Hierarchical Tucker Tensor Optimization - Applications to 4D Seismic Data Interpolation

Curt Da Silva and Felix J. Hermann

SLIM

University of British Columbia

Motivation

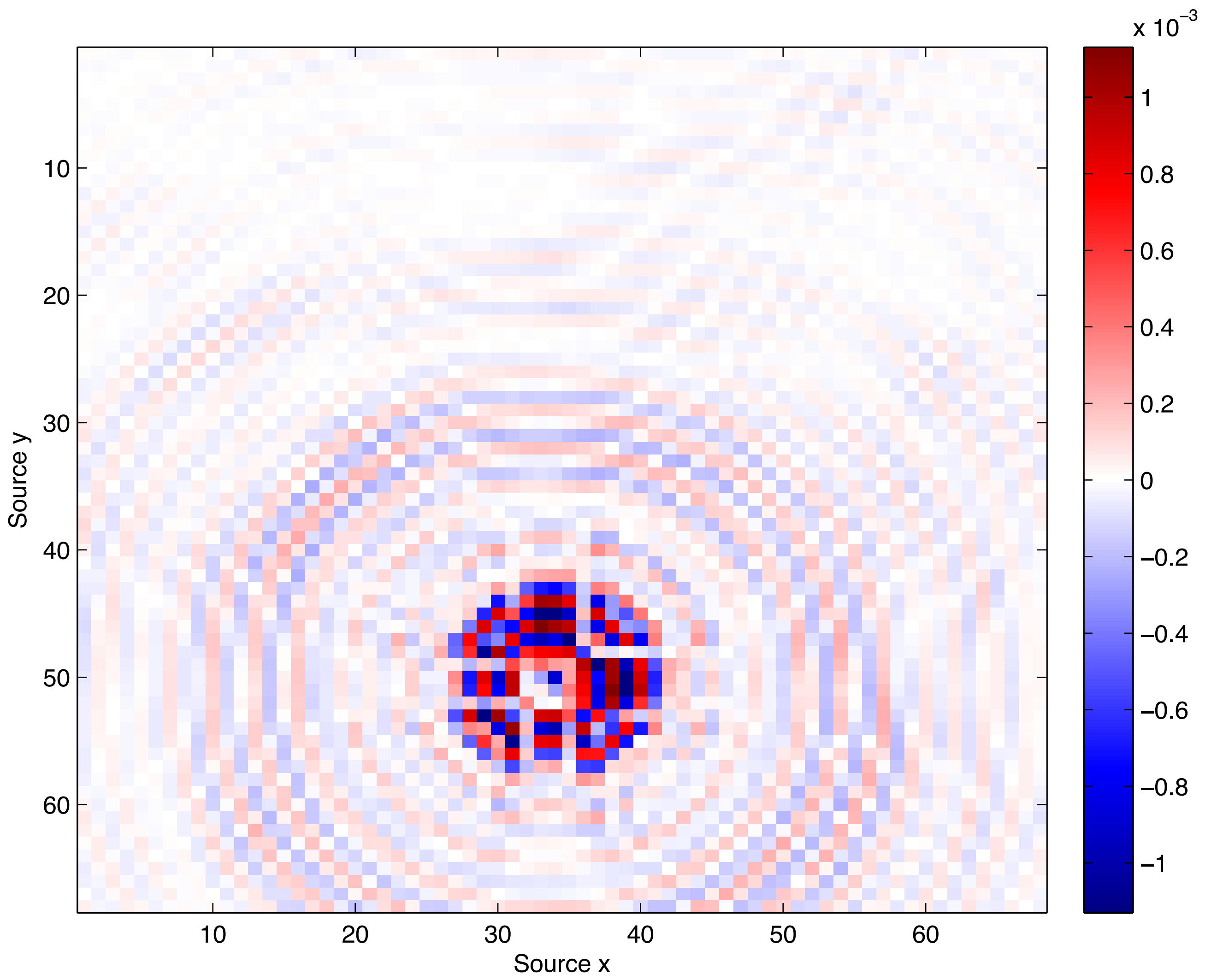
- 3D seismic experiments - 5D data
  - expensive to acquire, store
- Structured data - interpolation
- Fully sampled data
  - simultaneous sources in wave-equation based inversion
  - simulating multiples

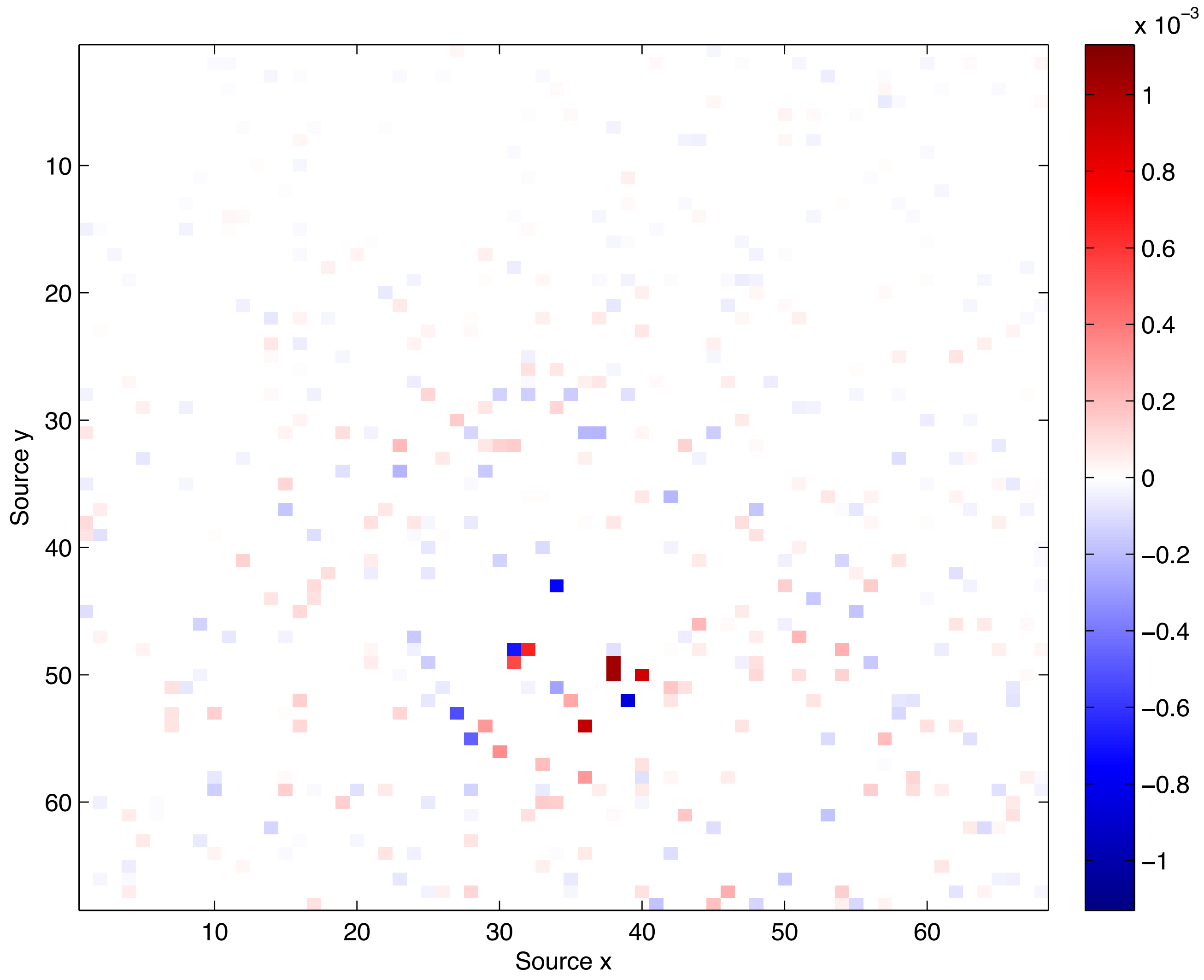
Goals

- Generalization of Compressive Sensing to multiple dimensions
- Randomized source/receiver acquisition
  - Reduce acquisition financial/time costs

7.34 Hz - 75% missing sources

Common receiver gather



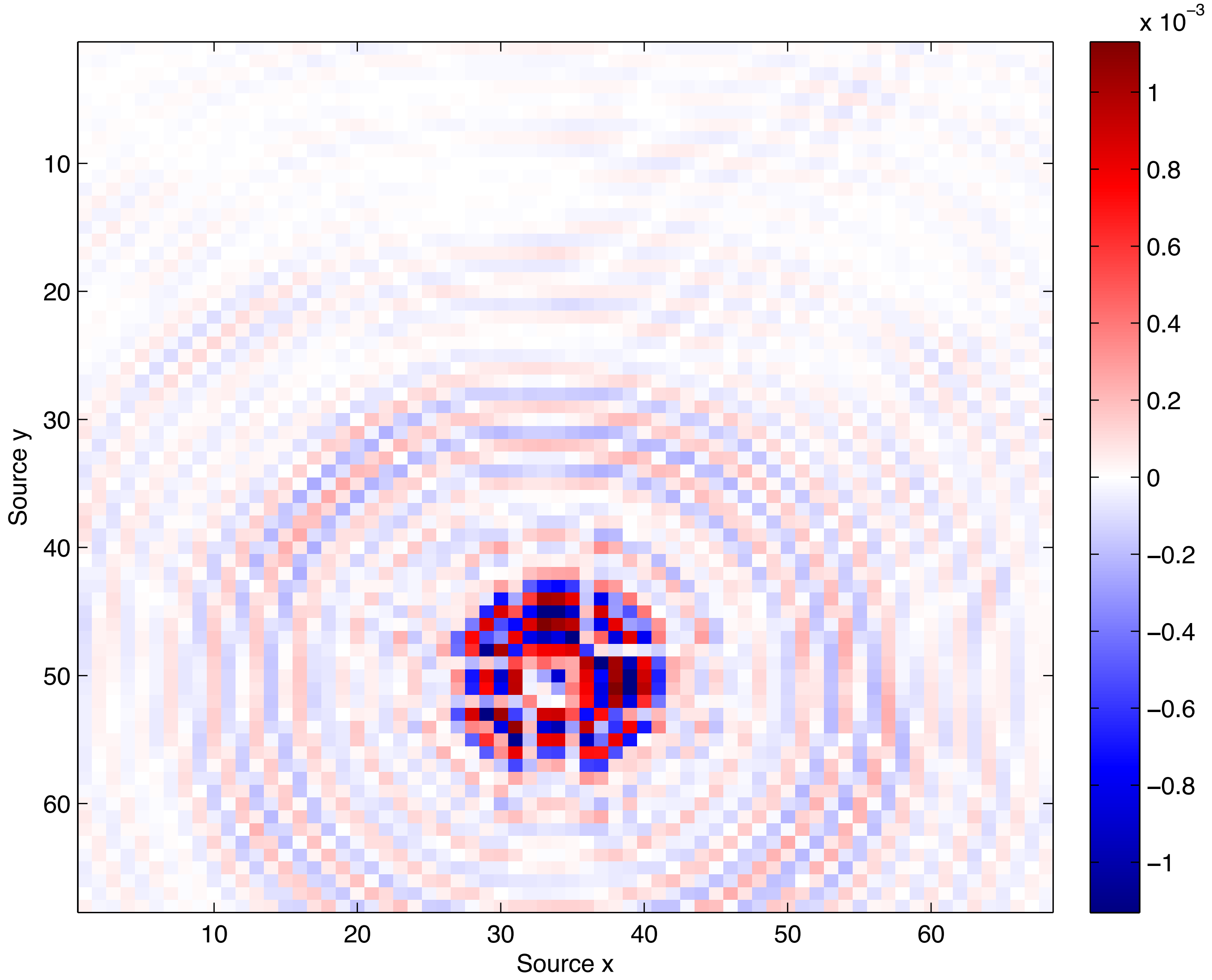


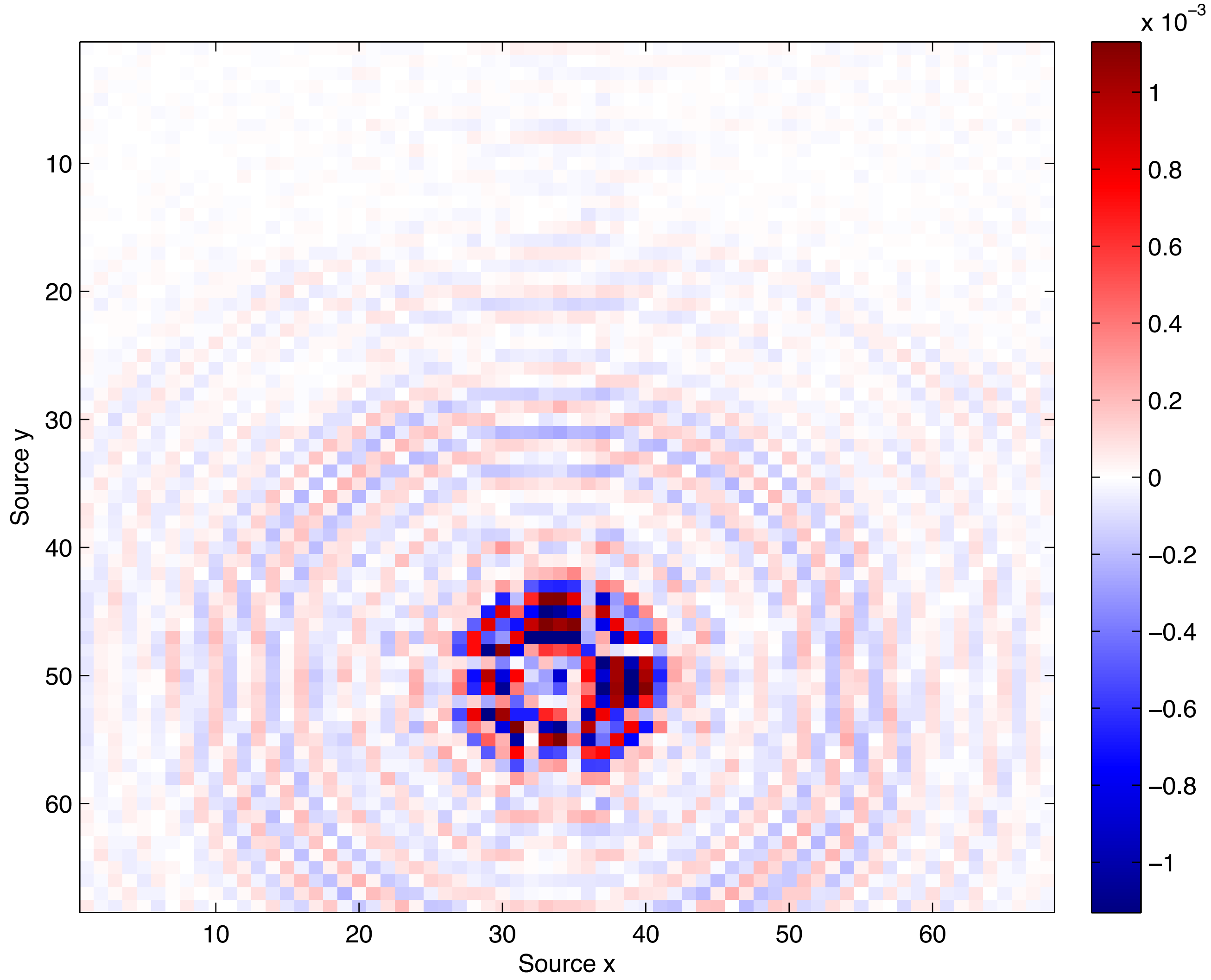
$(x_{rec}, y_{rec}) = (75,50)$

Subsampled Data

7.34 Hz - 75% missing sources

Common receiver gather





$(x_{rec}, y_{rec}) = (75,50)$

Interpolated Data

SNR 11.3 dB

Compressive sensing

with sparsity promotion

Successful reconstruction scheme

- Signal structure - *sparsity*
- Sampling - *subsampling decreases sparsity*
- Optimization - *look for sparsest solution*

Multidimensional interpolation
with Hierarchical Tucker

Successful reconstruction scheme

- **Signal structure - *Hierarchical Tucker***
- Sampling - *subsampling increases h-rank*
- Optimization - *fit data in the Hierarchical Tucker format*

Matricization

- The *matricization* of a tensor X with dimensions $1, \ldots, d$ along the dimensions $t = (t_1, \ldots, t_r)$ is the matrix formed by placing the dimensions t along the rows and dimensions t^c along the columns
- Denoted $X^{(t)}$

Example in Matlab

```
n1 = 20; n2 = 20; n3 = 20; n4 = 20;
% Tensor
x = randn(n1,n2,n3,n4);
% Matricization along dimensions 1 and 2
x12 = reshape(x,n1 * n2, n3 * n4);
% Matricization along dimensions 3 and 4
y34 = permute(x,[3 4 1 2]);
x34 = reshape(x, n3 * n4, n1 * n2);
% Matricization along dimensions 1 and 3
y13 = permute(x,[1 3 2 4]);
x13 = reshape(x,n1 * n3, n2 * n4);
```

$X^{(1,2)}$

$X^{(3,4)}$

$X^{(1,3)}$

Kronecker product

A, B Matrices

$B \otimes A$ Kronecker product

$$(B \otimes A)\mathrm{vec}(X) = \mathrm{vec}(AXB^T)$$

A acts along the columns of X

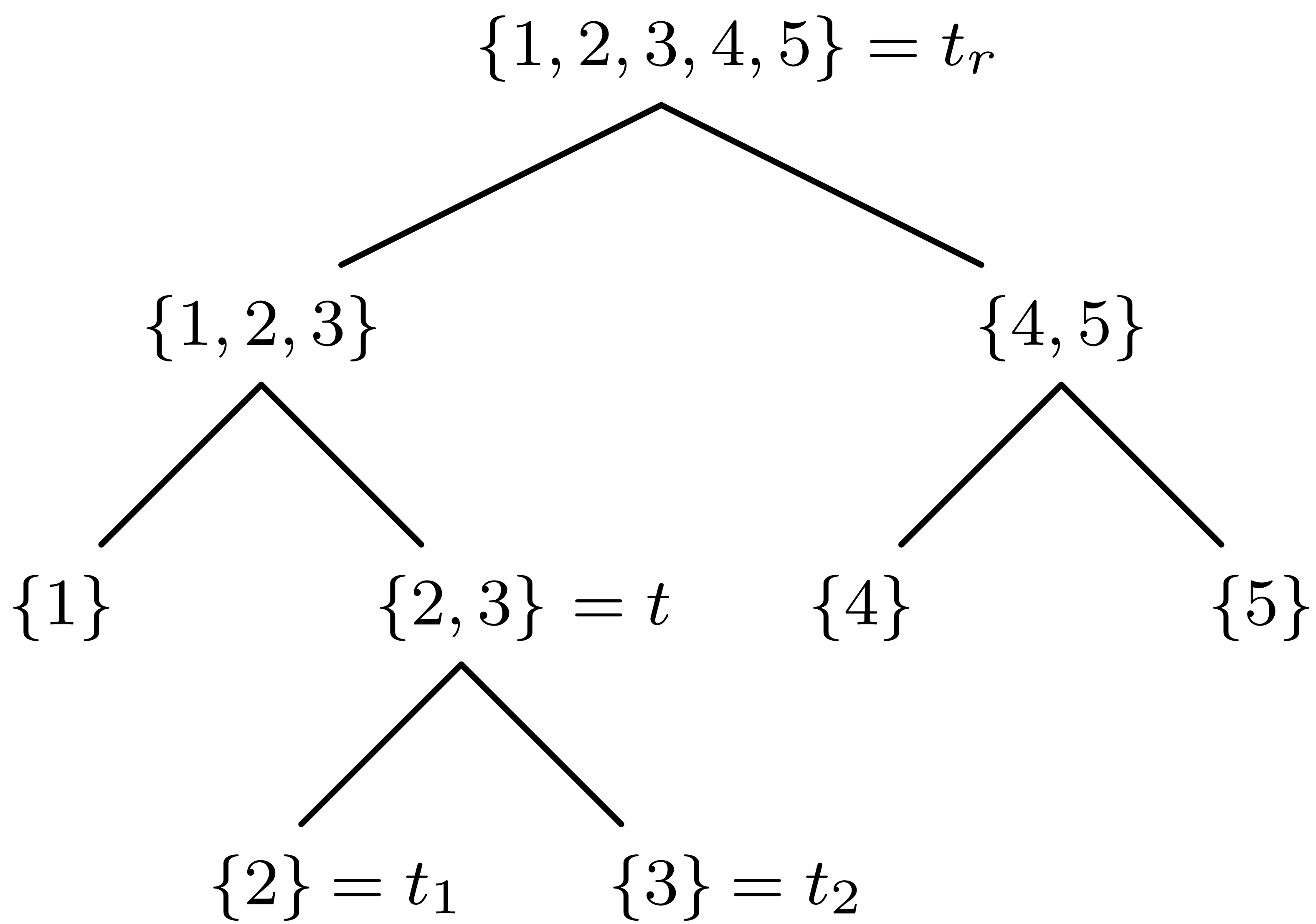
B acts along the rows of X

Hierarchical Tucker Format

- A *dimension tree* T for dimensions $\{1, \ldots, d\}$ is a non-trivial binary tree such that
  - the root, t_{root}, is has the label $\{1, \ldots, d\}$
  - each non-leaf node, t, can be written as $t = t_l \cup t_r$ $t_l \cap t_r = \emptyset$ where t_l is the left child of t, t_r is the right child of t

A. Uschmajew, B. Vandereycken. The geometry of algorithms using hierarchical tensors. 2012

Example



Hierarchical Tucker Format

- A tensor X can be written in the *Hierarchical Tucker format* corresponding to a dimension tree T and a vector of hierarchical ranks
$$(k_t)_{t \in T}, k_{\text{root}} = 1$$
if it can be written as

