

AN AUDITORY 3D FILE MANAGER DESIGNED FROM INTERACTION PATTERNS

Christopher Frauenberger, Veronika Putz, Robert Höldrich, Tony Stockman

University of Music and Dramatic Arts Graz, Institute of Electronic Music and Acoustics
Queen Mary University of London, Department of Computer Science

Graz University of Technology, Signal Processing and Speech Communication Laboratory

[frauenberger, hoeldrich]@iem.at, v.putz@gmx.at, tonys@dcs.qmul.ac.uk

ABSTRACT

This paper shows the design, implementation and evaluation of an auditory user interface for a file-manager application. The intention for building this prototype was to prove concepts developed to support user interface designers with design patterns in order to create robust and efficient auditory displays. The paper describes the motivation for introducing a mode-independent meta domain in which the design patterns were defined to overcome the problem of translating mainly visual concepts to the auditory domain. The prototype was implemented using the IEM Ambisonics libraries for Pure Data to produce high quality binaural audio rendering and used headtracking and a joystick as the main interaction devices.

1. INTRODUCTION

Audio in user interface technology has long been limited to simple beeps and alarm sounds. Despite the cases in which it was inevitable to switch from visual cues to another interaction modality, the visual screen remains to be the interface designers first choice to gap the bridge between machine and human beings. With graphical user interface design having a long and lively history resulting in vast knowledge about how to build efficient and usable interfaces. However, for a number of reasons this is about to change. The complexity of human computer interaction has reached a level at which visual-only displays sometimes reach their limits. Applications become increasingly complex offering new features to the user, but without being communicated to the user they remain unused or cannot provide their full power. Furthermore, the demands in human computer interaction changed significantly with the increasing mobility of users. Modern information technology gets integrated into our everyday life seamlessly and requires small, flexible, but powerful interface technologies. To bear with those challenges, the consideration of other interaction modalities is essential and the auditory domain is a good candidate.

Currently, audio is not an equal partner for vision in human computer interaction. Too little development has been seen in the field of auditory displays to claim that audio can equally replace visual interaction. This scientific field is mainly covered by the International Community of Auditory Displays¹ (ICAD) founded 1994, a time at which visual interface techniques already were highly sophisticated. Nevertheless, the community is catching up, making auditory displays increasingly efficient and building up knowledge about how auditory communication is working. Many prototypes have proven that audio is capable of playing a major role in human computer interaction and that using it, be it in conjunction with the

visual mode or as sole alternative to visual screens, is a promising approach to address the challenges in modern information technology described above.

Another aspect of using audio as interaction modality is the accessibility of computers for people with visual disabilities. As a matter of fact, computing highly dependent on graphical output is of little use for the visually impaired and blind. Relying on auditory and tactile interaction the people affected have to deal with information translated from the visual domain which basically is limited to the text available and much information presented through graphical means is lost. Screen reading software also represents the state of the art of auditory displays for computers on the consumer market. This means sequential, text orientated audio output for all what comes with a modern computer.

Given the major improvements in audio processing and all the possibilities provided by powerful sound hardware as well as software, this paper argues for using all those audio technologies in conjunction with HCI methodologies to improve the quality of auditory user interfaces. We have built a prototype of a file managing application using a spatial auditory rendering system to set up a virtual 3D environment as a user workspace.

Graphical design has produced a number of reliable concepts for robust and usable user interfaces and we propose to learn from these methodologies for the auditory domain while keeping in mind that there are significant differences in the way either domains communicate. This was the reason for adapting the well established pattern design method for our prototype by introducing a mode independent meta domain. With this concept user interfaces can be designed without determining their means of realisation and specialised transformations can be developed for each interaction domain (e.g. auditory or visual). With that, the strengths of every domain can be preserved and while using a method well known from the graphical design, we do not apply graphical concepts on auditory interaction.

The following section describes the current state of the art in audio rendering and auditory display development. Subsequently, section 3 shows the design method used and introduces the design process with mode independent interaction patterns. Section 4 describes in detail the prototype developed, its design, its implementation and the evaluation test conducted. Finally, section 5 concludes the paper and provides an outlook on future work.

