AUD-L6.4

ICASSP2020 Barcelona

Mutual-Information-Based Sensor Placement for Spatial Sound Field Recording

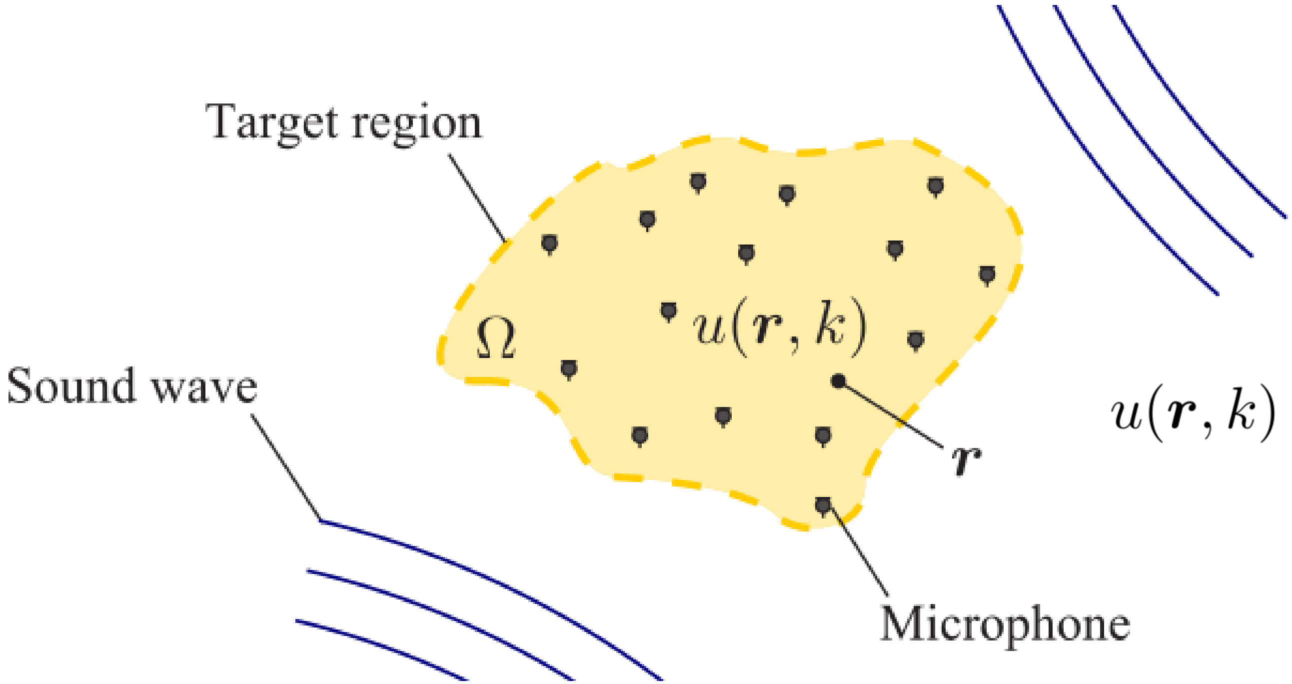
Kentaro Ariga[1], Tomoya Nishida[1], Shoichi Koyama[1,2],
Natsuki Ueno[1], and Hiroshi Saruwatari[1]

[1]The University of Tokyo, [2]JST PRESTO

https://ieeexplore.ieee.org/document/9053715

Sensor placement in sound field reconstruction

- Sound field reconstruction
  - Estimate and interpolate acoustic field with multiple sensors
  - Can be used in reproducing the sound field, visualization, active control, etc…