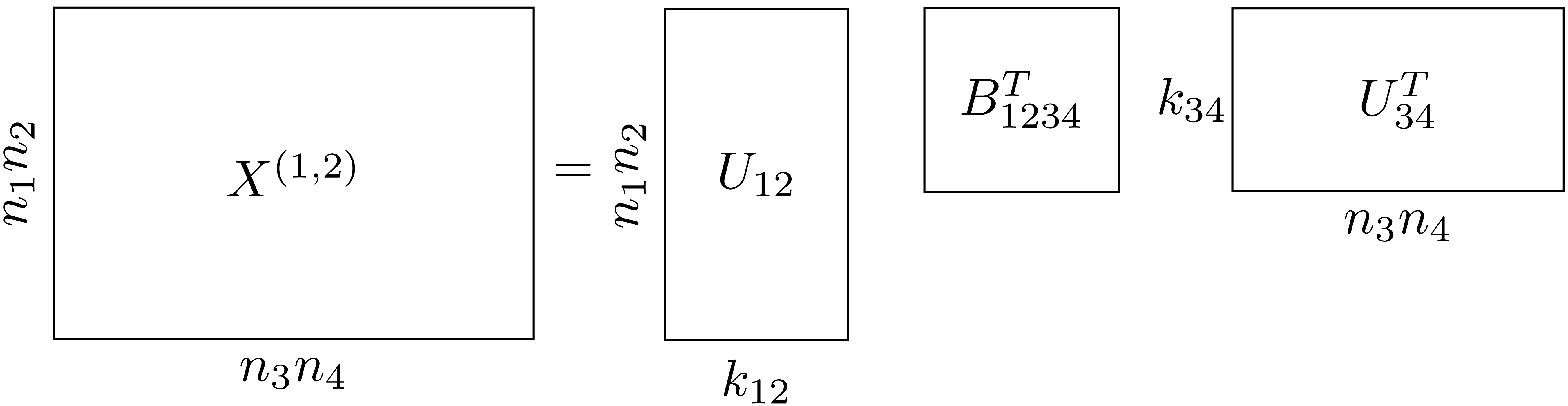
Hierarchical Tucker Format

$$\mathrm{vec}(X) = (U_{t_l} \otimes U_{t_r}) B_{t_{\mathrm{root}}}^{(1,2)} \quad t = t_{\mathrm{root}}$$

$$U_t = (U_{t_l} \otimes U_{t_r}) B_t^{(1,2)} \quad t \text{ not a leaf}$$

$$U_t \in \mathbb{C}^{n_t \times k_t} \qquad B_t \in \mathbb{C}^{k_l \times k_r \times k_t}$$

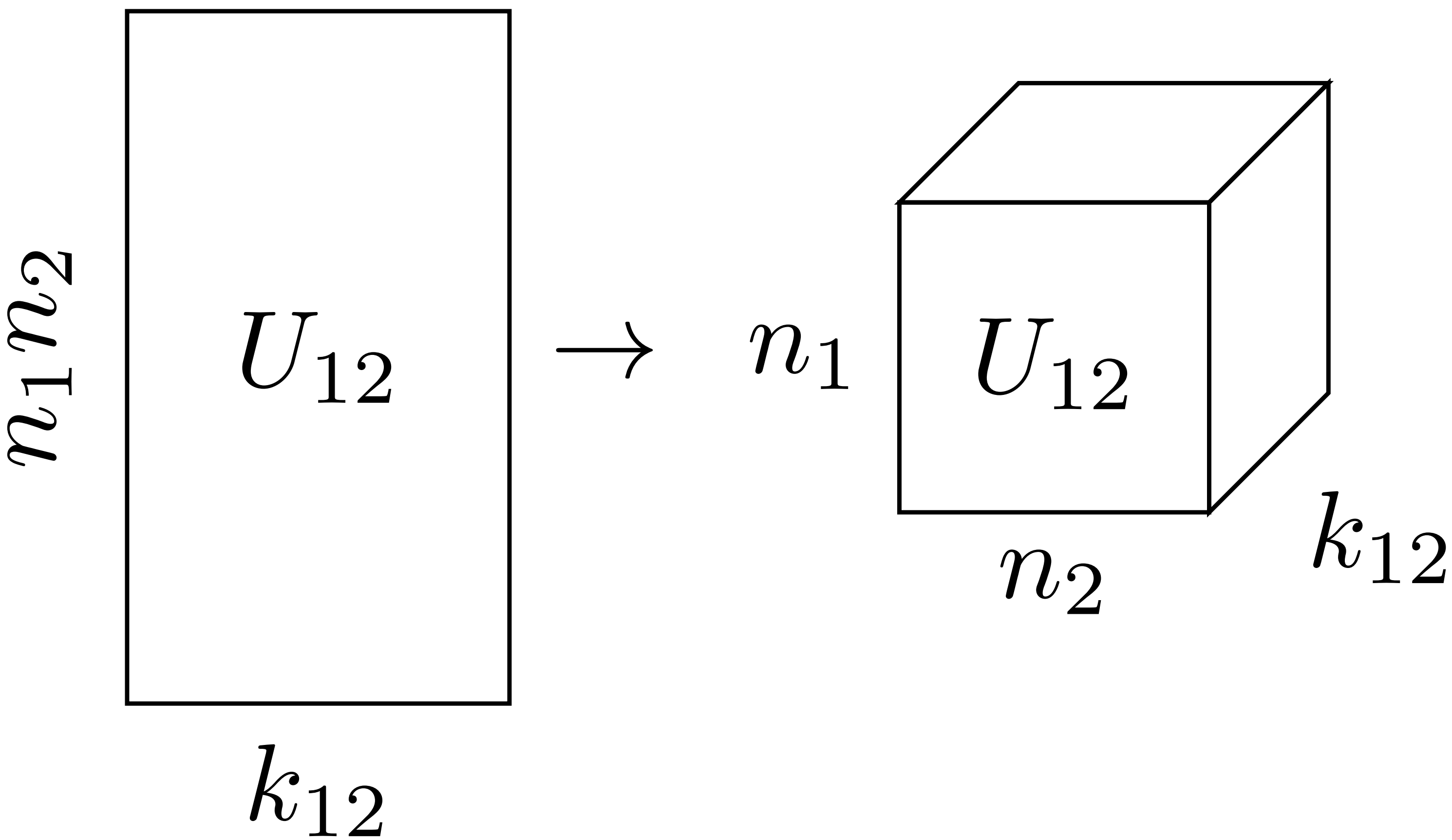
HT Format $X - n_1 \times n_2 \times n_3 \times n_4$ tensor



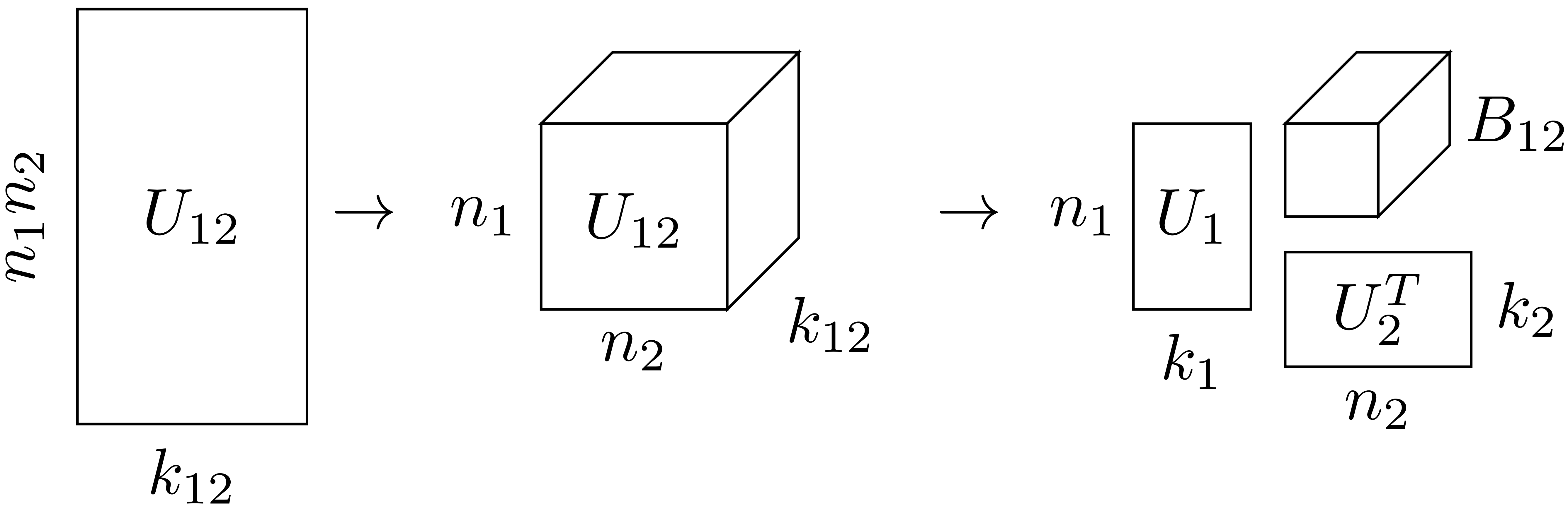
$$\mathrm{vec}(X) = (U_{12} \otimes U_{34})\mathrm{vec}(B_{1234})$$

$$X^{(1,2)} = U_{12} B^T_{1234} U^T_{34}$$

HT Format $X - n_1 \times n_2 \times n_3 \times n_4$ tensor



HT Format $X - n_1 \times n_2 \times n_3 \times n_4$ tensor

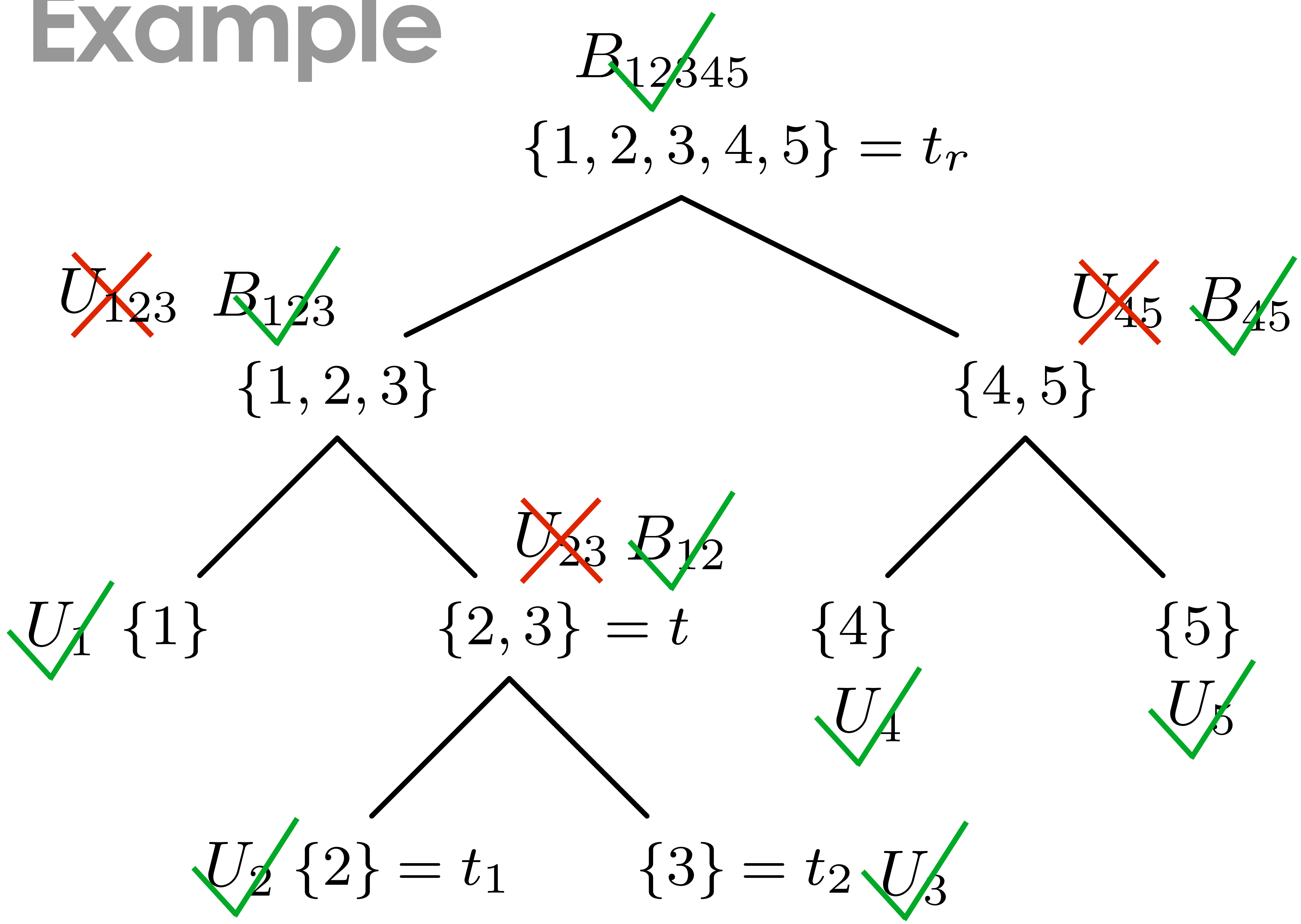


Hierarchical Tucker Format

- Intermediate matrices don’t need to be stored
- U_t, B_t - small parameter matrices
  - specify the tensor completely

A. Uschmajew, B. Vandereycken. The geometry of algorithms using hierarchical tensors. 2012

Example



Hierarchical Tucker Format

- Storage $\leq dNK + (d-2)K^3 + K^2$
- Compare to N^d storage for the full tensor
- Effectively breaking the curse of dimensionality when $K \ll N$
- Low frequency data compresses

Hierarchical Tucker Format

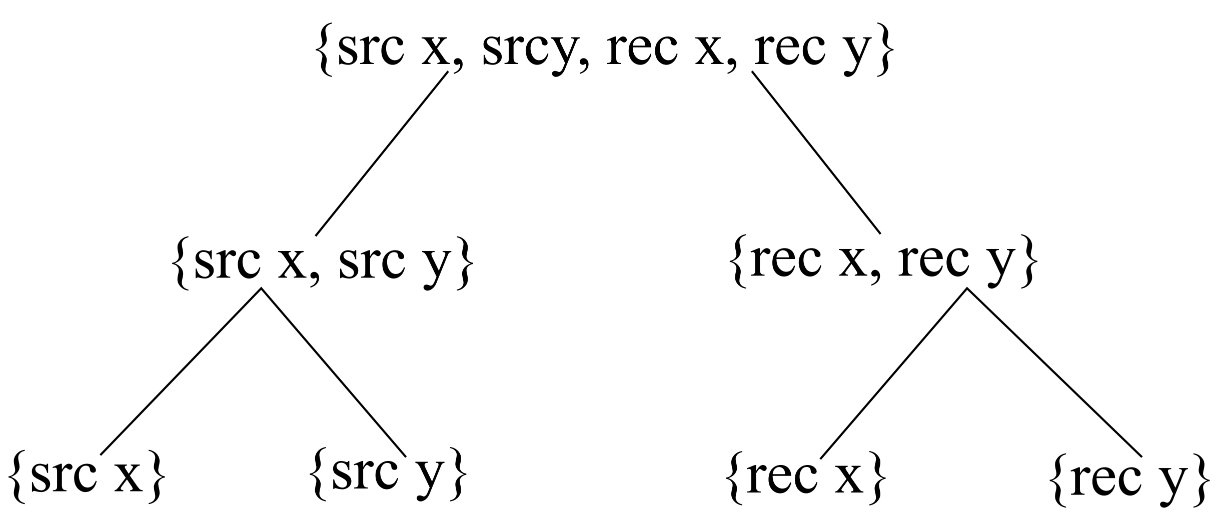
- E.g. for a 100 x 100 x 100 x 100 cube with max rank 20, $N = 100 \quad d = 4 \quad k = 20$
  Naive storage: $N^d = 10^8$ parameters
  HTucker storage: $= 24400$ parameters
  Compression of a factor of *99.97%*

Seismic HTucker

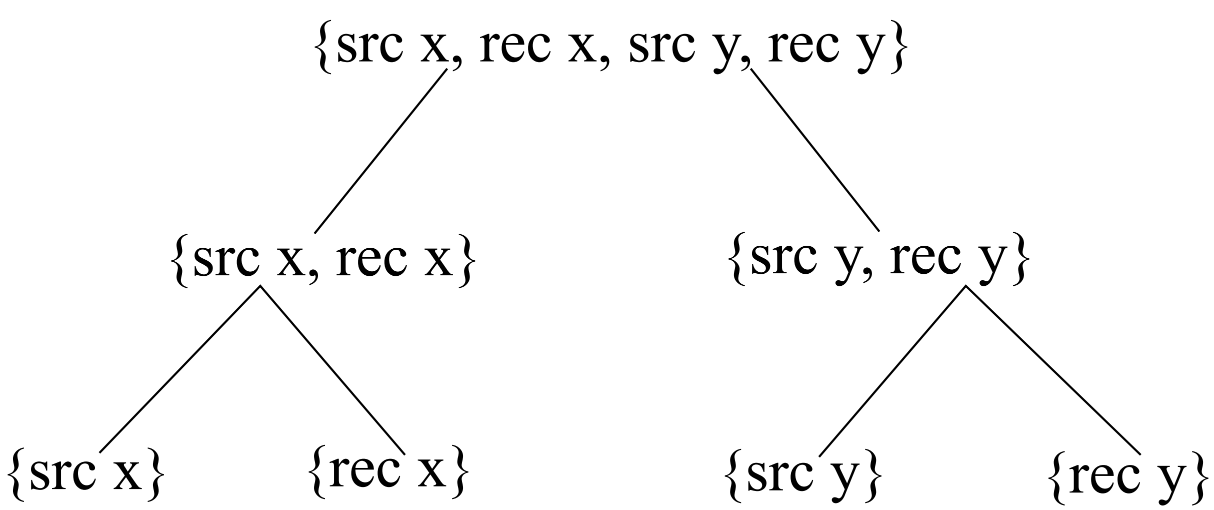
- We consider a 3D seismic survey with coordinates (src x, src y, rec x, rec y, time)
- We take a Fourier transform in time and restrict ourselves to a single frequency slice

Seismic HTucker

For a frequency slice with coordinates (src x, src y, rec x, rec y), there are essentially two choices of dimension splitting (by reciprocity)

