

Figure 15: Left: Amplitude of the B-format signal plotted along time and direction predictions. Right: Energy of the W-channel plotted according to SIRR analysis results.

Again, some of the similarities in the plots have been circled. As with figure 14, when just a single sound arrives in a time window, the analysis produces a quite precise estimate, as can be seen in the first circle (red).

5.3 Real world input

Finally a real world RIR from a concert hall [15] was analysed. This RIR has been analysed by the originators of SIRR and the results have been published [1]. This allows for comparisons between results from this project and the published results, which is especially useful when it comes to testing the diffuseness estimate. This is because the result from the diffuseness estimate are harder to predict than the results from the direction estimate, so it is difficult to produce test signals with known diffuseness values.

The diffuseness estimation part of the analysis was performed on the file ‘s1_r4_sf.wav’ with 128 sample long Hann windows with an additional 128 samples zero-padding and largely overlapping windows. This is different to the normal 256 sample long Hann window with 256 samples zero-padding and overlapping by half a window. This was done to achieve a smoother plot more similar to the one published in [1].

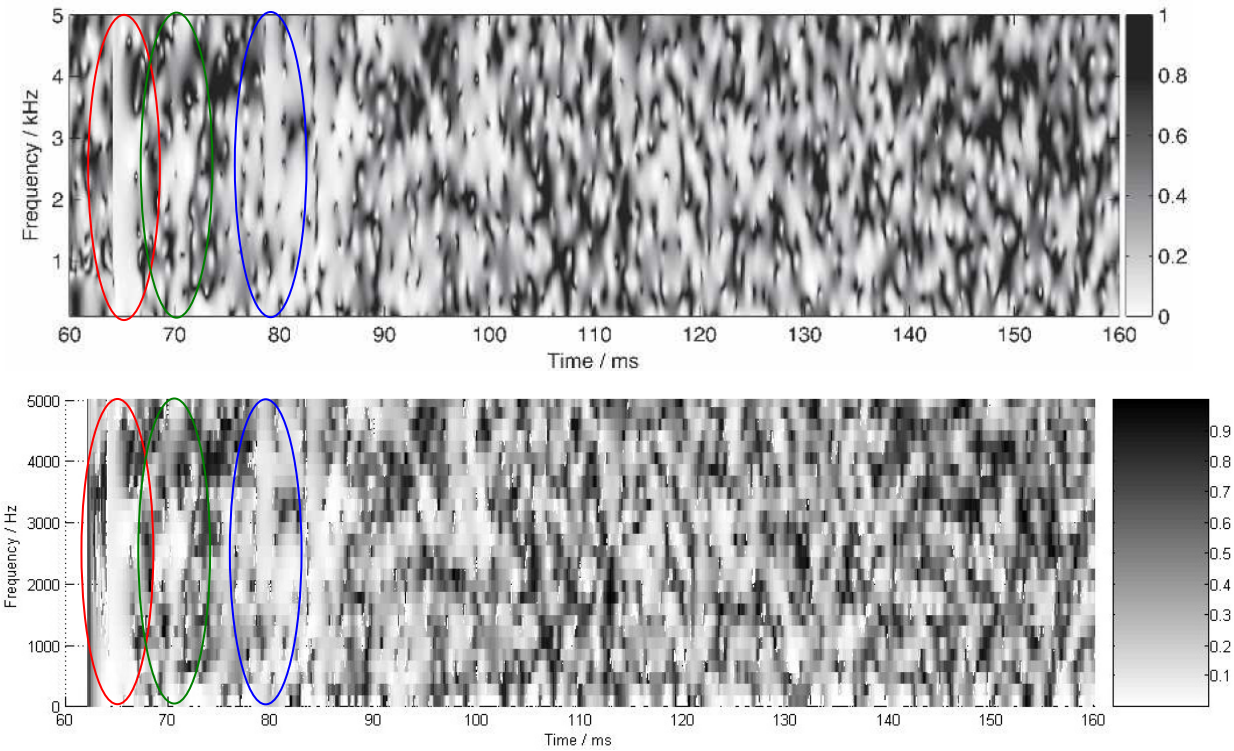

Figure 16: Diffuseness estimate plots. Top: Published results. Bottom: Results from this project.