2. STATE OF THE ART

The increasing computational power available for digital signal processing has made increasingly complex simulations of acoustical environments possible. The simulation of acoustical scenes

¹<http://www.icad.org>

with sound reproduction techniques create natural environments which are customisable and controllable in real time, very much as visual virtual environments were developed [1]. Acoustical rendering of objects (sound sources), the environment and the listener can be realised using a number of different approaches like Ambisonics [2], Wave Field Synthesis [3] or Vector Based Amplitude Panning [4]. All of those techniques are capable creating the sensation of various output formats ranging from large scale loudspeaker arrays to binaural rendering for headphones. This technology was just recently introduced to create auditory displays as a major step towards natural interfaces [5, 6].

A variety of auditory displays were developed for specific problem domains (e.g.: [7, 8]) and some efforts were taken towards a structured approach for more generic solutions. Early proposals include the *Mercator* project, the first framework targeting customary Unix desktops [9]. Another proposal was *Y-Windows* also following the idea of building alternative, audio rendering engines (servers) for existing clients requesting their user interface representation [10]. However, both approaches implied that graphical concepts were translated into the auditory domain and therefore had their limitations. A first attempt to depart from this approach and introduce a mode independent meta domain was made in [11] and subsequently led to the concept proposed in [6, 12] and in this paper.

Attempts to employ usability engineering methodologies in the design of auditory displays include the investigation of audio metaphors [13] and other structural approaches to include sound into human-computer interaction (Earcons [14]). Recently, the proposal of using patterns in sonification highlights the advantages of such methods in re-usable designs [15, 16].

However, much more may be exploited from the discipline of usability engineering. Auditory representations of user interfaces are in need of profound heuristics to assess user satisfaction similar to those in the graphical domain [17]. Examples of where work is needed to identify heuristics to guide the process of auditory display design include minimising the problems incurred due to the transient nature of sound, quantifying how the effectiveness of interactions can be improved through learning and providing guidelines for how sound can best be integrated with other media [18]. Also the design pattern method is a promising approach and other design principles may also give more control over the efficiency of auditory displays.

3. PATTERN DESIGN

Interaction pattern design has been successfully used to create robust user interfaces in the visual domain. The main advantages of this method can be summarised by

- re-usable solutions to common HCI tasks
- consistency of solutions in different interfaces
- easy and quick way to build interfaces from scratch.

There are various sets of patterns available with different objectives and at different levels of abstraction. To achieve the goal of finding a common base for HCI in different modalities we evaluated sets of patterns and found Welie and Trætteberg's patterns to be the most suitable for our purposes [19].

We used a reformulated set of those patterns to design our application in the auditory domain, but while developing the patterns we kept in mind that this approach should be applicable on every

other interaction domain equally. As stated above the main intention was to abstract the problem statement in human computer interaction to a level where a description does not imply the means of representation. Creating a real interface in the auditory domain can then be made considering all the special properties of this channel of interaction without having to deal with visual concepts. Figure 1 illustrates the approach.

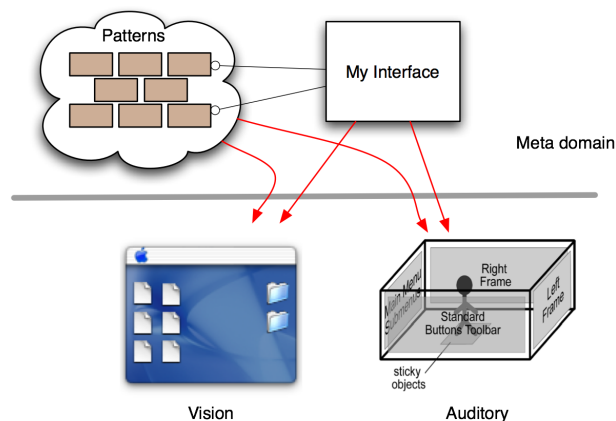


Figure 1: The meta domain as common base for different representations of a user interface.

The re-formulation of the patterns to make them independent from the used interaction mode demanded a change of the terminology for the description of the patterns. The *representation medium* means the domain or the combination of the domains in which the user interface will be realised. Within this representation medium there are *representation areas* defined which provide the boundary for *objects* of the user interface. These objects may result from one or more *interaction patterns* transformed into the representation medium. Despite these small changes to the terminology, the existing interaction patterns were easy to re-formulate so that they would not prejudice the process of realisation.

