Master thesis in Sound and Music Computing

Universitat Pompeu Fabra

Advancing a Wavelet-Based Spatial Audio Format

Samuel Narváez

Supervisor: Daniel Arteaga

Co-Supervisor: Davide Scaini

July 2022

Master thesis in Sound and Music Computing

Universitat Pompeu Fabra

Advancing a Wavelet-Based Spatial Audio Format

Samuel Narváez

Supervisor: Daniel Arteaga

Co-Supervisor: Davide Scaini

July 2022

Contents

Acknowledgement

I would like to express my sincere gratitude to:

- Daniel Arteaga and Davide Scaini – for all of their support, teaching, patience and advocating on my behalf.

- My family – for their long-distance support this past year.

- Friends, faculty and colleagues at the MTG and Taller de Musics for helping me in uncountably many ways.

- Friends, faculty and colleagues from Oberlin, especially the Math and TIMARA departments, for being a key step in the journey that brought me to this point

Abstract

This work further develops the theory of Spherical Wavelet Format (SWF), a spatial audio format inspired by Ambisonics that makes use of Spherical Wavelets as a basis to decompose the soundfield. In particular, we have specified a version of SWF that implements a method to build an arbitrary wavelet representation on an arbitrary triangular mesh. We make use of a modified lifting scheme to optimize the interpolating scaling functions for optimal playback reproduction, and we have demonstrated its functionality and competitiveness with state-of-the-art spatial audio algorithms on a 7.1.4 layout. The resulting SWF specification is available for use and further research in an open source python library. The python library is flexible enough to support any layout, and includes presets for the original Octahedral mesh from the first publication of SWF, as well as the 7.0.4-based SWF format used for objective and subjective evaluation in this report, and a Spherical Wavelet Format that naturally interpolates between standard surround sound formats (11.1.8,9.1.6,7.1.4,etc.). The library is intended to be used with the trivial decoding from the coarsest level of mesh, but can also be decoded using other strategies from less coarse representations.

Keywords: Second-Generation Wavelets; Spatial Audio; Spherical Wavelets; SWF; Ambisonics; VBAP; Triangular Mesh; Surround Sound

Chapter 1

Introduction

Spherical Wavelet Format (SWF) is a framework for spatial audio inspired by Ambisonics, published in 2020 by Davide Scaini and Daniel Arteaga. [1] [2]

The final concept they developed allows to encode sound sources to a cloud of points and to reduce (or recover) the dimensionality of the cloud at will. The spatial downsampling is implemented as a linear transformation that can be fully reverted. This construction allows for different coexisting spatial representations, that can scale based on various requirements, for example transmission bandwidth or the complexity of the destination playback system.

1.1 Motivation

The original exploration of SWF by Scaini and Arteaga was limited to a format based on the regular octahedral mesh. This limitation was necessary in that, it allowed them to optimize the format by brute force – taking advantage of the inherent symmetries of this mesh. However, the theory of SWF is not limited to regular meshes. Nor is it, neccessarily, limited to meshes that approximate the sphere or hemisphere. This work extends SWF to all triangular meshes, and introduces a psychoacoustically-motivated optimization technique that is general enough to be applied to any mesh, and fast enough to be carried out at the point of mesh

instantiation.

1.2 Objectives

The most significant objective of this work is the development of an open-source python library implementing a version of SWF for use in real-time applications that can serve as a starting point for further research.

We sought that this version of SWF should be flexible enough to handle all possible loudspeaker layouts by way of the base subivision mesh, particularly irregular layouts and industry standards such as Dolby Atmos surround sound formats. Additionally, this version of SWF should be optimized for certain psychoacoustic properties like spatial resolution and timbre preservation.

We have been largely successful in this. The resulting library implements a method to build an arbitrary wavelet representation on an arbitrary mesh, and a method to refine the format for optimal reproduction.

1.3 Structure of the Report

This report will introduce Spatial Audio broadly for the uninitiated reader, both its psychoacoustical foundations and the industry-standard methods for achieving 3D sound. We will then move into an overview of Multiresolution analysis: the framework for introducing wavelets, scaling functions, and an algorithm called the Lifting Scheme. We will dig into the fundamentals of SWF and expand upon how this has been implemented. We will explore the tools and methodology necessary to expand the SWF to general layouts. Subsequently, we will discuss the results generated by the library, focusing in particular on the case of the 7.1.4 standard layout, and talk about the Max patch and other realtime applications. Finally, we will perform an objective comparison to state-of-the-art spatial audio techniques.

Chapter 2

State of the Art

2.1 Localization of Sound by Humans

To motivate our understanding of spatial audio, let's ground ourselves in the physiological bases for sound localization. Humans achieve localization of sound events through assessing subtle differences in the sensations perceived by our ears.

These binaural (involving both ears) differences are key to how we perceive the location of a sound source. They are composed of the Interaural Time Difference (ITD) and Interaural Level Difference (ILD) [3] and can be measured by the head related transfer function (HRTF). The ITD describes the difference in time that it takes for a sound event to be percieved by the other ear once it has been detected in one ear. The ILD describes the difference in loudness sensation between the two ears as they percieve the same sound event.

Coherent signals from multiple loudspeakers combine linearly at the ears of the listener. This commonly is referred to as the Summing Localization Principle [4] and is the hypothesis that motivates amplitude-based panning. Summing localization assumes that the characteristics of this additive sound field are similar enough to the characteristics of the sound field that is produced by a single real source. A listener perceives a single auditory event at the location of this equivalent single sound source, commonly referred to as a phantom or virtual source.

2.2 Spatial Audio

Spatial audio refers to the set of tools, technologies and theories for creation or recreation of a subjective sound scene, that has to produce all the spatial characteristics of a sound located in a 2D or 3D space: direction, distance and size.

It is possible to classify the techniques to (re)create an auditory scene (2D or 3D) in three categories[1]:

- Discrete panning techniques (e.g. Stereo, Traditional Multichannel, VBAP, MDAP, ABAP, VBIP, ABIP): the known apparent direction of the source is used to feed a limited number of loudspeakers. This approach is based on the summing localization principle, and seeks to position phantom sources through either amplitude or intensity panning.
- Sound field reconstruction methods (e.g. Ambisonics, Wave Field Synthesis): the intent is to control the acoustical variables of the sound field (pressure, velocity) in the listening space.
- Head-related stereophony (binaural, transaural): the aim is to measure (binaural recording) or (re)produce (binaural synthesis) the acoustic pressure at the ears of the listener. Leverages ITD and ILD to create the impression of space via Head Related Transfer Functions (HRTFs).[5]

Besides the underlying theory of each technology, the spatial audio techniques can be also classified by analyzing how the whole encoding/decoding pipeline is structured:

- Layout-dependent : the whole encoding/decoding and recording/reproduction is based on a specific channel layout, e.g. 2.0, 5.1, 7.1, ...
- Layout-independent (channel-agnostic): the recording and encoding format is independent from the reproduction layout.

Within Layout-independent techniques, there exist two main categories:

- Channel Based : the format is transmitted in a fixed number of channels, which is specified at the beginning of the production chain.
- Object Based: the format is based on audio objects, which are representations of virtual sources that move in a digital representation of space.

Malham [6] developed two criteria that can be applied to any existing or future surround sound technology: the ideas of homogeneous and coherent sound reproduction systems. Quoting from [Malham, 1999]:

> "A *homogeneous* sound reproduction system is defined as one in which no direction is preferentially treated. A *coherent* system as one in which the image remains stable if the listener changes position within it, though the image may change as a natural soundfield does."

2.2.1 VBAP

Vector Based Amplitude Panning (VBAP) [1] is a channel-based discrete panning technique and in general is not homogeneous. In a horizontal plane around the listener, a virtual sound source at a certain position is created by applying the tangent panning law between the closest pair of loudspeakers. This principle was also extended to project sound sources onto a three dimensional sphere and assumes that the listener is located in the center of the equidistant speaker setup. In three dimensions, VBAP computes a triliinear interpolation between the points of the triangle of loudspeakers which contains the virtual source.

2.2.2 Ambisonics

Ambisonics [7] is a theory that aims at reconstructing the sound field, is layout-independent, is coherent and homogeneous. Ambisonics was developed in the UK in the 1970s as a full-sphere surround sound format. It decomposes the sound field by the spherical harmonics, an orthonormal basis for representing functions defined on the surface of the sphere. Sound sources must be encoded to Ambisonics B-format,

wherein the channels represent a truncated decomposition of the sound field by the spherical harmonics. Low-order ambisonics limit this decomposition to the first order spherical harmonics, which correspond to the three X,Y, and Z pressure gradients at a point in space with sound pressure W. Higher order ambisonics include channels that correspond with higher order spherical harmonics. Some of the drawbacks of low order Ambisonics, like large source spread and small sweet-spot, are directly related to the fact that spherical harmonics do not have compact support on the sphere.

An Ambisonic encoding distributes a sound source over the spherical harmonic components with different gains. This effectively represents the sound field in the basis of the spherical harmonics. Higher order Ambisonics include higher orders of spherical harmonics, thus increasing the number of transmission channels and the number of channels neccessary to reproduce the decoded soundfield.

2.2.3 Spherical Wavelet Format

Spherical Wavelet Format (SWF) [2] as a theory is layout-independent, channel-based, coherent and homogeneous. The specific version of SWF developed in this thesis is functionally layout-dependant although technically still decodable to any layout. More details on this later.

SWF is similar in spirit to Ambisonics, though SWF replaces the spherical harmonics by an alternative set of functions with compact support, the spherical wavelets.

Spherical wavelets are wave-like oscillations defined on the sphere that, differently to spherical harmonics, can be associated to a certain angular direction.

Wavelets typically have compact support: they are zero or decay very fast outside the region of interest, implying that they have an explicit directionality; they naturally offer a system to reduce/scale information and can be tuned to the signal so to have a set of desired properties. We will now introduce some tools and techniques needed to enable SWF, which will be discussed in further detail in Section 2.7.

2.3 Multiresolution Framework for Spatial Audio

Speaking generally, a virtual source distribution can be thought of as a function on the surface of the sphere. Any virtual source can have an arbitrary position in spherical space, with continuous azimuth and elevation coordinates (θ, σ). It is thus necessary for any sound field theory to be able to represent general functions defined on the sphere accurately.

Schroder [8] highlights wavelets' suitability for this task, stating:

> Wavelets have proven to be powerful bases for use in numerical analysis and signal processing. Their power lies in the fact that they only require a small number of coefficients to represent general functions and large data sets accurately. This allows compression and efficient computations.

2.3.1 Mesh Subdivision

The second generation spherical wavelets used in SWF are discrete, and defined procedurally by the repeated subdivision of a mesh approximating the sphere, which is typically built starting from a chosen base mesh and running a subdivision scheme. The result of an iteration of the subdivision scheme on a mesh of level n is a mesh of level $n + 1$, the base mesh being of level 0. The higher the level of subdivision of the mesh, the closer it will approximate the sphere.

The subdivision works as follows: At the coarsest level, we start with a triangular mesh, meaning each face of the mesh has exactly three vertices. At the midpoint of every unique edge, we impute a new vertex and project it onto the surface of the sphere. Then we draw new edges such that each triangular face from the coarse level becomes four smaller triangular faces. Our result is a finer (more dense) mesh that more closely approximates the sphere.

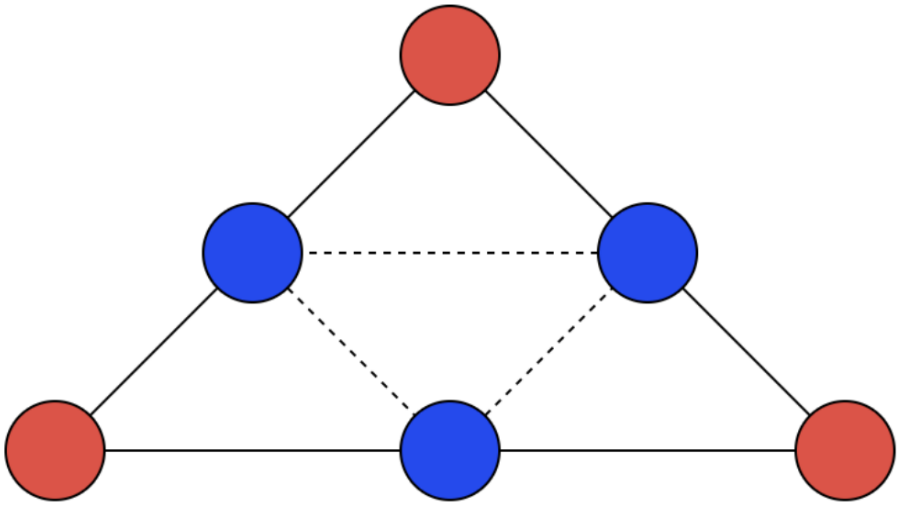


Figure 1: An Illustration of one iteration of the subdivision scheme on a single triangular face.

