Large Scale Seismic Data Interpolation with Matrix Completion

Okan Akalin, Curt Da Silva
Ben Recht, Felix Herrmann

UNIVERSITY OF WISCONSIN–MADISON
THE UNIVERSITY OF BRITISH COLUMBIA

Quick Summary

- Problem: Large Scale Seismic Data Interpolation
- Approach: Matrix completion on a 2-D representation of survey data
- Contribution: A scalable extendible algorithm
- Outcome: A simple folding of the tensor yields a matrix that can be successfully completed

Outline

- Introduction
- Our method
- Experiments
- Conclusion & Future Work

Outline

- Introduction
- Our method
- Experiments
- Conclusion & Future Work

Seismic Data Interpolation Problem

- Data is poorly sampled along a subset of modes
- Different from classical interpolation due to the nature of data

Challenges

- Seismic data is characterized by three main properties
  - Incomplete
  - Large volume
  - High dimensional
- Space efficient and fast interpolation is necessary for feasible analysis

Problem Setting

- 5-D data. Modes are time, source (x,y) coordinates, receiver (x,y) coordinates.
- Fourier transform is taken in time domain
- A certain frequency slice is selected from the Fourier transform
- Resulting data: a 4-D incomplete tensor.

Our Approach

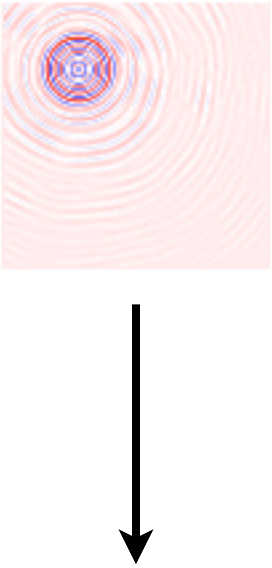
- We apply matrix completion methods to the seismic data interpolation problem.
- Matrix completion
  - solid theoretical results on necessary conditions for exact completion
  - Jellyfish: a state-of-the-art algorithm for large scale problems

Outline

- Introduction
- Our method
  - Encoding the data
  - Matrix Completion
  - Jellyfish & Tensor Completion Algorithm
- Experiments
- Conclusion & Future Work

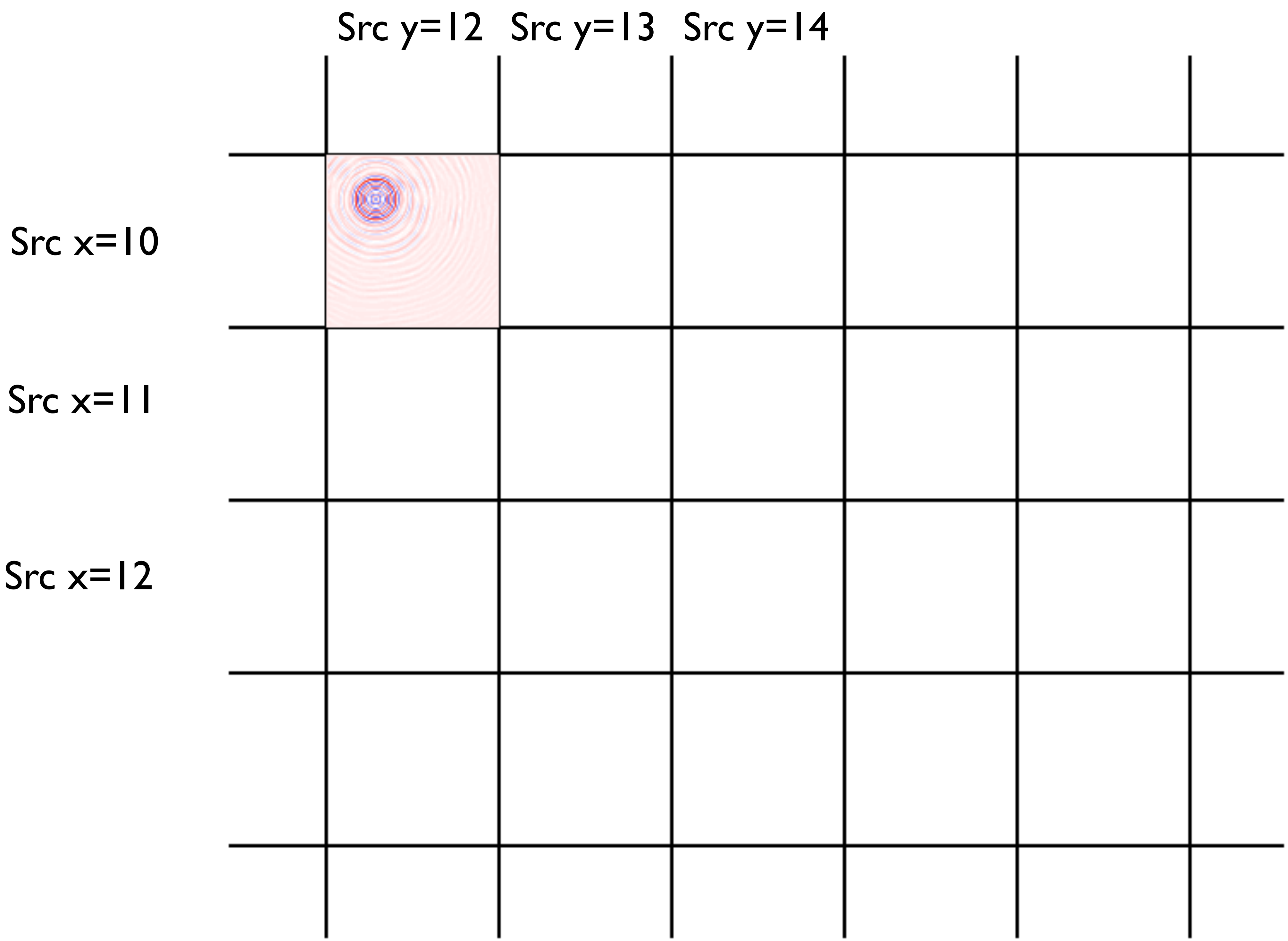
Encoding the survey data as a matrix

(src x, src y)=(10,12)

