A framework for dynamic spatial acoustic scene generation with Ambisonics in low delay realtime

Giso GRIMM
Medical Physics Group
Universität Oldenburg
26111 Oldenburg,
Germany
g.grimm@uni-oldenburg.de

Tobias HERZKE
HörTech gGmbH
Marie-Curie-Str. 2
26129 Oldenburg,
Germany,
t.herzke@hoertech.de

Abstract

A software toolbox developed for a concert in which acoustic instruments are amplified and spatially processed in low-latency is described in this article. The spatial image is created in Ambisonics format by a set of dynamic acoustic scene generators which can create periodic spatial trajectories. Parameterization of the trajectories can be selected by a body tracking interface optimized for seated musicians. Application of this toolbox in the field of hearing research is discussed.

Keywords

Ambisonics, digital processing, concert performance

1 Introduction

Spatially distributed presentation of music can contribute to an improved perception. In the 16th century, many composers used several choirs and groups of musicians at distributed places in church music. However, most conventional concert locations do not provide the capabilities to have the musicians placed around the audience. Spatial presentation with loudspeakers offers new possibilities.

The ensemble ORLANDO*viols*[1] developed and performed a concert program, in which the audience is surrounded by the five musicians. Additionally, the sound is processed by a low-latency Ambisonics system and presented in dynamic spatial configurations. The instruments virtually move around the audience. Trajectories matched to the music potentially provide easier access to the concepts in music and may add further levels of interpretation. A central piece in the program is an "In Nomine" fantasy from the 16th century by Mr. Picforth, in which each voice is related to one of the planets known in that time [Grimm, 2012]. For example, during this piece the musicians virtually follow the planet's trajectories.

Such a concert with simultaneous presentation of acoustic sources and their spatially processed images requires careful considerations of psychoacoustic effects, specifically the precedence effect. Appropriate amplification and placement of loudspeakers and musicians are a prerequisite for a spatial image which is dominated by the processed sound and not by the direct sound of the musicians [Grimm et al., 2011]. A set of software components is required to make such a concert possible. The Open Sound Control (OSC) protocol [Wright, 2002] facilitates communication of these components across the borders of processes and hardware machines. The several software components and their applications are described in the following sections.

2 Application in a concert setup

The tools described in this paper have been developed for a concert performance entitled "Harmony of the Spheres". The ensemble consists of five musicians playing on viola da gamba (or viols), a historic bowed string instrument. Viola da gambas come as a family - the smallest has the size of a violin, the largest that of the double bass. The instrument – except for the largest one – is held with the legs, which led to its name ('gamba' means leg in Italian).

In the concert, five musicians (including one of the authors) sit on the corners of a large pentagon. Within the pentagon a 10 channel 3rd order horizontal Ambisonics system is placed on a regular decagon with a radius of approximately 6 meters. Within this decagon sits the audience, with space for roughly 100 listeners. The instruments are picked up with cardioid microphones

[1] `http://www.orlandoviols.de/`

at a distance of approximately 40 cm, which is a compromise between close distance for minimal feedback problems and a large distance for a balanced sound of the historic acoustic instruments. The microphone signals are converted to the digital domain (Behringer ADA8000) and processed on a Linux PC (Athlon X2 250). The processed signals are then played back by the powered near field monitors (KRK RP5). Due to the large distance between the musicians – up to 15 meters – monitoring is required. The monitor signals are created on the hardware mixer of the sound card (RME HDSP9652) and played back to the musicians through in-ear monitor headphones. Two hardware controllers (Behringer BCF2000) are used to control the loudspeaker mix and the monitor. A separate PC is used for visualization of levels, monitor mixer settings and the spatial parameters. A foot switch (Behringer FBC1010) and a virtual foot switch (Microsoft Kinect connected to a netbook PC) are used for parameter control by one of the musicians. An additional netbook PC is used for projection of the trajectories of the virtual sources at the ceiling of the room.

Figure 1: Concert setup in a church, during rehearsal. The diameter of the loudspeaker ring was 13 m, fitting about 100 chairs.