It can be seen that the analysis results are similar. This diffuseness data is not weighted by the energy of the sound field, so it will look quite random in areas with low energy where it is difficult to estimate a diffuseness. The easiest features to recognise are the vertical white areas, where diffuseness is low; these are the direct sound and reflections. For example the area in the first circle (red) is the direct sound. It can be seen that in the upper part of that circle that both plots contain a similar high diffuseness (dark) area around 4 kHz.

6. A simple synthesis

6.1 Implementation

A simple version of synthesis was implemented in order to both test that the procedure to synthesise a multichannel RIR does indeed work, and as a method to further verify that the analysis data is correct and can be used.

The very first version of the synthesis simply did the time-frequency processing but nothing more. In other words the omni-directional signal was processed resulting in another omni-directional signal. This was done to make sure that the basic procedure behind the synthesis was working. A working synthesis would produce the same omni-directional impulse response except for round off errors in the process. Once this had been confirmed, the work could begin on the final version of the synthesis used in this project.

The synthesis implemented pans each time-frequency package to the desired direction based on the analysis data. The B-format impulse response is rendered to a multi channel impulse response. The target speaker setup is a six channel setup with the speakers spaced uniformly around the listener. The first speaker is at 30 degrees from the listener and the rest at every 60 degrees. The panning process is done with Vector Base Amplitude Panning (VBAP) as explained in section 2.6.5. Since panning is between six speaker channels spaced uniformly in the horizontal plane around the listener, only the azimuth data from the analysis is used and both the elevation and diffuseness estimates are discarded.

It should be noted that a full synthesis would include using the diffuseness estimate to split the energy in to direct and diffuse sound. This means that when the diffuseness is high, the sound would be presented more uniformly around the listener than in this implementation. This should be most noticeable when several reflections arrive around the same time. This is because as mentioned earlier the directional data will only show a single direction. When the sound field is diffuse this analysed direction will vary, but the results from section 5.2 suggest that it will not vary enough to represent several reflections arriving around the same time.

6.2 Results from the synthesis

The synthesis was tested on one of the RIR:s from the 2-D representation of a room used in section 5.2. This means that the same predictions about the direction of the direct sound and first order reflections are still valid. Additionally, the plots in that section can be used to make graphical estimations of the later reflections as shown in the following figure. The graphical estimations are done by looking for the middle of the horizontal line that represents a reflection.

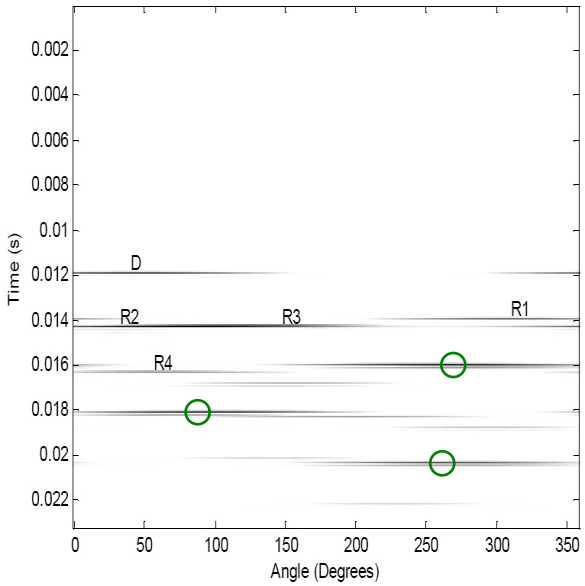

Figure 17: Amplitude of the B-format signal plotted with time and direction predictions from the ray-tracer as well as graphical estimations.

