The CUBEMixer a Performance-, Mixing- and Masteringtool

Thomas Musil
Institute of Electronic Music
and Acoustics
Inffeldgasse 10
Graz,Austria 8010
musil@iem.at

Winfried Ritsch
Institute of Electronic Music
and Acoustics
Inffeldgasse 10
Graz,Austria 8010
ritsch@iem.at

Johannes Zmölnig
Institute of Electronic Music
and Acoustics
Inffeldgasse 10
Graz,Austria 8010
zmoelnig@iem.at

Abstract

With the CUBEMixer we created a real time multi-purpose mixing- and masteringtool with integrated room-simulation possibilities for multichannel speakersystems or binaural production on base of the graphical programming language Pure-Data. I was designed to be used for Performances as Mixingdesks, 3D-Mastering application and virtual acoutics. It has parallel master sections for rendering inputchannels in either 3D-space with ambisonic for multichannel speaker-systems or binaural rendiering, or to a bus-subsystem for special use-cases or other spatializations strategy. Input-channels can be rendered on different multichannel speaker systems or binaural stereo-output. Also with his plugin- and extensions-system the mixer can be enhanced for any other purpose. It can be controlled with a PD-native GUI, over OSC using TCP/IP oder MIDI and has a dynamic parameter handling system for storing an loading settings, parameter on files. It can be easily extended for special compositions and performances or used as a library for PD.

Keywords

Spatialisation, Binaural, Mixer, Virtual Room Acoustics

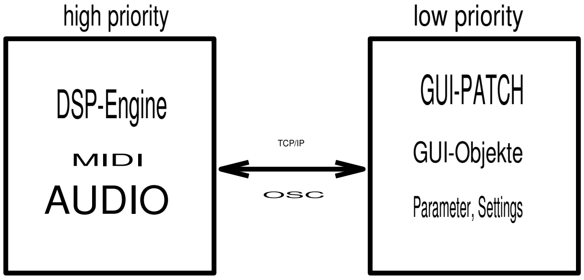
1 Introduction

The CUBEmixer[8] thinks of sound as a spatial, sculptural phenomenon and is a softwaretool which can render Sound in an 3D-audioenvironment. The control data, should be independend from the current playback/rendering situation. The interpretation of the 3D-spatialising data (azimuth, elevation, distance) for the inputchannels can be done for different domains, so also be renderd to sourround or even plain stereo.

The CUBEmixer is programmed with the graphical computermusic language Pure-Data [6] with few special external objects libraries [4] and can be extended with plugins, extensions and any other PD-Patch which respects the namespaces.

The basics of the software has ben developped since 1999 out from different projects for sound-installtion, performances in big concert halls with multichannel audio and mastering projects with a lot of research and experiments done at IEM. The software original intented to drive the CUBE [12], a concert hall with 82 speaker over 54 channels at IEM, and was generalized and adapted for the use in different audio environments. It is distributed under GPL. The goal is to find a common performance standard for Audio mixed in The 3D-domain for our and other projects.

2 Architecture of the system



The software is split into a DSP-Patch and a GUI-Patch to seperate the latency dangeroous GUI-control from the low latency DSP-Patch and also to remote control the audioworkstation over TCP/IP. The connection between them is done with OSC[1][10] and should be run over a TCP/IP Link,

[1] Open Sound Control Protocol

even also UDP-connections are sufficients.

2.1 Module Structure

The mixer is split into input section with an configurable number of inputs with the encoders and a master section with the decoder and other tools, like subwoofer-system, effects, 3D-reverbs, soundfile-player/recorder, etc. At the time of writing an Ambisonic Decoder for a 24.1 channel Speaker system in an Hemisphere doing 4th order ambisonic [1,2,3], a binaural rendering stage using the an ambisonic decoder with HRTF-Filters [5], a busmixer and 64x64-matrix for additional routing, a subwoofer mastersection, "room in room simulation", and 3D-reverb. In the ambisonic domain, the 25 ambisonic signals can be used as a storage format for already spatialized soundfiles.

