

An Introduction to the TU-Berlin WaveFieldSynthesis-System

1 Introduction

Wave Field Synthesis (WFS) is a new technique for sound spatialisation. It aims at a physical reconstruction of sound fields according to natural or artificial models by synthesizing the wavefronts of a defined virtual sound source (called primary source) with the superposition of wavefronts emitted by a closely spaced array of loudspeakers (called secondary sources). Thus, the spatial configuration of those virtual sound sources does not depend on a certain listener position (sweet spot), as it is the case with traditional stereophonic techniques such as two-channel stereo or surround. Moreover, while stereo setups allow the placement of sounds only on a line between the involved speakers and do not work well for lateral sources, WFS has no limitations concerning the placement of virtual sound sources outside and even inside of the reproduction array.



Fig. 1: TU Auditorium H104

The WellenFeld H104 in the main building of the TU Berlin is equipped with an array of 832 loudspeakers with 10 cm spacing in between. The rendering algorithm allows to generate two types of sound sources: point sources and plane waves. Point sources emit spherical wavefronts with a sound pressure decreasing by 6 dB per doubling of the distance. Plane waves are generated by point sources very far away. Due to the configuration of the loudspeaker array, they are approximated by cylindrical wave fronts exhibiting a sound pressure decreasing by 3 dB per doubling of the distance. Hence, if a sound should be reproduced at a certain position, use a point source. If a plane wave is used, the listener perceives only an angle where the sound is coming from but no distance. Plane waves can be used when no location should be noticed, i. e. for reproducing diffuse reverb.

Note that for sound sources inside the room (called focused sources), the sound is traveling from the speaker to the source position (marked red in the picture). When a listener is located between the speakers and the sound source, the focused sound source will not be located

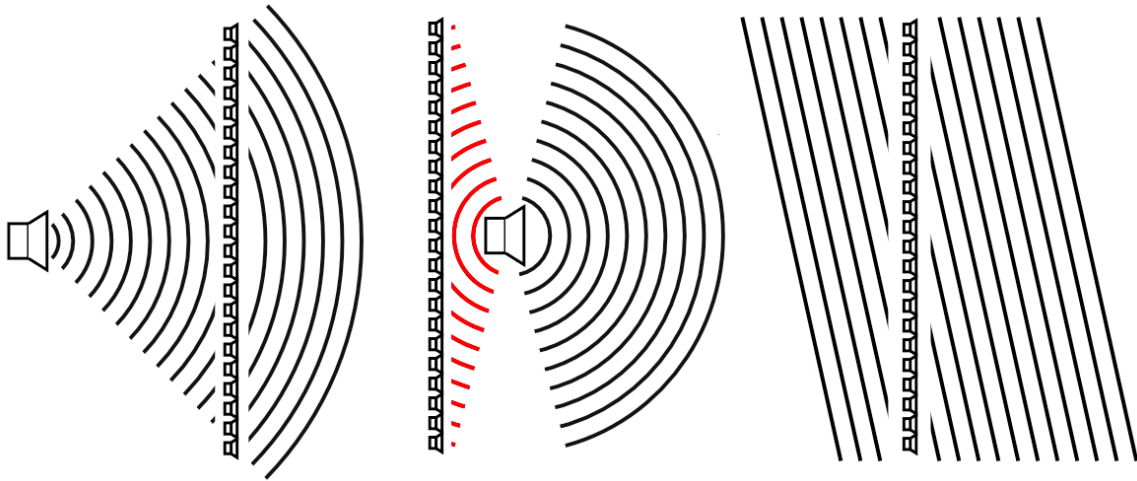


Fig. 2: Possible source types: Point source behind the speaker array, point source in front of the speaker array, plane wave. The large speaker is the virtual source. The small speakers are physically present in the lecture hall.

correctly because the wavefront travels towards the intended location rather than from this point. When moving focused sources across the center of the room, the array of speakers used for playback will change from one side of the room to the other, producing a noticeable change in sound incidence direction.

Depending on the distance between the speakers and the position of both the virtual source and the listener, the synthesized wavefield will be incorrect for higher frequencies above ca. 3000 Hz due to spatial aliasing. Depending on the audio content, this artifact can become audible as ambiguous location, spectral coloration or temporal smearing of transient sounds, particularly for focused sources.