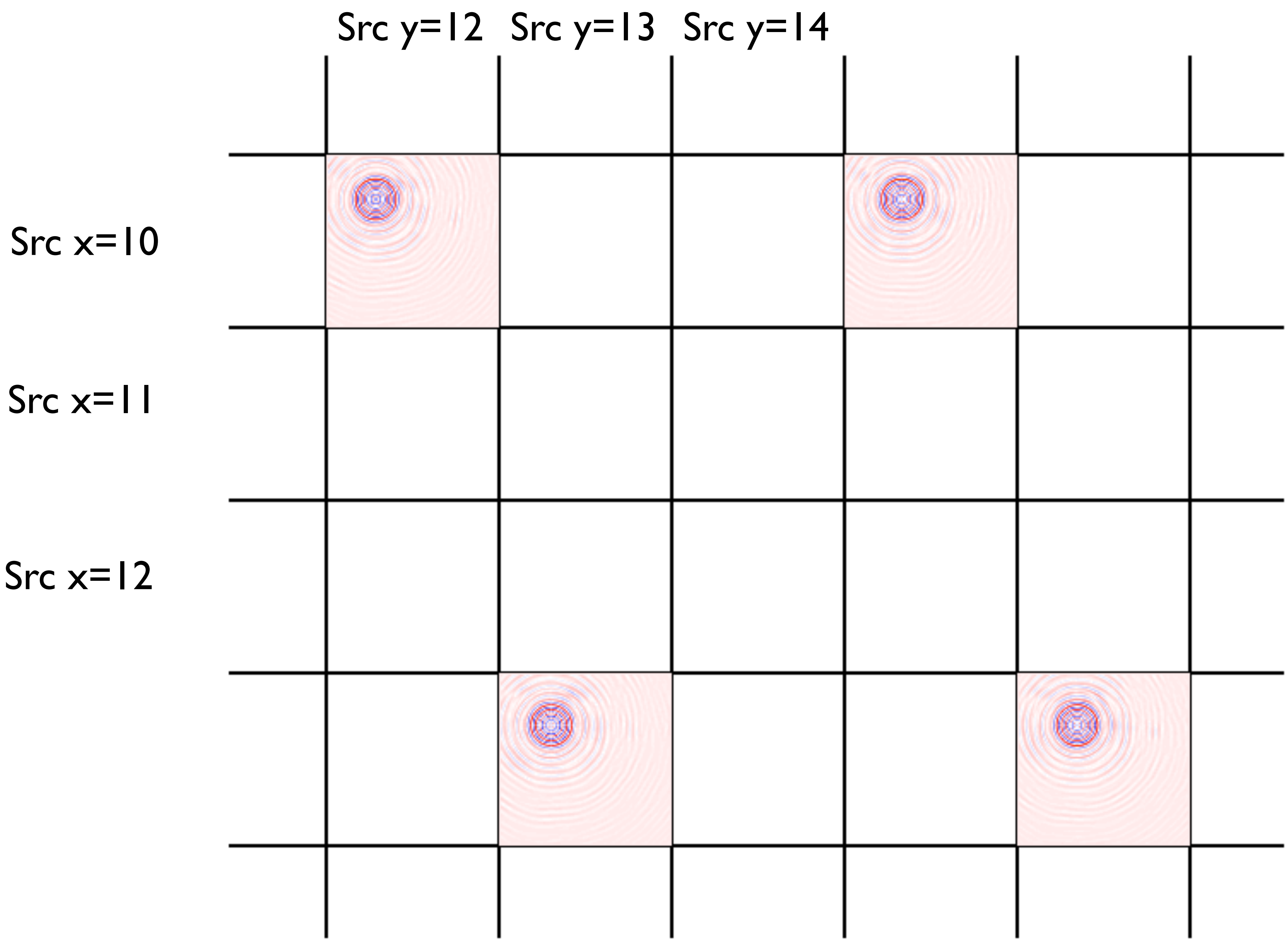


Fixing source coordinates, we obtain a specific shot

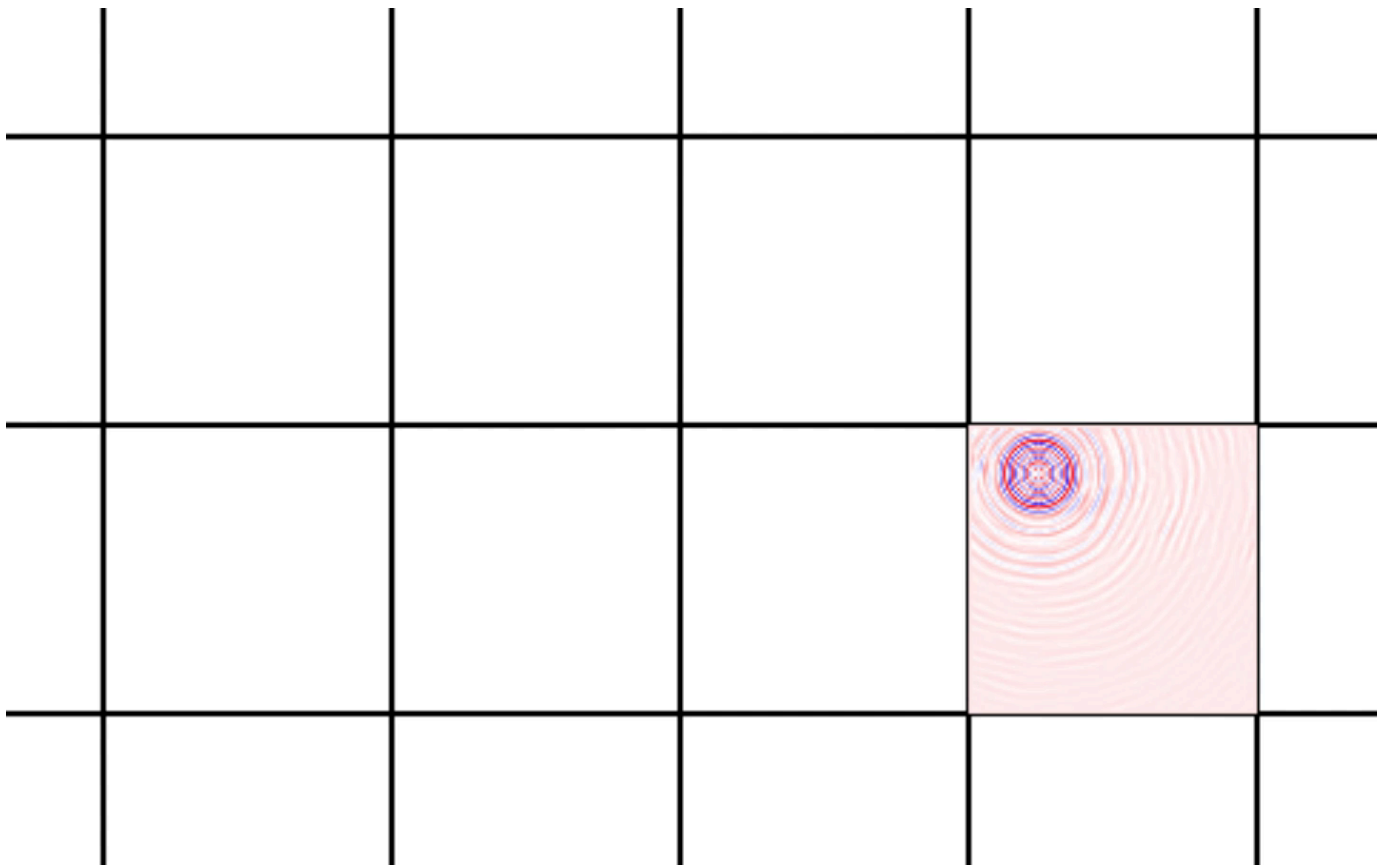
Encoding the survey data as a matrix



Encoding the survey data as a matrix



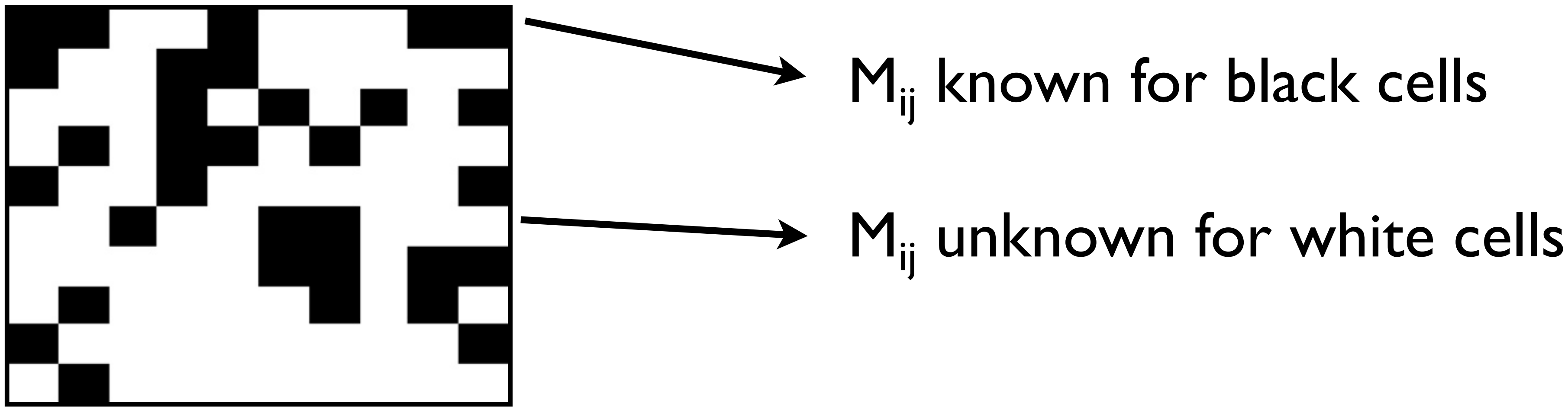
How does sampling on the grid look like?



Outline

- Introduction
- Our method
  - Encoding the data
  - Matrix Completion
  - Jellyfish & Tensor Completion Algorithm
- Experiments
- Conclusion & Future Work

Abstract Setup: Matrix Completion

$\mathbf{M} =$ 

- How do you fill in the missing data?
- Ill posed unless we assume a structure:
  - Low rank!

Rank

- Corresponding problem:

$$\begin{aligned} &\text{minimize} && \operatorname{rank}(\mathbf{X}) \\ &\text{subject to} && X_{ij} = M_{ij} \quad (i,j) \in \Omega \\ & && \mathbf{X} \in \mathbb{R}^{n \times n}, \end{aligned}$$

NP-Complete!

- Convex relaxation: approximate rank by nuclear norm:

$$\begin{aligned} &\text{minimize} && \|\mathbf{X}\|_* \\ &\text{subject to} && \mathrm{X}_{ij} = M_{ij} \quad (i,j) \in \Omega. \end{aligned}$$

$$\|X\|_* = \sum_i \sigma_i(X)$$

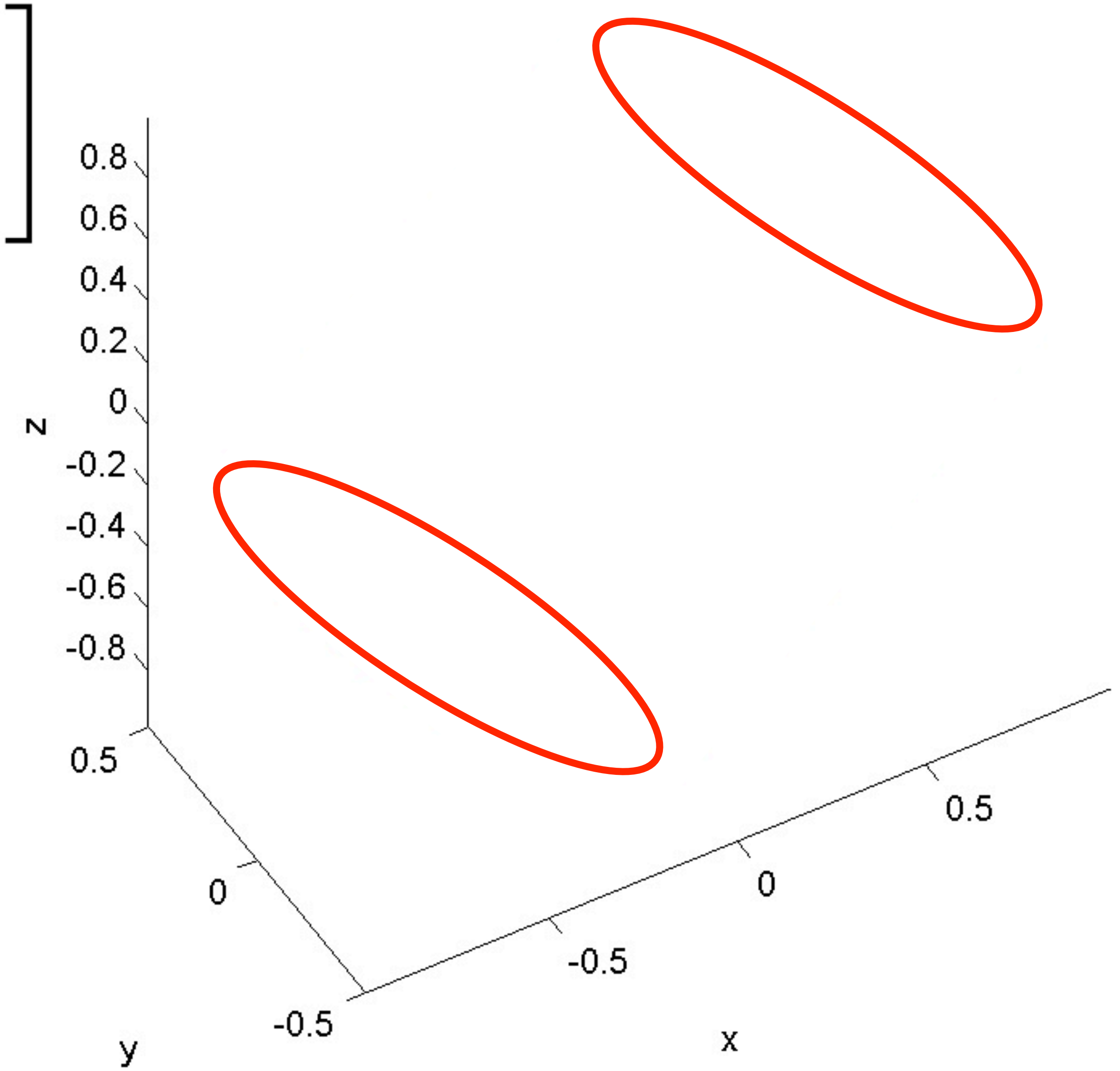
What is the benefit of nuclear norm?