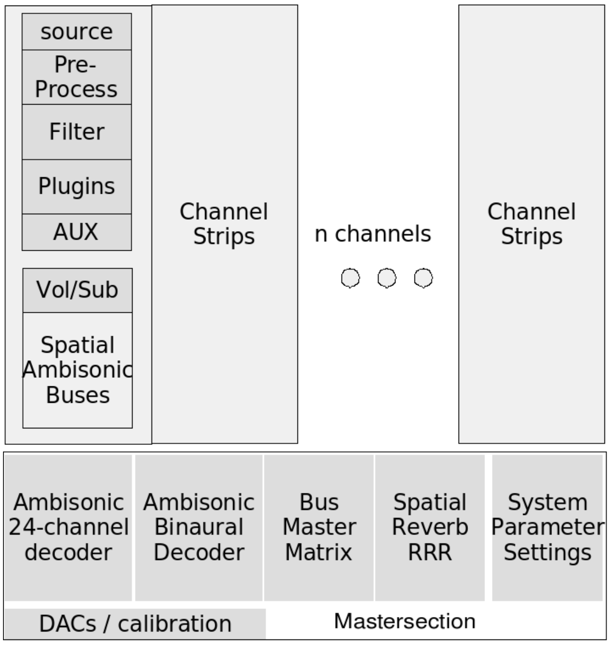


figure 1: mixer organisation

Additionally all paramters can be stored and loaded from files, where the parameter naming follows the OSC style. Testsignal generators and soundfile player, recoder complete the Version 2 of the CUBEmixer.

2.2 Input Section

Inputs are channels which can be assigned to ADCs of the soundcard or other sources in the system by effects, soundfile players, own extensions, AUX-routings and free definable signals from extensions.

Each input can be routed to AUX-Sums, to a ambisonic encoder or the to a bus matrix.
Also plugins can be configured, not only to modify signals as also control other parameter of the systems eg. to link channel effects.

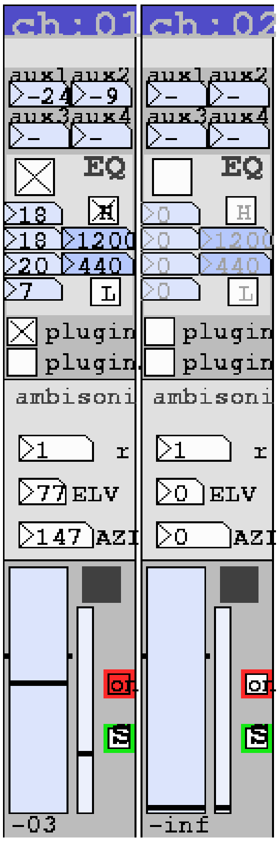


Abbildung 2: input channel overview GUI

The spatial control data is an abstract set of the angles elevation, azimuth and the distance r from the virtual ambisonic center. The volume section allows an additional output to the subwoofer master section.