2.3.2 Multiresolution Analysis

The subdivision mesh is the skeleton for our multiresolution analysis, the system that lets us represent data (our virtual source distribution) in different levels of detail.

More generally, a multiresolution analysis consists of a nested set of closed vector subspaces:

$$V_0 \subset V_1 \subset ...V_j \subset ... \subset V_n \tag{2.1}$$

with $n \geq 0$ such that:

- $V_j \subset V_{j+1} \forall j$
- for each j, the basis functions of V_j are called *scaling functions* and are denoted ϕ_k^j with $k \in \mathbb{K}(j)$. Where $\mathbb{K}$ is an index set with $\mathbb{K}(j) \subset \mathbb{K}(j+1)$
- since the vector spaces are nested, it is possible to to write each ϕ_k^j as a function

of the next level ϕ_{j+1} and obtain these refinement relations:

$$\phi_k^j = \sum_l p_{l,k}^{j+1} \phi_l^{j+1} \tag{2.2}$$

for some p, where $n > j \geq 0$, $k \in \mathbb{K}(j)$ and $l \in \mathbb{K}(j+1)$. We can write the scaling functions more concisely as a row vector:

$$\phi^j = (\phi_1^j, ..., \phi_{m^j}^j) \tag{2.3}$$

where m^j is the dimension of V^j

2.3.3 Wavelets

The wavelet spaces, W^j, are defined to be the orthogonal complement of V^j in V^{j+1}, such that $V^j \oplus W^j = V^{j+1}$, meaning that W^j includes all the functions in V^{j+1} that are orthogonal to all those in V^j under some inner product. The functions that form a basis of W^j are called *wavelets*, and are denoted with ψ_p^j. The refinement relation for wavelets is defined similarly to the scaling functions:

$$\psi_k^j = \sum_l q_{l,k}^{j+1} \phi_l^{j+1} \tag{2.4}$$

for some q, and they can be similarly condensed as a row vector:

$$\psi^j = (\psi_1^j, ..., \psi_{n^j}^j) \tag{2.5}$$

where n^j is the dimension of W^j, with $m^j + 1 = m^j + n^j$.

With this matrix notation now developed, it is clearer and more convenient to rewrite the refinement relations as:

$$\phi^j = \phi^{j+1} \mathbf{P}^{j+1} \tag{2.6}$$

$$\psi^j = \phi^{j+1} \mathbf{Q}^{j+1} \tag{2.7}$$

satisfying the biorthogonality condition:

$$\langle \phi^j | \psi^j \rangle = \mathbf{0} \tag{2.8}$$

where, $\langle \phi^j | \psi^j \rangle$ denotes the inner product.

Every multiresolution analysis has a dual multiresolution analysis consisting of nested spaces $\tilde{V}^j$ with bases given by dual scaling functions $\tilde{\phi^j}$, which are biorthogonal to the scaling functions:

$$\langle \tilde{\phi^j} | \phi^j \rangle = \mathbf{1} \tag{2.9}$$

And similarly, for any given wavelet basis there exists a dual wavelet basis $\tilde{\psi^j}$, the two of which are biorthogonal to each other:

$$\langle \tilde{\psi^j} | \psi^j \rangle = \mathbf{1} \tag{2.10}$$

and the duals satisfy similar refinement relations:

$$\tilde{\phi^j} = \tilde{\phi^{j+1}} [\mathbf{A}^{j+1}]^T \tag{2.11}$$

$$\tilde{\psi^j} = \tilde{\phi^{j+1}} [\mathbf{B}^{j+1}]^T \tag{2.12}$$

These operators, $\mathbf{A}^j$, $\mathbf{B}^j$, $\mathbf{P}^j$, and $\mathbf{Q}^j$ are the decomposition and reconstruction filters (respectively) that will be contextualized and discussed further in section 2.4.

They must satisfy the following biorthogonality relations:

- $\mathbf{B}^j \mathbf{P}^j = \mathbf{0}$
- $\mathbf{A}^j \mathbf{Q}^j = \mathbf{0}$
- $\mathbf{A}^j \mathbf{P}^j = \mathbf{1}$
- $\mathbf{B}^j \mathbf{Q}^j = \mathbf{1}$
- $\mathbf{B}^j \mathbf{A}^j + \mathbf{Q}^j \mathbf{B}^j = \mathbf{1}$

And can be used to compute the scaling functions, wavelets, and their duals as follows:

- $\phi_k^j(p) = (\mathbf{P}^n ... \mathbf{P}^{j+2}\mathbf{P}^{j+1})_{pk}$
- $\psi_k^j(p) = (\mathbf{P}^n ... \mathbf{P}^{j+2}\mathbf{Q}^{j+1})_{pk}$
- $\tilde{\phi}_k^j(p) = (\mathbf{A}^{j+1}\mathbf{A}^{j+2} ... \mathbf{A}^n)_{kp}$
- $\tilde{\psi}_k^j(p) = (\mathbf{B}^{j+1}\mathbf{A}^{j+2} ... \mathbf{A}^n)_{kp}$

A more complete exposition of these details can be found in Appendix A of [2].

2.4 Spherical Wavelet Format for Spatial Audio

We are now prepared to discuss in detail the Spherical Wavelet Format. [2] A Spherical Wavelet Format is completely determined by:

1. A recursive subdivision mesh over the sphere, ranging from the coarsest level 0 to the finest level n.
2. A set of filters $\{\mathbf{A}^j, \mathbf{B}^j, \mathbf{P}^j, \mathbf{Q}^j | j \in [1, n]\}$, defining a wavelet space, that satisfy the biorthogonal relations.
3. A truncation level $l \in [0, n]$, defining the order of the wavelet decomposition.

2.4.1 Encoding & Signal Deconstruction

Given a mesh, we have a set of data defined over the finest level of subdivision, $\mathbf{f} = (f_1, ..., f_n)^T$, which in the audio domain represents the original source distribution at the finest considered resolution (in this case, n being the number of vertices in the finest level of mesh). The process of downsampling decomposes the signal $\mathbf{f}$ into two signals, a coarse approximation, $\mathbf{c}^{n-1}$ and the details, $\mathbf{d}^{n-1}$ which are computed

using the encoding filters:

$$\mathbf{c}^{n-1} = \mathbf{A}^n \mathbf{f} \tag{2.13}$$

$$\mathbf{d}^{n-1} = \mathbf{B}^n \mathbf{f} \tag{2.14}$$

Where $\mathbf{A}^n, \mathbf{B}^n$ are the decomposition, encoding, or analysis filters at level n. The signal $\mathbf{c}^{n-1}$ represents a spatially low-passed and downsampled version of $\mathbf{f}$. The signal $\mathbf{d}^{n-1}$ represents the information missing from $\mathbf{c}^{n-1}$ relative to $\mathbf{f}$.

The decomposition can continue by iterating the decomposition of the coarse data, if the decomposition is followed up to the coarsest level available (level 0), there will be a list of $n-1$ detail signals or wavelet coefficients, $\mathbf{d}^0, ..., \mathbf{d}^{n-1}$ and one last coarse signal or scaling function coefficients $\mathbf{c}^0$; the representation $\{\mathbf{c}^0, \mathbf{d}^0, ..., \mathbf{d}^{n-1}\}$ constitutes the wavelet transform. The coarse signal $\mathbf{c}^0$ encodes a spatially downsampled version of the signal at the base mesh, and the detail signals $\mathbf{d}^0, ..., \mathbf{d}^{n-1}$ encode the difference between the representations in two successive levels. This representation is useful for transmission, and can be even decoded directly using the trivial decoding.

2.4.2 Trivial Decoding

The trivial decoding assumes that in the reproduction setting, there is a speaker located at each of the vertices at some level of mesh, typically level 0. Thus, the coarse signals $\mathbf{c}^l$ can be considered the coefficients of the trivial decoding. Details on nontrivial decodings can be found in the original paper [2] but they will not be considered here.

2.4.3 Signal Reconstruction

The upsampling process increases the spatial resolution of the coarse data $\mathbf{c}^{n-1}$ to the fine data $\mathbf{f}$, and if the details $\mathbf{d}^{n-1}$ are available, then the reconstruction process

will give back the original fine data:

$$\mathbf{f} = \mathbf{P}^n \mathbf{c}^{n-1} + \mathbf{Q}^n \mathbf{d}^{n-1} \tag{2.15}$$

where $\mathbf{P}^n$ and $\mathbf{Q}^n$ are the reconstruction, decoding, or synthesis filters at level n. In order to achieve compression, it may be advantageous to limit the decomposition up to a given level, in which case if the encoding has been truncated at decomposition level l, the detail coefficients with level equal to or greater than l will all be zero, and thus a reconstruction of the signal will result in a spatially low-passed version of the original signal $\mathbf{f}$.

2.4.4 Operator Reference

The reader may find these abuses of notation helpful to remember the functions of the operators mentioned so far.

$$\mathbf{P} : \mathbf{c} \mapsto \mathbf{f} \tag{2.16}$$

$$\mathbf{Q} : \mathbf{d} \mapsto \mathbf{f} \tag{2.17}$$

$$\mathbf{A} : \mathbf{f} \mapsto \mathbf{c} \tag{2.18}$$

$$\mathbf{B} : \mathbf{f} \mapsto \mathbf{d} \tag{2.19}$$

$$\mathbf{S} : \mathbf{c} \mapsto \mathbf{d} \tag{2.20}$$

$$\mathbf{T} : \mathbf{d} \mapsto \mathbf{c} \tag{2.21}$$

2.4.5 Filter Design

Arguably the most important factor to the success and performance of a particular Spherical Wavelet Format is the design of the filters $\mathbf{A}, \mathbf{B}, \mathbf{P}$, and $\mathbf{Q}$. Beyond satisfying the biorthogonality conditions, they should have as compact support as possible, have smooth spatial low-pass characteristics, preserve the total acoustic pressure between downsampling/upsampling, and limit out of phase components. Some example filters are included in the figures below. These filters were generated by the lifting scheme on a second-order subdivided octahedral mesh.

2.4.6 Lifting Scheme

The lifting scheme is a method invented by Wim Sweldens [9] to update the biorthogonal wavelet filters $\mathbf{A}^j$, $\mathbf{B}^j$, $\mathbf{P}^j$, and $\mathbf{Q}^j$. The whole idea of the lifting scheme is to start from one basic multiresolution analysis, which can be simple or even trivial, and construct a new, more performant one, i.e., the basis functions are smoother or the wavelets have more vanishing moments. In case the basic filters are finite we will have lifted filters which are also finite. One feature of the lifting scheme is

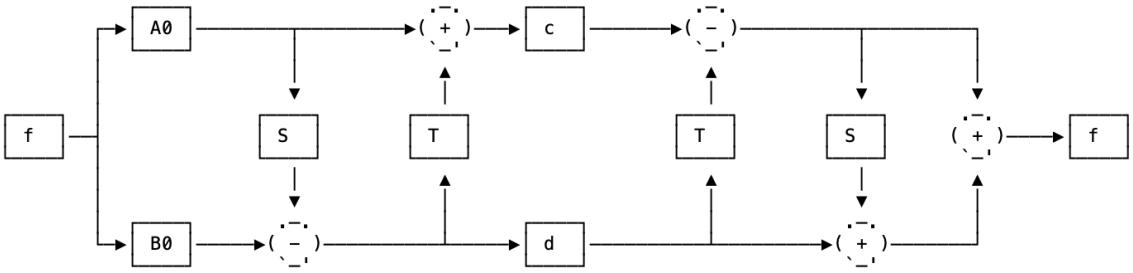


Figure 2: The lifting scheme.

that it always generates biorthogonal lifted filters if the basic filters provided are biorthogonal. This can be verified from the definition of the Lifting Scheme.