Point sources can not only be assigned to fixed locations, they can also be moved in realtime. However, fast movements (faster than walking speed) can cause artifacts. Plane wave sources should currently not be moved while playing sound on those sources.

2 Working in the H104

When working on two channel or multichannel stereo spatialisations, using a sequencer like Pro-Tools, pan pots are used to position sounds in the stereo panorama. Finally, a master file (two channel or multichannel) will be produced for playback on any other stereo system.

WFS follows a different strategy. Since producing a master file would require 832 audiotracks to be saved for playback in H104, and since other WFS systems have different loudspeaker numbers and array configurations, the signals for each individual speaker are computed in realtime. Only the original audio tracks (sources) are saved, along with the positions and movements for each virtual source in a separate file.

The system can currently synthesize up to 42 virtual sound sources. For each source you can define a sourcetype, position and whether the doppler shift is synthesized or not. Plane waves have their angle as an additional parameter. The volume for each individual source has to be controlled within the sequencer used for playback of audio sources rather than within the or WFS rendering software.

2.1 Using Ardour and xWonder

While WFS rendering is computed on a Linux PC cluster, we use Ardour as an open source sequencer to play back audio source tracks. This software is quite simple compared to ProTools and other common commercial programs. It has complex routing possibilities and allows basic filtering, volume control and automation. The editing of sound material is, however, not very comfortable in Ardour and should be done before starting to work on the WFS system. Both AIF and WAF files can be imported.

A graphical user interface called xWonder can be synchronized to Ardour. It allows to record and play back movements of single and grouped soundsources. The workflow is comparable to automating a plugin in a sequencer. The geometry of the room as well as source positions and movements are shown. Note that the looping of audio and spatialisation is not yet possible in the software.

