Universität Rostock Traditio et Innovatio

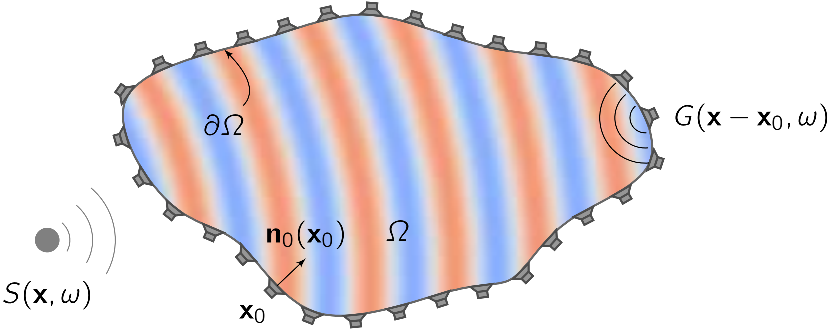
Time-Domain Realisations of 2.5-Dimensional Local Sound Field Synthesis

Fiete Winter and *Sascha Spors*

Universität Rostock
Institut für Nachrichtentechnik

DAGA 2017
09. März 2017, Kiel

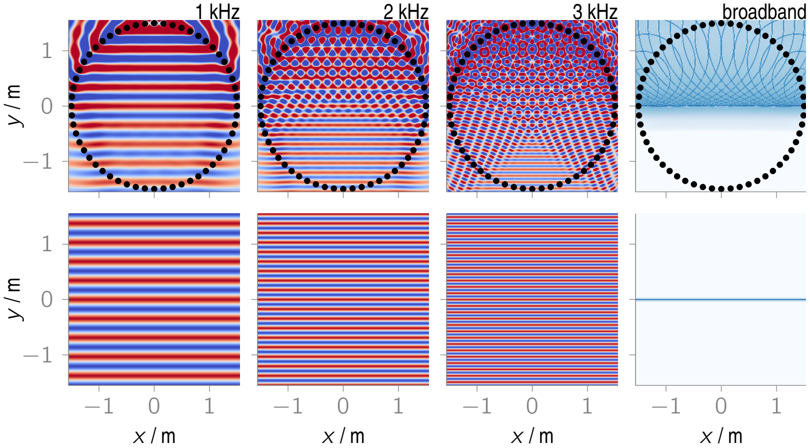
Sound Field Synthesis



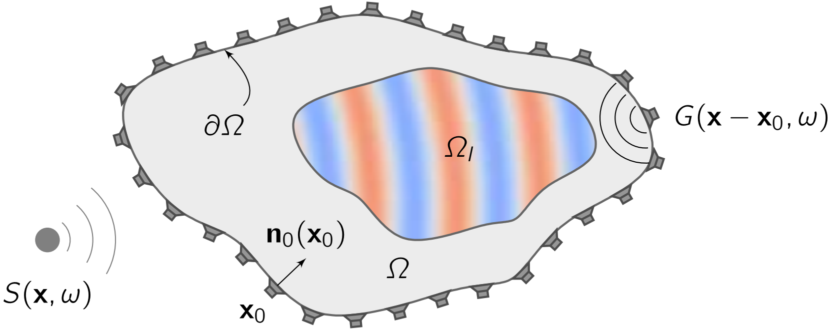
$$S(\mathbf{x},\omega) \stackrel{!}{=} \oint_{\partial\Omega} \underbrace{D(\mathbf{x}_0,\omega)}_{\text{driving signals}} \underbrace{G(\mathbf{x}-\mathbf{x}_0,\omega)}_{\text{loudspeaker}} \,\mathrm{d}A_0(\mathbf{x}_0) \quad \forall \mathbf{x} \in \Omega$$

