swonder3Dq: Auralisation of 3D objects with Wave Field Synthesis

Marije BAALMAN
Institute for Communication Sciences, Technische Universität Berlin
Einsteinufer 17
Berlin, Germany
baalman@kgw.tu-berlin.de

Abstract

swonder3Dq is a software tool to auralise three dimensional objects with Wave Field Synthesis. It presents a new approach to model the radiation characteristics of sounding objects.

Keywords

Wave field synthesis, spatialisation, radiation characteristic, physical modelling

1 Introduction

Wave Field Synthesis (WFS) is an interesting method for spatialisation of electronic music. Its main advantage is that it has no sweet spot, but instead a large listening area, making the technology attractive for concert situations.

Yet, whenever acoustic instruments are combined with electroacoustic sounds in concerts, a common problem is that the electroacoustic part tends to lack depth and extension in comparison with the sound from the acoustic instruments. Misdariis (1) and Warusfel (2) have proposed special 3D loudspeaker arrays to simulate different radiation characteristics. A problem with this technique is that the object itself is static; a movement of the source can only be created by physically moving the loudspeaker array.

In current Wave Field Synthesis applications there are only solutions for monopole point sources and plane waves. There are some reports of ad hoc solutions to simulate larger sources, such as the Virtual Panning Spots (3) and the auralisation of a grand piano (4) by using a few point sources. Currently, work is done on implementing radiation characteristics for point sources (5). By definition radiation characteristics are only applicable at a certain distance from the object modelled. Since WFS sources can get very close, it makes sense to look for a general solution for auralising arbitrarily shaped sources.

2 Theory

2.1 Source model

The wave field of a sounding object can be approximated by superposition of the wave fields of a number of point sources. The locations of these point sources and the number can be chosen arbitrarily, but it make sense to choose the locations on the surface of the object. This separates the calculation of the vibration of the object itself (which can be done with various methods, such as finite element methods or modal synthesis) from the calculation of the radiation of the sound from the object into the air.

For a correct calculation of the radiated sound field, the vibration of the surface must be spatially sampled. Here there is the danger of undersampling (6); this danger is twofold: first, spatial amplitude variations of the surface vibration may be lost, and secondly, depending on the frequency content of the sound, spatial aliasing may occur in a similar way as for the WFS speaker array itself.

In practice the sound emitted from different points on the object's surface will be correlated. For a practical implementation it is useful to assume that for a point Ψ on the surface:

$$S_{\Psi}(\vec{r}_{\Psi}, \phi, \theta, \omega) = S(\omega)G(\vec{r}_{\Psi}, \phi, \theta, \omega) \quad (1)$$

i.e. from each point source that is part of the distribution a source signal $S(\omega)$ filtered with G is emitted (G depending on the location $\vec{r}_{\Psi}$ of the point source Ψ, the angular frequency ω of the sound and the direction angles ϕ and θ). Applied to the reproduction of electronic music, this will allow a composer to determine the filter characteristics of his source object and the sound signal emitted by the source independently. Thus, the resulting filtering function for each speaker can be determined in advance and the signal input can be convolved in realtime with this filter.

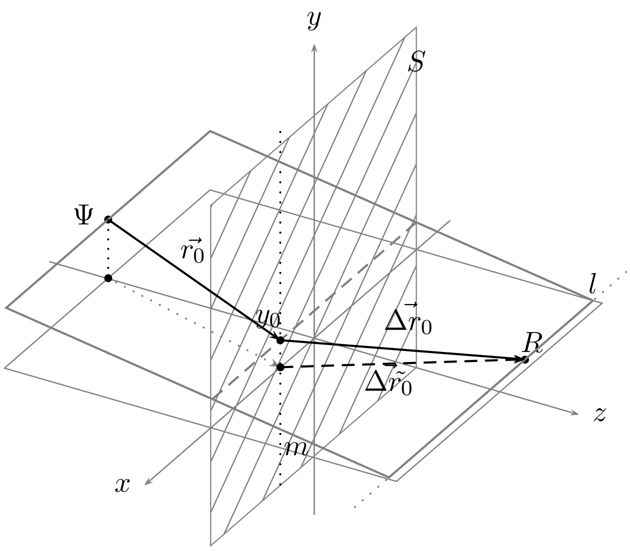

Figure 1: The stationary point $(x_m, y_0, 0)$ lies on the cross-section of the line m (in plane S) and the plane through Ψ and the reference line l.