2.3 Extensions

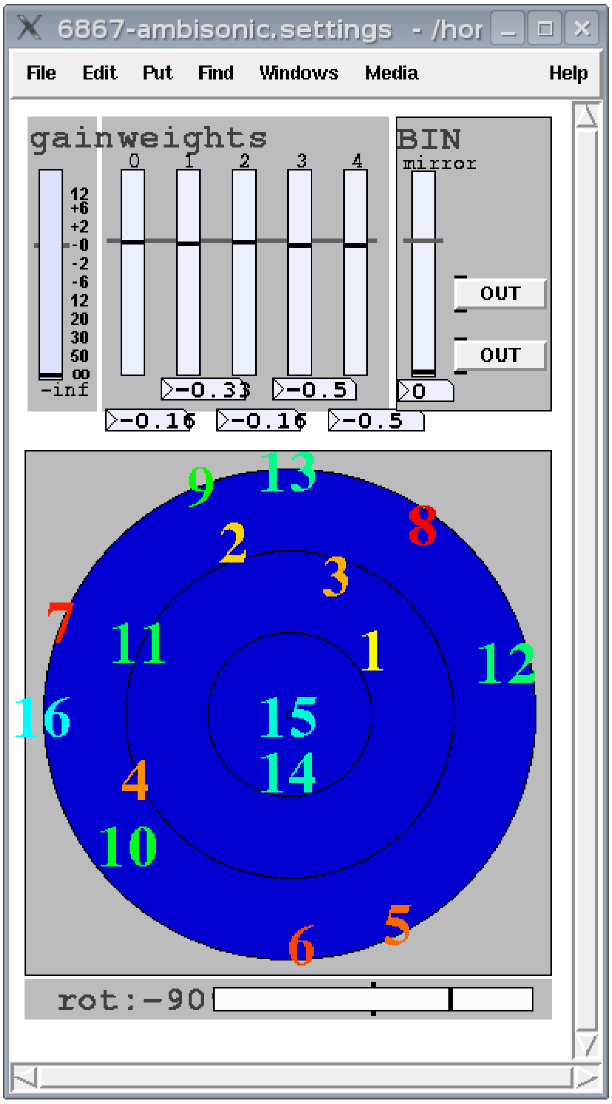
Extensions are modules in the master section which provide sinks and sources for the input channels. The ambisonic decoders are extensions, like the busmaster, reverbs, soundfile player, ... and different implementation can be present at the same time.

2.4 Controller

Since OSC with the dynamic naming mechanism is used as base for controls, the pattern matching system of OSC is used for filtering parameter groups and store them in seperate files or control a whole bunch of channels parallel. Also generative complex movement algorithms can be implemented as patches, with their own parameter room.

The OSC control data can be mapped to MIDI connections, so sequencers can store control data and play them back. Also the system is designed to be driven by MIDI-Controller like Fader-Boxes or other sensoric systems.

3 Ambisonic system



In this version ambisonic is used in 4th order for horizontal and vertical spatialization. The encoder can be used to feed a weightable order resolution of the 25-channel ambisonic signal to control the width of the source. This is used to code the distance of the signal toward the center.

The ambisonic-decoder renders the ambisonic signal to the actual speaker settings, in our case 24 speaker hemisphere, having an own decoding matrix parameter set. For each speaker setup an own parameter set for the decoder has to be calculatied. The speakers can be calibrated in volume and propagation time, so the offset in the distances of the speaker to the center can be equalized. Of course a well aranged speaker set sounds better than an odd one, but also not perfect speaker distributions can be handled efficient.

As an special feature, the ambisonic rendered signal can be stored and playbacked later without the use of inputs, enabling the playback of additional complex sound environments in parallel.

4 3D reverberation system

Since CUBEmixer can render audio signals in the 3D-domain, first reflections and late reverb can also be spatialized. With this feature the reverb sounds more natural and transparent. Especially the spatialization of the late reverb allows the construction of virtual acoustic rooms with different dimensions in different directions, e.g. staying in the coners of a room or simulate a room besides the main room.

5 Binauraul rendering

The ambisonic signals can also be rendered on an binaural ambisonic-decoder. Therefore a 4th order ambisonic decoder is used. The CUBEmixer can be a tool for production of binaural environments. This is used also to set up complex spatialization-productions with headphones for preproduction before moving to the concert hall.

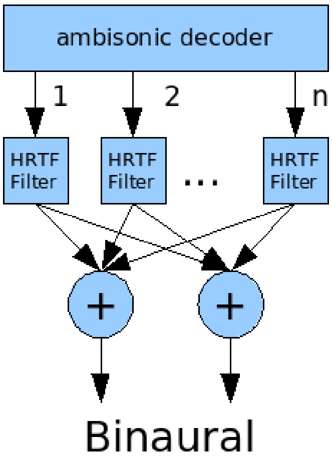
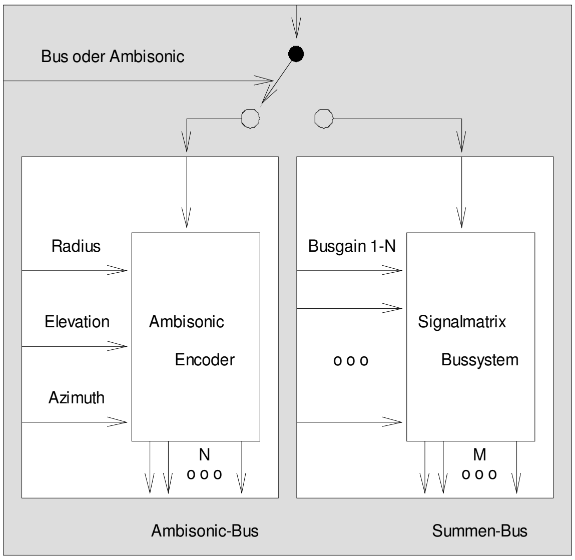