While developing the patterns, we recognised that certain tasks or parts of patterns recurred in other patterns too. This led to the concept of atoms and contextual attributes. Similar to a vocabulary for instantiating designs, a set of atoms were developed from which patterns may draw when addressing a particular set of user requirements. This also implies consistent representation of similar elementary units throughout the whole interface although atoms are not sufficient to solve any interaction problem. In order Not to end with a totally unrelated patchwork of small pieces of a user interface, each atom provides contextual attributes. These attributes need to be set by the parent pattern in order to indicate their context. In the graphical domain this would, for example, mean that certain elements like buttons or text fields are *in the same window* sharing the same frame and background colour. The following contextual attributes were identified for our set of atoms:

Similarity: Atoms in the same pattern share properties like timbre, rhythm or type of voice in their acoustical representation.

Proximity: Atoms in the same pattern are grouped based on the available dimensions of the representation area (space or pitch ranges).

Homogeneity: The same types of atoms should be placed adjacently in a pattern on the basis of the available dimensions of the representation area (space or pitch ranges).

It is important to state that not only the patterns and the atoms undergo the transformation process in order to form a real user interface, but also the contextual attributes must be mapped into the different representation media. Their realisation in the auditory domain will differ considerably from the visual domain.

The subsequent section shows the design and realisation of an auditory version of a file management application using a subset of these patterns. It would be out of scope of this paper to describe all patterns developed in great detail so here we shall only show the patterns used for the Audio Explorer. The full set of patterns and more information about them is available from [20].

4. THE AUDIO EXPLORER

To prove the concept of design patterns with auditory displays we implemented a prototype display of a real world application. Being fundamental to every operating system and fairly well known to most computer users, a file managing application was chosen for that matter. The goal was to analyse the existing graphical application, describe it using the set of mode-independent patterns and then create an auditory display with that description.

4.1. Design

Analysing the Microsoft Explorer application resulted in a description using the following patterns (atoms):

- Container Navigation (tree structure, list)
- Command Area (triggering element, selection)
- Context Menu (triggering element, selection)
- Message (triggering element, raw information)

Although not the whole functionality of the MS Explorer was considered in this analysis, the prototype was still covering the most important functions for a file manager.

The *Container Navigation* pattern was used to describe the two main frames of the Explorer. For the folder tree in the left frame the *tree structure* atom was used and the *list* atom described the right content frame. The *Command Area* pattern described the menu structure and the tool bar area. Finally, the *Contextual Menu* pattern solved the availability of the context menu and the *Message* pattern was used for all pop-up windows at their occurrence. Figure 2 shows the basic layout of the virtual audio environment into which the patterns were transformed. The container navigation pattern was realised as two different areas in the virtual environment, the walls to the front and to the right. The listing was put on the front wall using speech for the name of the items with different voices to indicate the type (folder or file). The tree structure was laid out on a grid on the wall to the right with the left-bottom corner being the root and the right-top corner the last hierarchical level. Unfolding and folding a node in the tree was also implemented. In both areas the user was able to select items and get a context menu. The contextual pattern was realised as sticky objects following the user wherever she moves. The content of the contextual menu pattern was again solved by triggering elements. The same concept was used to realise pop-up windows - sticky objects remaining to the front of the user, but with different background sound. The menu was created on the left wall of the room

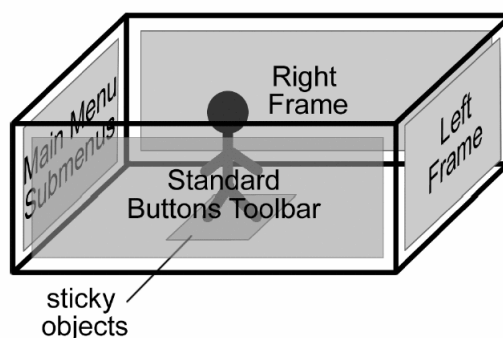


Figure 2: The layout for the virtual environment for the Audio Explorer

using speech sources lined up along the wall. When clicked they unfolded their content towards the ceiling and a series of raising tones indicated the number of menu items. The user could then virtually move vertically to select the item desired.

All sound used incorporated speech and non-speech sound and was audible within a certain range. This means that there was silence in the starting position, but moving towards a wall meant that one could hear the 5 menu items at once at different levels depending on the distance. Interaction with the prototype was done by joystick and keyboard. To avoid confusion while navigating through the virtual environment no relative movements are supported. Bringing the joystick to the starting position means moving to the centre of the room facing the front wall. Moving up, along the z-axis, in the environment was implemented using the throttle handle of the joystick. The localisation of different sound sources was improved by using a headtracker.

4.2. Implementation

The virtual environment was implemented in Pd² using binaural rendered Ambisonics. The Ambisonics libraries developed by the Institute of Electronics Music and Acoustics Graz as an extension to Pure Data were used to simulate the environment in an efficient and flexible way [21, 22].