3 Software components

The software components used in the toolbox for dynamic scene generation can be divided into a set of third party software, and specifically developed applications. The third party software includes the jack audio connection kit (jack) [Letz and Davis, 2011]. Jack2 is used to allow for parallel processing of independent signal graphs. The digital audio workstation 'ardour' [Davis, 2011] is used for signal routing, mixing, recording and playback. Convolution reverb is used for distance coding; the convolution is done by 'jconvolver' [Adriaensen, 2011]. The Ambisonics signals are decoded into the speaker layout with 'ambdec' [Adriaensen, 2011]. The tool 'mididings' [Sacré, 2010] triggers the execution of shell scripts upon incoming MIDI events.

3.1 Spatial processing: 'sphere' panning application

For the spatial processing a stand-alone jack panning application with an integrated scene generator is used. The scene generator is inspired by the planet movements. It can create Kepler ellipses with an additional epicyclic component and a random phase increment. The parameters are radius ρ, rotation frequency f, eccentricity ε, rotation of ellipse main axis θ, starting angle φ_0, and the epicycle parameters radius ρ_{epi}, rotation frequency f_{epi} and starting angle $\varphi_{0,epi}$. The source position on a complex plane is

$$z = \frac{\rho\sqrt{1-\varepsilon^2}}{1-\varepsilon\cos(\varphi-\theta)}e^{i\varphi} + \rho_{epi}e^{i\varphi_{epi}}. \qquad (1)$$

The azimuth $\angle z$ is the input argument of a 3rd order horizontal ambisonics panner. Distance perception in reverberant environments is dominated by the amount of reverberation [Sheeline, 1982; Bosi, 1990]. The distance $|z|$ is here coded by adding a reverberant copy of the signal. The reverberant signal is created by a convolution stereo-reverb. A virtual stereo loudspeaker pair centered around the target direction given by the virtual source azimuth is used for playback. The width of the virtual stereo pair is controlled by the distance. It is chosen to be maximal at the critical distance ρ_{crit}, and converging to zero for close and distant sources. Also the ratio between the dry signal and the reverberant stereo signal is controlled by the distance. The distance to the origin normalized by the critical distance is $\hat{\rho} = \rho/\rho_{crit}$. Then the parameters of the distance coding are:

$$w(\rho) = w_{max}\frac{\hat{\rho}}{\hat{\rho}^2+1} \qquad (2)$$

$$G_{dry} = \frac{1}{1+\hat{\rho}} \qquad (3)$$

$$G_{wet} = 1 - G_{dry} \quad (4)$$

The maximal width w_{max} and the critical distance ρ_{crit} can be controlled externally. Examples of trajectories used in the concert are shown in Figure 2.

The scene generator and panning application, named 'sphere', is a jack application with an input port for the dry signal and a stereo input port pair for the reverberated signal. The parameters can be controlled via OSC messages. The OSC receiver can be configured to listen at a multicast address, which allows to control multiple instances in a single OSC message. To achieve a dynamic behavior independent from the actual processing block size, the update rate of panning parameters is independent from the block rate. Changes of different parameters can be accumulated and applied simultaneously, either immediately or slowly within a given time.

Other features of the scene generator are sending of the current azimuth to other scene generator instances via OSC, application of parameter changes at predefined azimuths, randomization of the azimuth, and level-dependent azimuth.

The ambisonics panner uses horizontal panning only. For artistic reasons, a simulation of elevation is reached by gradually mixing the signal to the zeroth order component of the ambisonics signal. By doing so the resulting ambisonics signal does not correspond to any physical acoustic sound field anymore, however, the intention is to have the possibility of creating a virtual sound source without any distinct direction.

