

# OVBOX: REMOTE MUSIC AND SPEECH COMMUNICATION WITH INTERACTIVE LOW-DELAY VIRTUAL ACOUSTICS USING THE OPEN SOURCE TOOLBOX TASCAR

Giso Grimm\*

Digital hearing devices  
 Carl von Ossietzky Universität Oldenburg, Germany  
 g.grimm@uol.de

Volker Hohmann

Digital hearing devices  
 Carl von Ossietzky Universität Oldenburg, Germany  
 volker.hohmann@uol.de

## ABSTRACT

This workshop will show how to install and use a remote collaboration device *ovbox* [1] built upon the tool ‘zita-njbridge’ [2] and the Toolbox for Acoustic Scene Creation And Rendering (TASCAR) [3, 4, 5]. The *ovbox* is a remote collaboration box developed by the ensemble ORLANDOviols primarily to allow rehearsals during the lockdown due to Covid19 pandemia. This box is completely built upon open source software and open or standardized hardware. At the University of Oldenburg this project was extended towards application for low-delay real-time streaming in hearing experiments, e.g., to render complex virtual acoustic environments on a powerful PC in the lab, and stream the signals to the user via a device with calibrated reproduction hardware. Also low-delay transmission of head movements for head-tracked rendering is possible.

During the workshop we will explore some technical details of the system, and how to install it on your system. We will explain the applied 3D virtual acoustic rendering. At the end of the day you will be able to jam together with other participants of the LAC.

## 1. INTRODUCTION

The currently ongoing pandemia with frequent lockdowns and social distancing has a huge impact on music performance, but also on research with human participants.

Some tools are currently available for this purpose (e.g., Jamulus, soundjack, JamKazam). However, for music applications often a hardware-based system is preferred over desktop applications for best performance, especially for minimizing the delay, but also for optimal usability (“plug-and-play”). Furthermore, for research applications, it is difficult to extend and integrate those tools into existing laboratory frameworks.

To overcome these problems, the ensemble ORLANDOviols developed an own solution, mostly built upon the network audio tool “zita-njbridge” [2], which solves the problem of predictable delays and adaptive resampling. The sound sources from remote peers are rendered in the 3D audio engine, the Toolbox for Acoustic Scene Creation and Rendering (TASCAR) [3, 4, 5]. Room acoustics and spatially distributed source positions are simulated, optionally also including head tracking. The ‘ovbox’ is also be used in hearing research.

## 2. METHODS

The tool “zita-njbridge” can transmit audio within a local network. It can achieve a constant delay by transmitting time stamps along

\* Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Projektnummer 352015383 – SFB 1330 B 1

with the audio. It also provides adaptive resampling to compensate for clock drift or different sampling rate settings at the end points. While it is possible to build direct connections with zita-njbridge, it does not provide any methods for firewall and NAT traversal.

A relay server was developed, which manages the firewall and NAT traversal, handles peer-to-peer connections, multiplexes several UDP streams into a single UDP stream, and duplicates packages to multiple clients when audio is transferred via the relay server. A corresponding client serves as an endpoint and peer for the zita-njbridge. Since the relay server protocol is completely content-agnostic, it can also be used to transfer other data, such as head orientation sampled by motion sensors.

To handle the flexible collaboration between multiple ensembles, or in several labs with many participants, a configuration and “room booking” system was developed as a web service. The relation between these different components is sketched in Figure 1.

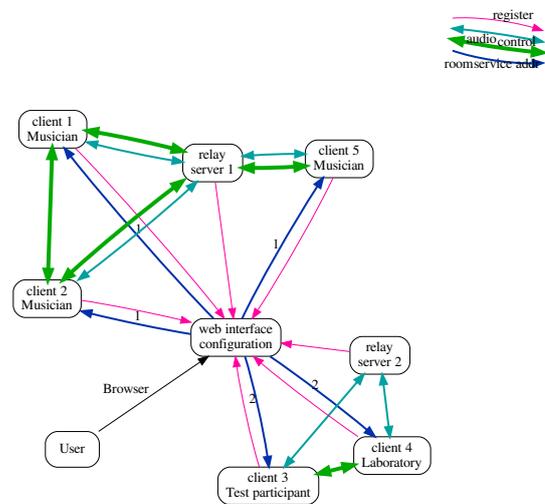


Figure 1: Server model of the *ovbox*. The user controls the devices and manages connections via a web interface. Devices and relay servers also request control information from the web configuration interface.

In the device, the virtual acoustic simulation tool TASCAR [5] is used to render the multiple input sound sources in a spatially distributed manner. It also provides basic room acoustic simulation controlled by the configuration interface.