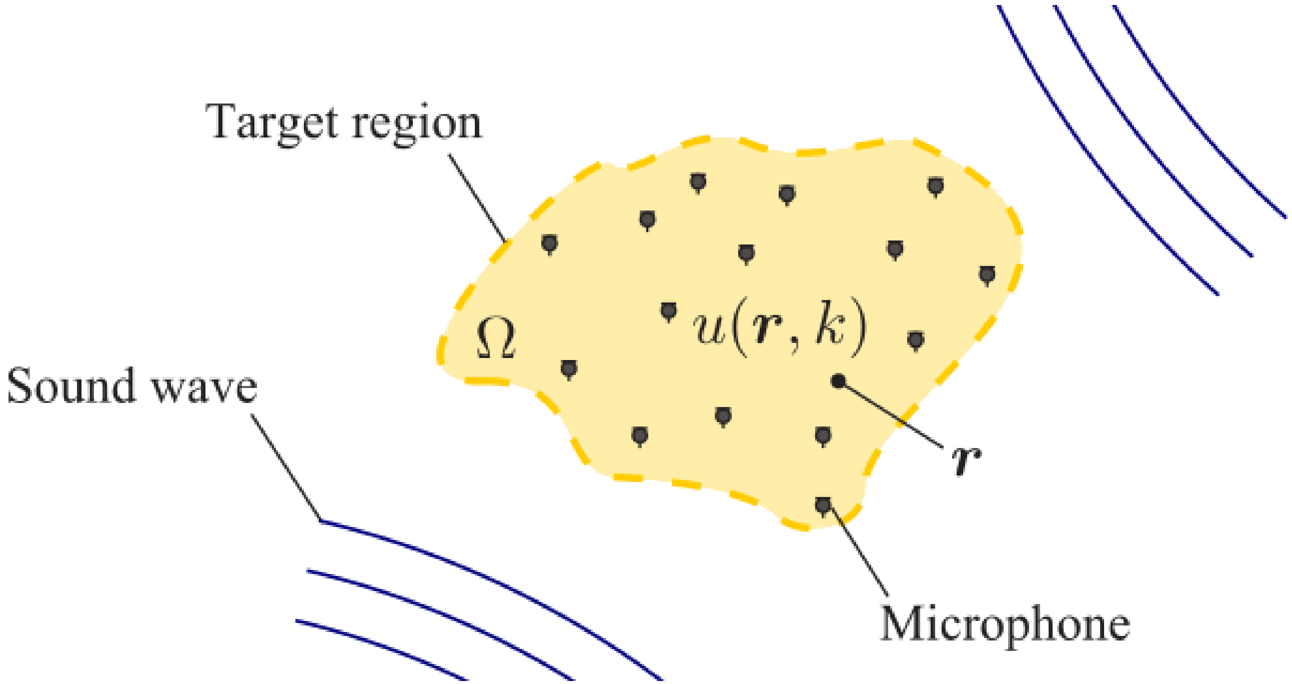


$u(\boldsymbol{r}, k)$: Pressure field at wavenumber k

Sensor placement in sound field reconstruction

How to determine the placement of sensors?

- Precision of sound field reconstruction largely depends on sensor placement
  - Number of sensors available is normally limited
  - Placing sensors on the boundary causes significant error at specific frequencies (nonuniqueness issue)



Mutual-information-based sensor placement

[Krause 2008]

- Select the placement positions from predefined discrete candidate positions $\mathcal{V}$ ($|\mathcal{V}| = N$)
- Select M positions that maximize the mutual information between selected and unselected positions $\mathcal{S}$ and $\mathcal{V} \backslash \mathcal{S}$.

($|\mathcal{S}| = M$)

$$\hat{S} = \arg\max_{\mathcal{S} \subseteq \mathcal{V}} \mathrm{MI}(\mathcal{S})$$

, where

$$\mathrm{MI}(\mathcal{S}) = I(\boldsymbol{u}_{\mathcal{S}}; \boldsymbol{u}_{\mathcal{V} \backslash \mathcal{S}})$$

Mutual information between selected and unselected positions

- Obtain nearly-optimal solution by greedy algorithm

Mutual-information-based sensor placement

- Field is modeled by Gaussian Process (GP)
  - Observations at M positions $\boldsymbol{u} = [u(\boldsymbol{r}_1) \quad \cdots \quad u(\boldsymbol{r}_M)]^{\mathsf{T}}$ are assumed to follow a multivariate Gaussian distribution:

$$\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K})$$

$$\boldsymbol{K} = \begin{bmatrix} \kappa(\boldsymbol{r}_1, \boldsymbol{r}_1) & \cdots & \kappa(\boldsymbol{r}_1, \boldsymbol{r}_M) \\ \vdots & \ddots & \vdots \\ \kappa(\boldsymbol{r}_M, \boldsymbol{r}_1) & \cdots & \kappa(\boldsymbol{r}_M, \boldsymbol{r}_M) \end{bmatrix}$$: Gram matrix