Let's start with the definition of our operators $\mathbf{A}^j$, $\mathbf{B}^j$, $\mathbf{P}^j$, and $\mathbf{Q}^j$ and derive new filters $\bar{\mathbf{A}}^j$, $\bar{\mathbf{B}}^j$, $\bar{\mathbf{P}}^j$, and $\bar{\mathbf{Q}}^j$ by chasing the diagram in Figure 2.

We have:

$$\mathbf{A}^j\mathbf{f} = \mathbf{c} \tag{2.22}$$

$$\mathbf{B}^j\mathbf{f} = \mathbf{d} \tag{2.23}$$

$$\mathbf{P}^j\mathbf{c} + \mathbf{Q}^j\mathbf{d} = \mathbf{f} \tag{2.24}$$

$$\tag{2.25}$$

Applying the steps from the diagram gives:

$$((\mathbf{1} - \mathbf{T}^j \cdot \mathbf{S}^j) \cdot \mathbf{A}^j + \mathbf{T}^j \cdot \mathbf{B}^j)\mathbf{f} = \mathbf{c} \tag{2.26}$$

$$(\mathbf{B}^j - \mathbf{S}^j \cdot \mathbf{A}^j)\mathbf{f} = \mathbf{d} \tag{2.27}$$

$$(\mathbf{P}^j + \mathbf{Q}^j \cdot \mathbf{S}^j)\mathbf{c} + (-\mathbf{P}^j \cdot \mathbf{T}^j + \mathbf{Q}^j \cdot (\mathbf{1} - \mathbf{S}^j \cdot \mathbf{T}^j))\mathbf{d} = \mathbf{f} \tag{2.28}$$

$$\tag{2.29}$$

Which implies that new filters $\bar{\mathbf{A}}^j$, $\bar{\mathbf{B}}^j$, $\bar{\mathbf{P}}^j$, and $\bar{\mathbf{Q}}^j$ are generated from the original filters as follows:

$$\bar{\mathbf{P}}^j = \mathbf{P}^j + \mathbf{Q}^j \cdot \mathbf{S}^j \tag{2.30}$$

$$\bar{\mathbf{Q}}^j = -\mathbf{P}^j \cdot \mathbf{T}^j + \mathbf{Q}^j \cdot (\mathbf{1} - \mathbf{S}^j \cdot \mathbf{T}^j) \tag{2.31}$$

$$\bar{\mathbf{A}}^j = (\mathbf{1} - \mathbf{T}^j \cdot \mathbf{S}^j) \cdot \mathbf{A}^j + \mathbf{T}^j \cdot \mathbf{B}^j \tag{2.32}$$

$$\bar{\mathbf{B}}^j = \mathbf{B}^j - \mathbf{S}^j \cdot \mathbf{A}^j \tag{2.33}$$

Where $\mathbf{S}^j$ and $\mathbf{T}^j$ are the lifting operators. $\mathbf{S}^j$ takes information from the coarse mesh and sends it to the details, and $\mathbf{T}^j$ takes information from the details and updates the coarse mesh. It is important to note that $\mathbf{S}^j$ and $\mathbf{T}^j$ can be any matrices of the right dimensions, but in spatial applications it is useful to define them in terms of neighborhoods of points in the mesh for locality. [8]

The lifting scheme is a powerful tool that allows us to customize the design of the wavelets and scaling functions for our applications, while at the same time limiting the scope of our optimization to the lifting operators, rather than the entire set of biorthogonal wavelet filters.

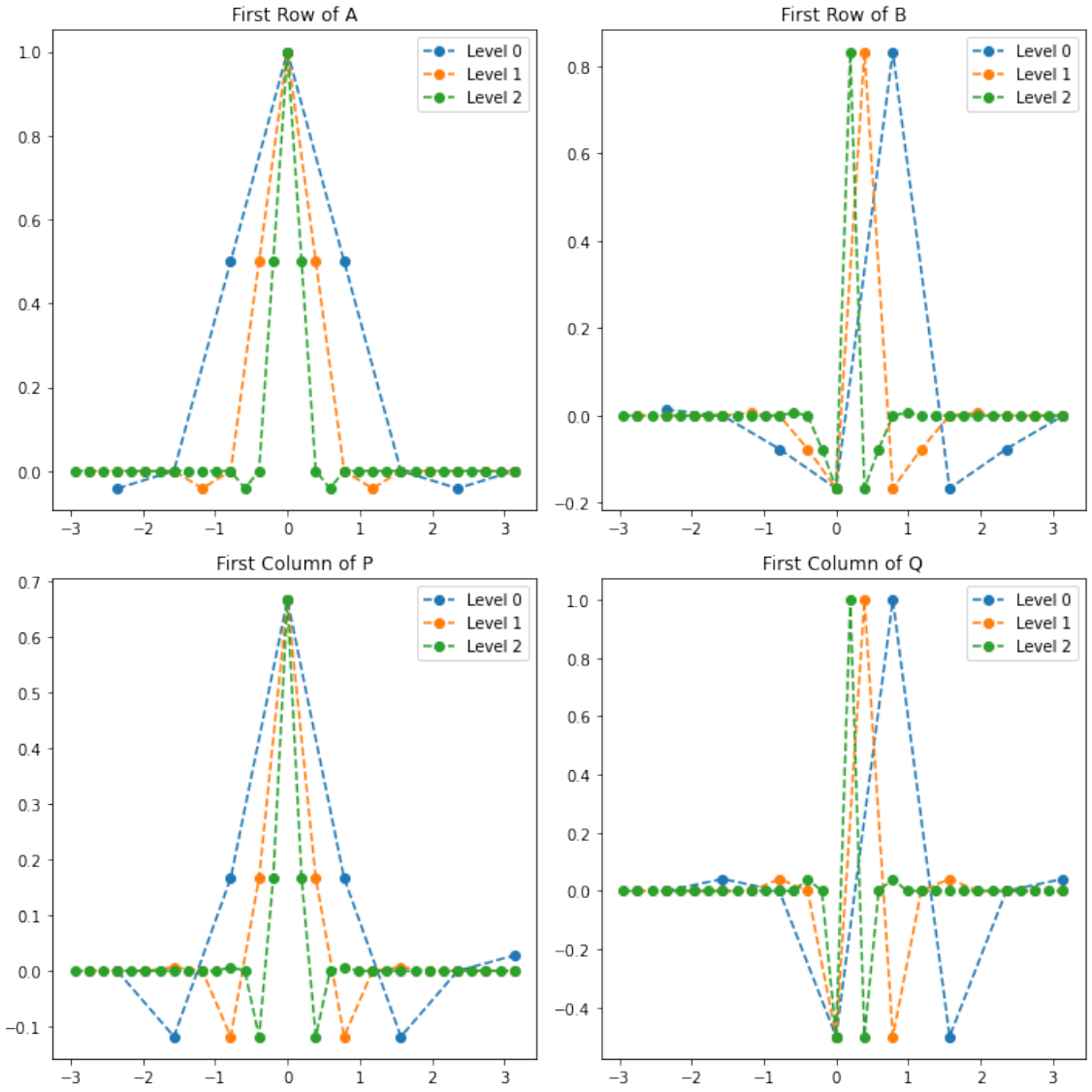


Figure 3: A horizontal cross-section of filters generated by the lifting scheme for the octahedral mesh.

Figure 4: The first row of A on the subdivided octahedral mesh. This illustrates how much signal is sent to the vertex at index 0 in the coarse mesh from all vertices in the fine mesh by applying the filter A

Chapter 3

Methods

3.1 Mesh Data Structure

Many considerations should be made when selecting a data structure for any implementation. A mesh needs both geometric (the locations of vertices in $\mathbb{R}^3$) and topological information (the graph representing the vertex connections) , and there is often a tradeoff between prioritizing either of the two depending on particular considerations like memory consumption, time complexity, etc.

We are particularly interested in performing three operations on a given mesh:

- Subdivision: Geometry and topology dependant. We calculate the midpoint of every edge in the graph and possibly project the result onto the unit sphere, as well as generate new subdivided faces.

- Lifting Scheme: Only topology dependant. We leverage information about the connectedness of each vertex to decide how to send signals defined over the finer mesh into the coarser mesh.

- Query Point: Only geometry dependant. We project any point in $\mathbb{R}^3$ onto the surface of the mesh and return the face that contains it.

In SWF, we expect to precompute the entire set of meshes and filters before they are ever used in the signal domain, so the time complexity of the Subdivision and Lifting Scheme operations on a mesh is less important and does not have to operate at real-time speed. However, our data structure should perform querying as fast as possible, as we expect to interpolate virtual sources over our mesh in real-time.

For the Modified Lifting Scheme introduced in 3.2, the adjacency matrix representation of a graph is particularly useful for the neighborhood-dependent generation of the lifting operator $\bar{\mathbf{T}}^j$. The adjacency matrix can be generated from an edge list at the point of calculating the lifting scheme, and later discarded to free up memory.

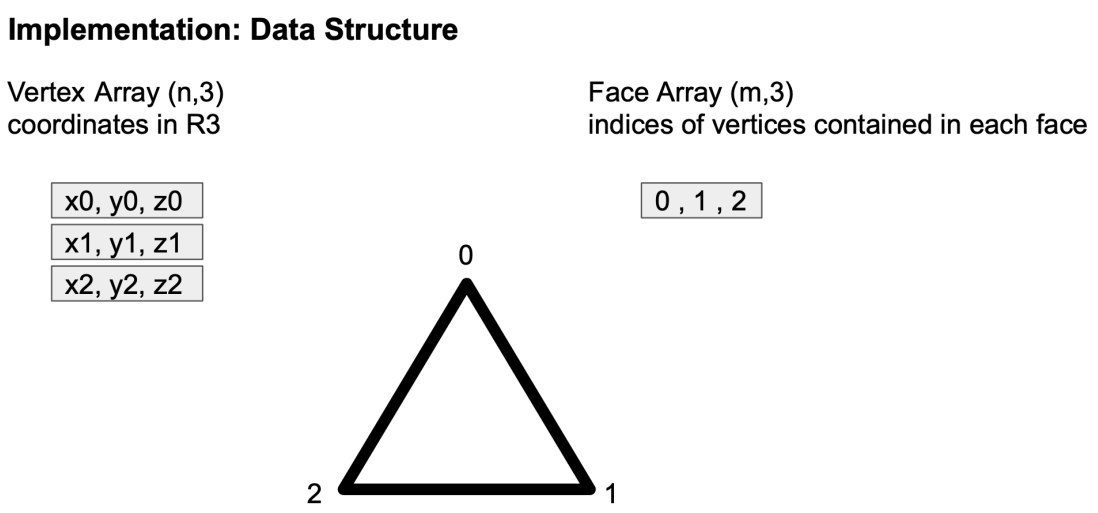


Figure 5: The indexed face list representation of a mesh.

We use an Indexed face list data structure for the general storage of the mesh, and the subdivision operation, as it is standard in many visualization libraries and simple to understand for users from a logical point of view. Further investigation into a more optimal internal data structure like Half-edge or Winged-edge [10] has not been carried out, but could be interesting and/or necessary for commercial applications.

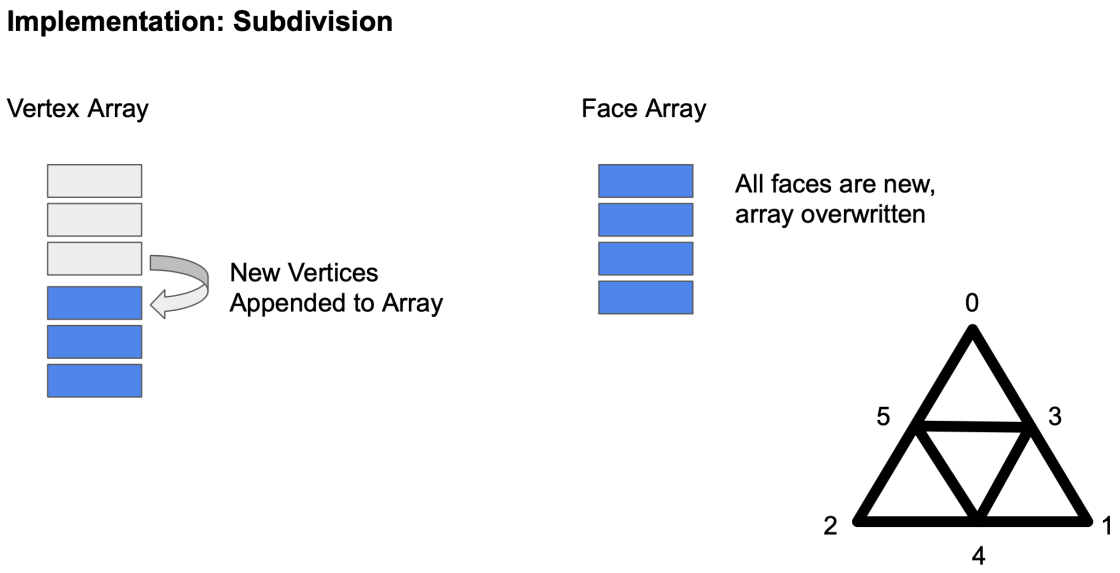


Figure 6: The subdivision method of the mesh with an index face list representation.