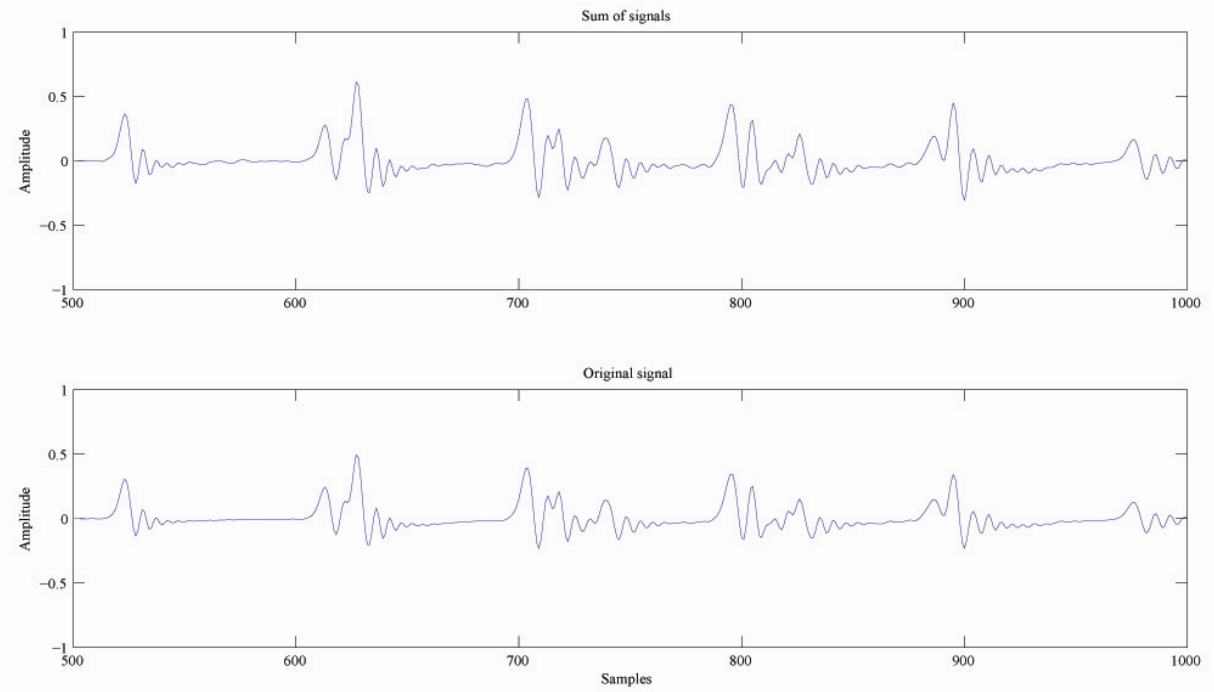

Figure 18: Waveform of the summation of all channels of the 6 channel RIR and the original waveform.

The above figure shows the waveform of all the six channels of the rendered multi channel impulse response summed together and the original omni-directional impulse response. It can be seen that the waveforms are similar. The higher amplitude in the sum signal is due to the fact that at any time at least two speakers will be active, and due to the nature of VBAP they will contain more than half the original energy each. The similarity indicates that the technique of synthesising a multichannel impulse response seems to work.