2.2 Adaptation of the WFS-operator

For WFS reproduction of a 3-dimensional source object the commonly used $2\frac{1}{2}D$-operator (7) is not sufficient as its derivation only sources in the same plane as the array and reference line are taken into account. In (8) the WFS operator for points outside of the horizontal plane was derived, starting from the Rayleigh integrals. The main difference from the $2\frac{1}{2}D$-operator is the calculation of the stationary point:

$$y_0 = y_R + (y_\Psi - y_R)\frac{z_R}{z_\Psi + z_R} \quad (2)$$

where z_R is the z-coordinate of the receiver and z_Ψ of the source point (see also figure 1). The driver function of a speaker for the contribution of one monopole source point becomes:

$$Q(x,\omega) = S(\omega)\sqrt{\frac{jk}{2\pi}}\sqrt{\frac{\Delta r_0}{\Delta r_0 + r_0}}cos(\phi_0)\frac{e^{-jkr_0}}{\sqrt{r_0}} \quad (3)$$

the speaker being assumed to be on the x-axis.

It should be noted that the actual elevation will not be heard, when the elevated source is played back by the WFS-array. The elevated points are mainly of interest, because their contributions will interfere with those of the points in the horizontal plane.

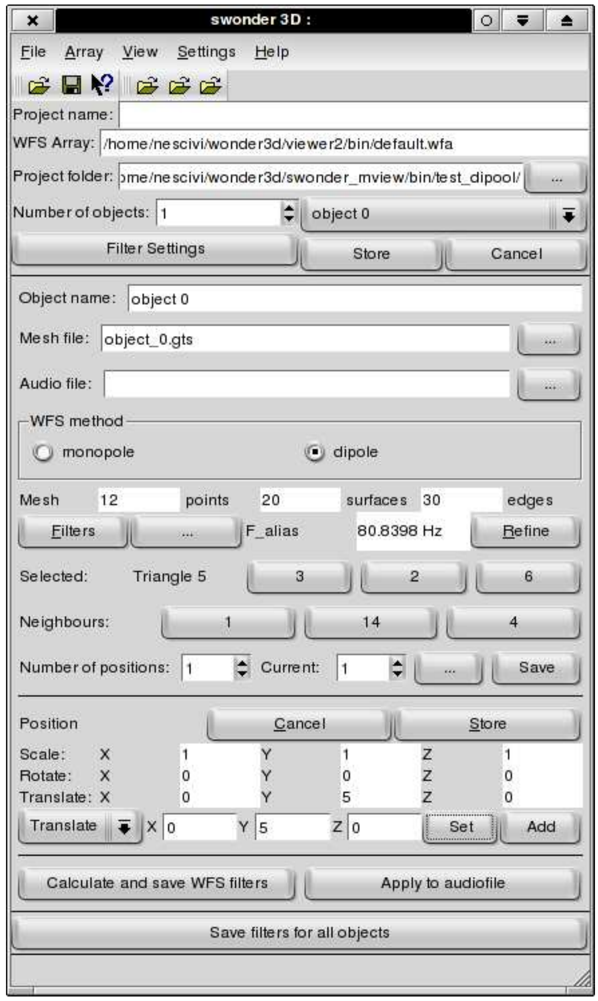

Figure 2: Snapshot of the graphical user interface of *swonder3Dq*

3 Implementation

The software enables to calculate the filters for WFS reproduction for several sound objects. The objects themselves are defined by their geometrical data and the radiation filters at several points on the surface. Objects can be positioned and given different orientations in space. Figure 2 is a snapshot of the graphical user interface. There is an OpenGL viewer included to look at the object. The user can choose between defining filters of each node, or choose a multichannel audiofile which has the sound for each node (e.g. calculated with *Modalys* (9)). The program then calculates the WFS operators or the loudspeaker signals and saves them to disk.