Kernel function

  - Typical kernel used in GP is the Gaussian kernel:

$$\kappa(\boldsymbol{r}_i, \boldsymbol{r}_j) = \exp(-\|\boldsymbol{r}_i - \boldsymbol{r}_j\|^2 / 2\gamma^2)$$

Constant parameter

Gaussian kernel is not necessarily optimal for sound field interpolation

Mutual-information-based sensor placement

- Field is modeled by Gaussian Process (GP)
  - Observations at M positions $\boldsymbol{u} = \left[u(\boldsymbol{r}_1) \quad \cdots \quad u(\boldsymbol{r}_M)\right]^{\mathsf{T}}$ are assumed to follow a multivariate Gaussian distribution:

$$\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K})$$

$$\boldsymbol{K} = \begin{bmatrix} \kappa(\boldsymbol{r}_1, \boldsymbol{r}_1) & \cdots & \kappa(\boldsymbol{r}_1, \boldsymbol{r}_M) \\ \vdots & \ddots & \vdots \\ \kappa(\boldsymbol{r}_M, \boldsymbol{r}_1) & \cdots & \kappa(\boldsymbol{r}_M, \boldsymbol{r}_M) \end{bmatrix}$$: Gram matrix

Kernel function

Find kernel appropriate for sound field interpolation

Kernel-induced sound field interpolation

[Ueno 2018]

- Define function space for sound field interpolation from solution set of Helmholtz equation (in 2D case)

$$\left(\Delta + k^2\right) u = 0$$

  - Expansion by cylindrical wave function with expansion center $\boldsymbol{r}_0$

$$u(\boldsymbol{r}) = \sum_{\nu=-\infty}^{\infty} \tilde{u}_\nu(\boldsymbol{r}_0) J_\nu(k\|\boldsymbol{r} - \boldsymbol{r}_0\|) e^{j\nu\angle(\boldsymbol{r}-\boldsymbol{r}_0)}$$

ν th-order Bessel function

  - Define Hilbert space $\mathcal{H}$

$$\mathcal{H} = \{u \in \mathcal{C}^2(\Omega; \mathbb{C}),\ \text{s.t. } (\Delta + k^2)u = 0 \mid \|u\|_{\mathcal{H}} < \infty\},$$

with

$$\|u\|_{\mathcal{H}} = \sqrt{\langle u, u \rangle_{\mathcal{H}}},$$

$$\sqrt{\langle u, u \rangle_{\mathcal{H}}} = \sum_{\nu=-\infty}^{\infty} \tilde{u}_\nu(\boldsymbol{r}_0)^* \tilde{v}_\nu(\boldsymbol{r}_0)$$

Kernel-induced sound field interpolation

[Ueno 2018]

- $\mathcal{H}$ is the reproducing kernel Hilbert space (RKHS) with reproducing kernel:

$$\kappa(\boldsymbol{r}, \boldsymbol{r}') = J_0(k\|\boldsymbol{r} - \boldsymbol{r}'\|)$$

0th-order Bessel function

- Appropriate Gram matrix for sound field interpolation

$$\boldsymbol{K} = \begin{bmatrix} J_0\left(k\|\boldsymbol{r}_1 - \boldsymbol{r}_1\|\right) & \cdots & J_0\left(k\|\boldsymbol{r}_1 - \boldsymbol{r}_M\|\right) \\ \vdots & \ddots & \vdots \\ J_0\left(k\|\boldsymbol{r}_M - \boldsymbol{r}_1\|\right) & \cdots & J_0\left(k\|\boldsymbol{r}_M - \boldsymbol{r}_M\|\right) \end{bmatrix}$$

