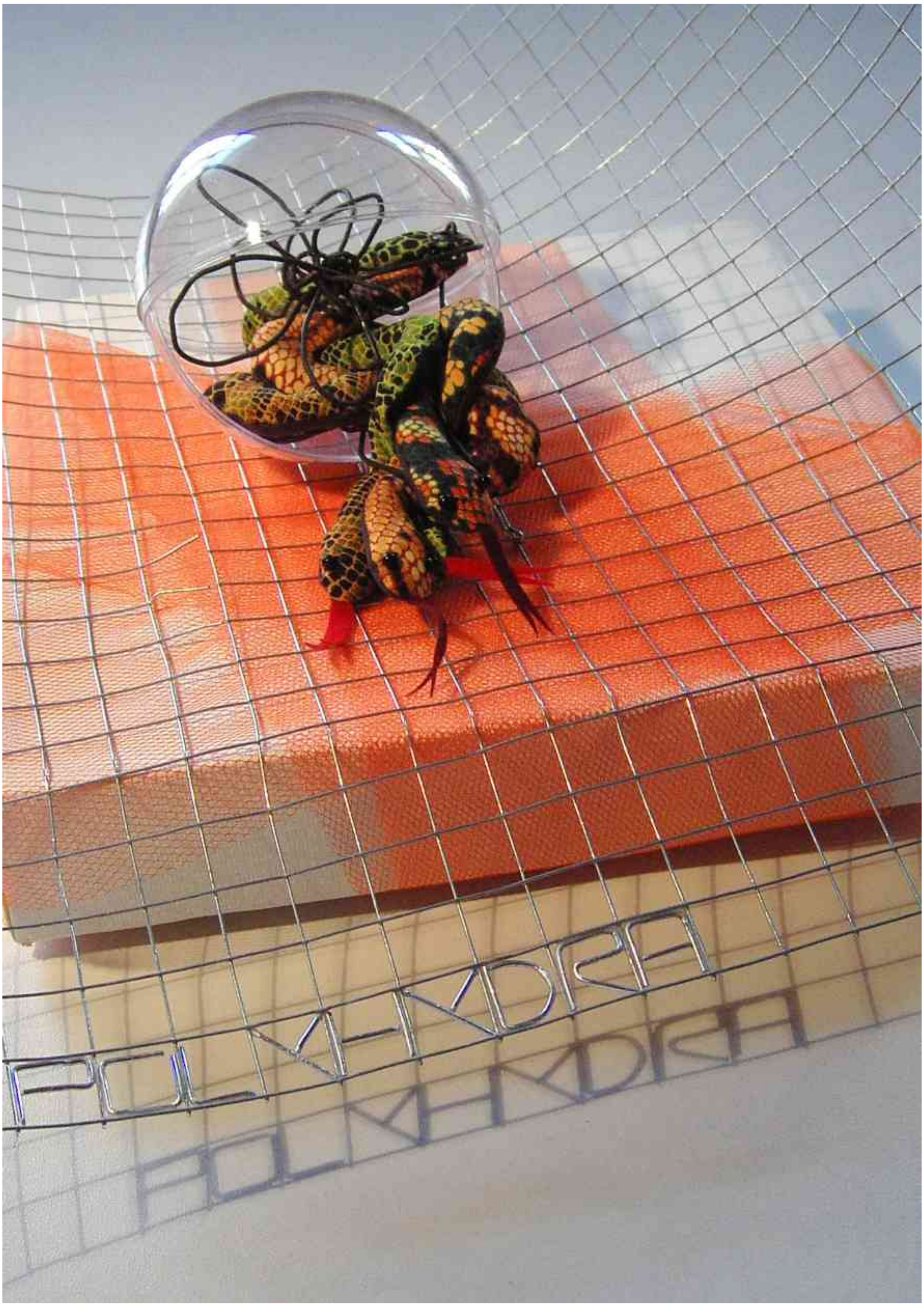
Periphonic Sound Spatialization in Multi-User Virtual Environments

Florian Hollerweger

corrected version, March 14th 2006

Institute of Electronic Music and Acoustics (IEM),
Center for Research in Electronic Art Technology (CREATE)

Advisor: O.Univ.-Prof.Mag. DI Dr. Robert Höldrich
Co-advisors: Stephen Travis Pope, Univ.Ass. DI Dr. Alois Sontacchi


POLYANDRA

Polyhydra by Manuela Meier 2006.
Photograph by Romana Rust.

Für Manuela.

Many thanks to Stephen Travis Pope and Robert Höldrich for supporting this project, and to Alois Sontacchi, Thomas Musil and Rudolf Rabenstein for providing lots of valuable information.

Only friends like Graham Wakefield, Jorge Castellanos and Will Wolcott can make the work on a thesis such an enjoyable task. The same is true for Lance Putnam and David Romblom, who always had a spare ear and opinion. August Black and Wesley Smith made California a home. And why aren't there more people like Elizaveta Pachepsky?

flo.H, March 1st 2006

Abstract

As a collaboration with the Center for Research in Electronic Arts Technology (CREATE) at the University of California at Santa Barbara (UCSB), this thesis addresses sound spatialization in the AlloSphere, a research environment currently under construction as a shared facility of the California Nanosystems Institute and UCSB's Media Arts and Technology program. This spherical construction, which has a diameter of approximately 10 meters and accommodates ten to twenty people on a centered bridge, features interactively controllable projection of visual and aural data on its entire inner surface.

The theoretical foundations of spatial hearing and a short history of periphonic sound spatialization are presented. An overview of spatialization techniques (Vector Base Panning, Higher Order Ambisonics, holophonic approaches) is given. A Higher Order Ambisonic C++ library for the CSL framework (CREATE Signal library) has been developed together with J. Castellanos and G. Wakefield, which can be used to set up Ambisonic systems up to third order on distributed computing systems. A periphonic loudspeaker layout design strategy is presented. It bases on a hybrid approach of geodesic spheres and minimal energy configurations, allowing to balance the design process between localization homogeneity and psychoacoustical layout optimization. An according Matlab library has been developed to generate loudspeaker layouts and evaluate them in combination with different spatialization algorithms.

Keywords: 3D sound, surround sound, periphony, sound spatialization, virtual environments, virtual reality, augmented reality, Higher Order Ambisonics, B-format, Vector Base Amplitude Panning, holophony, regular polyhedra, Platonic solids, geodesic spheres, minimimal energy configurations, CSL

Diese Diplomarbeit - eine Kooperation mit dem Center for Research in Electronic Arts Technology (CREATE) an der University of California at Santa Barbara (UCSB) - behandelt die Spatialisierung von Klängen in der AlloSphere, einer derzeit im Bau befindlichen Forschungseinrichtung des California Nanosystems Institute und des Media Arts and Technology Programmes an der UCSB. Diese kugelförmige Anordnung mit einem Radius von ca. 10 Metern bietet Raum für zehn bis zwanzig Leute und erlaubt eine interaktiv steuerbare Projektion von visuellen Daten und Klängen auf ihrer gesamten Innenfläche.

Die theoretischen Grundlagen räumlichen Hörens sowie eine kurze Geschichte periphoner Klangspatialisierung werden präsentiert. Es wird ein Überblick über entsprechende Techniken (Vector Base Panning, Higher Order Ambisonics, holophone Ansätze) gegeben. Mit einer gemeinsam mit J. Castellanos und G. Wakefield entwickelten Higher Order Ambisonic C++ Bibliothek für CSL (CREATE Signal Library) kann Ambisonic bis zu dritter Ordnung auf verteilten Rechnersystemen realisiert werden. Eine Strategie zum Design periphoner Lautsprecheranordnungen wird präsentiert. Dieser Hybridansatz geodäsischer Anordnungen und Anordnungen minimaler Energie ermöglicht Kompromisse zwischen homogener Lokalisation und psychoakustischen Optimierungen. Eine entsprechende Matlab Bibliothek erlaubt den Entwurf von Lautsprecheranordnungen und deren Evaluierung in Kombination mit verschiedenen Spatialisierungstechniken.

Schlagwörter: 3D Sound, Surround Sound, Periphonie, Klangspatialisierung, virtuelle Umgebungen, virtuelle Realitäten, erweiterte Realitäten, Higher Order Ambisonics, B-Format, Vector Base Amplitude Panning, Holophonie, reguläre Polyeder, platonische Körper, geodäsische Kugeln, Anordnungen minimaler Energie, CSL

Contents

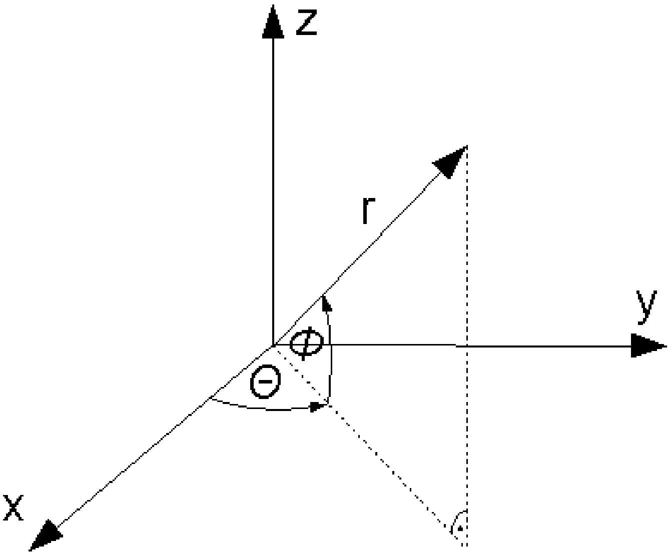


Figure 1: Coordinate system used in this thesis

Part I
Theory of Spatial Hearing

Our spatial hearing bases on two different sort of cues: those we use to evaluate the direction of a sound source and those we use to evaluate its distance. The first group can be further divided into different cues for *azimuth*[1] and *elevation*[2]. Considering these properties of human hearing, it is useful to introduce a spherical coordinate system for the description of periphonic soundfields. We will use a coordinate system as in figure 1 throughout the rest of this thesis.

In this left-oriented system, we assume the xy plane to be located at the height of the ears of an average listener. If there is any prefered direction a listener in the periphonic soundfield typically faces, we will assume this to be the direction of the x axis, which is also the zero degree direction for both, azimuth and elevation. The azimuth θ increases from 0 to π counterclockwise and decreases from 0 to $-\pi$ clockwise, whereas the elevation ϕ increases from 0 to $\frac{\pi}{2}$ towards the positive z axis, and decreases from 0 to $-\frac{\pi}{2}$ towards the negative z axis. The distance of a sound soucre is indicated by the radius r.

The x/y plane is also refered to as the *horizontal plane* or *ear plane.* The x/z plane (or *median plane*) cuts through the symetry axis of the listener's head and separates the acoustical environment into a left and right half, while the y/z plane (or *frontal plane*) is used to distinguish front and rear in our coordinate system.

While our angle naming convention follows a standard popular in the USA, the

[1] The azimuth is the horizontal angle in a spherical coordinate system.

[2] The elevation (or *zenith*) is the vertical angle in a spherical coordinate system.

letters ϕ and θ are often used exactly the other way around in European countries (ϕ for azimuth and θ for elevation). Another source of confusion is a convention widely used in mathematics, which assumes the elevation to be zero in the direction of the z axis and to increase towards the xy plane. To prevent any misinterpretations, different conventions used in the most important literature sources of this thesis are presented in the appendix (chapter B.1).

For a spherical coordinate system following these conventions, cartesian coordinates can be derived with the equations

$$\begin{aligned} x &= r\cos\theta\cos\phi \\ y &= r\sin\theta\cos\phi \\ z &= r\sin\phi \end{aligned}$$

and cartesian coordinates can be converted back into our coordinate system with the equations[3]

$$\begin{aligned} r &= \sqrt{x^2+y^2+z^2} \\ \theta &= \arctan\left(\frac{y}{x}\right) + \pi\cdot u_0\left(-x\right)\cdot\operatorname{sgn}\left(y\right) \\ \phi &= \arctan\frac{z}{\sqrt{x^2+y^2}} + \pi\cdot u_0\left(-\sqrt{x^2+y^2}\right)\cdot\operatorname{sgn}\left(z\right) = \arcsin\frac{z}{r} \end{aligned}$$

In the case of a coordinate system following different conventions regarding orientation or angels, these six equations will have to be adopted accordingly.

1 Localization in the Horizontal Plane

Spatial hearing in the horizontal plane is the result of a combination of the two major phenomena of binaural hearing: *Interaural Level Differences* (ILDs) and *Interaural Time Differences* (ITDs). The first are the major cue for localizing higher frequencies, whereas the latter are particularly relevant in the low-frequency area. Together, they allow for evaluation of the azimuth of a sound source at a very detailed resolution. Figure 2 shows the *localization blur*[4] in the horizontal plane for a fixed head. Note that head rotations greatly contribute to improved localization, which will be discussed in chapter 4. In this chapter, we will use the term *ipsilateral* as an attribute for the ear which is closer to a sound event than the *contralateral* ear. For example, for a sound wave arriving from the left (as seen from the listener's perspective), the left ear is considered ipsilateral and the right ear contralateral (and vice versa for sounds from the right).

[3] In these equations, u_0 (unity step function with $u_0\left(0\right)=0$) and *sgn* (the signum function) are used as logical switches for finding the value of θ and ϕ in the correct quadrant. For this purpose, the *atan2(y,x)* function is provided in many computer languages.

[4] The localization blur refers to the minimal difference required so that 50% of the subjects in a listening test will detect a change in the position of a sound source [Bla74, p.30]

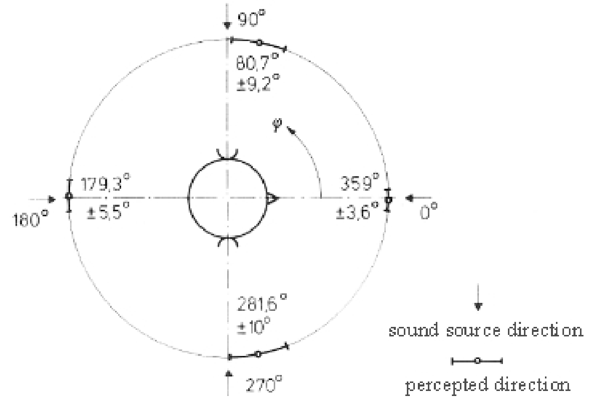


Figure 2: Localization and localization blur in the horizontal plane for 100 ms white noise impulses and fixed head position [Bla74, p.33]

1.1 Interaural Level Differences (ILDs)

Interaural Level Differences (ILD) are caused by two different effects: the attenuation of sounds on the contralateral ear by the human head, and the sound pressure amplitude decreasing with the distance of a sound source (1/r law). The latter effect can be neglected for sound sources with distances large compared to the distance between the two ears.[5] For close sources, however, the distance between the two ears becomes relevant, and both phenomena will add up to a higher ILD than would be caused by a sound source in the same direction but at a greater distance [Son03, p.9]. In this sense, the ILD serves as a distance cue as well (see chapter 3.6). On the other hand, the attenuation effect of the head is only relevant for frequencies above 300 Hz [Hub02, p.7], since diffraction occurs for wavelengths large compared to the size of the human head. Thus, Interaural Level Differences are a function of the distance of a sound source as well as of its frequency. Due to the complexity of the diffraction effects, it is difficult to mathematically model them[6].

1.2 Interaural Time Differences (ITDs)

Interaural Time Differences (ITD) occur because of the distance between the two ears, causing a sound wave to reach the ipsilateral ear earlier than the contralateral one. While the human hearing can detect such time differences as small as 30 μs [Gra00, p.68], [Ker03, p.8], the maximum ITD for a sound in the horizontal plane at an azimuth $\theta = \pm 90°$ is about 630 μs [Bla74, p.115][7]. Interaural Time Differences can be either detected in the shape of the signals themselves or in their envelopes. In the first case, they are also refered to as *Interaural Phase Differences* or *Interaural Phase Delays*. This mechanism is only relevant for frequencies below 1600 Hz, since the phase differences become ambiguous for signals with smaller wavelengths.

[5] The distance between the two ears is usually assumed with 17 cm.

[6] [PKH99] describes a binaural auditory model for evaluation of the ILDs and ITDs. This model bases on the *head-related transfer functions* (HRTFs) simulating the effects of the outer ear, and on a gammatone filterbank modelling the inner ear.

[7] [Bla74] assumes a distance of 21 cm between the ears and a speed of sound at a temperature close to 0°C. For a ear distance of 17 cm - as often assumed in the literature - and a more Californian climate (20°C), we get a value of 495 μs.

On the other hand, the detection of time differences in the envelopes (refered to as *Interaural Group Delays*) works for the entire frequency range above 100 Hz, since temporal changes in the envelope usually occur at rather low frequencies. However, this requires transients and thus does not work for stationary signals [Bla74, p.132].

Contrary to the ILD, interaural time differences can be modeled mathematically in a comparably simple way. Different models have been described to evaluate the ITD caused by a plane wavefront from a certain direction. All of them make use of a spherical head model. The simplest model [Dic97, p.118] assumes straight wave propagation to both ears, which is only possible for an acoustically transparent head. Although such heads are known to be rare, this is a valid assumption for low frequencies, where diffraction around the head occurs. Another model presented in [Gra00, p.68] assumes the wave propagation path to the contralateral ear to first follow a straight line and then curve spherically around the head, giving a good approximation of the ITD for higher frequencies. The most sophisticated model of Interaural Time Differences has been presented in [Dan00, pp.35-39], where an explicit distinction between low and high frequencies is made:

$$\begin{aligned} ITD^{LF}(\theta,\phi) &= \frac{D(f)}{c}\sin\theta\cos\phi \\ ITD^{HF}(\theta,\phi) &= \frac{R}{c}\left[\arcsin\left(\sin\theta\cos\phi\right)+\sin\theta\cos\phi\right] \end{aligned}$$

$D(f)$ is the so-called *equivalent head diameter*, c is the speed of sound, R is the radius of the head, θ is the azimuth and ϕ the elevation of the sound source. According to [Dan00, p.35], the transition frequency between high and low frequency area is somewhere around

$$f_T \approx 1.5\,\text{kHz} \;...\; 2\,\text{kHz}$$

Besides the fact that these equations also include the elevation of the sound source - accounting for the fact that ITDs can also occur for sounds outside the horizontal plane - equation (1) is identical to the one presented in [Gra00, p.68], and equation (1) is basically equivalent to the one in [Dic97, p.118]. Additionally, the *equivalent head diameter* $D(f)$ is introduced in equation (1). The reason for this concept is the observation that for a constant head diameter of $D = 2R$, the equation only yields right results for a small azimuth θ. For an azimuth close to $\theta = \pm\pi/2$, the measured ITD is actually bigger than predicted by the equation, depending on the frequency of the sound source. This irregularity can be avoided by assuming a bigger head diameter for lower frequencies. The equivalent head diameter $D(f)$ corresponds to the diameter that an acoustically transparent head needs to have in order to produce the same ITD over the entire frequency range (for a fixed direction of the sound source of course). Although Daniel presents no exact equation for $D(f)$, he notes that

$$D(f \rightarrow 0\,\text{Hz}) = 3R$$

A further desciption of this issue can be found in [DRP99, p.5].

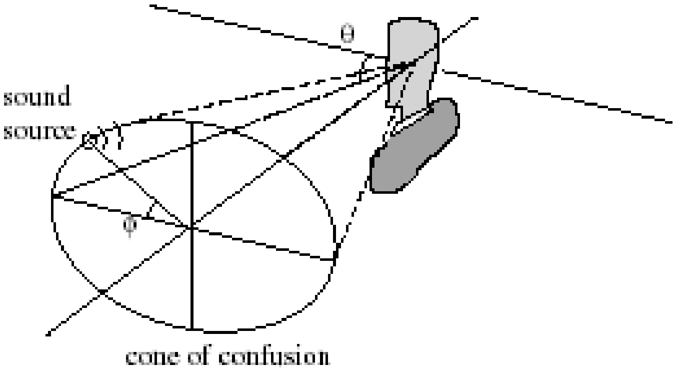


Figure 3: Cone of confusion [Pul01b, p.5]

1.3 Lateral Localization (Cone of Confusion)

Interaural differences alone are not sufficient for a model of spatial hearing in the horizontal plane, since they fail to explain how we distinguish sounds from the front and rear hemisphere creating identical interaural differences (think of two sound sources at azimuths 45° and 135°). The area of source positions causing identical ILDs and ITDs is usually refered to as the *cone of confusion*, shown in figure 3.

Two different mechanisms support our hearing in removing such ambiguities in the lateral localization of sound sources: First, the *spectral content* of soundwaves will be filtered due to interference with reflections at the pinna[8] and the torso. Due to the pinna's asymmetry, the characteristics of this filter depend on the position of the sound source, supporting our front-back discrimination. Second, *head rotations* will cause oppositely directed changes in the interaural differences for front and rear sound events. For example, the interaural differences created by a sound at an azimuth of 45° will increase if we rotate our head to the right, whereas they will decrease for a sound at an azimuth of 135°. Chapter 4 provides more information about the importance of head movements in spatial hearing.

2 Localization in the Median Plane

Sound sources in the median plane create identical ear signals. Thus, the human ear needs to rely on *monaural cues* in order to evaluate the elevation angle of a sound source. Due to the absence of interaural differences, our localization in the median plane is generally much worse than in the horizontal plane. Figure 4 shows the localization blur in the vertical plane[9].

As in the case of lateral localization (see chapter 1.3), we use the *spectral content* of a sound - which is filtered by the pinna and the torso depending on the direction of the sound source - to evaluate its position in the median plane. [Bla74] has shown that we tend to localize a sound source at a certain elevation depending on the presence of energy in certain frequency bands, which he refers to as *directional bands*, schematically shown in figure 5. Since narrowband sounds do not provide

[8] The pinna is the visible part of our ear.

[9] Note that the data on which figure 4 bases has been collected in a different study than in the case of figure 2.

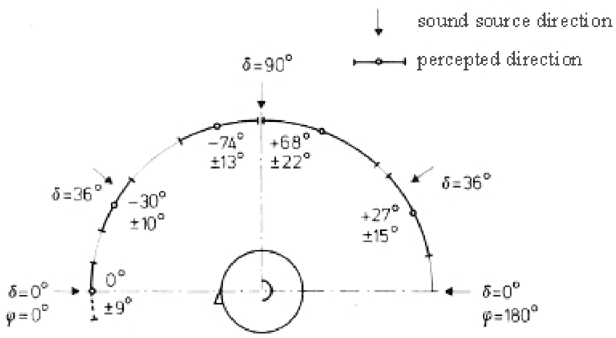


Figure 4: Localization and localization blur in the median plane for continuous speech of a familiar voice [Bla74, p.36]

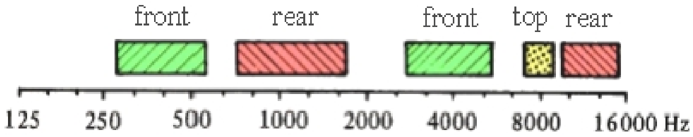


Figure 5: Directional bands according to [Bla74] (figure from [Hub02, p.8] with edits)

enough information regarding the directional bands, they can hardly be localized in the median plane. [Bla74] has observed that the perceived direction in the median plane of such a signal almost exclusively depends on its spectral content, which is demonstrated in figure 6.

Another important cue for localization in the median plane is the *familiarity* of a presented sound, which will be further discussed in chapter 5. Resembling a somewhat 'unnatural' listening environment, the localization of sounds from the *lower hemisphere* has hardly been the target of any research so far. The collection of according data would contribute imortant information regarding the design of fully periphonic sound spatialization systems such as the AlloSphere (see chapter 19.1).

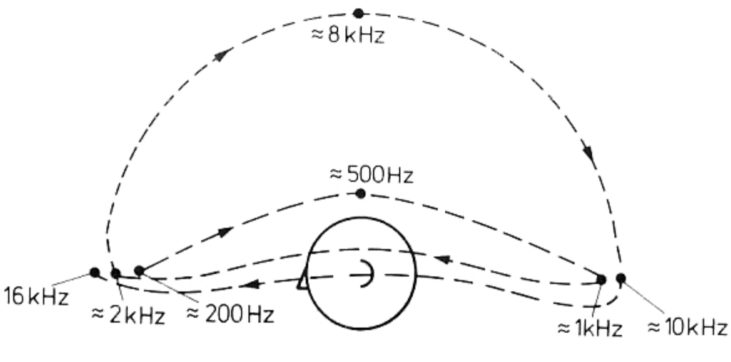


Figure 6: Perceived direction of a narrowband sound source located anywhere in the median plane, depending on its center frequency (schematic illustration) [Bla74, p.36]

3 Distance Perception

The perception of the distance of a sound source works much less reliable than the perception of its direction, since the according cues heavily depend on the spectral content and the familiarity of the presented content. A good summary of research conducted in this context is given in [Gro03, pp.41-44].

3.1 Loudness of the Direct Wavefront

Since the sound pressure amplitude of a spherical wavefront decreases with its distance (1/r law), its perceived loudness can be regarded a distance cue. However, this cue is not very reliable, since it heavily depends on the familiarity of the presented content. Also, in re-synthesized soundfields, the natural loudness of a sound source at a certain distance can easily be overridden by electroacoustical amplification. Besides, the spatial sound emission of natural sources usually follows complex, frequency dependent patterns, which cannot be sufficiently modelled by spherical wavefronts. Also, the 1/r law only holds for free field situations.[10] [Bla74, p.96] has noted that the greatest reliability of this cue can be expected for source distances in the range of 3 to 15 meters.

3.2 Atmospheric Absorption

For sound source distances over 15 meters, atmospheric sound absorption due to friction of air molecules can be regarded as an additional cue [Bla74, p.96]. The resulting change of the signal's spectral content will affect high frequencies more than low frequencies, giving a damped sound impression of distant sources. However, it has to be kept in mind that the required distances usually only occur in free field situations.

3.3 Proximity Effect

For very close sound sources ($r < 1$ m), the wavefront curvature becomes relevant compared to the wavelength, resulting in a proximity effect also familiar from recordings with velocity microphones: the low frequencies of close sources will be accentuated [BR99]. This effect is additionally supported by a property of the human auditory system which can be read from the *equal loudness contours* in figure 7. In the low-frequency region, these curves are flatter for higher sound pressure amplitudes - the low frequencies of loud signals do not have to be amplified as much in order to be perceived as loud as their high frequencies. This means that the low frequency content of a close (and thus louder) source will be perceived louder compared to its high frequencies than in the case of a distant (i.e. softer) source.

3.4 Relative Delay of Early Reflections to Direct Sound

In a closed room, it is possible to use the relative delay of the early reflections to the direct sound as a cue for the distance of a sound source. Due to the finite speed of sound, a listener will experience a delay of the direct wavefront, which increases with the distance to the sound source. The delay pattern of the statistically distributed early reflections will be less distance dependent. Thus, the relative delay of the

[10] In chapters 3.4 and 3.5, we will discuss distance cues in closed rooms.

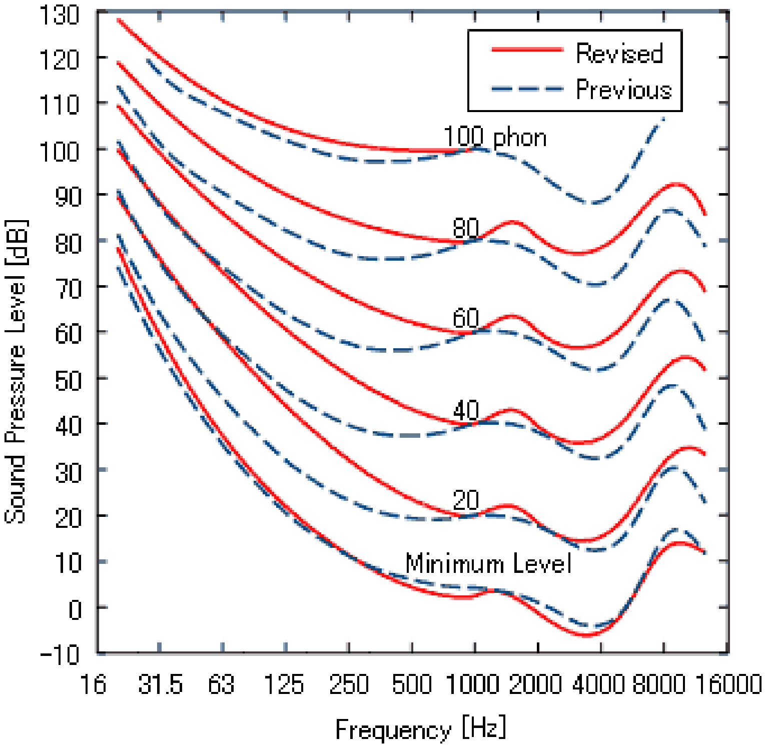


Figure 7: Equal loudness contours according to the old ISO 226 standard (dashed) and to the ISO 226:2003 revision (solid) [AIS]

early reflections to the direct sound will decrease with increasing distance and can serve as an according cue [Son03, p.9].

3.5 Intensity Ratio of Direct Sound to Reverberation

The ratio of direct sound intensity to reverberation intensity can be regarded as another important distant cue in closed rooms. While the intensity of the direct sound decreases with increasing distance of a sound source (1/r^2 law), the intensity of the reverberation will remain constant in a sufficiently diffuse room. The according ratio can be used to evaluate the distance of the source [Son03, p.9].

3.6 ILDs as a Function of Distance

It has already been mentioned in chapter 1.1 that the Interaural Level Differences (ILD) can be considered a function of distance for very close sound sources. The reason for this is the fact that for source distances of $r < 1$ m, the distance between the two ears cannot be neglected any more and the 1/r law becomes relevant. Compared to greater distances, where attenuation by the human head is exclusively responsible for the ILDs, this results in an amplification of the ILDs, which we can detect to evaluate the distance of close sound sources. [Ker03, p.13] refers to an according study of Brungart, which has shown that distance evaluation in this area is more exact for sources outside the median plane (for which ILDs occur). He also notes that Interaural *Time* Differences have turned out to be largely *in*dependent from the distance of a sound source.

4 Dynamic Localization

Beyond the localization of static sound sources, *dynamic localization* is important regarding the movement of sound sources, as it often occurs in natural and synthesized soundfields. Dynamic localization requires changes in the relative position of a sound source to the listener, i.e. movements of the source or movements of the listener.

In the case of a moving sound source, the *Doppler effect* causes changes in the pitch of the sound due to an interference of the speed of the sound source and the speed of the emitted soundwaves. It can be simulated by the means of variable delay lines in order to synthesize a natural acoustical environment [Zmö02, p.9].

Movements of the listener typically occur in form of small movements of the head, which greatly contribute to an improved localization of sound events. It is possible to distinguish two different groups of head movements:

- When trying to localize a sound source, we tend to unconsciously make small *head rotations* around the z axis to create variations of the Interaural Time Differences (ITD). These variations, which have their maximum for sounds in the horizontal plane [Dan00, pp.43-44], can remove problems in the lateral discrimination of sound sources caused by the *cone of confusion*, as described in chapter 1.3.
- *Head translations* can be considered a factor in evaluating the distance of a sound source, since the resulting variations regarding the relative position of the sound source to the listener will be bigger for close than for distant sources, leading to greater variations of the distance cues discussed in chapter 3. This is especially important for close sound sources, where the Interaural Level Differences are sensitive to translations of the head (see [Son03, p.9] and chapter 3.6).

5 Other Criteria of Spatial Hearing

Besides the signals picked up by our ears, other cues play an important role regarding the localization of sound sources:

- If any *visual cues* can be related to a sound source, the localization of this sound source will be primarily based on these[11]. This is an important factor in the design of virtual environments, which typically provide visual content as well.
- [Mal03a, p.34] has pointed out the lack in research regarding sound perception mechanisms like *chest cavity pickup* and *bone conduction*, which could play an important role in the localization of sounds in the low frequency area.
- The *familiarity* of the listener with a sound event has been described as an important factor in the evaluation of its elevation and distance [Bla74, pp.85,97]. Psychoacoustics have to be considered in this context as well: the perception of whispering will always be one of a close, intimate event, no matter how loud it is being reproduced.

[11] A popular example for this is the television, where the sound appears to be emitted by the screen rather than by the loudspeakers next to it [Hub02, p.9].

- Besides providing two important distance cues (see chapters 3.4 and 3.5), our *perception of room* gives us a good deal of information regarding the physical properties of a closed listening space. The pattern of the early reflections is important for evaluating the room's size, and the diffuse reverberation provides information about the surface structure of the walls [Zmö02, p.9]. According knowledge is essential in the design of reverberators and auralizers used for synthesizing virtual acoustical environments.
- A greatly underestimated criterion of periphonic soundfields are the *sound source properties*: size and frequency radiation pattern of a sounding object provide us with valuable information about the identity of that object as well as its position and orientation in the acoustical environment.[12] If addressed at all, this issue is usually treated globally in various techniques of synthesizing periphonic soundfields, meaning that only one size can be assigned to all sources of a field (e.g. chapter 13.3). An approach to encode size and radiation pattern for each single sound source is the Ambisonic O-format, discussed in chapter 16. The size of a sounding object also relates its direction and distance, since very close sources cannot be considered point-like any more - an assumption which we have implicitely made for the description of interaural differences in chapter 1.

6 Evaluating Periphonic Sound Localization

Most of the localization cues discussed so far can be electroacoustically synthesized in order to locally reconstruct desired properties of auditory localization, like in the case of stereophonic panning (see chapter 7). However, in multi-user virtual environments, we have to provide a much larger area of accurate sound localization than an isolated sweet spot. In the construction of such a *sweet area*, the question arises how a soundfield globally relates to its localization properties. According knowledge will also help us to accordingly focus our efforts, neglecting unpopulated areas. Several criteria of periphonic sound localization have been described in the literature for evaluating the localization properties of a given soundfield. In the case of fields which can be mathematically described, they serve as an important tool for simulating the localization properties in periphonic audio reproduction systems. Such a mathematical description is generally only possible for very simple circumstances, such as in the case of *complex monochromatic fields*[13]. The sound pressure in a complex monochromatic field of a frequency f can be described for any point in space $\vec{r}$ and time t as

$$p\left(\vec{r},t\right) = \left|p\left(\vec{r},t\right)\right| e^{j[\omega t-\phi(\vec{r})]} = P\left(\vec{r},t\right) e^{j[\omega t-\phi(\vec{r})]} \tag{1}$$

where ϕ is the phase angle, P is the pressure magnitude, and $\omega = 2\pi f$ is the radian frequency. For fields of such a form and assuming linear wavefront propagation under free-field conditions, we can derive the sound velocity[14] as [Pol00, p.1174]

$$\vec{v} = \frac{1}{k\rho c}\left(\nabla\phi + j\frac{\nabla P}{P}\right)p \tag{2}$$

[12] So far, we live in a world of electroacoustically synthesized soundfields in which all sound sources permanently face the listener directly.

[13] The term monochromatic means that the field consists of only a single frequency. For example, monochromatic soundfields are created by sine oscillators.

[14] The *sound velocity* is the speed at which the air molecules oscillate. It may not be confused with the *speed of sound*, which is much higher.

where c is the speed of sound (e.g. 331.3 m/s for air at 0 degrees Celsius), $k = \frac{2\pi f}{c}$ is the wave number, ρ is the air density (typically 1.29 kg/m^3 for air at 0 degrees Celsius and at sea level), and $\nabla\phi$ and ∇P are the gradients of phase and pressure magnitude. Note that in this equation, space and time dependence is given implicitely through p.

6.1 Velocity and Energy Vector according to Gerzon

Gerzon described how sound velocity and sound intensity both contribute to human auditory localization [Ger92]. He introduced the concepts of *velocity vector* and *energy vector*[15] as localization descriptors in periphonic soundfields. Gerzon considered the soundfield in the listening spot created by a superposition of plane wavefronts. For such a situation, he defined the velocity vector and the energy vector as [Dan00, pp.24, 61]

$$\begin{aligned}
\vec{V}_{gerz} &= \frac{\sum\limits_i a_i(\vec{r})\,\vec{u}_i(\vec{r})}{\sum\limits_i a_i(\vec{r})} = r_V \cdot \vec{u}_V \\
\vec{E}_{gerz} &= \frac{\sum\limits_i |a_i(\vec{r})|^2\,\vec{u}_i(\vec{r})}{\sum\limits_i |a_i(\vec{r})|^2} = r_E \cdot \vec{u}_E
\end{aligned}$$

where the unit vector $\vec{u}_i$ is the direction of the i-th plane wavefront relative to the listening spot $\vec{r}$, and the complex amplitude a_i describes its amplitude and phase. The unit vectors $\vec{u}_V, \vec{u}_E$ refer to the perceived direction in the soundfield. The according perception will be maximized for amplitudes r_V, r_E at their maximum value of one. Gerzon applied the velocity and the energy vector to evaluate the localization in a central listening spot $\vec{r} = 0$ of an array of loudspeakers emitting plane wavefronts by assumption. For $\vec{r} = 0$, the velocity vector describes the localization of low frequencies ($<$ 700 Hz according to Gerzon), and the energy vector is associated with high frequencies ($>$ 700 Hz). For example, if a high-frequency sound source is to be synthesized in a direction $\vec{u}_{src}$, we will try to optimize our playback system towards $\vec{u}_E = \vec{u}_{src}$ and $r_E \to 1$ in the sweet spot. Daniel has pointed out that the energy vector can also be applied to low frequency localization in the case of off-center listening positions [Dan00, p.76]. Thus, the energy vector is particularly interesting regarding the evaluation of large listening areas like in multi-user virtual environments.

6.2 Generalized Velocity Vector

While Gerzon's velocity vector based on the special case of superponed plane wavefronts, Daniel has generalized his concept to arbitrary soundfields [DRP99, p.3]. Also, he uses the velocity vector as a global soundfield descriptor rather than exclusively for a central listening spot. Poletti has used this complex velocity vector $\vec{V}$ at position $\vec{r}$ in the form [Pol00, p.1174]

$$\vec{V}(\vec{r}) = \rho c \frac{\vec{v}(\vec{r})}{p(\vec{r})} \tag{3}$$

[15] [Son03, p.20] has noted that although Gerzon uses the term *energy* vector, it is actually related to the sound *intensity*.

$\vec{v}(\vec{r})$ and $p(\vec{r})$ are sound velocity and sound pressure at position $\vec{r}$. $\vec{V}$ is a complex vector. Its real and imaginary parts describe different aspects of localization:

$$\vec{\Omega}(\vec{r}) = Re\left(\vec{V}(\vec{r})\right) = r_V \cdot \vec{u}_V$$
$$\vec{\Phi}(\vec{r}) = Im\left(\vec{V}(\vec{r})\right) = r_\Phi \cdot \vec{u}_\Phi$$

where $|\vec{u}_V| = 1$ and $|\vec{u}_\Phi| = 1$. Just as in equation (3), $\vec{\Omega}$ describes the perceived direction. $\vec{\Phi}$ is often refered to in the literature as *phasiness*. It is associated with *image broadening* and becomes large in areas where destructive interference among the wavefronts from different sound sources occurs [Pol00, p.1174].

For the special case of a complex monochromatic soundfield, we can derive $\vec{v}(\vec{r})$ and $p(\vec{r})$ using equations (1) and (2), yielding the *velocity vector for complex monochromatic fields* [Pol00, p.1174]:

$$\vec{V}_m = \frac{1}{k}\left(\nabla\phi + j\frac{\nabla P}{P}\right) \tag{4}$$

Note that the phase gradient $\nabla\phi$ is known to point in the direction of wave propagation, supporting our earlier statement according to which the real part of the velocity vector describes the perceived direction.