A second part of the software enables the user to load a project and listen to the desired object on the desired location with the desired orientation. This part of the software can be controlled with OSC. *BruteFIR* (10) is used as the convo-

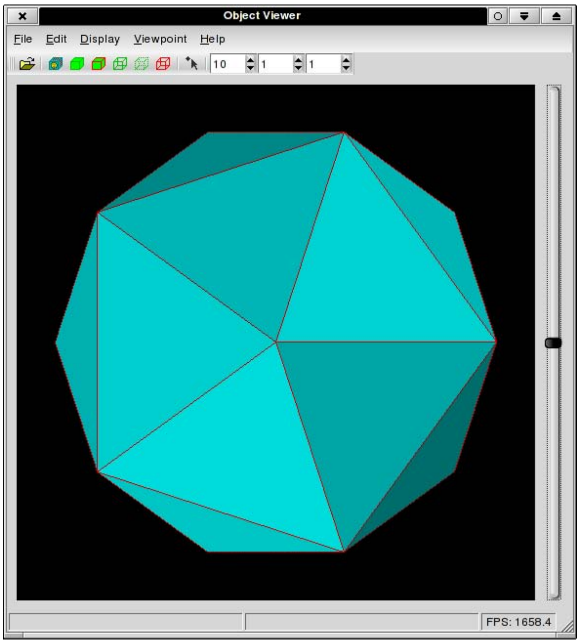

Figure 3: Graphical display of an object with *mview*

lution engine.

3.1 Geometry

There are a multitude of programs and libraries available for manipulating and visualising 3D data. The *GTS-library* (11) encompasses many functions to read, write and work with meshes. A disadvantage of this library is that the points on the mesh (the vertices) are not ordered or tagged while they are loaded, so it is not possible to connect the data for the radiation filters to them.

In the program *mview* (12) a lot of methods for working with meshes were already implemented and with only a few additions to implement the filter definition per source point, it was included into *swonder3Dq* for the graphical display of the objects.

GeomView (13) was used as a second viewer to view the whole scene: the WFS speaker array as well as several sounding objects. *GeomView* is used as an external program, interfaced via stdin and stdout.

3.2 Filter definition and calculation

There is a simple graphical representation of the frequency response of the filter, where the user can define breakpoints (figure 4). The filter settings can be copied between source points and there is an interaction between the picked triangle and its corner points (which can be selected in the main gui) and the current source point

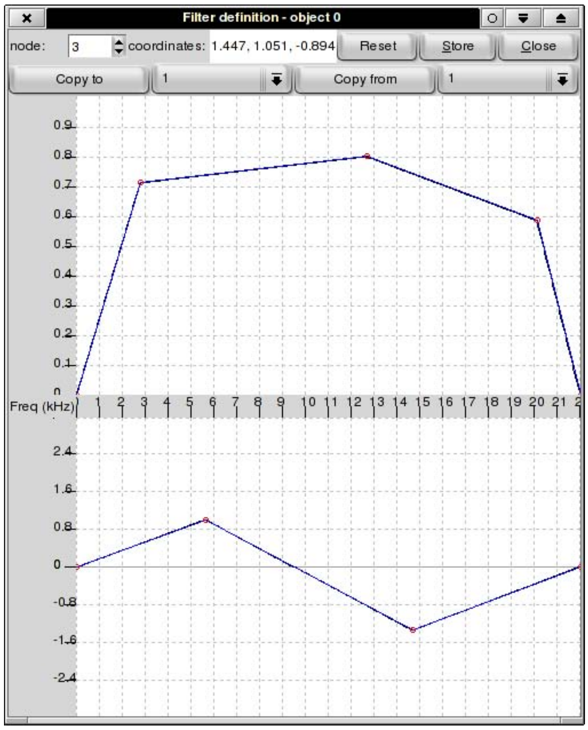

Figure 4: Snapshot of the dialog to define a filter for a point on the source object surface

for which a filter is defined. It is also possible to load filters from file.