Sound Field Synthesis

Spatial Aliasing



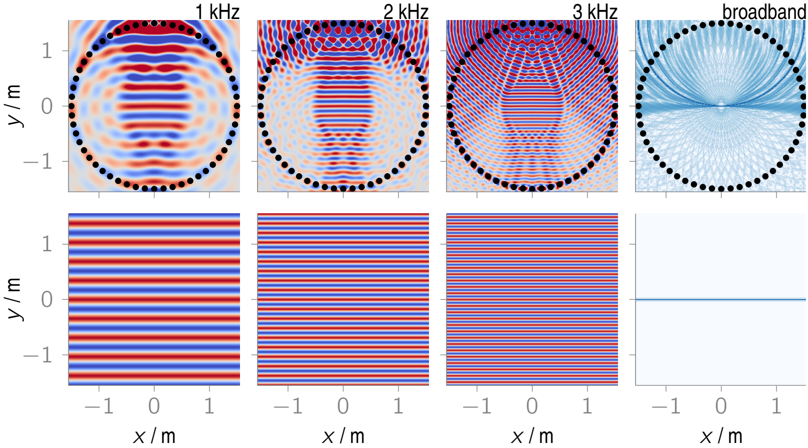
Local Sound Field Synthesis



$$S(\mathbf{x},\omega) \stackrel{!}{=} \oint_{\partial\Omega} \underbrace{D(\mathbf{x}_0,\omega)}_{\text{driving signals}} \underbrace{G(\mathbf{x}-\mathbf{x}_0,\omega)}_{\text{loudspeaker}} \, \mathrm{d}A_0(\mathbf{x}_0) \quad \forall \mathbf{x} \in \Omega_l$$

Local Sound Field Synthesis

Spatial Aliasing



Agenda

Wave Field Synthesis

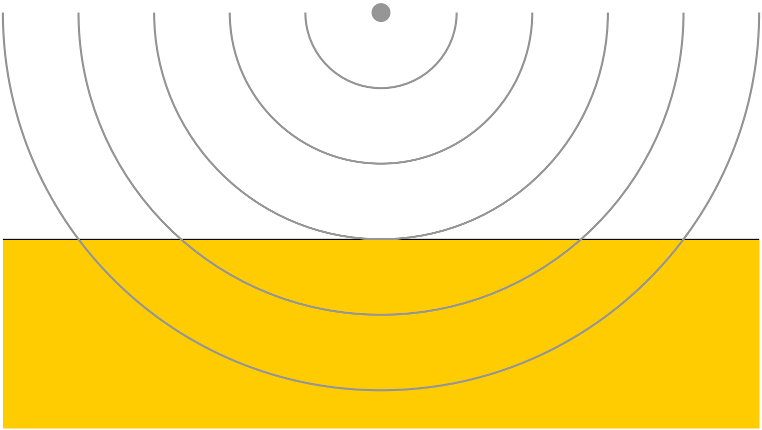
- basic concept
- time domain realisation

Local Wave Field Synthesis using Virtual Secondary Sources

- basic concept
- time domain realisation

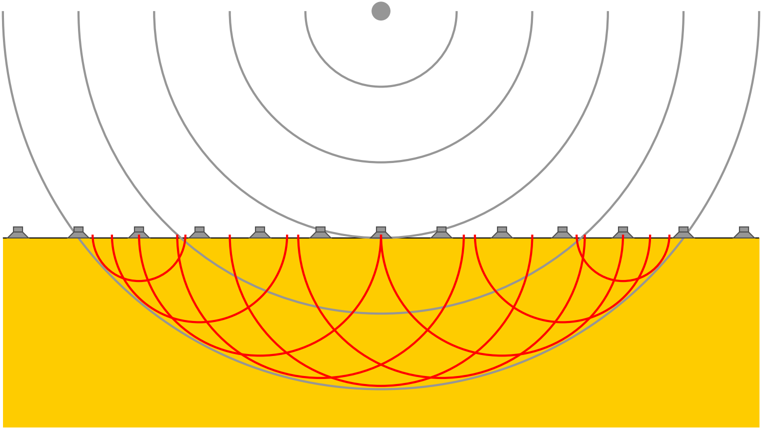
Wave Field Synthesis

Huygens–Fresnel principle



Wave Field Synthesis

Huygens–Fresnel principle



Wave Field Synthesis

Driving Signals - Model Based Rendering

Point source (at position $\mathbf{x}_{\mathrm{ps}}$)