3.2 Matrix mixer

Creating an individual headphone monitor mix for several musicians and many sources is a demanding task, and usually needed in situations when a quick and efficient solution is required. In this setup, the RME HDSP9652 sound card was used, which offers a hardware based matrix mixer. The existing interfaces on Linux - amixer (console) and hdspmixer (graphical) - provide full access to that mixer. However, the lack of hardware based remote control option makes these tools inefficient in time-critical situations. Therefore, a set of applications has been developed which attempts to provide a flexible solution: One application provides an interface to the audio hardware. A second application reads and stores the mixer settings from and to XML files. Mixer settings include panning – sin-cos-panning for two-channel output, vector-based amplitude panning (VBAP) for multi-channel output – and gains for a pre-defined set of inputs and outputs. Channels which are not used in a configuration will not be accessed. No logical difference is made between software and hardware inputs. A third application allows control via MIDI CC events (here coming from a Behringer BCF2000). Feedback is sent to the controller device. A fourth application is visualizing the gains. Editing of the gains is planned, but not implemented yet at time of writing. A screen-shot of the visualization tool is

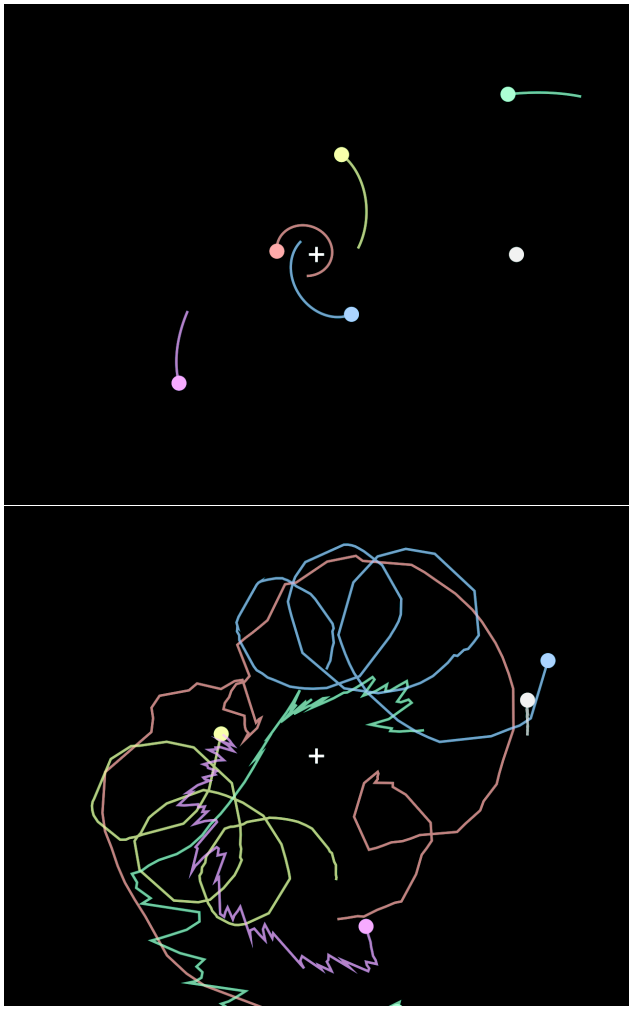

Figure 2: Example trajectories in two of the pieces: Kepler ellipses in the "In Nomine" by Mr. Picforth (top), and chaotic movements in the piece "Five" by John Cage (bottom).

shown in Fig. 3. The four applications share their settings via OSC messages. A feedback filter prevents from OSC message feedback.

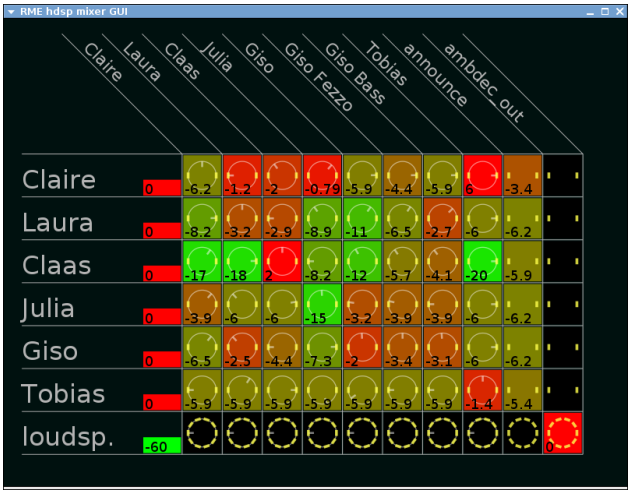

Figure 3: Screen shot of a matrix mixer example session.