6.3 Active and Reactive Sound Intensity

As we have seen from Gerzon's energy vector (chapter 6.1), the development of sound intensity in a soundfield is an interesting criterion regarding its localization properties. Poletti has pointed out that it is desirable to define a time-invariant intensity quantity in order to avoid averaging operations. This is possible by restriction to complex monochromatic fields as described by equations (1) and (2). Poletti thus introduced the *complex intensity for monochromatic fields* as [Pol00, p.1175]

$$\vec{I}_C = \frac{P^2}{2k\rho c}\left(\nabla\phi - j\frac{\nabla P}{P}\right) = \frac{P^2}{2\rho c}\vec{V}_m^*$$

where $\vec{V}_m^*$ is the complex conjugate of the velocity vector for monochromatic fields from equation (4). Using the real parts of pressure and velocity, Poletti derives from this the real intensity $\vec{I}_R$ (which is *not* the real part of the complex intensity) as

$$\vec{I}_R = \frac{P^2}{\rho\omega}\left(\nabla\phi\cos^2(\omega t - \phi) - \frac{\nabla P}{2P}\sin(2(\omega t - \phi))\right)$$

We are interested in the real and imaginary parts of the real intensity, given as

$$\begin{aligned}\vec{I}_a &= Re\left(\vec{I}_R\right) = \frac{P^2}{2k\rho c}\nabla\phi\cos^2(\omega t - \phi)\\ \vec{I}_r &= Im\left(\vec{I}_R\right) = -\frac{P}{2k\rho c}\nabla P\sin(2(\omega t - \phi))\end{aligned}$$

The *active intensity* is proportional to the gradient of the phase of the soundfield and thus points in the direction of wave propagation. Its average describes the average flow of sound energy. The *reactive intensity* - being proportional to the gradient of the pressure magnitude - is normal to surfaces of constant pressure. Its time average is zero. [Pol00, p.1175]

6.4 Complex u Velocity

An alternative criterion for describing the perceived direction in periphonic soundfields is the flux of sound energy. Poletti presented the *complex u velocity* as an according soundfield indicator. As in the case of the complex intensity (see chapter 6.3), he restricts its use to complex monochromatic soundfields in order to receive a time-invariant quantity. Poletti derives the complex u velocity $\vec{u}_c$ as [Pol00, p.1176]

$$\vec{u}_c = c \cdot \frac{\vec{V}_m^*}{\frac{1}{2}\left(1 + \left\|\vec{V}_m\right\|^2\right)} \tag{5}$$

Again, the real and imaginary parts of $\vec{u}_c$ describe different aspects of auditory localization:

$$\begin{aligned} \vec{v}_a &= Re\left(\vec{u}_c\right) = c \cdot \frac{\frac{1}{k}\nabla\phi}{\frac{1}{2}\left(1 + \left\|\vec{V}_m\right\|^2\right)} \\ \vec{v}_r &= Im\left(\vec{u}_c\right) = c \cdot \frac{\frac{1}{k}\frac{\nabla P}{P}}{\frac{1}{2}\left(1 + \left\|\vec{V}_m\right\|^2\right)} \end{aligned}$$

The real part has the same direction as the active intensity (see chapter 6.3) and is thus refered to as *active velocity*. It is *not* equivalent to the speed of the wavefronts (phase velocity c_p), which is $c_p = c\frac{k}{|\nabla\phi|} = \frac{c}{\left|Re\left(\vec{V}_m\right)\right|}$. The imaginary part of the complex u velocity is also called *reactive velocity*. It is not associated with sound energy transport, since its time average is zero. [Pol00, p.1176]

Part II
History of Periphonic Sound Spatialization

Since the electroacoustic techniques have made it possible to separate the recording and playback of sounds in terms of time and space, our attention has focused on the spatial qualities of sound as being an implicit part of our auditory experience. Composers have started to understand space as a parameter available for composition, and music performance is moving away from the the 19th century idea of a stage-centered experience towards immersive audiovisual environments in virtual and augmented realities. It has often been pointed out, that there are several examples of 'spatial music' - i.e. music in which space is considered an important factor in composition and performance - prior to the electroacoustic era [Mal03b], [R^+96, p.452]. However, the invention of the loudspeaker as a device separating sounds from their physical sources surely has contributed to new aesthetics regarding the role of space in music. Two major approaches to electroacoustic music have emerged from this aesthetical thinking:

- **The loudspeaker as an instrument**
  One approach is to see the loudspeaker as a powerful new instrument, capable of reproducing more different sounds than any other one available, but still being an instrument with its own body and characteristics. The French loudspeaker orchestras of the 1970s are examples for this kind of attitude towards electroacoustic music. Among them are the *Gmebaphone* (Groupe de Musique Experimentale de Bourges, France 1973) and the *Acousmonium* (Groupe de Recherches Musicales Paris, France 1974; see figure 8). The creators of these systems have considered the diversities of different loudspeaker models and irregular speaker layouts to be essential for the development of an art of sound spatialization. For example, the Acousmonium contained eighty speakers of different sizes placed across a stage at varying heights and distances. Their placement was based on their range, their power, their quality, and their directional characteristics [EMFa]. The *BEAST* (Birmingham ElectroAcoustic Sound Theatre) is also often mentioned in this context. It has been developed by Jonty Harrison and others at the University of Birmingham, UK in the mid-1980s as a thirty-channel concert system for playing electronic music in a concert hall [EMFb].
- **The invisible loudspeaker**
  The second approach bases on a 'dematerialization' of the loudspeakers, merely abusing them as a means of creating phantom sound sources and (re)synthesized wavefields. Playback systems designed following this concept typically use more symmetrical layouts of loudspeakers with identical characteristics. Generally, this is the suitable approach regarding sound spatialization in virtual environments, which considerably depend on the successfull hiding of the technology involved in their creation and usually call for a permanent sound spatialization system that can be used for various applications of both, artistic and scientific nature.

In this chapter, we will give an overview of the evolution from the first steps of spatial audio reproduction to the present day. We will discuss different techniques which have been and partly still are used in sound spatialization. Two examples of

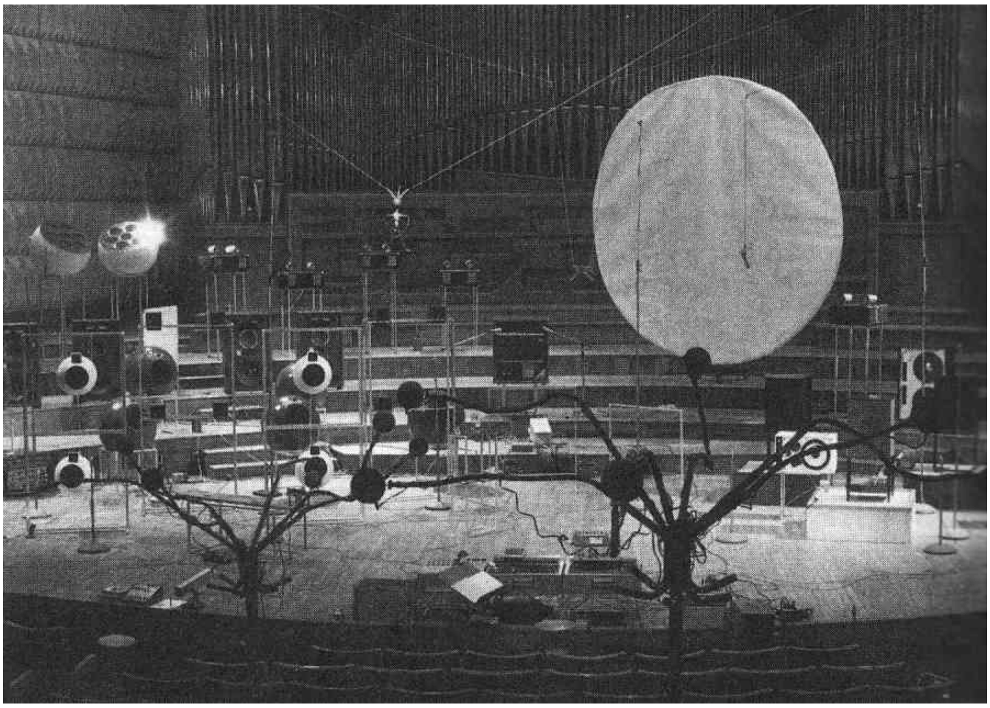


Figure 8: The Acousmonium at the Groupe de Recherches Musicales [R+96, p.454]

multi-user virtual environments providing capabilities for periphonic audio reproduction will be presented as well: Stockhausen's spherical auditorium at the world fair 1970 in Osaka as a historical example, and the CUBE at the Institute of Electronic Music and Acoustics at the University of Music and Dramatic Arts Graz, Austria as a more recent example.

7 Stereo

The first important breakthrough in the history of spatial electroacoustic music was the development of *stereophony* or - to be more precise - the invention of *phantom sound sources* created by pairs of loudspeakers. Although stereo has been developed independently in the USA and in the UK, the work of Alan Dower Blumlein can be pointed out. His patent from 1931 [Blu31] describes the stereophonic recording technique today refered to as the *Blumlein Pair.*[16] Since then, a variety of stereophonic recording techniques has been developed, which base either exclusively on amplitude (*XY* microphone technique, Blumlein pair) or phase differences (*AB* technique) between the microphones, or on a mixture of both (*ORTF* technique). Blumlein's work also included efforts of separately decoding the level and phase differences picked up by a stereo microphone to create phantom images on both the horizontal and vertical axis of a 2D loudspeaker array [Blu31][17]. It is also interesting to note, that the Blumlein pair as well as the *M/S* stereophonic microphone technique anticipated basic concepts of *Ambisonic*, a technique of periphonic audio reproduction extensively described in chapters 9 and 15.

[16] I.e. two coincident figure-of-eight microphones at a right angle with their symmetrical axis pointing towards the center direction.

[17] This design focused on movie theaters, with the loudspeaker array being hidden behind the projection screen.

The standard stereophonic loudspeaker layout consists of two speakers capable of reproducing the full audible frequency range and forming an equilateral triangle with the sweet spot in the horizontal plane. The loudspeakers are thus located at ±30 degrees from the listener's perspective, and their membranes typically face the sweet spot. Besides feeding the loudspeakers with the signals of a stereophonic recording, it is possible to synthesize phantom sound sources at any position between the two speakers. This process - which is refered to as *panning* - can be realized by feeding both loudspeakers with the same monophonic signal, either at different amplitudes, or with a small delay between them. In the fist case, the sound event will be 'dragged' towards the direction of the loudspeaker which is playing at a higher amplitude. In the second case, the sound event will appear closer to the loudspeaker which emits its wavefront earlier. These effects can be explained by the concept of interaural level and time differences, described in chapter 1. Due to its technical simplicity, panning by the means of amplitude differences has become far more popular. The *stereophonic law of sines* (Bauer, Clark, Dutton, Vanderlyn) is often used to derive the gains of the left and right loudspeaker for synthesizing a phantom source at a certain position:

$$\frac{\sin\theta}{\sin\theta_0} = \frac{g_L - g_R}{g_L + g_R}$$

θ_0 is half the angle between the two loudspeaker with $0^\circ < \theta_0 < 90^\circ$, θ is the desired angle of the phantom source with $-\theta_0 \leq \theta \leq +\theta_0$, and g_L, g_R are the gains of the left and right loudspeaker with $g_L, g_R \in [0, 1]$. However, this equation is only valid if the listener's head is exactly facing the symmetrical axis between the two loudspeakers. The *tangent law* (Bennett et al., Makita, Leakey, Bernfeld) is also correct for slight rotations of the listener's head [Bos01]:

$$\frac{\tan\theta}{\tan\theta_0} = \frac{g_L - g_R}{g_L + g_R} \tag{6}$$

Since both laws only describe the relation between the gain factors of the left and right speaker, a second equation is required to actually resolve the gain factors [Pul01b, p.12]:

$$\sqrt[P]{g_L^P + g_R^P} = 1 \tag{7}$$

Typically, P is chosen with $P = 1$ to keep the amplitude of a virtual sound source at a constant level for all possible positions, or with $P = 2$ to keep the sound energy constant.[18] [Pul01b, p.13] has pointed out that the first choice is often made in anechoic listening rooms, while the second approach is prefered in rooms with more extensive reflections.

Different panning laws have been proposed by John Chowning and others [Pul01a, p.13], [R$^+$96, pp.459-460]. The obvious drawbacks of the stereo system regarding periphonic sound spatialization are given by the small area of accurate audio reproduction (sweet spot) and the limitation to frontal sound source reproduction.

[18] Note that this does *not* refer to the techniques of *amplitude* and *intensity* panning, which are described in chapter 13!

8 Quadrophony, Octophony

In the 1970s, experiments have been made to extend the principles of stereo to quadrophonic layouts with four loudspeakers arranged in a rectangle. These systems suffered from the tradeoff between poor frontal versus poor lateral images: if the speakers are arranged in a square, the angle between the front speakers is 90 degrees, as opposed to the 60 degrees of a standard stereo layout, resulting in a 'hole in the middle' for frontal images. Trying to avoid this problem by approaching a more rectangular layout will at the same time decrease the quality of the lateral images. Also, the ear's lateral resolution is much worse than its frontal resolution, which furhter increased the problem. There has been a variety of quadrophonic audio formats (JVC CD4, CBS SQ, Sansui SQ, Nippon Columbia UD4, etc.), none of which was able to define a broad industry standard. Mostly, these systems were designed to be compatible with the stereo format, so they were used to reproduce quadrophonic recordings as well as to enhance the spatial quality of stereo recordings. To stay compatible with stereo recording media, the four channels of quadrophonic recordings were often encoded onto two channels using frequency modulation[19] [Zmö02, p.13]. In order to achieve the full bandwidth for all four signals, frequencies as high as 50 kHz were stored on vinyl, as in the case of the CD-4 format. Unfortunately, the high-frequency content was subject to wear, decreasing the spatial effect of often played quadrophonic records.

To escape the horizontal plane, quadrophonic systems sometimes had their speakers arranged in four corners of a rectangular parallelepiped (e.g. left bottom, right top for the front speakers and left top, right bottom for the rear speakers) [R$^+$96, p.456]. Obviously, such systems do not allow for independent control of azimuth and elevation of a sound source. To overcome the drawbacks of quadrophony, *octophonic* layouts with eight loudspeakers have been studied to increase the quality of phantom images in the horizontal plane (circular layouts) or to allow for true with-height reproduction (cubic layouts).

9 First-Order Ambisonic (B-Format)

The *Ambisonic* system, developed by Michael Gerzon from the University of Oxford, UK, was commercially even less successfull than quadrophony, but the theory it bases on had a lasting impact on periphonic sound reproduction. Gerzon's approach was the first which explicitely focused on the reproduction of entire sound*fields* rather than the creation of isolated phantom sound sources - a step which makes the introduction of full periphony a natural thing to do.[20] In its most basic version, the Ambisonic system allows for full periphony by the means of four loudspeakers arranged in a tetrahedron. If the reproduction quality of such a system is of course limited, its capabilities are still remarkable compared to the struggles of quadrophonic systems regarding horizontal-only reproduction with the same amount of loudspeakers. Even the current 5.1 standard - which uses even more loudspeakers - does neither support elevated sound sources nor arbitrary source positioning in the horizontal plane.

[19] For a description of the according encoding and decoding matrix operations used by different manufacturers, see [Mit]

[20] Gerzon's article *With-Heigth Sound Reproduction* [Ger73] literally took this step out of the horizontal plane.

The Ambisonic approach allows for recordings of natural soundfields by the means of one omnidirectional microphone (feeding the so-called *W channel*) and three figure-of-eight microphones in the directions of the x-, y-, and z-axis (feeding the *X*, *Y*, and *Z channel*).[21] This can also be interpreted as a threedimensional extension of the M/S microphone technique: the W channel carries the omnidirectional information, whereas the X, Y, and Z channels carry the directional information of the recorded soundfield. Together, the first-order Ambisonic channels W, X, Y, and Z are usually refered to as the *B Format.* Since the polar patterns of omnidirectional and figure-of-eight microphones can be described in a mathematical rather simple form, the Ambisonic approach can also be used to synthesize soundfields created by an arbitrary number of sound sources in different directions. The Ambisonic channels can then be derived as[22]

$$W = \sum_{i=1}^{k} s_i \frac{1}{\sqrt{2}} \tag{8}$$

$$X = \sum_{i=1}^{k} s_i \cos\theta_i \cos\phi_i \tag{9}$$

$$Y = \sum_{i=1}^{k} s_i \sin\theta_i \cos\phi_i \tag{10}$$

$$Z = \sum_{i=1}^{k} s_i \sin\phi_i \tag{11}$$

Here, s_i is the unencoded mono signal of the i-th sound source, and the azimuth θ_i and the elevation ϕ_i specify its direction. The number of sound sources is given by k. As can be seen from the equations, the sound sources are encoded by multiplication with simple sine and cosine terms and are then summed up onto the Ambisonic channels. Note that this summation can require a attenuation of the encoded channels in order to avoid clipping. The reason for the $\frac{1}{\sqrt{2}}$ weighting of the W channel is explained by David Malham:

> ”The 0.707 multiplier on W is present as a result of engineering considerations related to getting a more even distribution of signal levels within the four channels. This is particularly relevant when taking live sound from a Soundfield microphone or with synthesised soundfields containing many sources.” [Mal03a, p.46]

Since the Z channel only depends on the elevation but not on the azimuth, it can be neglected in the case of horizontal-only loudspeaker layouts, reducing the number of required Ambisonic channels to three. Gerzon has pointed out that this is the prefered approach for feeding quadrophonic speaker layouts in the horizontal plane [Ger92, Ger74].

[21] Ideally, the microphones have to be positioned in the same spot. A solution to this problem has been introduced in the *Soundfield microphone*, which arranges four cardioid microphone capsules in a tetrahedron and superimposes their polar patterns as required to create the described characteristics. The distances between the microphones are compensated by filters [Dan00, p.103].

[22] Note that the original B-format proposed by Gerzon featured an additional multiplication by $\sqrt{2}$ on all channels [Dan00, p.101].

The four Ambisonic channels have to be decoded in order to obtain the required driving signals for the loudspeakers. In the decoding process, each loudspeaker receives a weighted sum of all channels, with the weighting factors depending on its own position. A basic decoding equation for the j-th loudspeaker can be expressed as

$$p_j = \frac{1}{L}\left(W\frac{1}{\sqrt{2}} + X\cos\theta_j\cos\phi_j + Y\sin\theta_j\cos\phi_j + Z\sin\phi_j\right)$$

where p_j is the driving signal for the j-th loudspeaker and (θ_j, ϕ_j) describe its position. The number of loudspeakers L must be at least four (the number of Ambisonic channels) to decode all available directional information. However, it is possible to introduce more speakers than that. Ideally, they form a *regular* layout, e.g. a tetrahedron for four available speakers, or a cube in the case of eight speakers. In the case of horizontal-only layouts, the loudspeakers should be distributed at equal angles along a circle around the listener.[23]

The separation of the encoding (or recording) and decoding processes is the major advantage of the Ambisonic approach. First, because the exact properties of the loudspeaker layout do not have to be known at encoding time, which makes for flexible *portability* of the encoded material. Second, the Ambisonic encoding format is a powerful representation of three dimensional soundfields, allowing for straightforward manipulation of their spatial characteristics. Soundfield operations like *rotation*, *mirroring* and *dominance* (a zoom-like operation) can be implemented in a very efficient way and will be further described in chapter 15.3. In the 1990s, the Ambisonic approach has been extended to higher orders, introducing better localization quality (due to better soundfield approximation) as well as an extension of the sweet spot to a sweet area, at the cost of a higher number of encoded channels and required loudspeakers. This illustrates the *scalability* of Ambisonic representations, with its upper limit only determined by the available hardware (number of loudspeakers, CPU power). Higher Order Ambisonic (HOA) will be extensively described in chapter 15.

10 Dolby Standards, 5.1

Several techniques have been developed as extensions to the stereophonic layout, especially in the context of sound system design in movie theaters. The *Dolby* company has defined many according standards, often in combination with specific hardware. *Dolby Surround* and *Dolby Pro Logic* have been precedessors to the currently popular *Dolby Digital* standard. Dolby Digital uses a loudspeaker layout usually refered to as the *5.1* setup, which has become an industry standard itself, being used for reproduction of other formats as well. It consists of a classical 60-degree stereo pair of loudspeakers, a front center loudspeaker, a left and right surround speaker at ±110 degrees plus an additional subwoofer. Extended formats are available for 7.1 or 10.2 layouts, mostly found in movie theaters. [Son03] has pointed out that all these techniques aim at functionally recreating an acoustical environment rather than at a global soundfield recreation: they restrict themselves to the reproduction of an extended frontal stereo image, adding surround channels and subwoofers for ambience and low-frequency sound effects. This is adequate for

[23] Thus, a standard 5.1 layout is *not* the right choice for an Ambisonic speaker layout. Nevertheless, this layout has often been chosen for evaluating the qualities of the Ambisonic representation in general.

the dialogue-based sound design of many movies and for stage-centered situations like live recordings of classical and popular music. However, the 5.1 format does not allow for circular movements in the horizontal plane, and none of the mentioned techniques supports sound sources at arbitrary elevation angles. Thus, they are not suitable for periphonic sound spatialization in multi-user virtual environments.

11 Current Research

With the increasing computational power which has become available in the 1990s for digital signal processing applications, it has eventually become possible to realize large scale systems capable of rendering true periphonic sound spatialization in realtime. According techniques have been extended and improved. Among the most important are *Vector Base Panning*, a technique developed by Ville Pulkki at the Helsinki University of Technology, and *Higher Order Ambisonic*, an extension of the Ambisonic approach (see chapter 9) developed by several researchers around the world. These techniques will be described in chapter III. It will be shown that Vector Base Panning is a three dimensional generalization of the stereo panning law, and that Higher Order Ambisonic is actually a special case of *holophony*, the acoustical equivalent to holography, which bases on the idea of global soundfield reconstruction by the means of large loudspeaker arrays. Holophony also forms the theoretical basis for *Wave Field Synthesis*, a technique for soundfield reconstruction in the horizontal plane. We will also discuss the possibility of extending this technique to periphonic applications.

12 Examples of Periphonic Sound Spatialization Systems

To complete this chapter, we will present two examples of periphonic sound spatialization systems in multi-user virtual environments. The spherical auditorium designed by Karlheinz Stockhausen for the World Fair 1970 in Osaka, Japan will be presented due to its obvious relation to the concept of the *AlloSphere*, presented in chapter 19.1. As a more recent example, we will present the *CUBE* at the Institute of Electronic Music and Acoustics (University of Music and Dramatic Arts Graz, Austria).

12.1 The Spherical Auditorium at the World Fair 1970 in Osaka

The spherical auditorium designed by Karlheinz Stockhausen for the 1970 world fair at Osaka, Japan serves as an early example of a fully periphonic sound spatialization system. Stockhausen's original design featured eight rings of loudspeaker with eight loudspeakers each, including two manually operated 'rotation mills' for continuous sound rotation by the means of sliding contacts. An acoustically and visually transparent listening platform was planned to be located at the height of the sphere's equator. In the course of the project, the design has been scaled down to seven loudspeaker rings with seven speakers each, plus an additional subwoofer. Also, only one of the two rotation mills has been implemented. The listening platform was located at a height of three meters below the equator, which Stockhausen regarded suboptimal [Sto71, p.153]. Three loudspeaker rings and the subwoofer

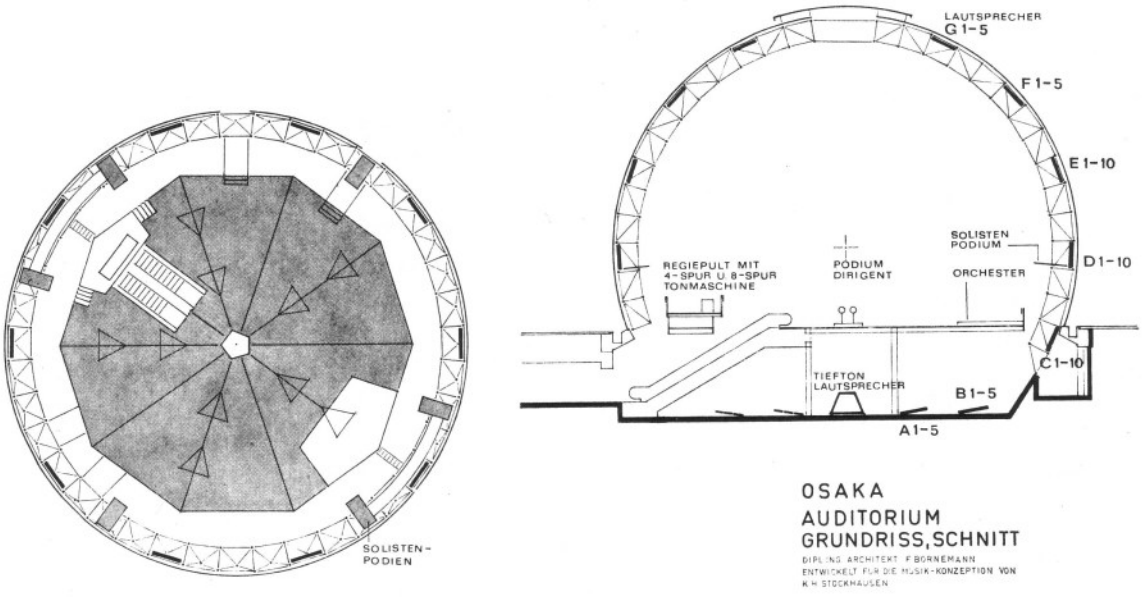


Figure 9: Ground plan and sectional view of the Osaka auditorium [Sto71, p.166]

were located below the listener platform, with the subwoofer and the lowest two rings arranged concentrically on the floor (see figure 9). The other four loudspeaker rings were mounted in the upper hemisphere. The loudspeakers were provided by the Altec company, the amplifiers by Klein & Hummel. An eight-track recorder of the Telefunken company (model M10A) was used for audio signal playback, and the sound spatialization was realized on a custom built mixing console. The musical concept originally consisted of a new composition by Stockhausen with the title 'Hinab - Hinauf', including a light concept by Otto Piene. This proposal was later rejected by the world fair committee. The new concept, which did not any more include the visualization part, featured music by several German classical composers as well as a total of four hours of Stockhausen's music per day. From March 14th to September 15th 1970, about a million people visited the auditorium. [Sto71, pp. 153-187]

12.2 The IEM CUBE

A more recent example of a multi-user virtual environment is the CUBE at the Institute of Electronic Music and Acoustics (IEM) of the University of Music and Dramatic Arts at Graz, Austria. The CUBE (**C**omputer-**u**nterstützte **B**eschallungs**e**inheit, i.e. 'computer assisted audio reproduction unit') serves as an electroacoustical laboratory and medium-sized concert hall. Its core audio reproduction system consists of a hemispherical arrangement of 24 loudspeakers (Tannoy System 1200 loudspeakers with Thomann TA-500 amps) and two subwoofers. An eight-by-six matrix of 48 loudspeakers (JBL Control 1) - refered to as 'the Sky' - is mounted on the ceiling and can be adressed over 24 separate channels for extended spatial effects. An eight channel array of Klipsch loudspeakers serves as an alternative PA system, especially for the reproduction of music from the historical repertoire of electroacoustic music. Visual content can be presented on a projection screen. A separate machine room hosts the amplifiers, converters, patchbays, servers and other machinery. [Zmö]

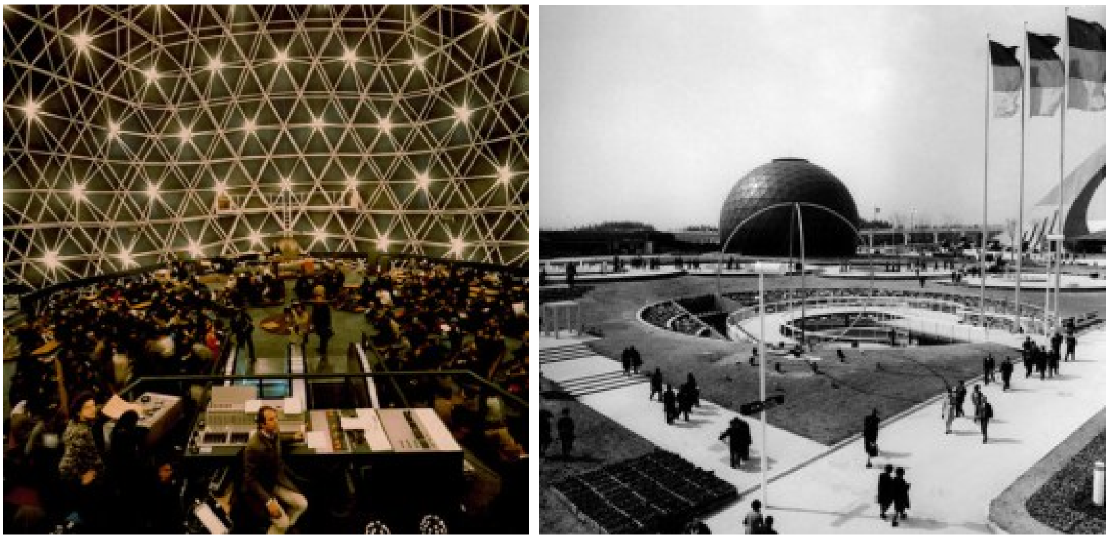


Figure 10: The Osaka auditorium from inside and outside [Der]

The 24 channel core system is arranged in three loudspeaker rings with twelve, eight, and four speakers from the bottom to the top. The loudspeakers are positioned in a way that allows for reproduction of the standard multichannel audio formats (stereo, quadrophony, 5.1). A derivation of the loudspeaker positions can be found in the appendix of [Son03]. The standard engine for periphonic sound spatialization is a fourth order Ambisonic system implemented in *Pure Data* [Puc96]. A 3D sound mixer, which has been designed and implemented by Thomas Musil, Johannes M. Zmölnig and Winfried Ritsch, includes a graphical frontend, 3D reverberation for far distance coding and a simulation of the Doppler effect by the means of variable delay lines. Alternatively, audio signals can be spatialized using a discrete input-output matrix with individually adjustable gains [RMZH05]. Informal latency measurements on this system have yielded results as low as 5 ms at a blocksize of 64 samples[24] using an OSS audio driver written by Winfried Ritsch. The spatialization engine runs on an off-the-shelf PC under the Debian GNU/Linux operating system. The graphical user interface and the digital signal processing engine have been separated as two independent tasks communicating over the TCP/IP protocol. This allows the operating system to grant realtime priority to the signal processing engine exclusively. Two additional machines serve as alternative playback devices for the source signals, one running Debian GNU/Linux and the other one using the Windows operating system.

The CUBE, which was opened in 2000, is used in research and development as well as for artistic productions. It hosts the *Open CUBE* concert series, featuring regular concerts with artists from the international electroacoustic music scene. Its connection to the main production studio of the institute and its acoustical properties allow to use it as a recording room. It has been used in pre- and postproduction of contemporary music theater by composers Peter Ablinger, Bernhard Lang, Olga Neuwirth and others. The scientific applications of the CUBE include acoustical measurements, listening tests, sonification of scientific data and prototyping of periphonic sound spatialization engines.

[24] At a sampling rate of 44.1 kHz, 64 samples equal about 1.45 ms. Considering a total of three buffers, one introduced by the Pure Data application and two by the input and output handling of the driver, the total latency adds up to $3 \cdot 1.45ms = 4.35ms$, roughly corresponding to the 5 ms measured.

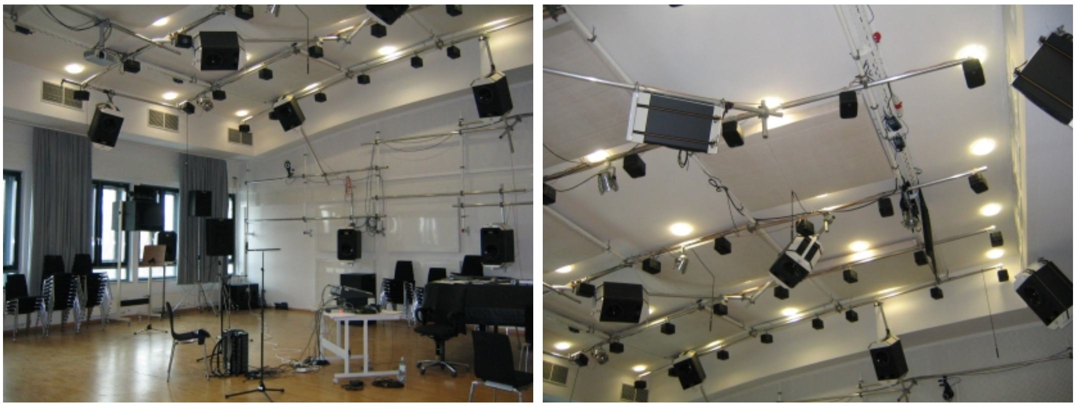


Figure 11: The IEM CUBE

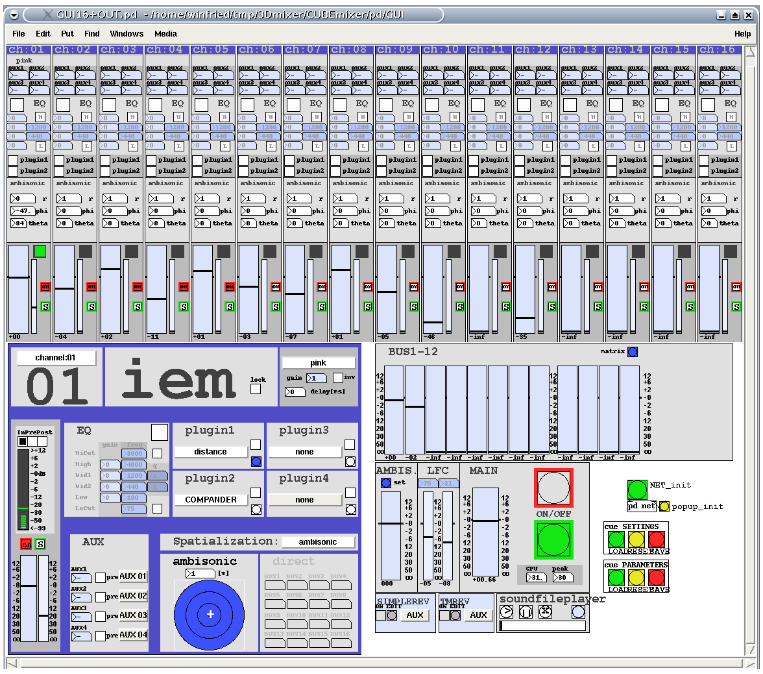


Figure 12: The graphical frontend of the 3D Sound Mixer in the IEM CUBE [RMZH05]

Part III
Theory of Periphonic Soundfield Reproduction

In the 1990s, several techniques for the reproduction of periphonic soundfields have been described or extended. Among them are *Vector Base Panning*, *holophony*, and *Higher Order Ambisonic.* We will give an introduction into each of them in the course of chapter III, including options of encoding not only the direction but also the distance of sound sources. The evaluation of the advantages and disadvantages of different spatialization techniques calls for subjective means of measuring the soundfield reproduction accuracy. Suitable approaches will be discussed in chapter 17. Chapter 16 is dedicated to the encoding of *sound object properties*, like radiation characteristics, surface shape and size, in order to allow for a richer auditory experience in virtual environments. And in chapter 18, different strategies for the design of periphonic loudspeaker layouts are presented, including a new, hybrid approach suitable for various spatialization techniques.

13 Vector Base Panning

Vector Base Panning (VBP) has been introduced by Ville Pulkki [Pul97] as an extension of stereophonic techniques. It is a sound-*source* oriented approach in the sense that it aims at the creation of phantom sound sources rather than at global soundfield reconstruction as in the case of holophony (chapter 14) or Higher Order Ambisonics (chapter 15). In Vector Base Panning, a 2D loudspeaker array is treated as an arrangement of subsequent stereo pairs, which allows for extending stereophonic panning techniques to the entire horizontal plane. For example, a standard 5.1 setup can be interpreted as a conglomeration of five stereo pairs of loudspeakers, each speaker forming a pair with both, his left and right neighbor. If a sound source is to be placed at an arbitrary position in the layout, we first have to find the right stereo pair, which is defined by the two loudspeakers enclosing the virtual source. Standard stereophonic panning laws (see chapter 7) are then applied to those two speakers. Extending this approach to *triples* rather than pairs of loudspeakers, 3D Vector Base Panning can be formulated as a means of periphonic sound spatialization. In this chapter, we will discuss the 3D case, from which 2D VBAP can be easily derived as well.

13.1 Panning within a Single Loudspeaker Base

In Vector Base Panning, the positions of the loudspeakers in each loudspeaker triple (or pair in the 2D case) are formulated as a vector base. The position $\vec{s}$ of a virtual sound source can then be defined as a linear combination of the loudspeaker vectors of the according triple:

$$\vec{s} = \sum_{i=1}^{D} g_i \cdot \vec{l_i}$$

$D \in [2, 3]$ refers to the two cases of 2D and 3D Vector Base Panning, g_i is the gain of the i-th loudspeaker in the triple and $\vec{l_i}$ is its position in cartesian coordinates.

We can rewrite this equation in a matrix representation:

$$\boxed{\vec{g} = \vec{s}^T \cdot \mathbf{L}^{-1}} \tag{12}$$

The (intermediate) gain factors of the loudspeakers are given by $\vec{g}$, and $\mathbf{L}$ is a matrix of row vectors $\vec{l_i}$ representing the loudspeaker positions. For the 2D case, equation (12) is equivalent to the stereophonic tangent panning law (eq. 6) [Pul97, pp.465-466]. For the 3D case, equation (12) can also be written as

$$[g_1\ g_2\ g_3] = [s_x\ s_y\ s_z] \begin{bmatrix} l_{1x} & l_{1y} & l_{1z} \\ l_{2x} & l_{2y} & l_{2z} \\ l_{3x} & l_{3y} & l_{3z} \end{bmatrix}^{-1}$$

The gain factors have to be scaled down in order to satisfy a condition like in equation (7), which can be generalized to

$$\sqrt[P]{\sum_{i=1}^{D} g_i^P} = 1$$

[Pul97] has suggested to apply constant power panning ($P = 2$). The final gain factors for the loudspeakers in a triple are thus given as

$$\boxed{\vec{g}_{VBAP} = \frac{\vec{g}}{\sqrt{\sum_{i=1}^{D} g_i^2}}} \tag{13}$$

It has been pointed out that this calculation of the loudspeaker gains leads to a reconstruction of the velocity vector (see chapters 6.1 and 6.2), which is known to describe the percepted direction of low frequency soundwaves (< 700 Hz). Pernaux et al. have thus suggested *Vector Base Intensity Panning* (VBIP) as an alternative to optimize the localization of frequencies > 700 Hz by resembling the energy vector [PBJ98]. The intermediate gain factors are calculated in the same way as for Vector Base Amplitude Panning (VBAP) (eq. (12)). However, the normalization of the gain factors is replaced by:

$$\boxed{\vec{g}_{VBIP} = \sqrt{\frac{\vec{g}}{\sum_{i=1}^{D} g_i}}} \tag{14}$$

It is possible to combine both approaches, using separate panning functions in two filtered frequency subbands, in order to optimize sound source reproduction in the entire frequency range.

13.2 Periphonic Panning

In the general case of fully periphonic sound spatialization, the loudspeaker layout needs to be divided once into non-overlapping triangles in a way that will optimize the homogeneity of localization, i.e. equal triangle sizes and avoiding very narrow triangles. An according algorithm has been presented in [PL98]. For panning a sound source to a certain direction, one needs to first find the triangle enclosing that direction. This can be done by first calculating the gain factors of *all* loudspeaker triples: the right loudspeaker base is the only one which yields positive gain factors for all three loudspeakers [Pul97, p.459], which are then applied to the triple. All other loudspeakers are silent. Movements of sound sources are realized by interpolating the gain factors of subsequent source positions sampled at the available control rate.

13.3 Source Spread (MDAP)

We have seen that for a single virtual sound source encoded by the means of 3D Vector Base Panning, a maximum of three loudspeakers is playing at a time. However, if the position of the source is identical with the position of a loudspeaker, only that loudspeaker will contribute to the reproduction of the source, like it is the case for hard-left or hard-right panned sources in stereo. For source positions on the connecting edge of two loudspeakers, only those two speakers will be active. For source positions at the center of a triangle, all three loudspeakers of that triple will play at equal volumes. It has turned out that these properties of Vector Base Panning make the spread of the virtual source appear incoherent: virtual sources close to a loudspeaker tend to 'collaps' into a point-like source at the speaker's position, whereas virtual sources close to the center of a loudspeaker triple appear bigger. Pulkki has presented *Multiple Direction Amplitude Panning* (MDAP) as an approach to avoid this problem [Pul99]. In MDAP, a sound source is panned to multiple directions close to the desired direction of the source, avoiding situations where less than three loudspeakers are active at a time. Adjustable parameters of this technique are the number of panning directions and the *spread angle*, which Pulkki defines as the maximum angle between two panning directions. Pulkki assumes that the 'average' of those multiple directions will be perceived as the direction of the source. This assumption implies that the virtual source quality will not be degraded if all panning directions are located in the same loudspeaker set, and that only source positions identical with the position of a loudspeaker or located on the connecting edge of two loudspeakers are affected, which makes for a more homogeneous localization quality of the virtual source. MDAP can be regarded an interesting technique for additional encoding of the size of a sound object, contributing to a more diverse sound spatialization experience (see also the paragraph on sound source properties in chapter 5).