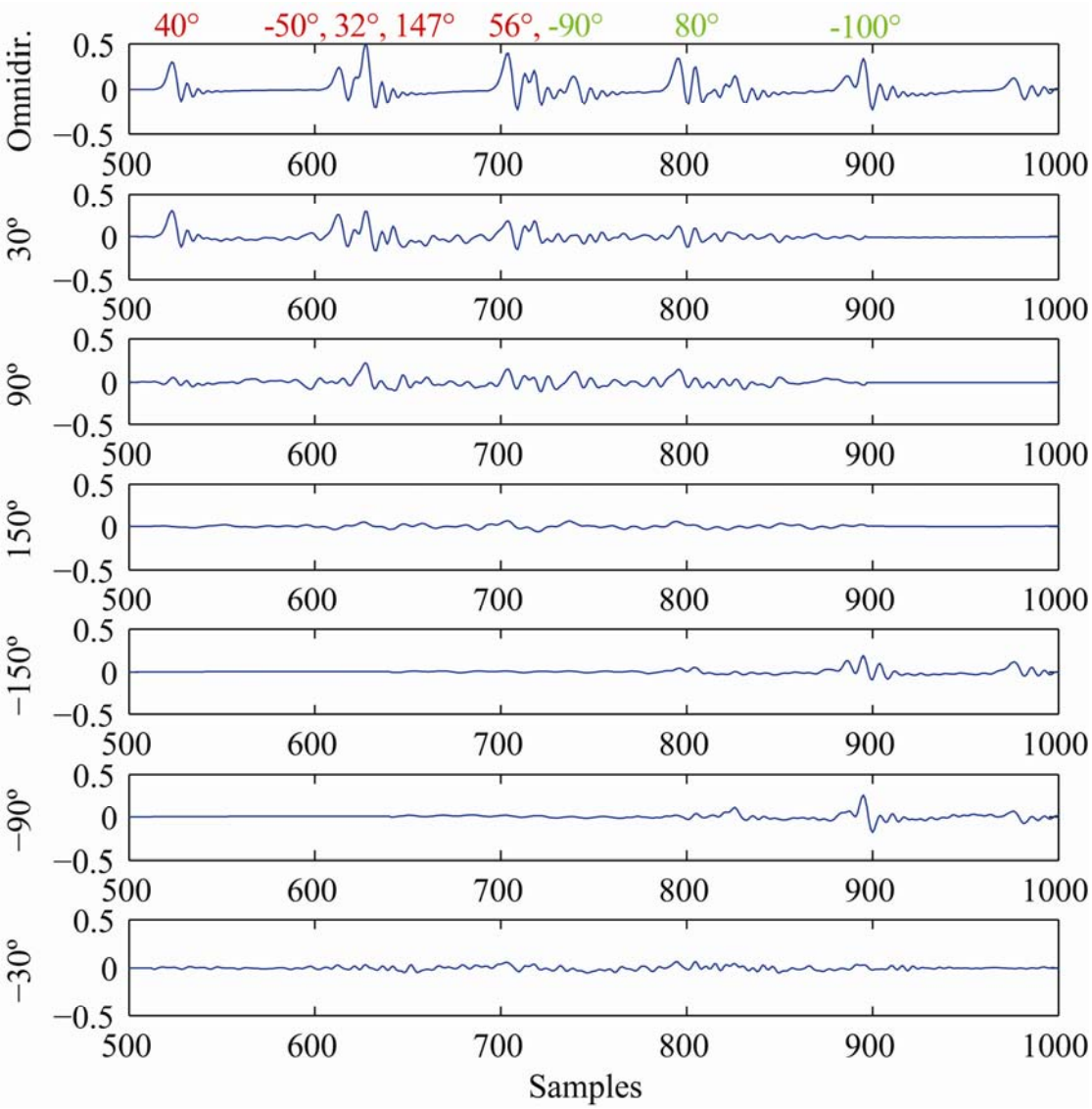

Figure 19: 6 channel RIR rendered from 1 channel omni-directional RIR.

The above figure shows the waveform of seven different speaker channels. The top one is the omni-directional RIR, followed by a multi-channel RIR that has been rendered from the omni-directional one based on analysis data. The direction of each speaker channel is noted on the left. Along the top of the figure is the expected direction of the direct sound and reflections. The first ones, in red, are the ones calculated with the ray-tracer used earlier, and the green ones are graphical estimates based on the plot. When several reflections arrive at the same time the directions are separated by a comma.

It can be seen in the figure that the energy from the direct sound and the different reflections are panned to the correct direction. For example the direct sound that is supposed to come from 40 degrees results in most of the energy in the 30 degree speaker channel. This confirms

that the synthesis does indeed pan the direct sound to the correct direction. Later reflections also confirm this, which is especially clear when only a single direction is “present”, like for instance the last reflection from -100 degrees.

7. Conclusions and further Work

7.1 Conclusions

The results from this project indicate that it is possible to use Spatial Impulse Response Rendering to reproduce the Room Impulse Response from a Digital Waveguide Mesh simulation. The SIRR analysis has been implemented in MATLAB, and tested with various inputs. It consistently provided accurate direction estimations. The test of the diffuseness estimate was encouraging as similar results were achieved as ones published by the originators of SIRR. There are no results that suggest that the technique would be any less suitable for using with artificially created input than real world recorded input.

Early work on the implementation of a synthesis shows promise. It has been demonstrated that the technique to pan each time-frequency component to the analysed direction gives results that are consistent with predictions. This indicates both that the basic principle of manipulating the omni-directional impulse response based on analysis data is viable, and that the analysis results that have been used are valid.

7.2 Further work

The obvious next step would be to implement a full synthesis. This would include adding the step of splitting the energy in to direct and diffuse parts based on the diffuseness estimate. The direct sound can be handled by the already implemented amplitude panning. It would be necessary to render the diffuse sound with one of the techniques described in [11], preferably the hybrid method chosen by the originators of the technique. It might also be necessary to restrict how fast the panning direction can change in order to avoid distortion that might occur due to switching panning direction too quickly. This is a problem that is mentioned briefly in [1], which in turn refers to [12] for further details.

With a working synthesis listening tests could be conducted to evaluate how well SIRR compares to existing techniques for spatial reproduction of the results from a DWM simulation. Another way of spatial reproduction that is available at the moment is doing a normal Ambisonics decoding of the B-format signal. Informal listening tests done in [11] suggest that SIRR should provide a better spatial experience than first order Ambisonics.