figure 3: Ambisonic Bibaural Renderer

Important for the good performance of the binaural rendering is a room simulation, wich is a multichannel reverb. Plugins can produce first reflections wich are also spatialized using a virtual room model.

6 Room in Room Simulation

With the usage of microphones in the performance room, preferable to be mounted on the walls, a “room in room reverberation” (RRR) can be build. The RRR module renders the reflections and reverb in the same room. So virtual acoustical rooms, on top of the physical room acoustic can be constructed for performances and rehearsals.



7 Tools, Control and Storage

The control of the cubemixer is done with OSC, which can be mapped to/from MIDI and can be adapted to a lot of environments for rendering 3D-audio and virtual acoustics.

8 Conclusion

The CUBEmixer has proofed to be general tool rendering 3D-Audio in different situations and is also ready to be extended with other render domains like vector-panning, etc. It can be run in dedicated computemusik workstation as a distribution system in soundinstallation and concert halls and on laptops for binaural productions.

The CUBEmixer is an open source project licenced under GPL and can be downloaded from sourceforge or checked out from svn from the IEM opensource projects site at http://iem.sourceforge.net/

9 References

[1] Gerzon, M. A. (1985). “Ambisonics in Multichannel Broadcasting and Video”, J. Audio Eng. Soc., 33(11), pp. 859-871.

[2] Malham, D. G. (1992). Experience with Large Area 3D Ambisonic Sound Systems. In Proc. Inst. Acoust.

[3] Musil, T. and J. Zmölnig, M. Noisternig, A. Sontacchi, Robert Höldrich: AMBISONIC 3D-Beschallungssystem 5.Ordnung für PD, IEM Report 15/2003

[4] Musil, T.: IEMLIB für PD, IEM Report 12/2003

[5] Noisternig N and A. Sontacchi, T. Musil, R. Höldrich: A 3D Ambisonic based Binaural Sound Reproduction System" AES 24th International Conference, 26-28 June 2003, Banff, Canada

[6] Puckette, M. 1996. "Pure Data: another integrated computer music environment." Proceedings, Second Intercollege Computer Music Concerts, Tachikawa, Japan, pp. 37-41.

[7] Ritsch W. and R. Höldrich, C. Frauenberger: *Internet Archive for Electronic Music IAEM-iARS internet Audio Rendering System*, AES 24th International Conference, 26-28 June 2003, Banff, Canada.

[8] Ritsch W. and Musil T., Zmölnig J., Höldrich R. (1995):*Implementation eines kombinierten Ambisonic- und Bus-Mixers für den Einsatz in 3D Audio Environments, IEM Report 28/05*

[9] Sontacchi, A.: Neue Ansätze der Schallfeldreproduktion, Dissertation der TU Graz 2001

[10] Wright, M. and A. Freed 1997. Open Sound Control: A New Protocol for Communicating with Sound Synthesizers. Proceedings of the International Computer Music Conference, Thessaloniki, Hellas, pp. 101-104.

[11] Zmölnig, J and W. Ritsch, A. Sontacchi: *"Der IEM CUBE - ein periphones (Re)Produktionssystem"*,22.TMT, Produktforum, Jahrestagung des Vereins Deutscher Tonmeister, Hannover, November 2002

[12] Zmölnig J. and W. Ritsch, A. Sontacchi: *"The IEM CUBE",* ICAD - International Conference on Auditory Display, July 7-9, 2003, Boston University, Boston, MA