13.4 Critique

Vector Base Panning is a very *simple and efficient* way of achieving periphonic sound spatialization. It is very *flexible regarding the loudspeaker layout*: a homogeneous loudspeaker distribution will provide more homogeneous localization quality, but irregularities in the layout will only locally affect the reproduction, due to the angular limitation of active loudspeakers. Daniel has pointed out that this limitation also gives a relative *good stability of localization at off-center listening positions* [Dan00, p.94], since a sound source can generally not be 'dragged' towards the loudspeakers that are closer to the listener like in holophonic techniques (Wave Field Synthesis, Ambisonics, etc.), where all loudspeakers are active at each moment.

However, Vector Base Panning suffers from certain drawbacks as well: Since VBP bases on resynthesizing interaural differences (see chapter 1), there are *problems regarding panning sources in the median plane*, where no interaural differences occur. Pulkki has conducted listening tests regarding the localization of virtual sources created in the median plane by the means of Vector Base Panning and has found that the source position is largely perceived individually [Pul01a]. It has been pointed out in chapter 2 that we rely on the spectral content of an audio signal to evaluate its elevation. Thus, it would be necessary to interpolate the spectra rather than the gain factors for panning in the median plane.

The *distance* of a sound source can not be reproduced by VBP. Generally, the virtual sources appear to be moving on the surface of the loudspeaker layout. Outside sources (with a radius bigger than the radius of the loudspeaker array) can be simulated by adding artificial reverberation. However, it is not possible to reproduce sound sources *within* the loudspeaker layout, like in the case of holophony (chapter 14) or Higher Order Ambisonics (chapter 15).

To store VBP synthesized soundfields, the source signals and their spatial positions either have to be stored separately, or the loudspeaker driving signals themselves have to be stored. This makes for less portability compared to Ambisonic representations, which allow for encoding the spatial information of a soundfield into standard audio signals, without any knowledge about the loudspeaker layout used for later reproduction. The same observation yields that recordings of natural soundfields cannot be achieved with Vector Base Panning either.

14 Holophony

Holophony has been described as the acoustical equivalent to holography [Jes73]. It aims at a global reconstruction of soundfields by the means of large loudspeaker arrays rather than at a local reproduction of phantom sources. Source-based algorithms like Vector Base Panning regard a single loudspeaker at the position of a virtual source as the optimum means to reproduce that source. Sound source positions not coincident with the position of a loudspeaker are reproduced as phantom sources by panning between several loudspeakers. However, this is generally regarded as degrading the quality of the reproduced source, and thus the number of active loudspeakers is kept at a minimum. Holophonic techniques, on the other hand, base on the reconstruction of an entire sound*field* by the means of a loudspeaker array. Here, the loudspeakers are merely expolited as a means to reproduce the field according to a mathematical description given by the *Kirchhoff-Hemholtz integral* (chapter 14.1). This has the effect, that generally all loudspeakers are contributing to the reproduction of a single virtual source. This technique allows for a significant extension of the area of accurate soundfield reproduction, which is particularly relevant in multi-user virtual environments. The mathematical background of holophony will be discussed in this chapter. In the last years, the according theory has been greatly extended and simplified towards horizontal-only holophonic soundfield reconstruction - a technique which has become known as *Wave Field Synthesis*. However, we are more interested in the holophonic theory with regards to three-dimensional implementations.

14.1 Theoretical Background

The *Huygens Principle*, formulated by Christian Huygens as early as 1690, states that each point of a wavefront may be regarded as the origin of a secondary spherical wavefront (see figure 13). Fresnel has completed this principle in 1818 by stating that the original (or: primary) wavefront can either be regarded as being emitted by primary sources, or also as being the superposition of the secondary wavefronts [SH00b, p.6]. It is therefore possible to replace the primary source(s) - e.g. one or more musical instruments - by a continous distribution of secondary sources - e.g. an infinitely dense loudspeaker array. Note that the situation in the second picture of figure 13 can also be inversed, so that the loudspeakers face *inwards* and reproduce the field due to sources *outside* of the array [DNM03, p.3].

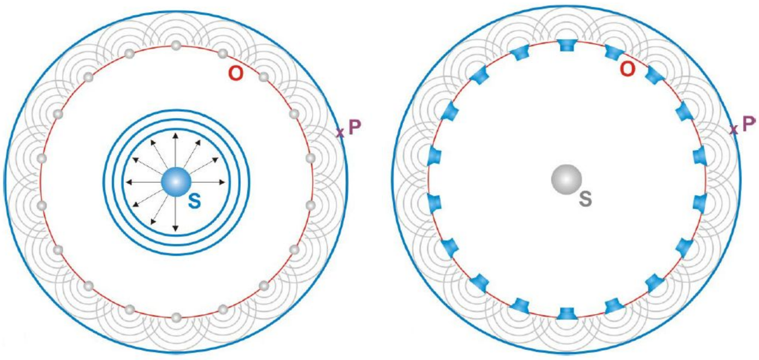


Figure 13: The Huygens principle [Hub02, p.12]

The mathematical description of the Huygens-Fresnel Principle is given by the *Kirchhoff-Helmholtz Integral* (KHI, eq. (15)). The KHI is derived from the wave equation and from Green's integral theorem. It states that for a volume V free of field sources, knowledge about the pressure and its gradient on the surface S enclosing V is sufficient to derive the field pressure within the entire volume V [SH01]:

$$\boxed{p\left(\vec{r}_R,\omega\right)=\frac{1}{4\pi}\oint\limits_S\left[\,p\left(\vec{r}_S\right)\cdot\nabla_S\,G\left(\vec{r}_R|\vec{r}_S\right)-G\left(\vec{r}_R|\vec{r}_S\right)\cdot\nabla_S\,p\left(\vec{r}_S\right)\,\right]\vec{n}\cdot dS} \tag{15}$$

for $r_R \in V$ and $r_S \in S$. G denotes Green's function, given as

$$G\left(\vec{r}_R|\vec{r}_S\right)=\frac{e^{-jk|\vec{r}_R-\vec{r}_S|}}{|\vec{r}_R-\vec{r}_S|}$$

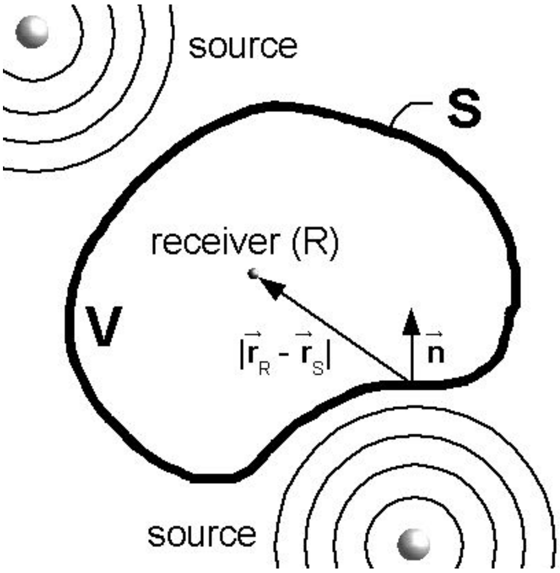


Figure 14: Geometry of the Kirchhoff-Helmholtz integral

$\vec{r}_R$ denotes a point in the source-free volume V enclosed by the surface S on which the points $\vec{r}_S$ are distributed. $\vec{n}$ is the vector normal to the surface S, and ∇_S is the Nabla operator with respect to the surface S. According to the KHI, the sound pressure field within a given volume V caused by sound sources *outside* of V can be reproduced by two continuous distributions of secondary dipole[25] and monopole sources on the enclosing surface S. The dipole source are driven by the pressure signals $p\left(\vec{r}_S, \omega\right)$, and the monopole sources are driven by the pressure gradient signals $\frac{\delta}{\delta \vec{n}} p\left(\vec{r}_S, \omega\right)$. Note that while the soundfield is reconstructed inside V, it becomes zero outside of V. The validity of the Kirchhoff-Helmholtz integral is limited to free-field situations. It can be understood as a generalization of the Huygens-Fresnel principle, since it assumes complex driving signals, while the Huygens-Fresnel principle only considers the magnitude, but not the phase of the signals. This has the effect that the distribution of secondary sources in the Kirchhoff-Helmholtz integral is arbitrary, but the Huygens-Fresnel principle requires the secondary sources to be distributed along the wavefront, which is a surface of equal phase [DNM03, p.2]. Holophonic soundfield recording is possible by using microphone arrays ideally identical with the loudspeaker positions in later soundfield reproduction. It has also been shown that soundfield interpolation can be applied to compensate for varying positions of microphones and loudspeakers. Holophonic recording techniques are also relevant regarding Higher Order Ambisonic Systems, due to the mathematical equivalency of the two approaches, which will be discussed in chapter 15.

14.2 Discretization of the Loudspeaker Array

The realization of the secondary source distribution by the means of a loudspeaker array requires a transition from a continuous to a discrete distribution due to the finite loudspeaker size and limitations regarding hardware and CPU. This discretization can be interpreted as spatially sampling the soundfield. The spacing of the loudspeakers determines the maximum frequency which can generally be reproduced without the introduction of artefacts due to *spatial aliasing*, an effect which has an analogy in temporal sound sampling, i.e. aliasing effects due to violations of the Nyquist theorem. The upper frequency border is refered to as the *spatial aliasing frequency*, given as [Sta97, p.73]

$$f_{sp} = \frac{c}{2 \cdot \Delta x \cdot \sin \alpha_{max}}$$

where c is the speed of sound, δx is the distance between the loudspeakers, and α_{max} is the maximum angle of incidence of the wavefront relative to the loudspeaker array. For example, the spatial aliasing frequency for a system with a distance of 10 cm between the loudspeakers can be as low as $f_{sp} = 1.7\,\text{kHz}$, which is clearly within the range of human perception. However, it has been pointed out that the human ear seems to be quite insensitive to spatial aliasing in the frequency range above 1.5 kHz [BSK04, p.3]. Although the perception of spatial aliasing artefacts still requires further examination, they can generally be described as sound coloration effects [Sta97, p.131], [DNM03, p.15]. In Wave Field Synthesis, the OPSI method (Optimised Phantom Source Imaging) combines holophonic reproduction of the low-frequency range on a woofer array with high-frequency phantom sources reproduced on a tweeter array [Wit02].

[25] Note that the axes of the dipoles need to coincide with the normal vector $\vec{n}$ [Sta97, p.15].

14.3 Restriction to Monopole Sources

Another drawback of the Kirchhoff-Helmholtz integral regarding implementations by the means of loudspeaker arrays is the requirement of speakers with monopole and dipole characteristics. It seems desirable to find a solution which requires loudspeakers of only one kind, ideally restricting itself to monopole characteristics, which are more common than dipoles. An according simplification of the Kirchhoff-Helmholtz integral (KHI, chapter 14.1) is possible, since the choice of the Green's function in the KHI (eq. 15) is not unique. Thus, it is possible to find Green's functions such that [Sta97, p.16]

$$\nabla_S \; G \cdot \vec{n} = 0$$

or

$$G = 0$$

However, the first case yields a rather complex description of G, and for $G = 0$, the description of $\nabla_S \; G \cdot \vec{n}$ becomes complicated [Sta97, p.16]. In *Wave-Field Synthesis* [Ber88]- a holophonic technique restricted to the horizontal plane - these descriptions are simplified by degenerating the closed surface S to an infinite plane surface, dividing the space into two halfs representing the domain of the sources and the domain of the receiver. An according derivation yields the two *Rayleigh integrals*, which describe how to reconstruct a wave field in the receiver domain by the means of a continuous infinite planar distribution of either monopole *or* dipole sources. For the Rayleigh integrals, S represents a perfectly reflecting surface, and thus they do *not* yield a null value for the soundfield in the source domain, as opposed to the KHI [Sta97, p.18]. 2D versions of the KHI and the Rayleigh integrals can then be derived, which replace the planar distribution of secondary sources with a distribution of line sources perpendicular to the horizontal plane (2D KHI and Rayleigh integrals, [Sta97, p.18-25]). These can again be replaced with a line array of secondary point sources in the horizontal plane ($2\frac{1}{2}$D KHI and Rayleigh integrals, [Sta97, ch.3]). The spatial truncation of this infinite array and its discretization allows for an implementation by the means of loudspeakers. Bent loudspeaker arrays can be used by interpreting them as piecewise linear arrays [Sta97, pp.104-108]. Circular array are used at the France Telecom R&D Labs [DNM03, p.4] and at the Institut für Rundfunktechnik at Munich, Germany [Rei02, p.37].

Fully periphonic implementations of the KHI are still in a rather early stage of research. Besides the mathematical difficulties of finding suitable Green's functions in order to restrict a periphonic system to one type of loudspeakers, the significant computational requirements of KHI implementations have been one of the major motivations for a restriction to the horizontal plane so far. However, it is expected that periphonic solutions will be available in the next years, since they currently are subject to wide international research. In the design of large-scale virtual environments like the AlloSphere (chapter 19.1), a possible periphonic KHI implementation has to be carefully considered, since its requirements regarding the number of loudspeakers and transmission channels is very different than in the case of Vector Base Panning (chapter 13) or Higher Order Ambisonics, a technique related to holophony (chapter 15).

14.4 Reconstruction of Enclosed Sources

We have seen that the Kirchhoff-Helmholtz integral (chapter 14.1) describes the reconstruction of a soundfield by the means of secondary sources within a volume free of sound sources. Thus, implementations of the KHI by the means of loudspeaker arrays allow for reproducing the distance of sound sources, but only if the source is located *outside* of the loudspeaker array. However, it is possible to create the illusion of enclosed sources by inverting the phase of the loudspeaker signals, which has the effect that the wavefront is focused at the position of the enclosed source. Note that the phase inversion is equivalent to temporally reversing the wavefront propagation: the wave actually propagates from the loudspeakers *towards* the inside source [DNM03, p.4], which differs from the natural case. Thus, this method suffers from the inversion of interaural time differences (see chapter 1), which so become contradictory to the correctly synthesized interaural level differences [DNM03, p.15].

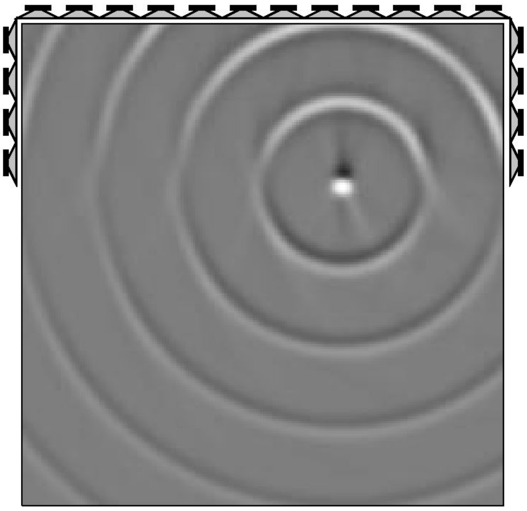


Figure 15: Reconstruction of an enclosed sound source [Boo01, p.4]

15 Higher Order Ambisonic

Higher Order Ambisonics (HOA) is an extension of the Ambisonic approach presented in chapter 9. It bases on the decomposition of a soundfield into a series of *spherical harmonic functions.* This series can be truncated at arbitrary orders. Higher orders result in a larger sweet area[26] and an extended upper frequency border of accurate soundfield reproduction, but also require more loudspeakers, channels for transimission and storage, and CPU. The resulting *scalability* is one of the major advantages of the HOA approach. As we have seen in chapter 9, the Ambisonic domain allows for efficient operations on the encoded soundfield (rotation etc.), which is also true for higher order representations.[27] We have also discussed the independence of the encoding and decoding stage in Ambisonic representations, resulting in *portability* of the encoded material, since no knowledge of the loudspeaker layout is required at encoding time. HOA is a *self-compatible* format: higher order encoded

[26] An according formula for the size of the sweet area depending on the system order has been presented in [NE99, p.448].

[27] Like the frequency domain is the prefered environment for operations on a signal's spectral content, the spatial properties of a soundfield represented in the Ambisonic domain are directly accessible.

material can be decoded to lower order reproduction systems, and vice versa.[28]

HOA can be regarded a holophonic approach, although it is not directly derived from the Kirchhoff-Helmholtz Integral (KHI, see chapter 14.1). However, both approaches provide an exact solution to the sound wave equation and can therefore be regarded as mathematically equivalent under certain assumptions [NE98, pp.2-3], [NE99, pp.442-443], [DNM03, p.11]. HOA can be described as a sweet-spot oriented low-frequency soundfield approximation: accurate reproduction of low frequencies is given over a large area, while with increasing frequency, the solution of the wave equation converges towards the center of the loudspeaker array [DNM03]. Thus, HOA offers an accurate reproduction of high frequencies at least in the center of the field, as opposed to holophonic approaches derived from the KHI, which suffer from spatial aliasing artefacts that can be globally perceived as sound colorations [DNM03, pp.14-15]. The Higher Order Ambisonic representation is not subject to spatial aliasing, since the theoretical distance of the microphones in an Ambisonic microphone arrays is zero (chapter 9). Since only a limited number of microphones can be positioned coincidently, the design of higher order microphones faces practical difficulties. Thanks to the holophonic nature of HOA, holophonic recording techniques (circular or spherical microphone arrays) can be used for Higher Order Ambisonic recordings [Pol00][29], unfortunately introducing spatial aliasing artefacts also in recorded HOA soundfields.

15.1 Derivation of the Ambisonic Representation

The Ambisonic representation bases on solving the wave equation for the *central listening spot* $\vec{r} = 0$ under the assumption of *sound sources and loudspeakers emitting plane wavefronts*. A plane wavefront is a wavefront with zero curvature. Its amplitude does not decay with distance, since its sound energy does not have to spread over an increasing surface as in the case of a spherical wavefront, which decreases with $1/r$. Note that this means that a plane wavefront does not carry any information about the distance of its source. From an isolated viewpoint - allowing us to evaluate the curvature but not the spatial envelope of a wavefront - a spherical wavefront can be regarded as plane at far distances from its source.

In the case of sound sources, the plane wavefront asumption does not impose a general restriction on us, since any soundfield can be interpreted as a superposition of plane waves [Son03, p.30]. Regarding the wavefronts emitted by the loudspeakers, these can be assumed to be plane for great loudspeaker distances. The plane wave assumptions together with the sweet spot restriction offer a remarkable simplification of the wave equation's solution, allowing for efficient implementations. However, they also remove any information about the distance of a sound source, which can thus not be encoded in the standard Higher Order Ambisonic format. In chapter 15.6 we will discuss an extended Ambisonic format capable of encoding the source distance as well.

We will now give a short description of the derivation of the Ambisonic encoding functions. More detailed derivations can be found in [Dan00], [Son03], [Zmö02],

[28] Of course, it is always the respective lower order which determines the accuracy of the decoded soundfield.

[29] Further information on Higher Order Ambisonic microphones can be found in [Dan00, pp.201-204], [DM03], and [Son03, p.87].

[Bam95]. The wave equation in a linear, lossless medium is given in the temporal domain as [Dan00, p.19]

$$\left(\Delta - \frac{1}{c^2}\frac{\delta^2}{\delta t^2}\right)\phi\left(\vec{r},t\right) = -q\left(\vec{r},t\right)$$

where c is the speed of sound, Δ is the *Laplace operator* in spherical coordinates, $\phi\left(\vec{r},t\right)$ is the velcoity potential, and $-q\left(\vec{r},t\right)$ is the volume flow rate. For the special case of monochromatic fields (see chapter 6), this simplifies to the time-invariant *Helmholtz equation* [Dan00, p.20]

$$\left(\Delta + k^2\right)\phi\left(\vec{r},\omega\right) = -q\left(\vec{r},\omega\right)$$

with the wave number $k = 2\pi f/c$. If we limit our observations to an area free of sound sources and assume free-field conditions (no reflections, diffractions, etc.), the term q becomes zero, yielding the *homogeneous Helmholtz equation* [Dan00, p.21]

$$\left(\Delta + k^2\right)\phi\left(\vec{r},\omega\right) = 0$$

By separation of variables, we can find a solution of the wave equation decomposing the soundfield into a spherical Fourier-Bessel series [Dan00, p.149]

$$p\left(\vec{r}\right) = \sum_{m=0}^{\infty}\left(2m+1\right)j^m j_m\left(kr\right)\sum_{0\leq n\leq m,\sigma=\pm 1} B_{mn}^{\sigma}Y_{mn}^{\sigma}\left(\theta_r,\phi_r\right) \tag{16}$$

where $j_m\left(kr\right)$ are the spherical Bessel functions and Y_{mn}^{σ} are the *spherical harmonic functions*.[30] Since the spherical harmonics form a set of orthogonal base vectors, they can be used to describe any function on the surface of a sphere. The components B_{mn}^{σ} will be discussed later. If we now consider a plane wavefront from direction θ,ϕ, transporting a signal s measured in the sweet spot $\vec{r}=0$, the pressure it causes at the position $\vec{r}$ of the soundfield can be described by a similar series [Dan00, p.150]:

$$p_{\theta,\phi}\left(\vec{r}\right) = s\cdot\sum_{m=0}^{\infty}\left(2m+1\right)j^m\sum_{0\leq n\leq m,\sigma=\pm 1} Y_{mn}^{\sigma}\left(\theta_s,\phi_s\right)Y_{mn}^{\sigma}\left(\theta_r,\phi_r\right)j^m\left(kr\right) \tag{17}$$

15.1.1 Derivation of the Encoding Process

By comparing equations (16) and (17) we can derive the terms B_{mn}^{σ} as [Dan00, p.150]

$$B_{mn}^{\sigma} = Y_{mn}^{\sigma}\left(\theta,\phi\right)\cdot s \tag{18}$$

Equation (18) states that the directional information of a plane wavefront from direction θ,ϕ can be encoded to a set of signals B_{mn}^{σ} by multiplying the signal s with the value of the respective spherical harmonic function Y_{mn}^{σ} in the direction θ,ϕ. Since this describes exactly the Ambisonic encoding process of a spatialized audio signal, the signals B_{mn}^{σ} are refered to as the *Ambisonic channels*. Due to practical limitations regarding computational power and storage, we have to truncate the

[30] A function with arguments azimuth θ and elevation ϕ is refered to as *spherical*, since its function value is defined on the surface of a sphere.

infinite series from equation (17) at an order M, which we call the *Ambisonic order*.

Let's take a closer look at the spherical harmonic functions Y_{mn}^{σ}: for each order m of the series, $0 \leq n \leq m$ different spherical harmonic functions exist for both values of the superscript $\sigma = \pm 1$. Since the functions Y_{m0}^{-1} are zero everywhere and for any m, they can be ignored, leaving us with $n = 2m + 1$ spherical harmonics for each order m. A 3D Ambisonic system of order M combines all spherical harmonics of orders $0 \leq m \leq M$, yielding the total number of Ambisonic channels N as [Zmö02, p.27]

$$\boxed{N = (M+1)^2} \qquad (19)$$

For example, the first-order B-format is given by the zeroth order function plus the three functions of first order (equation (8)). The spherical harmonic functions themselves consist of the associated Legendre functions[31] P_{mn} and various sine and cosine terms [Dan00, p.149]

$$Y_{mn}^{\sigma}(\theta, \phi) = N_{mn} \cdot P_{mn}(\sin\phi) \begin{cases} \cos(n\theta) & \text{for } \sigma = 1 \\ \sin(n\theta) & \text{for } \sigma = -1 \end{cases}$$

N_{mn} denotes a normalization term, available in different flavors. A popular version is the *Schmidt semi-normalization*, given as [Dan00, p.303]

$$N_{mn} = \sqrt{\epsilon_n \frac{(m-n)!}{(m+n)!}} \qquad \epsilon_0 = 1 \text{ and } \epsilon_n = 2 \text{ for } n \geq 1 \qquad (20)$$

For convenience, we will now rewrite equation (18) by combining all N Ambisonic channels in compact vector representation. For k sound sources, the *Ambisonic encoding equation* is then given as

$$\boxed{\vec{B} = \sum_{i=1}^{k} \vec{Y}(\theta_i, \phi_i) \cdot s_i} \qquad (21)$$

As equation(21) shows, the encoded soundfields of several sources can simply be superponed. Regarding the order of the Ambisonic channels in the rows of column vector $\vec{B}$, we will implicitely use Daniel's pattern for $\binom{\sigma}{mn}$ throughout this thesis [Dan00]

$$\binom{1}{00};\binom{1}{11},\binom{-1}{11},\binom{1}{10};\binom{1}{22},\binom{-1}{22},\binom{1}{21},\binom{-1}{21},\binom{1}{20};\;\cdots\;,\binom{-1}{M1},\binom{1}{M0}$$

Each element of $\vec{B}$ represents an Ambisonic channel of the encoded soundfield.

15.1.2 Derivation of the Decoding Process

Our task is now to find a set of loudspeaker signals which can be superponed to exactly reproduce the encoded soundfield in the sweet spot of the reproduction system. In order to express this in an equation system, we use a representation of the loudspeaker signals equivalent to the encoded soundfield, i.e. a spherical harmonic decomposition of a plane wavefront (eq. 17), depending on the position of

[31] The associated Legendre functions themselves are derived from the Legendre polynomials (see [Dan00, p.303]).

the respective loudspeaker. Equivalency between the encoded soundfield of a single source and the resynthesized soundfield is given if [Zmö02, p.41]

$$Y_{mn}^{\sigma}(\theta,\phi)\cdot s \equiv \sum_{j=1}^{L} Y_{mn}^{\sigma}(\theta_j,\phi_j)\cdot p_j$$

where p_j is the signal of the j-th loudspeaker at direction θ_j,ϕ_j. From this and equation (21), we can derive the *Ambisonic re-encoding equation* in compact matrix form as [Dan00, p.158]

$$\boxed{\vec{B} = \mathbf{C}\cdot\vec{p}} \tag{22}$$

$\vec{p}$ is a column vector with entries representing the loudspeaker signals. **C** is refered to as the *re-encoding matrix*, containing the re-encoded loudspeaker directions θ_j,ϕ_j.[32] The rows of **C** represent the N spherical harmonic components, and its columns represent the L loudspeakers of the reproduction array [Zmö02, p.41].

$$\mathbf{C} = \begin{pmatrix} Y_{00}^{1}(\theta_1,\phi_1) & Y_{00}^{1}(\theta_2,\phi_2) & \cdots & Y_{00}^{1}(\theta_j,\phi_j) & \cdots & Y_{00}^{1}(\theta_L,\phi_L) \\ Y_{11}^{1}(\theta_1,\phi_1) & Y_{11}^{1}(\theta_2,\phi_2) & \cdots & Y_{11}^{1}(\theta_j,\phi_j) & \cdots & Y_{11}^{1}(\theta_L,\phi_L) \\ Y_{11}^{-1}(\theta_1,\phi_1) & Y_{11}^{-1}(\theta_2,\phi_2) & \cdots & Y_{11}^{-1}(\theta_j,\phi_j) & \cdots & Y_{11}^{-1}(\theta_L,\phi_L) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ Y_{M0}^{1}(\theta_1,\phi_1) & Y_{M0}^{1}(\theta_2,\phi_2) & \cdots & Y_{M0}^{1}(\theta_j,\phi_j) & \cdots & Y_{M0}^{1}(\theta_L,\phi_L) \end{pmatrix}$$

From equation (22), the loudspeaker driving signals can be derived by inversion of the matrix **C**, yielding the *Ambisonic decoding equation* [Zmö02, p.41]

$$\boxed{\vec{p} = \mathbf{C}^{-1}\cdot\vec{B} = \mathbf{D}\cdot\vec{B}} \tag{23}$$

where the inverted **C** matrix is also refered to as the *decoding matrix* **D**. We will discuss the derivation of the decoding matrix extensively in chapter 15.4. **D** has L rows and N columns. To guarantee that the entire directional information encoded in the soundfield will be restored, the equation system has to satisfy the relation [Zmö02, p.42][33]

$$\boxed{L \geq N} \tag{24}$$

which means that there should be at least as many loudspeakers in our array as there are Ambisonic channels, i.e. $L \geq (M+1)^2$ for a 3D system. Daniel has mentioned that although it has been shown that the smallest average reconstruction error is given for a relation $L = N$, reproduction systems with large listening areas will benefit from additional loudspeakers such that $L > N$, since listeners at off-center positions are less likely to approach an isolated loudspeaker which then dominates their auditory localization. [Dan00, pp.113,179]. From equations (24) and 19, we can see that the number of available loudspeakers determines the maximum reproducable Ambisonic order M. As we have noted, the accuracy of soundfield reproduction increases with the order of a system. The error introduced due to the truncation at M can be described by the *complex normalized wavefront mismatch error* [VL87] or the *normalized truncation error* [WA01].

[32] Note that **C** is time-invariant, since it is an abstraction of the (static) loudspeaker array.

[33] Note that in [Zmö02], N denotes the number of speakers and L the number of Ambisonic channels. Different conventions used in the literature are also compared in the appendix B.2.

15.1.3 Equivalent Panning Functions

The Ambisonic encoding and decoding process can also be described by the means of *equivalent panning functions*, which can be used equivalently to spatialize a sound source onto a loudspeaker array. This is of interest regarding interpretation of Ambisonic systems: the polar diagram of an equivalent panning function describes the required directivity pattern that a microphone feeding the respective loudspeaker would need to have. For a basic 3D encoding/decoding system, the equivalent panning function is given as [DNM03, p.7]

$$G(\kappa) = \frac{1}{N} \sum_{m=0}^{M} (2m+1)\, g_m P_m (\cos(\kappa))$$

where κ is the angle between the directions of the loudspeaker and the virtual source.

15.2 Ambisonic Encoding Functions

Several modificated *encoding conventions* have been presented in the literature. Daniel distinguishes the following flavors for 3D encoding function flavors [Dan00, pp.156]:

- SN3D: semi-normalization 3D
- N3D: full normalization 3D
- MaxN: max normalization
- FuMa: Furse-Malham set

The SN3D encoding functions are the spherical harmonic functions with semi-normalization as applied in equation (20). Since they can be generically defined for arbitrary orders, Daniel suggests to use them as the reference encoding convention [Dan00, p.157]. The N3D convention is of interest regarding the evaluation of loudspeaker layouts in Ambisonic soundfield reproduction (see chapter 18). The MaxN convention features weighting factors of the spherical harmonics which ensure that the maximum absolute value of each function is 1, which is of relevance regarding digital signal representations [Mal03a, p.65]. The FuMa set is equivalent to the MaxN convention, except for an additional $1/\sqrt{2}$ weighting factor for the zeroth order channel, following a B-format tradition (see chapter 9).[34] Conversion factors among different encoding conventions are presented in [Dan00, pp.156-157] and [Dan03, p.13]. In table 1, the SN3D encoding functions are presented up to third order. They are also shown in figure 16. The abbreviations for the third order channel (O, L, M, etc.) (which [Dan00] does not provide) follow a convention applied in [Mal03a].[35]

It can be seen from figure 16, that for each order, the two functions $m = n$ only depend on the azimuth, but not on the elevation, which makes it possible to separate *horizontal* and *vertical* Ambisonic channels. This allows for an efficient implementation of horizontal-only Ambisonic systems, based on the *cylindrical harmonic functions*, presented here in their semi-normalized form:

[34] In [Dan00], the Furse-Malham set is only defined for orders $M \leq 2$. In [Mal03a, p.65], David Malham has suggested to follow the MaxN convention for higher orders of the FuMa set.

[35] Note that for third order [Zmö02] uses the same abbreviations, but for different channels. Various conventions used in the literature are compared in the appendix B.2.

order	**B**	$^{\sigma}_{mn}$	$Y^{\sigma(SN3D)}_{mn}(\theta,\phi)$
0	W	$^{1}_{00}$	1
1	X	$^{1}_{11}$	$\cos\theta\cos\phi$
	Y	$^{-1}_{11}$	$\cos\theta\cos\phi$
	Z	$^{1}_{10}$	$\sin\phi$
2	U	$^{1}_{22}$	$\frac{\sqrt{3}}{2}\cos(2\theta)\cos^2\phi$
	V	$^{-1}_{22}$	$\frac{\sqrt{3}}{2}\sin(2\theta)\cos^2\phi$
	S	$^{1}_{21}$	$\frac{\sqrt{3}}{2}\cos\theta\sin(2\phi)$
	T	$^{-1}_{21}$	$\frac{\sqrt{3}}{2}\sin\theta\sin(2\phi)$
	R	$^{1}_{20}$	$\frac{3\sin^2\phi-1}{2}$
3	P	$^{1}_{33}$	$\sqrt{\frac{5}{8}}\cos(3\theta)\cos^3\phi$
	Q	$^{-1}_{33}$	$\sqrt{\frac{5}{8}}\sin(3\theta)\cos^3\phi$
	N	$^{1}_{32}$	$\frac{\sqrt{15}}{2}\cos(2\theta)\sin\phi\cos^2\phi$
	O	$^{-1}_{32}$	$\frac{\sqrt{15}}{2}\sin(2\theta)\sin\phi\cos^2\phi$
	L	$^{1}_{31}$	$\sqrt{\frac{3}{8}}\cos\theta\cos\phi\left(5\sin^2\phi-1\right)$
	M	$^{-1}_{31}$	$\sqrt{\frac{3}{8}}\sin\theta\cos\phi\left(5\sin^2\phi-1\right)$
	K	$^{1}_{30}$	$\frac{1}{2}\sin\phi\left(5\sin^2\phi-3\right)$

Table 1: Higher Order Ambisonic SN3D encoding functions [Dan00, p.151]

$$\begin{aligned} Y^{1(SN2D)}_{m}(\theta) &= \cos(m\theta) \\ Y^{-1(SN2D)}_{m>0}(\theta) &= \sin(m\theta) \end{aligned}$$

Note that the SN2D encoding functions differ from their SN3D counterparts not only by neglecting the elevation ϕ, but also in different weightings of the higher order channels. Conversion factors from 3D to 2D representations can be found for both, the full normalized and semi-normalized encoding convention in [Dan03, p.13]. In a horizontal-only system, the number of Ambisonic channels N for an order M is given as

$$N^{2D} = 2M + 1$$

representing the W channel plus the two additional channels for each order $m \geq 1$. This separation of horizontal and vertical channels also allows for the implementation of *hybrid representations*, typically used in optimizing the number of channels in a HOA system by encoding the horizontal parts of the soundfield to a higher order than its vertical parts, taking into account the higher spatial resolution of the ear in the horizontal plane (see chapter I). The definition of hybrid order encoding functions is not trivial, due to the different appearances of 2D and 3D encoding functions. Encoding functions for a '3rd+1st' representation have been described in [Zmö02, pp.33,39]. Daniel discusses hybrid representations in [Dan00, pp.154] and describes an example of '2nd+1st' *de*coding of uniformly N3D encoded material in [Dan00, pp.199-201]. A disadvantage of hybrid order systems is that they do not allow for soundfield rotation about the x and y axis,

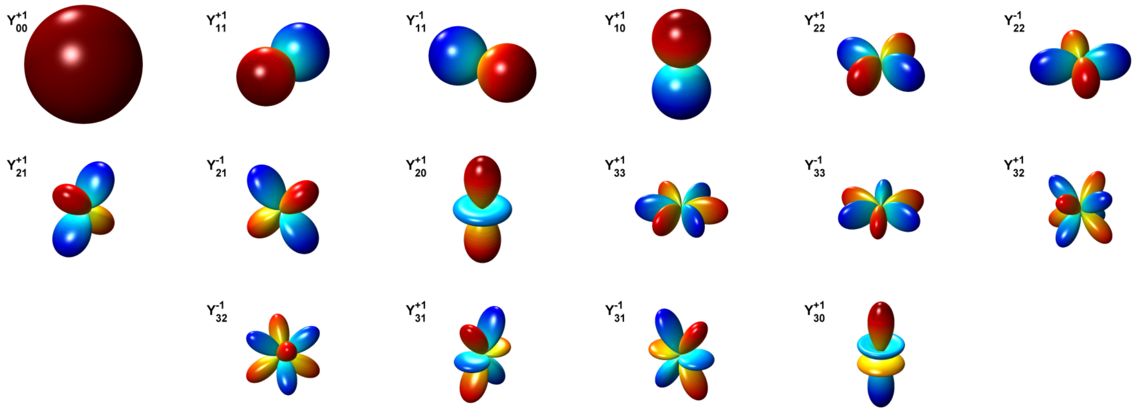


Figure 16: Spherical harmonic functions up to third order

due to the interdependence of horizontal and vertical Ambisonic channels in the tilt and tumble matrices (see chapter 15.3.1). Note that the term *mixed order systems* refers to a different technique describing the simultaneous reproduction of multiple Ambisonic soundfields encoded at different orders on one loudspeaker array [Dan00, pp.281-282].

15.3 Soundfield Operations

Similar to the frequency domain, which allows for straightforward operations on the spectral content of a signal, the Ambisonic domain is an ideal representation regarding operations on the spatial properties of a soundfield. From a mathematical point of view, an operation on an Ambisonic encoded soundfield $\vec{B}$ is achieved my multiplication with a matrix $\mathbf{T}$, representing the desired soundfield transformation:

$$\boxed{\vec{B}' = \mathbf{T} \cdot \vec{B}} \tag{25}$$

where $\vec{B}$ is the column vector representing the N Ambisonic channels, and $\mathbf{T}$ is a square matrix with N rows and columns. The appearance of $\mathbf{T}$ determines the soundfield operation. If it is a diagonal matrix, it represents a non-interdependent operation on the Ambisonic channels, like the *mirroring* operation described in chapter 15.3.3. For other transformations, like soundfield *rotation* (chapter 15.3.1), the Ambisonic channels depend on each other. Generally, the elements in $\mathbf{T}$ represent simple gain factors. More complex soundfield transformations, where the elements of $\mathbf{T}$ represent filters, delays, etc., might have interesting physical analogies or simply be of creative interest. Subsequent soundfield operations can be applied by operations $\vec{B}' = \mathbf{T}_1 \cdot \mathbf{T}_2 \cdot \vec{B}$.

15.3.1 Soundfield Rotation

The most popular operation in the Ambisonic domain is the rotation of entire soundfields, which turns out to be comparably easy to implement. A common application of this is found in headphone-based systems where the loudspeaker signals of a virtual Ambisonic system are encoded into a stereo signal by the means of head-related transfer functions (HRTFs): In combination with a head tracking device, soundfield rotation then allows for head rotations relative to the soundfield [Son03, p.122]. Rotation matrices can be defined for rotation about the x axis (*tilt* or *roll*), the

y axis (*tumble* or *pitch*), the z axis (*rotate* or *yaw*), or any combination of these, allowing for arbitrary rotation axes. There is a generic definition of matrices for rotation around the z axis. This is due to the symmetry of the spherical harmonic functions with regards to the horizontal plane, causing the channel interdependence to be limited to channels pairs $Y^1_{m\lambda}, Y^{-1}_{m\lambda}$ with $\lambda = 1, 2, ..., m-1$ for all orders m. For the SN3D encoding convention, the matrix elements of each pair $m = n$ for $\sigma = \pm 1$ are [Dan00, p. 166].

$$\mathbf{Z}^{\pm 1}_{m\lambda}(\gamma) = \begin{pmatrix} \cos(\lambda\gamma) & -\sin(\lambda\gamma) \\ \sin(\lambda\gamma) & \cos(\lambda\gamma) \end{pmatrix}$$

where γ is the rotation angle. The signal Y^1_{m0} is invariant to z axis rotation, resulting in a matrix element of 1. All other elements of the matrix are zero. For example, the z rotation matrix for a second order SN3D encoded Ambisonic soundfield using our standard channel ordering convention (see chapter 15.1.1) is

$$\mathbf{Z}^{M=2}(\gamma) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sin\gamma & \cos\gamma & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos 2\gamma & -\sin 2\gamma & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sin 2\gamma & \cos 2\gamma & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cos\gamma & -\sin\gamma & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Note that the z rotation matrix for the FuMa encoding convention looks exactly the same, since identical FuMa weightings are applied on the channel pairs $Y^1_{m\lambda}, Y^{-1}_{m\lambda}$ and cancel each other out during the operation.