The filter is then calculated based on the defined breakpoints with a method as described in (14)[1]. For the fast fourier transform the *FFTW Library* is used (15).

3.3 Refinement of the surface

To make the discretisation distance smaller, an algorithm is needed to calculate more points on the object surface. A simple method (partly taken from the *GTS Library*) is the midvertex insertion method. On each edge of a triangle, a point is added in the middle to divide the edge in two. Then, every midpoint is connected to those of the other edges (figure 5). This method can be applied more than once, to create a fine raster of points. The aliasing frequency (6) can be calculated by finding the longest edge in the mesh and calculate the corresponding aliasing frequency.

Secondly the filter for the newly calculated points needs to be determined. This is done by an average of the filter of the neighbouring points, using an inverse distance weighting method (16) to determine the contribution of each point:

[1]Chapter 17, pages 297-300

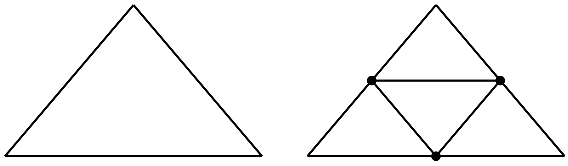

Figure 5: Refinement of a triangle with the *mid-vertex insertion* method

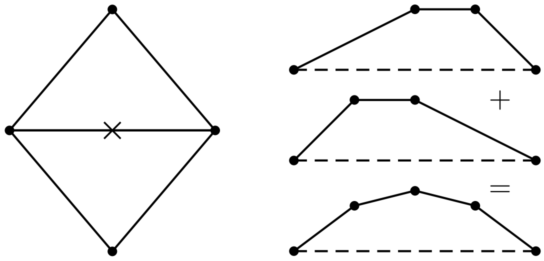

Figure 6: Averaging of the filter values. On the left is shown from which point the average is taken: $\times$ is the new point, $\bullet$ are the neighbourpoints. On the left is shown how the breakpoints are added when averaging.

$$Z_j = \frac{\sum_{i=1}^{n} \frac{Z_i}{h_{ij}^{\beta}}}{\sum_{i=1}^{n} \frac{1}{h_{ij}^{\beta}}} \tag{4}$$

Z_j is the value of the new point j, Z_i of the neighbour point i, h_{ij} the distance from point i to j, and β a factor that defines the weighting of the distance, usually set to $\beta = 2$. In this case the factor β can be interpreted as a kind of measure how well the sound is propagated through the material of the object.

The averaging between two filters is calculated as follows (see also figure 6): for each breakpoint from either filter the corresponding value on that frequency value is calculated for the other filter. The new filter then has a breakpoint value at that frequency value, which is an average of the two breakpoints of the two filters. The average is taken from the real and imaginary parts of the coefficients.

3.4 3D WFS Calculation

Steps in the calculation of the WFS filters (for each object, at each location specified):

1. Per source point:
   - Check: is the source point audible (visible) for the loudspeaker?
   - Calculation of delay and volume factor according to the WFS operator
   - Convolution of the WFS-delay and volume with the filter for that source point
2. Addition of all filters of the source for each loudspeaker
3. Save filter to disk

As the software is still in an experimental state, there is a choice between defining the source points as monopoles or as dipoles. In the case of a dipole, the main axis of radiation is in the direction of the normal (pointing outwards) on the surface; in the case of points on the corners of triangles which are not on the same plane, this normal is an average of the normals of the triangles it is a corner point of.

When a source point is at the back side of the object, a direct path of the sound to the loudspeaker is not possible and this source point should not be taken into account. Diffraction of sound waves around the object is neglected in this case.

4 First tests

Preliminary listening tests on a frontal WFS-speaker array of 24 loudspeakers (at 12.5cm distance) show that this approach does give a stronger spatial impression of a sound source.