3.2 The trivial filter bank

We generate a trivial filter bank as initial base to which we apply the modified lifting scheme, detailed below, generating the second generation wavelets described in Chapter 2. The trivial filter is simply the identity matrix corresponding to the vertices that the filter acts upon concatenated with a matrix of zeros for those vertices which it does not act upon. These graphics make clear how the trivial filters are implemented with the indexed face list.

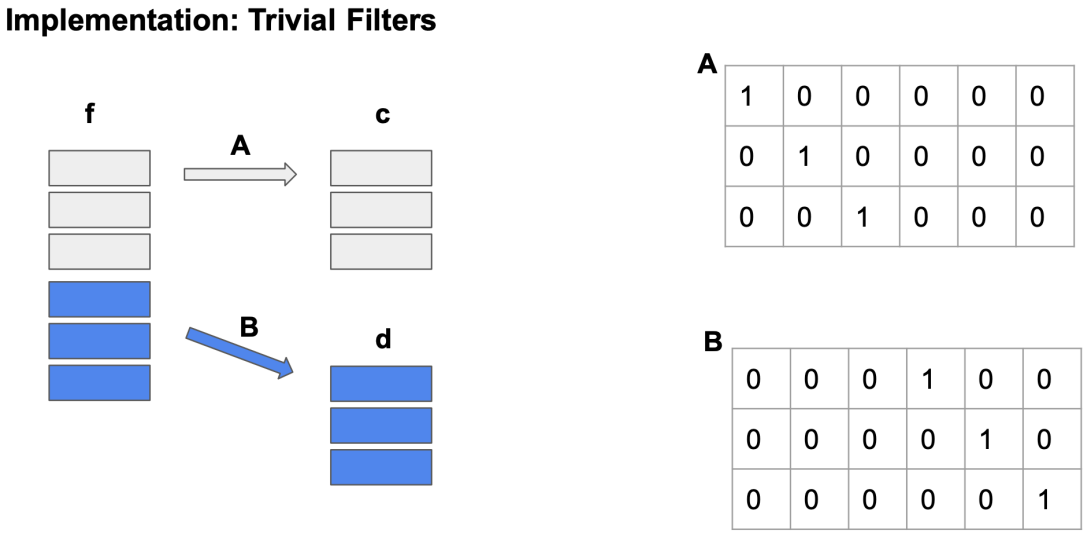
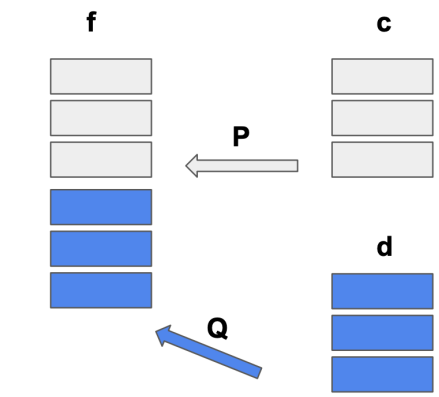


Figure 7: The trivial encoding filters.



P

1	0	0
0	1	0
0	0	1
0	0	0
0	0	0
0	0	0

Q

0	0	0
0	0	0
0	0	0
1	0	0
0	1	0
0	0	1

Figure 8: The trivial decoding filters.

3.3 Modified Lifting Scheme

The original lifting scheme, as described in 2.3.4, is a powerful tool, but its original formulation is not ideal for this spatial audio application. In the spatial audio context, the **A** filter is the most important, as it defines the way we transition from high spatial resolution (i.e. approximating continuous sphere) to low spatial resolution (i.e. approximating our speaker layout). The nonlinearities of the original Lifting Scheme make it poorly suited for constructing a more performant **A** filter, as **A** depends on both **S** and **T**. Fortunately, a modification can be made to the lifting

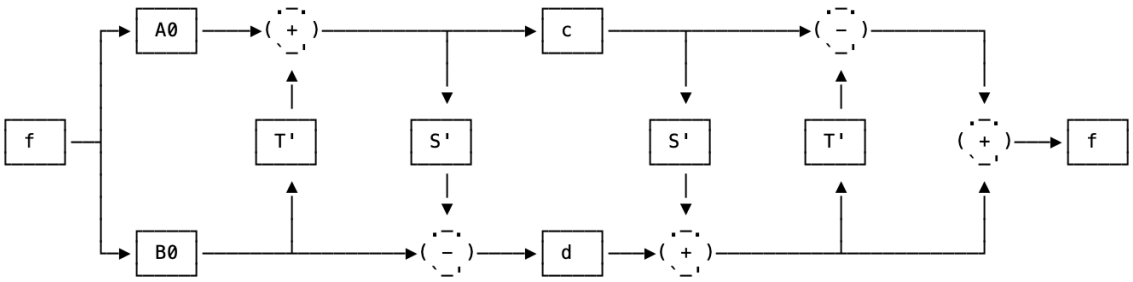


Figure 9: The modified lifting scheme.

scheme so that the **A** filter's linearity is prioritized. This will allow us to optimize the **A** filter directly by way of the $\bar{\mathbf{T}}$ operator.

$\bar{\mathbf{T}}^j$ takes information from the details, and updates the coarse representation. This is the behavior that we want to take advantage of in order to optimize **A**. Thus in the modified lifting scheme, we apply the $\bar{\mathbf{T}}^j$ operator first, adding information from the details (isolated by the initially trivial **B**) weighted by $\bar{\mathbf{T}}^j$ to **A**.

Like before, let's start with the definition of our operators $\mathbf{A}^j$, $\mathbf{B}^j$, $\mathbf{P}^j$, and $\mathbf{Q}^j$ and derive new filters $\bar{\mathbf{A}}^j$, $\bar{\mathbf{B}}^j$, $\bar{\mathbf{P}}^j$, and $\bar{\mathbf{Q}}^j$ by chasing the diagram in Figure 9.

We have:

$$\mathbf{A}^j\mathbf{f} = \mathbf{c} \tag{3.1}$$

$$\mathbf{B}^j\mathbf{f} = \mathbf{d} \tag{3.2}$$

$$\mathbf{P}^j\mathbf{c} + \mathbf{Q}^j\mathbf{d} = \mathbf{f} \tag{3.3}$$

$$\tag{3.4}$$

Applying the steps from the diagram gives:

$$(\mathbf{A}^j + \bar{\mathbf{T}}^j \cdot \mathbf{B}^j)\mathbf{f} = \mathbf{c} \tag{3.5}$$

$$((\mathbf{1} - \bar{\mathbf{S}}^j \cdot \bar{\mathbf{T}}^j) \cdot \mathbf{B}^j - \bar{\mathbf{S}}^j \cdot \mathbf{A}^j)\mathbf{f} = \mathbf{d} \tag{3.6}$$

$$(\mathbf{Q}^j \cdot \bar{\mathbf{S}}^j + \mathbf{P}^j \cdot (\mathbf{1} - \bar{\mathbf{T}}^j \cdot \bar{\mathbf{S}}^j))\mathbf{c} + (\mathbf{Q}^j - \mathbf{P}^j \cdot \bar{\mathbf{T}}^j)\mathbf{d} = \mathbf{f} \tag{3.7}$$

$$\tag{3.8}$$

Which implies that new filters $\bar{\mathbf{A}}^j$, $\bar{\mathbf{B}}^j$, $\bar{\mathbf{P}}^j$, and $\bar{\mathbf{Q}}^j$ are generated from the original filters as follows:

$$\bar{\mathbf{P}}^j = \mathbf{Q}^j \cdot \bar{\mathbf{S}}^j + \mathbf{P}^j \cdot (\mathbf{1} - \bar{\mathbf{T}}^j \cdot \bar{\mathbf{S}}^j) \tag{3.9}$$

$$\bar{\mathbf{Q}}^j = \mathbf{Q}^j - \mathbf{P}^j \cdot \bar{\mathbf{T}}^j \tag{3.10}$$

$$\bar{\mathbf{A}}^j = \mathbf{A}^j + \bar{\mathbf{T}}^j \cdot \mathbf{B}^j \tag{3.11}$$

$$\bar{\mathbf{B}}^j = (\mathbf{1} - \bar{\mathbf{S}}^j \cdot \bar{\mathbf{T}}^j) \cdot \mathbf{B}^j - \bar{\mathbf{S}}^j \cdot \mathbf{A}^j \tag{3.12}$$

3.3.1 Constructing $\bar{\mathbf{T}}^j$

$\bar{\mathbf{T}}^j$ takes information from the details, $\mathbf{d}$, and sends it to the coarse vertices, $\mathbf{c}$. For a fixed point in the details we can define neighborhoods [8] of points in the coarse mesh:

- **Def** Let the k-th neighborhood of some vertex v in the fine mesh, be the set of all vertices w in the coarse mesh connected to v such that the shortest path connecting v and w in the fine mesh has length k.

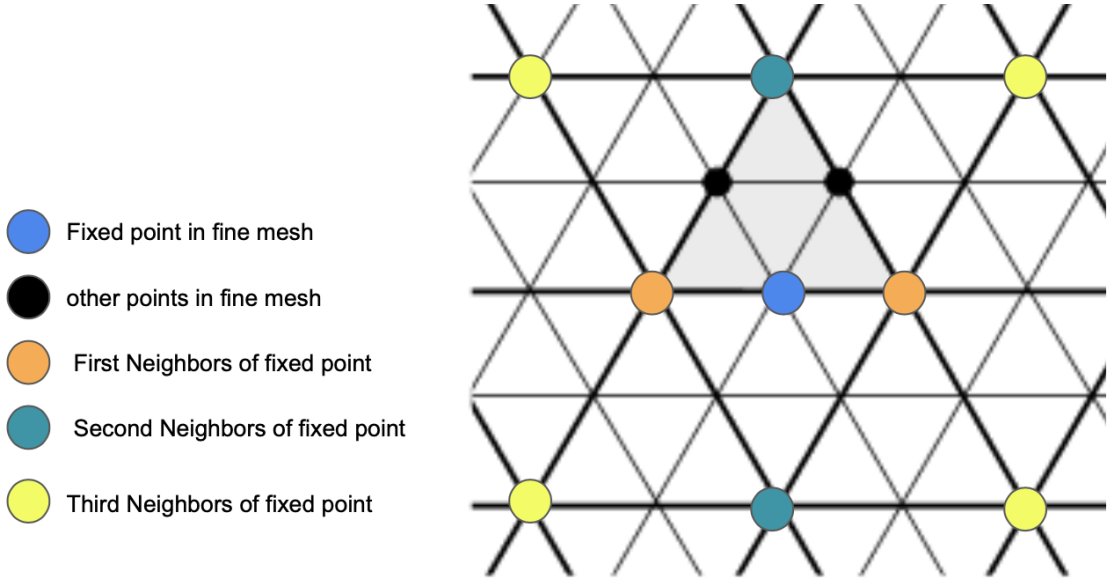


Figure 10: The neighborhoods of a general point on a regular triangular lattice

Using this definition, we can parameterize the construction of $\bar{\mathbf{T}}^j$ and thus the subsequent lifting of $\mathbf{A}^j$ through weights on these neighborhoods.

Let α be the weight on first neighbors, β be the weight on second neighbors, and γ be the weight on third neighbors. As seen in Figure 10, we expect, in a regular triangular lattice, any point in the fine mesh to have two first neighbors, two second neighbors, and four third neighbors in the coarse mesh. Thus, to ensure no information defined over the fine mesh is duplicated or lost due to the application of the encoding filter, it is necessary that the neighborhood weights satisfy the relation:

$$2\alpha, +2\beta + 4\gamma = 1 \tag{3.13}$$

This however, is not always the case, as seen in Figure 11.