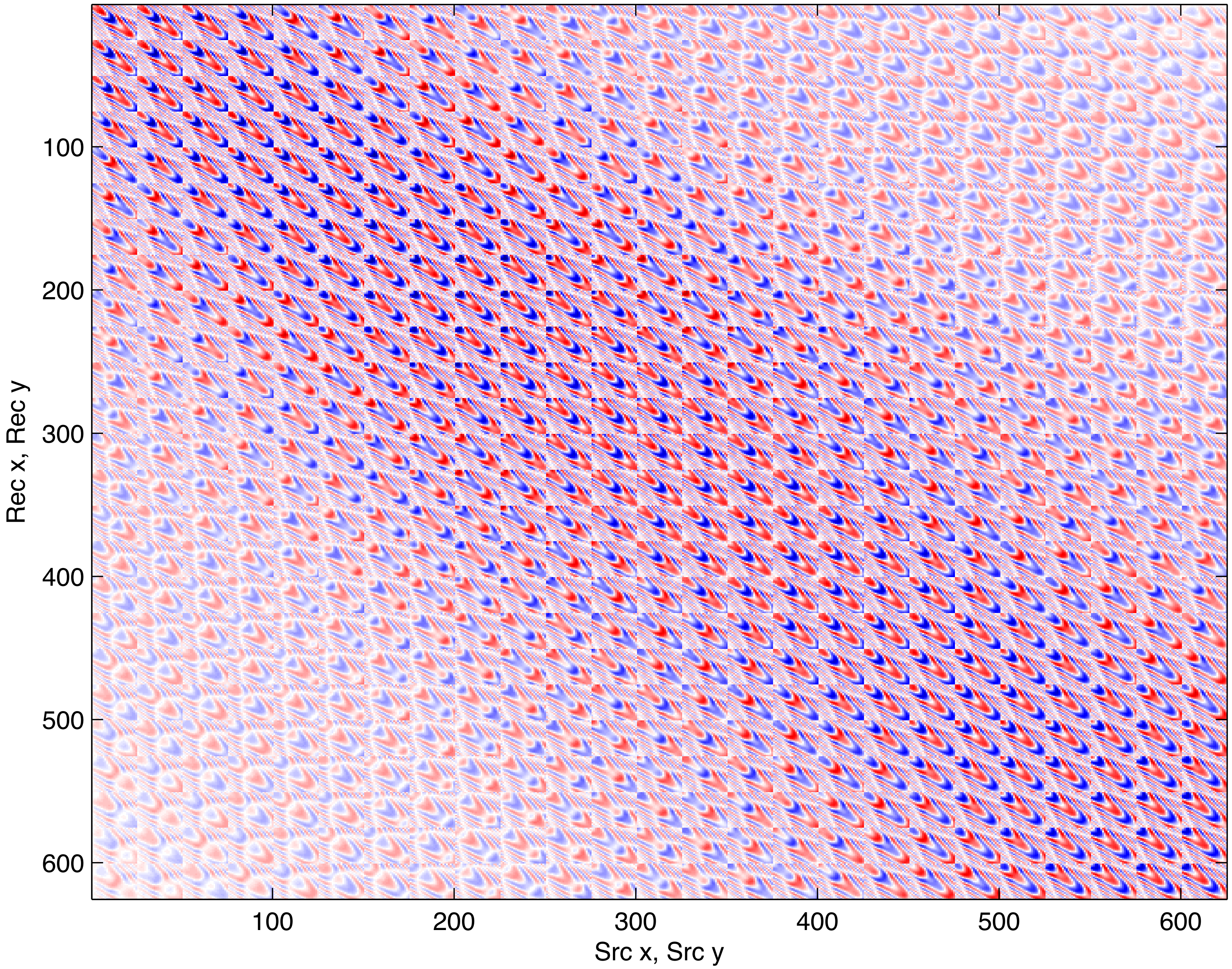


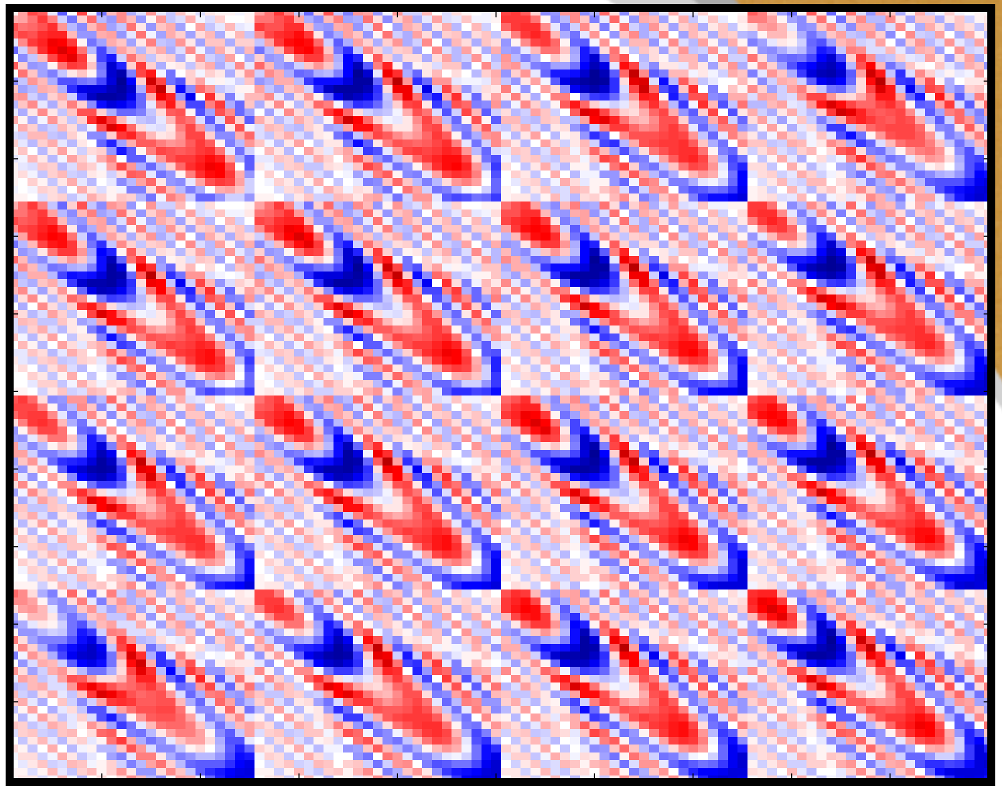
Canonical Decomposition

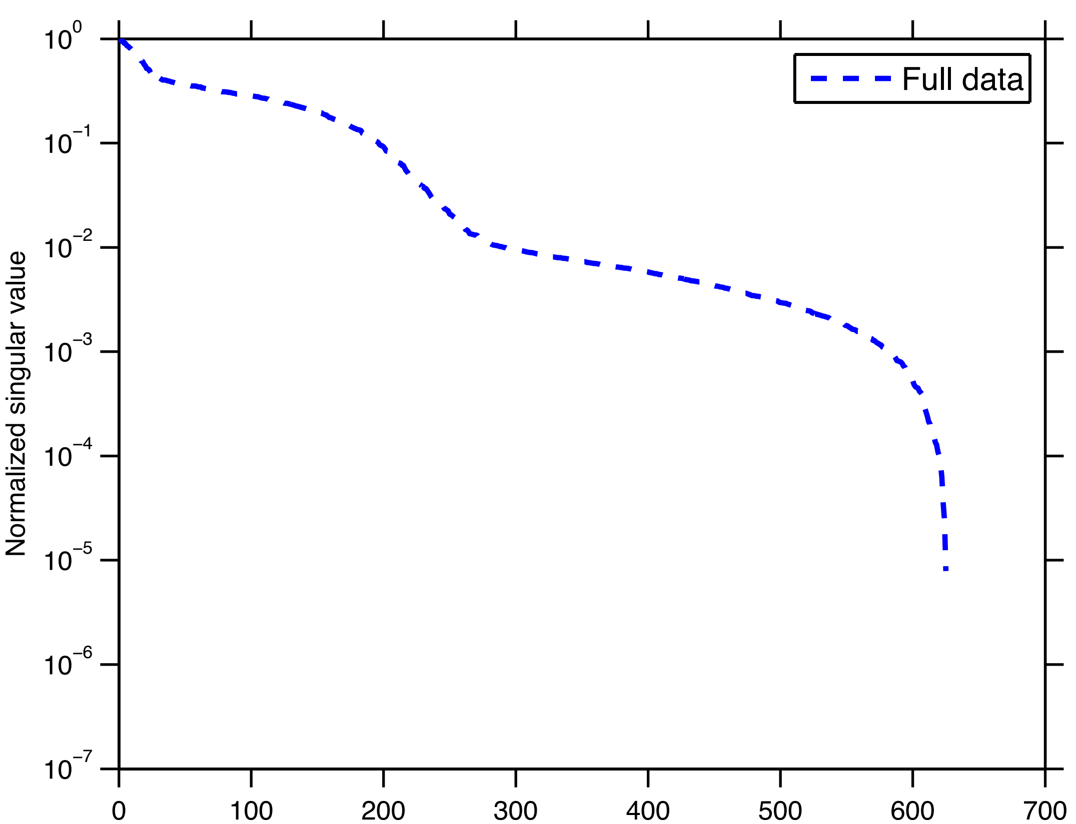


Non-canonical Decomposition

Matricizations

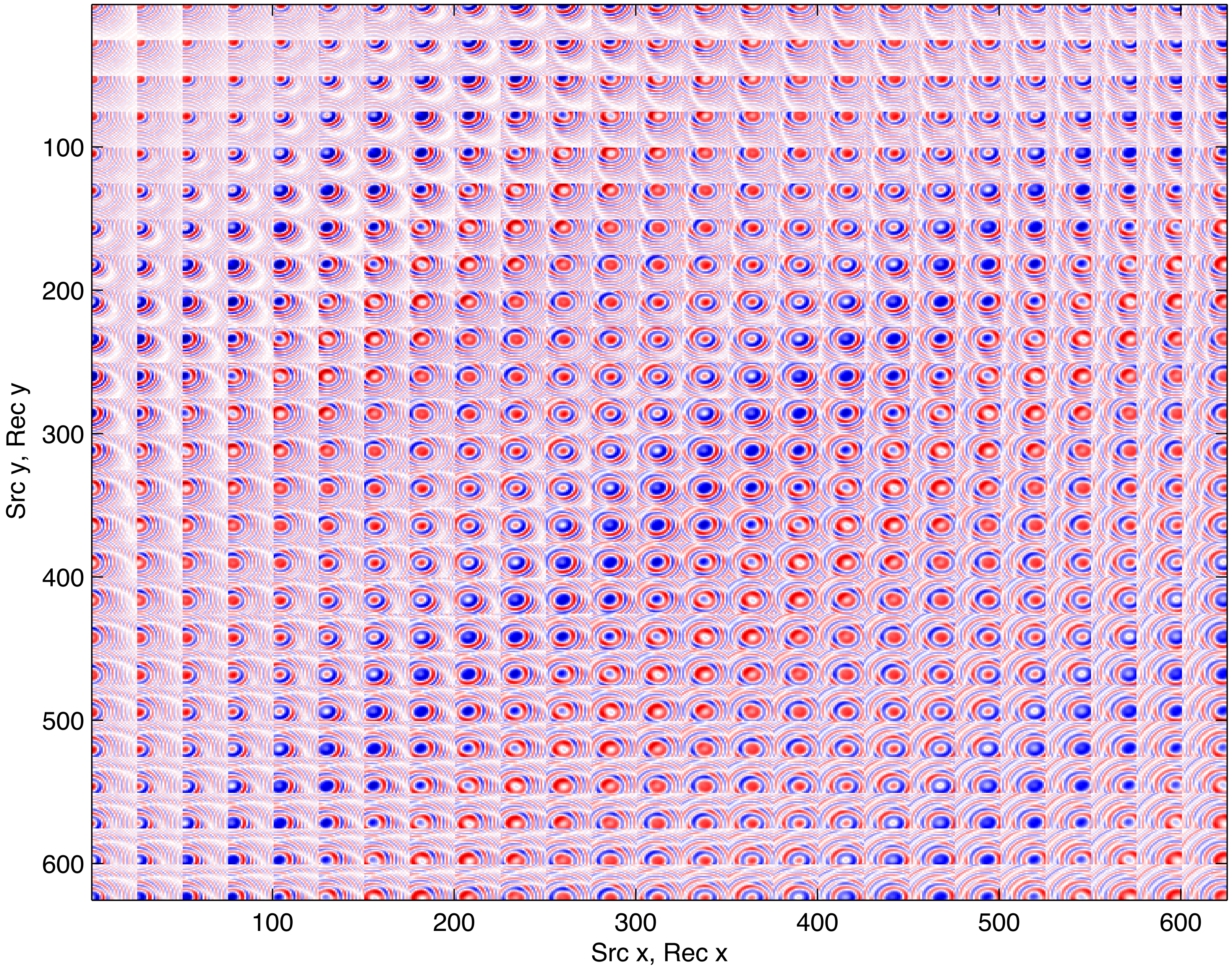


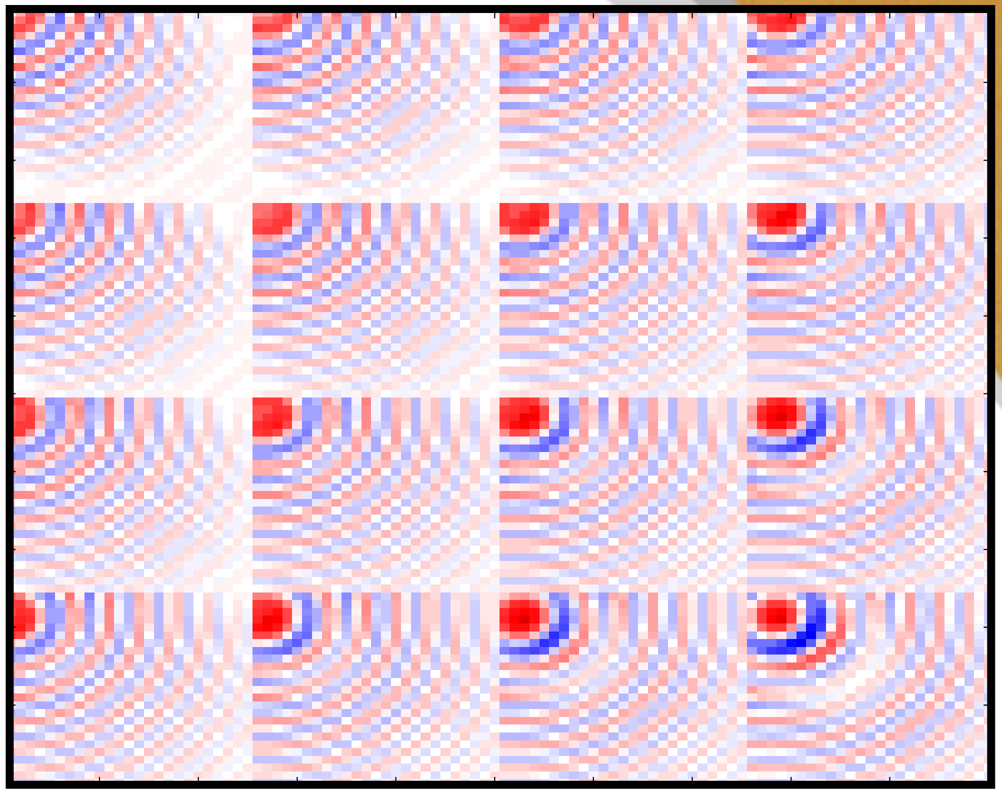


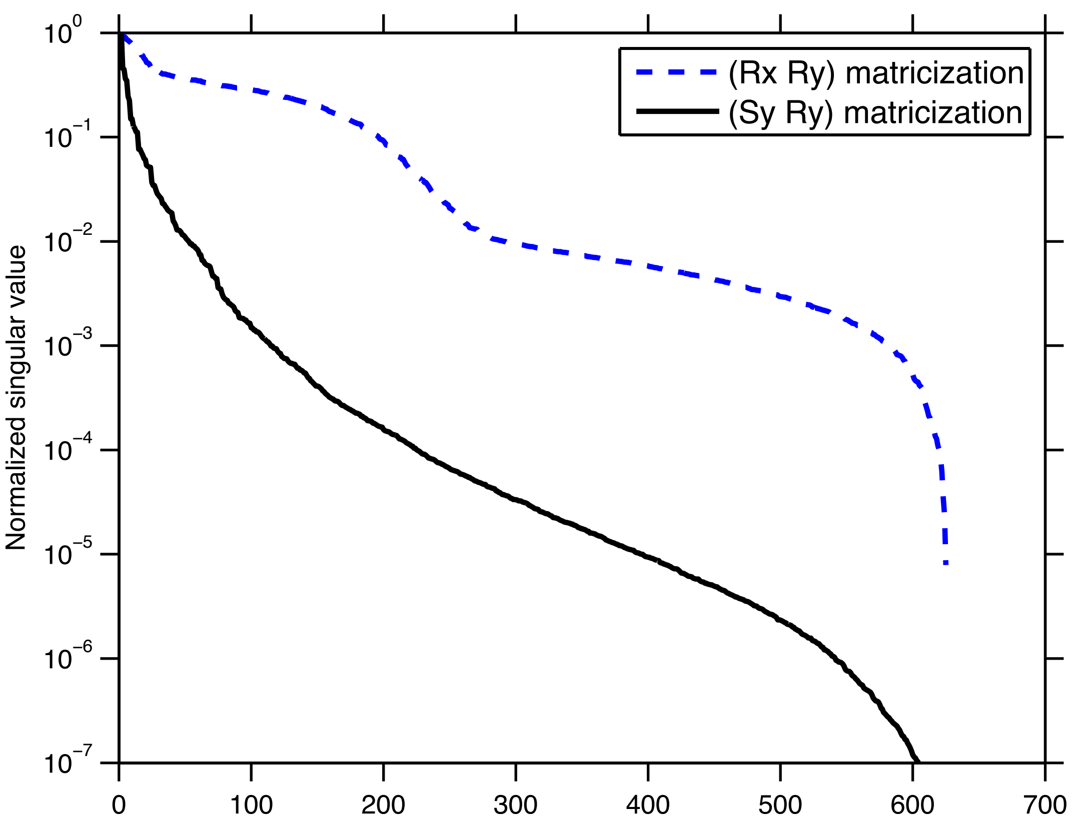


(Rec x, Rec y) matricization - full data

Matricizations







(Src y, Rec y) matricization - full data

Multidimensional interpolation

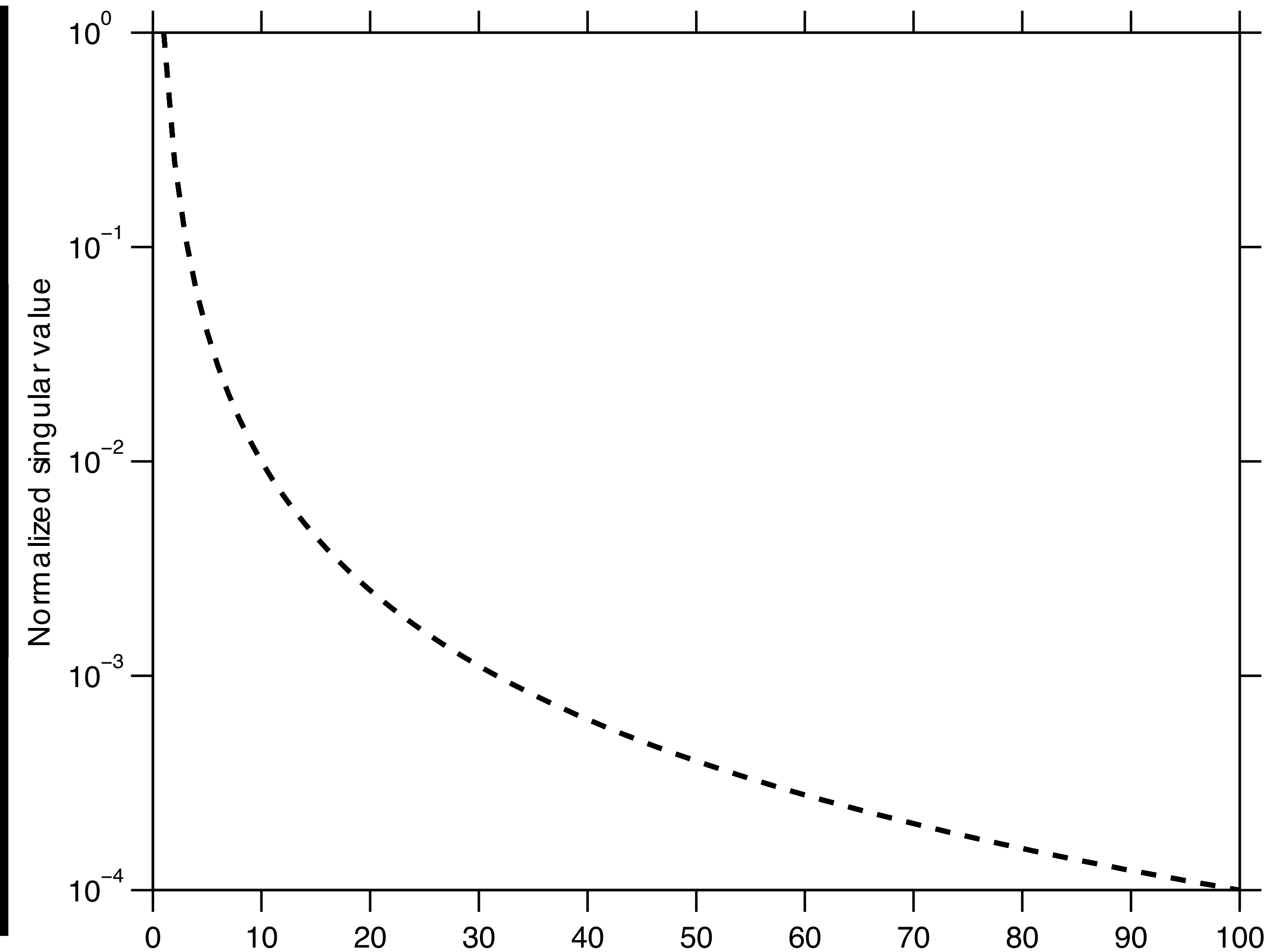
Successful reconstruction scheme

- Signal structure - *Hierarchical Tucker*
- **Sampling - *subsampling increases h-rank***
- Optimization - *fit data in the Hierarchical Tucker format*

Matrix Completion

$$\mathbf{X}$$

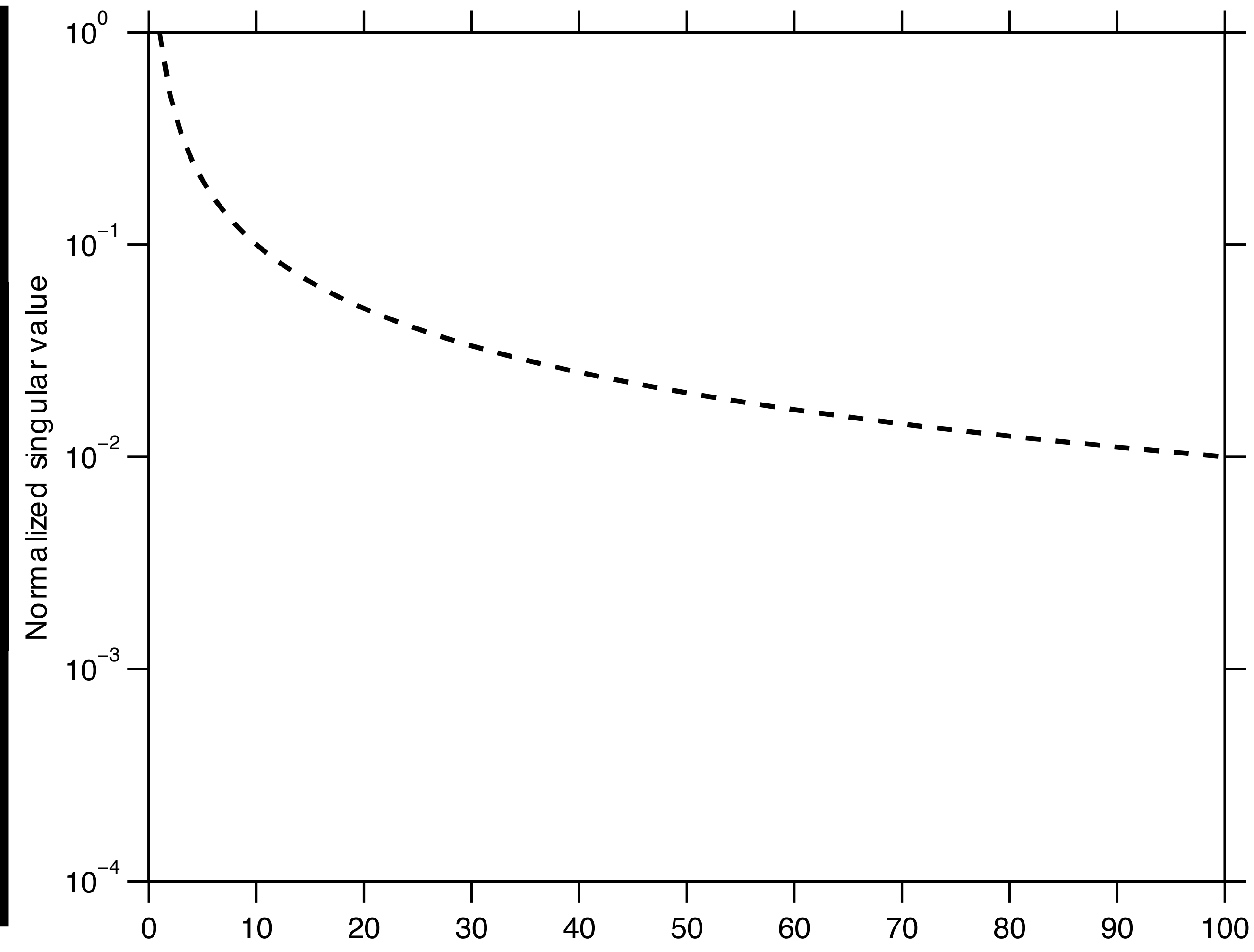
$$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}$$