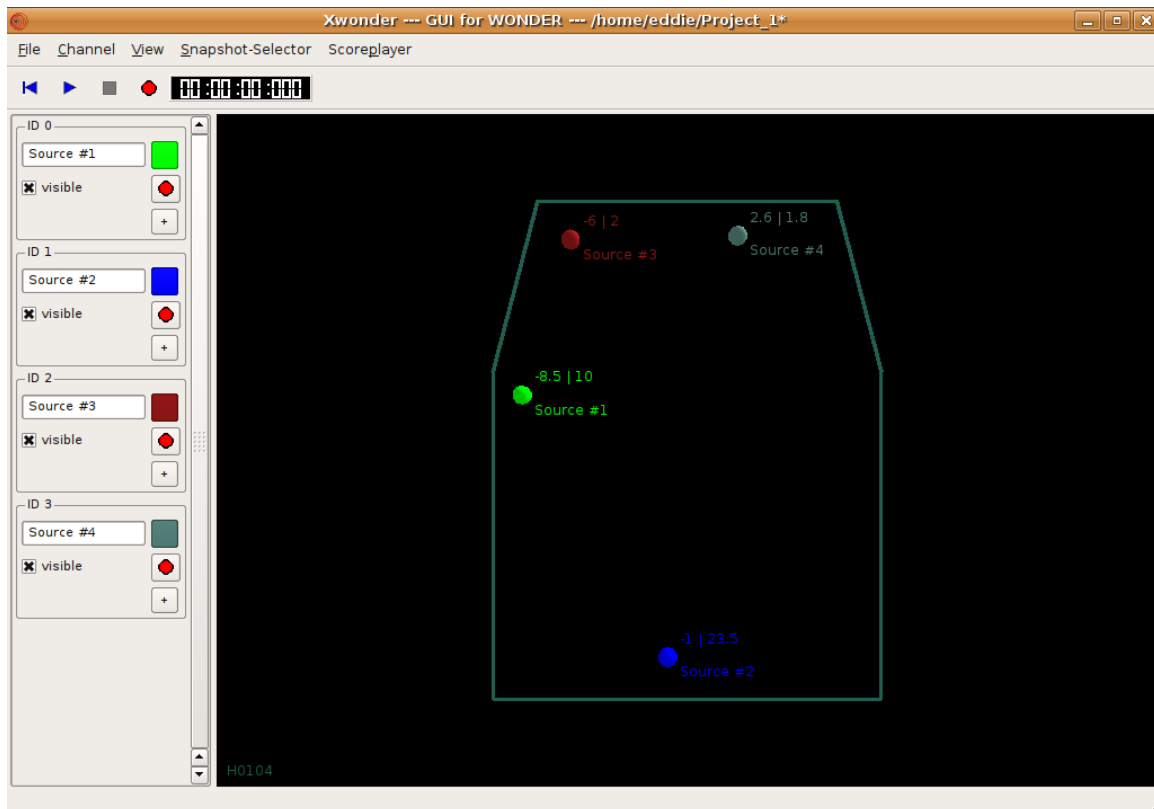


Fig. 3: The graphical user interface: xWonder

2.2 Using your Max/MSP patch

You can also control the WFS-system using OSC messages generated by programs such as Max/MSP, SuperCollider or PureData. In this case, audio can be fed via ADAT or MADI into the WFS computers. Note that the system has a latency of about 100ms. You can find the OSC commands and the room coordinates in the appendix. Note that the enumeration of the sources starts at 0. For a setup with 8 point sources located in a line in front of the room, the required messages are:

```

/WONDER/project/createWithoutScore myProjectName
/WONDER/source/activate           0
/WONDER/source/activate           1
/WONDER/source/activate           2
/WONDER/source/activate           3
/WONDER/source/activate           4
/WONDER/source/activate           5
/WONDER/source/activate           6
/WONDER/source/activate           7
/WONDER/source/position           0 -3.5 2 0 0
/WONDER/source/position           1 -2.5 2 0 0
/WONDER/source/position           2 -1.5 2 0 0
/WONDER/source/position           3 -0.5 2 0 0
/WONDER/source/position           4 0.5 2 0 0
/WONDER/source/position           5 1.5 2 0 0
/WONDER/source/position           6 2.5 2 0 0
/WONDER/source/position           7 3.5 2 0 0

```

2.3 A couple of things to keep in mind

- You can synthesize up to 42 sources.
- Do not move the sources too fast.
- Plane wave sources should not be moved.
- You can not control the volume of individual sources within the WFS software.
- Moving sound across the speaker from the inside of the room to the outside or vice versa causes a modulation.
- Placement of sources is only possible in a two dimensional plane. Elevation of soundsources is not possible.
- A listener situated between the speakers and a focused virtual source won't perceive the position of the sound correctly.