However comparisons with other techniques such as higher order Ambisonics and Wavefield Synthesis would be of interest if these techniques become available to the DWM application.

An aspect of the listening tests would also be to confirm if the technique does give the same spatial impression as the original sound field would. As the very basic idea behind the technique is to render a sound field that gives the same spatial impression as the original sound field this has to be verified. This is especially important when the DWM application is used for scientific purposes. If it is used to create reverberation for musical application it might be enough that it sounds nice to the listener.

Further testing of the analysis process might be necessary. The verification of the results done as a part of this project has been encouraging in every case. However, the tests have been conducted in an informal manner. To ensure that the analysis is correct for any input a more methodical testing process would be required.

Once a completely functional SIRR implementation is working it would also be possible to experiment with parameters such as window lengths and window overlapping, to see if a better result can be achieved. Another area of experimentation is the input used. A first order B-format signal is supposed to give complete information about the sound field at a certain point in space. However, it might be possible to achieve better results using another input, perhaps utilising more channels. The input is created with virtual receivers in the RoomWeaver application, so there is no practical limitation to the number of receivers as would be the case in the physical world.

It would also be possible to use the analysis data for other purposes than the SIRR synthesis. The data has to be used in a somewhat similar way since it applies to the time-frequency components. It could for example be used with a high resolution HRTF to produce a good quality binaural playback.

References

[1] J. Merimaa and V. Pulkki, “Spatial Impulse Response Rendering I: Analysis and Synthesis”, *J. Audio Eng. Soc.*, vol. 53, no. 12, pp. 1115-1127, 2005.

[2] B. C. J. Moore, *An Introduction to the Psychology of Hearing*, 5th ed., Elsevier Academic Press, London, UK, 2004.

[3] D. T. Murphy, "An Introduction to Spatial Sound", Audio technology (2F1410) Course Notes, Speech, Music and Hearing, KTH, Stockholm, October 2006.

[4] J. Liljencrants and S. Granqvist, *Elektroakustik*, Institutionen för Tal Musik och Hörsel, Kungliga Tekniska Högskolan, Stockholm, 2004.

[5] C. Faller and J. Merimaa, “Source Localization in Complex Listening Situations: Selection of Binaural Cues Based on Interaural Coherence” ”, *J. Acoust. Soc. Am.*, vol. 116, pp. 3075-3089, 2004.

[6] F. Rumsey, *Spatial Audio*, Focal Press, Oxford, UK, 2001.

[7] D. Murphy, A. Kelloniemi, J. Mullen and S. Shelley, “Acoustic Modeling using the Digital Waveguide Mesh”, *IEEE Signal Processing Letters*, In Press, 2007.

[8] D. Murphy, M. Beeson, S. Shelley and A. Moore, “Digital Waveguide Mesh Based Virtual Environment Modeling – The RoomWeaver System”, private communication.

[9] A. Southern and D. Murphy, “Spatial Encoding for Digital Waveguide Mesh Room Modeling Applictations”, *Proc. of the AES 28th Int. Conf.*, Piteå, Sweden, June 30-July 2, 2006, pp. 196-202.

[10] J. Merimaa and V. Pulkki, “Perceptually Based Processing of Directional Room Responses for Multichannel Loudspeaker Reproduction”, *Proc. IEEE Workshop on the Application of Signal Processing to Audio and Acoustics*, New Paltz, NY, USA, 2003.

[11] J. Merimaa and V. Pulkki, “Spatial Impulse Response Rendering II: Reproduction of Diffuse Sound and Listening Tests”, *J. Audio Eng. Soc.*, vol. 54, no. 1/2, pp. 3-20, 2006.

[12] F. Baumgarte and C. Faller, “Binaural Cue Coding. Part 1: Psychoacoustic Fundamentals and Design Principles” *IEEE Trans. on Speech and Audio Processing*, vol. 11, no. 6, pp. 509-519, 2003.

[13] V. Pulkki, “Virtual Sound SourcePositioning Using Vector Base Amplitude Panning”, *J. Audio Eng. Soc.*, vol. 45, no. 6, pp. 456-466, 1997.

[14] A. Southern, “Spatial rendering of digital waveguide mesh room acoustic models for multichannel sound,” M.Sc. thesis, Department of Electronics, The University of York, UK, 2006.