The tilt and tumble operations are somewhat more complex. The according matrices for orders up to $M = 2$ can be found quite easily, but for higher orders, the task becomes non-trivial. Tilt, tumble (and rotation) matrices up to third order are presented in [Zmö02, p.36] for the FuMa encoding convention. A description of research on higher-order matrices in other scientific fields is given in [Mal03a, p.67]. Also, the tilt and tumble operations cannot be applied to hybrid order encoded material (see chapter 15.2).

Tilt, tumble, and rotation matrices can be combined in a rotation matrix $\mathbf{R}$ in order to define an arbitrary rotation axis [Zmö02, p.35]:

$$\vec{B}' = \mathbf{R}(\alpha, \beta, \gamma) \cdot \vec{B} = [\mathbf{X}(\alpha) \cdot \mathbf{Y}(\beta) \cdot \mathbf{Z}(\gamma)] \cdot \vec{B}$$

Since the order in which the matrices are multiplied matters, other combinations can be used (e.g. $Z \cdot X \cdot Y$).

15.3.2 Dominance (Zoom) and Focus

The *dominance* operation is used to make sounds in the front direction louder, and reduce the gain of rear sounds. It is worth pointing out that this is an operation in the Ambisonic domain, independent from the loudspeaker layout. Although also known as *zoom*, there is no direct analogy to an optical zoom lense, since the angular spread of sounds in the zoom direction is reduced and that of sounds in the opposite direction is increased [Mal03a, p.115]. Daniel refers to the operation as 'distortion de la perspective' [Dan00, p.166]. For an Ambisonic order $M = 1$, the dominance operation is implemented by the means of a Lorentzian transformation, known from the theory of relativity [Mal03a, p.50]. Two equivalent versions of the according matrix have been described [Dan00, p.166]:

$$\mathbf{L}_{\lambda}^{(SN3D)} = \begin{pmatrix} \frac{\lambda+\lambda^{-1}}{2} & \frac{\lambda-\lambda^{-1}}{2} & 0 & 0 \\ \frac{\lambda-\lambda^{-1}}{2} & \frac{\lambda+\lambda^{-1}}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where $\lambda =\in [0 .. \infty]$, and

$$\mathbf{L}_{\mu}^{(SN3D)} = \begin{pmatrix} 1 & \mu & 0 & 0 \\ \mu & 1 & 0 & 0 \\ 0 & 0 & \sqrt{1-\mu^2} & 0 \\ 0 & 0 & 0 & \sqrt{1-\mu^2} \end{pmatrix}$$

where $\mu =\in [-1.. + 1]$.

Daniel has noted that the dominance function results in a directional distortion: a sound originally coming from the front $\theta = 0, \phi = 0$ will be transformed to a position $\theta', \phi = 0$ with

$$\theta' = \arccos \frac{\mu + \cos\theta}{1 + \mu \cos\theta}$$

For higher orders $M > 1$, this operation cannot be realized without distorting the plane wave characteristics. Different concepts have been discussed but not implemented yet [Mal03a, p.68], [Dan00, p.166]. The *focus* operation is a special case of dominance with $\mu = 1$. For this case, Daniel has described a possible extension to higher orders [Dan00, p.167].

15.3.3 Mirroring, W-Panning

In this chapter, we will discuss soundfield operations with non-interdependent Ambisonic channels, i.e. representing simple weightings of the Ambisonic channel gains. In this case, the transformation matrix becomes a diagonal matrix W

$$\mathbf{W} = \begin{pmatrix} g_{00}^{1} & 0 & 0 & \cdots & 0 \\ 0 & g_{11}^{1} & 0 & \cdots & 0 \\ 0 & 0 & g_{11}^{-1} & \cdots & 0 \\ \cdots & \cdots & \cdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & g_{M0}^{1} \end{pmatrix}$$

where $-1 < g_{mn}^{\sigma} < +1$ represents the gain factors. This matrix is used in the process of *W-panning* to control the apparent size of a sound source, similar to the spread control in Vector Base Panning (see chapter 13). The apparent spread of the source will increase when the ratio of the omnidirectional W channel is increased compared to the other channels, resulting in the extreme case of an *interior* effect: when only the W channel is playing and all other channels are muted, the same signal appears from all directions, giving an effect of being 'inside' the sound source.

A derivation of suitable panning functions is given in [Men02].

A special case of the matrix $\mathbf{W}$ is applied in soundfield *mirroring*:

$$\mathbf{M} = \begin{pmatrix} +1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & 0 \\ 0 & 0 & -1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

with $g_{00}^{1} = +1$ and $g_{mn}^{\sigma} = -1$ for $m \neq 0$. This operation has the effect that all sound sources will appear in positions diametrically opposite of their original ones. This is due to the symmetry of the spherical harmonic functions. As can be seen from figure 16, the value of a spherical harmonic function between a given position and its diametrically opposite position only varies by sign, but not by value, except for the W channel, which is constantly one everywhere. In first order systems, selected axes can be mirrored by adjusting the gain of either the X, Y or Z channel: for example, multiplying the gain factor of only the X channel will move all sounds at left front positions to the left rear [Mal03a, p.49].

15.4 Decoding Methods

We have already noted in chapter 15.1.2 that in order to decode an Ambisonic soundfield to a loudspeaker array, we need to invert the $\mathbf{C}$ matrix, which contains the re-encoded loudspeaker positions. However, a matrix can only be inverted if it is square. For $\mathbf{C}$, this is only the case if the number of loudspeakers L (number of columns in $\mathbf{C}$) matches the number of Ambisonic channels N (number of rows). We will discuss a *pseudoinverse* approach, which does not face this limitation. However, it introduces different problems, leading us to alternative decoding methods refered to as *projection* and *regularization*.

15.4.1 Pseudoinverse

A general approach to inverting non-square matrices ($L \neq N$) is given by the *pseudoinverse*. The pseudoinverse is defined as [BK]

$$\mathrm{pinv}\left(\mathbf{C}\right) = \mathbf{C}^{T} \cdot \left(\mathbf{C} \cdot \mathbf{C}^{T}\right)^{-1}$$

for $L \geq N$, and as

$$\mathrm{pinv}\left(\mathbf{C}\right) = \left(\mathbf{C}^{T} \cdot \mathbf{C}\right)^{-1} \cdot \mathbf{C}^{T}$$

for $L < N$ and full rank of C.

where $\mathbf{C}^{T}$ is the transposed $\mathbf{C}$ matrix, and both, $\mathbf{C} \cdot \mathbf{C}^{T}$ and $\mathbf{C}^{T} \cdot \mathbf{C}$ are always square matrices that can be inverted. The pseudoinversion of the re-encoding matrix yields the decoding matrix, which is given as

$$\mathbf{D} = \mathrm{pinv}\left(\mathbf{C}\right)$$

The quality of the pseudoinverse depends on the condition number of the $\mathbf{C}$ matrix, i.e. the regularity of the loudspeaker layout (see chapter 18.1.2). For instance, the direction of a sound source decoded with the pseudoinverse only matches the direction of the energy vector (chapter 6.1) if $L > N$ and if the loudspeaker array satisfies certain criteria regarding its regularity [Dan00, p.159]. Extended irregularities in the layout result in instabilities of the decoding matrix.

15.4.2 Projection

Daniel has shown that for loudspeaker layouts which are regular in the Ambisonic sense (chapter 18.1.2) and if the N3D encoding convention is applied, the pseudoinverse simplifies to [Dan00, p.179]

$$\operatorname{pinv}(\mathbf{C}) = \frac{1}{L} \cdot \mathbf{C}^T$$

However, it is common practice to use a decoding matrix

$$\mathbf{D} = \frac{1}{L} \cdot \mathbf{C}^T$$

also for non-regular layouts, which can be mathematically interpreted as a *projection* of the Ambisonic channels. The advantage of this technique is that the resulting decoding matrix **D** is generally more stable than if derived as the pseudoinverse of **C**. However, directional distortions and problems regarding the energy balance of a spatialized sound source are the drawbacks of this approach [Dan00, p.192].

15.4.3 Regularization

The technique of deriving the **C** matrix of a virtual regular loudspeaker layout, although the actual layout is irregular (see chapter 18.1.2), is refered to as *regularization* of a layout. Daniel has pointed out that while this technique preserves the energy balance of spatialized sound sources, directional distortions can occur as well [Dan00, p.191].

15.5 Decoder Flavors

We have already seen that in Ambisonic systems, *all* loudspeakers contribute to the reproduction of a virtual source in a specific direction. In theory, the signals from loudspeakers in directions other than the one of the source cancel each other out in the sweet spot. However, diffraction effects due to the presence of the listener's head disturb this destructive interference, which causes the signals of loudspeakers in the opposite direction of the sound source to be perceived as independent sources. This has led to a modification of the basic decoder, originally suggested for horizontal-only systems by David Malham [Mal99] and extended to periphonic systems by Daniel [Dan00]. It is refered to as the *in-phase* decoder, which avoids contributions from opposite loudspeakers for the price of an increased spread of the sound source (see figure 17). Other modifications have been suggested, which can generally be described as a multiplication of the decoding matrix **D** with a diagonal matrix **Γ**. The elements in **Γ** represent additional weighting factors for the spherical harmonic components of the loudspeaker signals, depending on their respective order m.

$$\boxed{\mathbf{D} = \mathbf{D} \cdot \mathbf{\Gamma}} \tag{26}$$

where

$$\mathbf{\Gamma} = \operatorname{Diag}\left(\begin{bmatrix} g_0 \cdots \overbrace{g_m \cdots \cdots g_m}^{2m+1} \cdots \overbrace{g_M \cdots g_M}^{2M+1} \end{bmatrix}^T\right)$$

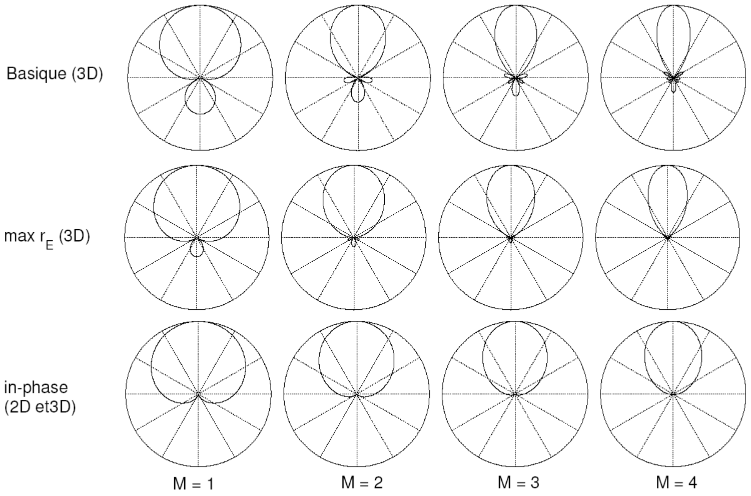


Figure 17: Equivalent panning functions of 3D basic, max r_E and in-phase decoders for orders one to four [Dan00, p.160]

- The *basic decoder* represents the unmodified decoding matrix (i.e. $\mathbf{\Gamma} = \mathbf{I_N}$, where $\mathbf{I_N}$ is the N-by-N unity matrix).
- It has already been noted that the *in-phase decoder* avoids contributions from loudspeakers in the direction opposite of the source. This is particularly relevant for off-center listening positions, where the danger is given that the direction of the closest loudspeaker is erroneously perceived as the source's direction. Thus, the in-phase decoder is regarded as suitable for systems with large listening areas, like multi-user virtual environments. The additional sound source spread can be compensated by increasing the Ambisonic order M [Son03, p.46]. The in-phase decoder can be applied to non-regular layouts if the decoder matrix $\mathbf{D}$ has been derived by the method of projection (chapter 15.4.2) [Dan00, p.192].
- The *max r_E decoder* represents a compromise between the basic and the in-phase decoder. Daniel has noted that this decoding flavor can also be applied to *semi*regular layouts (see chapter 18.1.2) [Dan00, p.188].
- *Window-applied decoding* has been described as a generalization of the decoder modification process [Son03, pp.43,133]. It bases on the observation that the modification matrix $\mathbf{\Gamma}$ is equivalent to the application of a windowing function on the equivalent panning functions (chapter 15.1.3), allowing for the typical tradeoffs known from FIR filter design (main lobe width vs. side lobe attenuation, etc.). The parameters of $\mathbf{\Gamma}$ can thus be derived by an according window design procedure.

A derivation of the matrix elements g_M for the in-phase and max r_E decoder is given in [Dan00, pp.184-185], including the factors for 2D and 3D decoders, as well as for both, amplitude and energy preserving decoders. Daniel has also described how useful application of different decoder flavors depends on the listening position and on the reproduced frequency [Dan00, p.160]: while the basic decoder gives good results in the central listening spot and for low frequency areas, the in-phase decoder is the right choice for high-frequency reproduction at off-center listening positions. The max r_E decoder's useful range is located between these two extremes (see figure 18. In [Dan00, p.175], transition frequencies between basic and max r_E decoding for an off-center position are presented for both, 2D and 3D systems of orders from

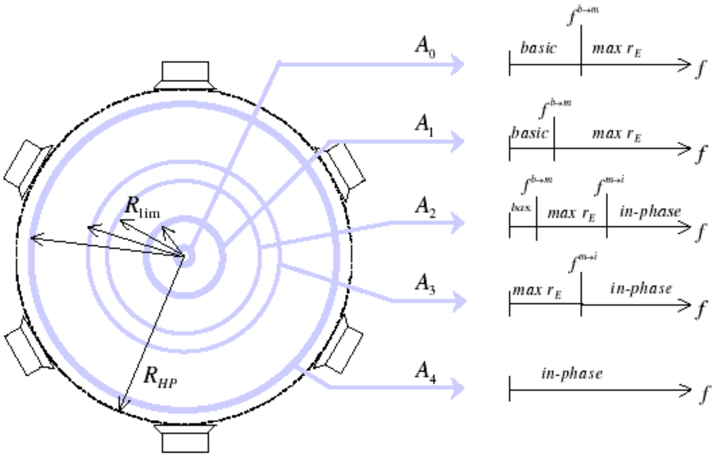


Figure 18: Prefered listening areas and frequency ranges for various decoders flavors [Dan00, p.160]

one to five. Daniel also notes that no frequencies for the transition from max r_E to in-phase decoding have been determined yet [Dan00, p.160].

15.6 Source Distance Encoding (NFC-HOA)

We have already noted that the reconstruction of the distance of a sound source is an essential feature of periphonic sound spatialization systems. However, due to the plane wave assumption used in the derivation of the Ambisonic approach, no distance information is included in the Higher Order Ambisonic encoding format, since a plane wave does not decay with distance. Like in Vector Base Panning (chapter 13), it is possible to simulate distance cues by adding artificial reverberation. An interesting approach within the Ambisonic domain has been discussed in [Zmö02, p.87-88]. Here, the four zeroth and first order signals (W,X,Y,Z) are extracted from a Higher Order Ambisonic soundfield and decoded to a virtual loudspeaker tetrahedron. The decoded signals are then reverberated and encoded back into the Ambisonic domain, where they are added again to the higher order soundfield. A dynamical weighted addition can be implemented in order to receive a 'distance panpot'. The resulting soundfield is decoded to the physical loudspeaker layout. This method allows for controlling the direction of the diffuse reverberation, while avoiding the impression of clearly localizable reverberation 'sources' by limiting the according Ambisonic order to one. However, this approach does only support 'outside' sources with a radius r greater than the loudspeaker radius R.

Two different methods have been described in the literature which are capable of reproducing 'inside' sources $r < R$ using Higher Order Ambisonic representations. For horizontal-only systems, Sontacchi and Höldrich have presented a hybrid method of Wave Field Synthesis and Higher Order Ambisonic [SH02a]. In the first stage of this process, the gain factors of a virtual loudspeaker array[36] are calculated by Wave Field Synthesis (or an equivalent least mean squares approach) for each sound source, considering its radius, but neglecting its direction. In the second stage, each signal feeding a virtual loudspeaker is Ambisonic encoded according to the loudspeaker's direction, resulting in one Ambisonic soundfield for each distance-

[36] The radius of the virtual loudspeakers should correspond to the average radius of the real loudspeaker array [SH02b]. Their number and spacing are regarded design parameters of the system

weighted source. Each source can be then be spatialized to its direction θ, ϕ by the means of soundfield rotation (chapter 15.3.1). The Ambisonic soundfields of all sources are then superponed, and the resulting field is subject to soundfield operations, storage and decoding like any other Ambisonic encoded material. Thanks to the Wave Field Synthesis engine, 'inside' sources ($r < R$) can be encoded (chapter 14.4), while the final Ambisonic representation guarantees a lower number of channels for transmission and storage. Any other spatialization algorithm can be chosen to encode the direction of the sources, but in Ambisonics, the required soundfield rotations are easy to achieve. A periphonic version of this distance coding algorithm could be implemented by replacing the Wave Field Synthesis engine with a 3D implementation of the Kirchhoff-Helmholtz integral (see chapter 14).

The second approach, presented by Daniel, works entirely within the Ambisonic domain [Dan03]. Its theory bases on the compensation of artefacts introduced by the *plane wave assumption* used in deriving the Ambisonic encoding format (see chapter 15.1.1). The signals emitted by most real sound sources and loudspeakers can be modelled much more accurately as spherical wavefronts, and the Ambisonic format can also be derived by a decomposition of a spherical rather than a plane wavefront. Daniel has shown [Dan03, p.3] that this changes the Ambisonic encoding equation (see eq. 21) to

$$B_{mn}^{\sigma} = F_m^{(r/c)}(\omega) \cdot Y_{mn}^{\sigma}(\theta, \phi) \cdot s$$

where $\omega = 2\pi f$. Compared to the original encoding format, transfer functions $F_m^{(r/c)}$ are introduced for the respective order m of each spherical harmonic component, where r/c represents the delay due the source distance r and the finite speed of sound c. These functions show that the finite sound source distance results in a *bass boost effect* which increases with the order of the spherical harmonic components. For higher orders, these functions become unstable, i.e. their bass boost is infinite. This shows that the standard Higher Order Ambisonic encoding format is physically unable to represent a finite source distance.

On the decoding side, the differences between spherical and plane wavefront propagation yields that the finite distance of the loudspeakers results in a *bass boost effect* [Dan03, p.5]. It has already been suggested by Gerzon to compensate for this bass boost effect at decoding time [Ger92, p.52] by adjusting the decoding matrix with a set of filters [Dan03, p.5]:

$$\vec{p} = \mathbf{D} \cdot \mathbf{\Gamma}\left(\frac{1}{F_m^{(R/c)}(\omega)}\right) \cdot \vec{B} \tag{27}$$

where $\mathbf{\Gamma}$ is a diagonal matrix with elements representing filters depending on the loudspeaker distance R. Fortunately, these inverse filters are stable. Note that this operation is equivalent to a filtering operation on the Ambisonic channels prior to decoding. The near-field compensation of the loudspeaker signals can thus also be applied at *en*coding time. Daniel has observed that at the same time, the finite distance of the sound sources can be considered as well, since the combination of both functions $F_m^{(r/c)}$ (finite source distance) and $F_m^{(R/c)}$ (finite speaker distance) results in a set of *distance coding filters* [Dan03, p.5]:

$${H_m}^{NFC(r/c,R/c)}(\omega) = \frac{F_m^{(r/c)}(\omega)}{F_m^{(R/c)}(\omega)}$$

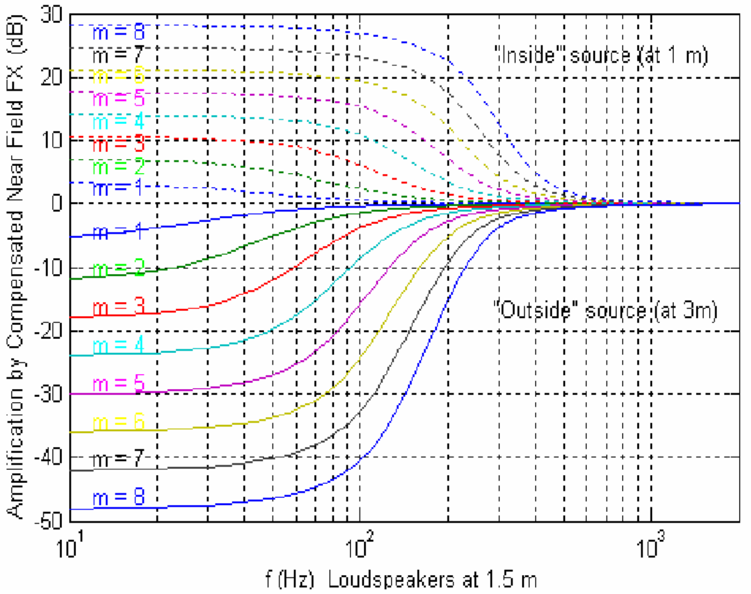


Figure 19: Distance coding filters ${H_m}^{NFC(r/c,R/c)}$ [Dan03, p.6]

Figure (19) shows two examples of distance coding filters ${H_m}^{NFC(r/c,R/c)}$ for Ambisonic orders one through eight: one for a source outside the loudspeaker array, and one for an inside source. These filters, which are stable also for high orders, can be used to define a new Ambisonic encoding format called *Near-Field Compensated Higher Order Ambisonic* (NFC-HOA) [Dan03]:

$$B_{mn}^{\sigma}{}^{NFC(R/c)} = {H_m}^{NFC(r/c,R/c)}(\omega) \cdot Y_{mn}^{\sigma}(\theta, \phi) \cdot s$$

These encoding functions now include the distance of the virtual source r as well as the loudspeaker radius R. The required knowledge of the loudspeaker radius at *en*coding time is the drawback of these functions. However, Daniel has shown how the transfer functions ${H_m}^{NFC}$ can also be used to adapt an NFC-HOA encoded soundfield to a different loudspeaker radius R' than was assumed at encoding time [Dan03, p.7]:

$$B_{mn}^{\sigma}{}^{NFC(R'/c)} = {H_m}^{NFC(R/c,R'/c)} B_{mn}^{\sigma}{}^{NFC(R'/c)}$$

Regarding the reconstruction of *inside sources* ($r < R$), Daniel has pointed out that NFC-HOA can be understood as an extrapolation of the soundfield from the sweet spot to the source radius r. Since no temporal reversed wavefront propagation is applied as in Wave Field Synthesis, the problem of inverted interaural time differences (see chapter 14.4) does not occur. [DNM03, p.13].

As can be seen from figure (19), the gains of the NFC filters are finite but still considerably high for large Ambisonic orders, introducing *noise* in practical filter implementations. Daniel has also described the equalization functions necessary to convert HOA signals recorded by microphone arrays to the NFC-HOA format [DNM03, p.10], which require even higher gain factors, revealing the increased effort required to extrapolate the soundfield from the radius of the microphone array R_{mic} to the loudspeaker radius R (e.g. 120 dB for a fourth order component at 100 Hz with $R_{mic} = 5cm$ and $R = 1m$!). Addressing this problem by choosing a greater R_{mic} on the other hand increases the problem of spatial aliasing. Daniel has also derived *equivalent panning functions* (chapter 15.1.3) for the NFC-HOA format in [DNM03, p.9].

15.7 Room Reflection Cancellation

We have already noted that the Ambionic approach assumes that the sound wave propagation happens under free field conditions (chapter 15.1). This means that room reflections will disturb the restored Ambisonic soundfield. In hemispherical arrangements like the IEM CUBE (chapter 12.2), reflections from the floor are particularly critical. Several approaches have been described for room reflection cancellation in periphonic sound spatialization systems by the means of destructive interference. A technique described in [Son03, pp.86-104] can be efficiently implemented as a modification of the Ambisonic decoding process. The method bases on an offline recording of each loudspeaker's directional impulse response by the means of an Ambisonic recording technique (like the Soundfield microphone for first order representations), yielding a set of L impulse responses, each encoded to N Ambisonic channels. From these signals, it is possible to build a set of compensation signals which are used to create wavefronts destructively interfering with the room reflections.

The impulse response of the j-th loudspeaker can be modelled as

$$h_j\left(i\right) = \delta\left(t - t_0\right) + \sum_{k=1}^{\infty} a_k \cdot \delta_{j,k}\left(t - t_j\right)$$

The first term represents the direct wavefront of the loudspeaker, and the summation term describes the disturbing reflections. By capturing the loudspeakers' impulse responses with an Ambisonic recording technique, we reveive them in form of N Ambisonic encoded channels. From these signals, the term describing the direct wavefront is removed, since we want to exclude it from our cancellation efforts. The remaining signal is multiplied by -1, representing a 180 degree phase shift in order to achieve the required destructive interference. For each loudspeaker, the result of this operation is then convolved with the actual loudspeaker signal, yielding a set of Ambisonic encoded compensation signals $\vec{B}_{comp}$. Decoding these signals and adding them to our original Ambisonic decoding equation (eq. (23)) yields

$$\vec{p} = \mathbf{D} \cdot \vec{B} + \mathbf{D} \cdot \vec{B}_{comp}$$

This is equivalent to the following representation, which uses the $\otimes$ symbol to denote convolution

$$\boxed{\vec{p} = \mathbf{D} \cdot \left(\mathbf{F} \otimes \vec{B}\right)} \tag{28}$$

Here, $\mathbf{F}$ is a N-by-N matrix of filters

$$\mathbf{F} = \begin{pmatrix} \delta\left(t\right) + \sum\limits_{j=1}^{L} g_{00_j}^1 Y_{00_{comp}}^1 & \sum\limits_{j=1}^{L} g_{11_j}^1 Y_{00_{comp}}^1 & \cdots & \sum\limits_{j=1}^{L} g_{M0_j}^1 Y_{00_{comp}}^1 \\ \sum\limits_{j=1}^{L} g_{00_j}^1 Y_{11_{comp}}^1 & \delta\left(t\right) + \sum\limits_{j=1}^{L} g_{11_j}^1 Y_{11_{comp}}^1 & \cdots & \sum\limits_{j=1}^{L} g_{M0_j}^1 Y_{11_{comp}}^1 \\ \cdots & \cdots & \cdots & \cdots \\ \sum\limits_{j=1}^{L} g_{00_j}^1 Y_{M0_{comp}}^1 & \sum\limits_{j=1}^{L} g_{11_j}^1 Y_{M0_{comp}}^1 & \cdots & \delta\left(t\right) + \sum\limits_{j=1}^{L} g_{M0_j}^1 Y_{M0_{comp}}^1 \end{pmatrix}$$

where $Y_{mn_{comp}}^{\sigma}$ are the Ambisonic encoded compensation signals in $\vec{B}_{comp}$ and $g_{mn_j}^{\sigma}$ is the gain factor of the according Ambisonic channel feeding the j-th loudspeaker, as derived from the decoding matrix $\mathbf{D}$. The number of required filters is

N^2 and thus only depends on the number of Ambisonic channels but not on the number of loudspeakers, which is an important advantage of this method. For periphonic Ambisonic representations, we receive the required number of filters from equation (19) with $N^2 = (M+1)^4$. A more detailed derivation of the process can be found in [SH00a]. It has to be pointed out that this method only gives good results in the low-frequency area (up to about 500 Hz), due to the offline recording process, which might happen under very different acoustical circumstances than later soundfield reproduction (empty room vs. room with listeners, differences in temperature, etc.) [Son03, pp.89].

16 Sound Object Encoding (O-Format)

It has been noted in chapter 5 that the properties of a sound source regarding size, orientation in the soundfield and radiation characteristics are rarely taken into account in periphonic sound spatialization. Since their successfull reproduction can greatly increase the richness of the acoustical experience in a virtual environment, we will discuss an approach presented by Dylan Menzies [Men02] in this chapter, which is refered to as *O-format*. Menzies' original approach features direction-dependent encoding of a sound source's gain factor. David Malham has extended this approach to include *frequency* dependent radiation characteristics. He has also presented a method to encode the surface shape and size of a sound object [Mal01]. Although the O-format has been developed within an Ambisonic context - the same spherical harmonic functions are used in sound object encoding - the encoding/decoding process of an O-format sound source is carried out prior to its directional encoding, allowing this approach to be used also with other spatialization techniques. Menzies and Malham have described their approaches as first order representations, noting the possibility of higher order extension. We have generalized the O-format encoding and decoding equations for application at arbitrary orders and will present them in this chapter.

16.1 Frequency-Invariant Radiation Pattern

It has been noted in chapter 15 that any function on a sphere can be described by a series of the so-called spherical harmonic functions. Thus, the direction-dependent gain factor of a sound source[37] can also be represented this way, resulting in an O-format encoding process identical to the one of an Ambisonic soundfield. However, rather than converging in a central listening point, the signal now spreads outwards from a point-like sound source. Menzies speaks of "turning the [B-format] signal inside-out" [Men02] in this context, refering to the interesting approach of re-using an Ambisonic encoded soundfield as an O-format sound source.

The encoding of a mono sound source signal to an O-format representation is identical to the Ambisonic encoding process (eq.(21)), which means that an available Ambisonic encoder can simply be re-used for O-format encoding. The *sound object encoding equation* for encoding k audio signals s_i to one sound object is given as

$$\boxed{\vec{O} = \sum_{i=1}^{k} \vec{Y}(\vartheta_i, \varphi_i) \cdot s_i} \qquad (29)$$

[37] We call this the *frequency-invariant radiation pattern*, since it does *not* include directional dependencies of the signal's spectral content - an effect which can be captured or synthesized by techniques discussed in chapters 16.3 and 16.4.

The result of this operation is a vector $\vec{O}$ of length $N_O = (M_O + 1)^2$ (see eq. (19)) for the 3D case, where M_O represents the order at which the sound object is encoded. The signals in $\vec{O}$ are the spherical harmonic components of the wavefront emitted by the sound object into the direction ϑ, φ, assuming a coordinate system with axes in the same directions as the soundfield's coordinate system, but with the origin at the position of the sound object. Once encoded to the sound object domain, it is possible to apply the same transformations to an O-Format encoded signal as the ones that have been described for Ambisonic encoded soundfields in chapter 15.3 (rotations, mirroring, etc.). This happens in the same manner as described in equation (25), i.e. by multiplication of the encoded signal vector with a matrix representing the *sound object operation* $\mathbf{T}_O$:

$$\boxed{\vec{O}' = \mathbf{T}_O \cdot \vec{O}} \tag{30}$$

For example, the sound object can be *rotated* by the means of the rotation matrices described in chapter 15.3.1, which allows to specify the orientation of the object in the soundfield. Other operations like *mirroring* or *W-panning* of an O-format encoded sound object can be used to dynamically change its radiation pattern.

After the desired operations have been applied to the object, it is decoded back to a standard audio signal before being embedded into the soundfield by the means of a prefered spatialization technique (e.g. Vector Base Panning, Ambisonic, etc.). The signal emitted by a sound object into a specific direction ϑ, φ is given as the sum of the respective values of the spherical harmonic O-format components. The only direction ϑ, φ which is of interest for us is the direction of the sweet spot (as seen from the object), which depends on the position θ, ϕ of the object in the soundfield:[38] For example, if the sound object is located 'in front', e.g. at position $\theta = 0, \phi = 0$, the sweet spot is - from the object's point of view - located in the direction $\vartheta = 180°, \varphi = 0°$. This observation can be generalized, yielding that the relevant sound emission direction ϑ, φ of the sound object is always exactly the 'mirrored' direction of θ, ϕ. In this context, it is possible to exploit the symmetry of the spherical harmonic functions like in the case of mirroring soundfields (see chapter 15.3.3) by simply multiplying each spherical harmonic component (except the W channel, which is 1 everywhere) with a factor of -1 in order ro receive its value in the opposite direction. For an O-format encoded sound object at position θ, ϕ of the soundfield, the *sound object decoding equation*[39] is therefore given as

$$\boxed{s_O = \vec{D}_O\left(\vartheta, \varphi\right) \cdot \vec{O}'} \tag{31}$$

where $\vec{D}_O$ is the *sound object decoding vector* of lenght N_O:

$$\vec{D}_O = \vec{Y}\left(\vartheta, \varphi\right) = \left[+Y^1_{00}\left(\theta, \phi\right) \quad -Y^1_{11}\left(\theta, \phi\right) \quad -Y^{-1}_{11}\left(\theta, \phi\right) \quad ... \quad -Y^1_{M0}\left(\theta, \phi\right) \right]^T$$

The multiplication in equation (31) represents a dot product. Thus, the result is a scalar, representing the signal emitted by an object at position θ, ϕ into the direction of the central listening spot. The decoding order can also be chosen lower than N_O by choosing a shorter vector $\vec{M}$ and ignoring the additional channels of

[38] Alois Sontacchi has noted in a private discussion that the other directions of sound propagation are of interest for synthesizing room reflections.

[39] Our equation (31) is a higher-order generalization of equation (11) in [Men02]

the O-format.[40]

The decoded sound object signal s_O can be treated like a regular (non O-format encoded) sound source signal. For example, it can be encoded into an Ambisonic soundfield using equation (21). Since the entire O-format encoding/decoding process is carried out before spatializing the sound object, the order M_O at which the object is encoded is completely independent from the Ambisonic order M of the soundfield, allowing for a tradeoff between sound object and soundfield accuracy [Mal01, p.56]. The same observation yields that any other spatialization technique can be used for the directional encoding of a decoded O-format sound source [Men02].

It is useful to assume a situation where the relative orientation of a sound object to the listener remains constant as the object moves in the soundfield (e.g. the voice of a person constantly facing the listener as she/he walks around the listening spot). This can easily be achieved by a simplification of the sound object decoding matrix $\vec{D}_O$:

$$\vec{D}_O = \vec{Y}(0,0) = \left[Y_{00}^{1}(0,0) \quad Y_{11}^{1}(0,0) \quad Y_{11}^{-1}(0,0) \quad Y_{10}^{1}(0,0) \quad ... \quad Y_{M0}^{1}(0,0) \right]^{T}$$

16.2 Surface Shape and Object Size

So far, we have considered point-like sound objects. David Malham has suggested a techniqe for encoding the *surface shape* of a sound object. The signals emitted by various points of the object's surface will interfere in the listening spot, resulting in a filtering process of the signal picked up by the listener, which depends on the relative orientation of the surface shape to the listener. It is important to note that this effect is a result of the surface shape, not of frequency-variant radiation patterns (see chapter 16.4) - it also occurs if an identical signal is emitted homogeneously from all points on the object's surface. The effect can be described by the impulse response of a sound object, i.e. the signal measured at the listening spot when the object is sending out an impulse. Thus, to add surface shape information to an audio signal s, we have to find its impulse response h_O and convolve it with the s, so that

$$s_{shape} = h_O(\theta, \phi) \otimes s \tag{32}$$

where s_{shape} represents the surface-shape encoded sound signal. Note that s can either denote a plain audio signal or also a signal which has already been encoded and decoded so that it includes frequency-invariant radiation characteristics (chapter 16.1). In the latter case, the orientation of the frequency-invariant radiation pattern (encoded in the original O-format) and the orientation of the surface shape (encoded by the impulse response) can be varied independently - i.e. one can dynamically change the radiation pattern of an object without moving the object itself.

It is possible to model the surface shape as a combination of spherical harmonics, and then derive h_O from it. However, for reasons of efficiency Malham has

[40] However, this is only relevant as an analogy to Ambisonic soundfield encoding. In practice, the multiplication with $\vec{M}$ represents a low-cost operation, and unlike the maximum decoding order of an Ambisonic soundfield, which depends on the number of available speakers, there are no such inherent limitations in the case of sound object decoding.

suggested to use a set of spherical harmonic impulse response components [Mal01, p.55], which represent the far field responses of objects with shapes resembling those of the spherical harmonic functions (figure 16). For example, figures 20 and 21 show zeroth and first order impulse responses. Their length is determined by the size of the object, and the amplitude of each peak is determined by the size of the radiating area at the respective distance. Note that neither the $1/r$ law of sound pressure amplitude decay nor effects due to varying off-axis distances of the radiating points are applied yet. For this reason, Malham uses the term *non-distance weighted impulse response* (ndw response) in this context.

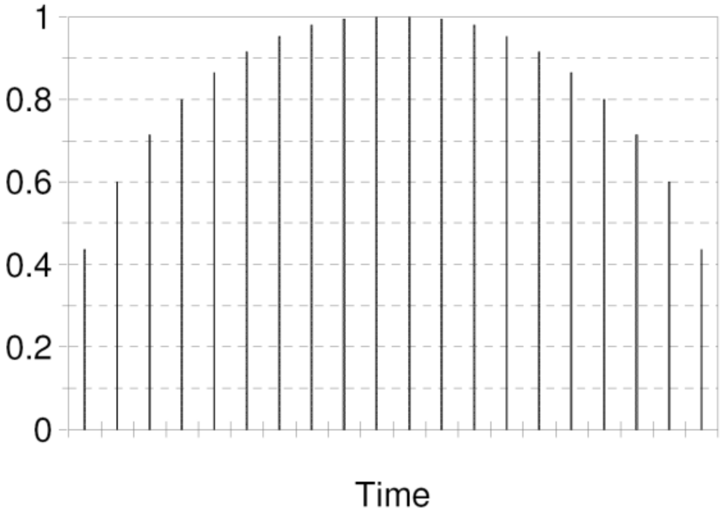


Figure 20: Zeroth order impulse response [Mal01, p.55]

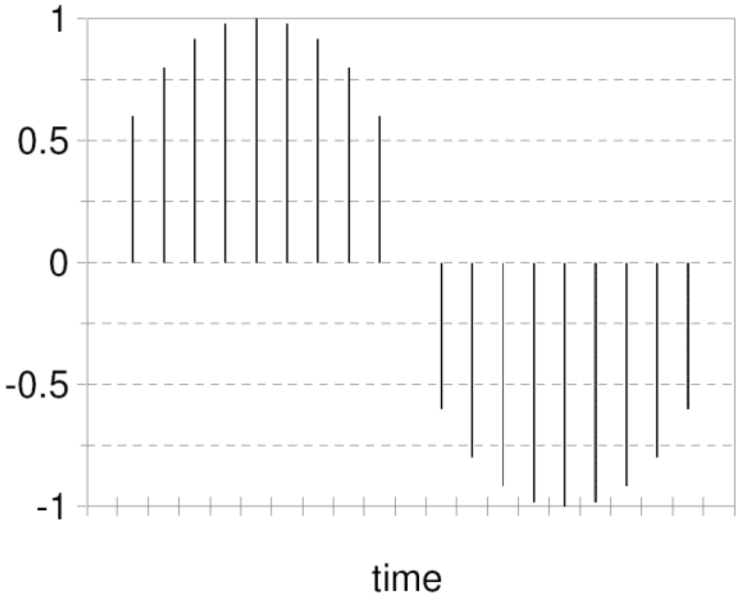


Figure 21: First order impulse response [Mal01, p.55]

The spherical harmonic decomposition of the surface shape is given as a matrix $\mathbf{O}_{shape}$ with rows representing the spherical harmonic components and columns representing the time samples of the ndw-responses. The usual transformation like rotation, etc. can be applied to the surface shape before the total ndw-response $h_{O,ndw}$ is derived as a weighted sum of its spherical harmonic components

$$h_{O,ndw} = \vec{D}_O\left(\vartheta, \varphi\right) \cdot \mathbf{O}_{shape}$$

where $\vec{D}_O$ denotes the decoding vector known from equation (31). The result $h_{O,ndw}$ represents the non-distance weighted impulse response of the object with the given relative orientation to the listener and is further modified in order to account for the effects due to on- and off-axis distances:

- For objects with significant extension on the axis connecting it and the listener, the amplitude decay along the axis has to be compensated. Although the peaks of the impulse response represent amplitudes ($1/r$ law) rather than intensities ($1/r^2$ law), Malham has suggested to weight each impulse in h_O with a factor $\left(\ T_S/T_C\right)^2$, where T_S is the time of appearance of the first component in the impulse response, and T_C that of the current component [Mal01, p.56].
- In the case of objects with significant off-axis extension, the extra delay due to the off-axis distance has to be compensated. This is particularly important for near field sources, for which the off-axis distance is relevant compared to the distance of the object to the listener. The compensation can be achieved by warping the time axis. [Mal01, p.56]

From the resulting impulse response h_O, the sound signal s_{shape} can be derived by convolution (eq. 32). s_{shape} includes the surface shape information of the sound object and can be spatialized using a prefered spatialization technique (e.g. Ambisonic, VBAP, etc.). David Malham has also described how the *sound object size* can be dynamically varied by changing the resampling rate of the spherical harmonic impulse response set, since changing the size of a sound object only affects the length, but not the general shape of the impulse responses [Mal01, p.56].