3.3 Level metering

For the polyphonic character of many of the pieces played in the above mentioned concert, well balanced levels of the five instruments are of major importance. Level differences between the instruments of only a few dB can easily destroy the musical structure in some pieces. Space limitations make it impossible to place the mixing desk and its operator within the Ambisonics system. Therefore, a level metering application was developed. This level meter can measure the root-mean-square (RMS) level and short time peak values of multiple inputs. To account for the loudness difference between low instruments (double bass size) and medium and high instruments, a band pass filter with a pass band from 300 Hz to 3 kHz is applied before metering. The RMS level meter operate on 10 s and 125 ms rectangular windows. The 125 ms window is overlapping by 50%. The peak levels are measured in the same windows as the short time levels. The levels are sent via OSC messages to a visualization application. The level meter application was implemented in the HörTech Master Hearing Aid (MHA) [Grimm et al., 2006].

To allow for optimal mixing even without hearing the balanced final mix at the sweet spot, an indicator is used which shows the long term target level. This level can be changed by the presets for each piece and also within pieces. Long term level deviations can then be corrected by the operator. Automatic adaption to the desired levels bares the risk of causing feedback problems.

3.4 Parameter exchange and preset selection

All applications exchange parameters and control data via OSC messages. To allow a flexible inter process communication across hardware borders, most applications subscribe themselves to UDP multicast addresses.

Settings of the scene generators 'sphere' can be loaded from setting files. Reading of preset files and sending them as OSC messages can be triggered either from the console, from the virtual footswitch controller, from the physical footswitch or by ardour markers. For the last option, an application has been developed which converts ardour markers into OSC messages: The application reads an ardour session file, subscribes to jack, and emits OSC messages whenever an ardour marker position falls into the time interval of the current jack audio block.

The several preset files can also control the transport of ardour, to control recording of inputs and trigger playback of additional sounds.

3.5 Visualization

Simple visualization classes have been implemented for display of the trajectories of the 'sphere' scene generators, for level monitoring and for visual control of the monitor mixer. With these classes applications can be built to contain any combination of visualization tools. No graphical user interaction is possible.

3.6 Implementation of virtual foot switch

In the concert application, one of the musicians is responsible for switching scene generation presets at musically appropriate times while playing. Since viola da gambas are played while sitting on a chair, and the instruments are held between the legs, it is sometimes difficult to use a real footswitch, especially elevating the heel, without interrupting the music. To circumvent these problems, a virtual footswitch is used which can be controlled by a minor movement of the tip of the foot, without the need of elevating the heel. Care

was taken that this footswitch is sufficiently robust to avoid false alarms and missed movements.

The virtual foot switch is implemented using the Prime Sense depth camera of the Microsoft Kinect game accessory [Kin, 2010]. The depth camera captures the area of the right foot of the musician in charge of the switching. The switch differentiates between 4 states: No foot present ("free"), and 3 different directions of the foot ("straight", "left" and "right"). The switch is inactive in the states "free" and "straight", and triggers actions in the states "left" and "right".

The depth camera uses an infrared laser diode to illuminate the room with an infrared dot pattern, which is then picked up by the camera sensor. From the distortion of the dot pattern, the camera is able to associate image pixels with depth information. The computation of depth information is performed inside the camera device. Because the infrared laser diode and the camera sensor are laterally separated, not the complete scene visible to the camera sensor is illuminated with the infrared dot pattern. For these shadow areas, the camera does not provide depth information. Also, for small surfaces like fingertips and reflecting surfaces like glass, the camera will often not return a valid depth information.