Matrix Completion

$$\mathcal{A}(\mathbf{X})$$

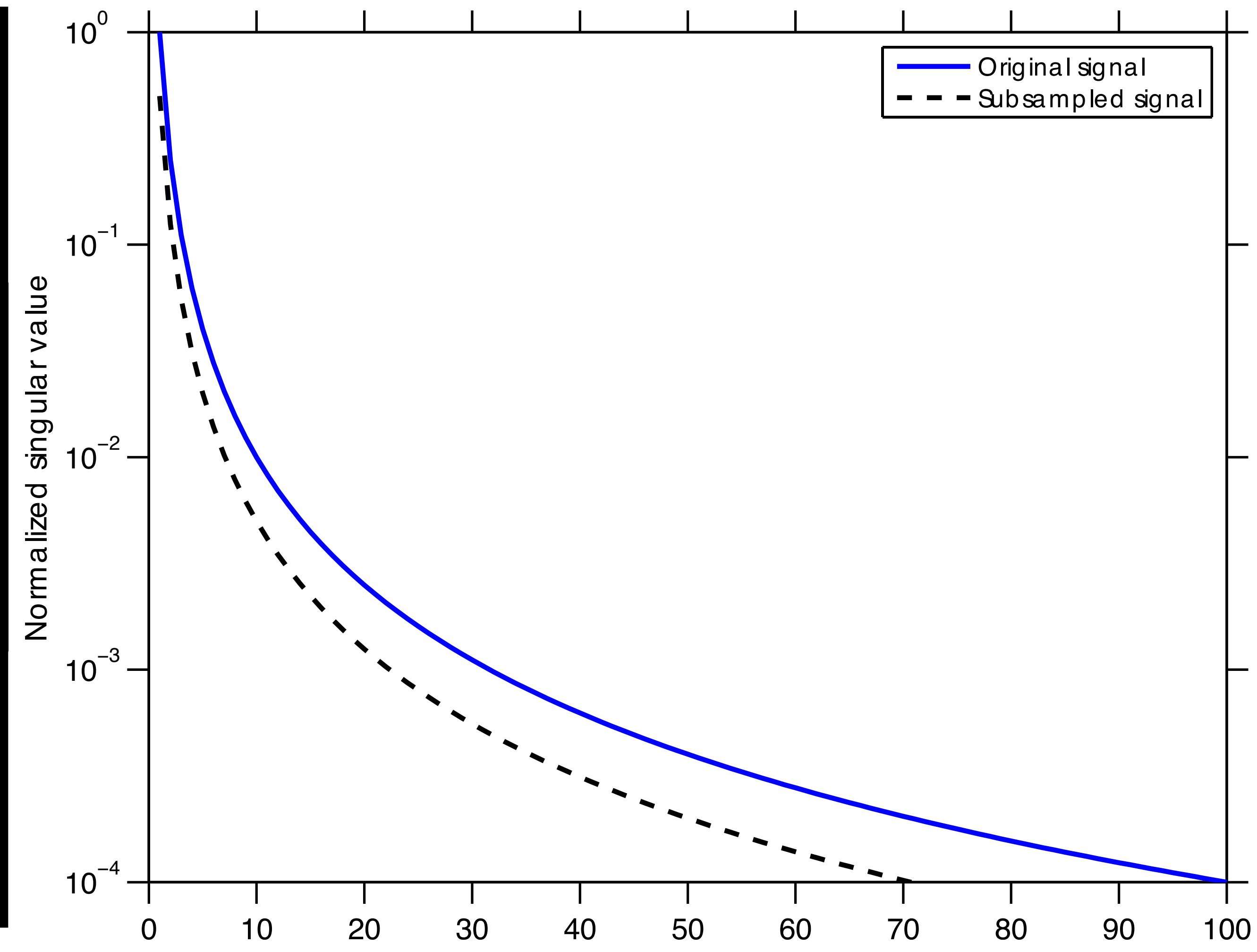
$$\begin{bmatrix} * & * & * & 0 & * \\ * & 0 & 0 & * & 0 \\ * & * & * & * & * \\ * & * & 0 & * & * \\ 0 & * & * & * & 0 \end{bmatrix}$$



Matrix Completion

$$\mathcal{A}(\mathbf{X})$$

$$\begin{bmatrix} * & * & * & 0 & * \\ * & * & * & 0 & * \\ * & * & * & 0 & * \\ * & * & * & 0 & * \\ * & * & * & 0 & * \end{bmatrix}$$



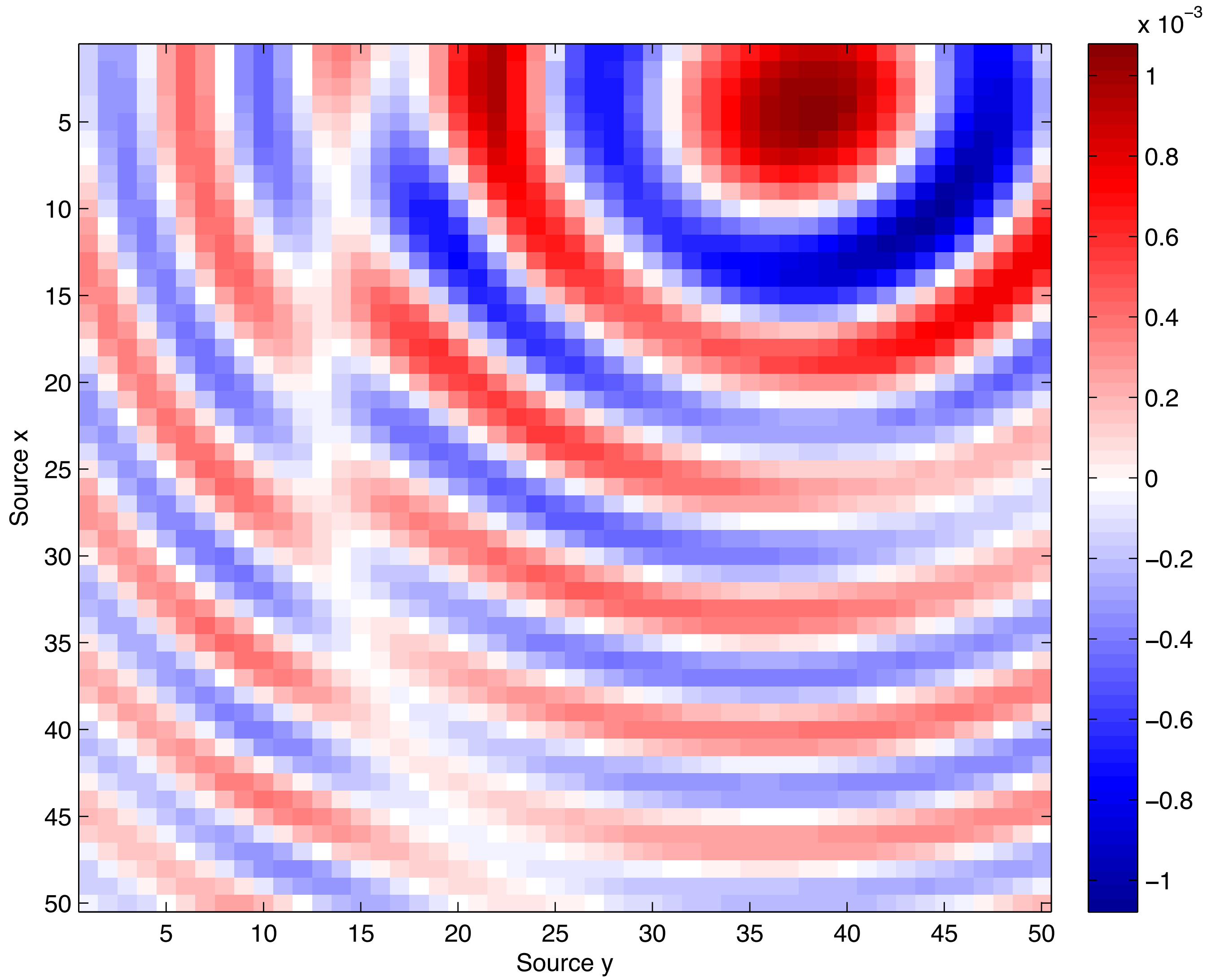
Tensor Completion

- *Structure* - recover a *tensor* $\mathbf{X}$ which has low *hierarchical* rank
  - Well represented in HT
- *Sampling* - random removal of points *increases* rank
  - Poorly represented in HT
  - Idealized sampling