With this system, Ambisonics is used as an intermediate stage towards a final binaural mixing. First, all sound sources are encoded in Ambisonics resulting in a number of Ambisonics channels. In this domain the encoded sources can easily be mixed and rotated before Ambisonics decoding calculates a set of corresponding loudspeaker signals. For binaural rendering these signals are then convoluted with an appropriate set of head related impulse responses (HRIRs). In order to improve the external localisation of sound sources and the naturalness of the virtual environment, there was also a room model implemented considering early reflections and late reverberation. The described Ambisonics rendering system has a number of advantages that makes it especially suitable for the task given:

- Very efficient for a large number of sound sources. This is particularly important as all mirror sources resulting from the room model are treated the same way as original sound sources, increasing the amount of sources significantly.

²Pure Data by Miller Puckette

- Efficient support for rotation in the Ambisonics domain with rotation matrices. Using headtrackers this is an important requirement.
- Easy adaptation of the output format. This rendering system can be used with either binaural output or multi-channel output for an array of loudspeakers.
- Efficient for binaural output because the number of HRIRs needed is independent from the amount of the sound sources and no interpolation between HRIRs needs to be calculated.
- Incorporated room model with early reflections and late reverb supporting the externalisation of sound sources.

The used library consists of six modules which contain objects and abstractions to be used with Pure Data. **iem.matrix** provides efficient matrix operations in the signal domain and is basically a collection of helper functions. **iem.ambi** contains all objects for encoding, decoding and rotation in the message domain meaning it is providing all necessary parameters to the objects that actually handle the signals. **iem.bin.ambi** is the module responsible for decoding Ambisonics signals to a virtual loudspeaker setup and applying HRIRs to the decoded signals. It currently uses non-personalised HRIRs from KEMAR as well as CIPICS databases, but may well be used with any others too. Because this module requires the most computational effort in the whole rendering process, a number of optimisations were developed to make this module highly efficient. These optimisations include:

- Reduced HRIR set: due to the symmetry in the virtual loudspeaker setup the number of HRIRs could be reduced to its half.
- Frequency domain filtering improves computational efficiency compared to time domain implementations.
- Reduced IFFT: By combining the decoder and HRIR filters and transforming them into the frequency domain reduces the number of required IFFTs to two, one for each ear-signal.

iem.roomsim provides the room simulation with calculation of early reflections of first and second order. **iem.reverberation** implements a computational efficient calculation of a reverberation algorithm based on former work of J.-M. Jot and A. Chainge and M. Puckette [23]. Finally, **iem.gui** is a module which allows for creating graphical user interfaces and scene representations for the virtual environments using the GEM³ extension for Pd.

Given the advanced functionality of this library, the implementation of the Audio Explorer was straight forward. The application used pre-recorded sound files and real-time synthesised sound as sources to the library and controlled the created scene with the parameters acquired from the input devices (keyboard, headtracker and joystick). The content of the file manager was faked due to the difficulties with involving calls to the operating system in the application.

Figure 3 shows a screenshot of the rendering system showing the rendering process with 6 input streams.

4.3. Evaluation

The test was performed by a group of 15 test participants divided into two groups. Group S were seven students of the Graz Univer-

³Graphics Environment for Multimedia is an external to Pure Data. See <http://gem.iem.at>

sity of Technology and one person already holding a masters degree. All of them were between 20 and 27 years old and had good experience with computers and Windows. Group B consisted of four persons who are totally blind and three persons with visual disabilities. The use of visual screens was only feasible for them using additional magnification software. Six participants in group B hold the ECDL (European Computer Driving License). They use a computer in their work and were very experienced with Windows software. One member had little experience with computers, but was attending the course for receiving the ECDL. On average Group B was a little older.

After instruction, the participants got 15 minutes of training time with the application, participants were given a list of 7 tasks to perform. The tasks involved finding out how many files are in a specific folder, finding the size of files, copying, moving and creating files or folders.

Throughout the test, different types of data were collected. On the one hand, the hierarchic structure of files and folders after the test is stored in a text file. Apart from that, two further lists report the whole test sequence, one list containing the movement of the joystick within the virtual room (x,y,z-coordinates, rotation around the z-axis) in a resolution of 50 ms, the other list reporting any action performed by the participants with a time index, so that both lists can be combined. With these two lists, the whole test performance of the participants can be reproduced and visualised. The list of reported events can also be used to compute the quantity of different events. Apart from that, the whole performance of the participants was attended by the test administrator via headphones who additionally took notes.