The camera produces depth images of 640x480 pixels at a frame rate of 30 Hz[2]. The depth information in each captured depth image consists of a matrix of 11 bit integer depth values d_{raw}. The highest bit indicates an invalid depth value, e.g. a shadow. The range of detectable depth is approximately 0.5m to 5m, with better resolution ($<$ 1cm) at the start of this range. The depth in meters is

$$d = 0.1236\text{m} \tan\left(\frac{d_{raw}}{2842.5} + 1.1863\right) \qquad (5)$$

[Magnenat, 2011]. For depth pixel values where this formula yields a depth $d < 0$m or $d > 5$m, we conclude an invalid depth value.

To capture the foot of the musician, the depth camera is mounted on a small pedestal next to the music stand. The camera is tilted downwards by 16 degrees to capture the floor in the area of the foot as shown in Figure 4.

[2]We use the libfreenect library from the OpenKinect [Blake, 2011] project to receive a stream of depth images.

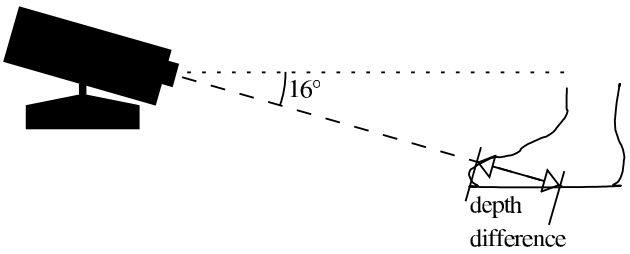

Figure 4: Kinect depth camera detects depth difference caused by foot

Our software for the virtual foot switch starts with a training phase before entering the detection phase. It first needs to capture the scene without the foot, then directs the musician to place the foot pointing naturally straight, or rotated, around the heel, to the left and right, respectively. 10 images are captured for each of the 4 situations. Mean and standard deviation for each pixel's depth is computed. Three differential depth images are computed by subtracting the mean depth information of the images containing a foot from the empty scene. Differential depth for pixels with invalid depth information or too high standard deviation in either image is set to 0, effectively excluding these pixels from further consideration. The presence of a foot where previously was only floor in the image will reduce the depth information captured by the camera as shown in Figure 4. As we are mainly interested in the position of the front part of the foot, we restrict the differential depth image to depth differences in the range of 3cm to 9cm. All depth differences outside this range are also set to zero and will not be considered for locating the foot. Figure 5 shows an example of such a differential depth image for a straight pointing foot. To identify the foot, the software searches for a cluster of depth difference pixels from bottom to top. The first such cluster of at least 100 pixels that is found is considered to be caused by the foot. The minimum rectangle that contains all three (straight, left, right) foot clusters defines the region of interest (ROI) to be used during the detection phase. Figure 6 shows the ROI from an example session, containing three foot clusters.

Also from depth images captured during detection, differential images are computed by subtracting their depth information from the empty scene training image, and considering only depth

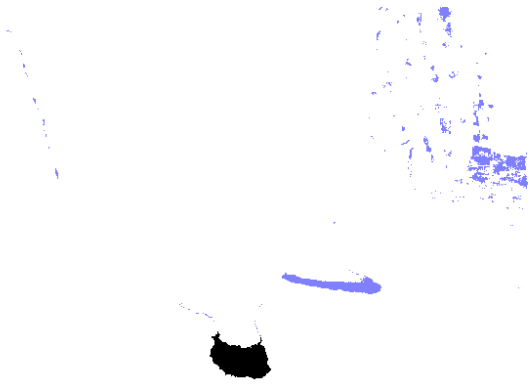

Figure 5: Differential depth image for the straight foot state from training, identified foot cluster painted black

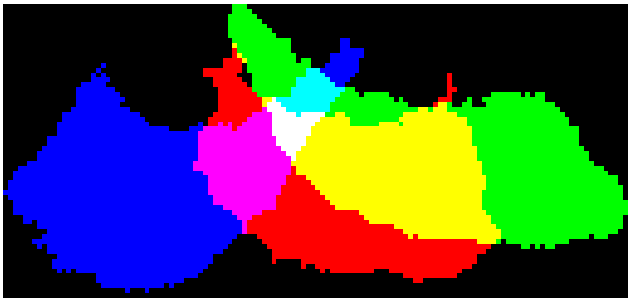

Figure 6: The region of interest (ROI) for the virtual foot switch containing the three foot clusters (displayed using additive color)

differences in the range of 3cm to 9cm. During detection, this processing is restricted to the ROI determined from the training data.