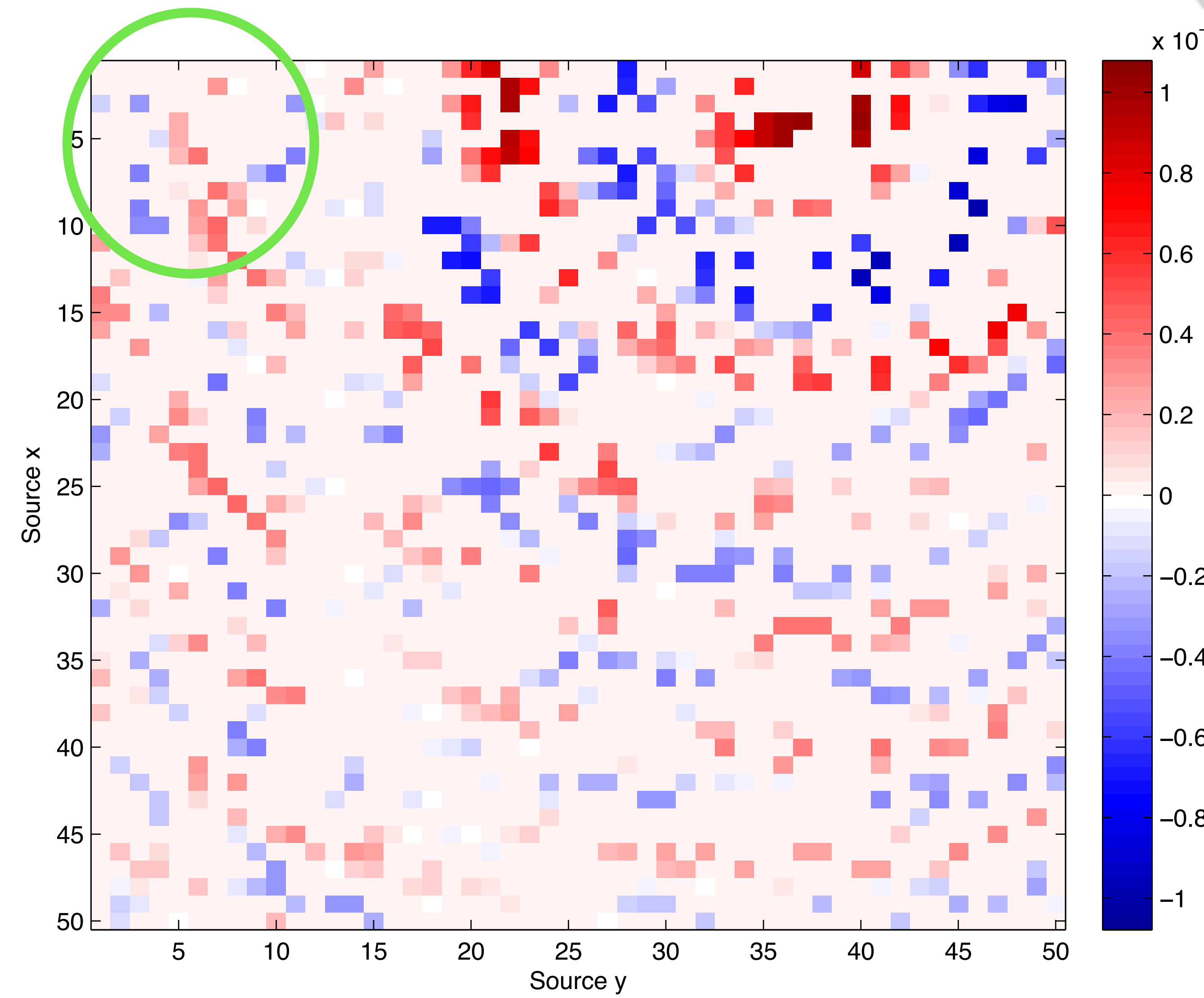
Idealized recovery

75% random entries removed

Common shot gather



True Data

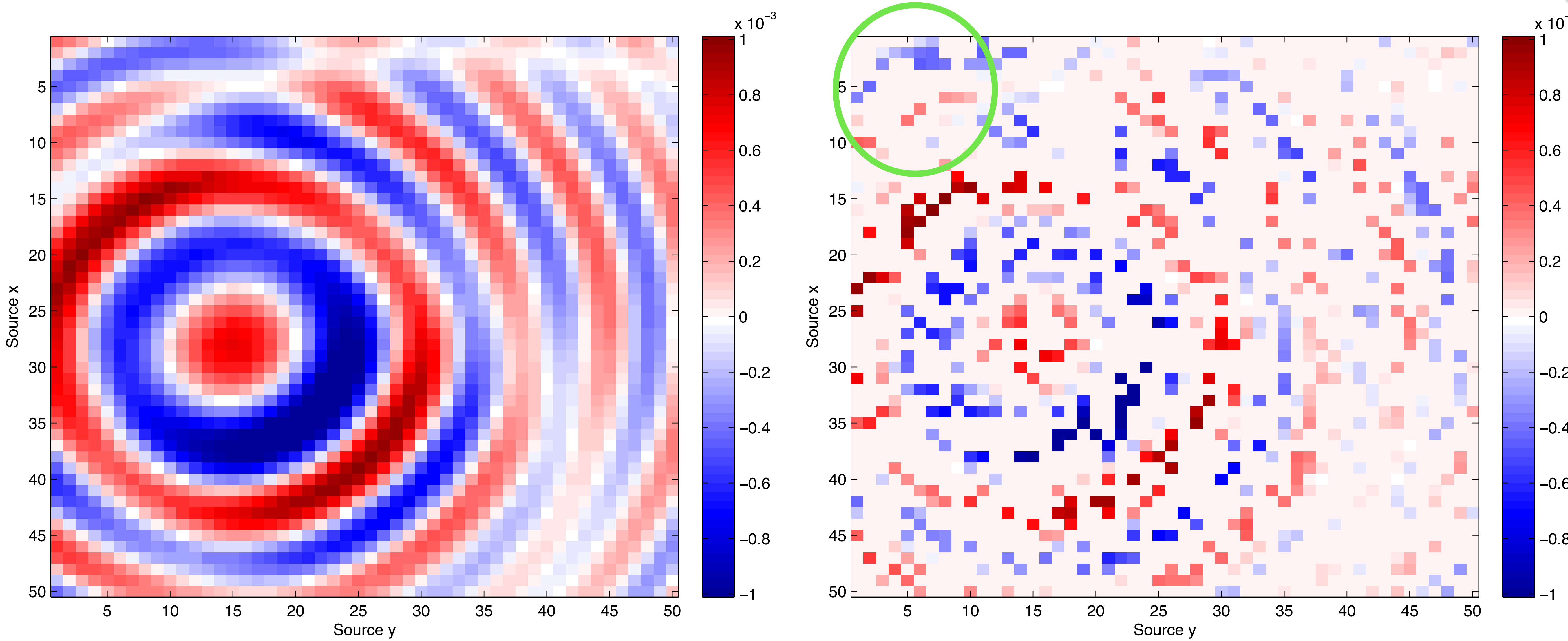


Subsampled Data

Idealized recovery

75% random entries removed

Common shot gather

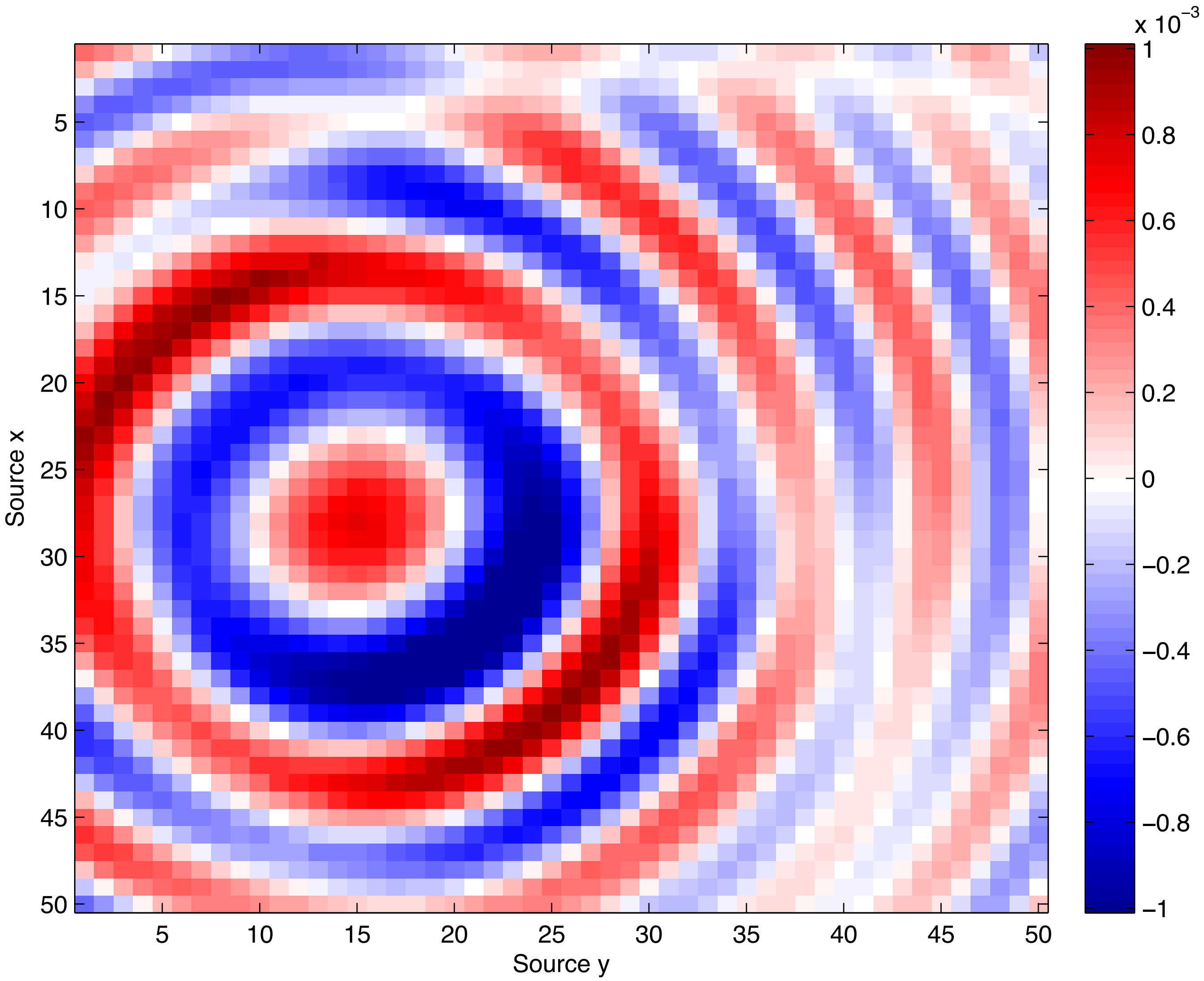


True Data

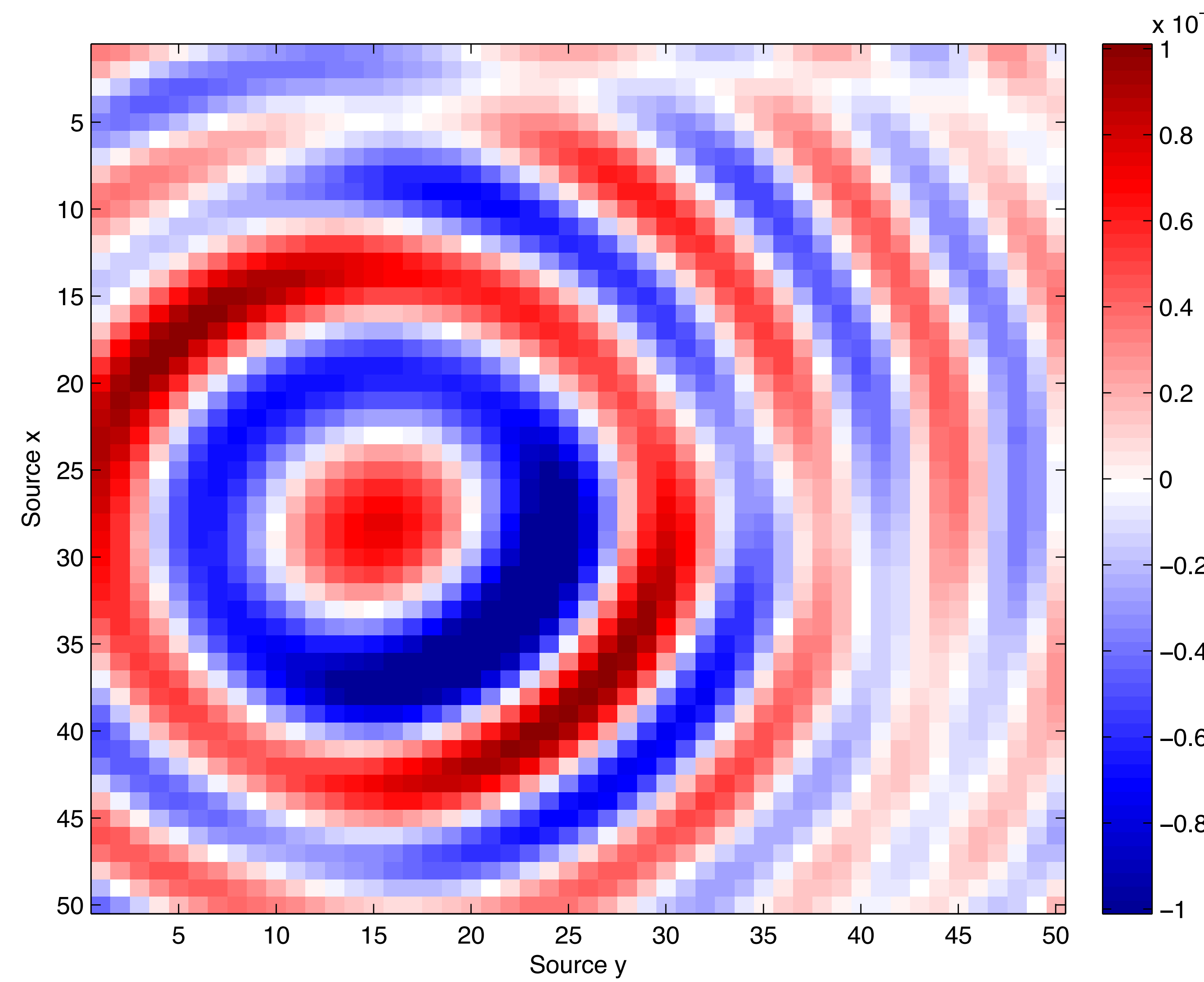
Subsampled Data

Idealized recovery

75% random entries removed
Common shot gather



True Data



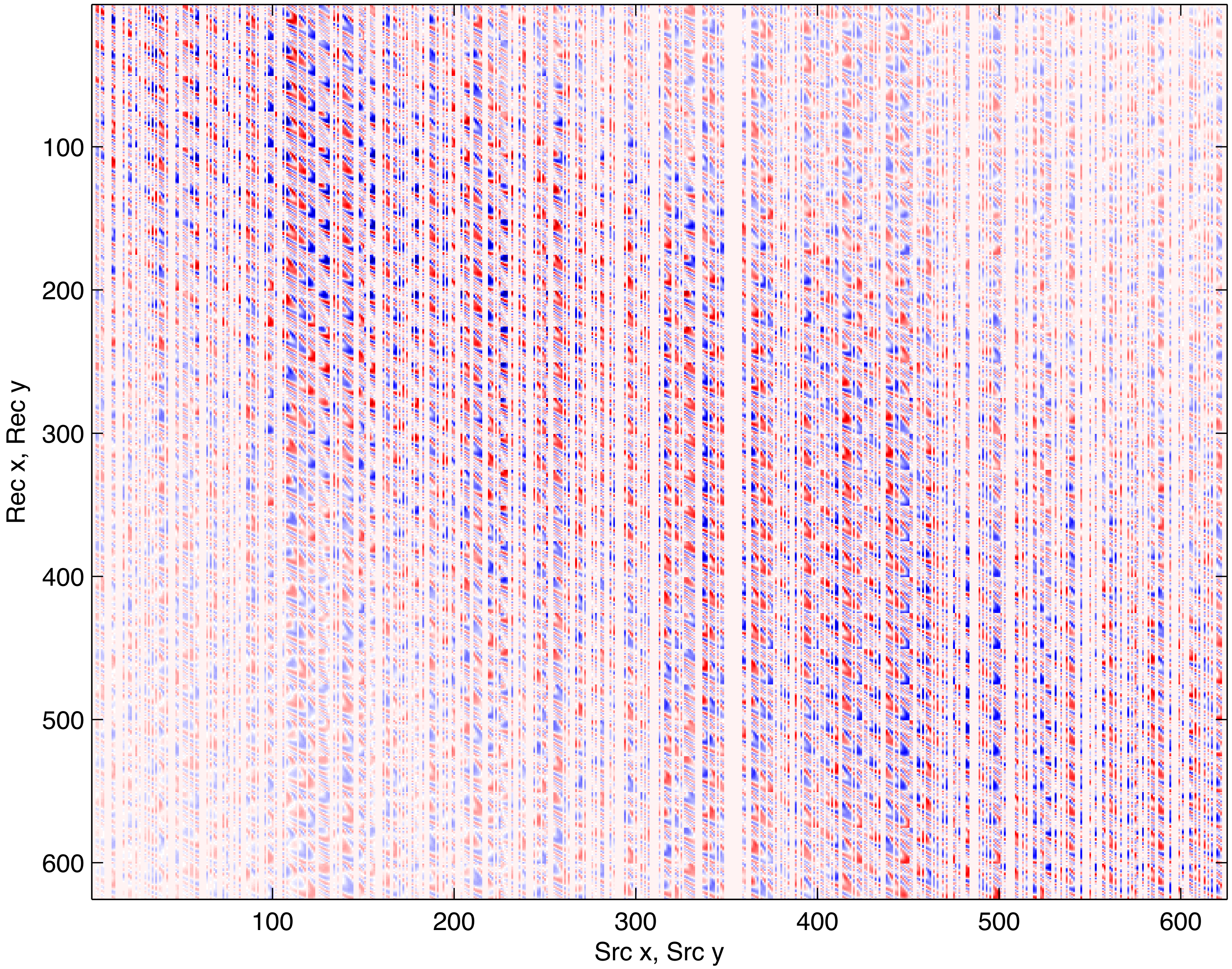
Recovered Shot

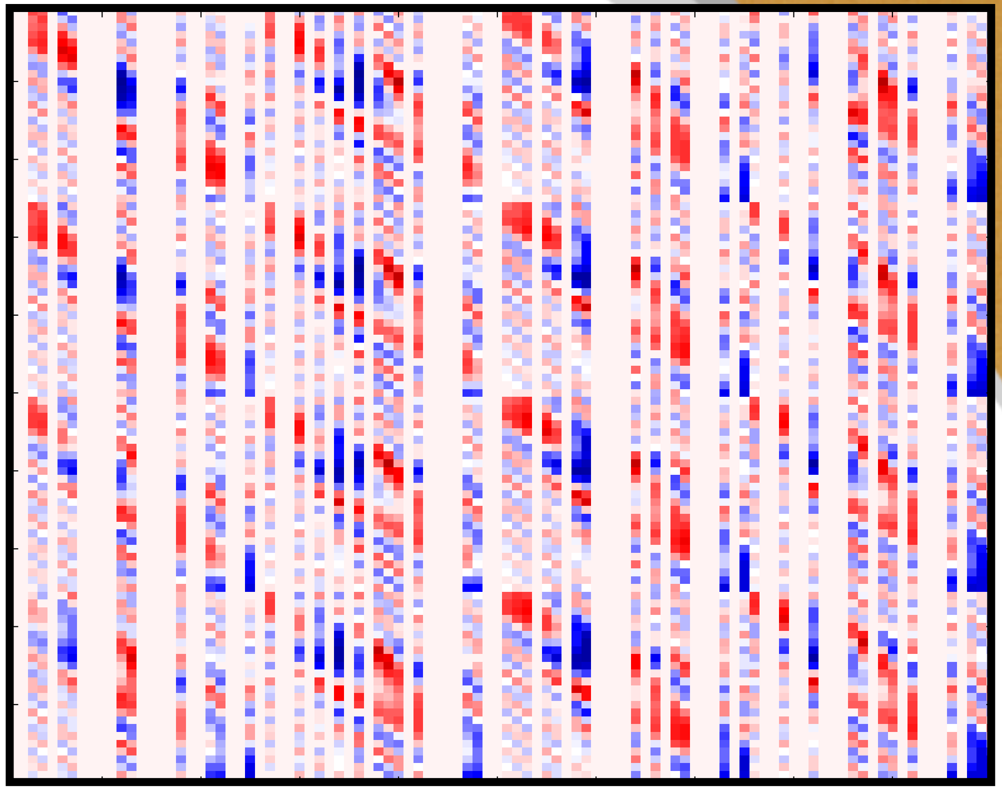
SNR 19.3 dB

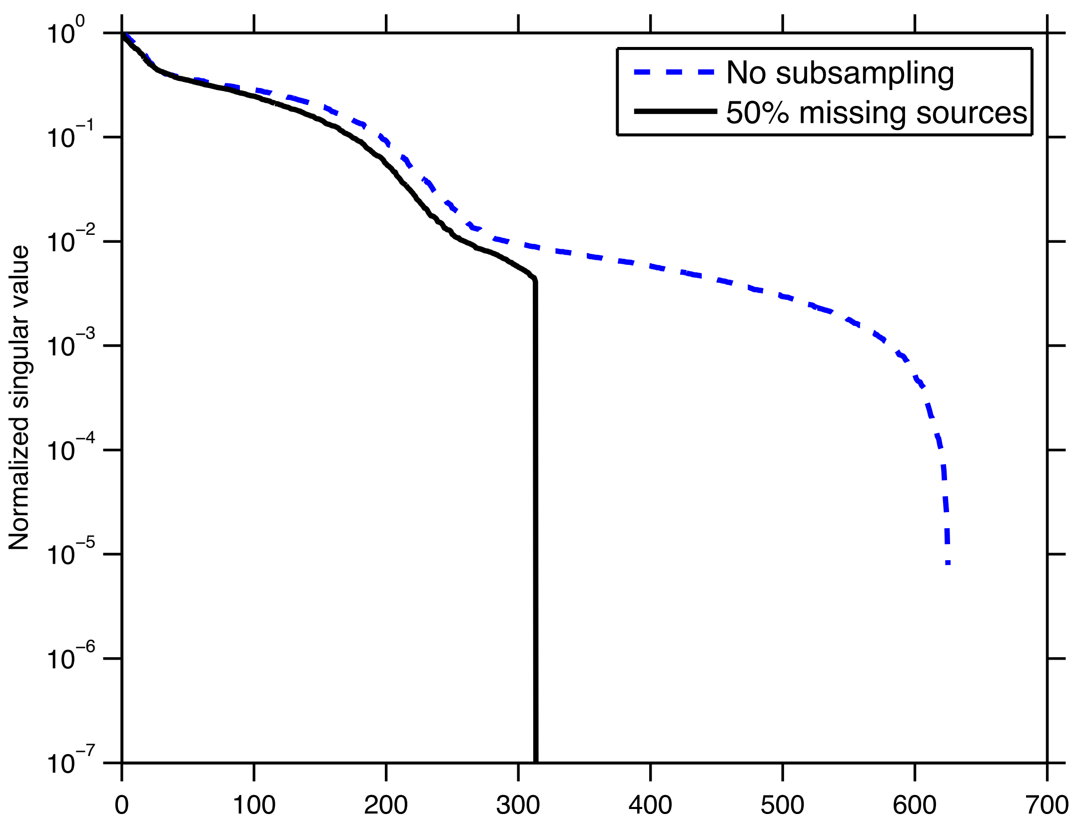
Sampling

- Sampling (src x, src y, rec x, rec y) points
  - Not physical
- How does our data behave under randomly missing sources or receivers?
  - Ocean-bottom node setup

Sampling



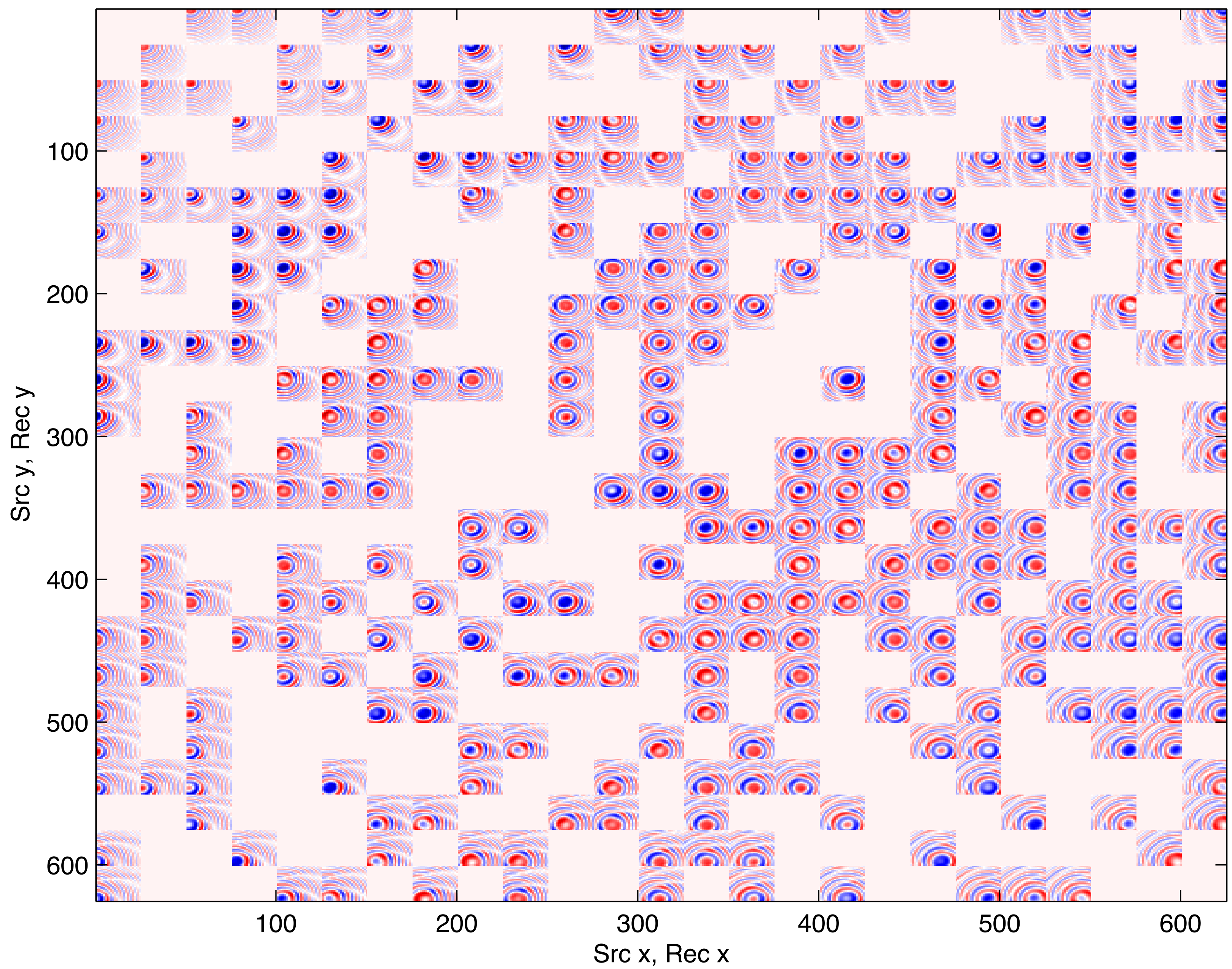


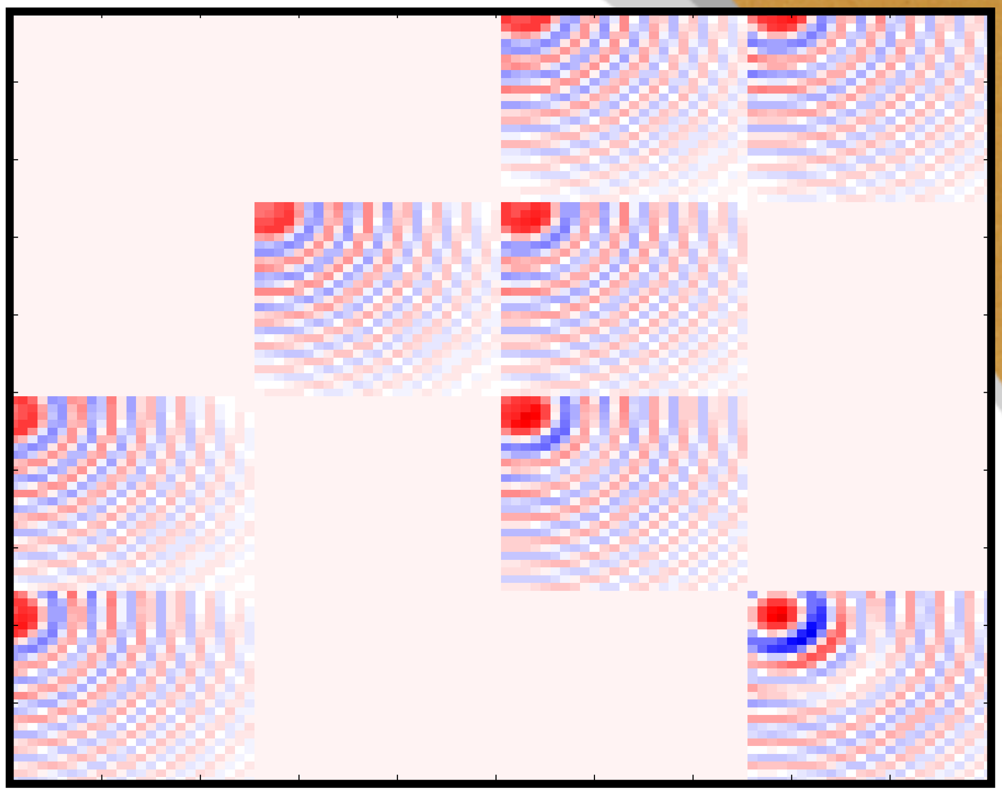


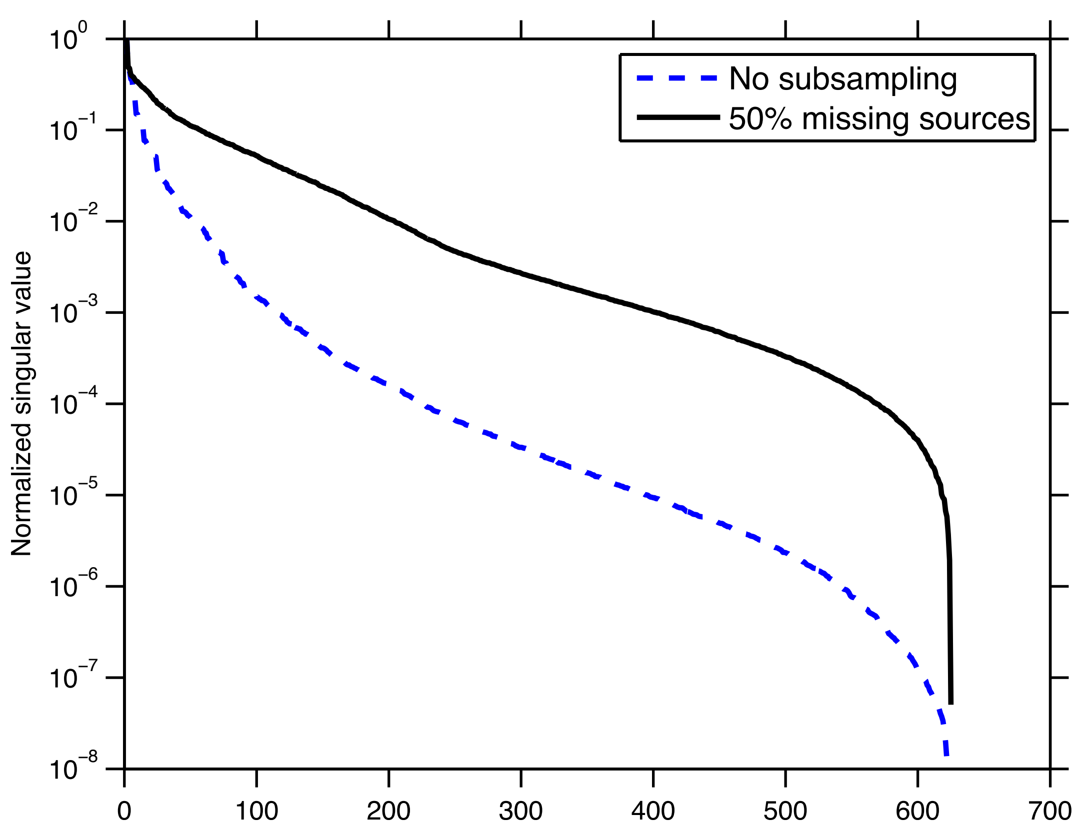
(Rec x, Rec y) matricization

50% randomly missing sources

Sampling







(Src y, Rec y) matricization

50% randomly missing sources

Data organization

- (Rec x, rec y) organization
  - High rank
  - Missing sources operator - removes columns
  - Poor recovery scenario

Data organization

- (Src y, rec y) decomposition
  - Low rank
  - Missing sources operator - removes blocks
  - Closer to ideal recovery scenario

Multidimensional interpolation

Successful reconstruction scheme

- Signal structure - *Hierarchical Tucker*
- Sampling - *subsampling increases h-rank*
- **Optimization - *fit data in the Hierarchical Tucker format***

Optimization Format

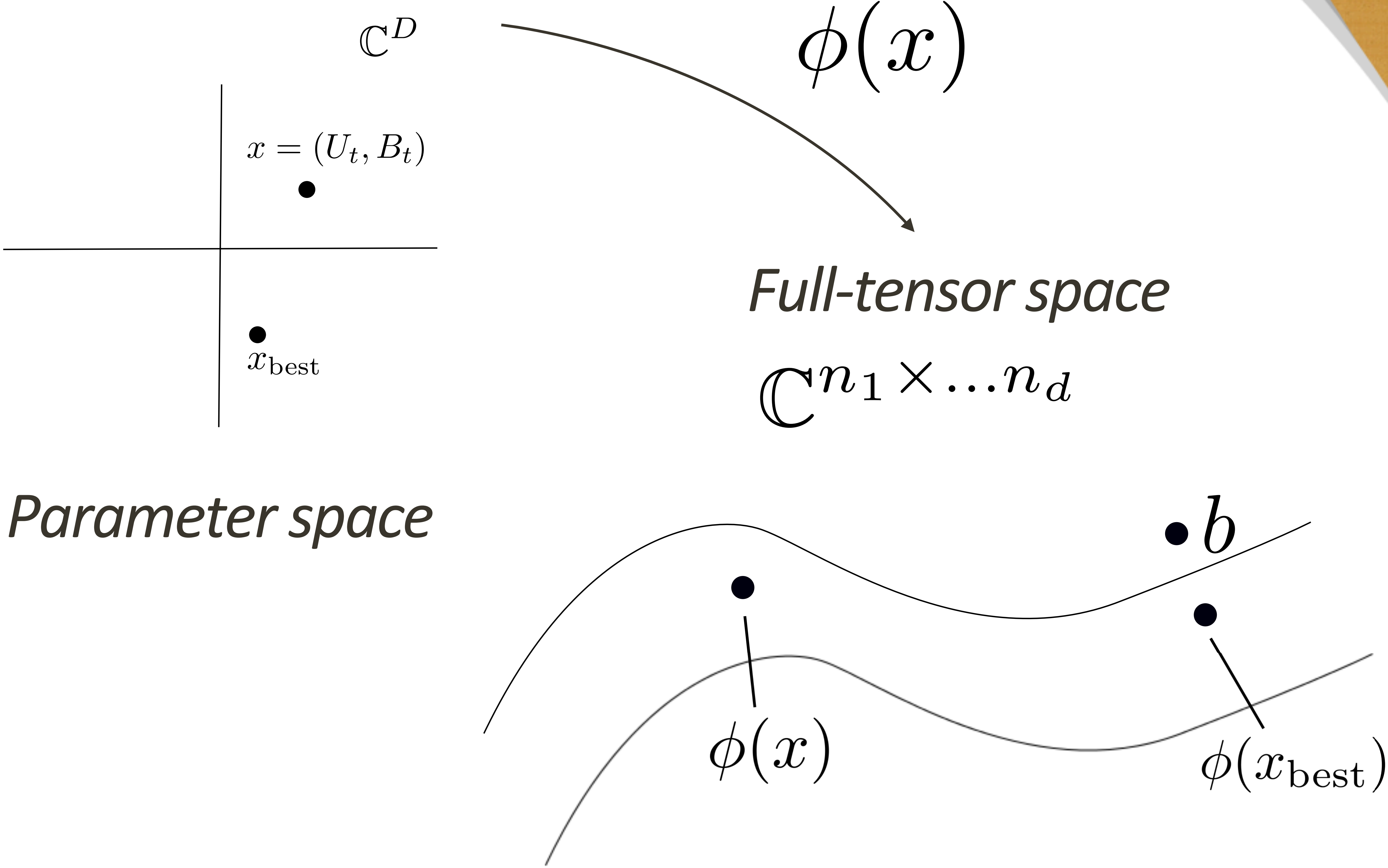
- Given data b with missing sources and/or receivers, subsampling operator A, full tensor expansion operator
$$\phi : (U_t, B_t) \to \mathbb{C}^{n_1 \times \cdots \times n_d}$$
solve
$$\min_{x=(U_t, B_t)} \|A\phi(x) - b\|_2^2$$

P.A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton Univ Press, 2008.
A. Uschmajew, B. Vandereycken. *The geometry of algorithms using hierarchical tensors*. 2012