$$D_{\mathrm{ps}}^{\mathrm{WFS}}(\mathbf{x}_0,\omega) = \frac{\sqrt{\frac{\mathrm{j}\omega}{\mathrm{c}}}}{\sqrt{2\pi}} \frac{\langle \mathbf{x}_0 - \mathbf{x}_{\mathrm{ps}} | \mathbf{n}_0(\mathbf{x}_0)\rangle a_{\mathrm{ps}}(\mathbf{x}_0)\sqrt{|\mathbf{x}_0 - \mathbf{x}_{\mathrm{ref}}|}}{|\mathbf{x}_0 - \mathbf{x}_{\mathrm{ps}}|^{3/2}\sqrt{|\mathbf{x}_0 - \mathbf{x}_{\mathrm{ps}}| + |\mathbf{x}_0 - \mathbf{x}_{\mathrm{ref}}|}} \mathrm{e}^{-\mathrm{j}\frac{\omega}{\mathrm{c}}|\mathrm{x}_0 - \mathrm{x}_{\mathrm{ps}}|}$$

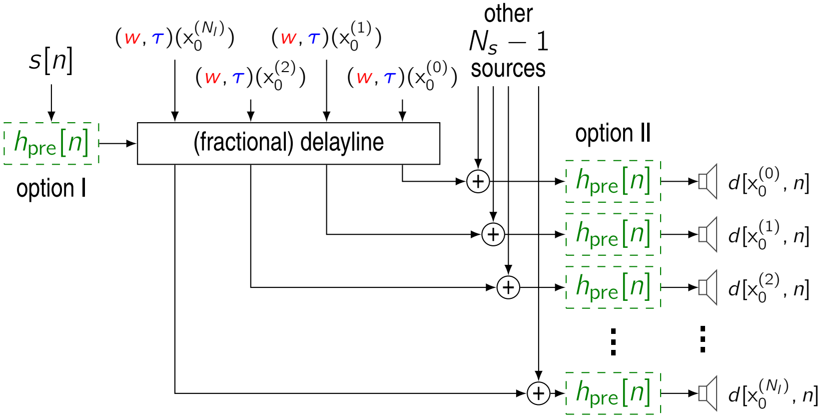
Plane wave (propagating in the direction of $\mathbf{n}_{\mathrm{pw}}$)

$$D_{\mathrm{pw}}^{\mathrm{WFS}}(\mathbf{x}_0,\omega) = \sqrt{\frac{\mathrm{j}\omega}{\mathrm{c}}}\sqrt{8\pi|\mathbf{x}_0 - \mathbf{x}_{\mathrm{ref}}|}\, a_{\mathrm{pw}}(\mathbf{x}_0)\langle \mathbf{n}_{\mathrm{pw}} | \mathbf{n}_0(\mathbf{x}_0)\rangle \mathrm{e}^{-\mathrm{j}\frac{\omega}{\mathrm{c}}\langle \mathrm{x}_0 | \mathrm{n}_{\mathrm{pw}}\rangle}$$

1. geometry independent pre-filter $H_{\mathrm{pre}}(\omega)$
2. geometry dependent weighting $w(\mathbf{x}_0)$
3. geometry dependent delaying $\tau(\mathbf{x}_0)$

Wave Field Synthesis

Time-Domain Realisation, e.g. SoundScape Renderer



I. $c_{\text{I}}^{\text{WFS}}(N_s, N_\text{l}) = N_s c_{\text{conv}}^{\text{pre}} + N_s c_{\text{write}}^{\text{dl}} + N_s N_\text{l} c_{\text{read}}^{\text{dl}}$

II. $c_{\text{II}}^{\text{WFS}}(N_s, N_\text{l}) = N_s c_{\text{write}}^{\text{dl}} + N_s N_\text{l} c_{\text{read}}^{\text{dl}} + N_\text{l} c_{\text{conv}}^{\text{pre}}$