Virtual acoustic scenes in TASCAR consist of a geometric image source model, containing sound sources, receivers and reflectors, and diffuse sound field model (see Figure 2 for an overview). The acoustic content can be taken from sound files, generator plugins or from external sound sources, e.g., microphones or other audio software. The acoustic modelling is performed in real time. The receiver type defines the render format. Possible render formats range from virtual omni-directional microphones to 3D Higher Order Ambisonics of arbitrary order.

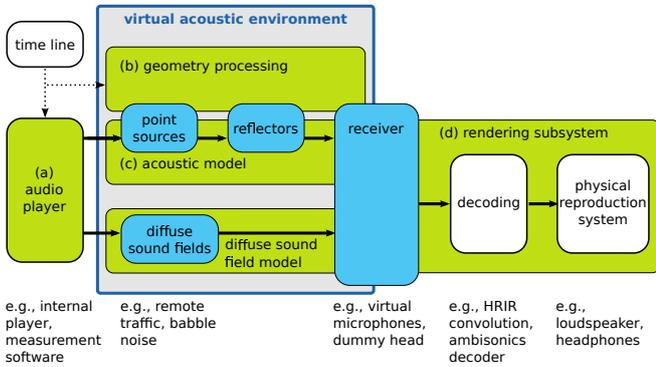


Figure 2: Components of TASCAR.

All objects – sources, receivers, reflectors and diffuse sound fields – can move along predefined trajectories or change their position in real time, e.g., controlled by head tracking systems or game controllers. Due to the time-domain implementation effects like the Doppler shift or time variant coloration of sounds are simulated using physical models. However, the time-domain implementation also results in a limited complexity of the virtual environments. Depending on the computer performance, approximately 200-1000 sound sources (including image sources) can be rendered simultaneously in real time. This means that the software is not suitable for room acoustic simulation of high accuracy. On a Raspberry Pi 4B in low-delay settings, only approximately 7 sound sources can be rendered next to the adaptive resampling and reverberation.

The ‘ov-client’ software (the core of ‘ovbox’) is a command line interface for headless operation, configured via a web interface. TASCAR comes with a command line interface and a GUI program. The GUI provides a map view (see Figure 3, a mixer view, a source code view (no editor), an OSC variable view, content license information, warnings and a news page.

### 3. INSTALLATION

The documentation of the ovbox, including installation instructions, can be found on github at

<https://github.com/gisogrimm/ovbox/wiki>.

The source code of the client is at

<https://github.com/gisogrimm/ov-client>.

Precompiled debian packages for Ubuntu LTS and Raspberry Pi OS are available (see Wiki for details).

### 4. REFERENCES

[1] Giso Grimm, “ORLANDOviols remote collaboration box (ovbox),” <https://github.com/gisogrimm/ovbox>, 2020.

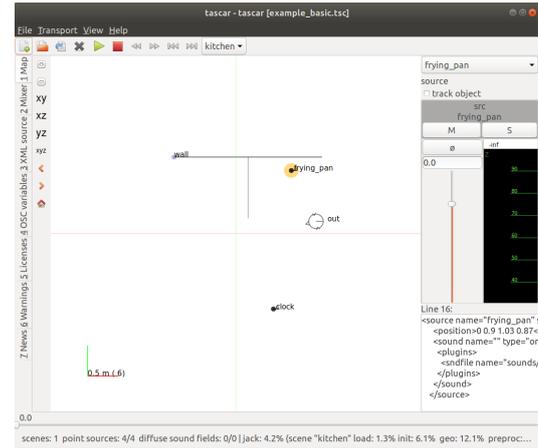


Figure 3: Map view in the main window of TASCAR.

[2] Fons Adriaensen, “Controlling adaptive resampling,” in *Linux audio conference*, Stanford, USA, 2012.

[3] Giso Grimm and Tobias Herzke, “A framework for dynamic spatial acoustic scene generation with Ambisonics in low delay realtime,” in *Proceedings of the Linux Audio Conference*, Frank Neumann, Ed., Stanford, CA, USA, 2012, Center for Computer Research in Music and Acoustics, Stanford University.

[4] Giso Grimm, Joanna Luberadzka, Tobias Herzke, and Volker Hohmann, “Toolbox for acoustic scene creation and rendering (TASCAR): Render methods and research applications,” in *Proceedings of the Linux Audio Conference*, Frank Neumann, Ed., Mainz, Germany, 2015, Johannes-Gutenberg Universität Mainz.

[5] Giso Grimm, Joanna Luberadzka, and Volker Hohmann, “A Toolbox for Rendering Virtual Acoustic Environments in the Context of Audiology,” *Acta Acustica united with Acustica*, vol. 105, no. 3, pp. 566–578, may 2019.