- 2x2 matrices
- Plotted in 3d

$$\begin{bmatrix} x & y \\ y & z \end{bmatrix}$$

— rank 1

$x^2 + z^2 + 2y^2 = 1$



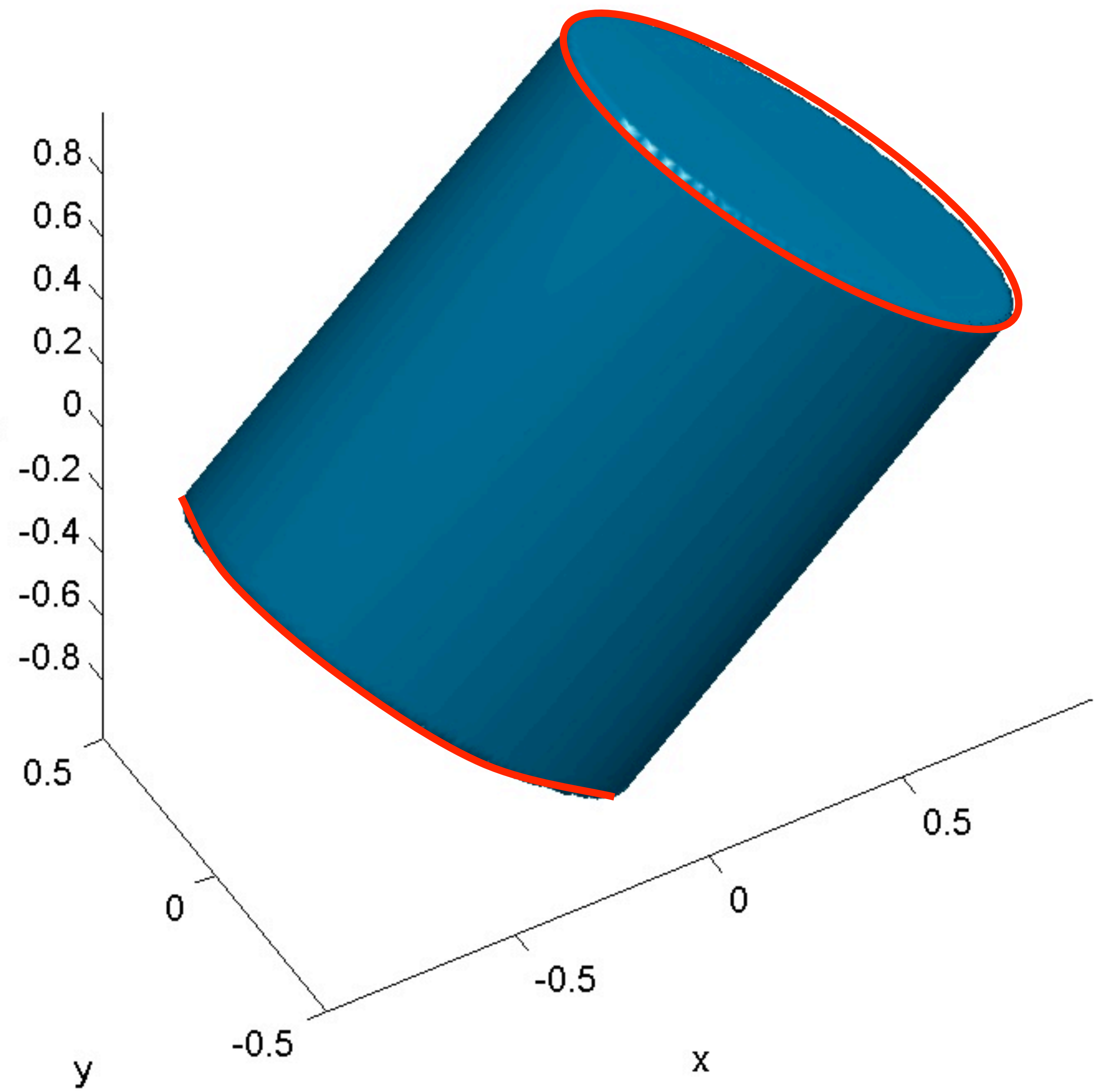
What is the benefit of nuclear norm?

- 2x2 matrices
- Plotted in 3d

$$\begin{bmatrix} x & y \\ y & z \end{bmatrix}$$

— rank 1

$x^2 + z^2 + 2y^2 = 1$



Convex hull:

$$\{X \;:\; \|X\|_* \leq 1\}$$

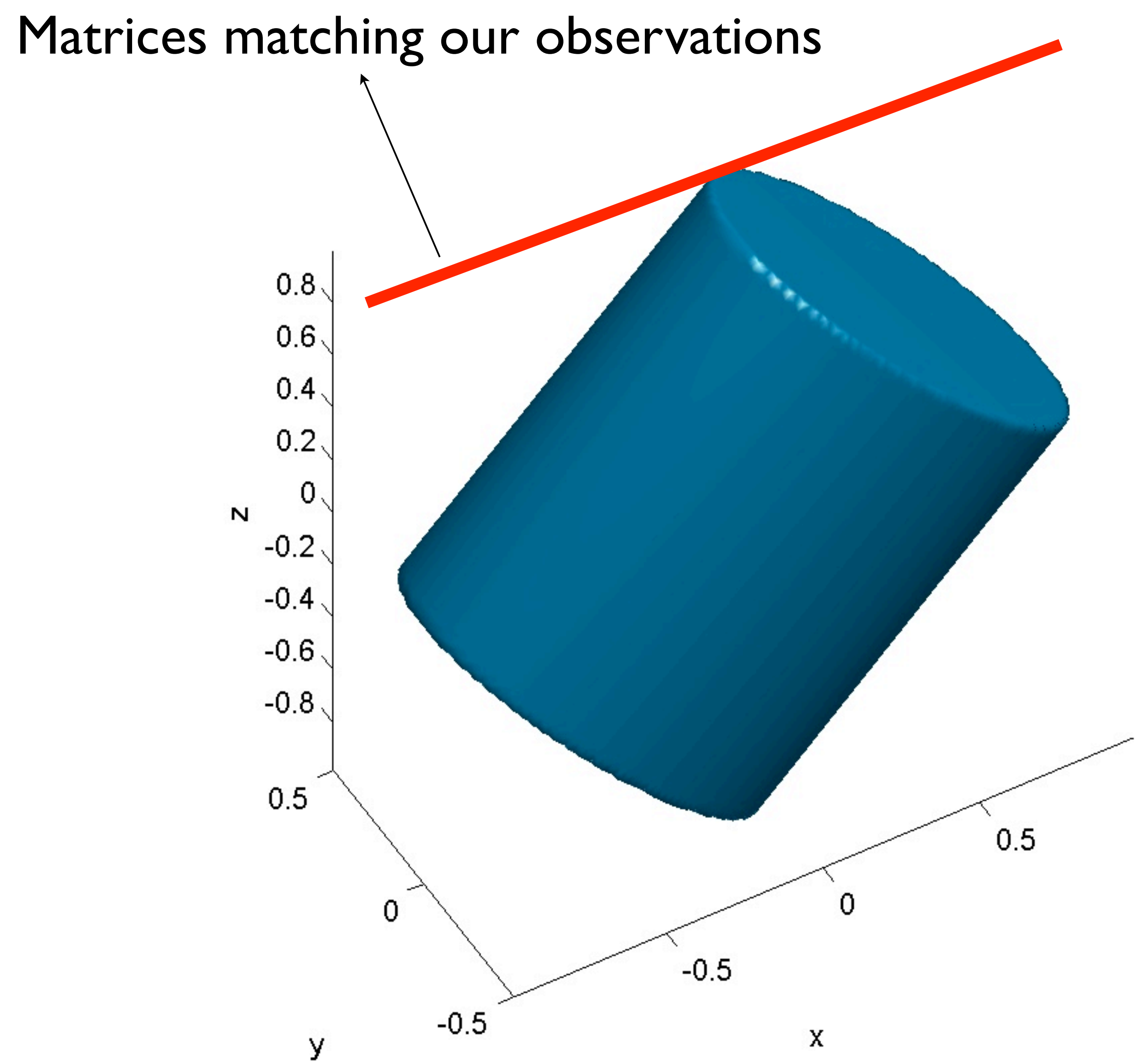
$$\|X\|_* = \sum_i \sigma_i(X)$$

What is the benefit of nuclear norm?