To be able to classify depth images captured during detection, an error function to compute the distance of the current state to each of the three training states containing a foot is used. Within the ROI, for each scan line, the horizontal coordinate $x_{y,i}$ with the maximum depth difference (within the range of 3cm to 9cm) is located, where y is one of the scan lines in the ROI, and $i \in \{\text{current}, \text{straight}, \text{left}, \text{right}\}$. The error function that we use is

$$e_i = \sum_y (x_{y,\text{current}} - x_{y,i})^2 \qquad (6)$$

For some combinations of y and i, $x_{y,i}$ is not defined. These combinations do not contribute to the error summation.

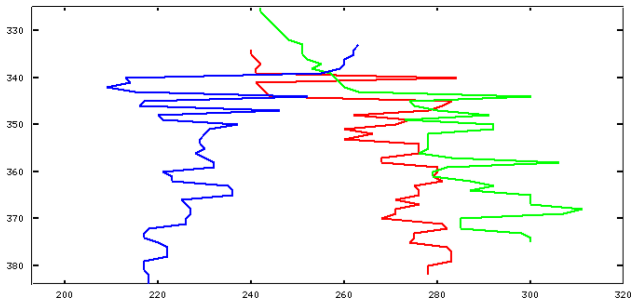

Figure 7: For each foot cluster from Figure 6 and for each scan line, the horizontal coordinate with the maximum depth difference between the training image with the foot and the empty scene is shown

Figure 7 shows these horizontal coordinates for the example foot clusters of Figure 6.

The training state with the lowest squared error sum e_i is then considered recognized. If, however, the lowest error is zero, then the empty scene is recognized.

If the left or right foot state is recognized for seven consecutive frames (i.e. for a quarter of a second), then the switch performs the configured action. The straight foot state and the empty scene do not trigger actions.

The virtual foot switch is initialized with a list of actions to perform. When the foot is recognized in the right or left position reliably, then the software triggers the next or previous action from this list, respectively. Actions are system commands and executed in a shell. For the concert, the actions influence the trajectory generator by sending appropriate OSC messages.

4 Application in hearing research

The scene generation toolbox described in the previous sections is planned to be used also in hearing research and hearing aid development, after some modifications and extensions. Here, the task is to generate dynamic acoustic scenes which are fully reproducible and controllable. Dynamic acoustic scenes can be presented to real listeners, who - within a certain area - can move freely. Hearing aid algorithms can directly be evaluated[3].

[3]For the evaluation of hearing aid algorithms in Ambisonics systems, the effect of the limitations of Ambisonics

In hearing aid research, discrete loudspeaker setups are commonly used for spatial evaluation of algorithms. However, discrete setups are inable to acoustic scenes with moving sources. Vector based amplitude panning (VBAP) as a simple extension of discrete speaker setups have the disadvantage of creating phantom sources when the virtual source is between two speakers and real sources when the virtual source direction is matched by a speaker. Phantom sources are a general problem for directional filter algorithms. Higher order Ambisonics offers the advantage that moving sources can be created with a steady image. A detailed evaluation of applicability for hearing aid algorithm testing is content of an ongoing study.

5 Discussion

Low-latency real-time spatial processing of acoustic instruments is a field with many gaps. Although much effort has been done to find solutions to most problems, many open questions still remain. One major flaw is the coding of distances: While distance coding by changing the amount of reverberation may work in low-reverberant rooms like the monitor room used during development, the effect will break down in more reverberant rooms, like most performance spaces. A potential solution might be to simulate the Doppler effect in order to at least create an image of distance changes, at the cost of additional delay. Also spectral changes which are observed in large distance differences may help in the distance perception.