16.3 Natural Sound Objects

David Malham has presented two methods to determine the impulse response of a natural sound object:

- The surface shape of an object can be measured and used to derive its non-distance weighted impulse response (equivalent to h_O from equation (31)). The measurement can be done using an inwards facing microphone array around the object or laser scanning techniques. In order to be able to apply the typical sound object operations (rotation etc.) before decoding, the object has to be re-encoded into its spherical harmonic components by the means of a Fourier analysis. [Mal01, p.56-57]
- The impulse response of the object can also be measured directly. This has to be done in the far field in order to receive a non-distance weighted impulse response. The measurement has to be repeated over an appropriate grid in order to derive the spherical harmonic components of the impulse response. Here, the frequency-dependent radiation pattern of the sound source (chapter 16.4) is captured as well. [Mal01, p.57]

16.4 Frequency-Variant Radiation Pattern

We have already shown how the original O-format allows for encoding the spherical gain function of a point-like sound object (chapter 16.1). However, for most sources in natural sound sources, it is not only the loudness of the emitted signal that varies with direction, but also its spectral content. Robert Höldrich has noted in a private discussion that frequency-variant radiation characteristics could be encoded by application of a filter bank on the source signal, followed by separate sound object encoding of the resulting frequency subbands according to Menzies' original O-format proposal. The superposition of the resulting O-format representations then yields the encoded sound object. David Malham has presented a different approach: Since filters can be described by their impulse response, he has suggested to use a set of spherical harmonic impulse responses like the one used to describe the surface shape (chapter 16.2) of the object. A combination of both sets would be equivalent to directly measured impulse responses described in chapter 16.3. [Mal01, p.57]

17 Evaluation of Synthesized Soundfields

In chapter 6, different criteria for evaluating the localization in periphonic soundfields have been discussed, with a focus on monochromatic fields synthesized from spherical or plane wavefronts. In the last chapters, we have also discussed the mathematical descriptions of different sound spatialization algorithms. With this knowledge, we can now simulate the reconstruction of a 'natural' soundfield through a spatialization algorithm of our choice. By applying our localization criteria to both, the original and the resynthesized soundfield, and comparing the differences, we have now a tool to evaluate the quality of a periphonic sound spatialization system. In chapter 21 we will also present a Matlab library as an according practical implementation.

17.1 Integrated Wavefront Error (D-Error)

Bamford has introduced the *integrated wavefront error* or *D-error* as a means to compare the reconstruction of a soundfield with its original [Bam95, p.19]. Although he has developed this criterion to evaluate Ambisonic systems, the D-error can also be applied to soundfields reconstructed by other spatialization techniques. The D-error, which originally was applied to horizontal-only systems, bases on the integration of the local difference between the original (reference) pressure field S_{ref} and the reconstructed pressure field S_{synth} over the circumference of a circle. By integrating over the surface of a sphere, it can be generalized to periphonic situations [Son03, p.45]. For those cases, the D-error is defined as

$$D^{3D} = \frac{1}{\left|4\pi P_{ref}^2\right|} \int\limits_{\theta=0}^{2\pi} \int\limits_{\phi=-\pi/2}^{+\pi/2} \left|p_{ref}\left(\vec{r}\right) - p_{synth}\left(\vec{r}\right)\right| \cdot d\theta \cdot d\phi$$

Due to the normalization term in front of the integral, the error becomes 1 for an absent field S_{synth}. Sontacchi has suggested to consider the D-error as a function of $k\vec{r}$, i.e. as a function of frequency[41] and r (distance from the sweet spot) for systems with large listening areas [Son03, p.45].

[41] $k = 2\pi f/c$ is the wave number.

17.2 Sound Pressure Errors

The *squared sound pressure error* has been used as a descriptor of soundfield reconstruction by [Pol00] and [Son03]. It is defined as the squared difference of the original and the reconstructed sound pressure field.

$$\Delta p^2 = \left[p_{ref}\left(\vec{r}\right) - p_{synth}\left(\vec{r}\right)\right]^2$$

The *sound pressure amplitude error* is defined as the absolute value of the difference between the original and the reconstructed complex sound pressure field

$$|\Delta p| = \left|p_{ref}\left(\vec{r}\right) - p_{synth}\left(\vec{r}\right)\right|$$

17.3 Direction Deviation

Sontacchi has used the *direction deviation* of two vector fields describing the perceived direction in a soundfield (e.g. the real part of the complex velocity vector or the active velocity; see chapter 6) to evaluate the reconstruction quality of horizontal-only sound reproduction systems [Son03]. The direction deviation is a scalar field which represents the angle between the directions of the original and the reconstructed soundfield. A 3D version of the direction deviation using the active velocity as a direction indicator can be specified as

$$\delta\left(\vec{r}\right) = \arccos \frac{\vec{v}_a^{ref}\left(\vec{r}\right) \cdot \vec{v}_a^{synth}\left(\vec{r}\right)}{\left|\vec{v}_a^{ref}\left(\vec{r}\right)\right| \cdot \left|\vec{v}_a^{synth}\left(\vec{r}\right)\right|}$$

where the term in the numerator denotes a dot product. It remains to evaluate how the direction deviation coresponds with human perception.

18 Loudspeaker Layout Design

The process of designing a suitable loudspeaker layout for a periphonic sound spatialization system is not a trivial task. Often, like in the case of the AlloSphere (chapter 19.1), the system has to support multiple spatialization algorithms rather than restricting itself to a single technique. However, different spatialization algorithms have different requirements regarding the loudspeaker layout, which need to be considered in the course of the design process. Other criteria, like the spatial resolution of the human ear (see chapter I) and architectural restrictions play an important role in the design process. In this chapter, we will first adress these criteria and then present different strategies for finding suitable layouts. Although these are all based on sphere approximations, we will show how they can be adopted to situations in which loudspeaker mounting on a spherical surface enclosing the listening area is not possible. They can also be used to derive hemi-spherical layouts, which are often used in periphonic sound spatialization (see [Dan00, p.193] and chapter 12.2). Finally, we will present a new loudspeaker layout design strategy, which is a hybrid approach of the methods discussed.

18.1 Design Criteria

18.1.1 Vector Base Panning Layouts

For implementation of a Vector Base Panning system, a triangulation of the surface defined by the loudspeaker layout has to be performed (see chapter 13). To achieve a homogeneous localization quality, the shape and size of the different triangles should not differ from each other too much. Rather than first defining a loudspeaker layout which is then triangulated, we might find suitable layouts defined as structures of vertices (loudspeaker positions) and facets (loudspeaker triangles, or also squares, which can always be interpreted as two triangles) from the theory of *geodesic spheres* (chapter 18.3), keeping the differences between the triangles at a minimum. Best results are given if the loudspeakers are placed at equal distances from the sweet spot. The number of loudspeakers is only determined by the available hardware (loudspeakers, CPU, bandwidth, disk space).

18.1.2 Higher Order Ambisonic Layouts

In Higher Order Ambisonic systems, the regularity of the loudspeaker layout plays a more important role than in the case of Vector Base Panning. Although the homogeneity of localization in a VBP system will also benefit from a regular layout, irregularities affect it only locally (for sources in the according directions). Since in Ambisonic systems generally all loudspeakers contribute to the reproduction of a sound source, irregularities in the layout globally affect the soundfield reconstruction. Daniel has given definitions regarding regularity of a loudspeaker layout in the Ambisonic sense in [Dan00]. A loudspeaker layout is refered to as *regular*, if it preserves orthonormality. This can be examined using the N3D encoding to check whether [Dan00, p.176]

$$\frac{1}{N} \cdot \mathbf{C} \cdot \mathbf{C}^T = I_N \tag{33}$$

holds for the given loudspeaker layout. N is the number of Ambisonic channels, $\mathbf{C}^T$ is the transposed Ambisonic re-encoding matrix from chapter 15.1, and I_N is the N-by-N unity matrix. Equation (33) depends on the Ambisonic order M, but not on the order of rows (loudspeakers) and columns (Ambisonic channels) in $\mathbf{C}$, nor on the orientation of the loudspeaker layout in the coordinate system. Also note that layouts which are regular for an order M are always regular for all lower orders as well, and that regularity of a layout implies that the $L \geq N$ criterion (eq. 24) is fullfilled. Daniel further defines the class of *semi-regular* layouts, which preserve orthogonality [Dan00, p.176].

$$\frac{1}{N} \cdot \mathbf{C} \cdot \mathbf{C}^T = \mathrm{Diag}\left(\mu_N\right)$$

where $\mathrm{Diag}\left(\mu_N\right)$ is a diagonal matrix with $\mu_N = [\mu_1\ \mu_2\ ...\ \mu_N]^T$. Again, this relation only holds for the N3D encoding convention.

Sontacchi has suggested to use the *condition number* of the re-encoding matrix $\mathbf{C}$ to evaluate the regularity of an Ambisonic loudspeaker layout [Son03, p.39]. The condition number is defined as the ratio of the largest singular value of a matrix to the smallest. It is used in mathematics to evaluate the accuracy of matrix inversion, with a value near 1 indicating a well-conditioned matrix. In Higher Order Ambisonic systems, we can use the condition number as a criterion regarding the quality of a decoding matrix $\mathbf{D}$ derived as the pseudoinverse of $\mathbf{C}$ (chapter 15.4). The condition number offers the advantage of returning a representative value rather than a mere classification of the loudspeaker layout. However, it depends not only on the Ambisonic order, but also on the sorting of the rows in columns in $\mathbf{C}$, which represent the Ambisonic channels and loudspeakers. Thus, it should be used as an indicator rather than as a definite loudspeaker layout evaluation criterion. Also, the ratio of the singular values might hide some irregularities. While for horizontal-only systems, regularity in the Ambisonic sense can be simply achieved by loudspeakers arranged in a regular polygon (i.e. at equal angles along a circle), the number of regular layouts reduces dramatically in the 3D case. Daniel has noted that only five layouts fullfill the regularity criterion defined in equation (33), which are known as the *Platonic solids*, discussed in chapter 18.2.1. However, these provide 20 vertices at the most and regularity in the Ambisonic sense for a maximum order of $M = 2$. Daniel has thus suggested to look for *quasi-regular* layouts for higher orders using the method of geodesic spheres. It will be shown in chapters 18.3 and 22.1 that this technique is indeed suitable for the design of well-conditioned Higher Order Ambisonic loudspeaker layouts.

18.1.3 Homogeneity vs. Psychoacoustics

While regularity in the loudspeaker distribution is required to make for a homogeneous localization in periphonic sound spatialization systems, we can also optimize such a system according to psychoacoustical criteria. As we have seen in chapter I, the spatial resolution of the human ear greatly varies with the direction of an incoming sound event. This suggests to define a varying loudspeaker density which increases in areas of good auditory resolution (front direction) and decreases in areas of poor localization (rear direction, elevated sources). For periphonic loudspeaker systems, it is not obvious how to achieve this. We will discuss an according method in chapter 18.6. But : Ambisonics, VBAP not so sensitive. However, it has to be kept in mind that psychoacoustical layout optimization can be critical regarding Ambisonic soundfield reconstruction, since it can negatively affect the regularity of a loudspeaker layout (chapter 18.1.2). Vector Base Panning (chapter 13) systems, where the active loudspeakers are always located in a small angular region, are less affected.

18.1.4 Horizontal Plane

A number of reasons speak for paying special attention to the horizontal plane in a periphonic sound spatialization system. We have already noted in the last chapter that areas of good auditory resolution (which the horizontal plane is) can be equipped with more loudspeakers for psychoacoustical layout optimization. Many standard multichannel formats (see chapter II) are restricted to the horizontal plane, as is (2D) Wave Field Synthesis. Even for a fully periphonic spatialization system it makes sense to provide possibilities for an application of these techniques. We have also noted that 3D Vector Base Panning systems give best results in the horizontal plane, since they base on the reproduction of interaural differences (chapter 13.4). All these reasons speak for introducing a loudspeaker ring in the horizontal plane, which we refer to as the *equator* of a loudspeaker layout. For example, the dodecahedron does not provide an equator, whereas the octahedron does (see figure 22). Also, hemispherical layouts can be more easily derived from spherical layouts with loudspeakers in the horizontal plane. An example for this is the IEM CUBE (chapter 12.2).

18.1.5 Architecture

The real-world design of a virtual environment is usually not exclusively devoted to the loudspeaker positions, leaving us in the uncomfortable situation of not being able to position our speakers where we would like to see them. Reasons for this might be given by the architecture of the reproduction space and/or problems of loudspeaker accessibility and maintainance. The resulting restrictions can involve deviations from the prefered angle as well as the radius of loudspeakers. We will speak of *forbidden areas* where loudspeakers cannot be mounted, and consider a *loudspeaker radius function* to define the radius of a loudspeaker as a function of its direction in the layout.

18.2 Polyhedra

Generally, it is possible to approximate a spherical surface by the means of various polyhedra, which can be classified according to the properties of their vertices, faces, and edges [Wike]

- A polyhedron is *convex* if the line segment joining any two points of the polyhedron is contained in the polyhedron or its interior.
- A polyhedron is *vertex-uniform* if all vertices are the same, in the sense that for any two vertices there exists a symmetry of the polyhedron mapping the first isometrically onto the second.
- A polyhedron is *edge-uniform* if this symmetry is also given for its edges, and *face-uniform* if given for its faces.
- A polyhedron is *regular* if it is vertex-uniform, edge-uniform and face-uniform. This implies that every face is a regular polygon and all faces have the same shape.
- A polyhedron is *quasi-regular* if it is vertex-uniform and edge-uniform, and every face is a regular polygon.
- A polyhedron is *semi-regular* if it is vertex-uniform and every face is a regular polygon. The convex ones consist of the prisms and antiprisms and the *Archimedean solids* (chapter 18.2.2).
- A polyhedron is *uniform* if it is vertex-uniform and every face is a regular polygon, i.e. it is regular, quasi-regular, or semi-regular, but not necessarily convex.

Note that the definitions regarding regularity and semi-regularity do *not* match the according definitions provided by Daniel to evaluate the regularity of an Ambisonic loudspeaker system (see chapter). For example, the *dodecahedron* (chapter 18.2.1) is regular according to the above definition, but it is not regular in the Ambisonic sense for an Ambisonic order of $M = 3$, although it does provides enough loudspeakers for a third order system according to the $L \geq N$ criterion (eq. (24)).

An interesting property of polyhedra is the *Euler characteristic*, which relates the number of edges E, vertices V, and faces F of a simply connected polyhedron:

$$V - E + F = 2$$

In this chapter, we will present different polyhedra with interesting properties regarding loudspeaker distribution in periphonic sound spatialization systems. Since we are primarily interested in playouts approximating a spherical surface, we will only consider convex polyhedra, starting with the five regular convex polyhedra, which are also called the *Platonic solids*, and moving on to some convex semi-regular (the *Archimedian* solids) and face-regular polyhedra (the *Johnson solids*).

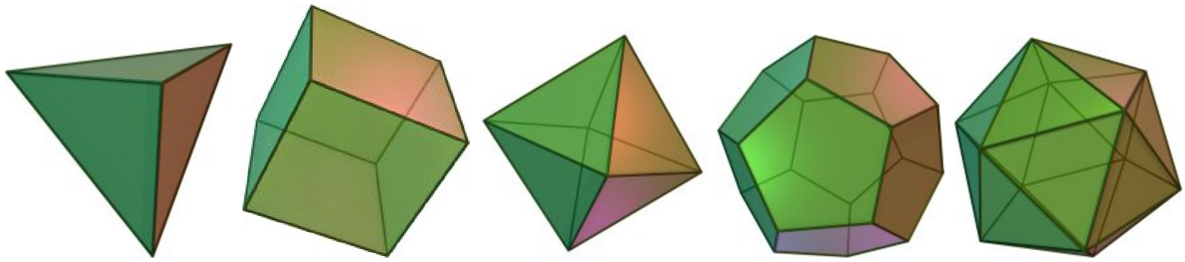


Figure 22: The Platonic solids: tetrahedron, hexahedron, octahedron, dodecahedron, icosahedron [Wikd]

18.2.1 Platonic Solids

There are only nine polyhedra which are regular by the definition given above. The five convex ones are also known as the *Platonic solids.* These are the *tetrahedron*, the *hexahedron* (i.e. the cube), the *octahedron*, the *dodecahedron*, and the *icosahedron.* They are shown in figure 22, and their properties are summarized in table 18.2.1.

Polyhedron	Vertices	Facets	Facet Shape	Equator
tetrahedron	4	4	triangular	no
hexahedron (cube)	8	6	square	no
octahedron	6	8	triangular	yes
dodecahedron	20	12	pentagonal	no
icosahedron	12	20	triangular	yes

Table 2: The five Platonic solids

The tetrahedron, the hexahedron, and the cube are regular in the Ambisonic sense (chapter 18.1.2 only for first order. The icosahedron is regular also for second order [Dan00, p.177]. Note that although the dodecahedron provides enough loudspeakers to fullfill the $L \geq N$ criterion for third order (see chapter 15.1.2), it is only Ambisonic regular for first and second order.

Obviously, the Platonic solids offer only a very limited amount of vertices with a maximum of 20 vertices in the case of the dodecahedron. For a periphonic sound-spatialization system in a large-scale virtual environment, this does not satisfy our requirements. Also, only the octahedron and the icosahedron provide a ring of loudspeakers in the horizontal plane. We will discuss possible extensions of these arrangements by the method of *geodesic spheres* in chapter 18.3.

18.2.2 Archimedian and Catalan Solids

By excluding the infinite set of prisms and antiprisms - which are no sphere approximations - from the class of convex semi-regular polyhedra, we receive the 13 *Archimedian solids* [Wika]. Five of them are truncated versions of the Platonic solids.[42] Probably the most famous Archimedian solid is the *truncated icosahedron*, better known as the *Bucky ball* and as the shape of a *soccer ball* (see figure 23). Two other Archimedian solids are convex quasi-regular polyhedra, i.e. they have

[42] Truncation of a polyhedron refers to the process of 'chopping off' its corners.

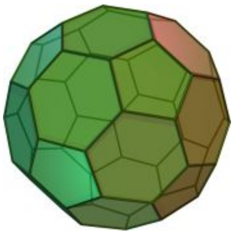


Figure 23: The truncated icosahedron (or Bucky ball) [Wika]

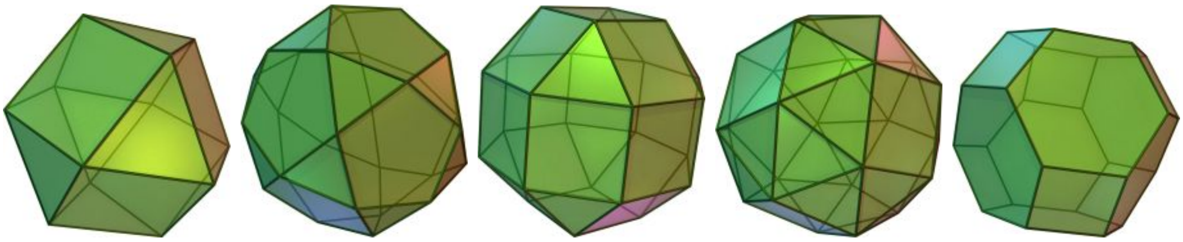


Figure 24: Some Archimedian solids: cuboctahedron, icosidodecahedron, rhombicuboctahedron, snub hexahedron, truncated octahedron[Wika]

the additional property of being edge-uniform. These are the *cuboctahedron*, consisting of triangles and squares, and the *icosidodecahedron*, built from triangles and pentagons. They are shown with three other Archimedian solids in figure 24. The remaining Archimedian solids differ from each other too much regarding the shape and size of their faces to seem interesting as periphonic loudspeaker layouts.

The duals of the Archimidean solids are the *Catalan solids*, also known as *canonical polyhedra* [Wikb]. We show four of them in figure 25, which seem to have interesting properties regarding application in periphonic loudspeaker layouts.

18.2.3 Johnson Solids

In 1966, Norman Johnson published a list of 92 convex, non-uniform, and face-regular polyhedra, which are generally refered to as the *Johnson solids*. In 1969, Victor Zalgaller proved that Johnson's list was complete [Wikc]. Although most of the Johnson solids are no sphere approximations, some of them have interesting properties, like the *elongated square gyrobicupola*. Others might serve as a good basis for the design of hemispherical layouts, like the *gyroelongated pentagonal cupola* (figure 26).

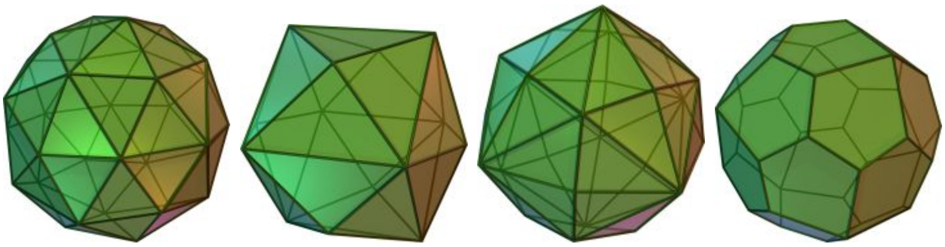


Figure 25: Some Catalan solids: The pentakis dodecahedron, tetrakis hexahedron, disdyakis dodecahedron, and pentagonal icositetrahedron [Wikb]

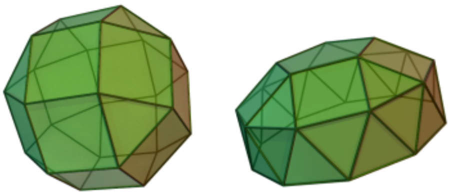


Figure 26: Two Johnson solids: The elongated square gyrobicupola and the gyroelongated pentagonal cupola [Wikc]

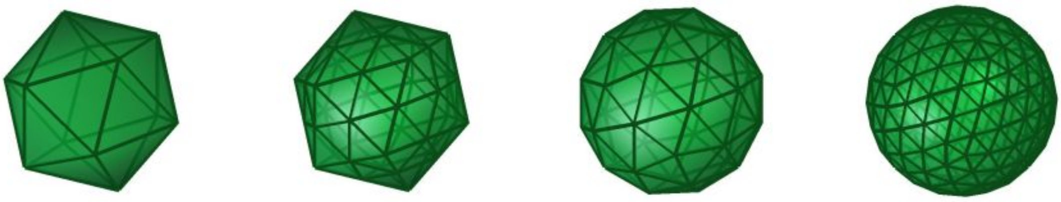


Figure 27: Building a geodesic sphere from an icosahedron

18.3 Geodesic Spheres

In chapter 18.2.1 we have seen, that mathematical regularity of a polyhedron unfortunately translates to a rather low number of vertices. Arrangements with more vertices can be found among the semi-regular and other classes of polyhedra. However, the number of loudspeakers in these arrangements is still pre-determined by the solid itself. A way to introduce a higher number of vertices (i.e. loudspeakers) while retaining the advantages regarding the regularity of the Platonic solids is given by the method of *geodesic spheres*. This method is used in architecture to distribute stress on 'geodesics', i.e. large circles around a spherical structure. Usually, it is described as the tessellation of the faces of a Platonic solid. This process creates new vertices which are then 'pushed out' to the radius of the polyhedron's circumscribed sphere. The faces are re-arranged as well in order to include the new vertices. This process can be repeated in an iterative way. Figure 27 shows the process of building a geodesic sphere from an icosahedron. The first picture shows the original icosahedron, being triangulated in the second picture. The third picture shows the triangulated icosahedron with the new vertices pushed out to its circumsphere. Repeated application of the process of pictures 1 and 2 on the polyhedron from picture 3 gives the resulting geodesic sphere in picture 4. Note that also for the maximum Ambisonic order which can be reproduced on this layout according to the $L \geq N$ criterion (eq. 24), the condition number of its re-encoding matrix (chapter 18.1.2) remains remarkably low with $cond\,(\mathbf{C}) = 1.74$.

Since we aim at a choice of the total number of vertices (i.e. loudspeakers) as arbitrary as possible, we will generalize the concept of geodesic spheres, so it can be applied to other structures than the Platonic solids as well. For this purpose, we can define the following *tessellation rules* for differently shaped faces of a polyhedron:

- Triangles can either be midpoint-triangulated or triangulated at an arbitrary 'frequency'.
- Rectangles can be midpoint-triangulated or rectangulated at an arbitrary 'frequency'

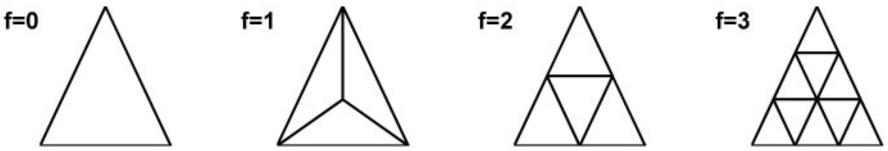


Figure 28: A triangle, its midpoint-triangulation, and two triangulations at different frequencies

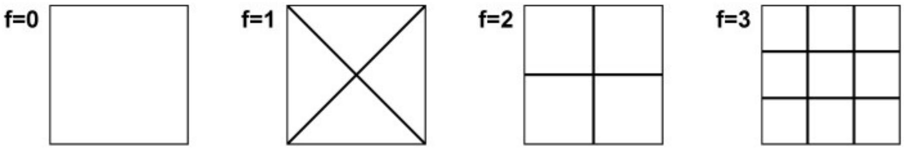


Figure 29: A rectangle, its midpoint-triangulation, and two rectangulations at different frequencies

- Polygons with more than four vertices (pentagons, hexagons, etc.) can only be midpoint-triangulated

We use the term *midpoint-triangulation* to describe the process of adding a new vertex in the center of an arbitrary polygon and connecting it to each of the existing vertices in the polygon, resulting in a number of new facets which is equal to the number of vertices in the original polygon. Midpoint-triangulations of a triangle, a rectangle, and a pentagon are shown in the second picture of figures 28, 29, and 30. Another tessellation strategy bases on the subdivision of a polygon's edges, resulting in a *triangulation* of triangles or a *rectangulation* of rectangles. Examples are shown in the last two pictures of figures 28 and 28. The *tessellation frequencies* $f > 2$ refers to the number of subsections in one edge. Additionally, we use $f = 0$ to denote the original polygon, and $f = 1$ to denote its midpoint-triangulation.

The presented tessellation strategies can be also applied to non-regular polygons, which allows us to build geodesic structures from non-uniform polyhedra as well. Another source of flexibility in the process of designing a geodesic sphere is given through independently specifying the tessellation behaviour of each face shape in each iteration. For example we could consider a polyhedron built from triangles, squares, and pentagons: In a first step, we decide to midpoint-triangulate every triangle, rectangulate every square at a frequency $f = 2$, and not subdivide the pentagons at all. In a second step we do not tessellate the triangles and midpoint-triangulate the squares and pentagons, leaving us with a structure consisting of triangles exclusively.

However, increased flexibility is traded off for growing irregularities in the re-

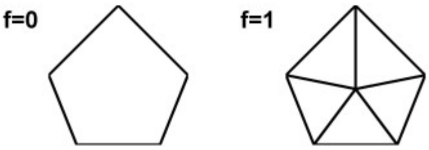


Figure 30: A pentagon and its midpoint-triangulation

sulting layout. The amount of irregularity is determined by the choice of the initial polyhedron[43], and the frequencies of the applied tessellations. The process of 'pushing out' the new vertices to the circumsphere of the polyhedron results in a variation regarding the length of the polyhedron's edges (which again affects the size of the faces). Since the number of different lengths is given by the tessellation frequency[44], high frequencies and mixed frequencies are to be avoided in order to keep the introduced irregularity at a minimum.

We have noted a special interest in periphonic layouts with loudspeaker equators (chapter 18.1.4). It is interesting to note that the method of geodesic spheres can affect this property of an arrangement in a positive way: For instance, any tessellation of the tetrahedron, icosahedron, or hexahedron at an even frequency ($f = 2$, $f = 4$, etc.) will result in a structure with a ring of vertices in the horizontal plane. As an example, consider figure 27.

Daniel has also mentioned two interesting examples of geodesic spheres built from Platonic solids [Dan00, p.177]: The compound of an icosahedron and a dodecahedron can also be interpreted as a $f = 2$ triangulation of either of the two. The same holds for the compound of octahedron and cube.

18.4 Minimal Energy Configurations

We have seen in the last chapter that although the method of geodesic spheres can be somewhat generalized towards a more arbitrary choice of the number of loudspeakers in a periphonic layout, this freedom has to be traded off for the regularity of the resulting layout. To overcome this disadvantage, it is possible to employ a different strategy from the field of physics. The resulting sphere approximations are refered to as *minimal energy configurations*. The process is initialized by randomly distributing an arbitrary number of electrons on the surface of a sphere. The electrons have only two degrees of freedom, i.e. they may only move on the sphere. Due to the repulsion forces the electrons exercise on each other, they will distribute themselves over the sphere until they find a natural equilibrum of minimal potential energy.

In nature, the repulsion forces among two electrons are proportional to their squared distance. However, this factor can be chosen differently in a computer simulation of the process, which can be easily implemented as a nested `for` loop, in which the forces of each electron onto each other electron are calculated, and the results are superponed. After all electrons have been moved to their new positions, they are brought back to the radius of the sphere while retaining their directions. The entire procedure can be applied ieratively until the configuration fullfills the specified requirements regarding homogeneity. The advantage of this method is complete freedom regarding the number of electrons/loudspeakers. However, minimal energy configurations usually lack any symmetry, also regarding the horizontal plane, i.e. they do not provide an 'equator' of loudspeakers (see chapter 18.1.4). Figure 31 demonstrates the process of electrons distributing on a sphere. The re-encoding matrix of the resulting layout after 200 iterations features a condi-

[43] Note that even the Platonic solids are of varying quality as initial layouts for geodesic spheres. For example, a tetrahedron generally produces greater irregularities than an icosahedron [Bou].

[44] For example $f = 2$ tessellation of a Platonic solid results in two different edge lengths, whereas this number increases to three in the case of $f = 3$ tessellation.

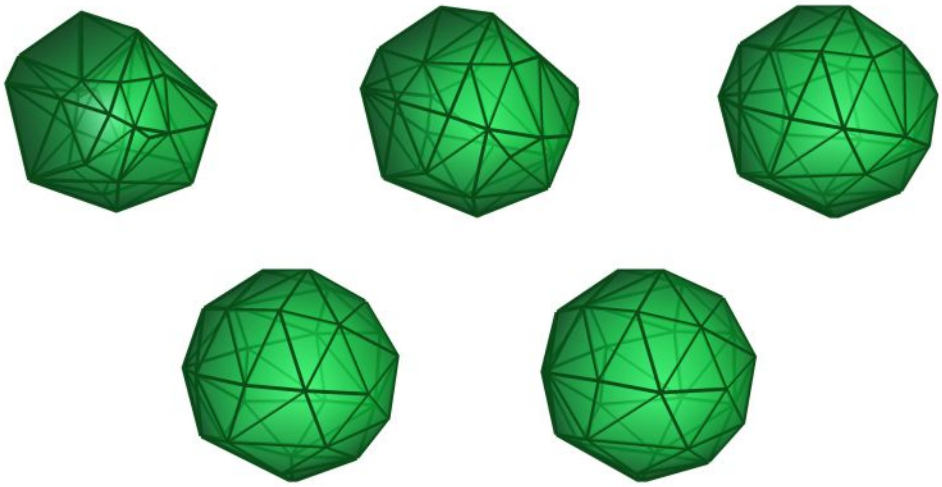


Figure 31: The result of a minimal energy configuration algorithm with 30 loudspeakers after 0, 10, 30 , 80, and 200 iterations

tion number (chapter 18.1.2) as low as $cond(\mathbf{C}) = 1.94$ for fourth order Ambisonic soundfield reproduction, which is the maximum order of the layout according to the $L \geq N$ criterion (eq. 24).

18.5 Loudspeaker Array Calibration

We have discussed possible restrictions regarding the loudspeaker positions due to the architectural environment of a periphonic sound spatialization system (chapter 18.1.5. These can also result in deviations of the loudspeaker radii. Generally, the actual radius will vary as a function of direction. Within reasonable limits, it is possible to compensate for such irregularities by the means of calibrating the gains and delays of the loudspeakers. Assuming the j-th loudspeaker in our layout, its gain factor g_j and delay T_j can be calculated as:

$$g_j = \frac{R_j}{R_{max}}$$

$$T_j = \frac{R_{max} - R_j}{c}$$

where R_{max} is the radius of the loudspeaker at the greatest distance, and c is the speed of sound. The first equation is derived from the 1/r law, resulting in lower gains for closer loudspeakers. The second equation compensates for the finite speed of sound by delaying the signals coming from closer loudspeakers. The gain and delay factors are normalized to the speaker with the maximum radius, i.e. the gain of that speaker is one and its delay is zero. By the means of layout calibration, the gain and phase factors of a wavefront can be restored such that they resemble those of a sound source moving at a distance $R_m ax$. However, the curvature of the wavefront cannot be restored, which causes problems when NFC-HOA distance coding is applied (chapter 15.6). If the differences of the loudspeaker radii are within reasonable limits, the average loudspeaker distance can be used in the NFC-HOA encoding process, which is given as

$$\overline{R} = \frac{\sum_{j=1}^{L} R_j}{L}$$

where L is the number of loudspeakers in the layout.

18.6 Towards a Hybrid Loudspeaker Layout Design Strategy

So far, we have focused our efforts on different methods for homogeneous distribution of loudspeakers over the surface of a sphere. We have seen how to use more exotic polyhedra or geodesic extensions in order to trade some of this homogeneity off for more freedom regarding the number of loudspeakers in a layout. We have also discussed what to expect from these methods regarding the horizontal plane, and how to compensate for slight irregularities of the loudspeaker radii. However, none of the approaches mentioned so far offers the possibility of varying loudspeaker densities for psychoacoustical layout optimization (see chapter 18.1.3). Ideally, we would like to be able to specify a spherical *loudspeaker density function*, which resembles the spatial resolution of the human ear as a function of direction. The task is to define an intuitive method which can be applied to arbitrary layouts, rather than empirically adopting a given arrangement.

It is suggested to use the *electron charge in the minimal energy configuration algorithm as a handle to the loudspeaker density.* Defining the charge of an electron as a function of its position on the sphere[45] will result in varying repulsion forces among the electrons: Higher forces will occur in areas where the charge of electrons is high, resulting in a low loudspeaker density, whereas more loudspeakers will be found in low-charge areas with weak repulsion. The spherical electron charge function can be derived by simple inversion of the loudspeaker density function. The definition of a loudspeaker density function can be significantly simplified using superpositions of weighted spherical harmonic functions, which are also used in the Higher Order Ambisonic spatialization technique (chapter 15) and sound object encoding (chapter 16).

Note that if the electron charge is a function of direction, the initial positions of the electrons cannot be chosen randomly any more: for example, in a situation with all electrons being initially placed in areas with very high loudspeaker density, the electrons would remain in those areas forever. It is thus important to use an initial layout in which the electrons are already to some degree homogeneously distributed. This results in a separation of the loudspeaker layout design process into two subsequent parts dedicated to different tasks: first, an initial layout is designed according to the homogeneity criterion, which is then psychoacoustically refined in the second stage (see figure 32). The initial layout can be generated by the means of any method discussed so far, i.e. as a polyhedron, its geodesic extension, or a minimal energy configuration (with constant loudspeaker density). However, we suggest to use a polyhedron-based approach in order to integrate the advantages of those layouts (symmetry, loudspeaker equator). It can be seen from table 3, that the advantages and disadvantages of minimal energy vs. polyhedron-based approaches contemplate each other very well. Geodesic extension can be used to keep the restrictions regarding the number of loudspeakers at a minimum.

Due to the use of a minimal energy configuration stage for psychoacoustical layout refinement, we have to be prepared to give up any symmetries in the initial

[45] A concept unknown to nature, but very convenient for our purposes.

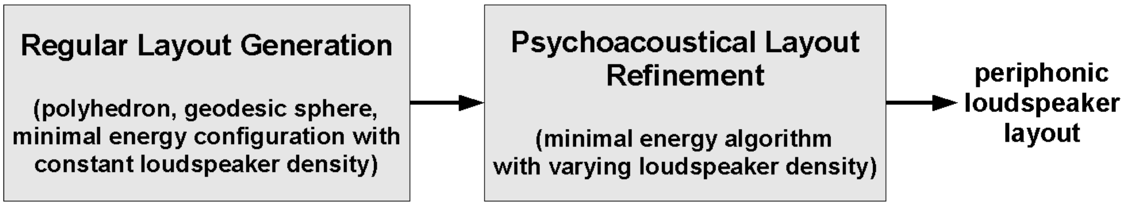

Regular Layout Generation
(polyhedron, geodesic sphere, minimal energy configuration with constant loudspeaker density)
Psychoacoustical Layout Refinement
(minimal energy algorithm with varying loudspeaker density)
periphonic loudspeaker layout

Figure 32: Loudspeaker layout design strategy

Criterion	**Polyhedra, Geodesic Spheres**	**Minimal Energy Configs**
number of loudspeakers	limited, quality tradeoff	arbitrary
loudspeaker density	no obvious handle	handle via electron charge
equator	easy to achieve	generally not given
symmetry	yes	generally not given
forced positions	not obvious to achieve	via 'locked' speakers
forbidden areas	not controllable	[via low speaker density]

Table 3: Polyhedron-based vs. minimal energy strategies

layout, including the horizontal plane, i.e. a loudspeaker equator in the original layout. Enforcing specific loudspeaker positions (*forced positions*) - be it on the equator or elsewhere - can be achieved by 'locking' certain loudspeakers in the refinement process: A locked loudspeaker can be simulated as an electron which exercises repulsion forces on the other electrons, but is immune to the forces exercised on itself and therefore remains in its initial position.

Deviations of the loudspeaker radii due to architectural circumstances (see chapter 18.1.5) can be considered in the psychoacoustical refinement loop by including a *loudspeaker radius function* which defines the actual radius of a loudspeaker as a function of the direction in which it is mounted. Rather than mapping the loudspeakers to the surface of a sphere after each iteration (as done in the 'regular' minimal energy algorithm), they are mapped to the surface specified by this function. Alternatively, it is also possible to first assume a spherical distribution in the refinement process and map the loudspeakers to their respective radii afterwards. Like a loudspeaker density function, the radius function can be defined as a superposition of weighted spherical harmonic functions. The required gain and delay factors for layout calibration (see chapter 18.5) can be calculated as a side product of the mapping procedure.

We have also discussed the possibility of areas which do not allow for mounting loudspeakers (*forbidden areas*, chapter 18.1.5). Loudspeakers in those areas can to some degree be avoided by assigning low values of loudspeaker density to forbidden areas in the refinement stage. If the values are chosen too low, however, this will result in an unstable behaviour of the configuration during the refinement stage. A more straightforward approach is to manually move loudspeakers which have ended up in forbidden areas after the refinement process. If the total number of loudspeakers is flexible, they can also be deleted. Since this affects the properties regarding the homogeneity of the layout, additional minimal energy refinement might be required after such operations. It is expected that the best results will be achieved through a combination of a suitable loudspeaker density function and manually moving or deleting loudspeakers, depending on the circumstances regarding the number and size of the forbidden areas and the required number of loudspeakers. Using the ideas proposed in this chapter, an *extended loudspeaker layout design strategy* can be defined, which is described in figure 33.

Note that the introduction of a non-constant loudspeaker density disturbs the convergence of the minimal energy algorithm towards a natural equilibrum: the result of the such extended algorithm actually does not represent a minimal energy configuration in the physical sense any more. With increasing inhomogeneity of the loudspeaker density, stability of the configuration becomes harder to achieve, i.e. also after a high number of iterations, the configuration is still subject to significant changes. The introduction of 'locked' loudspeakers radius functions has the same effect. Thus, the freedom of psychoacoustical layout optimization and the consideration of architectural circumstances has to some degree be traded off for ease of implementation. Nevertheless, we believe that the presented method represents a useful approach to the design of periphonic loudspeaker layouts, which will be demonstrated at the example of the *AlloSphere* in chapter 22.