Differential Geometry

- *The geometry of algorithms using hierarchical tensors* - theoretical analysis of HT tensors
  - Nonlinear, nonconvex space - Riemannian manifold

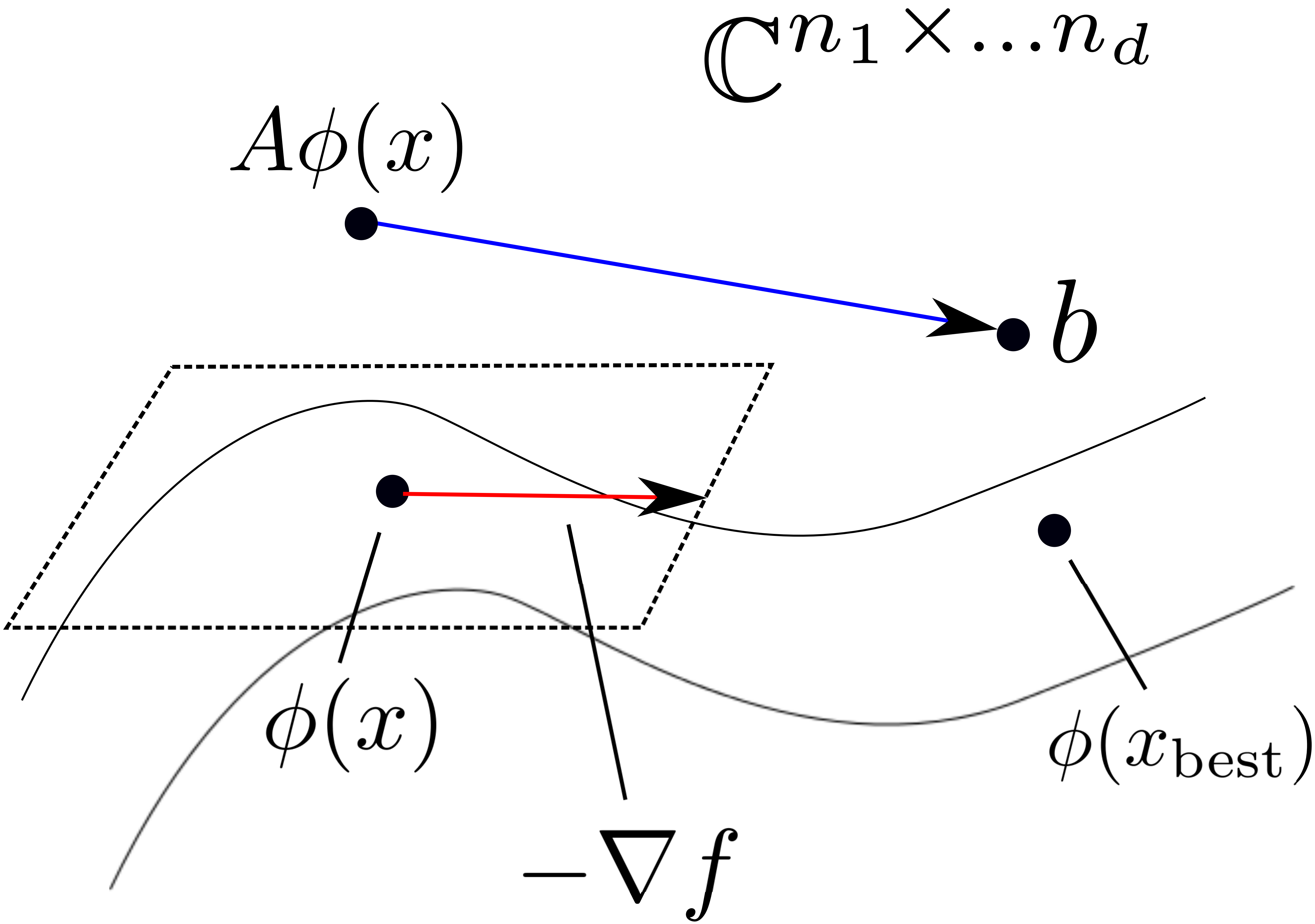
Optimization program



Our contributions

- Theoretical -> Practical
- Steepest Descent, Conjugate Gradient, Gauss-Newton
- Efficient optimization framework that *avoids* computing SVDs

Optimization program



Derivatives

- Derivatives of a particular node with respect to its children can be computed efficiently
- The chain rule gives the gradient of the function ϕ

Derivative code

```
function [dUdL,dUdR,dUdB] = dHTuck(Uleft, Uright, B)
    Bhat = matricize(B,1,[2 3]);
    Bhat = dematricize(Uright * Bhat,[nr,kl,k],1,[2,3]);
    Bhat = matricize(Bhat,[1 3],2);
    dUdL = opFunction(nl*nr*k, nl*kl, @(x,mode) dUdL_func(x,mode,
Bhat,nr,nl,k,kl));
    BhatR = dematricize( Uleft*matricize(permute(B,[2 1 3]),1, [2 3]),
[nl,kr,k],1,[2 3]);
    dUdR = opKron(matricize(BhatR,[1 3],2),opDirac(nr));
    dUdB = opKron(opDirac(k),Uleft,Uright);
end
function y = dUdL_func(x,mode,Bhat,nr,nl,k,kl)
    if mode == 1
        v = reshape(x,nl,kl);
        q = Bhat * v';
        result = reshape(q,nr,k,nl);
        result = permute(result,[1 3 2]);
        y = vec(result);
    else
        v = reshape(x,nr,nl,k);
        q = Bhat' * matricize(v,[1 3],2);
        q = reshape(q,kl,nl);
        y = vec(q');
    end
end
```

Derivatives

- Only involve matrix-matrix multiplication of small matrices compared to the ambient, full-tensor space
- Immediately parallelizable - Kronecker products done in parallel

Gauss-Newton Hessian

$$\min_{x} \frac{1}{2}\|A\phi(x) - b\|_2^2$$

Linearize

$$\min_{dx} \frac{1}{2}\|AD\phi(x_k)dx - b\|_2^2$$

Gauss-Newton Hessian

Solution

$$D\phi(x)^* A D\phi(x)\delta x = -D\phi(x)^* r(x)$$

Ill-conditioned

Instead solve

$$D\phi(x)^* D\phi(x)\delta x = -D\phi(x)^* r(x)$$

Gauss-Newton

$$D\phi(x) : \text{params} \to \mathbb{C}^{n_1 \times \dots n_d}$$

$$D\phi(x)^* : \mathbb{C}^{n_1 \times \dots n_d} \to \text{params}$$

$$H_{GN} = D\phi(x)^* D\phi(x) : \text{params} \to \text{params}$$

Gauss-Newton Hessian

- Simplify expressions
- Avoid forming intermediate tensors of size $O(\mathbb{C}^{n_1 \times \dots n_d})$
- GN Hessian is block-diagonal - easy to invert

Results

Synthetic BG Data

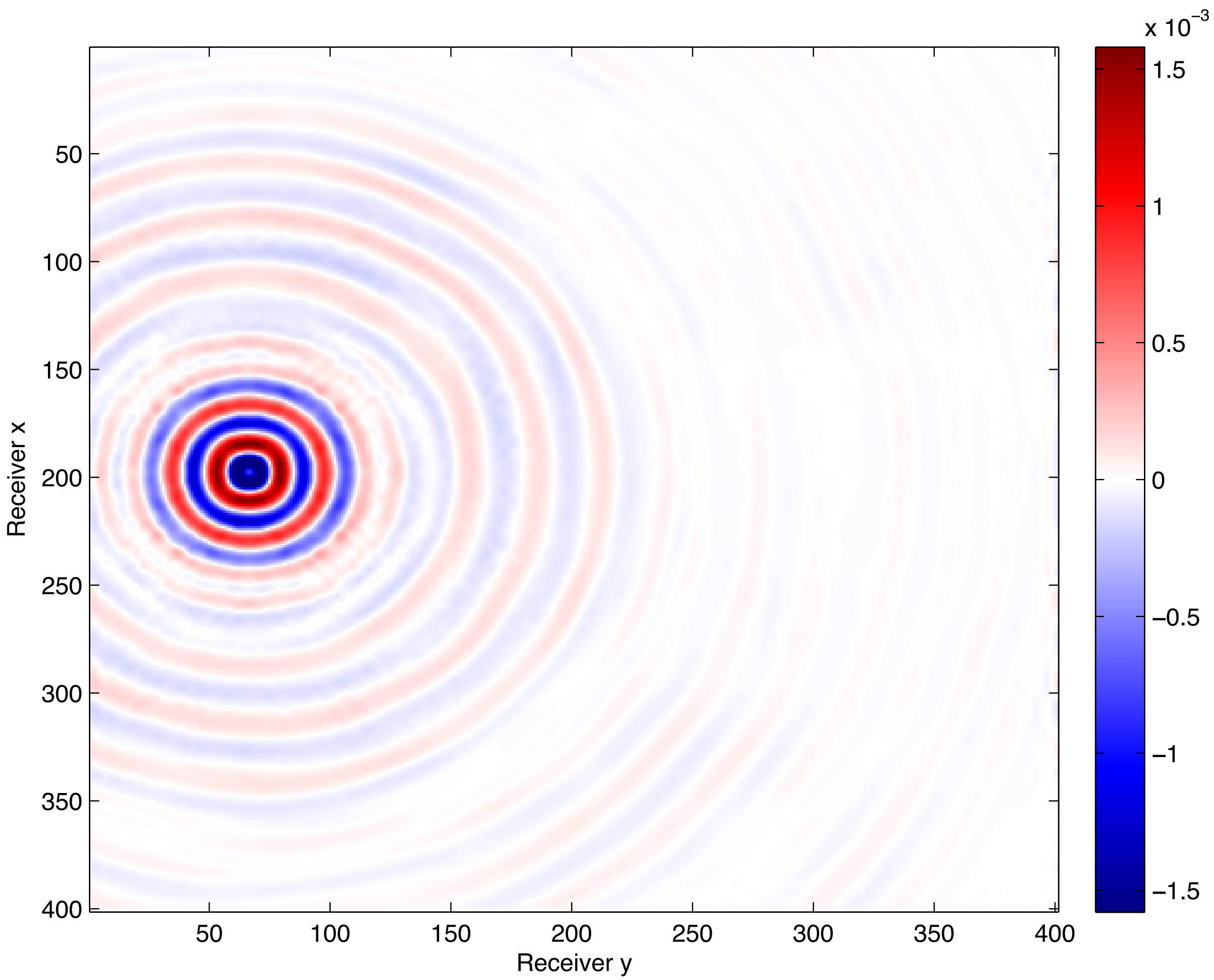
- Unknown model
- 68 x 68 sources with 401 x 401 receivers, data at 4.68Hz, 7.34 Hz, 12.3 Hz
  - Receivers subsampled, Fourier interpolated back to 401 x 401 to produce figures

Synthetic BG Data

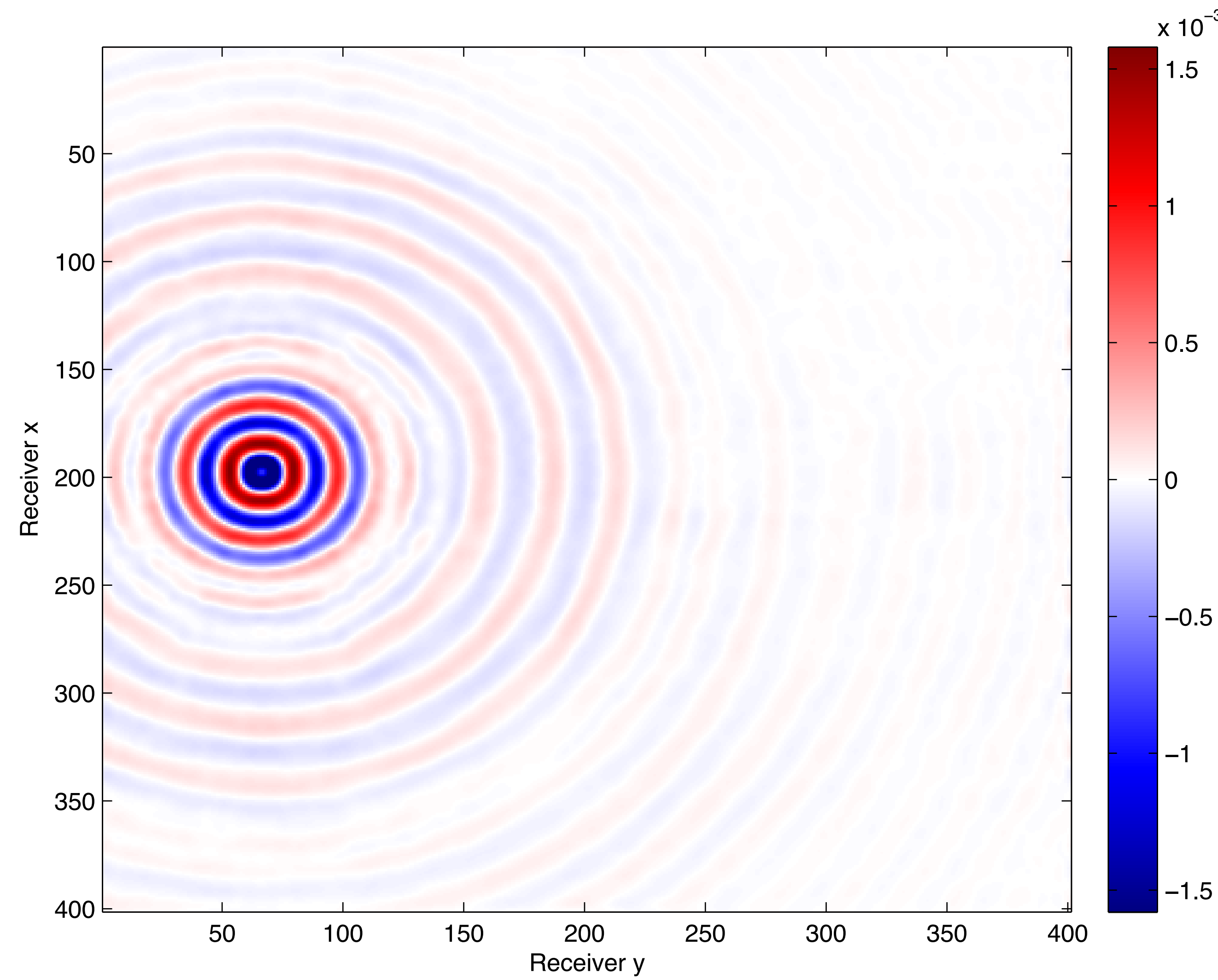
- Single frequency slice (real part) - scaled to unit norm
- Varying percentages of sources have been randomly removed
- Recovered with nonlinear CG