After the test, the participants had to answer two questionnaires. One concerning the individual background of the participants, the other trying to catch the subjective impression of the participants after the test.

The three-dimensional layout of the virtual room with the different meanings of the four surrounding walls and the two-staged movement (ground-plane movement towards the walls, vertical movement for selection) proved to be sufficient to host the elements of a real-world application. On the ground floor, at least 20 items (5 on each wall) can be represented, not to mention the potential with regard to vertical placement. The thematic grouping of elements on the different walls was easy to memorise for the test users. The usability of the mappings of particular interaction patterns is different. While the menu structure was easy to use for most test participants, the representation of the folder hierarchy is in need of improvement: hardly any test user had a clear overview of the file and folder structure. According to the participants the static grid layout was confusing because they lost track of the absolute position while navigating through the tree structure. This was fostered by the fact that hardly anyone could re-construct the file structure correctly after the test.

Overall, the test results proved the concept and showed that this approach is promising in terms of efficient user interfaces in virtual audio environments. Further results and detailed analysis of the collected bottom line data is available from [20, 12].

5. CONCLUSION

The evaluation of the prototype system clearly showed that the approach to design and implement auditory displays in auditory spaces is feasible and can produce efficient user interfaces. This work showed the concepts behind the design, the implementation

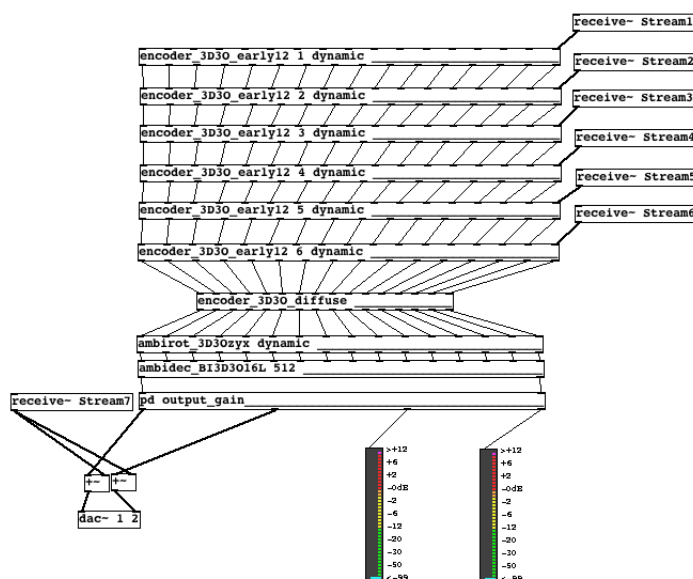


Figure 3: Pd screenshot with the Ambisonics rendering system.

of an auditory file-manager application using the IEM Ambisonics library and described the evaluation process of the resulting prototype. The quality of the auditory realisation clearly met the requirements while flaws were detected in the realisation of certain design patterns. However, the concept made it possible to isolate the usability problems and assign them to certain patterns or atoms in design. This makes it possible to improve smaller pieces of an interface instead of dealing with the whole.

Future work will need to improve the patterns to be consistent and complimentary. This will need further investigations in psychoacoustics, acoustic communication, sound design, aesthetics and navigation and orientation in virtual audio environments. On the concept side, work needs to proceed towards formalising the patterns and the development of supporting tools in order to support user interface designers.

6. ACKNOWLEDGEMENTS

We would like to thank all who volunteered to participate in the evaluation test. Especially the people from the ISIS group (Integration, Service, Information and Schooling) in Graz / Austria for the close collaboration and for giving us an idea about the needs the visually impaired have when it comes to interaction with computers. ISIS is a project run by BFI (Berufsförderungsinstitut) Styria.

Support by the SonEnvir (<http://sonenvir.at>) project sponsored by the Zukunftsfonds Steiermark is gratefully acknowledged.