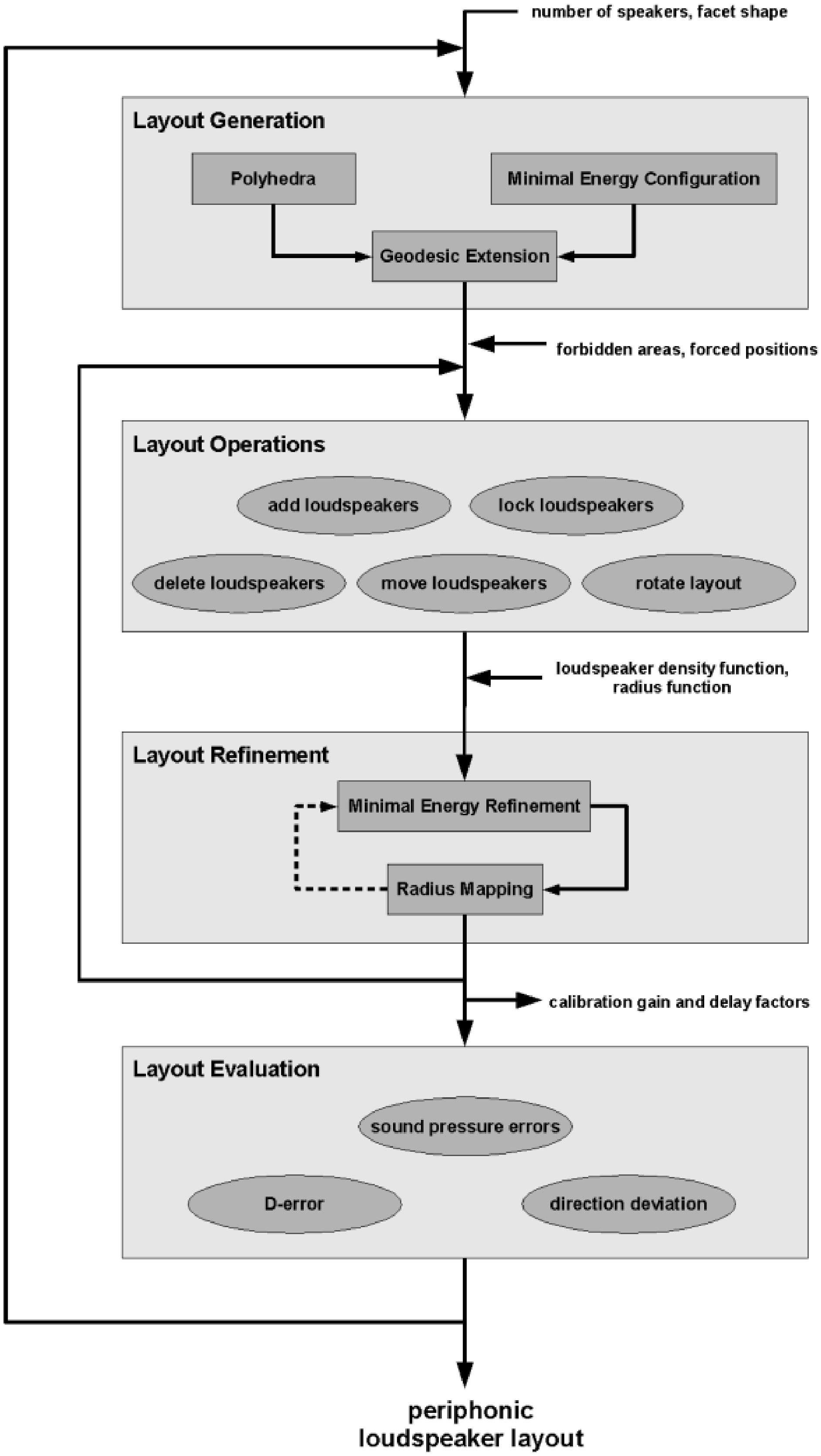


Figure 33: Extended loudspeaker layout design strategy

Part IV
Practical Contributions

In this chapter, we will discuss practical contributions to the *AlloSphere* project at the Media Arts and Technology Program of the University of California at Santa Barbara (UCSB), which has provided the background of this thesis. We will give an introduction into the AlloSphere project itself, as well as into the *CREATE Signal Library* (CSL, 'Sizzle'), a C++ framework for digital sound synthesis developed at UCSB's Center for Research in Electronic Art Technology [PR03]. Then we will discuss a set of Higher Order Ambisonic classes for CSL, which has been developed together with Graham Wakefield and Jorge Castellanos in 2005 and has been integrated in the farmework. Also, a Matlab library for the design of periphonic loudspeaker layouts will be presented, which has been developed by the author in a project related to this thesis. Eventually, we will demonstrate the application of this library by developing a scenario of a periphonic sound spatialization system for the AlloSphere.

19 Project Background

19.1 The AlloSphere

The *AlloSphere* is a joint project of the California Nanosystems Institute (CNSI) and the graduate program in Media Arts and Technology (MAT) at the University of California at Santa Barbara (UCSB). It is designed as an immersive virtual environment for multiple users, and is currently being constructed as part of an CNSI building at the UCSB campus. The completion of the building, which is already a part of the final instrument, is scheduled for spring 2006. It will be a three-story cubic anechoic chamber with extensive sound absorption treatment (4-foot foam wedges on almost all inner surfaces), containing a built-in spherical screen with a diameter of approximately 10 meters. The screen consists of perforated aluminum, designed to be optically opaque but acoustically transparent. A bridge suspended in the center of the two hemispheres accomodates about ten to twenty people. Once this construction is completed, the visual, sonic, and interactive components of the system will be integrated in the system, which will eventually represent one of the largest immersive instruments in the world. Fifteen high-resolution video projectors mounted around the seam between the two hemispheres will provide 3D projection of visual data on the entire surface of the sphere. An array of hundred loudspeakers or more will be suspended behind the projection screen, hung from the steel infrastructure of the building. This will allow for truly periphonic sound spatialization, including sounds from the lower hemisphere. The projection equipment will be driven over gigabit ethernet LAN by custom-built software running on a server farm. Interactive control of the data projection will be achieved by the means of various sensing and tracking techniques, like microphone arrays, camera tracking, mechanical and magnetic input tracking. [MAT05]

The following laboratories at UCSB will be involved in using the AlloSphere as a large-scale immersive instrument for their research:

- Center for Research in Electronic Art Technology (CREATE)
- Four Eyes Lab

- Networking and Multimedia Systems Laboratory
- VivoNets Lab
- Interactive/Algorithmic Visualization Research Lab
- Center for Bio-Image Informatics
- Research Center for Virtual Environments and Behavior
- Spatial Thinking Lab
- Vision and Image Understanding Laboratory
- National Center for Geographic Information and Analysis
- Nanoelectronics and Circuits Research Group

The research carried out by these institutions will include a wide range of scientific fields. Immersive audio systems are studied at the CREATE, including periphonic sound spatialization and sound synthesis technologies (JoAnn Kuchera-Morin, Curtis Roads, Xavier Amatriain, Stephen Travis Pope). The Four Eyes Lab will focus on immersive visual systems, contributing research on high-resolution video projection and holography (Tobias Hollerer). Sensor and camera tracking systems for gesture recognition and interpretation are developed by Edward Chang and the Four Eyes Lab (Mathew Turk, Tobias Hollerer). The Networking and Multimedia Systems Laboratory (Kevin Almeroth), the VivoNets Lab (Jerry Gibson) and other researchers (Steven E. Butner, Forrest Brewer) are concerned with system design and integrated software/hardware research. The Center for Bio-Image Informatics, B.S. Manjunath, and Edward Chang are interested in using the AlloSphere as an advanced interface for their research on multidimensional knowledge discovery. Kaustav Banerjee and the Nanoelectronics and Circuits Research Group will carry out simulations in the AlloSphere for the analysis of complex structures and systems like cells, circuits, or the Internet. Human perception, behaviour and cognition is a research area of the Spatial Thinking Lab (Mary Hegarty), the Research Center for Virtual Environments and Behavior (James J. Blascovich), and the Vision and Image Understanding Laboratory (Miguel Eckestein). This also includes Jack Loomis' research on spatial hearing. The National Center for Geographic Information and Analysis (Keith Clarke, Michael Goodchild, Jim Frew, Dan Montello) will benefit from the AlloSphere as a tool for cartographic display and information visualization, for example 'inside-out' displaying of global data (e.g. earthquake activity) as tools for collective decision-making. Several researchers will use the AlloSphere for artistic scientific data visualization and auralization. This includes George Legrady's research at the Interactive/Algorithmic Visualization Research Lab, Marcos Novak's work on visual architecture, the mixed media works by JoAnn Kuchera-Morin's , Curtis Roads and his research on microsound, web visualization software by Lisa Jevbratt, and Marko Peljhan's work on non-linear and non-hierarchical data display and usage. Altogether, the AlloSphere will serve as a research environment for some sixty students of the master and PhD programs in Media Arts and Technology, along with approximately thirty professors and post-doctoral researchers at MAT and its partner departments. [MAT05]

19.2 Audio in the AlloSphere

This thesis project is dedicated to periphonic sound spatialization in multi-user virtual environments like the AlloSphere. We have presented the theoretical foundations of spatial hearing (chapter I) as well as historical aspects of sound spatialization (chapter II). An overview of suitable algorithms which allow for reconstruction

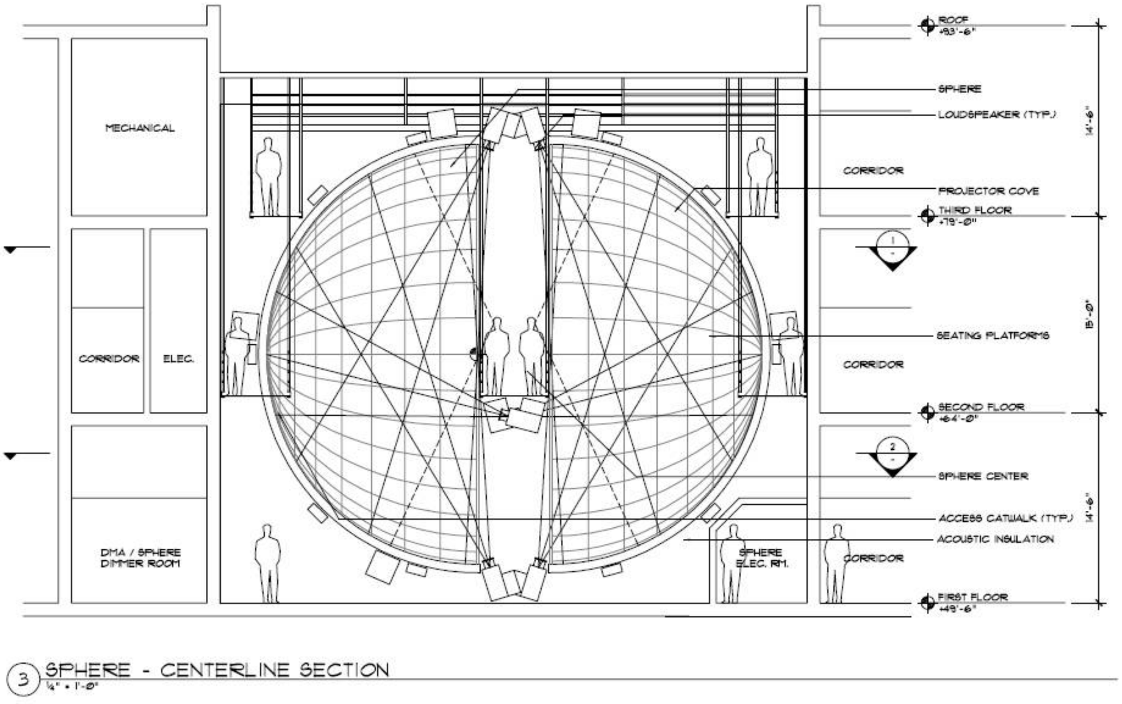


Figure 34: AlloSphere [MAT]

of periphonic soundfields has been given in chapter III. Since these have different requirements regarding the loudspeaker array, we have discussed various techniques of generating periphonic loudspeaker layouts (chapter 18), culminating in a new hybrid approach presented in chapter 18.6. Regarding its application for the design of the AlloSphere's sound spatialization engine, this strategy allows for taking into account the actual shape of the AlloSphere (radius as a function of direction) and its architectural environment (forbidden areas and forced positions for loudspeaker distribution). It allows for psychoacoustical optimization of the directional loudspeaker density and for a special focus on sound reproduction in the horizontal plane, allowing for co-existing 3D/2D systems like a 3D Higher Order Ambisonic and a 2D Wave Field Synthesis engine.

Several other aspects of the sound spatialization engine design in the AlloSphere have *not* been addressed within this project: In particular, we have not addressed the room acoustics of the AlloSphere. According decisions regarding shape and acoustical treatment have been made long before the start of this project. Severe problems regarding sound focusing in the center of the AlloSphere due to its spherical shape are expected. These worries are grounded on experience with curved surfaces in sound spatialization at the Institute of Electronic Music and Acoustics, Graz, Austria. The metal bridge in the center of the AlloSphere will be critical regarding reflections and shielding of sounds from the lower hemisphere. It will also remain to evaluate how the noise created by the video projectors can be kept at a minimum within the listening area. The hardware implementation of the AlloSphere's sound spatialization engine was not a subject of this thesis either. The main challenge in this context will be the implementation of a distributed computing system carrying out the processing of the source signals and their spatialization. The audio input section design will require the development of microphone arrays and specifications regarding microphone preamps and analog-to-digital converters.

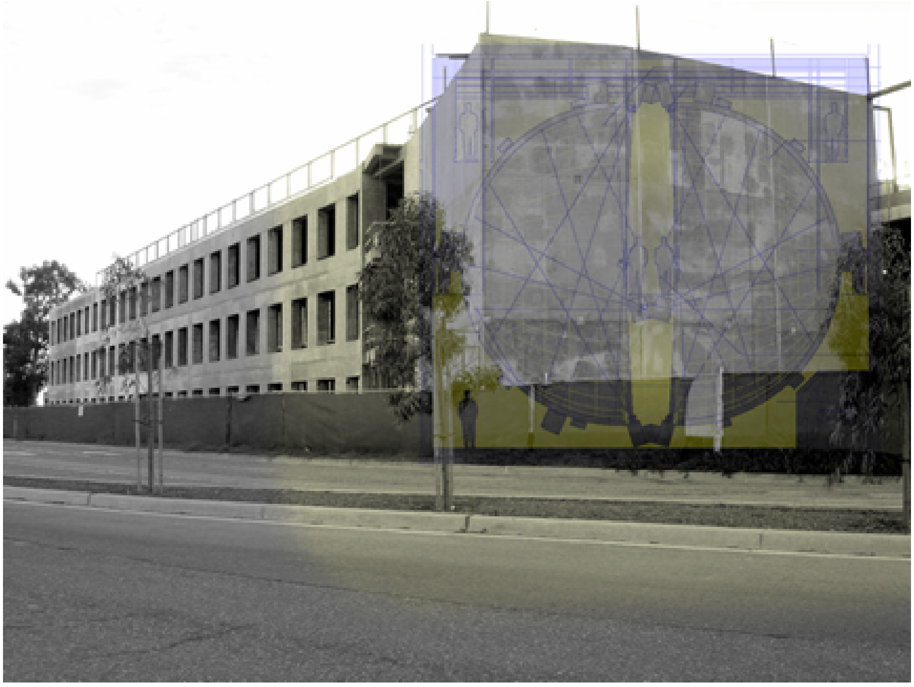


Figure 35: The AlloSphere building [MAT]

The transmission and distribution of audio signals in the AlloSphere involves the design of a gigabit ehternet LAN, a central mixing unit and decisions regarding patchbays and wiring. The audio output section requires decisions regarding the number of loudspeakers and the specification of loudspeaker models used. Stephen Travis Pope has proposed the use of electrostatic loudspeaker panels (ESLs) in the AlloSphere [Pop05], which have the advantage of low weight, facilitating loudspeaker maintainance. However, they require voltages in the kV range and their lower frequency border is as high as 200 Hz to 450 Hz, depending on the size of the panel. A system built from these speakers will require additional mid-woofers and sub-woofers, possibly distributed at lower densities than the high-frequency-range ESLs. It is not obvious how to generate loudspeaker driving signals for such a hybrid system by the sound spatialization techniques presented in this thesis. Stephen Travis Pope has suggested to address the loudspeakers over custom-built interface boxes, each carrying the required gigabit ethernet interface, the digital-to-analog converter, power amplifier, and step-up transformer [Pop05]. Also, this thesis was not concerned with the interactive control of spatialized sounds in the AlloSphere by the means of human interface devices and graphical user interfaces.

The following chapters will be dedicated to two software packages developed in the course of this thesis project: Higher Order Ambisonic classes have been developed for the CSL framework (CREATE Signal Library, [PR03]; see also chapter 19.3) with Graham Wakefield and Jorge Castellanos (chapter 20). Together with Doug McCoy's VBAP classes [McC05], they will allow for the implementation of various periphonic sound spatialization engines in the AlloSphere. Doug McCoy's work also included a hardware interface based on a 3D motion tracking system combined and a data glove, which allows the user to 'grab' sounds with their hand, and 'release' them again to different modes of movement (orbit, bouncing, resting). The second software package is the *3LD*, the Library for Loudspeaker Layout Design, which has been implemented by the author of this thesis as a set of Matlab functions

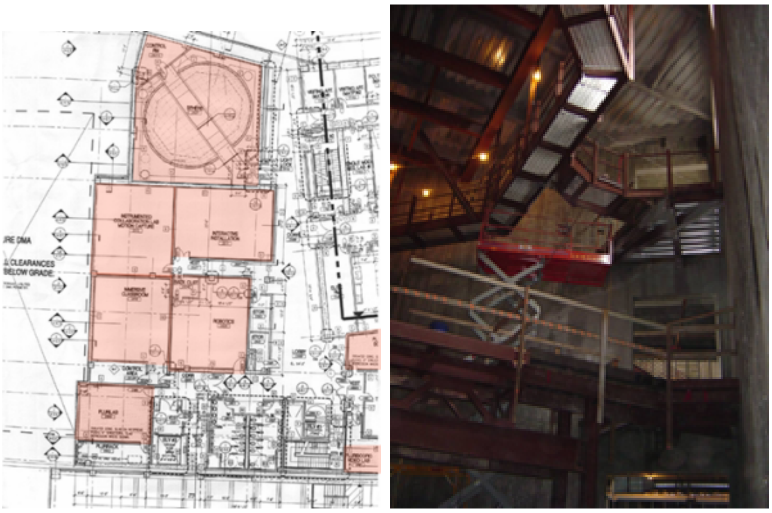


Figure 36: AlloSphere environment and interior construction (fall 2005) [MAT]

for the generation and evaluation of periphonic loudspeaker layouts (chapter 21). An application of this library is demonstrated in the final chapter of this thesis, in which we derive a hypothetical scenario of a sound spatialization system for the AlloSphere (chapter 22).

19.3 The CREATE Signal Library (CSL)

The *CREATE Signal Library* (CSL, pronounced 'Sizzle') [PR03] is a cross-platform software framework for sound synthesis and digital signal processing. It is implemented as a C++ class library and was initially written by Stephen Travis Pope as the CREATE Oscillator (CO) in 1998. In 2002, CSL was rewritten by students of the graduate program in Media Arts and Technology at the University of California at Santa Barbara. In Spring 2003, the framework has been revised again, mainly by Chandrasekhar Ramakrishnan (CSL 3). In the last months of 2005, another revision of the program has been carried out by Stephen Travis Pope, Xavier Amatriain, Jorge Castellanos, Graham Wakefield, Lance Putnam, Ryan Avery, and others (CSL 4). The new sources are already available, but not all of the classes have been ported to the current version yet. CSL runs on Linux, Unix (Solaris, Iris, OpenBSD) and MacOSX. Windows is supported, although some features (abstractions for network and thread) are missing. CSL is an open source project - the sources can be downloaded at `http://create.ucsb.edu/CSL`.

Similar to Jsyn, CLM, STK, CLAM and Cmix, CSL is a class library in a general-purpose programming language (C++), rather than a stand-alone 'sound compiler'. It is a low level synthesis and processing engine, but does not provide its own music-specific programming language, such as the Music-N languages or SuperCollider. CSL programs are written as the `main` function of a C++ program, which is linked with the CSL library, and use hardware abstraction classes for I/O ports, network interfaces and thread APIs. It is designed from the ground up to be used in distributed systems, with several CSL programs running as servers on a local-area network, streaming audio and control signals between them. [PR03]

A CSL program sets up a DSP graph, following the dataflow concept known from other sound synthesis software. The root object of the graph - for example a spatializer - is connected to the output, represented by a direct audio output API (PortAudio, CoreAudio), a socket-based network protocol (UDP, TCP/IP), a sound file (using libsndfile), or a MIDI output (PortMIDI). Signal processing in the DSP graph is done in sample blocks and follows a *pull principle*, i.e. the root object requests a buffer of samples from its inputs using the `nextBuffer` method, which is then passed on upwards the DSP graph. Once the DSP graph is worked off and the root object receives its requested buffer, the `nextBuffer` method proccesses it (for example spatializes an incoming audio signal) and passes it on to the output. Different subgraphs can run at different block sizes and sample rates, and a DSP graph can be distributed to several machines in a local-area network. [PR03]. The source code of CSL is organized in the following directories:

- **CRAM:** hosts the CREATE Realtime Applications Manager.
- **Instruments:** includes FM-, soundfile- and white noise-instruments.
- **IO:** hosts the hardware abstraction classes for audio, MIDI, soundfile, and network input and output.
- **Kernel:** contains the core classes of CSL, which will be further described below.
- **OSC:** includes the sources of the CNMAT Open Sound Control library.
- **Processors:** provides binary signal operations, filters, reverberators, etc.
- **Sensors:** includes classes to address human interface devices (P5 data glove, EBeam whiteboard, Flock of Birds magnetic tracker, etc.)
- **Sources:** contains CSL's unit generators, like oscillators, noise sources, envelopes, physical models, a wave shaper, a granulator, etc..
- **Spatializers:** provides classes for sound spatialization by the means of Higher Order Ambisonics, Vector Base Amplitude Panning, an Auralizer, and several panners.
- **Tests:** contains examples of CSL programs (i.e. `main` functions) for testing and demonstration of the CSL classes.
- **Utilities:** hosts a block resizer and other utility functions.

The classes which represent the top level of the framewok's inheritance hierarchy are located in the 'Kernel' folder. The following four of them represent CSL's *core classes*:

- `Buffer` represents a multi-channel audio sample buffer, which is passed around between unit generators and I/O objects.
- `BufferCMap` is a sample buffer used for multichannel processing.
- `Port` represents signal and control inputs and outputs.
- `UnitGenerator` is an object which fills a buffer with samples. This is CSL's central DSP abstraction.

Note that in CSL 3, it was the `FrameStream` class which was responsible for filling the audio buffers with samples. In CSL 4, this functionality has been integrated in the `UnitGenerator` class, and `FrameStream` has been removed. However, the Higher Order Ambisonic classes presented in chapter 20 have not been ported to CSL 4 yet, so in their constructors presented in chapter 20, you will still find references to the `FrameStream` class. A class new to CSL 4 is `Port`, which is an abstraction of a unit generator's inputs and outputs for audio and control signals.

Besides these core classes, the CSL kernel also provides several *mix-in classes*. For example, the `Effect` class (which was called `Processor` in CSL 3) represents a `UnitGenerator` which operates on an input port, like a filter or panner. Other subclasses can be used to add multiplicative and additive inputs (`Scalable`) or frequency inputs (`Phased`) to unit generators. Additional kernel classes include a multiplexer (`Joiner`) and demultiplexer (`Splitter`) and an abstraction of the I/O driver (`IO`).

20 Higher Order Ambisonic Classes for CSL

Together with MAT students Jorge Castellanos and Graham Wakefield, several classes for building Higher Order Ambisonic systems within the CSL framework (chapter 19.3) have been developed. These classes allow for the implementation of Ambisonic systems with the following characteristics:

- Up to third Ambisonic order
- Encoding functions using the Furse-Malham set (see chapter 15.2)
- Soundfield rotation about all three axes x, y, and z
- Decoding by the means of projection or pseudoinverse (see chapter 15.4)
- Basic, max r_E and in-phase decoder flavors (see chapter 15.5)

The Higher Order Ambisonic classes have been incorporated into the CSL framework - in the distribution they can be found in `/CSL/Spatializers/Ambisonic`. However, at the point of writing, they have not yet been ported to the new version (CSL 4). The constructors and the code example presented in this chapter are still based on CSL 3. Thus, they still refer to the `FrameStream` class, which has been abandoned in CSL 4 (chapter 19.3).

20.1 The HOA_AmbisonicFramestream Class

`HOA_AmbisonicFramestream` represents the superclass from which most other Higher Order Ambisonic classes inherit. Generally, any class which requires Ambisonic encoded material at its input and/or output represents an Ambisonic framestream, which knows about its order and the number of channels it carries. The motivation of introducing this class was that in hybrid order systems (see chapter 15.2), the order of the input cannot be guessed any more from the number of channels. Thus, an Ambisonic framestream (e.g. a decoder) needs to be able to ask its input (e.g. an encoder) about the order it provides. However, it has to be pointed out that although we have included them into our considerations regarding the design of our classes, hybrid orders are currently not supported in the right way. It is possible for each Ambisonic framestream to separately specify a horizontal and a vertical order, but the same 3D encoding equations are applied in the encoding process of vertical as well as horizontal Ambisonic channels. This problem will be addressed when the classes are ported to CSL 4 (expected for spring 2006). In the meantime, it is recommended not to use the constructors of the HOA classes which allow for separate specification of horizontal and vertical order, but to to restrict oneself to global Ambisonic order definition using the according constructors which will also be presented in the following chapters.

20.2 The HOA_Encoder Class

The `HOA_Encoder` class inherits from `HOA_AmbisonicFramestream`. It allows for encoding mono audio signals to Higher Order Ambisonic soundfields of up to third order. A typical constructor of the class is

```
HOA_Encoder(FrameStream &input, unsigned int order = 1,
double azimuth = 0.f, double elevation = 0.f);
```

As can be seen from the constructor, the encoder defaults to encoding a sound source to the zero (i.e. front) position at first order. Note that the actual encoding functions are hosted by the `HOA_Utilities` class. The sound source azimuth and elevation can be set using the following methods:

```
void setAzimuth(double azimuth);
void setElevation(double elevation);
```

Note that once the `HOA_xxx` classes have been ported to the new version 4 of CSL, the audio input will be a `UnitGenerator` rather than a `FrameStream`, and the azimuth and elevation values will be represented by instances of the `Port` class (see chapter 19.3).

20.3 The HOA_Mixer Class

This class, which also inherits from `HOA_AmbisonicFramestream`, is used to mix Ambisonic framestreams - for example superponing several encoded sound sources to a single Ambisonic soundfield. Our original design intended to provide the `HOA_Encoder` class with a framestream with multiple channels representing the audio signals intended for encoding. Two additional framestreams with the same amount of channels were supposed to provide the azimuth and elevation information of the sound sources. In order to mix several mono audio signals into one multichannel framestream, we wrote a class named `Glue`, which did not work as intended, due to reasons outside of the scope of our project. Our temporary solution to the problem was to encode each sound source separately using its own instance of the `HOA_Encoder` class, and eventually mix them together by the means of `HOA_Mixer`. The problems we encountered have been considered in the current re-design of CSL, and thus a more sophisticated solution can be expected once the HOA classes are ported to CSL 4. A typical constructor of the `HOA_Mixer` class is

```
HOA_Mixer(FrameStream &input, unsigned maxInputs = 64);
```

and new inputs can be added to the mixer using the method

```
bool addInput(FrameStream &input);
```

20.4 The HOA_Rotator Class

The `HOA_Rotator` class inherits from `HOA_AmbisonicFramestream` as well. It is capable of rotating Ambisonic-encoded soundfields around any combination of the three x (tilt), y (tumble) and z (rotate) axes. Note that inside the class, the operations are always executed in the same order, which is first tilt, then tumble, and eventually rotation. You will have to provide the three angles accordingly in order to obtain your desired final rotation axis. Some standard constructors of the `HOA_Rotator` class are

```
HOA_Rotator(FrameStream &input);
HOA_Rotator(FrameStream &input, unsigned order);
```

The first constructor uses the order of the encoded input material. The second constructor can be used to override this parameter and rotate the soundfield at a lower order than given by the input. This makes sense whenever you know that later you are going to decode the soundfield at a lower Ambisonic order than is provided by the encoded material, for example because there are not enough loudspeakers to fullfill the $L \geq N$ criterion (eq. (24)). In such situations, CPU resources can be saved by not rotating Ambisonic channels which will never be decoded. The rotation angles can be set by the following methods:

```
void setTilt(float input);
void setTumble(float input);
void setRotate(float input);
```

20.5 The HOA_SpeakerLayout Class

The HOA_SpeakerLayout class allows for reading and writing text files containing the loudspeaker position information. It does not represent an HOA_AmbisonicFramestream nor does it inherit from any other CSL class, since it only includes a parser which processes the loudspeaker position information so it can be used by the decoder class (chapter 20.6) for setting up the re-encoding matrix of the loudspeaker layout. The loudspeaker positions can be specified in either spherical (azimuth, elevation, radius) or cartesian (x, y, z) coordinates, and angles can be specified in degrees or radians. This is achieved by adding an according keyword (SPHERICAL-DEGREES, or SPHERICAL-RADIANS, or CARTESIAN) in the header of the text file. Note that the radius information of the loudspeakers is included, keeping future system extensions towards distance coding by the means of the NFC-HOA format (chapter 15.6) in mind. An example for a text file which can be read by the HOA_SpeakerLayout class is

```
/ An Octahedron Loudspeaker Layout
SPHERICAL-DEGREES
/azimuth  elevation   radius
    45         0         1         / left front speaker
   135         0         1         / left rear speaker
  -135         0         1         / right rear speaker
   -45         0         1         / right front speaker
     0        90         1         / top speaker
     0       -90         1         / bottom speaker
```

Everything right of a '/' is interpreted as a comment. Some typical constructors of the class are

```
HOA_SpeakerLayout();
HOA_SpeakerLayout(char *filePath);
```

In the second constructor, the path to the text file containing the loudspeaker position information is passed as an input argument. If this path is not specified as in the case of the first constructor, the class defaults to a loudspeaker layout used in the laboratory of UCSB's Graduate Program in Media Arts and Technology, which consists a ring of eight loudspeakers in the horizontal plane, a ring of four loudspeakers on the floor, a ring of four loudspeakers on the ceiling, a central loudspeaker on the ceiling, and a subwoofer.

20.6 The HOA_Decoder Class

This class inherits from HOA_AmbisonicFramestream and decodes Ambisonic soundfields encoded by the HOA_Encoder class to the loudspeaker layout specified by the

HOA_SpeakerLayout class. The decoding method can be specified to either use projection of the Ambisonic channels onto the loudspeaker layout (chapter 15.4.2) or to derive the pseudoinverse of the re-encoding matrix (chapter 15.4.1). The decoder flavors specified in chapter 15.5 have also been implemented, i.e. a basic decoder as well as an in-phase decoder and a max-r_E decoder. To build the re-encoding matrix of the loudspeaker array, the HOA_Decoder class uses the function fumaEncodingWeights in HOA_Utilities, which is also employed by HOA_Encoder for virtual sound source encoding. This avoids unnecessary code duplication. Constructors of the decoder class can look like

```
HOA_Decoder(FrameStream &input,
HOA_DecoderMethod method = HOA_PSEUDOINVERSE,
HOA_DecoderFlavour flavour = HOA_BASIC);
```

or

```
HOA_Decoder(FrameStream &input, unsigned order,
HOA_SpeakerLayout &spkLayout,
HOA_DecoderMethod method = HOA_PSEUDOINVERSE,
HOA_DecoderFlavour flavour = HOA_BASIC);
```

In the first case, the decoder defaults to the standard loudspeaker layout provided by HOA_SpeakerLayout (chapter 20.5). It can also be seen that by default the decoder uses the pseudoinverse decoding method and a basic decoder flavor. The order at which the decoder operates is given as the minimum of

- the order of the encoded input material,
- the order specified in the constructor,
- the maximum order which can be realized on the given loudspeaker layout so that $L >= N$ (eq. (24)).

20.7 The HOA_Utility Class

The HOA_Utility class serves as a container for several functions which are shared by multiple Higher Order Ambisonic classes or represent low-level functions not included in the class which uses them in order to keep the code concise. Examples of functions in HOA_Utility are

```
unsigned int orderToChannels(const ambisonicOrder order);

void SingularValueDecomposition(sample** a, int m, int n,
sample* w, sample** v);

void fumaEncodingWeights(sample *weights, const ambisonicOrder &order,
sample &azimuth, sample &elevation);
```

The first function is an implementation of equation 19, which is used to derive the number of Ambisonic channels from the order of a 3D system. The second function is used in the derivation of the pseudoinverse of the re-encoding matrix by the HOA_Decoder class (chapter 20.6). The third function hosts the Ambisonic encoding equations (table 1), which are used by the encoder class for encoding the sound sources, and by the decoder class for re-encoding the loudspeaker positions.

20.8 Code Example

```
int main (...) {

... ... ...

gIO = new PAIO(...);

... ... ...

// Encode first source at third order to the zero (i.e. front) direction
HOA_Encoder source1(signal1, 3, 0.0, 0.0);

// Encode decond source at third order to 90 degrees left, 45 degrees up
HOA_Encoder source2(signal2, 3, CSL_PI/2.0, CSL_PI/4.0);

// Mix both sources to a single Ambisonic soundfield
HOA_Mixer ambiMix(source1);
ambiMix.addInput(source2);

// Rotate the Ambisonic soundfield about the z axis
HOA_Rotator rotator(ambiMix);
rotator.setRotate(2.f);

// Get the (default) loudspeaker layout information
HOA_SpeakerLayout speakerlayout;

// Decode the Ambisonic soundfield at second order using projection.
// The decoder flavor will default to 'basic'.
HOA_Decoder decoder(rotator, 2, speakerlayout, HOA_PROJECTION);

// Set the root of the DSP graph and start the audio driver
gIO->set_root(decoder);
gIO->open();
gIO->start();

... ... ...

}
```

20.9 Future Work

The design of the CSL sound spatialization classes (`VBAP`, `HOA_xxx` classes, etc.) is expected be somewhat generalized in the new version 4 of CSL, so that various spatializers share a common format representing the spatial position information of virtual sound sources and loudspeakers. This could include Pulkki's loudspeaker layout triangulation algorithm implemented by Doug McCoy in his CSL Vector Base Panning classes [McC05] and loudspeaker array calibration as described in chapter 18.5. Regarding possible extensions of the `HOA_xxx` classes, it is recommended to focus on the following issues:

- The extension to even higher orders $M > 3$ is the most important issue regarding implementations in the AlloSphere. A decision has to be made whether this extension will be implemented by the means of hardcoded encoding functions, or if the Ambisonic encoding functions are generated on-the-fly, using their generic definition. Note that a generic definition of higher-order rotation matrices is only possible for rotation about the z axis, but not for the tilt and tumble operations.

- Hybrid order representations (chapter 15.2) have been considered but not yet implemented in the right way, which could be done in a future version of the Higher Order Ambisonic CSL classes.
- Various encoding conventions (N3D, SN3D, FuMa, etc.) should be supported. An Ambisonic framestream will have to include information about the convention according to which it was encoded as a member variable, so that classes operating on the encoded material (rotators, decoders) can adopt themselves accordingly. For example, the tilt and tumble matrices vary for different encoding conventions, and an Ambisonic decoder has to apply the same (re-)encoding convention which has also been used in the source encoding process.
- New soundfield operation classes could be added, providing capabilities for mirroring, W-Panning, dominance, etc. See chapter 15.3 for background information. More experimental operations in the Ambisonic domain could be implemented by the means of existing CSL classes (filters, delays, etc.).
- Sound source distance coding by the means of the NFC-HOA format (chapter 15.6) should be included. This would require an additional encoder class, applying filter operations on the Ambisonic channels. This might be achieved by the means of existing CSL filter classes. Note that no additional rotation and decoding classes have to be written to handle NFC-HOA encoded material.
- Sound Object encoding in its simplest form (chapter 16.1) is already possible by re-using the `HOA_Encoder` class accordingly. However, the encoding of surface shapes and frequency-variant radiation patterns requires spherical harmonic impulse responses (chapter 16.2), which would have to be additionally implemented.
- The decoder class could be extended towards a very generic design: Note that the max-r_E and in-phase weighting factors can be defined generically for arbitrary orders. A more sophisticated design would allow for window-applied decoding rather than a mere choice between the extremes of basic, max-r_E or in-phase decoding. Such an extension would involve the design of a user interface, similar to the ones known from FIR filter design software.
- Room reflection cancellation (chapter 15.7) is an interesting issue, yet it faces practical difficulties in the high-frequency range. Regarding its application in the AlloSphere, an evaluation of the necessity for a room reflection cancellation engine should be considered after the building is finished and its acoustical properties are known in detail. Since the required impulse responses of the loudspeakers are always restricted to a specific playback situation, it has to be doubted that a wide range of users will benefit from a CSL implementation of Ambisonic room reflection cancellation.

21 3LD: Library for Loudspeaker Layout Design

The *3LD* (*'Trip-L-D'*) **L**ibrary for **L**oudspeaker **L**ayout **D**esign has been implemented by the author of this thesis in the course of a project related to this thesis. It provides functions for generating periphonic loudspeaker layouts and evaluating them in combination with different sound spatialization algorithms. The library is available at `http://iem.at`. In this chapter, we will give a short description of the 3LD functions and and their useful combination. Note that the source code is well documented, and that typing `help <functionname>` in Matlab will return a description of the functionality and the input and output arguments of any 3LD function. Some demos are included in the package as well. All functions use the cartesian and spherical coordinate systems known from this thesis (figure 1) as well as from native Matlab functions like `sph2cart`.

21.1 Core Functions

The two functions presented in this chapter have been implemented as useful extensions to native Matlab functions, with a range of possible applications clearly exceeding the field of periphonic loudspeaker layout generation.

21.1.1 spharmonic

This function uses Matlab's native `legendre` to derive the spherical harmonics at the given azimuth and elevation angles. For the specified order, all spherical harmonics are calculated. Note that `legendre` uses the term 'degree' and the subscript 'n' for what we refer to in this thesis as 'order' and denote with the subscript 'm'. On the other hand, our subscript 'n' is replaced by 'm' and refered to as 'order' in `legendre`. Since `spharmonic` represents a direct extension to this native Matlab function, it uses the same naming convention. However, in 3LD's Ambisonic encoding/decoding functions (chapter 21.3), the naming conventions known from this thesis (chapter 15.1) have been applied, since these are common to Ambisonic literature. Also, the order of the spherical harmonic functions in the rows of the `spharmonic` output array represents a logical extension of the convention applied in `legendre`, which differs from the channel order used in this thesis (chapter 15.1.1). In 3LD's Ambisonic functions, it is possible to choose among both conventions. `spharmonic` provides the same normalization options as `legendre`, which are 'unnorm', 'sch', and 'norm'. Note that 'sch' is identical to the SN3D Ambisonic encoding convention (chapter 15.1.1), but 'norm' seems to be *not* identical to the N3D encoding convention.

21.1.2 ezspherical

This function is a spherical extension of the native Matlab function `ezpolar`. `ezspherical` plots spherical functions defined as function handles with two arguments representing azimuth and elevation. For example, the handle to a spherical harmonic function Y_{11}^{1} can be expressed in Matlab as `X = @(az,elev) cos(az).*cos(elev)`. The command `X(pi,pi/2)` then returns the value of that function in the direction $\theta = \pi, \phi = \pi/2$. `ezspherical(X)` allows for direct plotting of the function handle rather than first calculating the function values for the points of a grid. `ezspherical(X)` creates such a grid internally - its resolution can be specified as an optional input argument.