Proposed algorithm

- Greedy algorithm

while $|\mathcal{S}|$ reaches to M **do**

$\mathcal{S} \leftarrow \mathcal{S} \cup \{\hat{m}\}$, s.t. $\hat{m} = \arg\max_{m \in \mathcal{V} \backslash \mathcal{S}} \mathrm{MI}(\mathcal{S} \cup \{m\}) - \mathrm{MI}(\mathcal{S})$

end while

Increase of the mutual information in the next step

- Greedy algorithm obtains suboptimal solution
- Increase of mutual information can be computed directly

$$\mathrm{MI}(\mathcal{S} \cup \{m\}) - \mathrm{MI}(\mathcal{S}) = \frac{1}{2} \log \frac{\sigma_m^2 - \mathbf{\Sigma}_{m,\mathcal{S}} \mathbf{\Sigma}_{\mathcal{S},\mathcal{S}}^{-1} \mathbf{\Sigma}_{\mathcal{S},m}}{\sigma_m^2 - \mathbf{\Sigma}_{m,\bar{\mathcal{S}}} \mathbf{\Sigma}_{\bar{\mathcal{S}},\bar{\mathcal{S}}}^{-1} \mathbf{\Sigma}_{\bar{\mathcal{S}},m}}$$

$\bar{\mathcal{S}} = \mathcal{V} \backslash (\mathcal{S} \cup \{m\})$

$\mathbf{\Sigma}_{\mathcal{A},\mathcal{B}}$: Covariance matrix between $\mathcal{A}$ and $\mathcal{B}$

$\bar{\mathcal{S}} = \mathcal{V} \backslash (\mathcal{S} \cup \{m\})$

Proposed algorithm

- Greedy algorithm

while $|\mathcal{S}|$ reaches to M **do**

$$\mathcal{S} \leftarrow \mathcal{S} \cup \{\hat{m}\}, \text{ s.t. } \hat{m} = \underset{m \in \mathcal{V} \setminus \mathcal{S}}{\arg\max} \, \mathrm{MI}(\mathcal{S} \cup \{m\}) - \mathrm{MI}(\mathcal{S})$$

end while

Broadband case

- Kernel function depends on frequency
  - Introduce a set of frequency bins $\mathcal{F}$
  - Assume independency between measurements of different frequencies

$$\mathrm{MI}(\mathcal{S} \cup \{m\}) - \mathrm{MI}(\mathcal{S}) = \frac{1}{2} \sum_{f \in \mathcal{F}} \log \frac{\sigma_{m,f}^2 - \mathbf{\Sigma}_{m,\mathcal{S},f} \mathbf{\Sigma}_{\mathcal{S},\mathcal{S},f}^{-1} \mathbf{\Sigma}_{\mathcal{S},m,f}}{\sigma_{m,f}^2 - \mathbf{\Sigma}_{m,\bar{\mathcal{S}},f} \mathbf{\Sigma}_{\bar{\mathcal{S}},\bar{\mathcal{S}},f}^{-1} \mathbf{\Sigma}_{\bar{\mathcal{S}},m,f}}$$

$\mathbf{\Sigma}_{\mathcal{A},\mathcal{B},f}$: Covariance matrix between $\mathcal{A}$ and $\mathcal{B}$ at frequency f

$\bar{\mathcal{S}} = \mathcal{V} \setminus (\mathcal{S} \cup \{m\})$

Efficient computation of the algorithm

- Total computational cost: $O(N^4 M)$

Computational cost is high due to the matrix inversion calculation for each $m \in \mathcal{V} \backslash \mathcal{S}$

$(N \gg M)$

- Each iteration calculates

$$\mathrm{MI}(\mathcal{S} \cup \{m\}) - \mathrm{MI}(\mathcal{S}) = \frac{1}{2} \log \frac{\sigma_m^2 - \mathbf{\Sigma}_{m,\mathcal{S}} \mathbf{\Sigma}_{\mathcal{S},\mathcal{S}}^{-1} \mathbf{\Sigma}_{\mathcal{S},m}}{\sigma_m^2 - \mathbf{\Sigma}_{m,\bar{\mathcal{S}}} \mathbf{\Sigma}_{\bar{\mathcal{S}},\bar{\mathcal{S}}}^{-1} \mathbf{\Sigma}_{\bar{\mathcal{S}},m}}$$

$O(N^3)$

$\bar{\mathcal{S}} = \mathcal{V} \backslash (\mathcal{S} \cup \{m\})$

for $m \in \mathcal{V} \backslash \mathcal{S}$

Efficient computation of the algorithm

- Total computational cost: $O(N^4 M)$ ➡ $O(N^3)$

Computational cost is high due to the matrix inversion calculation for each $m \in \mathcal{V} \backslash \mathcal{S}$