Delaylines

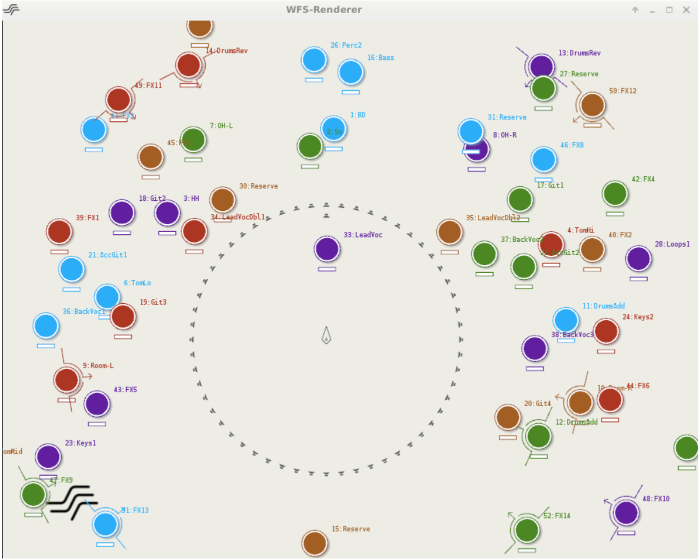
Integer Delays

- \+ sufficient for stationary scenarios
- – likely to cause artefacts for dynamic scenarios
- \+ low $c^{\mathrm{dl}}_{\mathrm{write}}$, low $c^{\mathrm{dl}}_{\mathrm{read}}$

Delay Interpolation (two options)

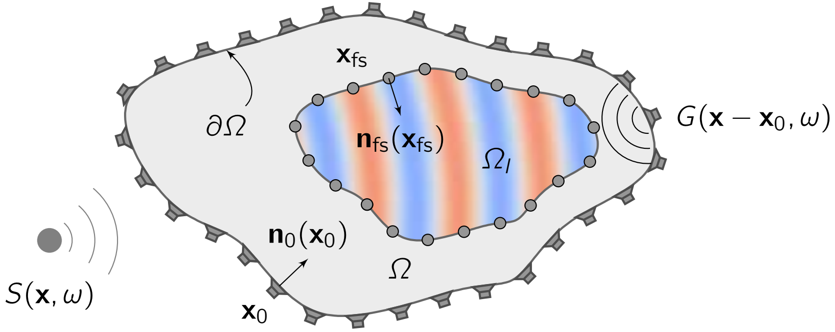
1. interpolation upon request
   - \+ no additional memory required
   - \+ low $c^{\mathrm{dl}}_{\mathrm{write}}$
   - – high $c^{\mathrm{dl}}_{\mathrm{read}}$
2. delay independent preprocessing, e.g. oversampling about factor R
   - – R times more memory required
   - – high $c^{\mathrm{dl}}_{\mathrm{write}}$
   - \+ low $c^{\mathrm{dl}}_{\mathrm{read}}$

Example


WFS-Renderer
14:DrumsRev
26:Perc2
16:Bass
13:DrumsRev
27:Reserve
49:FX11
50:FX12
7:OH-L
1:BD
31:Reserve
8:OH-R
46:FX8
42:FX4
17:Git1
30:Reserve
18:Git2
3:HH
39:FX1
35:LeadVocDbl2
33:LeadVoc
37:BackVoc
4:TomHi
40:FX2
28:Loops1
21:AccGit1
6:TomLo
19:Git3
36:BackVoc1
11:DrumsAdd
24:Keys2
38:BackVoc3
9:Room-L
43:FX5
44:FX6
20:Git4
12:DrumsAdd
23:Keys1
47:FX9
51:FX13
15:Reserve
52:FX14
48:FX10

Local WFS using Virtual Secondary Sources

Basic Principle



- focused sources used as virtual secondary sources on $\partial\Omega_l$
- virtual secondary sources driven as real loudspeakers

! choose $N_{\mathrm{fs}} \gg N_{\mathrm{l}}$ to avoid additional aliasing

Local WFS using Virtual Secondary Sources

Driving Signal

Focused source (at position $\mathbf{x}_{\mathrm{fs}}$)