4.86 Hz - 50% missing sources

Common source gather - no data originally



$(x_{src}, y_{src}) = (34,17)$

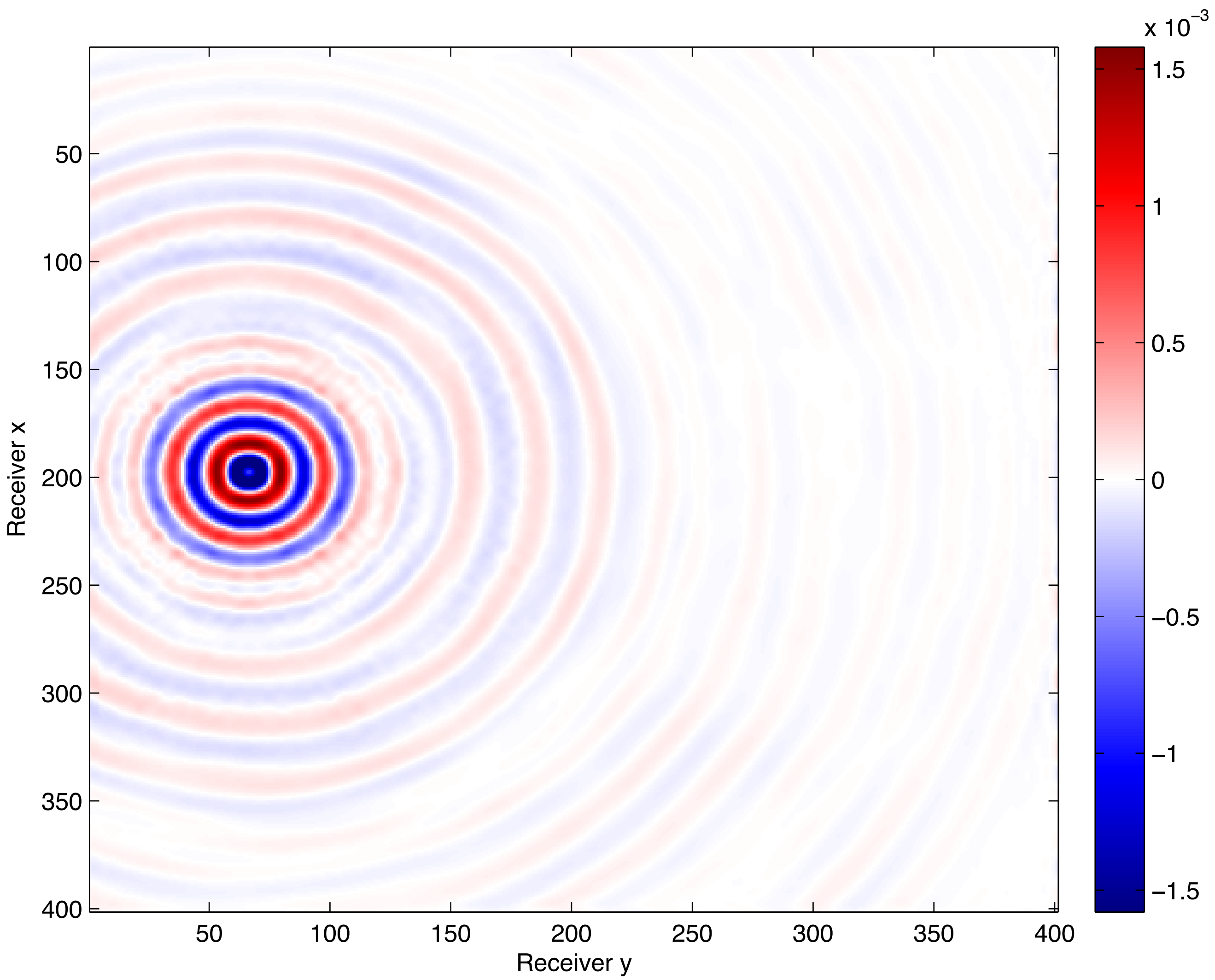


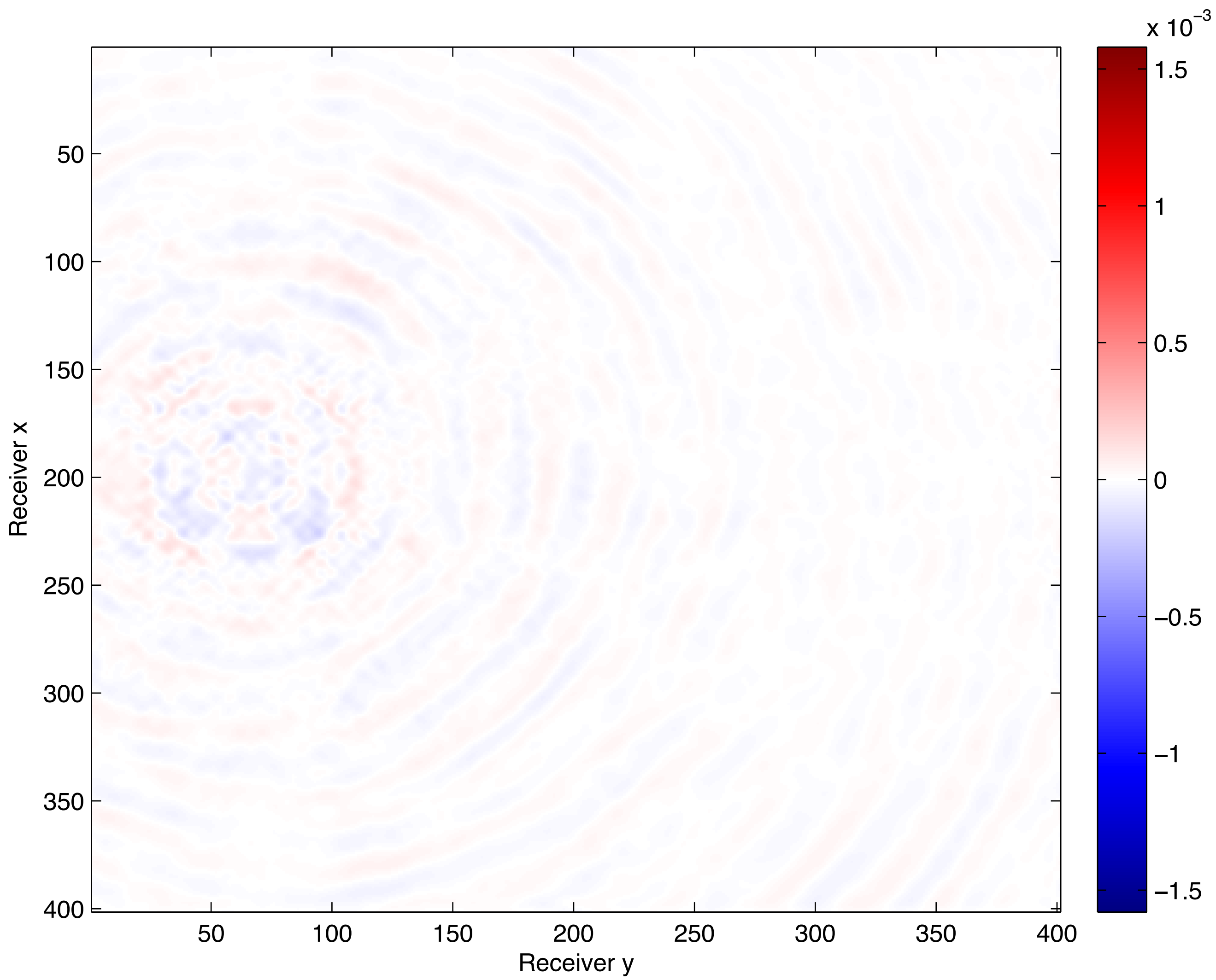
Interpolated Data

SNR 17.5 dB

4.86 Hz - 50% missing sources

Common source gather - no data originally



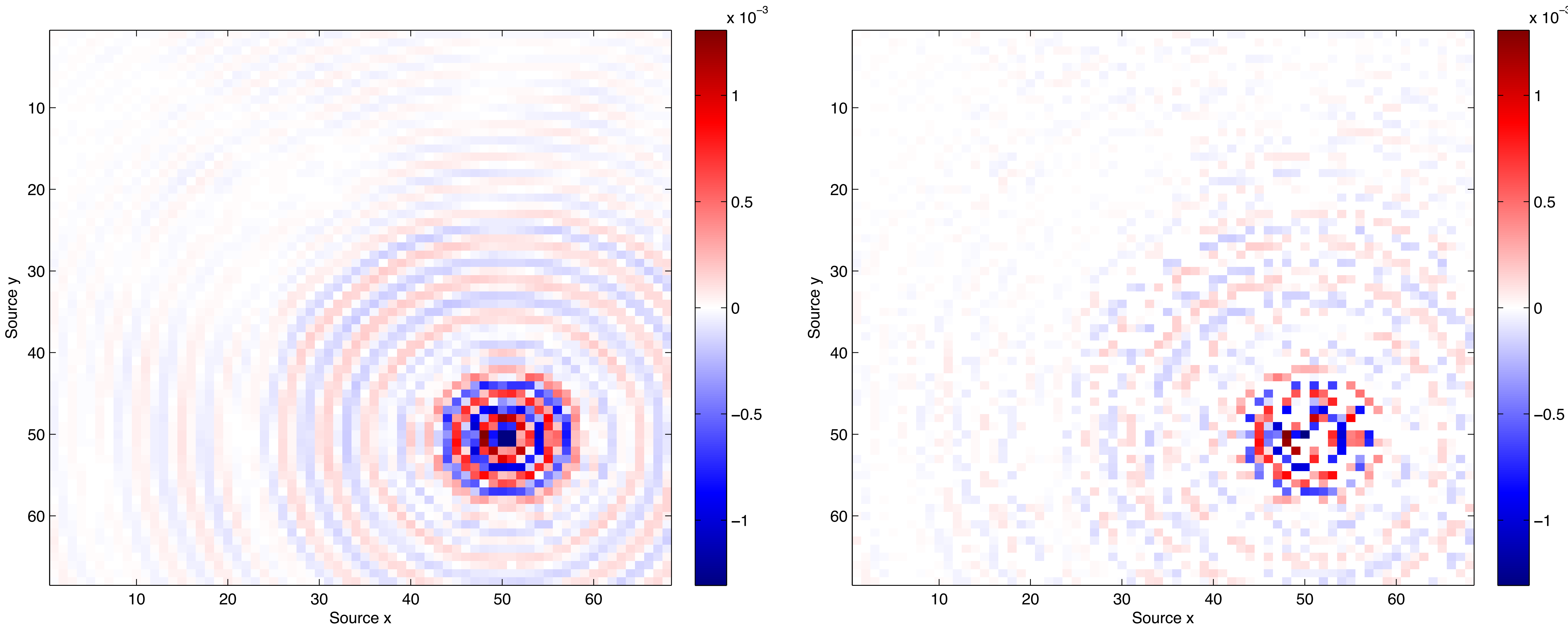


$(x_{src}, y_{src}) = (34,17)$

Difference

4.86 Hz - 50% missing sources

Common receiver gather

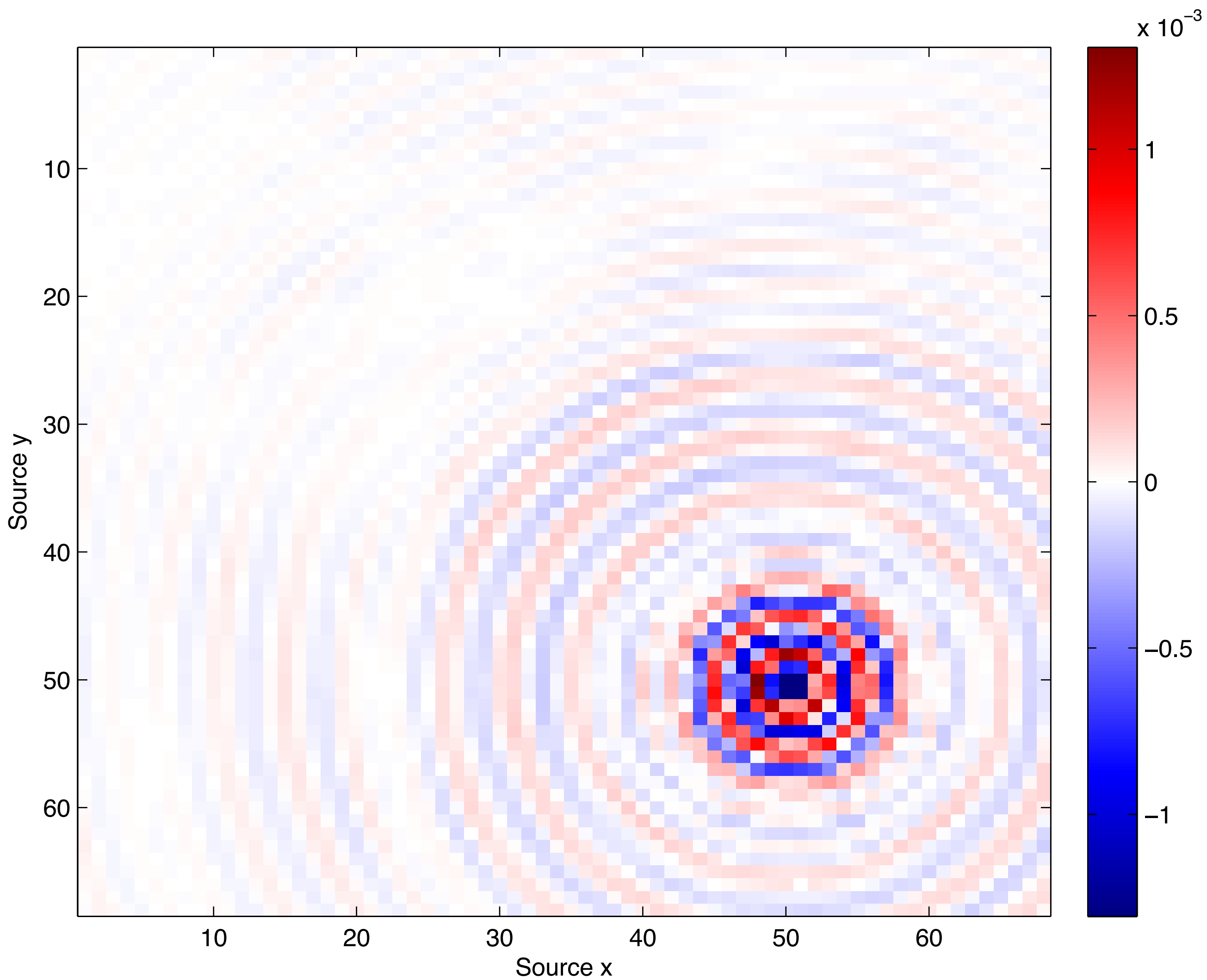


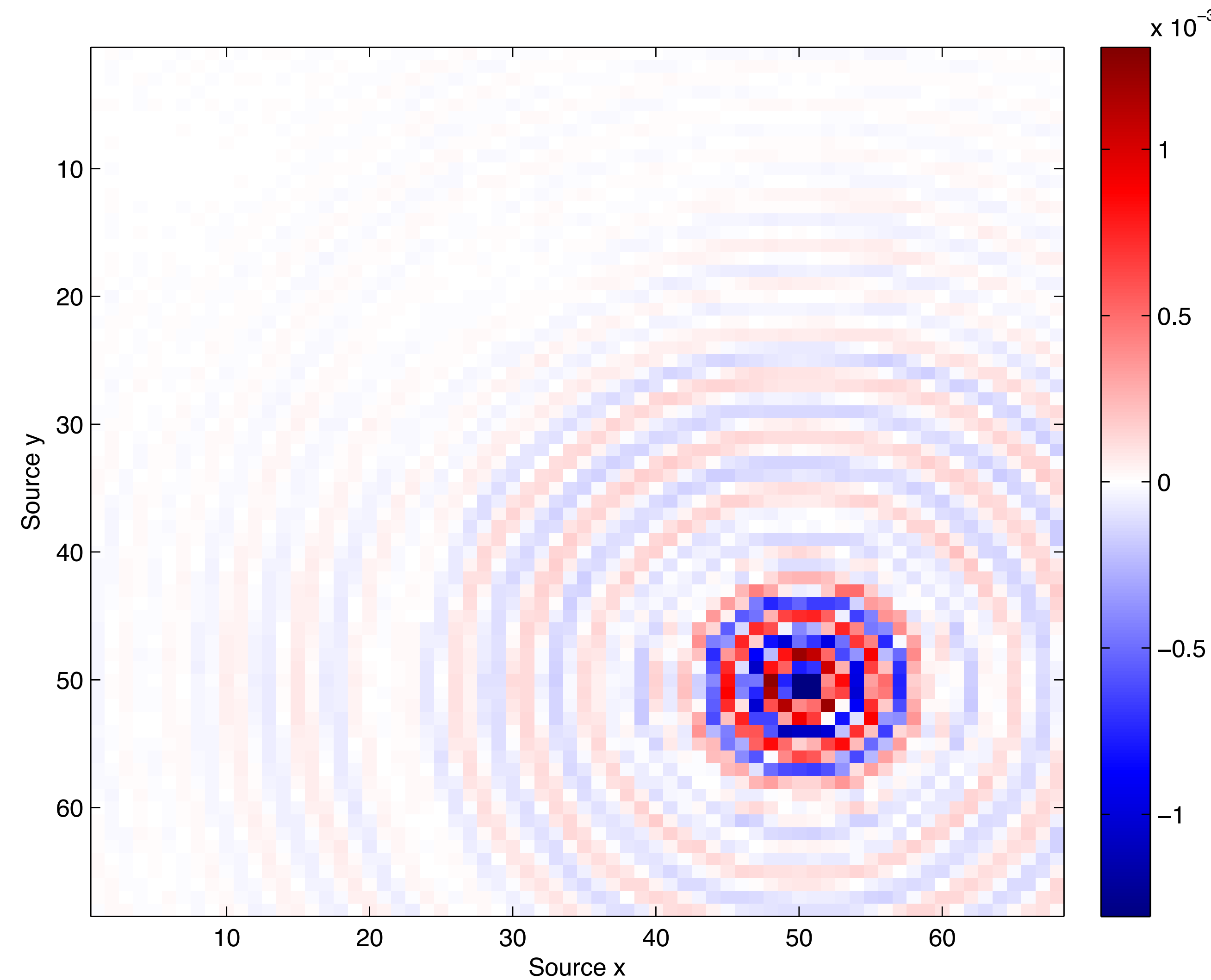
$(x_{rec}, y_{rec}) = (75,75)$

Subsampled Data

4.86 Hz - 50% missing sources

Common receiver gather





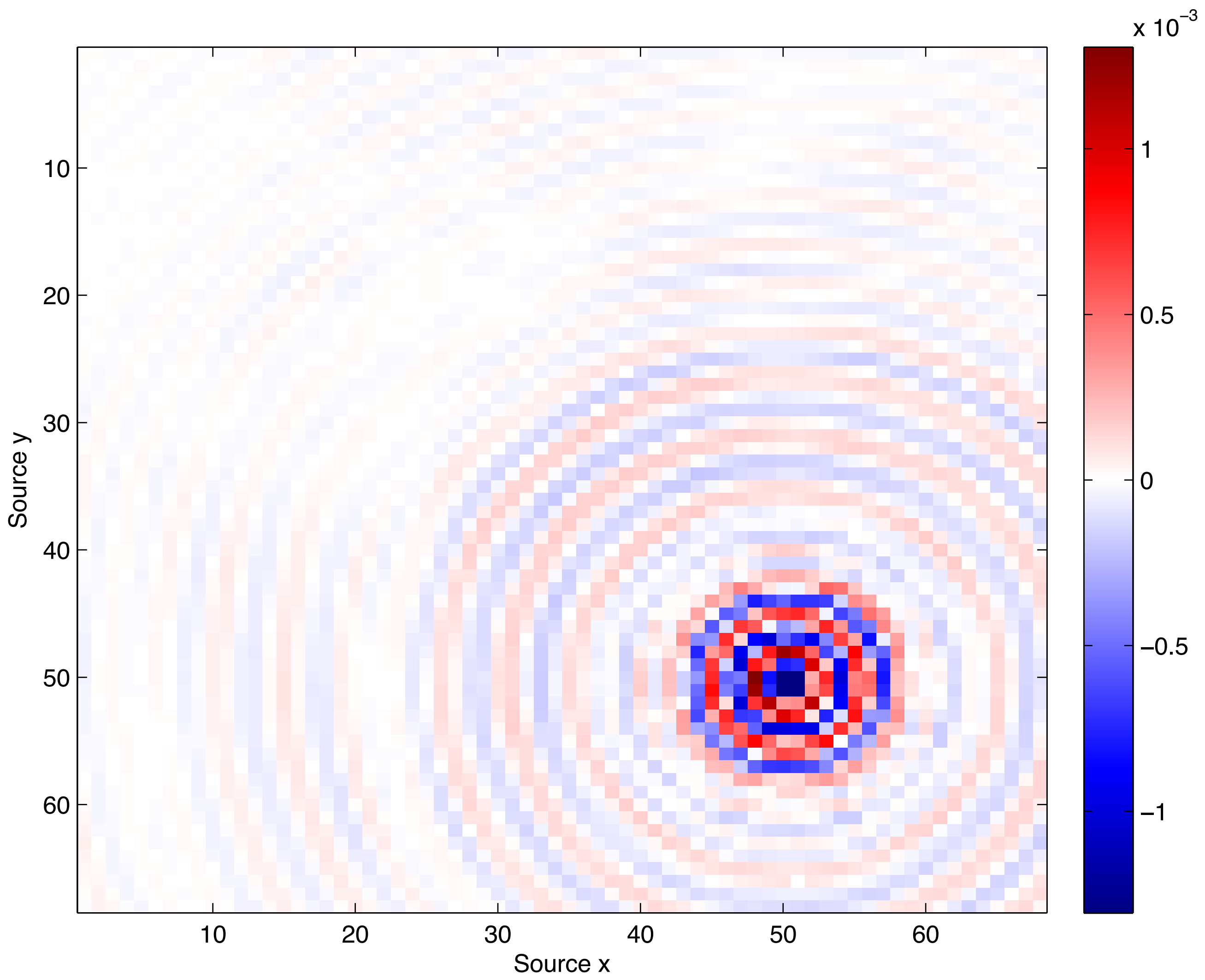
$(x_{rec}, y_{rec}) = (75,75)$

Interpolated Data