[15] J. Merimaa, T. Peltonen and T. Lokki, “Concert Hall Impulse Responses – Pori, Finland”. [Online] Available: http://www.acoustics.hut.fi/projects/poririrs/

Acknowledgements

This work was supported by The Swedish Foundation for International Cooperation in Research and Higher Education (STINT). STINT Contract No. IG2002-2049.

I would like to especially thank my supervisor Damian Murphy both for the support with this project and for welcoming me to York. I would also like to thank Sten Ternström for giving me the opportunity to go to York and do my thesis. Finally I would like to thank the people in the Music Technology Lab and all the students I met at the University of York for making me feel so welcome.

Appendix A – Analysis code

```
function [azimuth, elevation, diffuseness] = analysis(filename, plotit)
% analysis - Spatial Impulse Response Rendering analysis
%
% input:
%   filename    -   Name of a 4-channel wave file (.wav), where the channels
%                   represent the W, X, Y and Z channels of a first order
%                   B-format signal as used in Ambisonics
%
%   plotit      -   Any non-empty input (for example any number) will
%                   create a plot at the end of the analysis
%
% output:
%   azimuth     -   A 512xN matrix with the azimuth estimates for the 512
%                   FFT components in N timewindows.
%
%   elevation   -   A 512xN matrix with the elevation estimates for the 512
%                   FFT components in N timewindows.
%
%   diffuseness -   A 512xN matrix with the diffuseness estimates for the 512
%                   FFT components in N timewindows.
%
if nargin == 1
    plotit = [];                                                %any type of additional input
besides the filename will add a plot to the analysis
end

[input,fs,bits] = wavread(filename);
input = [zeros(128,4);input;zeros(256,4)];                      %zero-pad the entire input
ilength = length(input);

w = hann(256)*ones(1,4);                                        %create a window function
wlength = length(w);

Z0 = 410;

pos = 1;                                                        %sample to start the window
t = 1;                                                          %time window index
s = warning('off','MATLAB:divideByZero');
while (pos+wlength <= ilength) && (t < 150)
    winput = input(pos:pos+wlength-1,:).*w;                     %create a window
    winput = [zeros(128,4);winput;zeros(128,4)];                %zero-pad the window


    W = fft(winput(:,1),512);
    X = fft(winput(:,2),512);
    Y = fft(winput(:,3),512);
    Z = fft(winput(:,4),512);

    %active intensity
    Ix(:,t) = (sqrt(2)/Z0)*real(conj(W(1:257,:)).*X(1:257,:));
    Iy(:,t) = (sqrt(2)/Z0)*real(conj(W(1:257,:)).*Y(1:257,:));
    Iz(:,t) = (sqrt(2)/Z0)*real(conj(W(1:257,:)).*Z(1:257,:));

    %diffuseness
    U = [X Y Z];
    for i = 1:257
        diffuseness(i,t) = 1 - (sqrt(2)*norm(real(conj(W(i,:))*U(i,:)))) / (abs(W(i,:))^2 +
(norm(U(i,:))^2)/2);
    end

    if ~isempty(plotit) %plot stuff
        p = W.*conj(W) / 512;
        p2 = p';
        Pw(:,t) = p2(1:257);
    end %plot stuff

    t = t+1;
    pos = pos + wlength/2;                                      %move window position by half
the window size
end

azimuth = atan2(Iy, Ix);
elevation = atan((Iz)./(sqrt(Ix.^2+Iy.^2)));
azimuth = erbsmooth(azimuth, fs);
```

```
elevation = erbsmooth(elevation, fs);
diffuseness = erbsmooth(diffuseness, fs);
warning(s)

if ~isempty(plotit) %plot stuff
    Pwmin = max(max(Pw))/1000;
    Pw = 10*log10(Pw/Pwmin);
    [M, N] = size(Pw);
    for m = 1:M
        for n = 1:N
            if Pw(m,n) < 0
                Pw(m,n) = 0;
            end
        end
    end

    for n = 1:N
        Pw2(:,n) = interp1(1:257, Pw(:,n), 1:5:257);
        az(:,n) = interp1(1:257, azimuth(:,n), 1:5:257);
    end

    if N > 55
        plength = 55;
    else
        plength = N;
    end
    T = ((0:plength-1)*wlength/2)/fs*1000;
    F = fs*(0:10)/512/1000*5;
    F2 = fs*(0:54)/512/1000;
    [Xcoord,Ycoord] = pol2cart(az(1:11,1:plength),Pw2(1:11,1:plength));
    subplot(2,1,1);
    quiver(T, F, Xcoord, Ycoord, 0.5);
    axis([50 150 0 5]);
    grid on;
    subplot(2,1,2);
    surf(T,F2,diffuseness(1:55,1:plength),'EdgeColor','none');
    view(0,90);
    axis xy; axis([0 50 0 5]);
    colormap(1-gray);
end %plot stuff
```