- 2x2 matrices
- Plotted in 3d

$$\left\| \begin{bmatrix} x & y \\ y & z \end{bmatrix} \right\|_* \leq 1$$

$$\|X\|_* = \sum_i \sigma_i(X)$$

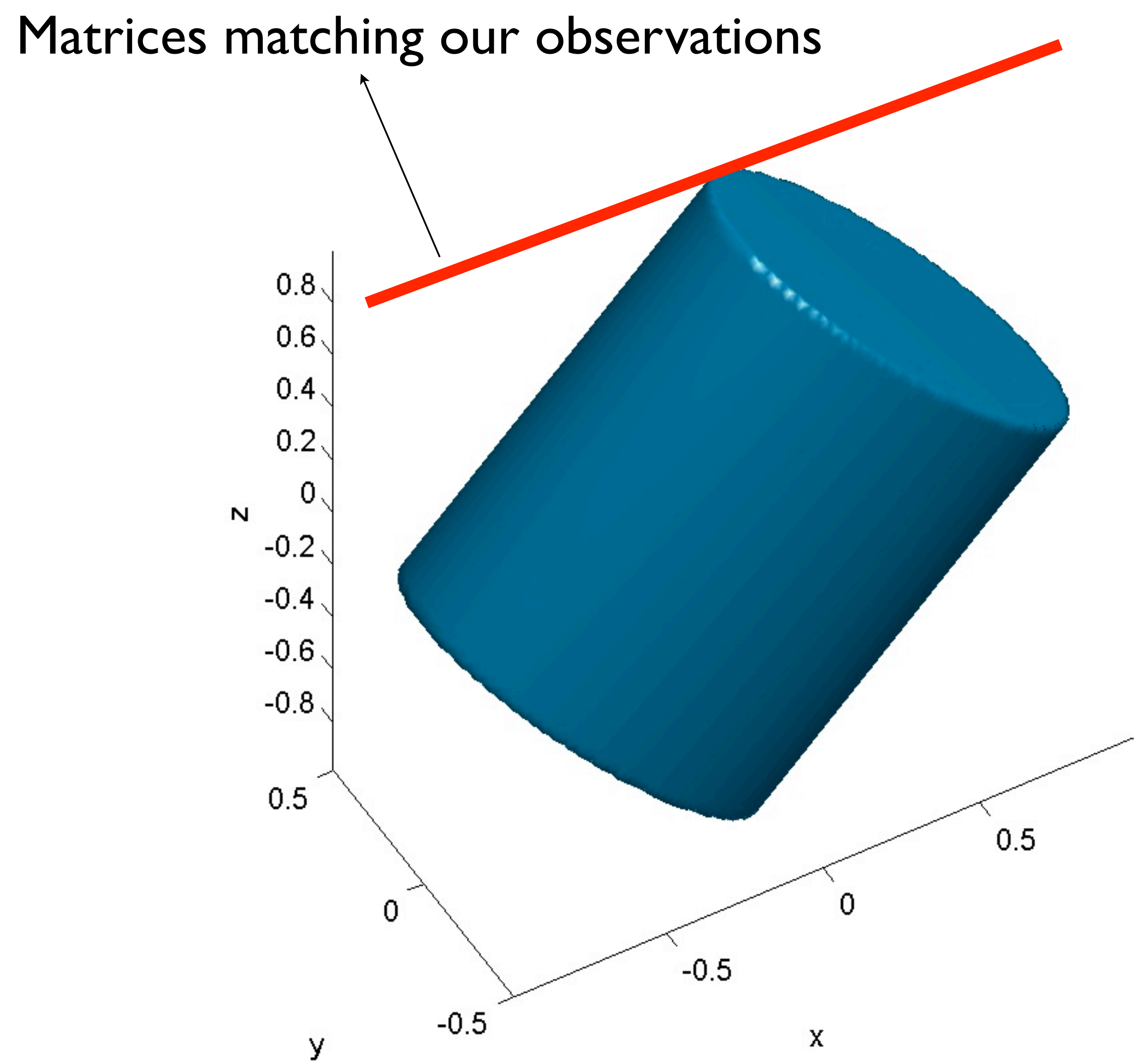


What is the benefit of nuclear norm?

- 2x2 matrices
- Plotted in 3d

$$\left\| \begin{bmatrix} x & y \\ y & z \end{bmatrix} \right\|_* \leq 1$$

$$\|X\|_* = \sum_i \sigma_i(X)$$



Fazel 2002.
Recht, Fazel, and Parillo 2007
Candes and Recht 2009
Rank Minimization/Matrix Completion

Outline

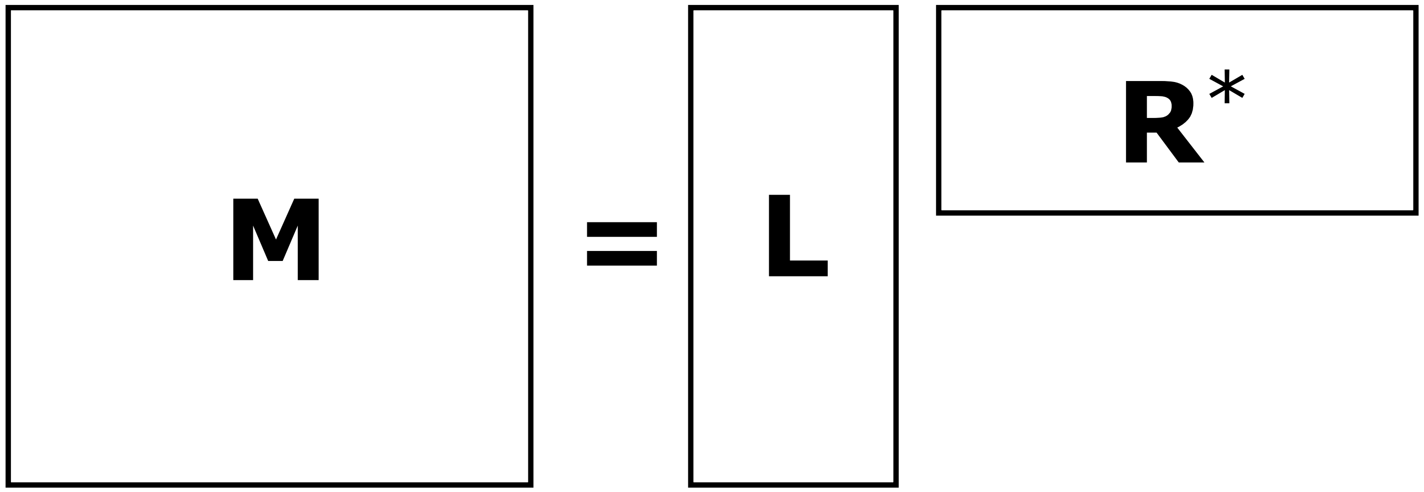
- Introduction
- Our method
  - Encoding the data
  - Matrix Completion
  - Jellyfish & Tensor Completion Algorithm
- Experiments
- Conclusion & Future Work

Jellyfish

SGD for Matrix Factorizations.

Ben Recht and Christopher Ré

- Nuclear norm minimization can be written as a semidefinite program.
  - Does not scale to large datasets!
- Idea: approximate



k x n — k x r — r x n

kn entries — r(k+n) entries

Jellyfish

- Based on explicit factorization:

$$\text{minimize}_{(\mathbf{L},\mathbf{R})} \sum_{(u,v)\in E} \left\{ (\mathbf{L}_u \mathbf{R}_v^T - M_{uv})^2 + \mu_u \|\mathbf{L}_u\|_F^2 + \mu_v \|\mathbf{R}_v\|_F^2 \right\}$$

- Update steps:
  - Step 1: Pick (u,v) and compute residual:

$$e = (\mathbf{L}_u \mathbf{R}_v^T - M_{uv})$$

  - Step 2: Take a gradient step:

$$\begin{bmatrix} \mathbf{L}_u \\ \mathbf{R}_v \end{bmatrix} \leftarrow \begin{bmatrix} (1-\gamma\mu_u)\mathbf{L}_u - \gamma e \mathbf{R}_v \\ (1-\gamma\mu_v)\mathbf{R}_v - \gamma e \mathbf{L}_u \end{bmatrix}$$

- Possible to scale to GB sized matrices by proper sampling

Algorithm

- Matricize data on (src x, rcv x) x (src y, rcv y) grid
  - Storage in sparse matrix form
- Factorize matrix with Jellyfish
- Multiply rows in L and R to obtain elements in the tensor

Outline

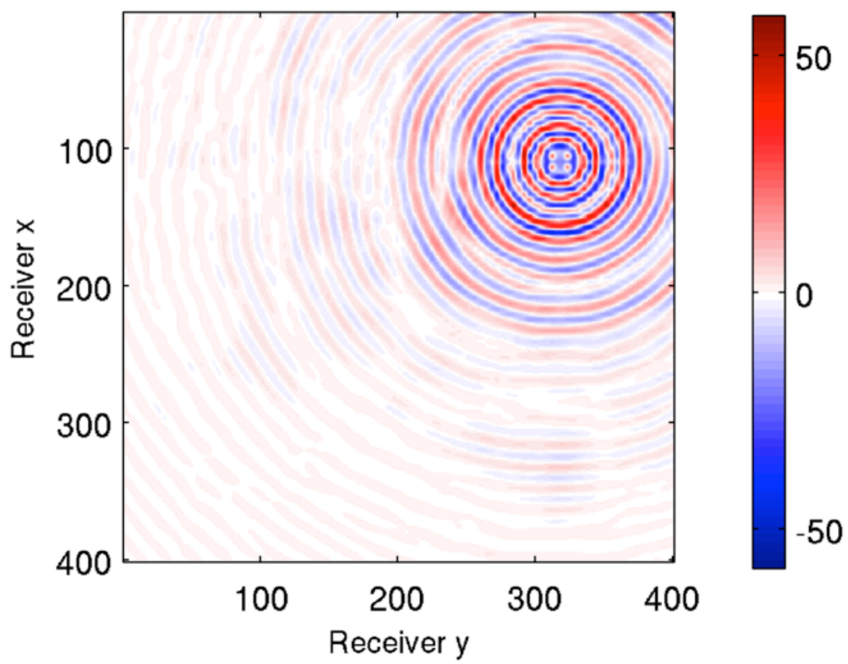
- Introduction
- Our method
- Experiments
- Conclusion & Future Work