21.2 Loudspeaker Layout Generation and Modification

3LD provides functions for generating periphonic loudspeaker layouts by the means of various techniques discussed in chapter 18. All of them use an indentical layout representation format: the coordinates of the loudspeakers are returned as L-by-3 arrays, containing the x,y,z coordinates of the L loudspeakers. Some of the functions also provide tessellation of the surface defined by the loudspeakers, which is then represented by an F-by-P array of F facets. P is the maximum number of vertices in a facet of the polyhedron, e.g. $P = 3$ in the case of a triangulated solid like the *octahedron* (chapter 18.2.1). Polyhedra combining facets of different shape (like the bucky ball from figure 23) are supported by filling up the according rows of the faces matrix with `NaN`s.[46] Note that faces are always oriented counterclockwise as seen from the center of the polyhedron. Functions which provide both, coordinates and tessellation of a polyhedron return a structure with fields `vertices` and `faces`, representing the two arrays. This structure can be plotted directly using the native Matlab function `patch`.

21.2.1 platonicsolid

This funtion returns a vertices/faces structure of either one of the five Platonic solids (see chapter 18.2.1). The radius of the solid can be specified as an optional input argument. For example, to plot an icosahedron with a radius $r = 3$, type

```
p = platonicsolid('ico',3);
plot3LD(p);
```

21.2.2 bucky2

This function returns the vertices and faces of a *truncated icosahedron*, also known as *bucky ball* (see chapter 18.2.2). It uses the internal Matlab function `bucky`, which returns the coordinates of the vertices and an adjacency matrix defining the edges of the polyhedron. `bucky2` uses the first as an output argument, whereas the adjacency matrix is replaced by a matrix representing the faces. The radius of the polyhedron can be passed as an input argument and will default to one if not specified.

21.2.3 geosphere

The `geosphere` function is capable of tessellating an arbitrary polyhedron. It is an implementation of the generalized approach to building geodesic spheres which we have presented in chapter 18.3. Sphere approximations can be built from Platonic solids, like in the function call

```
p = geosphere('oct',2,1);
plot3LD(p);
```

Note that you can plot the output of this function directly using 3LD's `plot3LD` (chapter 21.5.4) or Matlab's `patch`, which is true for the following examples as well. You can also define an existing polyhedron as the first input argument of `geosphere`, like in

```
b = bucky2;
p = geosphere(b,1,1);
plot3LD(p);
```

[46] In Matlab, 'NaN' stands for 'not a number'.

The second input argument of `geosphere` specifies the frequency f of the geodesic sphere (see chapter 18.3). For $f = 1$, the faces are midpoint-triangulated, whereas for $f > 1$, they are tessellated at the frequency f if possible,[47] and midpoint-triangulated otherwise. The frequency can also be specified as a row vector in order to execute multiple iterations of the tessellation process at different frequencies, like in

```
p = geosphere('oct',[2 3]);
plot3LD(p);
```

Note that in this example, we have not provided a third input argument, which represents the radius of the geodesic sphere. This has the effect that the polyhedron is tessellated, but the new vertices are not pushed out to the radius. The frequency input argument can also be extended to a matrix, with columns representing different tessellation frequencies for different facet shapes of a polyhedron. In the following example, we first build a bucky ball, replace one of its faces by a triangle and a rectangle, and then pass the resulting polyhedron to `geosphere`:

```
p = bucky2;
p.faces(1,:) = [];
p.faces(end+1:end+2,:) = [3 5 4 NaN NaN NaN ; 3 2 1 5 NaN NaN];
p = geosphere(p,[2 0; 3 1; 1 4; 7 1]);
plot3LD(p); view(111,83), zoom(2)
```

The frequency matrix can be read as follows: the first column represents the tessellation frequency for triangular faces, the second row for rectangles, the third for pentagons, and so on. Thus, in the first iterations, triangles are triangulated at $f = 2$, rectangles are rectangulated at $f = 3$, pentagons are midpoint-triangulated ($f = 0$), and hexagons are midpoint-triangulated as well ($f \neq 0$), since there is no other option for them (see chapter 18.3). In the second iteration, triangles, are ignored, rectangles are midpoint-triangulated, and the entries for pentagons and hexagons are without any effect, since there are no more left. For the radius input argument, you can also specify a handle to a spherical function. Its first argument will be interpreted as the azimuth, and its second function as the elvation.

```
radius = @(az,elev) abs(cos(az) .* cos(elev)) + 2;
p = geosphere('oct',[2 2 2],radius);
plot3LD(p);
```

Note that spherical function handles can be easily designed using `handlespharm` (chapter 21.5.2) and `ezspherical` (chapter 21.1.2).

21.2.4 minenergyconf

`minenergyconf` returns the vertices of a minimal energy configuration (see chapter 18.4). Configurations can be built for an arbitrary number of electrons and iterations, like in

```
v = minenergyconf(20,2);
plot3(v(:,1),v(:,2),v(:,3),'o'); grid on;
```

[47] I.e. if the facet is a triangle or rectangle; see chapter 18.3)

for 20 electrons and 2 iterations. You can also plot the output by first triangulating the configuration with Tianli Yu's `TriangulateSpherePoints`[48] and then plotting the output with 3LD's `plot3LD`:

```
p.vertices = minenergyconf(20,10);
p.faces = TriangulateSpherePoints(p.vertices);
plot3LD(p);
```

The returned polyhedron should remind you more and more of a sphere as you apply additional iterations. To refine an existing polyhedron, like the one returned from the last example, try:

```
q.vertices = minenergyconf(p.vertices,50);
q.faces = TriangulateSpherePoints(p.vertices);
plot3LD(q);
```

which is here refined for another 50 iterations. Non-homogeneous electron distributions can be defined with an additional input argument, representing a handle to a spherical electron density function. Similarly, the radius of the configuration can be specified as a handle to a spherical function simply as a scalar as in `geosphere` (chapter 21.2.3). Note that inhomogeneous electron densities and radii will directly affect the stability of the configuration as described in chapter 18.4. Also, you have to keep in mind that `TriangulateSpherePoints` can only tessellate the surface of a unit sphere and will thus be confused whenever the radius of all vertices is not constant one. Density and radius functions can be generated using the 3LD functions `handlespharm` and `ezspherical`. Selected electrons can be locked to their initial positions in the refinement process by adding another input argument which is a vector representing the status of each electron (0 for unlocked, $\neq 0$ for locked electrons). The power of the repulsion forces p $(1/d^p)$ can be specified with another input argument and defaults to $p = 2$ if not specified. Consider the following example:

```
% Build the initial layout
p = geosphere('ico',[2],1);

% Lock the loudspeakers on the equator
lock = zeros(size(p.vertices,1),1);
[az,elev] = cart3sph(p.vertices);
lock(elev == 0) = 1;

% Define the electron density function and plot it
density = @(az,elev) abs(sin(elev)) + 0.1;
plot3LD(density);

% Plot the initial loudspeaker layout
h = plot3LD(p,lock);view(0,0); view(125,30)

% Refine the layout and plot each iteration
```

[48] This function is available at `http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=5964&objectType=file`. It triangulates any configuration of vertices on the unit sphere. However, it cannot be applied to other surfaces, which has to be kept in mind when using 3LD's minimal energy algorithm with a radius function which is not constant one everywhere.

```
for i=1:60
    p.vertices = minenergyconf(p.vertices,1,density,[],lock);
    set(h,'Vertices',p.vertices);
    drawnow;
end
```

The plot clearly shows that in the refinement process, the vertices move away from the equator towards the north and south pole, as would have been expected from the plot of the electron density function. To increase the stability, the electrons at the equator have been locked before the refinement process. Try running the same example without locking the equator loudspeakers, which destroys the symetry of the layout. Note that in order to plot the entire process, the single iterations of the refinement process have been processed in a for loop. For increased efficiency, and if no plotting of the process itself is required, this loop can be omitted and all iterations be calculated at once.

21.2.5 rotate_xyz

This function can be used to rotate a polyhedron about either of the x, y, and z axes. Note that this actually uses rotation matrices of first order, which are also applied in Ambisonic soundfield rotation (chapter 15.3.1). Input arguments are a V-by-3 array representing the vertex coordinates of a polyhedron, and the desired axis and angle of the rotation. An arbitrary rotation axis can be realized by subsequent application of this function with different axes. The function returns the rotated vertices array. For example to rotate an octahedron 45 degrees around the x axis, type

```
p = platonicsolid('oct');
p.vertices = rotate_xyz(p.vertices,'x',pi/4);
```

21.2.6 map_to_surface

This function maps the vertices of a polyhedron to a given surface while retaining their directions. Spherical surfaces can be specified as scalars representing their radius, while more complex surfaces have to be defined as handles to a spherical function of azimuth and elevation. Such functions can be conveniently generated using 3LD's `handlespharm` (chapter 21.5.2) and `ezspherical` (chapter 21.1.2). For example, to map the vertices of a dodecahedron to a non-spherical surface, try and `ezspherical`.

```
p = platonicsolid('dodec');
radius = handlespharm('abs(Y(1,0)) + 1');
p.vertices = map_to_surface(p.vertices,radius);
plot3LD(p);
```

A fancier example is

```
p = geosphere('ico',[2,2],1);
radius = handlespharm('abs(3*Y(7,-5))+1');
p.vertices = map_to_surface(p.vertices,radius);
plot3LD(p);
```

21.3 Loudspeaker Driving Signal Calculation

3LD provides function for simulating Vector Base Panning and Higher Order Ambisonic systems. Note that these do not actually process any audio signals, but calculate the loudspeaker gains required to reproduce a number of sound sources at the requested positions. Together with the soundfield rendering functions presented in chapter 21.4, they can be used to simulate an original soundfield and its reconstruction by a Vector Base Panning or Higher Order Ambisonic system.

21.3.1 amb3d_encoder

This function calculates the gain factors of the Ambisonic channel for an arbitrary number of virtual sound sources in a periphonic soundfield. Required input arguments are the Ambisonic order and the azimuth and elevation angles of the sound sources. Different gain factors can be applied to the sound sources. Additionally, it is possible to specify whether the sources are fed by the same audio signal, in which case their Ambisonic channel gain factors are superponed. Another input argument allows to specify how the Ambisonic channels are sorted in the output array to either match the convention used in `spharmonic` (chapter 21.1.2), or the one used in this thesis (chapter 15.1.1). If not specified, the first is assumed. Note that the SN3D encoding convention is applied (chapter 15.2). The following example returns the Ambisonic channel gains of two sound sources in front and on top of the listening position, encoded at third order.

```
source_position = [0 0;0 pi/2];
source_gain = [1 0.5];
B = amb3d_encoder(3,source_position,source_gain)
```

Note that the output B is a structure with fields 'gain', representing the Ambisonic channel gains, and 'sort', specifying how the channels are sorted (in this case as in `spharmonic`). The decoder requires this information for re-encoding the position information of the loudspeaker array. The rows in 'gain' represent the Ambisonic channels, and the columns the different sound sources.

21.3.2 amb3d_decoder

This function derives the loudspeaker gain factors for a periphonic Ambisonic system from the Ambisonic channel gains as calculated with `amb3d_encoder` (chapter 21.3.1). Additionally, the azimuth and elevation angles of the loudspeakers have to be specified. Another input argument represents the Ambisonic order at which the decoder operates. If not specified, the order of the encoded input is used. Naturally, the decoder's order can only degrade but not increase the order of the encoded input. The decoding method (projection or pseudoinverse, see chapter 15.4) and the flavor of the decoder (basic or in-phase, see chapter 15.5) can be specified as well. For the calculation of the loudspeaker layout's re-encoding matrix (chapter 15.1.2), this function re-uses the `amb3d_encoder` function. To get the re-encoding right, the decoder needs to know how the Ambisonic channels are sorted in the field 'gain' of its input structure, an information which is obtained from the field 'sort'. To decode the output of the example from the last chapter to an icosahedron at second order, type

```
p = platonicsolid('ico');
[az elev] = cart3sph(p.vertices);
g = amb3d_decoder(B,[az elev],2,'pseudoinverse','inphase')
```

The rows in the output array g represent the loudspeakers, and the columns the different sound sources.

21.3.3 amb3d_regularity

To evaluate the regularity of a loudspeaker layout regarding Ambisonic sound spatialization (see chapter 18.1.2), this function can be used to determine the condition number of the re-encoding matrix of a given loudspeaker layout. Another return argument specifies whether the layout is regular, semi-regular or irregular as defined by Daniel. For example to find out that a dodecahedron is regular for second order Ambisonics, type

```
p = platonicsolid('dodec');
[az elev] = cart3sph(p.vertices);
amb3d_regularity(2,[az elev])
```

21.3.4 vbp

This function calculates the loudspeaker gain factors of a 2D or 3D Vector Base Panning System (chapter 13) for an arbitrary number of virtual sound sources. The azimuth and elevation angles of the sources and loudspeakers are required input arguments, as well as a definition of the loudspeaker pairs (2D) or triples (3D). Note that in the latter case, the faces field of an output structure provided by `platonicsolid` or `geosphere` can be used if the faces of the according polyhedron are triangular. As in the case of the `amb3d_encoder` function (chapter 21.3.1), it is possible to specify whether the sources are fed by the same signal, i.e. whether the loudspeaker gains due to different sources can be superponed. Varying gain factors can be applied to the sound sources as well. Another input argument allows for a choice between Vector Base Amplitude and Vector Base Intensity Panning. The following example demonstrates how to receive the loudspeaker gains of two sound sources in front and below the listener for an intensity panning system with an octahedron loudspeaker layout.

```
source_position = [0 0;0 -pi/2];
p = platonicsolid('oct');
[az elev] = cart3sph(p.vertices);
are_identical = 1;
g = vbp(source_position,[az elev],p.faces,'vbip',[],are_identical)
```

The rows in the output array g represent the loudspeakers, and the columns the different sound sources. In this case, there is only one column, since the source signals are identical and their loudspeaker gains have thus been superponed.

21.3.5 calibrate_layout

This function returns the gain and delay calibration values of a loudspeaker layout required to compensate for differing radii of the loudspeakers in the array (see chapter 18.5). To get the calibration values of a tetrahedron layout in which one loudspeaker has been pushed away from the sweet spot, try

```
p = platonicsolid('tetra',1);
p.vertices(1,:) = [0.8, 0.8, 0.8];
p.vertices
[g,t] = calibrate_layout(p.vertices)
```

where `g` represents the calibration gains and `t` represents the time delays in seconds for the four loudspeakers in the layout.

21.4 Soundfield Rendering and Evaluation

21.4.1 soundfielder

With the `soundfielder` function, soundfields in air created by multiple monochromatic sound sources with either plane or spherical wavefront propagation characteristics can be rendered. The positions of the sources and sinks (i.e. the 'measuring points') of the soundfield have to be specified as well as the frequencies and radiation types (plane or spherical) of the sources. The direction of wavefront propagation (outgoing or incoming) can be specified separately for each source. Soundfields can also be rendered for multiple points in time by a single function call and at arbitrary temperatures (default: 273.15 Kelvin = 0 degrees Celsius). Generally, the soundfields created by all specified sources are superponed, but they can also be returned separately if desired. The four output fields of the function represent:

- the complex sound pressure (eq. (1))
- the complex sound velocity (eq. (2))
- the complex velocity vector (eq. (3))
- the complex $\vec{u}$ velocity (eq. (5))

The following example shows the wavefront propagation of a Hanning-weighted impulse built from several monochromatic sources:

```
% Number of pics in the animation
num_pics = 20;

% Specify sound source position and radiation pattern
src_position = [2 0 0];
type = 'spherical';

% Define frequencies of Hanning-weighted impulse partials
center_freq = 100; upper_freq = 500;
freq = [center_freq:center_freq:upper_freq];
num_sources = length(freq);

% Define gains of Hanning-weighted impulse partials
window = hanning(2 * length(center_freq:center_freq:upper_freq));
gain = window( ceil(length(window)/2)+1 : length(window) );

% Use same source position for all impulse partials
src_position = repmat(src_position,num_sources,1);

% Specify sink positions (here in the x/y plane)
x = -1:0.05:1; y = -1:0.05:1; z = 0;

% Define time points for rendering
time = 0 : 1/(center_freq*num_pics) : 1/center_freq - 1/(center_freq*num_pics);

% Render the sound pressure field
```

```
pressure = soundfielder(src_position,freq,type,x,y,z,time,gain);

% Plot the wavefront propagation
for i = 1:num_pics
    surf(x,y,real(pressure(:,:,i))),
    shading interp, colormap gray, view(0,90), grid off,
    axis([-1 1 -1 1 -1 1 -1 1]), axis square
    animation = getframe;
end
```

The next example shows how to use `soundfielder` together with `vbp` to resynthesize this soundfield by the means of Vector Base Amplitude Panning on a tetrahedron loudspeaker layout. The loudspeakers are assumed to emit spherical wavefronts.

```
% Get sound source direction
[src_az, src_elev] = cart3sph(src_position);

% Define loudspeaker layout properties
spk = platonicsolid('tetra',2);
[spk_az, spk_elev] = cart3sph(spk.vertices);
spk_type = 'spherical';

% Derive VBP loudspeaker gains
spk_gain = vbp([src_az, src_elev],[spk_az, spk_elev],spk.faces,'vbap',gain);

% Each loudspeaker contributes to the reconstruction of each partial, so we
% accordingly replicate our frequency matrix
num_speakers = size(spk_gain,1);
freq_rep = repmat(freq,num_speakers,1);
freq_rep = freq_rep(:);

% Superpone the num_speakers*num_sources soundfields
pressure_vbp = soundfielder(repmat(spk.vertices,num_sources,1),...
freq_rep,spk_type,x,y,z,time,spk_gain(:),1);

% Plot the reconstructed wavefront propagation
for k = 1:num_pics
    surf(x,y,real(pressure_vbp(:,:,k))),
    shading interp, colormap gray, view(0,90), grid off,
    axis([-1 1 -1 1 -1 1 -1 1]), axis square
    animation = getframe;
end
```

Try to use an octahedron instead of the tetrahedron layout to see how the reconstruction drastically improves due to the fact that a loudspeaker is located directly at the position of the virtual source.

21.4.2 direction_deviation, pressure_errors

The function `pressure_errors` calculates the *squared sound pressure error* and the *sound pressure amplitude error* (see chapter 17.2) of a reconstructed sound pressure field with regards to a reference field. The first output argument of the `soundfielder` function (chapter 21.4.1) can be used to derive the input arguments of this function. The output arrays represent the sound pressure errors at each point of the field.

The function `direction_deviation` calculates the direction deviation (see chapter 17.3) among two complex soundfields, using input arrays denoting the direction of the two soundfields, like their *velocity vectors* or their *active velocities* as calculated by the `soundfielder` function (chapter 21.4.1). The output array represents the direction deviation at each point of the two soundfields in radians.

21.5 Helper Functions

The 3LD library also provides several helper functions to facilitate the process of periphonic loudspeaker layout generation.

21.5.1 solospharm

This function can be used to evaluate a single spherical harmonic function Y_{mn}^{σ} rather than calculating *all* spherical harmonics of a given order m, as in the case of `spharmonic` (chapter 21.1.1). For example, `solospharm(3,-2,pi,0)` returns $Y_{mn}^{\sigma}(\theta, \phi) = Y_{32}^{-1}(\pi, 0)$. The first argument of the function is the m subscript, the second argument is the product $n\sigma$.[49] Multiple functions can be obtained by using vectors for the second argument, as in `solospharm(3,-2:0,0,0)`. An additional input argument can be used to specify the normalization of the spherical harmonic function. `solospharm` is useful in all situations in which isolated spherical harmonics are used, e.g. in designing loudspeaker density or radius functions (chapter 18.6) and plotting them. However, the function is inefficient, since it calls `spharmonic` internally, and simply dismisses the spherical harmonics not required. This is due to the fact that `spharmonic` uses Matlab's native `legendre`, which does not return isolated functions either. Rather than rewriting `legendre` accordingly, it has been decided to accept an inefficiency in a helper function. The more critical application of spherical harmonics regarding efficiency is the encoding of Ambisonic soundfields (chapter 21.3.1), where we always require *all* functions of a given order anyway.

21.5.2 handlespharm

This function returns handles to combinations of spherical harmonic functions, which are specified as `Y` in the input string of the function. For example

```
Y = handlespharm('2*Y(1,-1) + abs(Y(6,5)) - 1')
ezspherical(Y); view(150,15);
```

returns a handle to a function $2 \cdot Y_{11}^{-1} \cdot \left| Y_{65}^{+1} \right| - 1$. The first argument of the `Y` expressions is the m subscript, the second argument is the product $n\sigma$. Internally, `handlespharm` translates the input string into function calls of `solospharm` and native Matlab functions like `abs`. Schmidt-seminormalization is applied to the spherical harmonic functions. The output of this function can be plotted directly using `ezspherical`. Together, these two functions can be used for straightforward design of loudspeaker density and radius functions (see chapter 18.6), which can then be applied to the loudspeaker layout design process using the functions `geosphere`

[49] Note that Matlab's legendre uses the subscripts the other way around.

(chapter 21.2.3) and minenergyconf (chapter 21.2.4). However, due to the inefficiency inherent to the function solospharm (chapter 21.5.1), the such designed functions should be hardcoded in the actual refinement process.

21.5.3 cart3sph, sph3cart, deg2rad, rad2deg

These functions work in exactly the same way as Matlab's native functions cart2sph, sph2cart, i.e. they convert among cartesian and spherical coordinates. However, the input argument to 3LD's cart3sph and the output argument of sph3cart are single N-by-3 array containing the x, y, z components of N points. This differs from Matlab's functions, which require three separate arrays. Thus, cart3sph and sph3cart are convenient for direct conversion from and to cartesian coordinates, for example to derive the loudspeaker directions required by vbp (chapter 21.3.4) from a vertices matrix returned by platonicsolid (chapter 21.2.1). Another application is the editing of loudspeaker radii, demonstrated here at the example of an octahedron layout:

```
p = platonicsolid('oct');
p.vertices
[az,elev,r] = cart3sph(p.vertices);
r = 3.14;
p.vertices = sph3cart(az,elev,r);
p.vertices
```

The functions rad2deg, deg2rad can be used to convert an angle array of arbitrary size from radians to degrees and vice versa. For example, to get the loudspeaker directions from the example before in degrees, type

```
az = rad2deg(az), elev = rad2deg(elev)
```

21.5.4 plot3LD

This is 3LD's plotter function. It can be used to plot loudspeaker layouts given as vertices/faces structures, as returned by functions platonicsolid, geosphere and bucky2, and handles to spherical functions, as provided by handlespharm. The function knows which plotting routine to apply from the appearance of its input argument. For layouts, Matlab's patch is used, and 3LD's ezspherical (which uses Matlab's surf) for spherical functions. Loudspeaker layout plots include the indices of the loudspeakers, which appear black for unlocked and red for locked speakers if their lock status is provided as an additional input argument. Other optimizations regarding the appearance of the plots are included as well. Several application examples of plot3LD have been given throughout this chapter.

22 AlloSphere Scenario

In this final chapter of the thesis, we will briefly develop a scenario for a periphonic sound spatialization engine in the AlloSphere. This will serve as an example for a useful application of the 3LD Library for Loudspeaker Layout Design (chapter 21), following the extended loudspeaker layout design strategy presented in chapter 18.6. However, it has to be pointed out that the actual design phase for the AlloSphere loudspeaker layout will have to involve much more extensive and detailed simulations than we can provide here. Nevertheless, the presented layout can serve as a useful starting point for such a process.

22.1 Loudspeaker Layout

Regarding the amount of loudspeakers in the AlloSphere, numbers between 80 and 512 have been discussed. For our hypothetical scenario, we have aimed at a layout of approximately 300 loudspeakers. Following our proposals from chapter 18.6, we have decided to first find an initial layout with homogeneous loudspeaker distribution, and to psychoacoustically refine this layout in a second step. The high number of required loudspeakers suggests to use a geodesic sphere (chapter 18.3) as the initial layout. The Platonic solids (chapter 18.2.1) serve as a good starting point for the design of geodesic spheres regarding regularity. Thus, various tessellations of the five Platonic solids have been experimented with, resulting in five different layouts with approximately 300 loudspeakers, which are described in table 4. In this table, we list the initial Platonic solid which the geodesic sphere has been built from, the frequencies of the iteratively applied tessellations, the number of loudspeakers in the resulting layout, and the condition number of its Ambisonic re-encoding matrix (chapter 18.1.2) for an Ambisonic order of $M = 14$, which can be decoded by all five layouts so that the $L \geq N$ criterion is fullfilled (eq. 24).

Initial Polyhedron	Frequencies	Loudspeakers	Condition Number
Icosahedron	2, 3	362	1.27
Octahedron	3, 3	326	2.25
Cube	2, 2, 2	386	3.60
Dodecahedron	1, 3	272	1.80
Dodecahedron	1, 2, 2	482	1.28

Table 4: Various initial layouts for an AlloSphere scenario

Note that all of the geodesic spheres provide a remarkably low condition number, which confirms that this approach is suitable for the design of Higher Order Ambisonic loudspeaker layouts. We have decided to choose the geodesic sphere built from the icosahedron as the initial layout, since it provides the lowest condition number as well as advantageous properties regarding loudspeaker distribution in the horizontal plane. With 362 loudspeakers (and 720 triangular facets), Ambisonic systems up to 18th order can be implemented on this system according to the $L \geq N$ criterion (eq. 24). After a rotation of the layout according to our requirements, it has been refined with a minimal energy algorithm using the following loudspeaker density function:

$$d(\theta, \phi) = \frac{1}{3 + |\sin \phi|} + |\cos \theta \cdot \cos \phi|$$

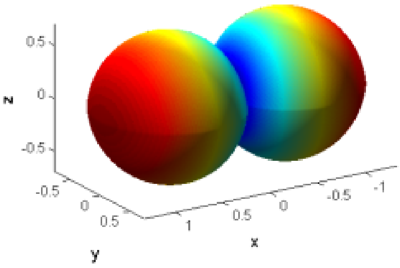


Figure 37: Loudspeaker density function in the AlloSphere scenario

The function is plotted in figure 37. Its first term resembles the higher resolution of the human ear in the horizontal plane, while the second term has been added in order to account for the architectural circumstances of the AlloSphere: the maximum density is given in the front and back directions of the AlloSphere, i.e. in the direction of the positive and negative x axis. It decreases with the absolute value of the elevation, but also towards the seam between the front and back hemisphere, which is a critical area for loudspeaker mounting and also not a prefered viewing direction, since the video projectors will be mounted in this area.

A more precise design of the density function could include an additional top-bottom asymmetry, so that lower loudspeaker densities are given in the lower hemisphere, resembling the spatial resolution of the human ear in more detail. However, it is recommended not to introduce a front-back asymmetry, since the AlloSphere provides two equally important directions (positive and negative x direction) rather than a 'front' and 'back' direction. The density function, which was designed using 3LD's `handlespharm` and `ezspherical` (chapter 21), has been defined as a hardcoded function handle in the actual simulation, in order not to degrade the performance due to the inefficiency of `handlespharm` (chapter 21.5.2).

The layout has been refined in 100 iterations using 3LD's `minenergyconf` (chapter 21.2.4). Note that in order not to degrade the stability of the refinement process, the loudspeaker positions have been mapped to the actual radius of the AlloSphere after rather than during the refinement process (chapter 18.6). The radius of the AlloSphere as a function of direction is given as (see appendix 34 for a derivation)

$$r(\theta, \phi) = \frac{2.44\,|\cos\theta\cos\phi| + \sqrt{5.95\,(\cos^2\theta\cos^2\phi - 1) + 95.26}}{2}$$

No loudspeakers have been locked during the refinement process, but the loudspeakers initially located in the horizontal plane have been set back to zero elevation afterwards. Note that the ear plane in the AllosSphere is actually located somewhat above the equator, which would have to be considered in an extended simulation.

The refined and radius mapped loudspeaker layout is shown in figure 38. The results of the refinement process can be clearly seen in the side view, but especially in the top view of the layout, which reveals a minimum of the loudspeaker density in those areas. In a more sophisticated AlloSphere model, the density in the area of the seam between the two hemispheres will propbably have to be additionally decreased. For instance, the layout still features an unfortunate loudspeaker in the entrance area to the bridge. In figure 39, the condition numbers of the initial layout and the final layout are plotted as functions of the Ambisonic order. It can be seen that for the refined layout, the Ambisonic re-encoding matrix remains quite well-conditioned also for higher orders. For $M = 18$, which is the maximum order of the layout according to the $L \geq N$ criterion, the condition number becomes considerably higher (greater than 10k) for both, the initial and the final layout.[50]. Although the different ranges of the y axis in the plots have to be pointed out, the layout refinement obviously does not dramatically degrade the quality of the layout regarding Ambisonic soundfield reproduction. Note however, that the condition number is a criterion which knows nothing about the radii of the loudspeakers, i.e. the irregularites introduced by the process of radius mapping are not represented.

22.2 Soundfield Simulations

A soundfield reconstruction simulation has been carried out on the refined loudspeaker layout presented in the last chapter, using different spatialization techniques. As a source signal, we have chosen a hanning-weighted impulse with a base frequency of 200 Hz and a top frequency of 1400 Hz.[51] Spherical wavefront propagation has been assumed for the source, and its position has been chosen with $\theta = 57°$ and $\phi = 29°$. Note that no loudspeaker is located at this position. The loudspeakers in the triple responsible for the Vector Base Panning reconstruction of a source at this position are located at $\theta_1 = 61.3°, \phi_1 = 26.9°$; $\theta_2 = 48.6°, \phi_2 = 25.8°$, and $\theta_3 = 51.3°, \phi_3 = 36.2°$. The loudspeaker density in the area of the source position is comparably low.

Besides a simulation of the original soundfield produced by this source, we have considered a virtual source reproduction by the means of a Vector Base (Amplitude) Panning system and two Higher Order Ambisonic engines, one at sixth, and one at seventeenth order. For both Ambisonic systems, a basic decoder (chapter 15.5) has been used, with a decoder matrix derived by the projection method (chapter 15.4). Both Ambisonic soundfields have been decoded to *all* loudspeakers of the layout. The loudspeaker gains derived from the spatialization engines have been calibrated according to the procedure described in chapter 18.5 in order to compensate for the varying loudspeaker distances. Therefore, virtual sources are assumed to move on the surface of the layout's circumscribed sphere. For the AlloSphere, this yields a virtual source radius of $r = 6.1$ m. It has been assumed in the simulation that the loudspeakers emit spherical wavefronts.

The results of the simulation are shown in figures 40 to 42. The real parts of the original and the reconstructed complex sound pressure fields are shown in figure 40. Figure 41 shows the sound pressure amplitude errors of the synthesized fields, and figure 42 their squared sound pressure errors (chapter 17.2). The plotted area resembles the dimensions of the bridge in the center of the AlloSphere (top view), minus one meter at the left and right ends, i.e in the area of the entrances. Note that all three spatialization engines provide accurate soundfield reproduction in the center of the listening area, and how the sweet area increases with the order of an

[50] The according value is not plotted in figure 39.

[51] A Hanning-weighted impulse consists of several monochromatic sources with linearly spaced frequencies (200 Hz, 400 Hz, 600 Hz, etc.) and amplitudes weighted with the right half of a hanning window, i.e. decaying for increasing frequency. See chapter 21.4.1 for an example of a Hanning-weighted impulse built in Matlab.

Ambisonic system. The wavefronts of the three active loudspeakers in the VBP reproduction can be clearly distinguished in the top left corner of the VBP plot in figure 40. In the same figure, the plot of the sixth order Ambisonic system reveals that Ambisonic uses more loudspeakers than Vector Base Panning in the reproduction of a static source.

Additional informal simulations have yielded that the pseudoinverse Ambisonic decoding method seems to provide less accurate soundfield reproduction for high Ambisonic orders (e.g. 17) than the projection method, which can be interpreted as a result of the irregularities introduced by the geodesic extension and psychoacoustical refinement of the initial icosahedron layout. Also, the effects of layout calibration, i.e. artefacts in the reproduction of the wavefront curvature, have been observed.

22.3 Conclusion

It has been shown in this chapter that the 3LD Library for Loudspeaker Layout Design (chapter 21) can be usefully applied in the design of a periphonic sound spatialization engine for the AlloSphere. However, the scenario developed in the course of this chapter can only serve as a starting point for more detailed simulations. The question of how many loudspeaker are actually required for appropriate soundfield reproduction in the AlloSphere must be answered. Different loudspeaker layouts need to be evaluated and compared. Various Ambisonic systems should be evaluated in order to determine the required system order as well as a suitable decoding method and a prefered decoder flavor. Vector Base Amplitude and Intensity Panning systems have to be simulated with regards to their application in the AlloSphere, possibly combined in a hybrid system for optimizing soundfield reconstruction over the entire frequency range. Also, different 3D holophonic approaches could be tested, which is particularly relevant regarding the required number of loudspeakers. More sophisticated loudspeaker density functions can be defined, which consider the properties of spatial hearing and the architectural circumstances of the AlloSphere in more detail. The theory and the practical contributions presented in this thesis should facilitate both, the design as well as the use of the AlloSphere's periphonic sound spatialization system.

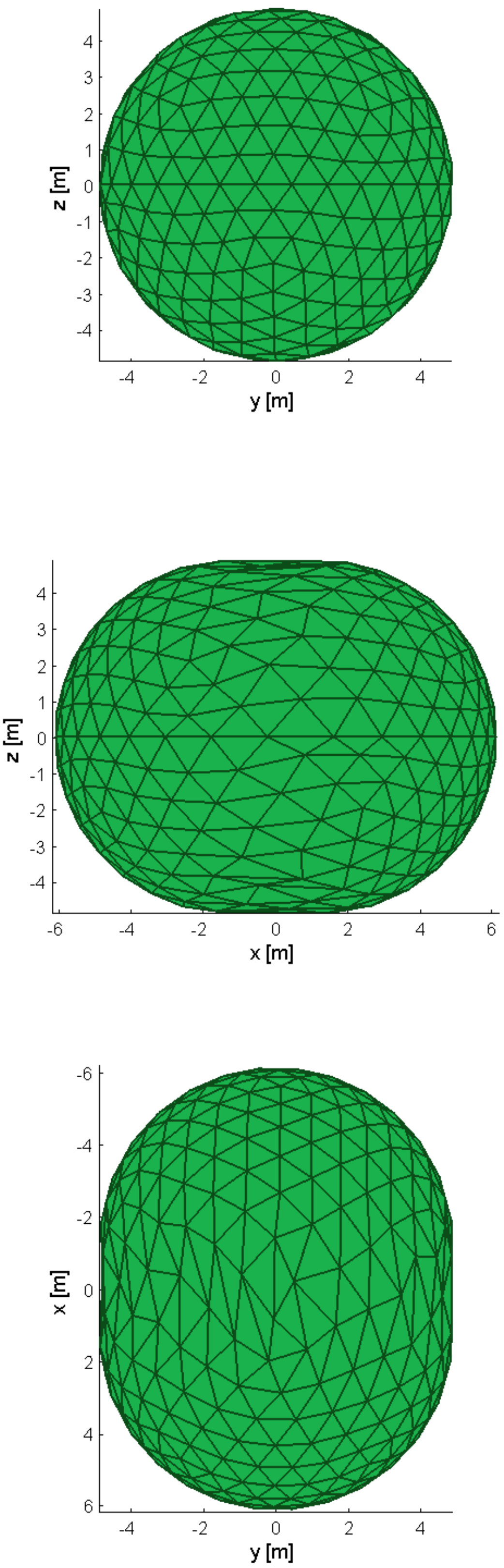


Figure 38: Loudspeaker layout in the AlloSphere scenario: front, side, and top view

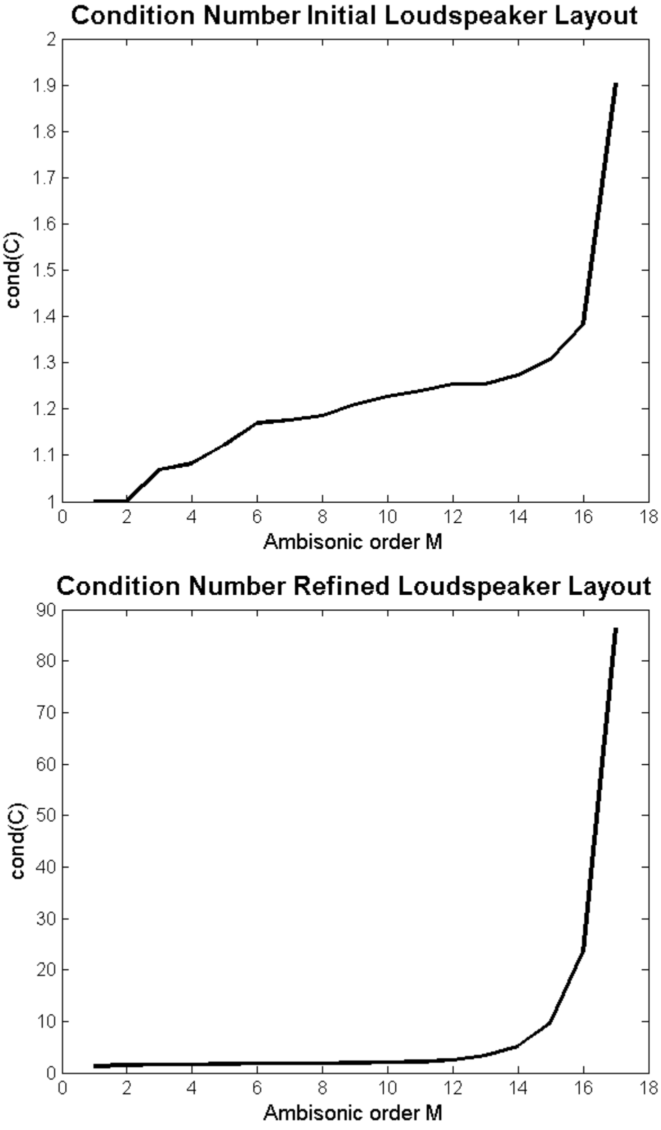


Figure 39: Condition numbers of the initial and final loudspeaker layout

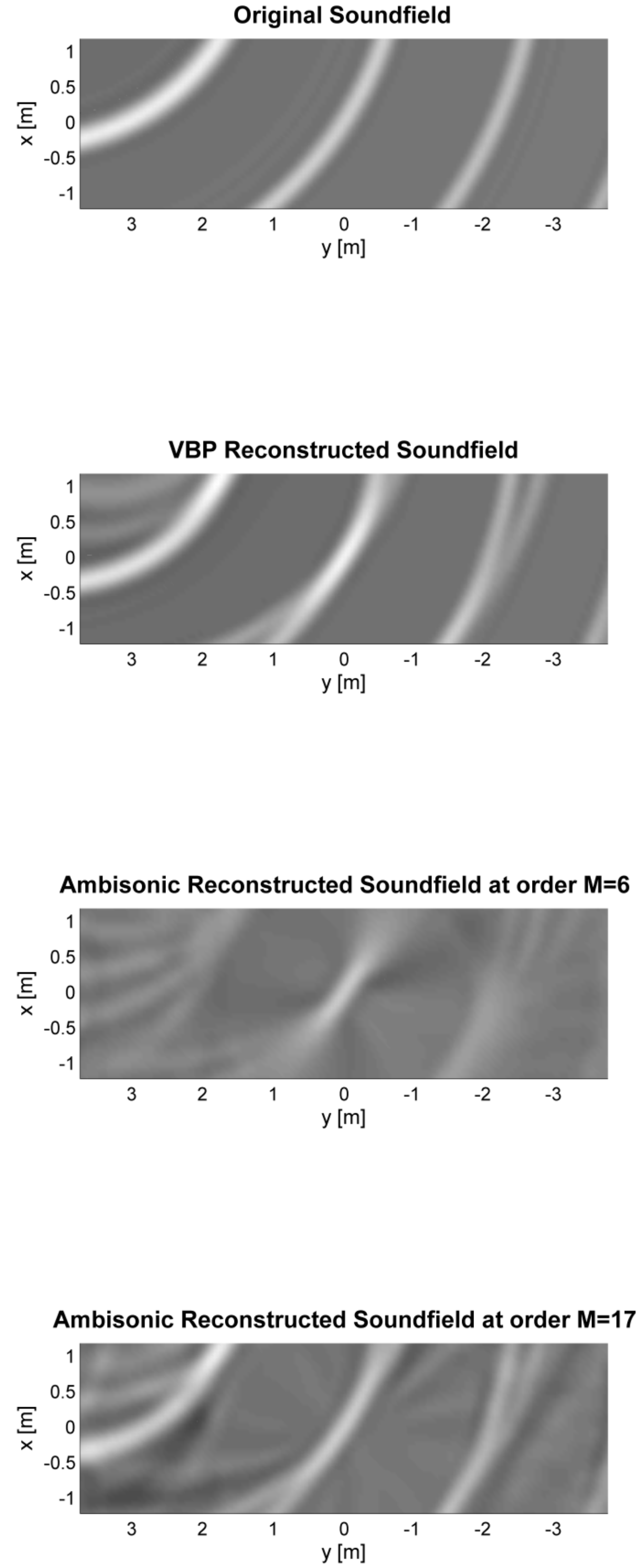
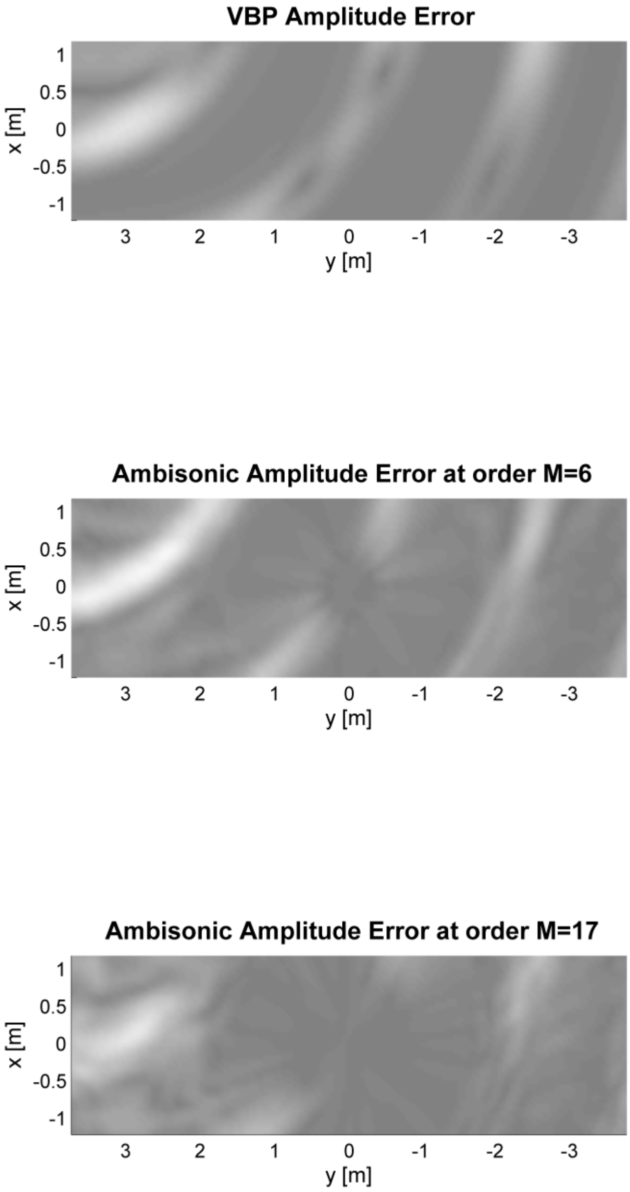