Appendix B – Code to smooth analysis results

```
function output = erbsmooth(input, fs)
% erbsmooth - Smoothes the input from the SIRR analysis according to the
%             ERB scale.
%
% input:
%   input       -   The analysis result to be smoothed.
%
%   fs          -   Sampling frequency of the input to the analysis.
%
% output:
%   output      -   Smoothed result.
%
[M, N] = size(input);

f = fs*(0:256)/512;
erb = 24.7*(4.37*f'+1);
erb2 = round(erb/(f(2)*1000));

for n = 1:N
    for m = 1:M
        if erb2(m) <= 1
            w = 1;
            output(m,n) = input(m,n);
        else
            if M >= m+round(erb2(m)/2-0.1)
                output(m,n) = mean(input(m-floor(erb2(m)/2-0.1):m+round(erb2(m)/2-0.1),n));
            else
                output(m,n) = mean(input(M-length(w):M,n));
            end
        end
    end
end
```

Appendix C – Test files code

```
function null = testfile(filename, azimuth, elevation)
% testfile - Creates a B-format signal of an impulse encoded to a desired
%            direction and saves the result as a wave file.
%
% input:
%   filename    -   filename of the output without the .wav extension
%
%   azimuth     -   desired azimuth of the output signal
%
%   elevation   -   desired elevation of the output signal
%
input = [zeros(1999,1);1;zeros(1000,1)];
W = input*(1/sqrt(2));
X = input*cosd(azimuth)*cosd(elevation);
Y = input*sind(azimuth)*cosd(elevation);
Z = input*sind(elevation);
bformat = [W X Y Z];
wavwrite(bformat,48000,24,filename);

function null = testfile2(filename, f1, f2, azimuth1, azimuth2)
% testfile2 - Creates a B-format signal of two sine waves with different
%             frequency and direction and saves the result as a wave file.
%
% input:
%   filename                -   filename of the output without the .wav extension
%
%   f1 & f2                 -   frequency of sinewave 1 & 2
%
%   azimuth1 & azimuth2     -   azimuth of sinewave 1 & 2
%
sine1 = sin(2*pi*f1*[0:(1/48000):0.1])/2;
sine2 = sin(2*pi*f2*[0:(1/48000):0.1])/2;
input1 = sine1';
input2 = sine2';
W = (input1+input2)*(1/sqrt(2));
X = input1*cosd(azimuth1)+input2*cosd(azimuth2);
Y = input1*sind(azimuth1)+input2*sind(azimuth2);
Z = (input1+input2)*0;
bformat = [W X Y Z];
wavwrite(bformat,48000,24,filename);

function null = testfile3(filename);
% testfile3 - Creates a B-format signal of 10 impulses coming one after
%             another from 10 different directions around the listener.
%
% input:
%   filename    -   filename of the output without the .wav extension
%
input = [zeros(150,1);1;zeros(149,1)];
W = [];
X = [];
Y = [];
Z = [];
for i = 1:10
    azimuth = 36*i;
    W = [W; input*(1/sqrt(2))];
    X = [X; input*cosd(azimuth)];
    Y = [Y; input*sind(azimuth)];
    Z = [Z; input*0];
end
bformat = [W X Y Z];
wavwrite(bformat,48000,24,filename);
```