Experimental Setup

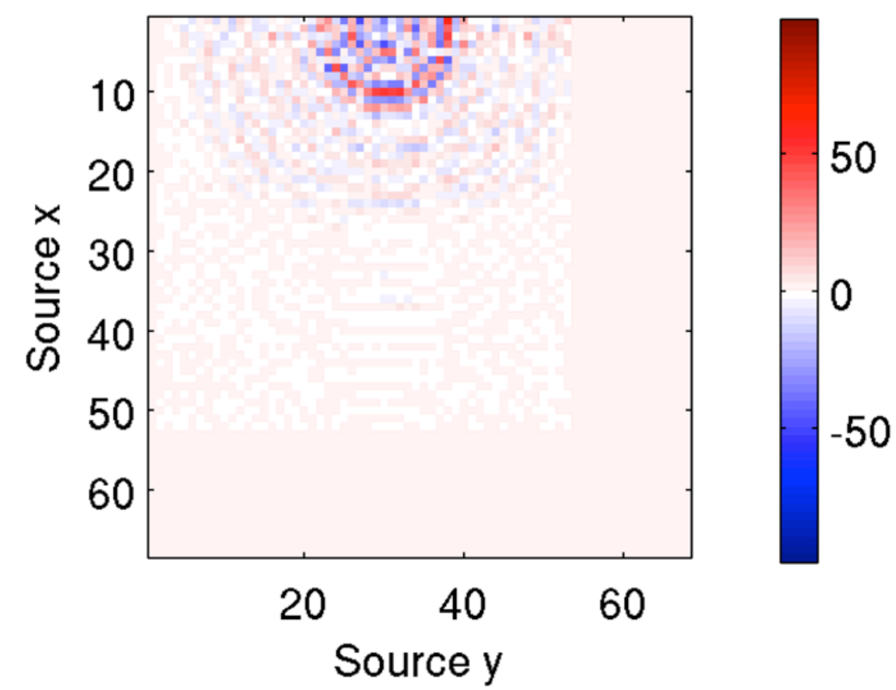
- Data set is a 54x54x101x101 tensor.
- Out of 54x54 shots, 200 are observed.
- 197 shots were used in training
  - Remaining 3 used for parameter selection

Experiments

Fix source coordinates:
Shot data

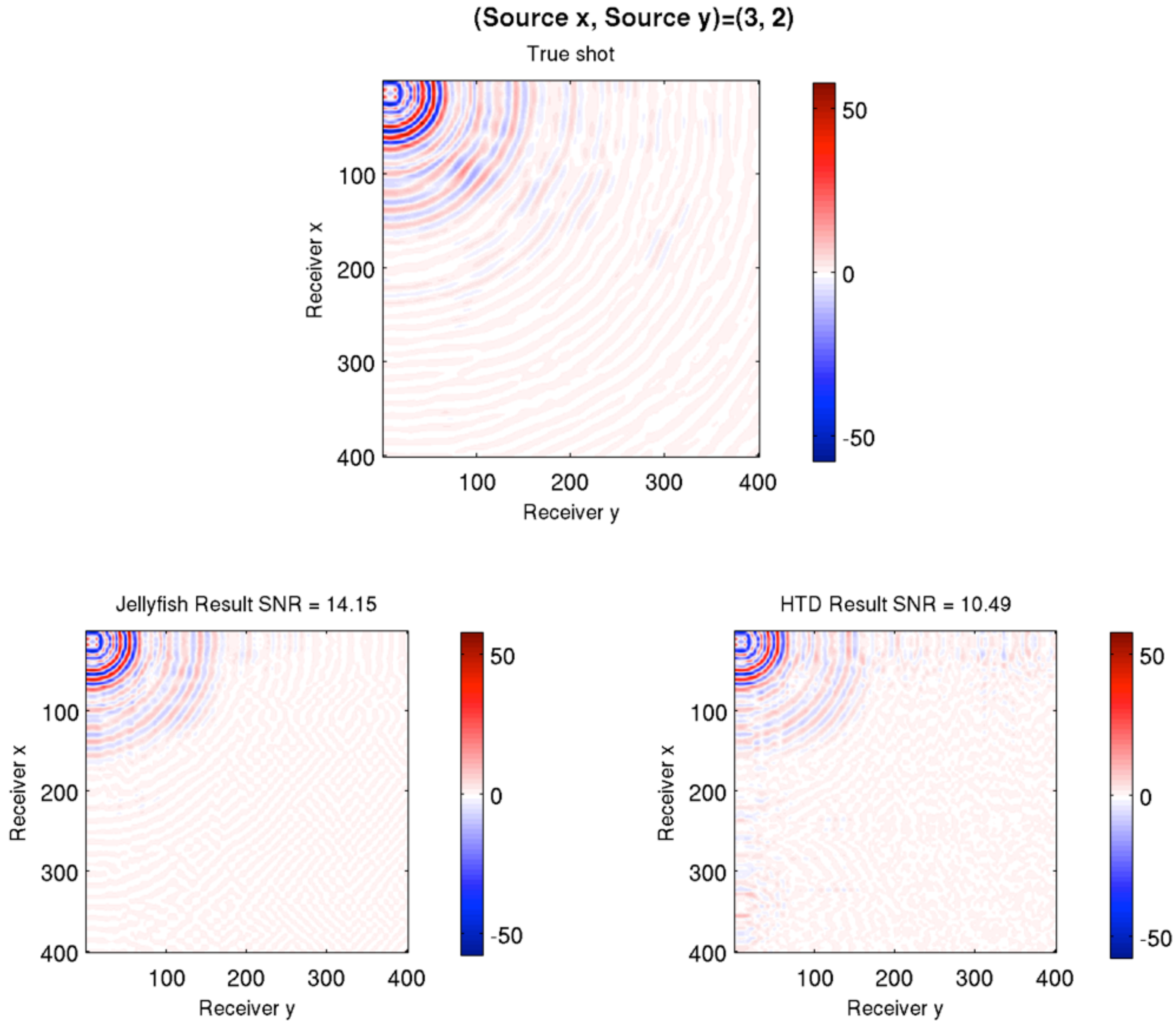


Fix receiver coordinates:
Receiver gathers

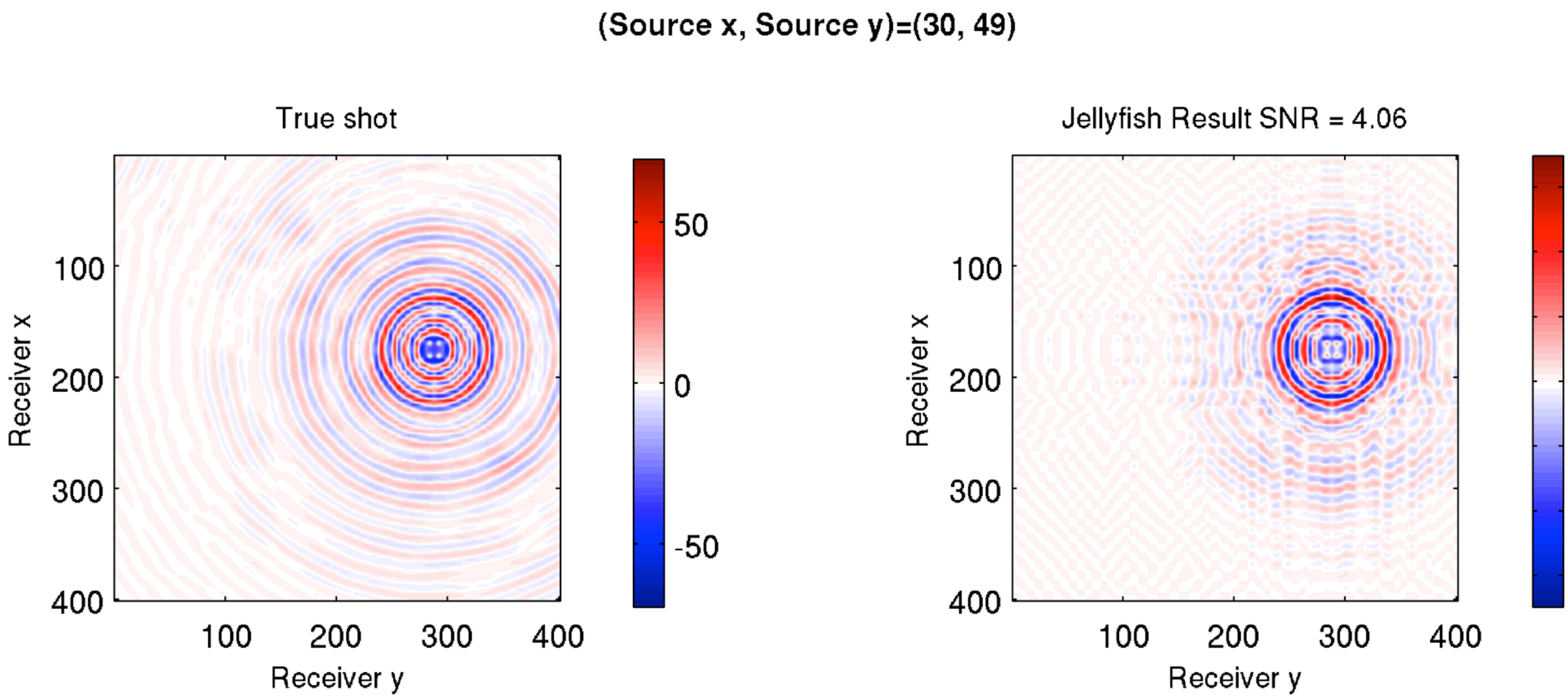


- We evaluate our method using Signal-to-Noise ratio
- Hierarchical Tucker Decomposition results are also presented for comparison