On the edge of a half open mesh (approximating a hemisphere), the edge points in the fine mesh have only one second neighbor and two third neighbors. Hence it is

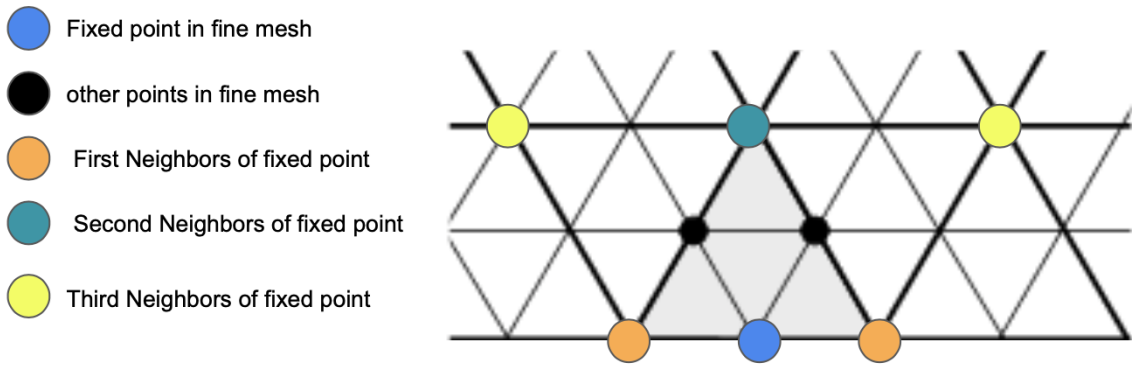


Figure 11: The neighborhoods of an edge point

necessary that we define an adjusted α', β', γ' for each vertex in the fine mesh as follows:

$$\alpha'(v) = \begin{cases} \frac{2\alpha}{s_1} & \text{if } s_1 \neq 0 \\ 0 & \text{if } s_1 = 0 \end{cases} \tag{3.14}$$

$$\beta'(v) = \begin{cases} \frac{2\beta}{s_2} & \text{if } s_2 \neq 0 \text{ and } s_3 \neq 0 \\ \frac{2\beta+4\gamma}{s_2} & \text{if } s_2 \neq 0 \text{ and } s_3 = 0 \\ 0 & \text{if } s_2 = 0 \end{cases} \tag{3.15}$$

$$\gamma'(v) = \begin{cases} \frac{4\gamma}{s_3} & \text{if } s_3 \neq 0 \\ 0 & \text{if } s_3 = 0 \end{cases} \tag{3.16}$$

Where s_1, s_2, s_3 are the number of first, second, and third neighbors of the vertex v. This ensures that, irregardless of the local topology of each point, the weight is adjusted accordingly. Note that, there exist cases with no third neighbors, in which the weight allotted to the third neighbors is absorbed by the second neighbors. It is left to the reader as an exercise to determine why there are no cases without second neighbors.

In practice, we can isolate these neighborhoods using the adjacency matrix. If M

is the adjacency matrix of the directed or undirected graph G, then the matrix M^n (i.e., the matrix product of n copies of M) has an interesting interpretation: the element (i, j) gives the number of (directed or undirected) walks of length n from vertex i to vertex j. If n is the smallest nonnegative integer, such that for some i, j, the element (i, j) of M^n is positive, then n is the distance between vertex i and vertex j.

3.4 Optimization and Evaluation Metrics

To evaluate the performance of a spatial audio format, it is necessary to be able to describe the perceived location of phantom sources objectively. Several models exist to predict phantom source perception. We use a vector model based on properties of the sound field at the listening position. This model has the advantage that the relevant psychoacoustical indicators can be calculated directly from the loudspeaker positions and gains generated by a given spatial audio format, rather than having to be measured like ILD and ITD. (although they *could* be measured with using a coincident arrangement of a pressure microphone [omni-directional] and pressure difference microphones [figure-of-eight] aligned with the axis of the vector space.) [4] The relevant psychoacoustical indicators are defined below:

3.4.1 Total Acoustic Pressure

The total acoustic pressure p under the hypothesis of coherence at the listening position is defined as:

$$p = \sum_{k=1}^{N} s_k \tag{3.17}$$

where N is the number of loudspeakers and s_k is the signal on loudspeaker k.

3.4.2 Total Energy

Similarly, the total energy e under the hypothesis of incoherence at the listening position is defined as:

$$e = \sum_{k=1}^{N} |s_k|^2 \tag{3.18}$$

3.4.3 Acoustic Velocity

The acoustic velocity vector for a phantom source is a normalized plane-wave decomposition of 1st order. [4] A statistical estimator of the acoustic velocity under the hypothesis of coherence is given by:

$$\mathbf{V} = \sum_{k=1}^{N} \frac{s_k \mathbf{u}_k}{p} \tag{3.19}$$

where $\mathbf{u}_k$ is the unit vector pointed in the direction of the loudspeaker k. The velocity vector can be decomposed into two scalar components, namely the longitudinal and transverse components for a phantom source with direction of arrival $\mathbf{s}$.

$$V_l = \mathbf{V} \cdot \mathbf{s} \tag{3.20}$$

$$V_t = ||\mathbf{V} \times \mathbf{s}|| \tag{3.21}$$

$$\text{such that, } ||\mathbf{V}|| = \sqrt{{V_t}^2 + {V_l}^2} \tag{3.22}$$

3.4.4 Sound Intensity

The sound intensity vector indicates the direction of the sound field at the listening position for a phantom source. [4] A statistical estimator of the sound intensity under the hypothesis of incoherence is given by:

$$\mathbf{I} = \sum_{k=1}^{N} \frac{|s_k|^2 \mathbf{u}_k}{e} \tag{3.23}$$

Similar to velocity, the Intensity vector can be decomposed into its longitudinal and transverse components.

$$I_l = \mathbf{I} \cdot \mathbf{s} \tag{3.24}$$

$$I_t = ||\mathbf{I} \times \mathbf{s}|| \text{such that, } ||\mathbf{I}|| = \sqrt{{I_t}^2 + {I_l}^2} \tag{3.25}$$

3.5 Energy Normalization

By construction of the modified lifting scheme, SWF guarantees that the total acoustic pressure is conserved, i.e. for all possible phantom source locations, the sum of the gains at the destination layout is equal to one. This is ideal assuming that we have coherent sources at all loudspeakers, however many use cases for spatial audio exist with incoherent sources. A better normalization exists for incoherent sources which prioritizes unit total energy at all possible phantom source locations. We derive new loudspeaker gains g' as:

$$g'_i = \frac{g_i}{\sqrt{\sum_{k=1}^{N} {g_k}^2}} \tag{3.26}$$

3.6 Optimization

The psychoacoustic indicators we use for evaluation extend nicely as metrics for optimization of the $\mathbf{A}^j$ via the modified lifting scheme by way of the parameterization of $\bar{\mathbf{T}}^j$. In particular, we want:

- the total acoustic pressure to be preserved by the encoding filter, which is guaranteed through the relation $2\alpha, +2\beta + 4\gamma = 1$, and the relevant adjustments for the local topology of each vertex.
- Ideally, for a point source encoded to the coarse mesh from any vertex in the

fine mesh: the transverse velocity, V_t should be as close to zero as possible, and the longitudinal velocity, V_l should be as close to one as possible.

These statements combined give rise to a cost function:

$$Cost = \sum_{i=0}^{n} w_l(V_{l_i} - 1)^2 + w_t V_{t_i}^2 \tag{3.27}$$

where n is the number of vertices in the fine mesh, w_l, w_t are weights for the longitudinal and transverse components, respectively.

We have three parameters, α, β, γ and one constraint, $2\alpha, +2\beta + 4\gamma = 1$, and a cost function to minimize. This minimization can be computed with any optimization library of choice, but in this implementation we use *scipy.optimize.minimize*.

3.7 Evaluation

The evaluation is based on a horizontal panning of a constant signal over a 7.1.4 layout computed with a virtual source on the unit circle at $z = 0$. At each sample in the panning, we record all the channel gains and calculate the total acoustic pressure, total energy, and longitudinal and transverse intensities for each of the algorithms.

Ideally, the transverse intensity, which measures the proportion of energy that is coming from directions other than the intended direction-of-arrival should be zero. Similarly, in an ideal case, the longitudinal intensity, which measures the proportion of energy that is coming from the intended direction-of-arrival should be one. It is important to note that this ideal will never be realized with any finite speaker array.

All of the gains have been renormalized for unit energy as described in Section 3.5 for every algorithm. This ensures that the performance of each algorithm is presented on a consistent scale.

Chapter 4

Results

4.1 Objective Evaluations

In this section, we perform an objective comparison between SWF (defined on 7.0.4 base mesh subdivided to level 2 with the trivial decoding and optimization) and other state of the art spatial audio algorithms. In the figures, SWF is always presented on the left, and the other algorithm on the right.

4.1.1 VBAP

VBAP, as a generalization of the tangent law for amplitude panning, represents the most directional panning possible for a given layout. In figure 13, we see issues with L/R symmetry in VBAP in due to the triangulation of the irregular mesh. This is especially evident for a virtual source around π radians. SWF, although generated from the same base mesh, handles L/R symmetry in the horizontal plane through the subdivision, as can be seen in figure 12.

The optimized filters of the SWF format generate negative gains, as can be seen in figure 14. This is an issue because counterphase components can be heard outside of the listening sweet spot, and can contribute to changes in timbre as a virtual source moves. This explains the smaller peaks in the total acoustic pressure relative to VBAP, but is generally not a desired result. The loudspeakers firing in counterphase

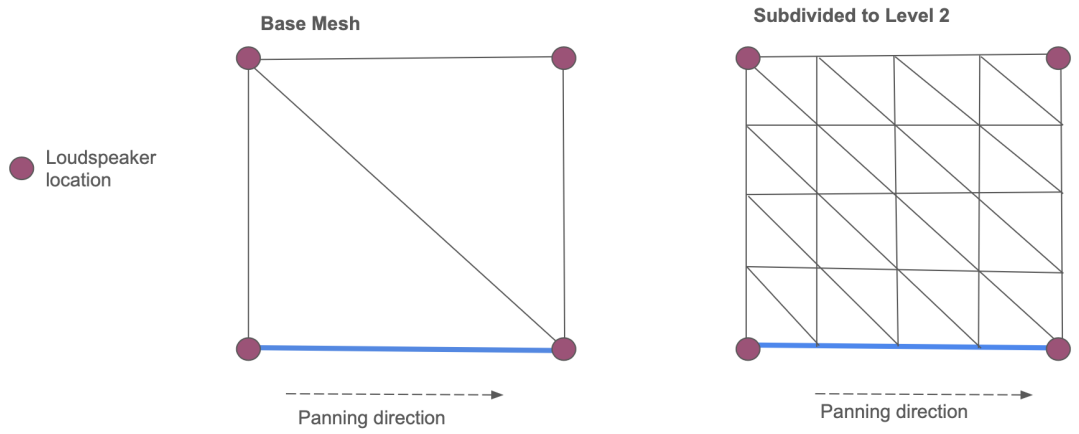


Figure 12: A look at L/R symmetry in the base vs subdivided mesh – VBAP computes trilinear interpolation over the base, SWF computes trilinear interpolation over the subdivision and then scales down to the base via the filters described in 3.2.1

do contribute to a less jumpy panning experience when listening in the sweet spot for SWF, but for applications with a large potential listening area this is not ideal.