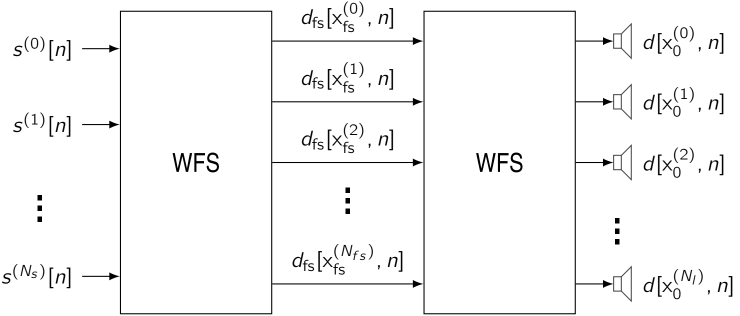
$$D_{\mathrm{fs}}^{\mathrm{WFS}}(\mathbf{x}_0, \mathbf{x}_{\mathrm{fs}}, \omega) = \frac{\sqrt{\frac{-\mathrm{j}\omega}{c}}}{\sqrt{2\pi}} \frac{\langle \mathbf{x}_{\mathrm{fs}} - \mathbf{x}_0 | \mathbf{n}_0(\mathbf{x}_0) \rangle a_{\mathrm{fs}}(\mathbf{x}_0) \sqrt{|\mathbf{x}_{\mathrm{ref}} - \mathbf{x}_0|}}{|\mathbf{x}_{\mathrm{fs}} - \mathbf{x}_0|^{3/2} \sqrt{\big||\mathbf{x}_{\mathrm{ref}} - \mathbf{x}_0| - |\mathbf{x}_{\mathrm{fs}} - \mathbf{x}_0|\big|}} e^{\mathrm{j}\frac{\omega}{c}|x_{\mathrm{fs}} - x_0|}$$

Resulting Driving Signal

$$\begin{aligned} D^{\mathrm{LWFS}}(\mathbf{x}_0, \omega) &= \sum_{x_{\mathrm{fs}} \in \mathcal{X}_{\mathrm{fs}}} D^{\mathrm{WFS}}(\mathbf{x}_{\mathrm{fs}}, \omega) D_{\mathrm{fs}}^{\mathrm{WFS}}(\mathbf{x}_0, \mathbf{x}_{\mathrm{fs}}, \omega) \text{ with} \\ &= |H_{\mathrm{pre}}(\omega)|^2 \sum_{x_{\mathrm{fs}} \in \mathcal{X}_{\mathrm{fs}}} w(\mathbf{x}_0, \mathbf{x}_{\mathrm{fs}})\, e^{-\mathrm{j}\omega\tau(x_0, x_{\mathrm{fs}})} \end{aligned}$$

Local WFS using Virtual Secondary Sources

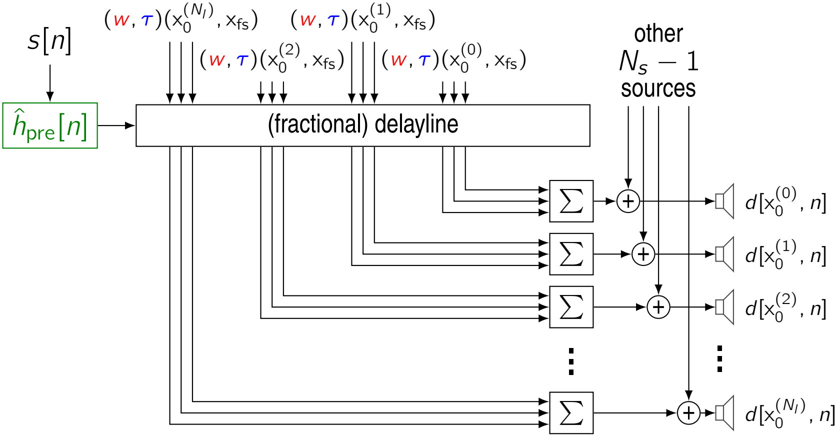
2-Stage Realisation



$$\begin{aligned} c_{\mathrm{l,l}}^{\mathrm{LWFS}}(N_s, N_{\mathrm{fs}}, N_{\mathrm{l}}) &= c_{\mathrm{l}}^{\mathrm{WFS}}(N_s, N_{\mathrm{fs}}) + c_{\mathrm{l}}^{\mathrm{WFS}}(N_{\mathrm{fs}}, N_{\mathrm{l}}) \\ &= (N_{\mathrm{s}} + N_{\mathrm{fs}}) c_{\mathrm{conv}}^{\mathrm{pre}} + (N_{\mathrm{s}} + N_{\mathrm{fs}}) c_{\mathrm{write}}^{\mathrm{dl}} + N_{\mathrm{fs}}(N_{\mathrm{s}} + N_{\mathrm{l}}) c_{\mathrm{read}}^{\mathrm{dl}} \end{aligned}$$