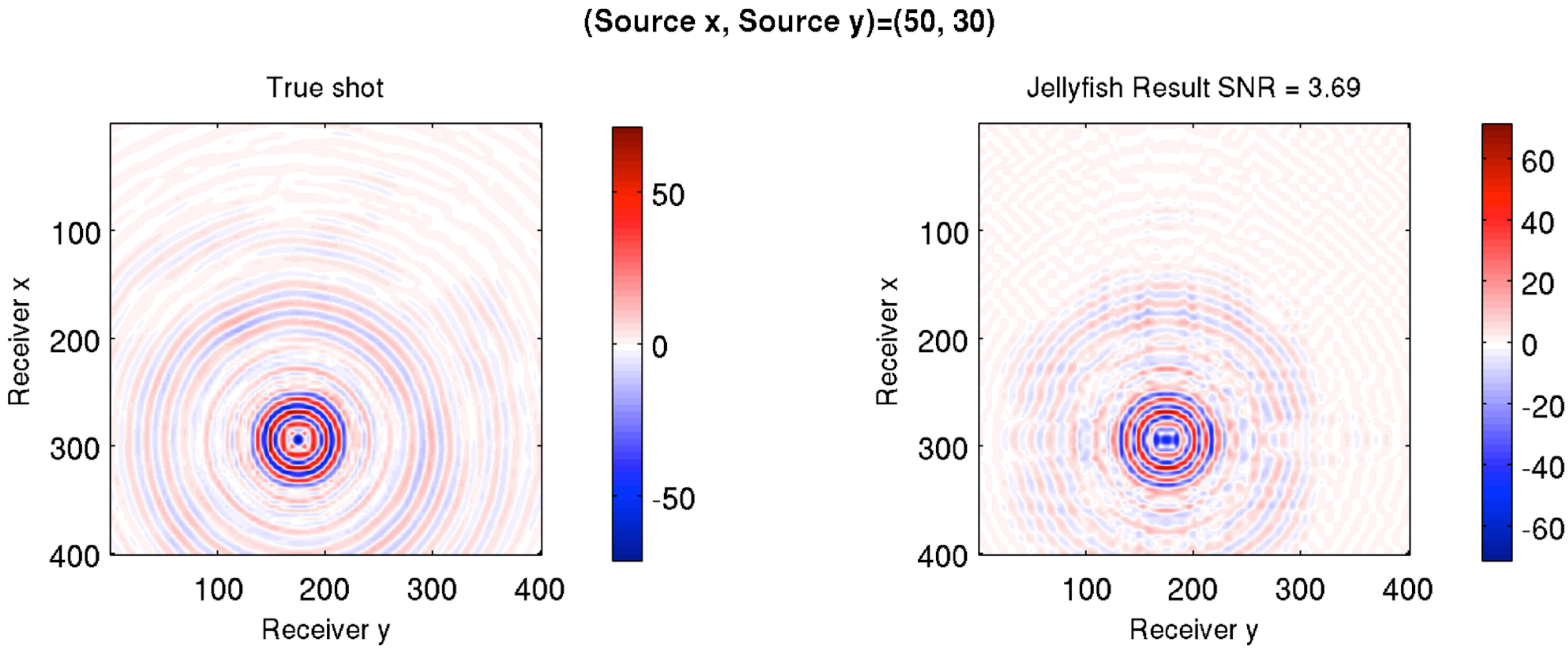
Reconstruction of available shot data



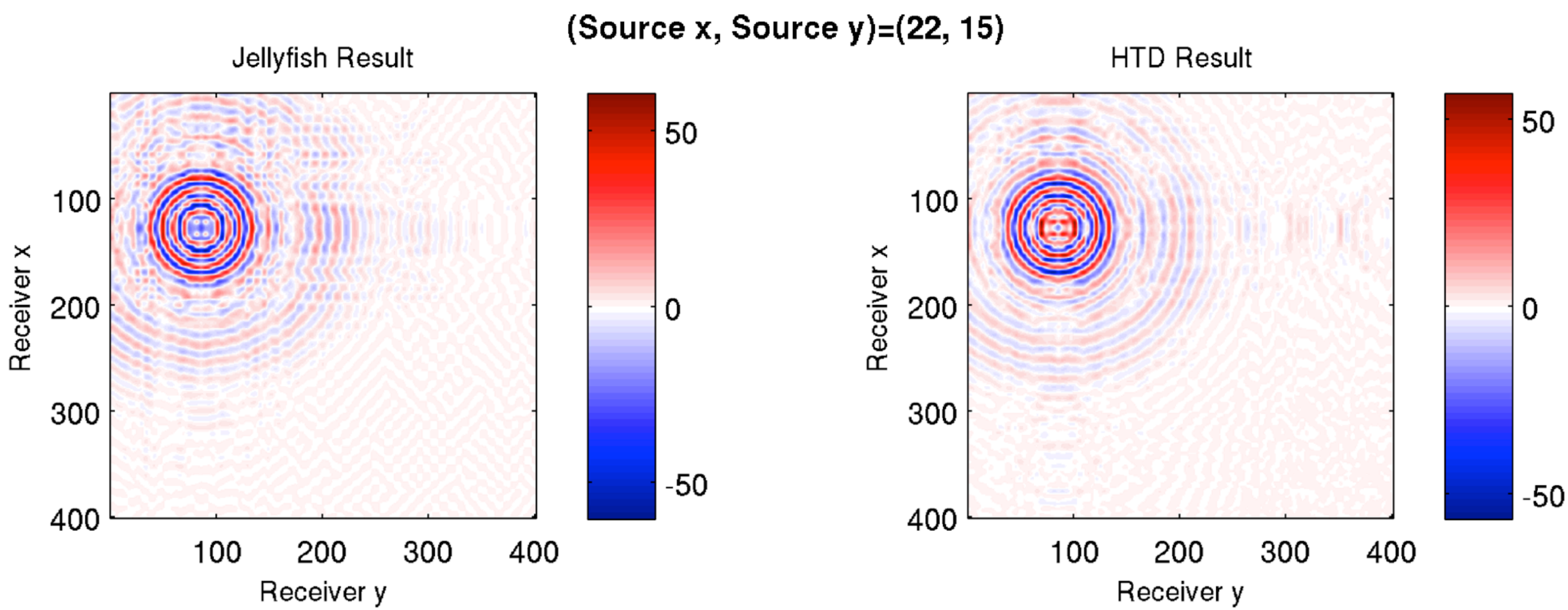
Reconstruction of unseen shot data in the test set



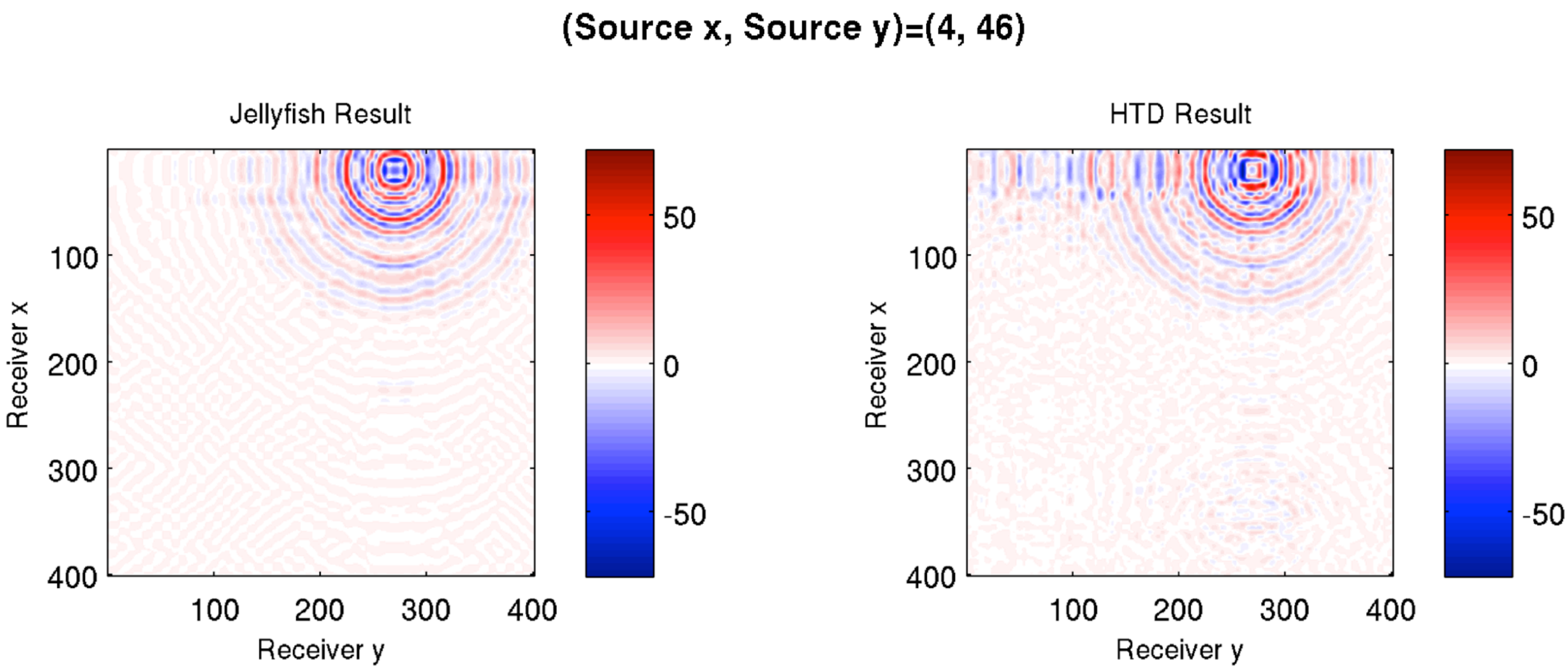
Reconstruction of unseen shot data in the test set