Appendix D – Modified code to plot diffusion

```
function rho = diffusion2(filename)
% diffusion2 - Spatial Impulse Response Rendering analysis of the diffusion
%              with largely overlapping windows. The result is plotted.
%
% input:
%   filename    -   4-channel wave file (.wav) where the channels represent
%                   the W, X, Y and Z channels of a first order B-format
%                   signal as used in Ambisonics
%
% output:
%   rho         -   A 512xN matrix with the diffuseness estimates for the
%                   256 FFT components in N timewindows.
%
[input,fs,bits] = wavread(filename);
input = [zeros(64,4);input;zeros(128,4)];
ilength = length(input);

w = hann(128)*ones(1,4);                                %create a window function
wlength = length(w);

pos = 1;
t = 1;                                                  %time window index
s = warning('off','MATLAB:divideByZero');
while (pos+wlength <= ilength) && (t <= 2000)
    winput = input(pos:pos+wlength-1,:).*w;             %create a window
    winput = [zeros(64,4);winput;zeros(64,4)];          %zero-pad the window

    W = fft(winput(:,1),256);
    X = fft(winput(:,2),256);
    Y = fft(winput(:,3),256);
    Z = fft(winput(:,4),256);

    %diffuseness
    U = [X Y Z];
    for i = 1:129
        rho(i,t) = 1 - (sqrt(2)*norm(real(conj(W(i,:))*U(i,:)))) / (abs(W(i,:))^2 +
(norm(U(i,:))^2)/2);
    end

    t = t+1;
    pos = pos + wlength/16;                             %move window position by half the
window size
end
%rho = erbsmooth(rho, fs);                              %skip the smoothing
warning(s)

    [M, N] = size(rho);
    plength = N;
    T = ((0:N-1)*wlength/16)/fs*1000;
    F = fs*(0:54)/256;
    surf(T,F,rho(1:55,:),'EdgeColor','none');
    view(0,90);
    axis xy; axis([55 155 0 5000])
    colormap(1-gray);
```

Appendix E – Synthesis code

```
function output = synthesis(infile, outfile);
% synthesis - Spatial Impulse Response Rendering synthesis
%
% input:
%   infile      -   Name of a 4-channel wave file (.wav), where the channels
%                   represent the W, X, Y and Z channels of a first order
%                   B-format signal as used in Ambisonics
%
%   outfile     -   Name of a 6-channel wave file that will be created as
%                   output from the synthesis.
%
% output:
%   output      -   The same output that is written to the wave file.
%
[azimuth, elevation, diffuseness] = analysis(infile);

[input,fs,bits] = wavread(infile);
input = [zeros(128,4);input;zeros(256,4)];                      %zero-pad the entire input
ilength = length(input);

w = hann(256)*ones(1,4);                                        %create a window function
wlength = length(w);

pos = 1;
t = 1;                                                          %time window index
speakers = dlmread('speakers.txt');
for i=1:length(speakers);
    rad = deg2rad(speakers);
    [l(i,1) l(i,2)] = pol2cart(rad(i), 1);
end
L12 = inv([l(1,:);l(2,:)]);
L23 = inv([l(2,:);l(3,:)]);
L34 = inv([l(3,:);l(4,:)]);
L45 = inv([l(4,:);l(5,:)]);
L56 = inv([l(5,:);l(6,:)]);
L61 = inv([l(6,:);l(1,:)]);
output = zeros(ilength,length(speakers));
while (pos+wlength <= ilength) && (t < 56)
    winput = input(pos:pos+wlength-1,:).*w;                     %create a window
    winput = [zeros(128,4);winput;zeros(128,4)];                %zero-pad the window

    W = fft(winput(:,1));

    %----------amplitude panning
    [p(1:257,1) p(1:257,2)] = pol2cart(azimuth(1:257,t), 1);
    result = [];
    for i=1:257
        g(1,:) = p(i,:)*L12;
        g(2,:) = p(i,:)*L23;
        g(3,:) = p(i,:)*L34;
        g(4,:) = p(i,:)*L45;
        g(5,:) = p(i,:)*L56;
        g(6,:) = p(i,:)*L61;
        ming = min(g');
        [value, index] = max(ming);
        if value < 0
            for i=1:2
                if g(index,i) < 0
                    g(index,i) = 0;
                end
            end
        end
        gscaled = (1/sqrt(g(index,1)^2+g(index,2)^2))*g(index,:);
        gain = zeros(1,6);
        gain(index) = gscaled(1);
        if index == 6
            gain(1) = gscaled(2);
        else
            gain(index+1) = gscaled(2);
        end
        result(i,:) = W(i)*gain;
    end
```

```
    %----------amplitude panning
    resultr = real(result);
    resulti = imag(result);
    result2 = flipud(complex(resultr, -resulti));
    result(258:512,:) = result2(2:256,:);

    inverse = ifft(result);
    out = inverse(129:384,:);
    output(pos:pos+wlength-1,:) = output(pos:pos+wlength-1,:)+out;

    t = t+1;
    pos = pos + wlength/2;                                    %move window position by half
the window size
end
output(ilength-255:ilength,:) = []; output(1:128,:) = [];       %remove zero-padding from
output
wavwrite(output, fs, bits, outfile);
```