Pseudo-elevation was used in the concert to create the image of sources without any distinct direction. Mixing a source to the zeroth order Ambisonics channel results in an identical signal from all loudspeakers. The effect of no distinct direction, however, can only be perceived right in the middle of the ring. At any other listening postion, the precedence effect will cause the nearest loudspeaker to be perceived as the dominant direction. The main difference to correct panning is that the perceived direction will differ for all listeners. A solution might be to use a full 3-dimensional Ambisonics system with at least first order in vertical direction, or to find alternative panning strategies.

systems on the respective algorithms (e.g., spatial aliasing, unmatched wave fronts, high frequency localization) has to be carefully considered.

6 Conclusions

A set of software tools for application in a concert performance with spatial processing of acoustic instruments has been described in this paper. The technical components of the performance were completely build on top of Linux Audio. All major parts are implemented as open source applications.

The authors believe that spatial processing can substantially contribute to a concert experience with Early and contemporary music. This believe was supported by statements of many concert listeners – "the experience of musicians moving behind your back sends you a shiver down your spine", "I felt like being in a huge bell, right in the heart of the music". However, this software toolbox leaves still space for further development, for improvement of the concert experience and for application in hearing research.

7 Acknowledgements

Our thanks go to the members of the ensemble ORLANDO*viols*, who all shared the enthusiasm in creating this concert project. Special thanks also go to Fons Adriaensen for his great linux audio tools and for helpful advice on Ambisonics. This study was partly funded by the German research funding organization Deutsche Forschungsgemeinsschaft (DFG) in the Research Unit 1732 "Individualisierte Hörakustik" and by "klangpol – Neue Musik im Nordwesten" (Contemporary Music in north-west Germany).

References

Fons Adriaensen. 2011. Linux audio projects at kokkini zita. `http://kokkinizita.linuxaudio.org/`.

Joshua Blake. 2011. Openkinect. `http://openkinect.org`.

Marina Bosi. 1990. An interactive real-time system for the control of sound localization. *Computer Music Journal*, 14(4):59–64.

Paul Davis. 2011. Ardour. `http://ardour.org/`.

Giso Grimm, Tobias Herzke, Daniel Berg, and Volker Hohmann. 2006. The Master Hearing

Aid – a PC-based platform for algorithm development and evaluation. *Acustica · acta acustica*, 92:618–628.

Giso Grimm, Volker Hohmann, and Stephan Ewert. 2011. Object fusion and localization dominance in real-time spatial processing of acoustic sources using higher order ambisonics. In György Wersény and David Worrall, editors, *Proceedings of the 17th Annual Conference on Auditory Display (ICAD)*, Budapest, Hungary. OPAKFI Egyesület. ISBN 978-963-8241-72-6.

Giso Grimm. 2012. Harmony of the spheres - cosmology and number aesthetics in 16th and 20th century music. *Musica Antiqua*, 1(1):18–22, Jan-Mar.

2010. Kinect. `http://microsoft.com/Presspass/press/2010/mar10/03-31PrimeSensePR.mspx`.

Stephane Letz and Paul Davis. 2011. Jack audio connection kit. `http://jackaudio.org/`.

Stéphane Magnenat. 2011. `http://openkinect.org/wiki/Imaging_Information`, version from June 7 2011. the wiki page attributes this formula to Stéphane Magnenat.

Dominic Sacré. 2010. Mididings. `http://das.nasophon.de/mididings/`.

Christopher W. Sheeline. 1982. *An investigation of the effects of direct and reverberant signal interactions on auditory distance perception*. Ph.D. thesis, CCRMA Department of Music, Stanford University, Stanford, California.

Matthew Wright. 2002. Open sound control 1.0 specification. `http://opensoundcontrol.org/spec-1_0`.