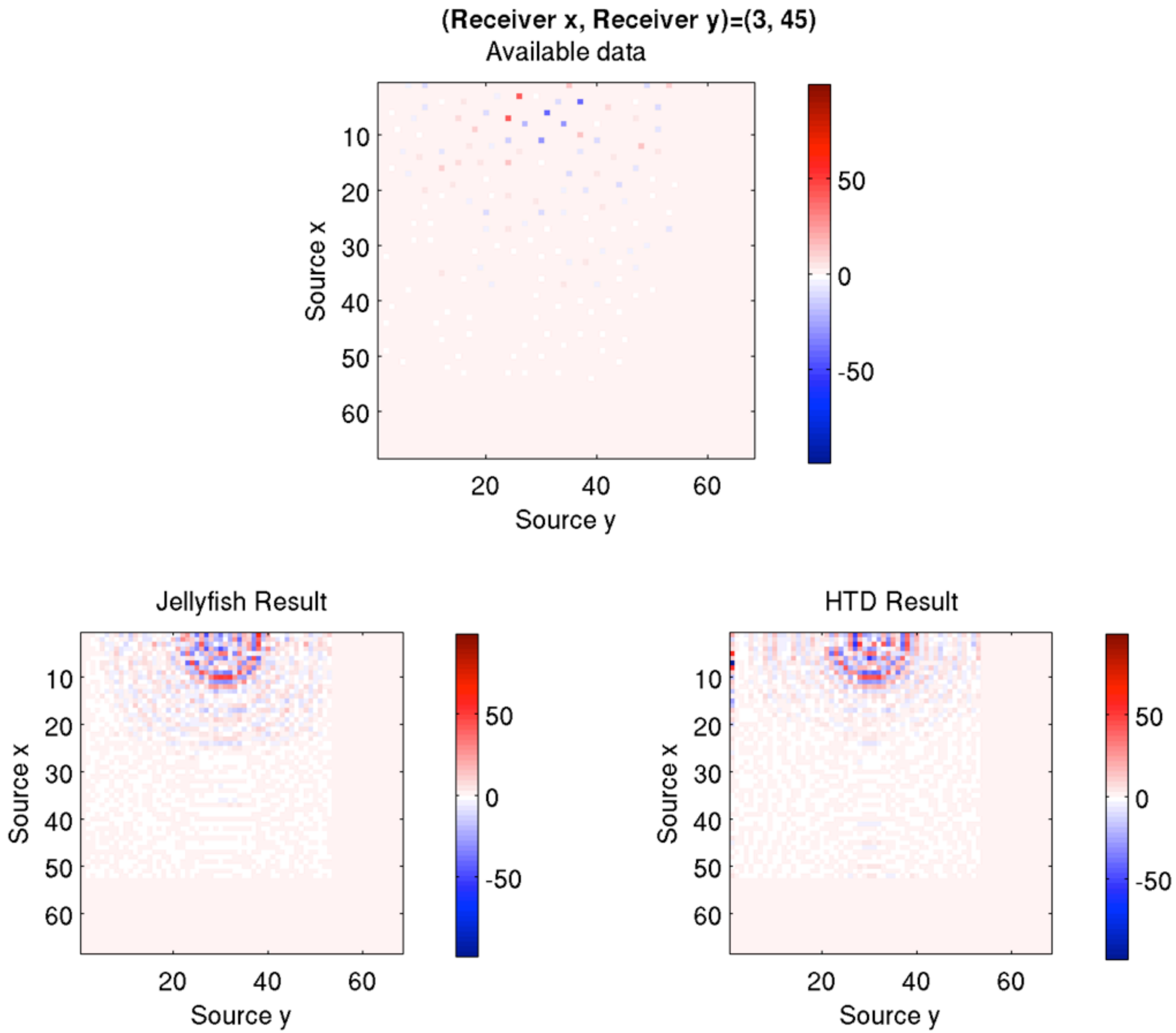
Extrapolation of unseen shot data



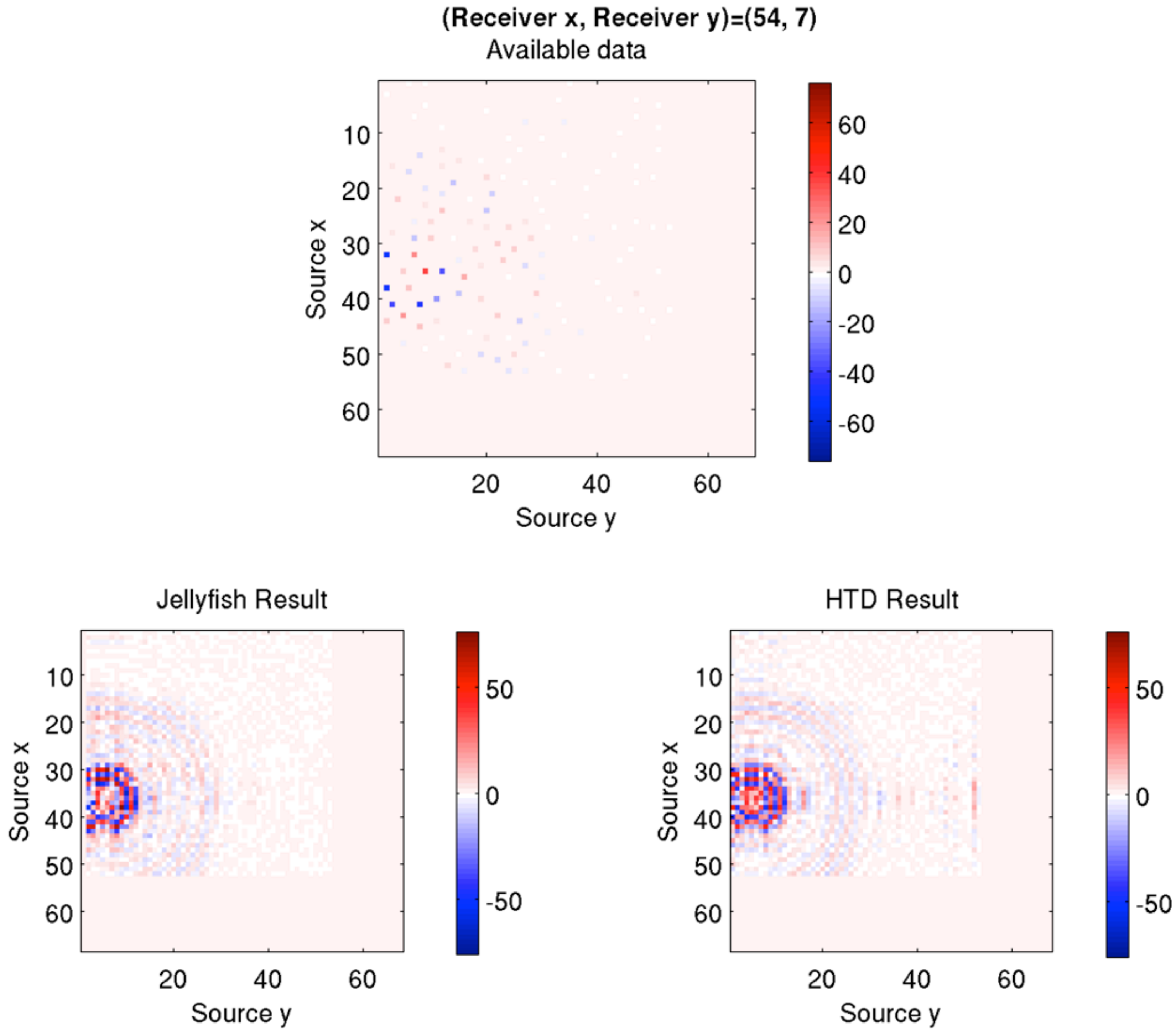
Extrapolation of unseen shot data



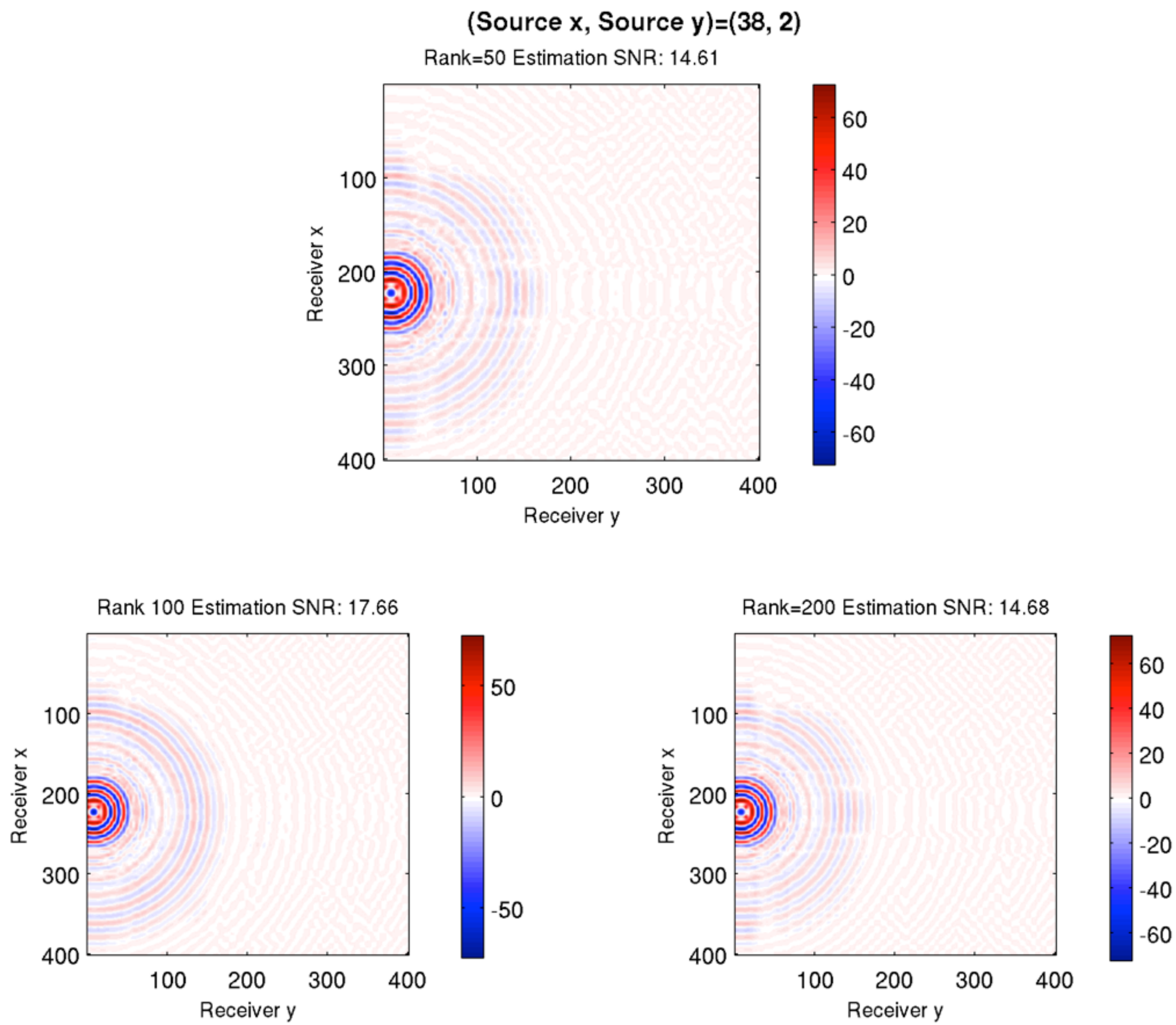
Extrapolation for fixed receiver coordinates