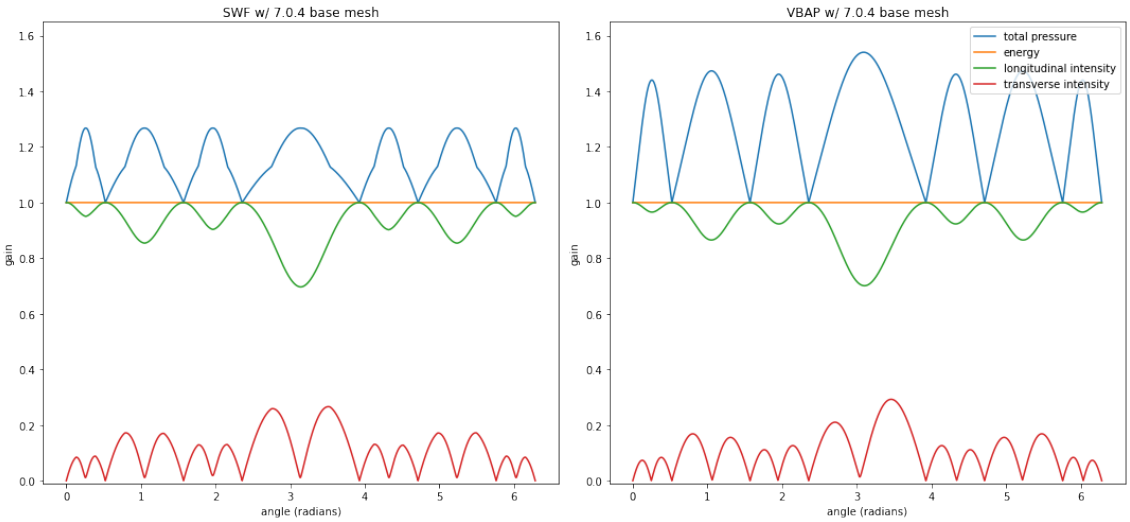


Figure 13: Comparison of SWF 7.0.4 and VBAP psychoacoustic indicators

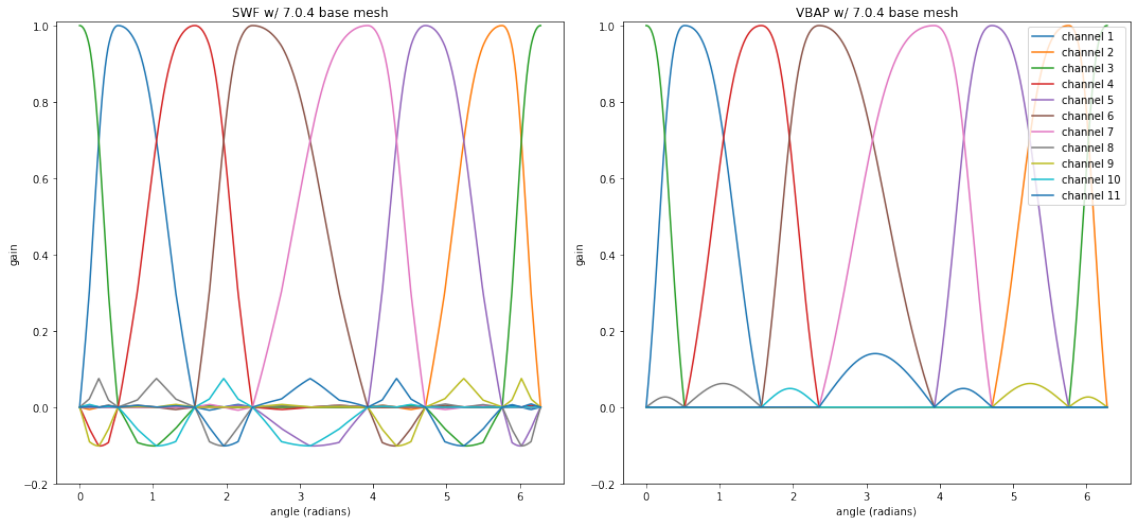


Figure 14: Comparison of SWF 7.0.4 and VBAP gains

4.1.2 3rd Order Ambisonics with various decodings

The horizontal panning encoded to 3rd Order Ambisonics is decoded using three techniques: AllRAD basic [11], AllRAD with maxrE weights, and IDHOA [12]. As can be seen in the figures, All of the Ambisonics-based pannings generate negative gains, although those generated with maxrE weights are the least dramatic. No Ambisonics-based panning handles L/R symmetry correctly. Ambisonic decoders never pan entirely to a single channel, which contributes to very smooth-sounding motion in the result, but without a very clear spatial resolution, often with the virtual source completely dispersed through the room.

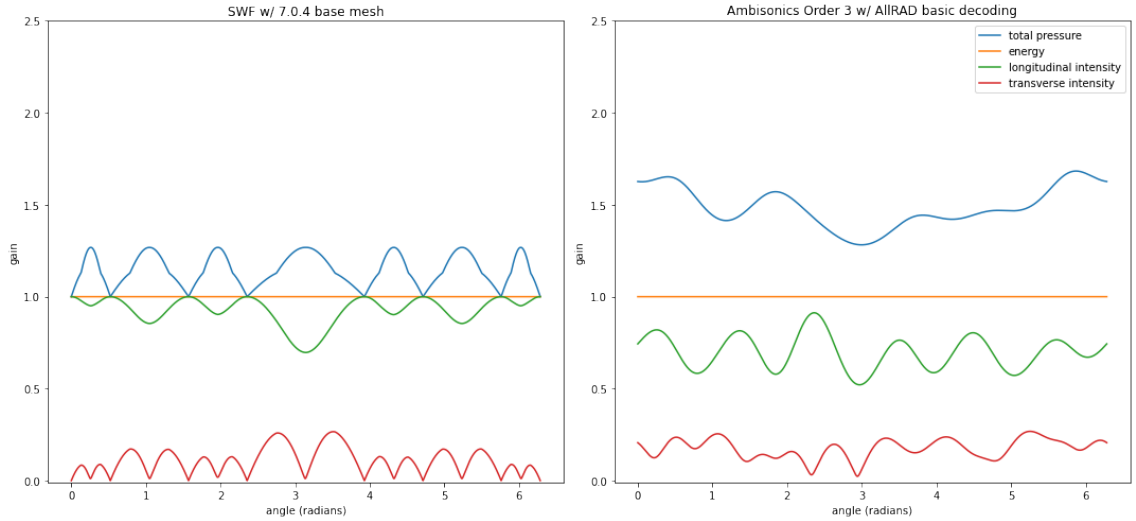


Figure 15: Comparison of SWF 7.0.4 and Ambisonics order 3 – AllRAD Basic decoding psychoacoustic indicators

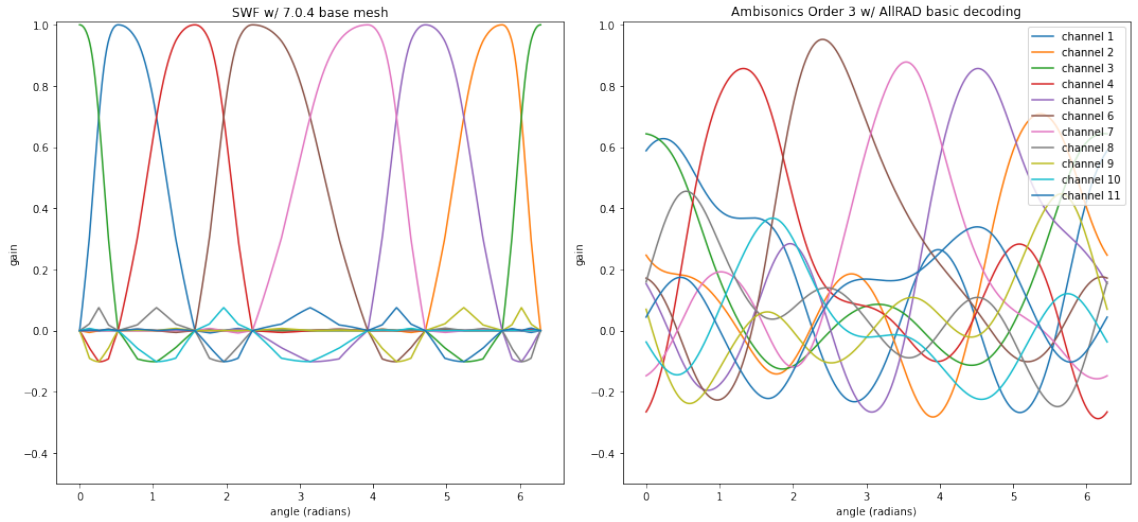


Figure 16: Comparison of SWF 7.0.4 and Ambisonics order 3 – AllRAD Basic decoding gains

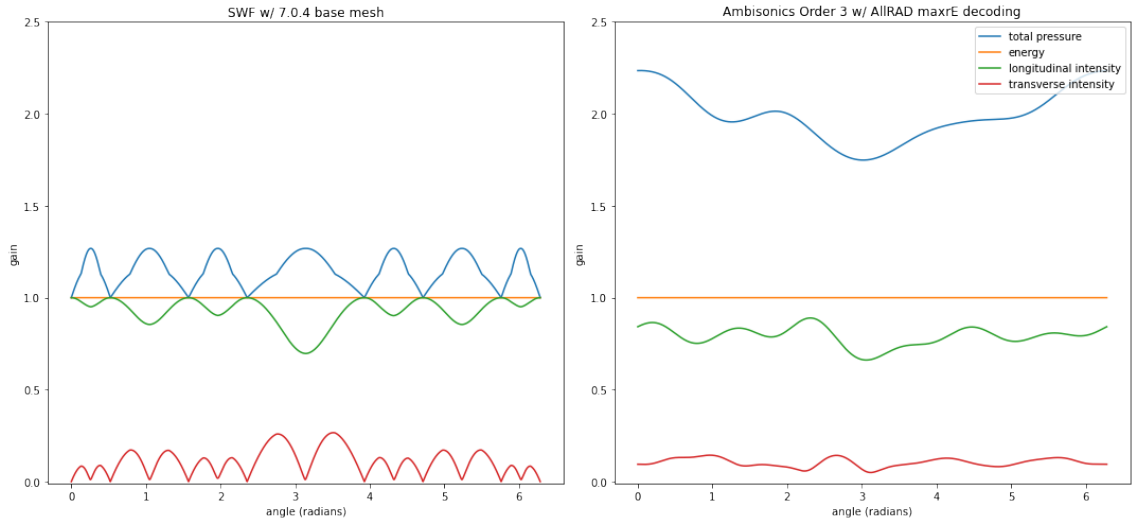


Figure 17: Comparison of SWF 7.0.4 and Ambisonics order 3 – AllRAD maxrE decoding psychoacoustic indicators

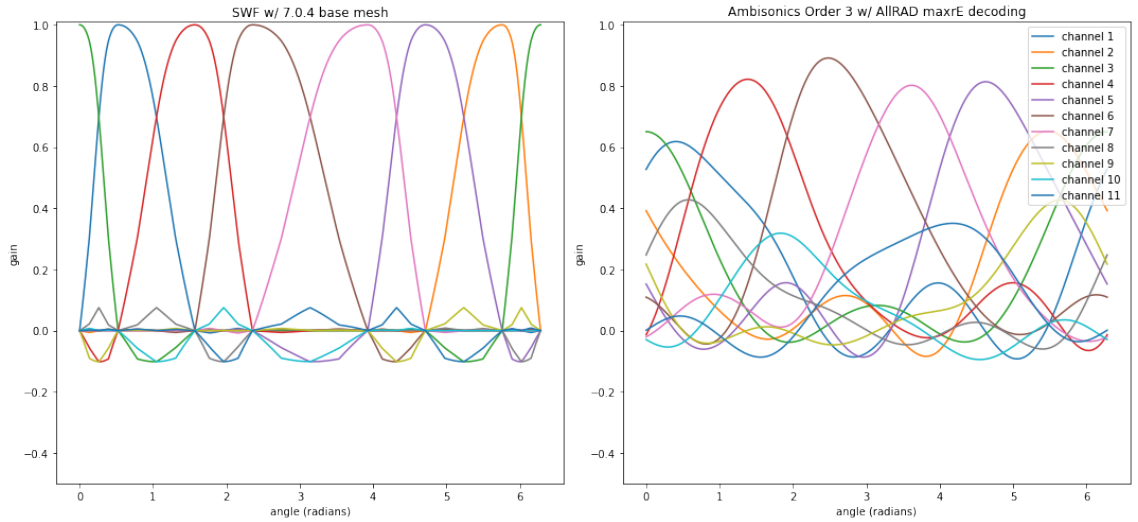


Figure 18: Comparison of SWF 7.0.4 and Ambisonics order 3 – AllRAD maxrE decoding gains