$(N \gg M)$

➡
- Avoided by applying the Woodbury matrix identity
- Updating inverse matrices at each iteration are calculated sequentially

- Each iteration calculates

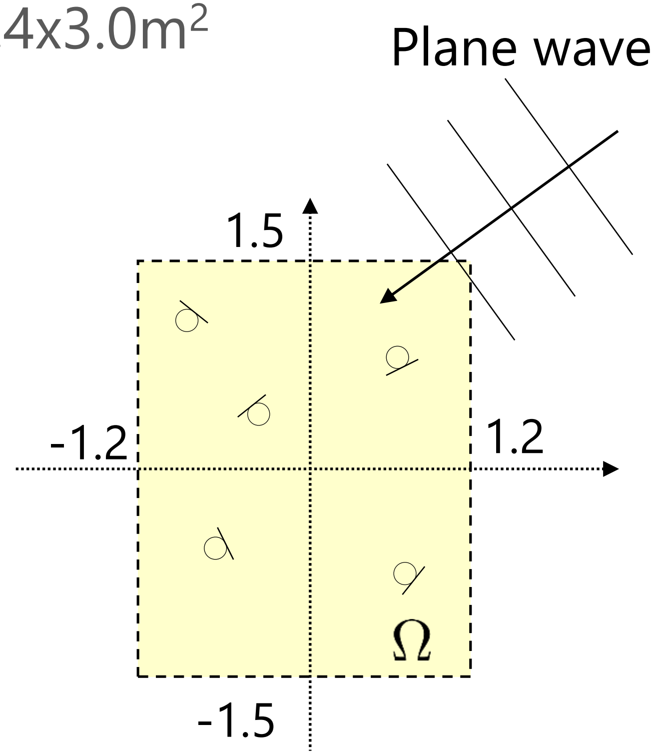
$$\mathrm{MI}(\mathcal{S} \cup \{m\}) - \mathrm{MI}(\mathcal{S}) = \frac{1}{2} \log \frac{\sigma_m^2 - \mathbf{\Sigma}_{m,\mathcal{S}} \mathbf{\Sigma}_{\mathcal{S},\mathcal{S}}^{-1} \mathbf{\Sigma}_{\mathcal{S},m}}{\sigma_m^2 - \mathbf{\Sigma}_{m,\bar{\mathcal{S}}} \mathbf{\Sigma}_{\bar{\mathcal{S}},\bar{\mathcal{S}}}^{-1} \mathbf{\Sigma}_{\bar{\mathcal{S}},m}}$$

$$= \frac{1}{2} \log \left(\sigma_m^2 - \mathbf{\Sigma}_{m,\mathcal{S}} \mathbf{\Sigma}_{\mathcal{S},\mathcal{S}}^{-1} \mathbf{\Sigma}_{\mathcal{S},m} \right) \times \left(\mathbf{\Sigma}_{\mathcal{V} \backslash \mathcal{S}, \mathcal{V} \backslash \mathcal{S}}^{-1} \right)_{m,m}$$

for $m \in \mathcal{V} \backslash \mathcal{S}$

Numerical simulations

- Experiments in 2D sound free field
- Target field Ω: Rectangular region of 2.4x3.0m^2
- Candidate sensor positions:
  - Discretized every 0.04m (N=4636)
- Comparison
  - Proposed method (Proposed)
  - Gaussian kernel (Gauss)
  - Random (Rand)
  - Regular (Reg)
- Number of Sensors M
  - Determined by a threshold for the increase in the mutual information



Numerical simulations

- Evaluated by sound field interpolation accuracy
  - True field: plane-wave field (every 1 deg)
  - Signal-to-noise ratio of measurement: 20dB
  - Evaluation measure : Signal-to-Distortion Ratio (SDR)

$$\mathrm{SDR}(f) = 10\log_{10}\frac{\int_{\Omega}|u_{\mathrm{true}}(\boldsymbol{r},k_f)|^2\mathrm{d}\boldsymbol{r}}{\int_{\Omega}|u_{\mathrm{true}}(\boldsymbol{r},k_f)-\hat{u}(\boldsymbol{r},k_f)|^2\mathrm{d}\boldsymbol{r}}$$