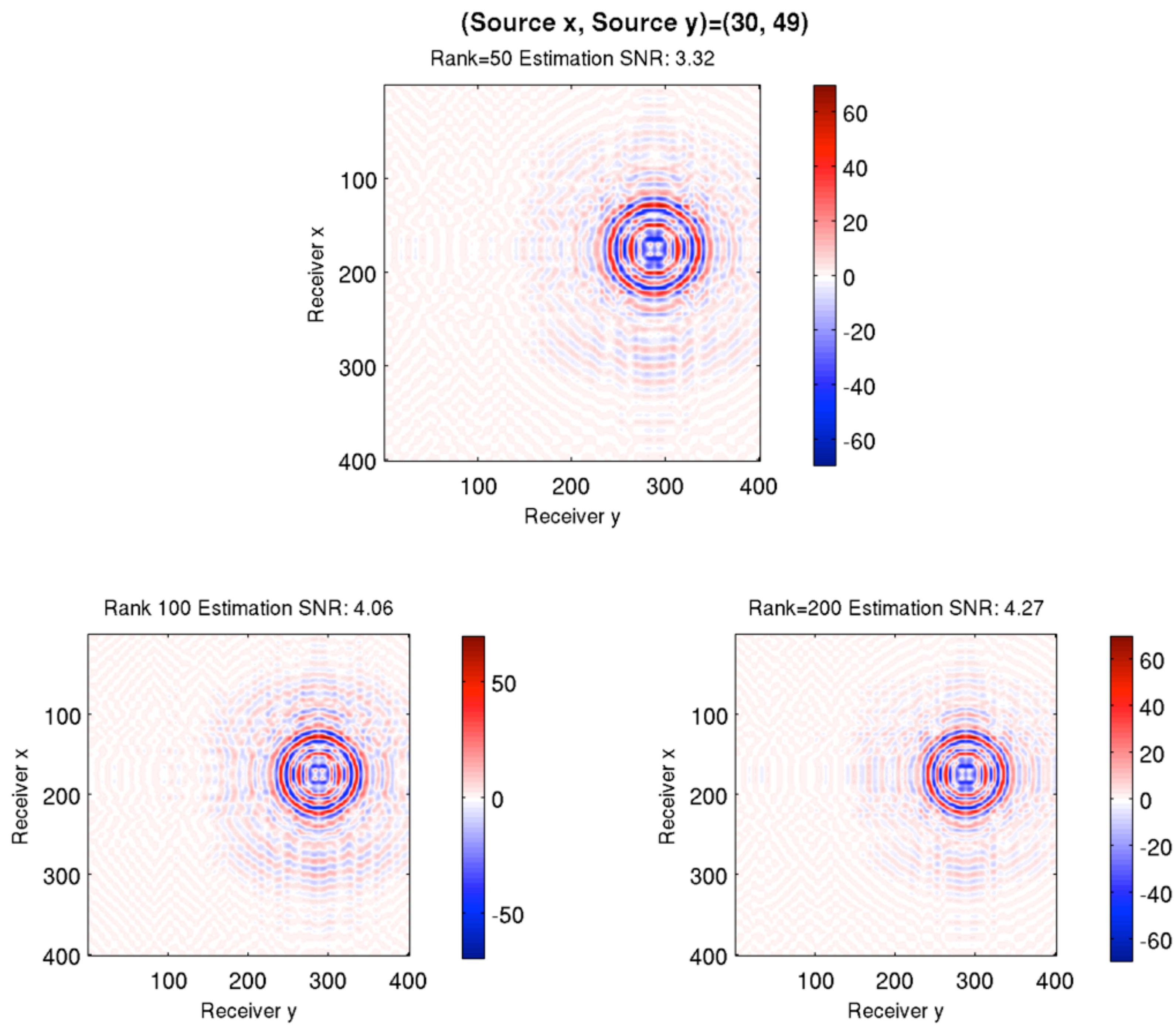
Extrapolation for fixed receiver coordinates



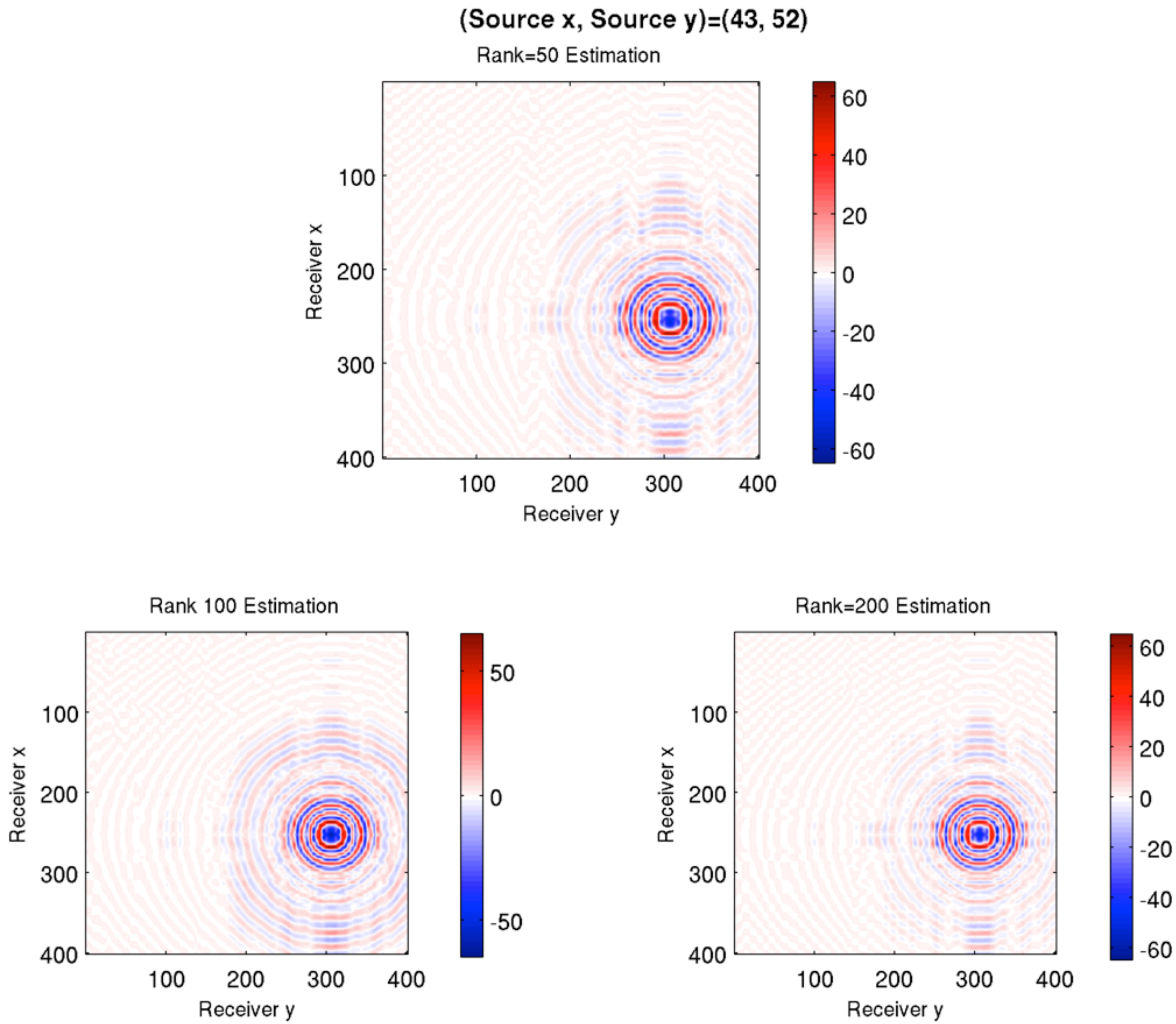
Effects of varying rank parameter on reconstructing available shot data



Effects of varying rank parameter on reconstructing shot data from test set



Effects of varying rank parameter on extrapolation of unavailable shot data



Experiments

- ~2M observations, ~30M elements in the completed tensor
- Factorization: ~30 seconds on 40 cores
- Parameter validation: 144 runs in 51 minutes

Experiments

- Recap of our results:
  - Low rank representation which can capture the inherent structure of seismic data
    - Especially evident in the receiver gather results
  - Efficient algorithm which can scale to gigabytes on workstations

Outline

- Introduction
- Our method
- Experiments
- Conclusion & Future Work

Conclusion

- Seismic data can be mapped to a low rank matrix structure
- Practical benefits:
  - Large scale interpolation
- Theoretical benefits
  - A better understanding of properties of sampling
  - Bounds on the number of necessary observations

Future Work

- Integrate the spatially continuous structure of survey in low-rank matrix completion
- Find other rank lowering transforms of seismic data to lower measurement demands in surveys
- Explicitly use low-rank structure in waveform inversion
- Related work: scaling the matrix factorization to TB sized data sets

Thank you for your attention!