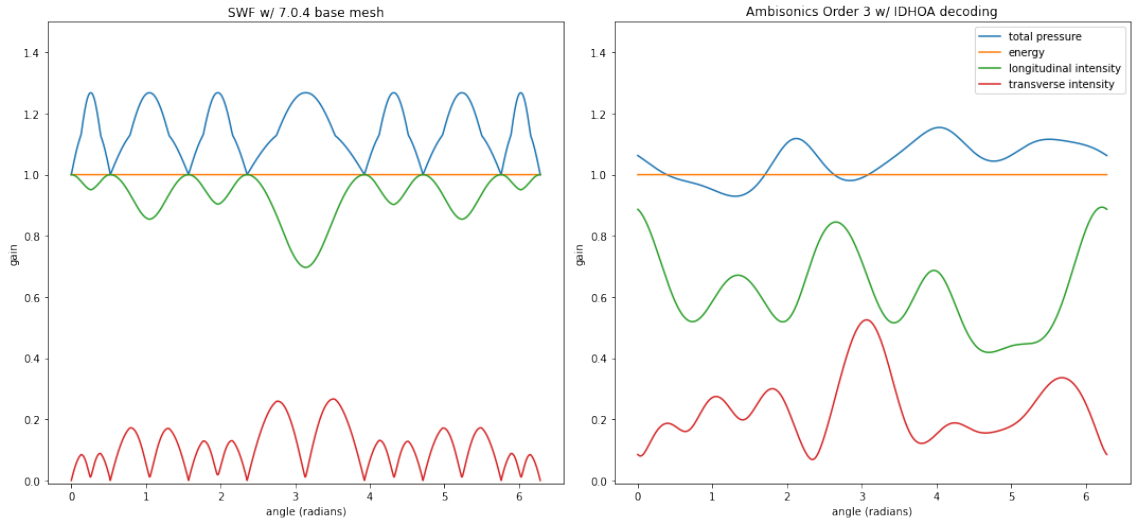


Figure 19: Comparison of SWF 7.0.4 and Ambisonics order 3 – IDHOA decoding psychoacoustic indicators

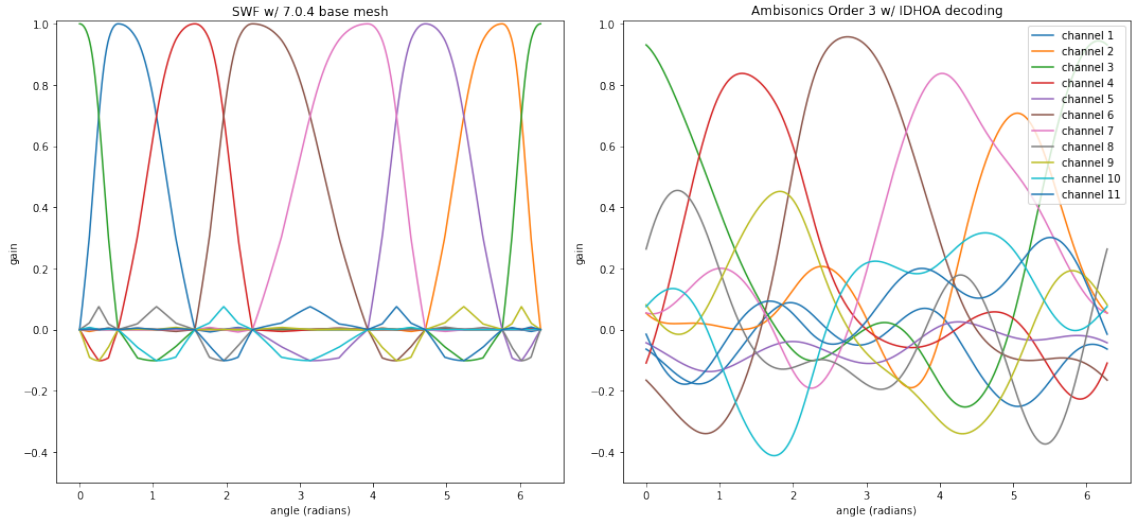


Figure 20: Comparison of SWF 7.0.4 and Ambisonics order 3 – IDHOA decoding gains

4.2 The Library

The major result of this work is an open-source python library implementing SWF with the modified lifting scheme and psychoacoustic optimization. The library has minimal dependencies; only numpy and scipy are required for the function of all the core objects. Optionally, matploblib and plotly are used in visualization helper functions, and python-osc for interfacing with the Max Patch. Michael Dawson-Haggerty's Trimesh[13] library provided many crucial building blocks for the python implementation of SWF. More detailed documentation on the usage of the library, as well as all the code can be found on GitHub. [14]

With this library we have built and tested various SWF formats which are available as preset variables in the constants.py file. The SWF formats available are those based on the regular octahedral mesh, a mesh approximating a 7.0.4 layout, a mesh approximating a 3.0.1 layout, and an interesting transcoding mesh structure that includes a subdivison mesh for the formats: 3.0.0, 5.0.0, 5.0.2, 7.0.4, 9.0.6, and 11.0.8 as explained in the next section.

4.3 Transcoding Mesh Structures

We use loop subdivision to uniformly increase the spatial resolution of a mesh, but a subdivision step does not neccessarily have to be uniform, we can add just a few vertices or focus our subdivision in a particular area of the mesh. This is useful in that, SWF can be used to optimally transcode between different layouts. Let's consider speaker configurations using the Dolby Atmos channel notation standard: a layout denoted $n.l.k$ has n channels in the horizontal plane, k overhead channels, and l LFE channels. For example, 7.1.4 is a layout with five horizontal channels, four overhead and one subwoofer.

Using this notation, we construct a matrix of intersecting surround sound formats, as seen in Figure 21. Moving down the vertical axis, we add horizontal channels symmetrically, and moving right along the horizontal axis, we add overhead channels symmetrically.

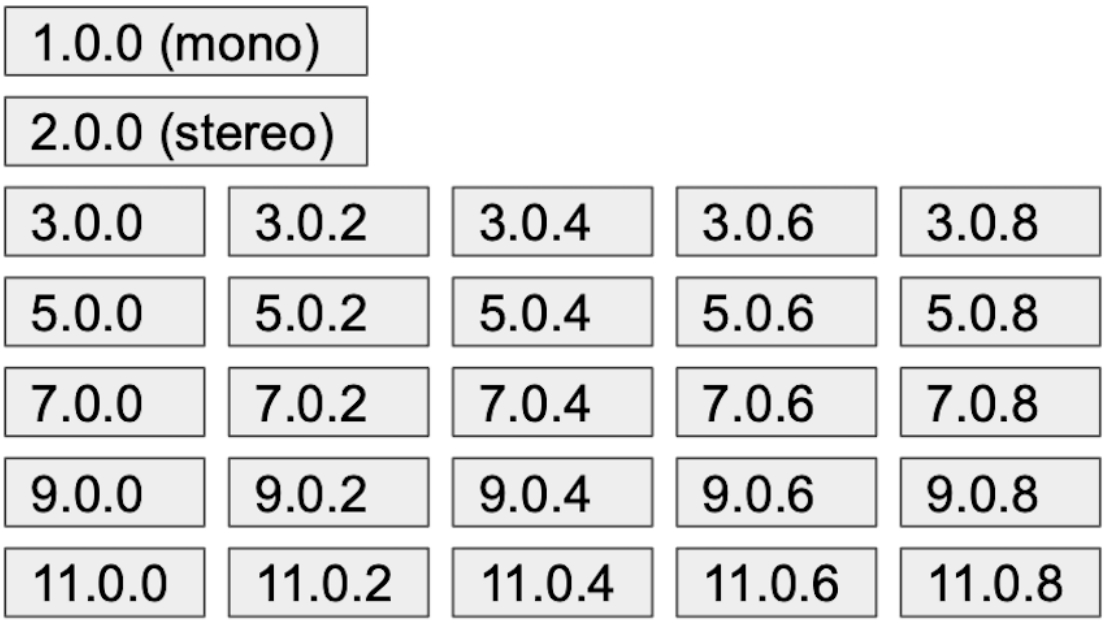


Figure 21: the L/R symmetric subsets of 11.0.8

This model can be used to construct a transcoding SWF between any two (or more) formats by fixing a particular path through the subdivisions, and using the relevant meshes to generate appropriate A,B,P,Q filters, as seen in Figure 22. This highlighted path is the preset included in the library, but any other set of intersecting formats could be used to construct a similar SWF. At any point, we can transition to the loop subdivision scheme to generate subdivisions uniformly. This result allows a user to downmix media produced for 7.0.4 to 3.0.0, for example. Or upmix media produced for 5.0.2 for playback in a 9.0.6 system.

In this structure, we optimize for each format, moving from most dense to least dense and fixing the parameters α, β, γ for each level. In the highlighted example, we would first fix the parameters for transitioning through the loop subdivision to the 11.0.8 format, then optimize new α, β, γ to transcode from 11.0.8 to 9.0.6 fixing the generated A filters from the previous step. We continue this proccess until the coarsest level has been reached.

The result is a SWF format with 6 levels that approximate different speaker layouts, and that can be subdivided uniformly from the densest mesh to gain more spatial resolution.

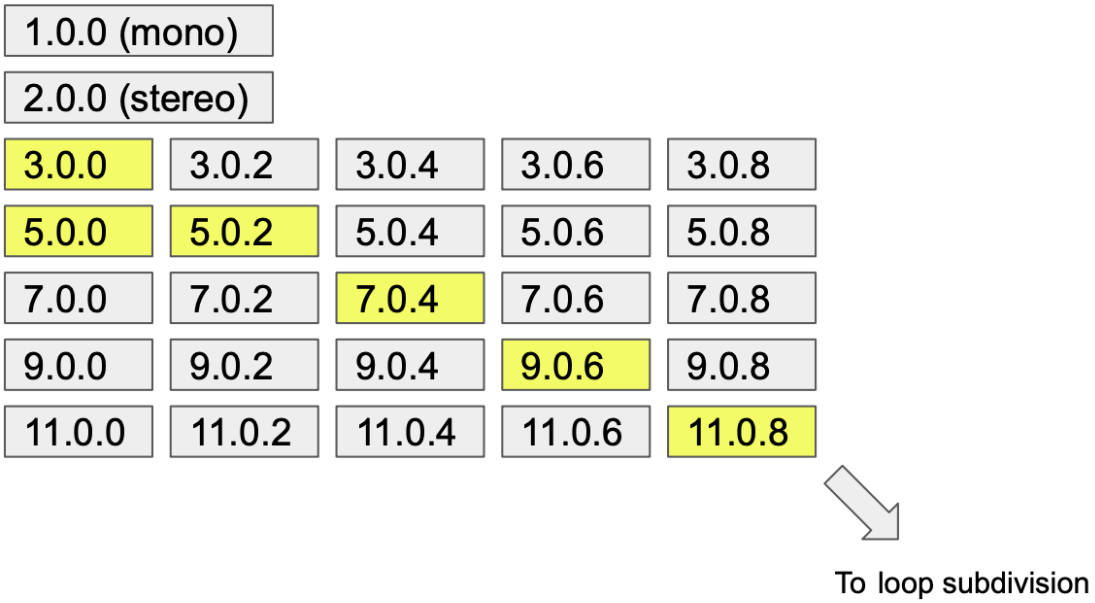


Figure 22: a path of subsets of 11.0.8

4.4 Max Patches

The majority of the listening tests for this library took place in the 7.1.4 surround studio at Dolby Laboratories in Barcelona. The Max patches I have designed are preconfigured to work with this layout, but with some adjustment can easily be adapted to other layouts. There are two patches, which can both be found in the GitHub repository linked in the appendix.

The first patch can be used to do comparative listening tests of 5 different pre-computed horizontal pannings over a 7.1.4 layout. The available options are SWF, VBAP, or Ambisonics 3rd order decoded with IDHOA, AllRAD basic, or AllRAD maxrE algorithms.

The second patch can be used to compute a SWF encoding of a virtual source in realtime. The patch is quite minimal and designed for only one virtual source, but could easily be extended to an arbitrary number of sources by overlaying an object-based interface like the one included in the ICST Ambisonics toolkit for Max. The patch queries python over OSC with the location of the virtual source and expects to receive the interopolation over the finest level of mesh in return over OSC. The

encoding from the finest to coarsest level happens in the max patch, but if there are hardware limitations on the number of channels that can be transmitted I would recommend computing the encoding of the interpolation result in python before returning to Max.