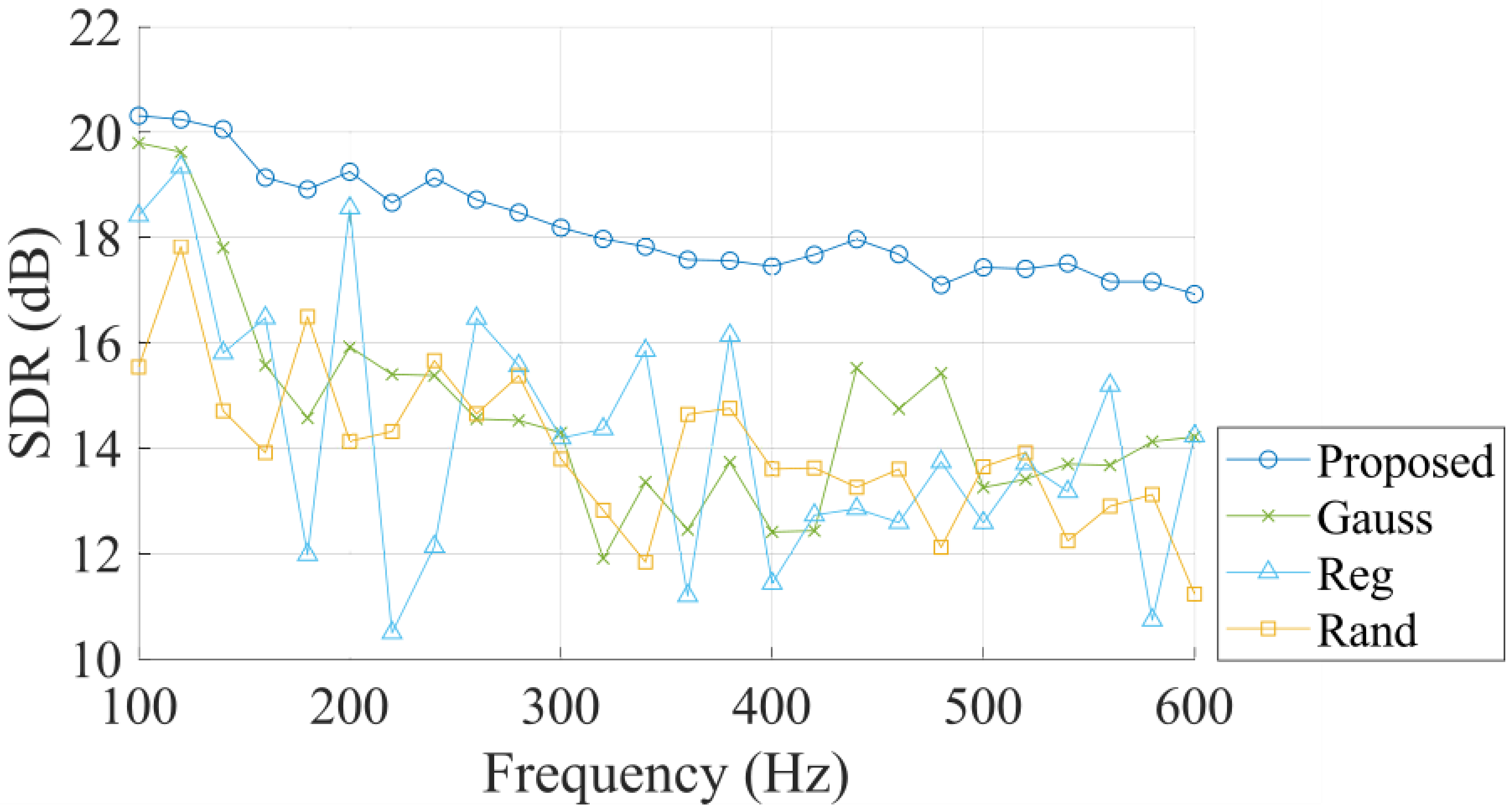
True pressure field

Interpolated pressure field

- Regional integral calculated by discretizing Ω at intervals of 0.01 m.