Figure 40: AlloSphere scenario: original and reconstructed soundfields

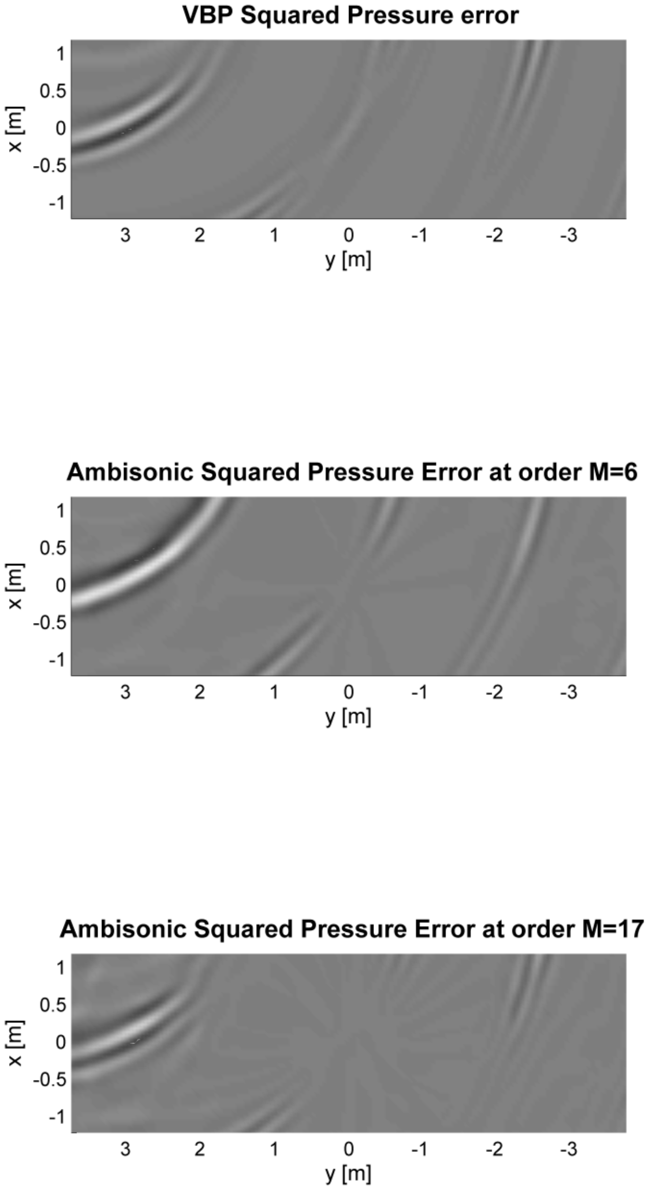


VBP Amplitude Error

Ambisonic Amplitude Error at order M=6

Ambisonic Amplitude Error at order M=17

Figure 41: AlloSphere scenario: sound pressure amplitude errors of the reconstructed fields



VBP Squared Pressure error

Ambisonic Squared Pressure Error at order M=6

Ambisonic Squared Pressure Error at order M=17

Figure 42: AlloSphere scenario: squared sound pressure errors of the reconstructed fields

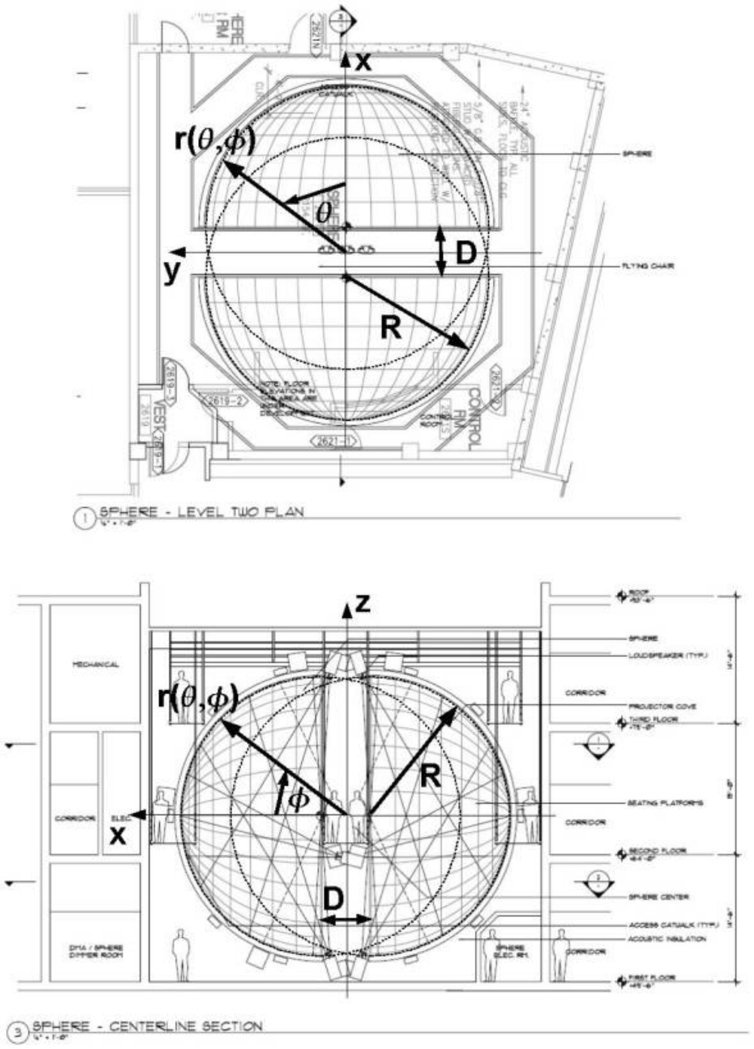


Figure 43: Geometry of the AlloSphere

Part V

Appendix

A AlloSphere Geometry

In this chapter, we will derive the most important geometrical properties of the AlloSphere, i.e. its radius as a function of direction, its circumference in the horizontal plane, and the size of its surface. The radius R of the two hemispheres and the width D between their centers have been assumed with

$$\begin{aligned} R = 16\text{ ft} &= 4.88\text{ m} \\ D = 8\text{ ft} &= 2.44\text{ m} \end{aligned}$$

Note that the distance D is not exactly identical with the width of the bridge in the center of the AlloSphere, which is approximately 7 feet.

A.1 Radius of the AlloSphere

In the area of the centered bridge in the AlloSphere, the curvature of the two hemispheres actually continues, and the radius of the AlloSphere can thus be described as in figure 43. Will Wolcott has contributed the following derivation of the AlloSphere's radius as a funtion of azimuth and elevation:

A simple pythagorean equation yields

$$R^2 = \left(|x| - \frac{D}{2}\right)^2 + y^2 + z^2$$

By converting from cartesian to spherical coordinates we get

$$R^2 = \left(|r\cos\theta\cos\phi| - \frac{D}{2}\right)^2 + r^2\sin^2\theta\cos^2\phi + r^2\sin^2\phi$$

$$R^2 = r^2\cos^2\theta\cos^2\phi - rD\,|\cos\theta\cos\phi| + \frac{D^2}{4} + r^2\sin^2\theta\cos^2\phi + r^2\sin^2\phi$$

$$R^2 = r^2\cos^2\phi\left(\cos^2\theta + \sin^2\theta\right) - rD\,|\cos\theta\cos\phi| + \frac{D^2}{4} + r^2\sin^2\phi$$

$$R^2 = r^2\cos^2\phi - rD\,|\cos\theta\cos\phi| + \frac{D^2}{4} + r^2\sin^2\phi$$

$$R^2 = r^2\left(\cos^2\phi + \sin^2\phi\right) - rD\,|\cos\theta\cos\phi| + \frac{D^2}{4}$$

$$r^2 - rD\,|\cos\theta\cos\phi| + \frac{D^2 - 4R^2}{4} = 0$$

Solving this quadratic equation gives us

$$r(\theta,\phi) = \frac{D\,|\cos\theta\cos\phi| \pm \sqrt{D^2\left(\cos^2\theta\cos^2\phi - 1\right) + 4R^2}}{2}$$

Only the positive square root term guarantees to give positive and thus useful results. The radius of the AlloSphere as a function of azimuth and elevation is thus given as

$$\boxed{r(\theta,\phi) = \frac{D|\cos\theta\cos\phi| + \sqrt{D^2(\cos^2\theta\cos^2\phi - 1) + 4R^2}}{2}} \qquad (34)$$

It is interesting to note that the term $\cos\theta\cos\phi$ happens to be a spherical harmonic function of first order. Thus, we can also express the function as

$$r(\theta,\phi) = \frac{D\,|Y_{11}^1| + \sqrt{D^2\left[\left(Y_{11}^1\right)^2 - 1\right] + 4R^2}}{2} \qquad (35)$$

It is interesting to have a look at the maximum and minimum values of $r(\theta,\phi)$:

$\mathbf{r_{max}} = r(0,0) = R + D/2 = \mathbf{6.10\ m}$

$\mathbf{r_{min}} = r\left(\pm\frac{\pi}{2},\phi\right) = \frac{\sqrt{4R^2 - D^2}}{2} = \mathbf{4.73\ m}$

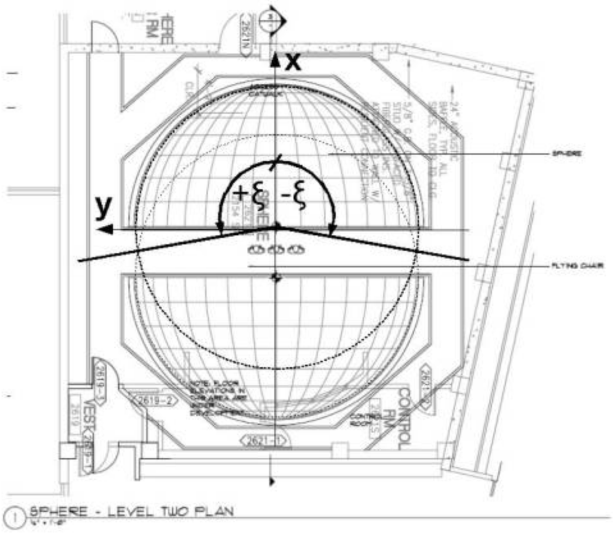


Figure 44: Equator length of the AlloSphere

A.2 Circumference in the Horizontal Plane

Considering the symmetry of the AlloSphere in the x/y plane, we base the derivation of its circumference in the horizontal plane on an integration over the front half of the AlloSphere. For the scope of this calculation, we will move the origin of our coordinate system from the center of the AlloSphere (as in figure 43) the center of its front hemisphere (as in figure 44). Limiting our integration to the front hemisphere and doubling the result will result in the desired value.

The arc length of a curve is given in polar coordinates by

$$c_{arc} = \int_{\alpha}^{\beta} \sqrt{r^2 + (r')^2} \cdot d\theta$$

where r' is the first derivative of r. For a circle, $r = const. \Rightarrow r' = 0$, we get

$$c_{arc} = \int_{\alpha}^{\beta} \sqrt{r^2} \cdot d\theta = \int_{\alpha}^{\beta} r \cdot d\theta \ = r\,(\beta - \alpha)$$

The limits of our integration (see figure 44) are given as $\alpha = -\xi$ and $\beta = +\xi$. It is easy to prove that ξ is

$$\xi = \pi - \arctan\left(\frac{2R}{D}\right) = 104.04^\circ$$

Thus

$$c_{arc} = 2\,r\,\xi = 2r\left[\pi - \arctan\left(\frac{2R}{D}\right)\right]$$

Also, we know that in our case

$$r = R$$

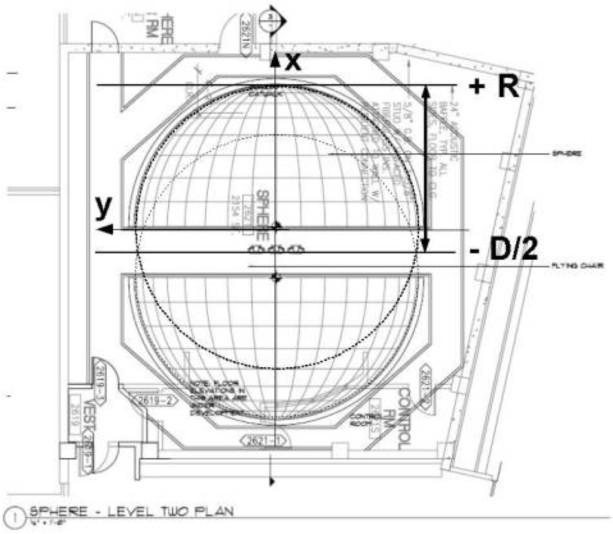


Figure 45: Surface of the AlloSphere

and that we have to double our result to get the full equator length of the AlloSphere

$$c = 2 \cdot c_{arc}$$

$$\boxed{c = 4R\left[\pi - \arctan\left(\frac{2R}{D}\right)\right]} \tag{36}$$

$$\mathbf{c = 35.44 \quad m}$$

A.3 Surface of the AlloSphere

It is possible to derive the size of the AlloSphere's surface by revoluting its circumference in the horizontal plane about the x axis. Again, it is easier to consider only the front hemisphere and double the result. Also, the origin of our coordinate system will again be assumed in the center of the front hemisphere (figure 45).

The general formula for a surface of revolution is given as

$$A_{rev} = 2\pi \int_a^b f(x)\sqrt{1 + [f'(x)]^2} \cdot dx$$

Our function $f(x)$ is given implicitly by the function of a circle

$$x^2 + y^2 - r^2 = 0$$

We use implicit differentiation

$$\frac{dx^2}{dx} + \frac{dy^2}{dx} - \frac{dr^2}{dx} = 0$$

$$2x + 2y\frac{dy}{dx} = 0$$

$$\frac{dy}{dx} = -\frac{x}{y}$$

and obtain

$$A_{rev} = 2\pi \int\limits_a^b y\sqrt{1+\frac{x^2}{y^2}} \cdot dx = 2\pi \int\limits_a^b \sqrt{y^2+x^2} \cdot dx = 2\pi \int\limits_a^b r \cdot dx = 2\pi r\left(b-a\right)$$

The limits of our integration (see figure 45) are given as

$$b = R$$

$$a = -D/2$$

Also, we know that in our case

$$r = R$$

and that we have to double our result to get the full surface of the AlloSphere

$$A = 2 \cdot A_{rev}$$

Thus, the surface of the AlloSphere A equals

$$\boxed{A = 4\pi R\left(R + D/2\right)} \qquad (37)$$

$$\mathbf{A = 374.08\ m^2}$$

B Conventions used in the Literature

This chapter compares different conventions and abbreviations used in literature on sound spatialization (and in the Matlab software package) to describe coordinate systems and Ambisonic representations. The tables in this chapter are intended to facilitate further research in the field of periphonic sound spatialization.

B.1 Coordinate System Conventions

In table 5, we present different conventions used for spherical and cartesian coordinate systems, which determine the appearance of the Ambisonic encoding functions and rotation matrices in different literature. In this table, the last row ('elev. dir.') refers to the direction of increasing elevation, e.g. $xy \rightarrow +z$ if the elevation is zero in the horizontal plane and increases towards the positive z axis. In [Bam95], only 2D systems have been considered.

	this doc.	**[Dan00]**	**[Mal03a]**	**[Son03]**	**[Zmö02]**	**[Bam95]**	**Matlab**
orientation	left	left	left	right	left	left	left
azimuth	θ	θ	θ	θ	φ	ϕ	θ
elevation	ϕ	δ	ϕ	ϕ	ϑ	—	ϕ
elev. dir.	$xy \rightarrow +z$	$xy \rightarrow +z$	$xy \rightarrow +z$	$+z \rightarrow xy$	$xy \rightarrow +z$	—	$xy \rightarrow +z$

Table 5: Coordinate system conventions in spatial audio literature

B.2 Ambisonic Conventions

In table 6, different Ambisonic notations are summarized (Ambisonic order, number of Ambisonic channels and loudspeakers, etc.). The subscripts and superscripts refer to the Y_{mn}^{σ} notation of the spherical harmonic functions used in this thesis. Note that in one exceptional case, [Son03, pp.38-39] uses N to denote the number of loudspeakers rather than the number of Ambisonic channels. In table 7, different conventions for sorting and abbreviating the Ambisonic channels up to third order are presented. The channel sorting convention used by each author is represented by the order of the rows in the table. Each entry lists the channel abbreviation and the spherical harmonic function which is actually refered to by the abbreviation. The abbreviations used for each channel by the specific author is listed as well as the spherical harmonic it ctually refers to, where the Y_{mn}^{σ} notation used in this thesis serves as a reference. Daniel does not use any abbreviations for the third order channels. Malham's third order abbreviations have been used in this thesis while staying coherent with Daniel's channel sorting convention, which lists the two horizontal channels first for each order. [Mal03a] and [Zmö02] use the same abbreviations for the third order channels (and the same order L, M, N, O, P, Q), but actually refer to different third order spherical harmonics with them.

	this doc.	**[Dan00]**	**[Mal03a]**	**[Son03]**	**[Zmö02]**	**[Bam95]**	**Matlab**
Amb. order	M	M	—	M	M	m	—
Amb. ch.	N	K	—	N	L	—	—
speakers	L	N	—	L	N	N	—
1st subscript	m	m	m	n	m	m	n
2nd subscript	n	n	n	m	η	—	m
superscript	σ	σ	ς	σ	σ	—	—

Table 6: Ambisonic conventions in spatial audio literature

C Platonic Solids

In tables 8 through 12, the cartesian coordinates of the five *Platonic solids* are presented. These are the unit solids, i.e. their vertices are located on the unit sphere and thus have a radius of $R = 1$. In these tables, we use χ to denote the *golden mean* with

$$\chi = \frac{\sqrt{5}+1}{2}$$

We also use the abbreviations

$$\tau = \frac{\chi}{\sqrt{1+\chi^2}}$$

$$\zeta = \frac{1}{\sqrt{1+\chi^2}}$$

this document	**[Dan00]**	**[Mal03a]**	**[Zmö02]**
$W = Y_{mn}^{\sigma} = Y_{00}^{1}$			
$X = Y_{11}^{1}$			
$Y = Y_{11}^{-1}$			
$Z = Y_{10}^{1}$			
$U = Y_{22}^{1}$		$R = Y_{20}^{1}$	
$V = Y_{22}^{-1}$		$S = Y_{21}^{1}$	
$S = Y_{21}^{1}$		$T = Y_{21}^{-1}$	
$T = Y_{21}^{-1}$		$U = Y_{22}^{1}$	
$R = Y_{20}^{1}$		$V = Y_{22}^{-1}$	
$P = Y_{33}^{1}$	Y_{33}^{1}	$K = Y_{30}^{1}$	
$Q = Y_{33}^{-1}$	Y_{33}^{-1}	$L = Y_{31}^{1}$	$L = Y_{31}^{-1}$
$N = Y_{32}^{1}$	Y_{32}^{1}	$M = Y_{31}^{-1}$	$M = Y_{31}^{1}$
$O = Y_{32}^{-1}$	Y_{32}^{-1}	$N = Y_{32}^{1}$	$N = Y_{32}^{-1}$
$L = Y_{31}^{1}$	Y_{31}^{1}	$O = Y_{32}^{-1}$	$O = Y_{32}^{1}$
$M = Y_{31}^{-1}$	Y_{31}^{-1}	$P = Y_{33}^{1}$	$P = Y_{33}^{-1}$
$K = Y_{30}^{1}$	Y_{30}^{1}	$Q = Y_{33}^{-1}$	$Q = Y_{33}^{1}$

Table 7: Ambisonic channel conventions in spatial audio literature

x	**y**	**z**
$+\frac{\sqrt{3}}{3}$	$+\frac{\sqrt{3}}{3}$	$+\frac{\sqrt{3}}{3}$
$-\frac{\sqrt{3}}{3}$	$-\frac{\sqrt{3}}{3}$	$+\frac{\sqrt{3}}{3}$
$-\frac{\sqrt{3}}{3}$	$+\frac{\sqrt{3}}{3}$	$-\frac{\sqrt{3}}{3}$
$+\frac{\sqrt{3}}{3}$	$-\frac{\sqrt{3}}{3}$	$-\frac{\sqrt{3}}{3}$

Table 8: The coordinates of the unit tetrahedron

x	**y**	**z**
$+\frac{\sqrt{3}}{3}$	$+\frac{\sqrt{3}}{3}$	$+\frac{\sqrt{3}}{3}$
$+\frac{\sqrt{3}}{3}$	$+\frac{\sqrt{3}}{3}$	$-\frac{\sqrt{3}}{3}$
$+\frac{\sqrt{3}}{3}$	$-\frac{\sqrt{3}}{3}$	$+\frac{\sqrt{3}}{3}$
$+\frac{\sqrt{3}}{3}$	$-\frac{\sqrt{3}}{3}$	$-\frac{\sqrt{3}}{3}$
$-\frac{\sqrt{3}}{3}$	$+\frac{\sqrt{3}}{3}$	$+\frac{\sqrt{3}}{3}$
$-\frac{\sqrt{3}}{3}$	$+\frac{\sqrt{3}}{3}$	$-\frac{\sqrt{3}}{3}$
$-\frac{\sqrt{3}}{3}$	$-\frac{\sqrt{3}}{3}$	$+\frac{\sqrt{3}}{3}$
$-\frac{\sqrt{3}}{3}$	$-\frac{\sqrt{3}}{3}$	$-\frac{\sqrt{3}}{3}$

Table 9: The coordinates of the unit hexahedron

x	**y**	**z**
$+1$	0	0
-1	0	0
0	$+1$	0
0	-1	0
0	0	$+1$
0	0	-1

Table 10: The coordinates of the unit octahedron

x	**y**	**z**
0	$+\frac{1}{\chi}$	$-\chi$
$+1$	$+1$	-1
$+\frac{1}{\chi}$	$+\chi$	0
-1	$+1$	-1
$-\frac{1}{\chi}$	$+\chi$	0
$-\chi$	0	$-1\frac{1}{\chi}$
-1	$+1$	$+1$
$-\chi$	0	$+\frac{1}{\chi}$
-1	-1	$+1$
0	$-\frac{1}{\chi}$	$+\chi$
$+\frac{1}{\chi}$	$-\chi$	0
$+1$	-1	$+1$
$+\chi$	0	$+\frac{1}{\chi}$
$+1$	-1	-1
$+\chi$	0	$-\frac{1}{\chi}$
$-\frac{1}{\chi}$	$-\chi$	0
0	$-\frac{1}{\chi}$	$-\chi$
-1	-1	-1
$+1$	$+1$	$+1$
0	$+\frac{1}{\chi}$	$+\chi$

Table 11: The coordinates of the unit dodecahedron

x	**y**	**z**
$+\tau$	$+\zeta$	0
$-\tau$	$+\zeta$	0
$-\tau$	$-\zeta$	0
$+\tau$	$-\zeta$	0
$+\zeta$	0	$+\tau$
$+\zeta$	0	$-\tau$
$-\zeta$	0	$-\tau$
$-\zeta$	0	$+\tau$
0	$+\tau$	$+\zeta$
0	$-\tau$	$+\zeta$
0	$-\tau$	$-\zeta$
0	$+\tau$	$-\zeta$

Table 12: The coordinates of the unit icosahedron

References

[AIS] Full revision of international standards for equal-loudness level contours (ISO 226). `http://www.aist.go.jp/aist_e/latest_research/2003/20031114/20031114.html`. Online; retrieved in February 2006.

[Bam95] Jeffery Stephen Bamford. An analysis of Ambisonic sound systems of first and second order. Master's thesis, University of Waterloo, Ontario, Canada, 1995.

[Ber88] A.J. Berkhout. A holographic approach to acoustic control. In *AES Journal*, volume 36, pages 977–995. Audio Engineering Society, 1988.

[BK] R.K. Bock and W. Krischer. The data analysis briefbook - pseudoinverse. `http://rkb.home.cern.ch/rkb/AN16pp/node220.html`. Online; retrieved in January 2006.

[Bla74] Jens Blauert. *Räumliches Hören.* S. Hirzel Verlag Stuttgart, first edition, 1974.

[Blu31] Alan Dower Blumlein. British patent specification 394325. `http://www.doramusic.com/patents/394325.htm`, 1931. Online; retrieved in April 2005.

[Boo01] Marinus M. Boone. Acoustic rendering with wave field synthesis. In *Proceedings of the ACM Siggraph and Eurographics Campfire: Acoustic Rendering for Virtual Environments, Snowbird, Utah*, 2001.

[Bos01] Xie Bosun. Signal mixing for a 5.1-channel surround sound system - analysis and experiment. In *AES Journal*, volume 49, no.4, pages 263–274. Audio Engineering Society, 2001.

[Bou] Paul Bourke. Sphere generation. `http://astronomy.swin.edu.au/~pbourke/modelling/sphere/`. Online; retrieved in February 2006.

[BR99] Douglas. S. Brungart and William M. Rabinowitz. Auditory localization of nearby sources - head-related transfer functions. In *ASA Journal*, volume 106, pages 1465–1479. Acoustical Society of America, 1999.

[BSK04] Herbert Buchner, Sascha Spors, and Walter Kellermann. Full-duplex systems for sound field recording and auralization based on Wave Field Synthesis. In *Covention Papers of the 116th Convention of the Audio Engineering Society, Berlin, Germany*, 2004.

[Dan00] Jérôme Daniel. *Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia.* PhD thesis, Université Paris 6, 2000. available at `http://gyronymo.free.fr/audio3D/download_Thesis_PwPt.html#PDFThesis`.

[Dan03] Jérôme Daniel. Spatial sound encoding including near field effect: Introducing distance coding filters and a viable, new Ambisonic format. In *Proceedings of the 23rd International Conference of the Audio Engineering Society, Copenhagen, Denmark*, 2003.

[Der] Manuel Derra. Expo 1970 image gallery. `http://www.antonraubenweiss.com/expo/webimages/`. Online; retrieved in January 2006.

[Dic97] Michael Dickreiter. *Handbuch der Tonstudiotechnik*, volume 1. K. G. Saur Verlag, Munich, Germany, 6th, improved edition, 1997.

[DM03] Jérôme Daniel and Sébastien Moreau. Theory and design refinement of Higher Order Ambisonic microphones - Experiments with a 4th order prototype. In *Proceedings of the 23rd International Conference of the Audio Engineering Society, Copenhagen, Denmark*, 2003.

[DNM03] Jérôme Daniel, Rozenn Nicol, and Sébastien Moreau. Further investigations of Higher Order Ambisonics and Wavefield Synthesis for holophonic sound imaging. In *Covention Papers of the 114th Convention of the Audio Engineering Society, Amsterdam, Netherlands*, 2003.

[DRP99] Jérôme Daniel, Jean-Bernard Rault, and Jean-Dominique Polack. Acoustic properties and perceptive implications of stereophonic phenomena. Corrected version 29/03/99. In *Proceedings of the 16th International Conference of the Audio Engineering Society on Spatial Sound Reproduction*, 1999.

[EMFa] EMF Institute infohub - The Acousmonium. `http://emfinstitute.emf.org/exhibits/acousmonium.html`. Online; retrieved in January 2006.

[EMFb] EMF Institute infohub - BEAST. `http://emfinstitute.emf.org/exhibits/beast.html`. Online; retrieved in January 2006.

[Ger73] Michael A. Gerzon. With-heigth sound reprodution. In *AES Journal*, volume 21, pages 2–10. Audio Engineering Society, 1973.

[Ger74] Michael A. Gerzon. What's wrong with quadrophonics. *Studio Sound Magazine*, 1974. `http://www.audiosignal.co.uk/Resources/What\_is\_wrong\_with\_quadraphonics\_A4.pdf`.

[Ger92] Michael A. Gerzon. General metatheory of auditory localisation. In *Proceedings of the 92th convention of the Audio Engineering Society, Vienna, Austria*, 1992.

[Gra00] Gerhard Graber. *Tontechnik und interdisziplinäres Sinnen. Eine grundlegende Fragestellung.* Peter Lang Europäischer Verlag der Wissenschaften, Frankfurt am Main, Germany, 1st edition, 2000.

[Gro03] Tillmann Gronert. Distanzwahrnehmung bei virtueller Tondarstellung mittels Wellenfeldsynthese. Master's thesis, Technische Fachhochschule Berlin, Germany, 2003.

[Hub02] Thomas Huber. Zur Lokalisation akustischer Objekte bei Wellenfeldsynthese. Master's thesis, Institut für Rundfunktechnik, Munich, Germany, 2002.

[Jes73] M. Jessel. *Acoustique Théorique: Propagation et Holophonie.* Masson, Paris, 1973.

[Ker03] Stefan Kerber. Zur Wahrnehmung virtueller Quellen bei Wellenfeldsynthese. Master's thesis, Technische Universität Munich, Germany, 2003.

[Mal99] David George Malham. Second order Ambisonics - the Furse-Malham set. `http://www.york.ac.uk/inst/mustech/3d_audio/secondor.html`, 1999. Online; retrieved in January 2006.

[Mal01] David George Malham. Spherical harmonic coding of sound objects - the Ambisonic 'O' format. In *Proceedings of the 19th International Conference of the Audio Engineering Society, Schloss Elmau, Germany*, pages 54–57, 2001.

[Mal03a] David George Malham. *Space in Music Music in Space*. PhD thesis, University of York, 2003.

[Mal03b] David George Malham. Spatial elements in music prior to the technological era. Private copy of an unpublished article, 2003.

[MAT] AlloSphere at the California Nanosystems Institute. `http://mat.ucsb.edu/allosphere/`. Online; retrieved in February 2006.

[MAT05] The AlloSphere: An immersive instrument for interactive computing and scientific data exploration, November 2005.

[McC05] Doug McCoy. Ventriloquist: A performance interface for real-time gesture-controlled music spatialization. Master's thesis, University of California at Santa Barbara, 2005.

[Men02] Dylan Menzies. W-panning and O-format, tools for object spatialization. In *Proceedings of the 22nd International Conference of the AES on Virtual Synthetic and Entertainment Audio, Espoo, Finland*, 2002.

[Mit] Doug Mitchell. Concepts and development of multichannel sound. `http://www.mtsu.edu/~dsmitche/rim456/Quad/Quad_Formats.html`. Online; retrieved in January 2006.

[NE98] Rozenn Nicol and Marc Emerit. Reproducing 3D-sound for videoconferencing: a comparison between holophony and Ambisonic. In *Proceedings of the DAFX98, Barcelona, Spain*, 1998.

[NE99] Rozenn Nicol and Marc Emerit. 3D-sound reproduction over an extensive listening area: A hybrid method derived from holophony and Ambisonic. In *Proceedings of the 16th International Conference of the Audio Engineering Society, Rovaniemi, Finland*, page 436453, 1999.

[PBJ98] Jean-Marie Pernaux, Patrick Boussard, and Jean-Marc Jot. Virtual sound source positioning and mixing in 5.1. Implementation on the real-time system genesis. In *Proceedings of the DAFX98, Barcelona, Spain*, 1998.

[PKH99] Ville Pulkki, Matti Karjalainen, and Jyri Huopaniemi. Analyzing virtual sound source attributes using a binaural auditory model. In *AES Journal*, volume 47, no.4, pages 203–217. Audio Engineering Society, 1999.

[PL98] Ville Pulkki and Tapio Lokki. Creating auditory displays to multiple loudspeakers using VBAP: A case study with DIVA project. In *Proceedings of the International Conference on Auditory Displays, Glasgow, Scotland*, 1998.

[Pol00] M.A. Poletti. A unified theory of horizontal holographic sound systems. In *AES Journal*, volume 48, no.12, pages 1155–1182. Audio Engineering Society, 2000.

[Pop05] Stephen Travis Pope. Audio in the UCSB CNSI AlloSphere. Technical report, Graduate Program in Media Arts and Technology, University of California at Santa Barbara, 2005. Online; retrieved in August 2005.

[PR03] Stephen Travis Pope and Chandrasekhar Ramakrishnan. The CREATE Signal Library ('Sizzle'): design, issues, and applications. In *Proceedings of the International Computer Music Conference 2003*, 2003.

[Puc96] Miller Puckette. Pure Data: another integrated computer music environment. In *Proceedings of the Second Intercollege Computer Music Concerts, Tachikawa, Japan*, pages 37–41, 1996.

[Pul97] Ville Pulkki. Virtual sound source positioning using Vector Base Amplitude Panning. In *AES Journal*, volume 45, no. 6, pages 456 – 466. Audio Engineering Society, 1997.

[Pul99] Ville Pulkki. Uniform spreading of amplitude panned virtual sources. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, New York*, page W99.1 W99.4, 1999.

[Pul01a] Ville Pulkki. Localization of amplitude-panned virtual sources II: Two- and three-dimensional panning. In *AES Journal*, volume 49, no. 9, pages 753–767. Audio Engineering Society, 2001.

[Pul01b] Ville Pulkki. *Spatial Sound Generation and Perception by Amplitude Panning Techniques.* PhD thesis, Helsinki University of Technology, 2001.

[R$^+$96] Curtis Roads et al. *The Computer Music Tutorial.* MIT Press, Cambridge, Massachusetts, 1996.

[Rei02] Markus Reisinger. Neue Konzepte der Tondarstellung bei Wiedergabe mittels Wellenfeldsynthese. Master's thesis, Fachhochschule Düsseldorf, 2002.

[RMZH05] Winfried Ritsch, Thomas Musil, Johannes Zmölnig, and Robert Höldrich. 3D Sound-Mixer. Implementation eines kombinierten Ambisonic- und Bus-Mixers für den Einsatz in 3D Audio Environments. IEM Report 28/05, Institute of Electronic Music and Acoustics Graz, 2005.

[SH00a] Alois Sontacchi and Robert Höldrich. Enhanced 3D sound field synthesis and reproduction system by compensating interfering reflections. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00), Verona, Italy*, 2000.

[SH00b] Alois Sontacchi and Robert Höldrich. Konzepte zur Schallfeldsynthese und Schallfeldreproduktion. In *50. Jahrestagung der ÖPG FA-Akustik, Graz, Austria*, 2000.

[SH01] Alois Sontacchi and Robert Höldrich. Further investigations on 3D sound fields using distance coding. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-01), Limerick, Ireland*, 2001.

[SH02a] Alois Sontacchi and Robert Höldrich. Distance coding in 3D sound fields. In *Proceedings of the 21st Conference of the Audio Engineering Society, St. Petersburg, Russia*, 2002.

[SH02b] Alois Sontacchi and Robert Höldrich. Investigations on distance coding in 3D sound fields. In *Proceedings of the 6th International Conference on Signal Processing, Beijing, China*, 2002.

[Son03] Alois Sontacchi. *Dreidimensionale Schallfeldreproduktion fuer Lautsprecher- und Kopfhöreranwendungen.* PhD thesis, Technische Universität Graz, Austria, 2003.

[Sta97] Evert Start. *Direct Sound Enhancement by Wave Field Synthesis.* PhD thesis, Technische Universiteit Delft, 1997.

[Sto71] Karlheinz Stockhausen. *Texte zur Musik 1963-1970*, volume 3. Verlag M.DuMont Schauberg, Köln, Germany, Dieter Schnebel, 1971.

[VL87] John Vanderkooy and Stanley Lipshitz. Anomalies of wavefront reconstruction in stereo and surround-sound reproduction. In *Proceedings of the 83rd Convention of the Audio Engineering Society*, 1987.

[WA01] B. Ward and T.D. Abhayapala. Reproduction of a plane-wave sound field using an array of loudspeakers. In *IEEE Transactions on Speech and Audio Processing*, volume 9, no. 6. IEEE, 2001.

[Wika] Wikipedia. Archimedian solid - Wikipedia, The Free Encyclopedia. `http://en.wikipedia.org/wiki/Archimedean_solid`. Online; retrieved in February 2006.

[Wikb] Wikipedia. Catalan solid - Wikipedia, The Free Encyclopedia. `http://en.wikipedia.org/wiki/Catalan_solid`. Online; retrieved in February 2006.

[Wikc] Wikipedia. Johnson solid - Wikipedia, The Free Encyclopedia. `http://en.wikipedia.org/wiki/Johnson_solid`. Online; retrieved in February 2006.

[Wikd] Wikipedia. Platonic solid - Wikipedia, The Free Encyclopedia. `http://en.wikipedia.org/wiki/Platonic_solid`. Online; retrieved in February 2006.

[Wike] Wikipedia. Polyhedron - Wikipedia, The Free Encyclopedia. `http://en.wikipedia.org/wiki/Polyhedron`. Online; retrieved in February 2006.

[Wit02] Helmut Wittek. OPSI - Optimised Phantom Source Imaging of the high frequency content of virtual sources in Wave Field Synthesis. a hybrid WFS/phantom source solution to avoid spatial aliasing, 2002. available at `http://www.irt.de/wittek/hauptmikrofon/OPSI_Paper.pdf`.

[Zmö] Johannes Zmölnig. Der IEM CUBE. Eine Dokumentation. `http://iem.at/services/studios/cube/hardware/`. Online; retrieved in January 2006.

[Zmö02] Johannes Zmölnig. Entwurf und Implementierung einer Mehrkanal-Beschallungsanlage. Master's thesis, University of Music and Dramatic Arts Graz, Austria, 2002.