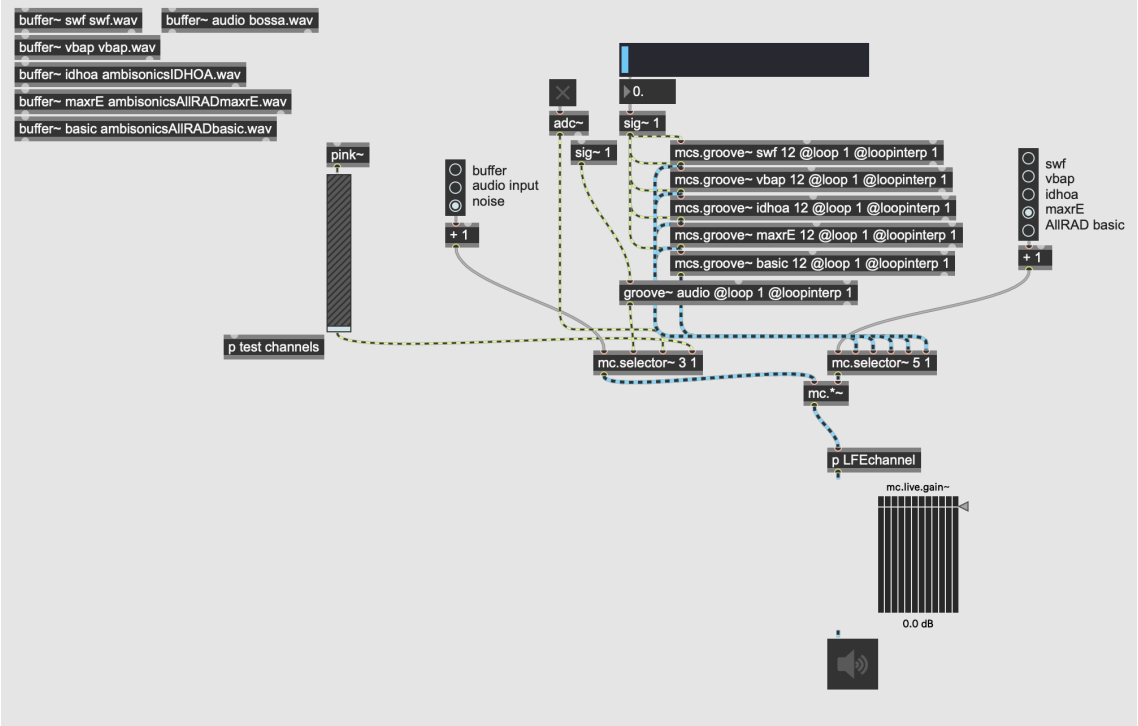


Figure 23: A patch for comparing precomputed horizontal pannings over 7.1.4 in SWF,VBAP, and Ambisonics with three different decodings

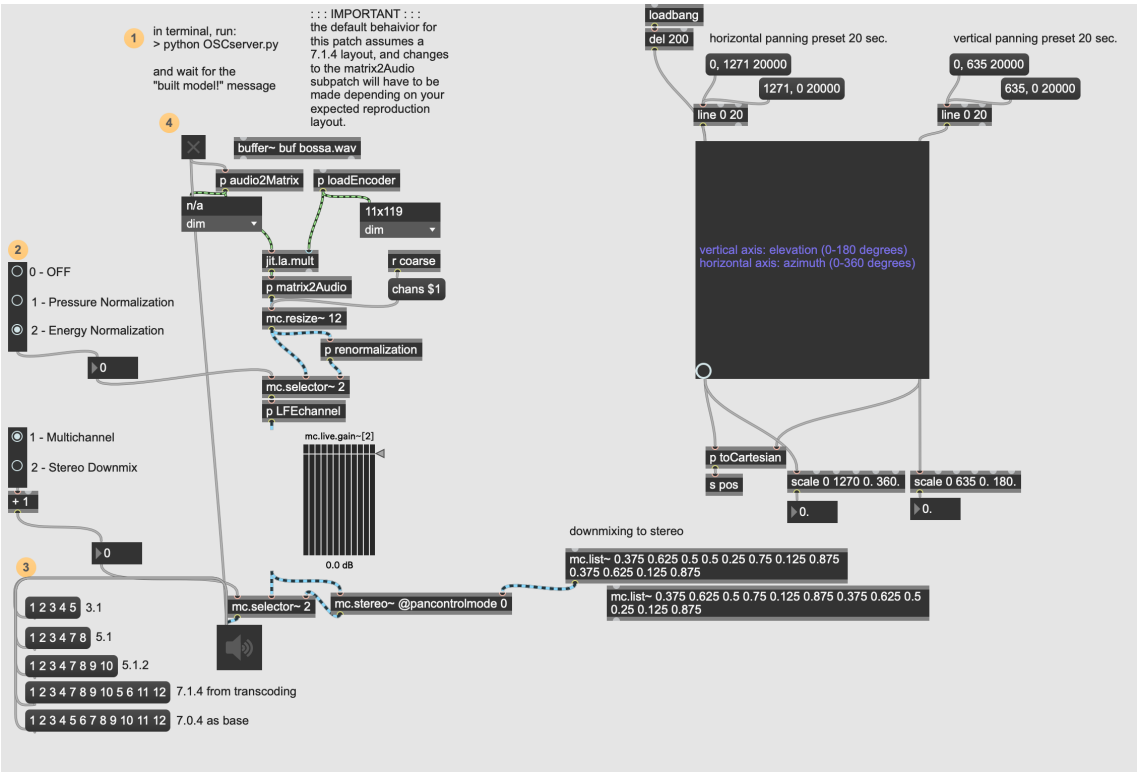


Figure 24: the realtime panning interface for a single virtual source

Chapter 5

Discussion and Conclusions

5.1 Conclusion

In this work, we have been able to develop a version of SWF that implements a method to build an arbitrary wavelet representation on an arbitrary mesh. We make use of a modified lifting scheme to optimize the interpolating scaling functions for optimal playback reproduction, and we have demonstrated its functionality and competitiveness with state-of-the-art spatial audio algorithms on a 7.1.4 layout. The python library is flexible enough to support any layout however, as has been demonstrated with a Spherical Wavelet Format that naturally interpolates between standard surround sound formats (11.1.8,9.1.6,7.1.4,etc.). The library is intended to be used with the trivial decoding from the coarsest level of mesh, but can also be decoded using other strategies from less coarse representations. This allows the user to decide what bandwith they want to transmit at based on the limitations or needs of their system.

5.2 Discussion

Ultimately, SWF offers advantages in that it is channel-based, like Ambisonics. Depending on the computing resources available, SWF can be transmitted at arbitrary resolutions, from the finest level of detail (which can reach orders of 800 channels

depending on the number of subdivisions), to any intermediate coarser level, to the same number of channels as loudspeakers. This offers advantages in terms of flexibility over object-based interfaces, while still being adaptable to any layout. Unlike Ambisonics, this formulation of SWF does not require a decoding step (although it can be used).

5.3 Future Work

Further work should be done to introduce penalties on negative gains in the cost function used to optimize the encoding filter. This will limit out-of-phase components especially when using SWF with multiple virtual sources.

Additionally, optimization should be carried out on the decoding filter, $\mathbf{P}^i$, which takes coarse information $\mathbf{c}^i$ to the next finest level $\mathbf{c}^{i+1}$. This optimization should be carried out discarding the details $\mathbf{d}^i$ to ensure the best possible reconstruction without the use of the details.

It's worth noting that the use of triangular meshes and trilinear interpolation is for convenience. The mechanics of SWF are not tied to this triangularity. Future work could include generalizing to meshes based on other polygons – or even mixed-polygon meshes – with appropriate finite subdivision rules and interpolation at the finest level via a generalized form of barycentric coordinates for irregular, convex n-sided polygons. [15] This could let us define multiresolution formats that prioritize multiple planes of symmetry, or symmetries much more complex than L/R.

We could even design meshes to fit spaces that don't approximate the sphere, which could be useful for spatial audio installations in architechtural spaces that were not necessarily designed for acoustic listening like botanical gardens, art galleries, classrooms, or other spaces where loudspeaker layouts wouldn't necessarily have spherical symmetry or a dedicated listening position.

Within the spherical context, subgraphs of the mesh and their subdivisions isomorphic to those presented in Figure 25, can create symmetry issues when a virtual source is located directly in the middle of the figure, along the diagonal axis. The

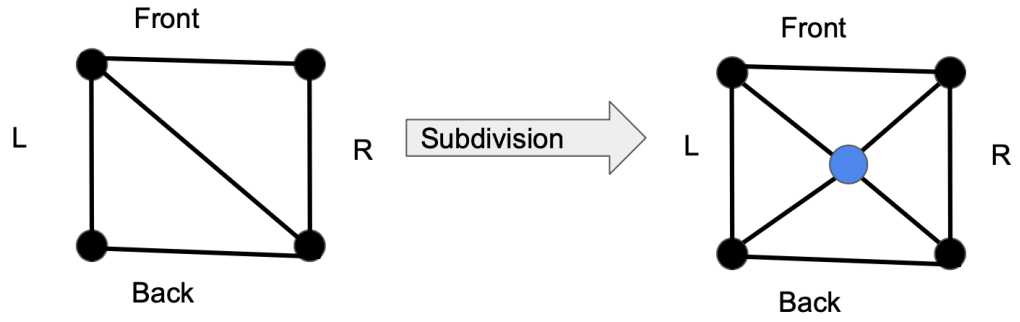


Figure 25: An alternative subdivision method that prioritizes L/R symmetry for certain subgraphs oriented as notated.

SWF formulation presented in this report will always prefer a diagonal panning between two loudspeakers in this scenario, when a more symmetric-sounding result would be a 4-way interpolation between all loudspeakers. An algorithm could be developed to identify these subgraphs and compute a non-uniform subdivision step to impute vertices that prioritize L/R (and thus ultimately binaural) symmetry.

5.4 Final Thoughts

Incredibly interesting problems in graph theory, geometry, computer graphics, and acoustics arise in studying and developing SWF. I think any young researcher interested in spatial audio and applied mathematics would find here a rich selection of directions to explore this work further. I hope that spatial audio practitioners and industries will take interest in how SWF could serve their purposes. I am so grateful to my advisors Daniel and Davide for shepherding me through their invention and none of my work would have been possible without them.

List of Figures

Bibliography

[1] Scaini, D. Wavelet-based spatial audio framework : from ambisonics to wavelets: a novel approach to spatial audio. *TDX (Tesis Doctorals en Xarxa)* (2019).

[2] Scaini, D. & Arteaga, D. Wavelet-based spatial audio format. *AES: Journal of the Audio Engineering Society* **68** (2020).

[3] Eguinoa, R., Martin, R. S., Arteaga, D. & Scaini, D. Subjective evaluation of the localization performance of the spherical wavelet format compared to ambisonics (2021).

[4] Frank, M. Phantom sources using multiple loudspeakers in the horizontal plane. *University of Music and Performing Arts Graz Dissertation* 1–119 (2013). URL `http://www.kug.ac.at/fileadmin/media/iem/projects/2013/frank_matthias_diss.pdf`.

[5] Cheng, C. I. & Wakefield, G. H. Introduction to head-related transfer functions (hrtfs): Representations of hrtfs in time, frequency, and space. *AES: Journal of the Audio Engineering Society* **49** (2001).

[6] Malham, D. G. Homogeneous and non-homogreneous surround sound systems (1999).

[7] Zotter, F. & Frank, M. *Ambisonics: A Practical 3D Audio Theory for Recording* (2019).

[8] Schroder, P. & Sweldens, W. Spherical wavelets: efficiently representing functions on the sphere (1995).

[9] Sweldens, W. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis* **3** (1996).

[10] Tobler, R. F. & Maierhofer, S. A mesh data structure for rendering and subdivision (2006).

[11] Zotter, F. & Frank, M. All-round ambisonic panning and decoding. *AES: Journal of the Audio Engineering Society* **60** (2012).

[12] Scaini, D. & Arteaga, D. Decoding of higher order ambisonics to irregular periphonic loudspeaker arrays. vol. 2014-January (2014).

[13] Dawson-Haggerty et al. trimesh. URL `https://trimsh.org/`.

[14] Samuel Narvaez et al. Swf. URL `https://github.com/SamuelNarvaez/SWF`.

[15] Meyer, M., Barr, A., Lee, H. & Desbrun, M. Generalized barycentric coordinates on irregular polygons. *Journal of Graphics Tools* **7** (2002).