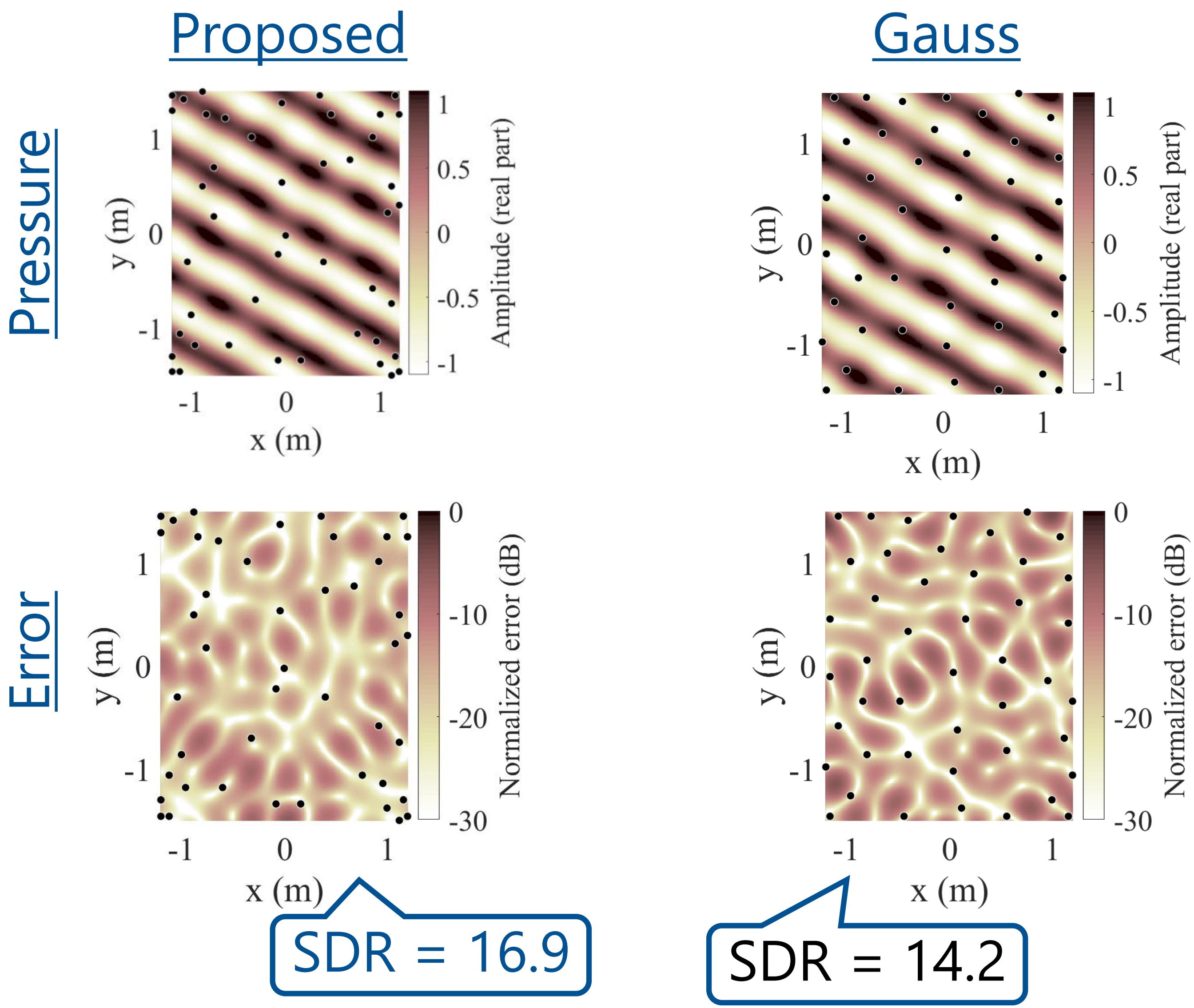
Results – narrowband case

- Average SDR w.r.t. frequency



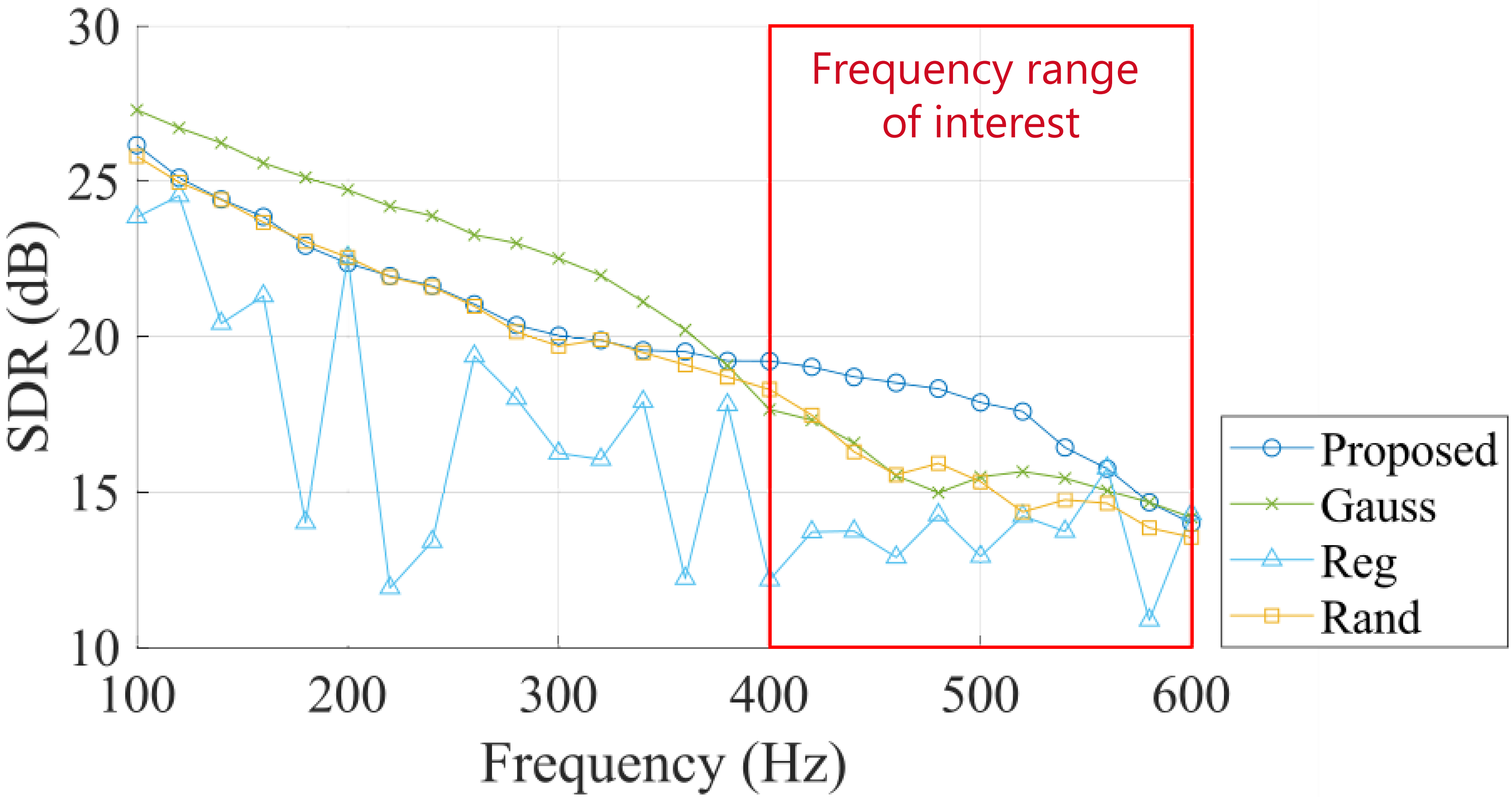
Results – narrowband case

- 600Hz, arrival angle=60° , 45 microphones



Results – broadband case

- Average SDR w.r.t. frequency
- Frequency range of interest set to 400Hz-600Hz



Conclusion

- Sensor placement method for sound field recording
  - Sensor placement based on mutual information
  - Sound-field-Interpolation kernel used as the kernel for GP
  - Efficient computation of the greedy algorithm was derived
  - Better performance than the typically used Gaussian kernel