Local WFS using Virtual Secondary Sources

Direct Realisation



$$c_{\mathrm{direct}}^{\mathrm{LWFS}}(N_s, N_{\mathrm{fs}}, N_{\mathrm{l}}) = N_{\mathrm{s}} c_{\mathrm{conv}}^{\mathrm{pre}} + N_{\mathrm{s}} c_{\mathrm{write}}^{\mathrm{dl}} + N_{\mathrm{fs}} N_{\mathrm{l}} N_{\mathrm{s}} c_{\mathrm{read}}^{\mathrm{dl}}$$

Local WFS using Virtual Secondary Sources

Comparison

$$c^{\text{LWFS}}_{\text{l,l}}(N_s, N_{\text{fs}}, N_{\text{l}}) = (N_{\text{s}} + N_{\text{fs}})\, c^{\text{pre}}_{\text{conv}} + (N_{\text{s}} + N_{\text{fs}})\, c^{\text{dl}}_{\text{write}} + N_{\text{fs}}(N_{\text{s}} + N_{\text{l}})\, c^{\text{dl}}_{\text{read}}$$

$$c^{\text{LWFS}}_{\text{direct}}(N_s, N_{\text{fs}}, N_{\text{l}}) = N_{\text{s}}\, c^{\text{pre}}_{\text{conv}} + N_{\text{s}}\, c^{\text{dl}}_{\text{write}} + N_{\text{fs}} N_{\text{l}} N_{\text{s}}\, c^{\text{dl}}_{\text{read}}$$

Break-even-Point without Pre-Filtering

$$c^{\text{LWFS}}_{\text{l,l}}(N_s, N_{\text{fs}}, N_{\text{l}}) \overset{?}{>} c^{\text{LWFS}}_{\text{direct}}(N_s, N_{\text{fs}}, N_{\text{l}})$$

$$c^{\text{dl}}_{\text{write}} > [(N_{\text{l}} - 1)N_{\text{s}} - N_{\text{l}}]\, c^{\text{dl}}_{\text{read}}$$

- Break-even-Point depends on required accuracy of delay interpolation
- 2-stage realisation requires more delaylines $\rightarrow$ more memory

Local WFS using Virtual Secondary Sources

Comparison

$$c_{\mathrm{l,l}}^{\mathrm{LWFS}}(N_s, N_{\mathrm{fs}}, N_{\mathrm{l}}) = (N_{\mathrm{s}} + N_{\mathrm{fs}})\,c_{\mathrm{conv}}^{\mathrm{pre}} + (N_{\mathrm{s}} + N_{\mathrm{fs}})\,c_{\mathrm{write}}^{\mathrm{dl}} + N_{\mathrm{fs}}(N_{\mathrm{s}} + N_{\mathrm{l}})\,c_{\mathrm{read}}^{\mathrm{dl}}$$

$$c_{\mathrm{direct}}^{\mathrm{LWFS}}(N_s, N_{\mathrm{fs}}, N_{\mathrm{l}}) = N_{\mathrm{s}}\,c_{\mathrm{conv}}^{\mathrm{pre}} + N_{\mathrm{s}}\,c_{\mathrm{write}}^{\mathrm{dl}} + N_{\mathrm{fs}} N_{\mathrm{l}} N_{\mathrm{s}}\,c_{\mathrm{read}}^{\mathrm{dl}}$$

Break-even-Point without Pre-Filtering

$$c_{\mathrm{l,l}}^{\mathrm{LWFS}}(N_s, N_{\mathrm{fs}}, N_{\mathrm{l}}) \overset{?}{>} c_{\mathrm{direct}}^{\mathrm{LWFS}}(N_s, N_{\mathrm{fs}}, N_{\mathrm{l}})$$

$$c_{\mathrm{write}}^{\mathrm{dl}} > \left[(N_{\mathrm{l}} - 1)N_{\mathrm{s}} - N_{\mathrm{l}}\right] c_{\mathrm{read}}^{\mathrm{dl}}$$

- Break-even-Point depends on required accuracy of delay interpolation
- 2-stage realisation requires more delaylines $\rightarrow$ more memory

Conclusion & Future Work

Conclusion

- computational cost of pre-filtering can be reduced for some scenarios
- delay independent preprocessing for delay interpolation reduces computational costs but requires more memory
- 2-stage LWFS implementation reused existing software components for WFS
- direct LWFS is more efficient for high-resolution delay interpolation

Future Work

- **fair** evaluation and comparison of run time
- perceptually motivated guidelines for delay interpolation

Thank you for your attention!