3 Appendix

3.1 The room coordinates

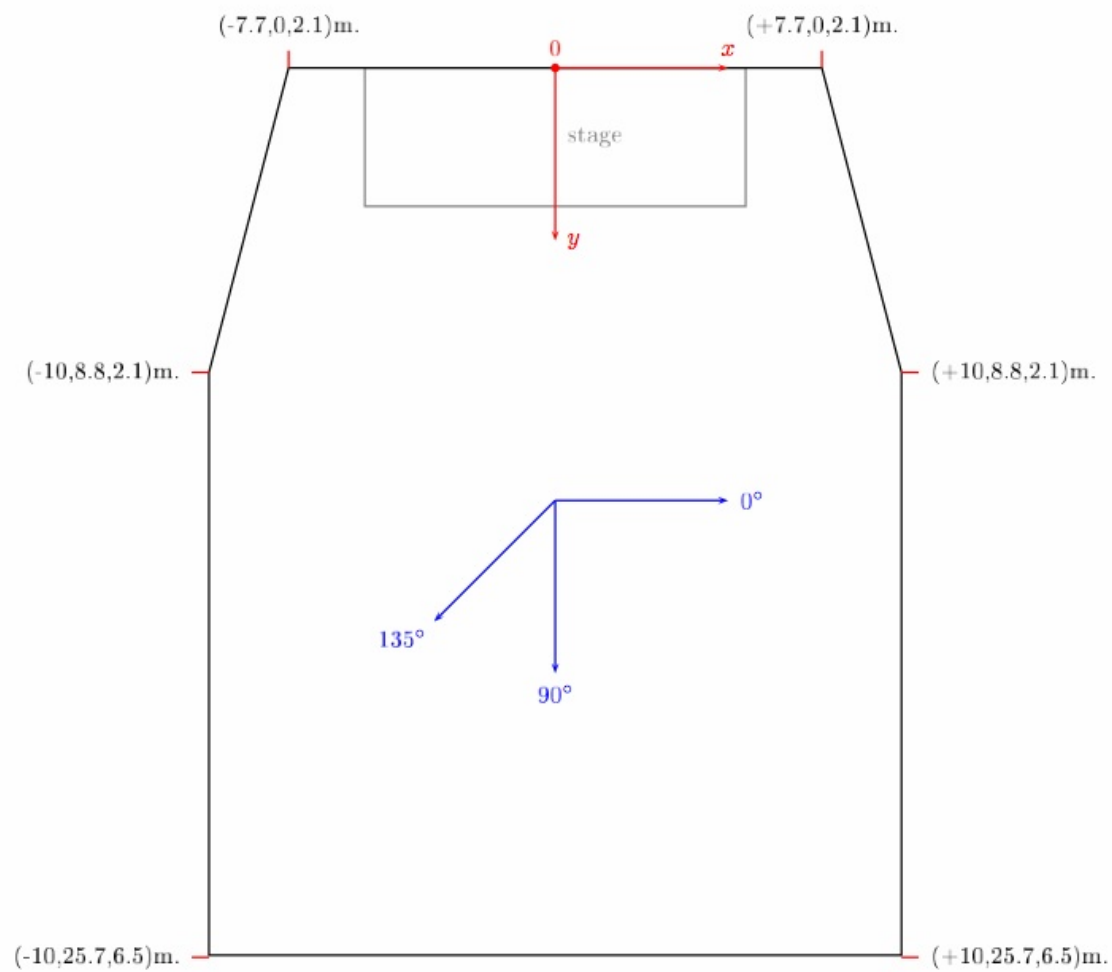


Fig. 4: The coordinates of H104

3.2 The OSC Commands

OSC-Path	Datatypes	Description
/WONDER/source/activate	i	id
/WONDER/source/deactivate	i	id
/WONDER/source/type	i i	id, type [0=plane, 1=point]
/WONDER/source/type	i i f	id, type [0=plane, 1=point], timestamp [seconds]
/WONDER/source/angle	i f	id, angle [degrees]
/WONDER/source/angle	i f f	id, angle [degrees], duration [seconds]
/WONDER/source/angle	i f f f	id, angle [degrees], duration [seconds], timestamp [seconds]
/WONDER/source/position	i f f	id, x coordinate [meters], y coordinate [meters]
/WONDER/source/position	i f f f	id, x coordinate [meters], y coordinate [meters], duration [seconds]
/WONDER/source/position	i f f f f	id, x coordinate [meters], y coordinate [meters], duration [seconds], timestamp [seconds]
/WONDER/source/position	i f f f f f	id, x coordinate [meters], y coordinate [meters], not Used, duration [seconds], timestamp [seconds]
/WONDER/source/dopplerEffect	i i	id, on [0 = false, 1 = true]
/WONDER/project/createWithScore	s	projectname
/WONDER/project/createWithoutScore	s	projectname
/WONDER/project/addScore		
/WONDER/project/load	s	projectname
/WONDER/project/save		
/WONDER/project/save	s	projectname
/WONDER/project/snapshot/take	i	snapshotId
/WONDER/project/snapshot/take	i s	snapshotId, name
/WONDER/project/snapshot/recall	i f	snapshotId, duration [seconds]
/WONDER/project/snapshot/delete	i	snapshotId
/WONDER/project/snapshot/rename	i s	snapshotId, name
/WONDER/project/snapshot/copy	i i	snapshotId (from), snapshotId (to)
/WONDER/score/play		
/WONDER/score/stop		
/WONDER/score/setStartScenario		
/WONDER/score/enableRecord	i	on [0 = false, 1 = true]
/WONDER/score/enableRead	i	on [0 = false, 1 = true]
/WONDER/score/reset		
/WONDER/score/newtime	i i i i	hours, minutes, seconds, milliseconds
/WONDER/score/enableMMC	i	on [0 = false, 1 = true]
/WONDER/score/enableMSRC	i	on [0 = false, 1 = true]
/WONDER/score/source/enableRecord	i i	id, on [0 = false, 1 = true]
/WONDER/score/source/enableRead	i i	id, on [0 = false, 1 = true]