7. REFERENCES

- [1] T. Lokki et al., "Creating interactive virtual auditory environments," *IEEE Computer Graphics and Applications, special issue "Virtual Worlds, Real Sounds*, vol. 22, no. 4, pp. 49–57, July/August 2002, Electronic publication <http://www.computer.org/cga/>.
- [2] J. S. Bamford, "An analysis of ambisonic sound systems of first and second order," M.S. thesis, University of Waterloo, <http://audiolab.uwaterloo.ca/~jeffb/thesis/thesis.html>, 1995.
- [3] E. Verheijen, *Sound Reproduction by Wave Field Synthesis*, Ph.D. thesis, TU Delft, 1998.
- [4] V. Pulkki, "Virtual sound source positioning using vector base amplitude panning," *Journal of the Audio Engineering Society*, vol. 45, no. 6, pp. 456–466, June 1997.
- [5] M. Strauss et al., "A spatial audio interface for desktop applications," in *AES Proceedings, International Conference on Multichannel Audio*, Banff, Canada, June 26–28 2003, AES: Audio Engineering Society.
- [6] R. Höldrich C. Frauenberger, V. Putz, "Spatial auditory displays - a study on the use of virtual audio environments as interfaces for users with visual disabilities," in *DAFx04 Proceedings*, Naples, Italy, October 5–8 2004, 7th Int. Conference on Digital Audio Effects (DAFx'04).
- [7] C. Schmandt M. Kobayashi, "Dynamic soundscape: Mapping time to space for audio browsing," in *ACM/SIGCHI 97 Proceedings*, Los Angeles, CA, March 22–27 1997, ACM Conference on Human Factors in Computing Systems, pp. 194–201.
- [8] C. Schmandt, "Audio hallway: A virtual acoustic environment for browsing," in *Proc. ACM Conference of Computer Human Interactions*, April 18–23 1998, pp. 163–170, ACM Press, Los Angeles, California, USA.
- [9] T. Rodriguez W. K. Edwards, E. D. Mynatt, "The mercator project, a nonvisual interface to the x window system," *The X Resource*, 1993, O'Reilly Publishers.
- [10] M. Kaltenbrunner, "Y-windows: Proposal for a standard au environment," in *ICAD Proceedings*, Kyoto, Japan, July 2–5 2002, International Community for Auditory Display.

- [11] A. de Campo C. Frauenberger, R. Höldrich, "A generic, semantically based design approach for spatial auditory computer displays," in *ICAD Proceedings*, Sydney, Australia, July 6–9 2004, International Conference on Auditory Display.
- [12] C. Frauenberger et.al., "Interaction patterns for auditory user interfaces," in *ICAD Proceedings*, Limerick, Ireland, July 6–9 2005, International Conference on Auditory Display.
- [13] W. K. Edwards E. D. Mynatt, *Extraordinary Human-Computer Interaction*, chapter Metaphors for Nonvisual Computing, Cambridge University Press, 1995.
- [14] R. M. Greenberg M. M. Blattner, D. A. Sumikawa, "Earcons and icons: Their structure and common design principles," *Human-Computer Interaction*, vol. 4, no. 1, pp. 11–44, 1989.
- [15] S. Barrass, "Sonification design patterns," in *ICAD Proceedings*, Boston, USA, July 6–9 2003, International Conference on Auditory Display.
- [16] S. Barrass M. Adcock, "Cultivating design patterns for auditory displays," in *ICAD Proceedings*, Sydney, Australia, July 6–9 2004, International Conference on Auditory Display.
- [17] J. Nielsen, *Usability Engineering*, Academic Press, London, 1993, ISBN 0125184050.
- [18] G. Kramer et. al., "Sonification report: Status of the field and research agenda," <http://icad.org/websiteV2.0/References/nsf.html>, February 2005.
- [19] H. Trætteberg M. van Welie, "Interaction patterns in user interfaces," in *PLOP Proceedings*, Monticello, Illinois, USA, August 13–16 2000, 7th. Pattern Languages of Programs Conference.
- [20] V. Putz, "Spatial auditory user interfaces," M.S. thesis, Institute of Electronic Music and Acoustics, University of Music and dramatic Arts Graz, 2004, <http://iem.at/projekte/dsp/spatial/dp-putz>.
- [21] M. Noisternig et al., "3d binaural sound reproduction using a virtual ambisonic approach," in *VECIMS Proceedings*, Lugano, Switzerland, 27–29 July 2003, International Symposium on Virtual Environments, Human-Computer Interfaces, and Measurement Systems, IEEE.
- [22] M. Noisternig et.al., "A 3d ambisonic based binaural sound reproduction system," in *AES 24, International Conference on Multichannel Audio*, Banff, Canada, June 26–28 2003, Audio Engineering Society.
- [23] A. Chainge J. M. Jot, "Digital delay networks for designing artificial reverberators," in *Proc. of the 90 th Conv. of the Audio Eng. Soc.* Audio Engineering Society, Februar 1991.