However, it became apparent that the neglection of diffraction cannot be allowed. As for different speakers, different points of the object will be at the backside (figure 7), sound will arrive from speakers to parts of the listening area that should be obscured when neglecting diffraction. Thus some sense of diffraction is created, but at a much later stage than should

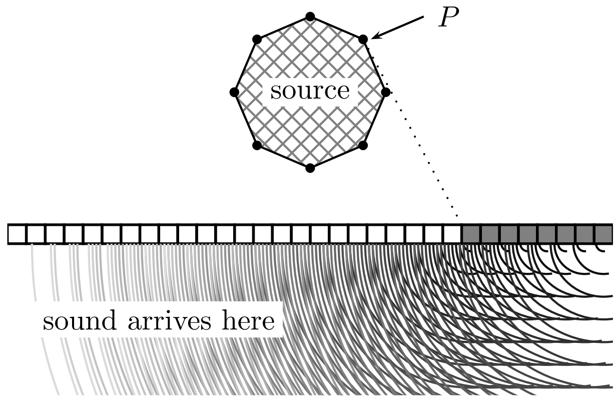

Figure 7: Illustration of the audibility problem. The point P is only audible for the gray speakers on the right, yet the sound from these speakers will arrive on the left side of the listening area.

be. We will need to find a different approach in order to take the diffraction of the sound around the object into account properly.

5 Conclusions and future work

An extension of the WONDER software (17) was presented which enables the calculation of WFS auralisation for complex sound sources. While its first aim is to be used for WFS auralisation, several concepts introduced could be used in other 3D audio applications, such as binaural representation.

Further research will be done on how to take the diffraction of waves around objects into account, before implementing that feature. Future work will include listening experiments, as well as doing usability tests by working with composers using the software.

6 Acknowledgements

The work on *swonder3Dq* is part of a Ph.D. research project funded by the *NaFöG* in Berlin, Germany.

7 References

[1] N. Misdariis, O. Warusfel & R. Caussé. Radiation control on a multi-loudspeaker device. In *ISMA 2001*, 2001.

[2] N. Misdariis & T. Caulkins O. Warusfel, E. Corteel. Reproduction of sound source directivity for future audio applications. In *ICA 2004*, 2004.

[3] G. Theile, H. Wittek & M. Reisinger. Wellenfeld-synthese verfahren: Ein weg für neue möglichkeiten der räumlichen tongestaltung. In *22nd Tonmeistertagung, Hannover, Germany, 2002 November*, 2002.

[4] I. Bork, & M. Kern. Simulation der schallabstrahlung eines flügels. In *DAGA '03*, 2003.

[5] O. Warusfel & N. Misdariis. Sound source radiation synthesis: from stage performance to domestic rendering. In *AES 116th Convention, Berlin, Germany, 2004, May, Preprint 6018*, 2004.

[6] M.A.J. Baalman. Discretisation of complex sound sources for reproduction with wave field synthesis. In *DAGA '05, 14 - 17 March 2005, München*, 2005.

[7] E.N.G. Verheijen. *Sound Reproduction by Wave Field Synthesis*. PhD thesis, TU Delft, The Netherlands, 1998.

[8] M.A.J. Baalman. Elevation problems in the auralisation of sound sources with arbitrary shape with wave field synthesis. In *ICMC 2005, 1-6 September 2005, Barcelona, Spain*, 2005.

[9] IRCAM. Modalys. http://www.ircam.fr/, 1991-2005.

[10] A. Torger. Brutefir. http://www.ludd.luth.se/∼torger/brutefir.html, 2001-2005.

[11] Gnu triangulated surface library. http://gts.sourceforge.net/, 2000-5.

[12] Helmut Cantzler. Mesh viewer. http://mview.sourceforge.net/, 2001-5.

[13] Geometry Technologies. Geomview. http://www.geomview.org/, 1992-2005.

[14] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1997.

[15] Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. special issue on "Program Generation, Optimization, and Platform Adaptation".

[16] M.L. Green. *GEOGRAPHIC INFORMATION SYSTEM BASED MODELING OF SEMI-VOLATILE ORGANIC COMPOUNDS TEMPORAL AND SPATIAL VARIABILITY*. PhD thesis, University of New York at Buffalo, 2000.

[17] M.A.J. Baalman. Updates of the wonder software interface for using wave field synthesis. In *3rd International Linux Audio Conference, April 21-24, 2005, ZKM, Karlsruhe*, 2005.