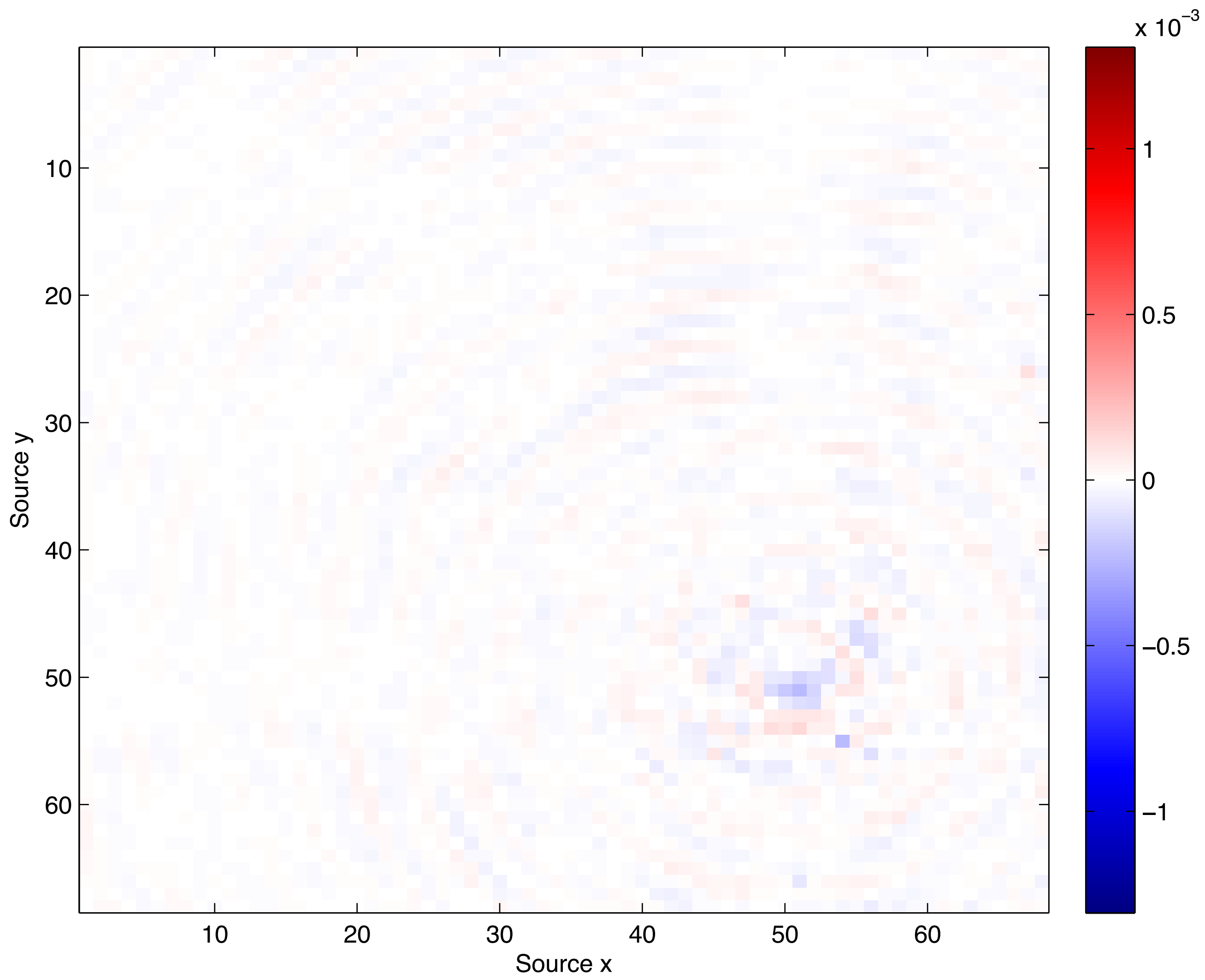
SNR 17.3 dB

4.86 Hz - 50% missing sources

Common receiver gather



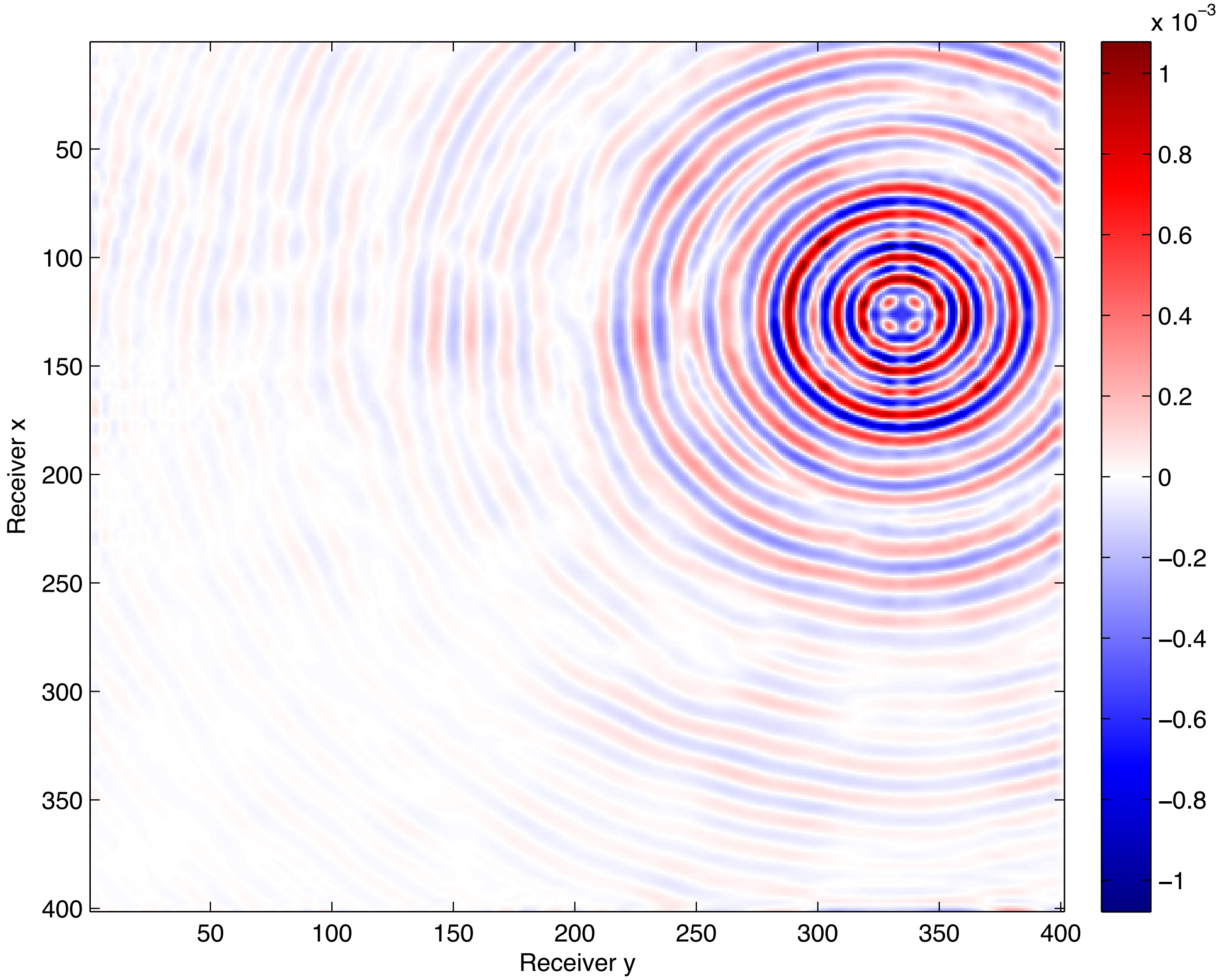
$(x_{rec}, y_{rec}) = (75,75)$

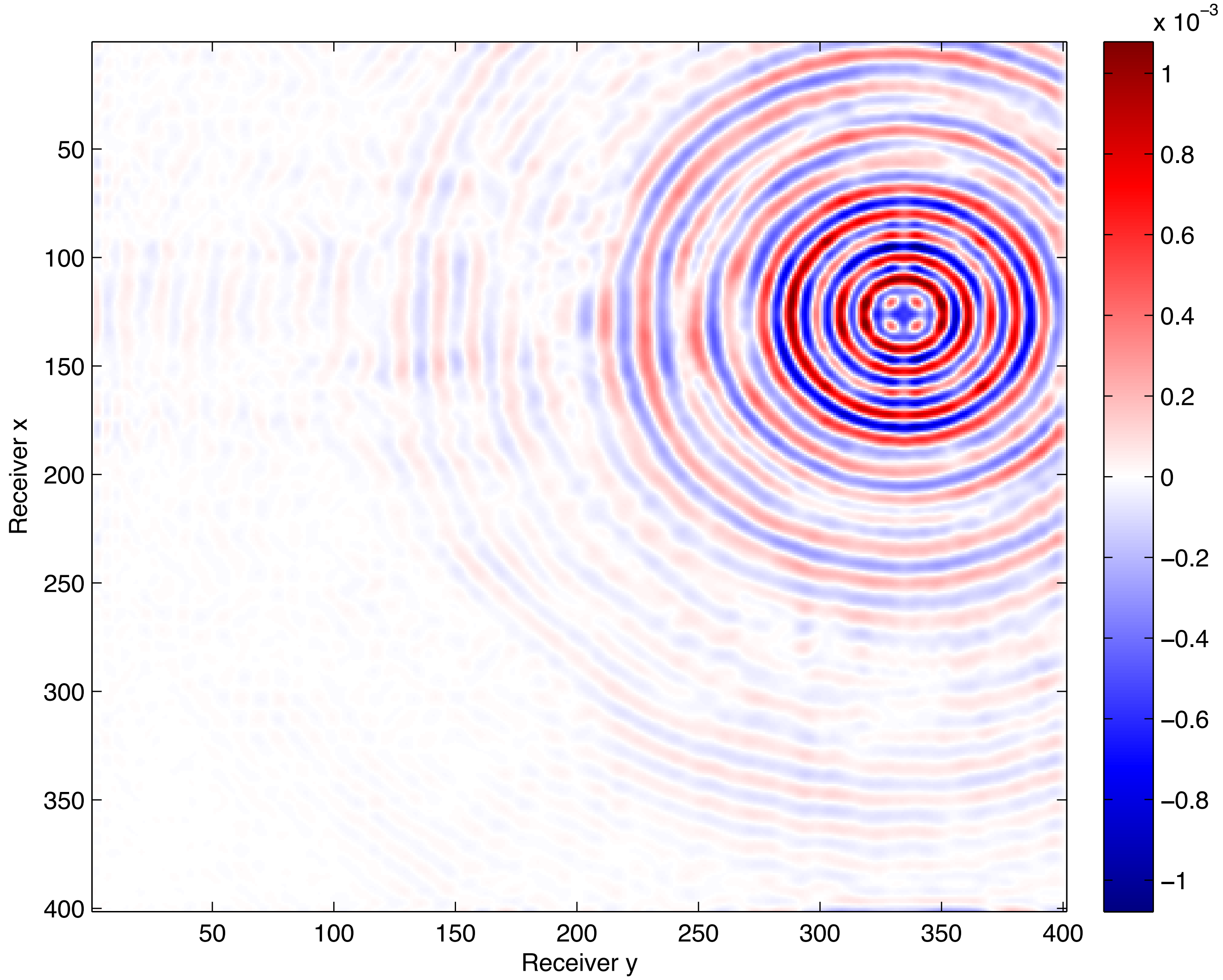


Difference

7.34 Hz - 75% missing sources

Common source gather - no data originally





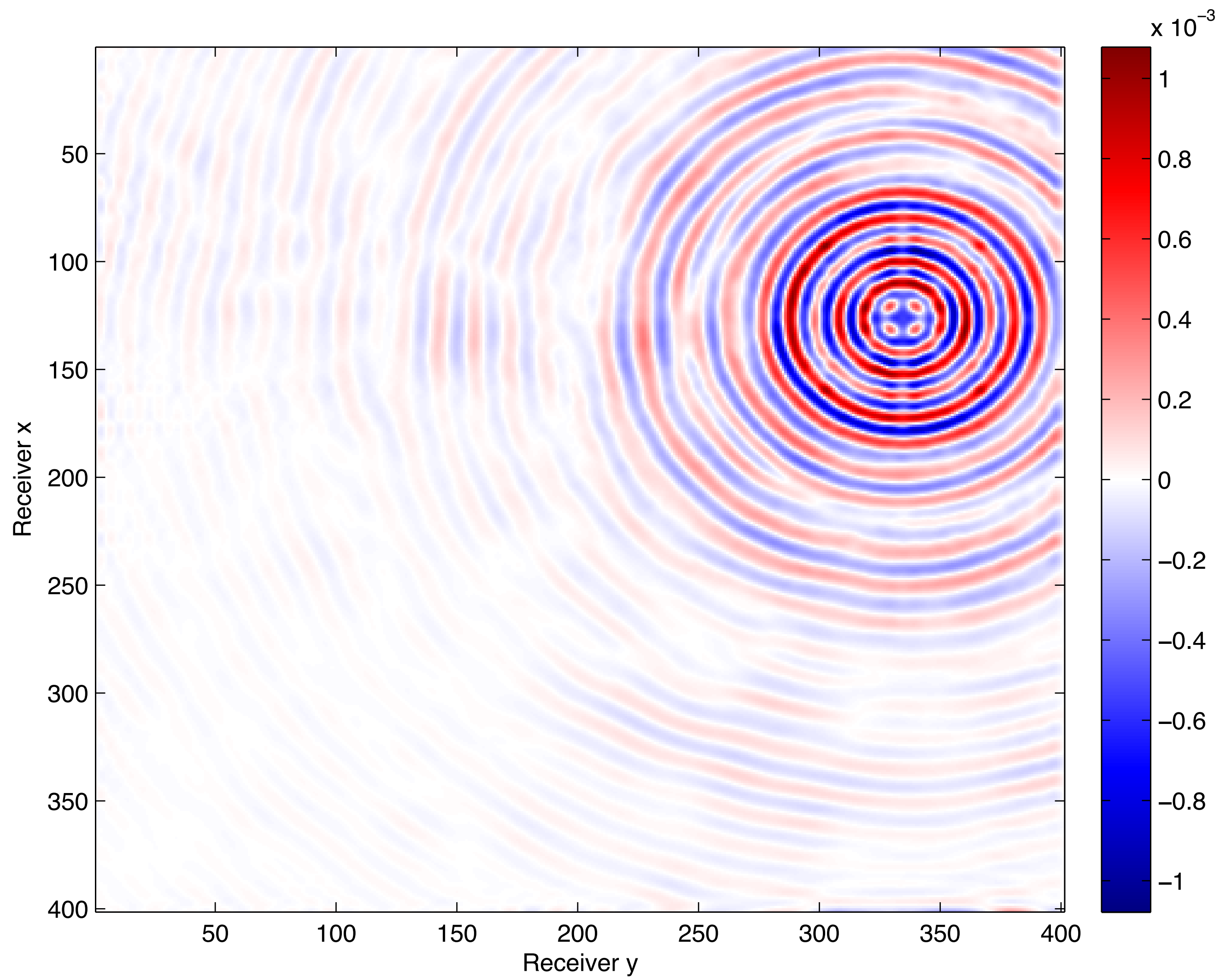
$(x_{src}, y_{src}) = (22,57)$

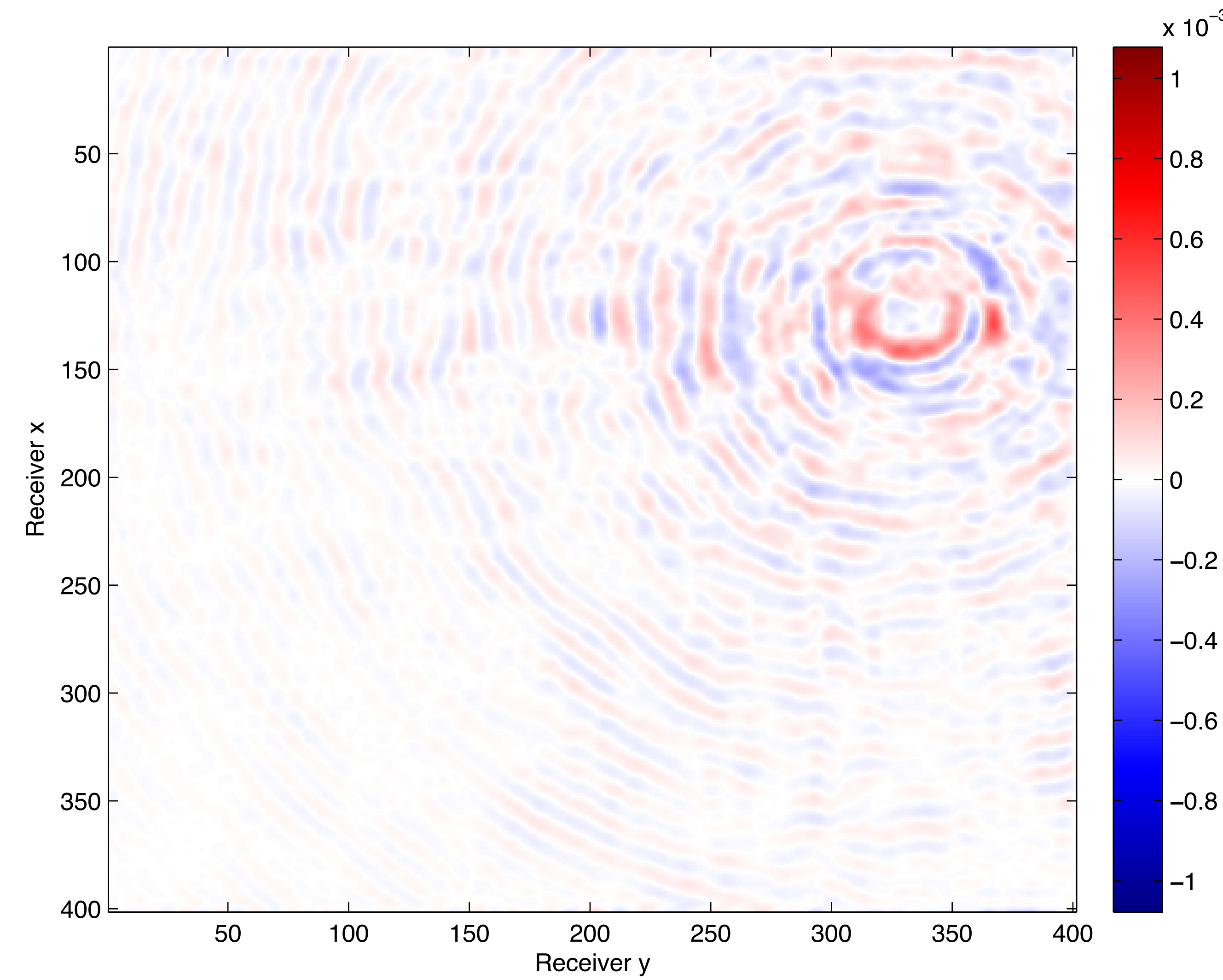
Interpolated Data

SNR 10.6 dB

7.34 Hz - 75% missing sources

Common source gather - no data originally



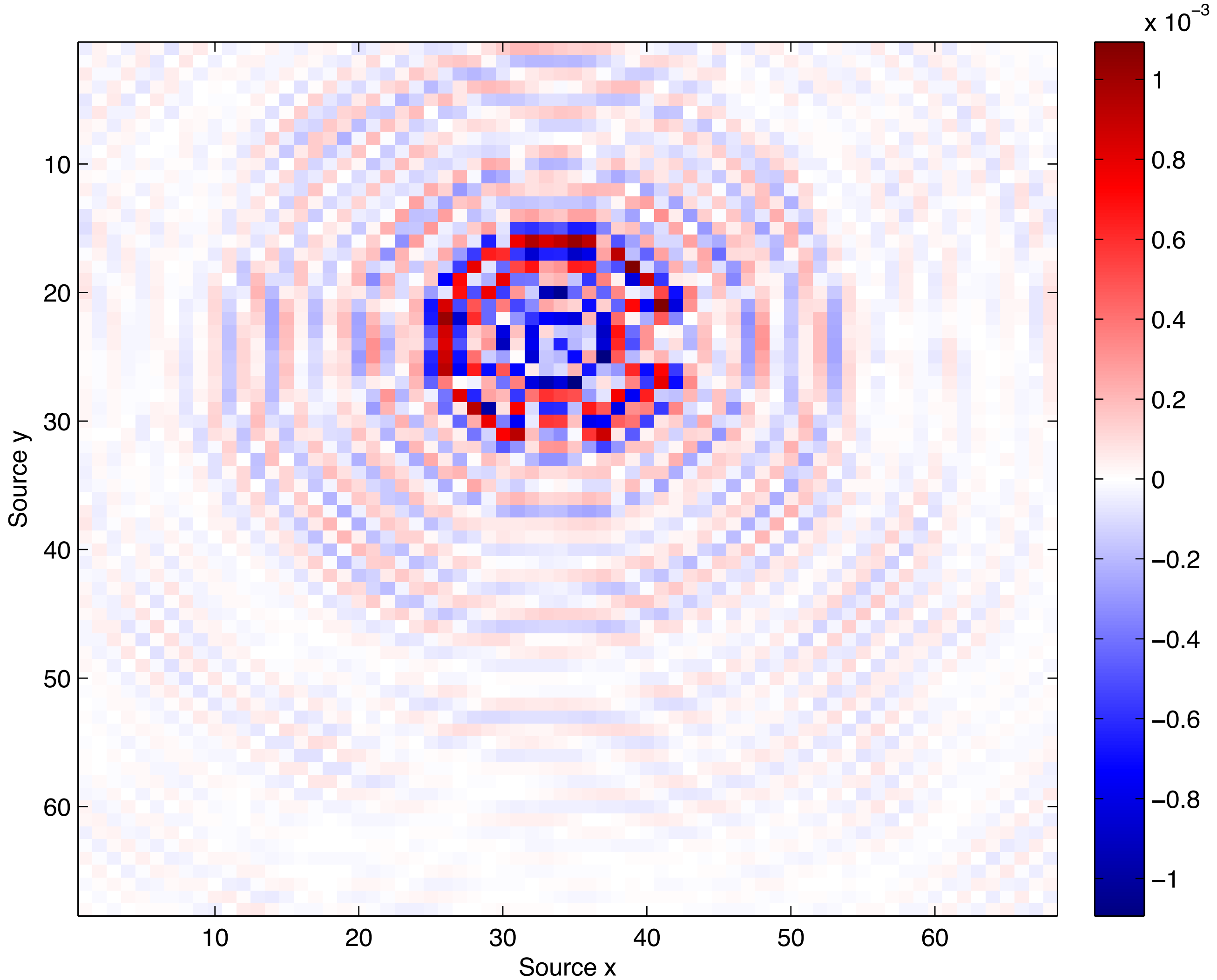


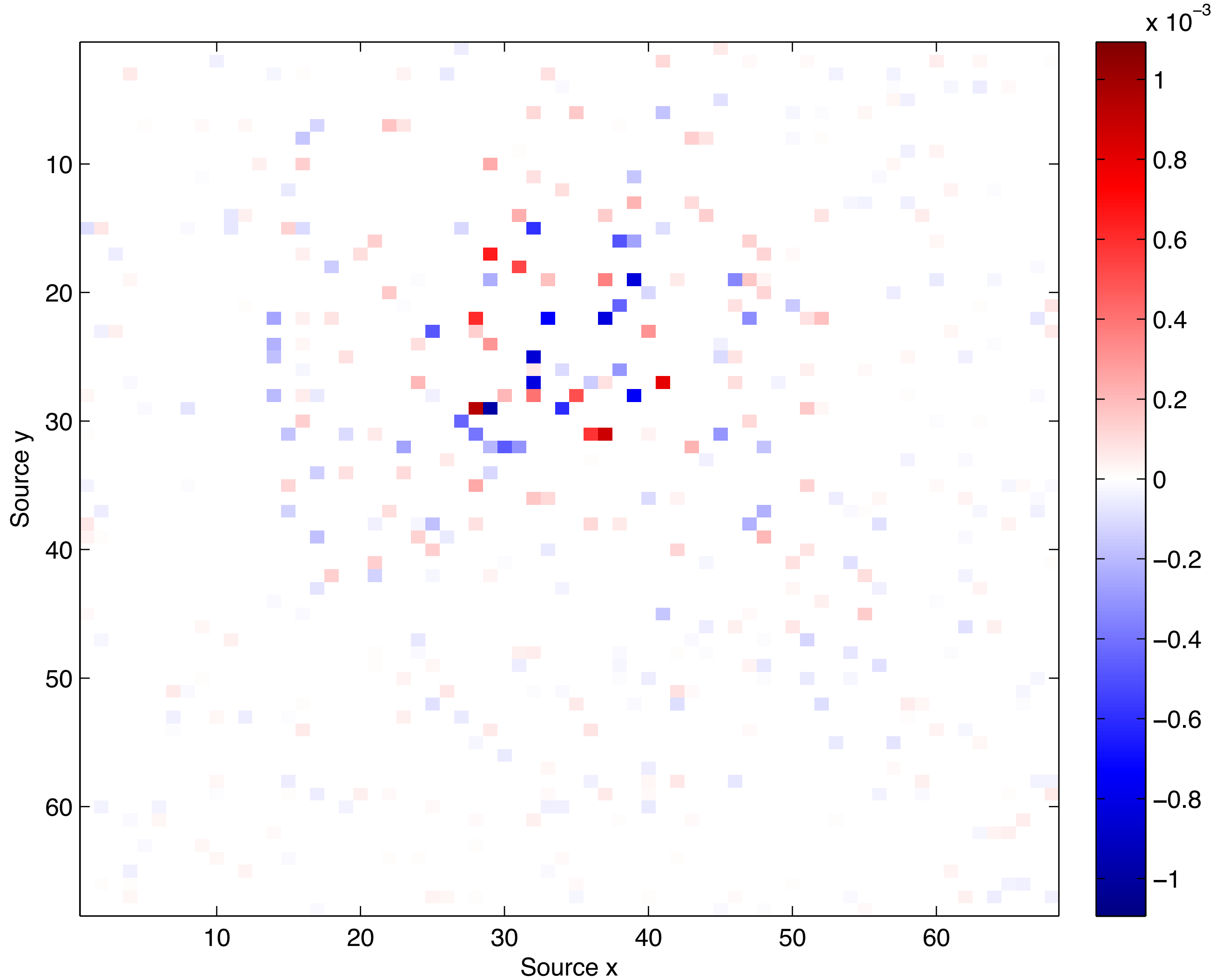
$(x_{src}, y_{src}) = (22,57)$

Difference

7.34 Hz - 75% missing sources

Common receiver gather



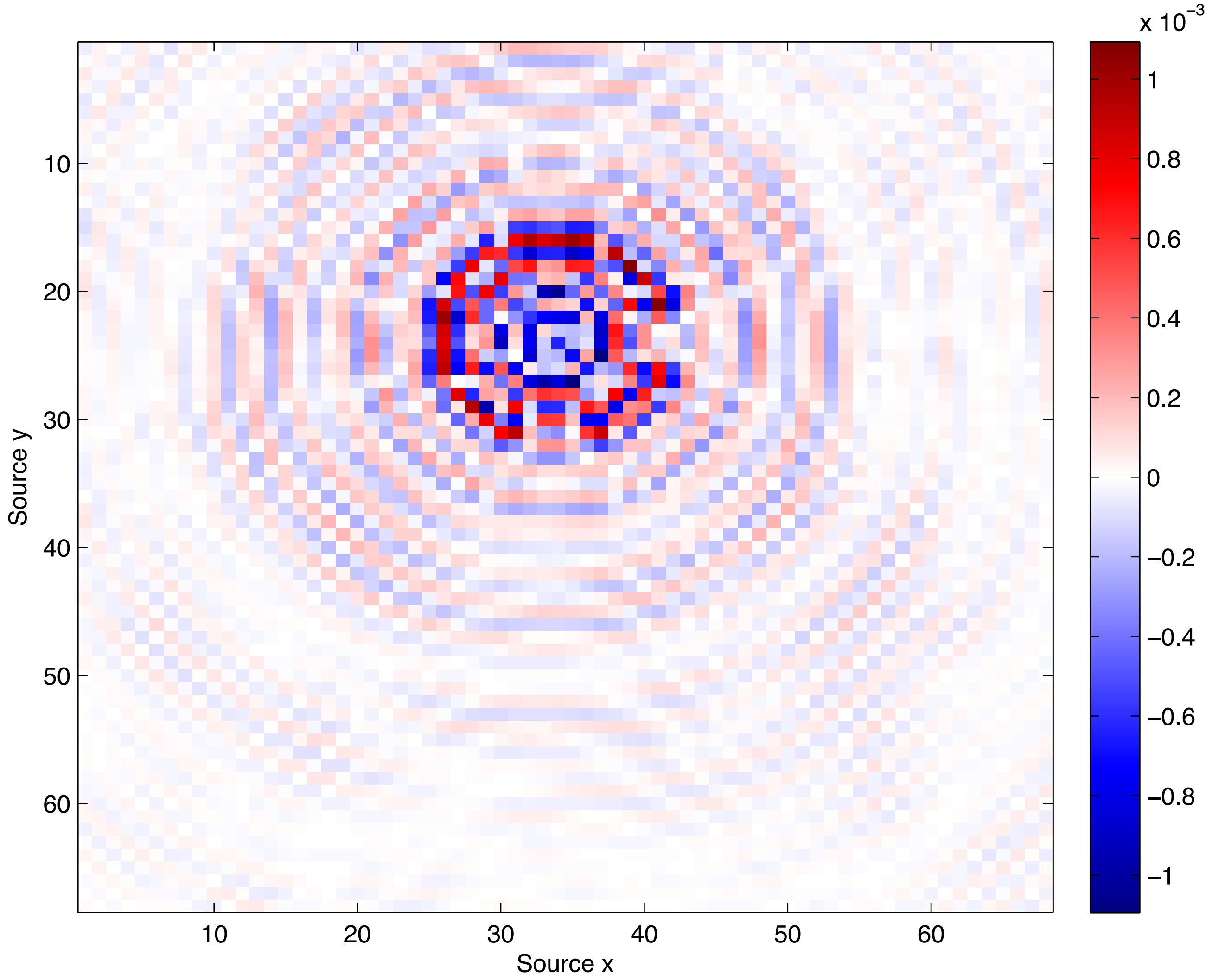


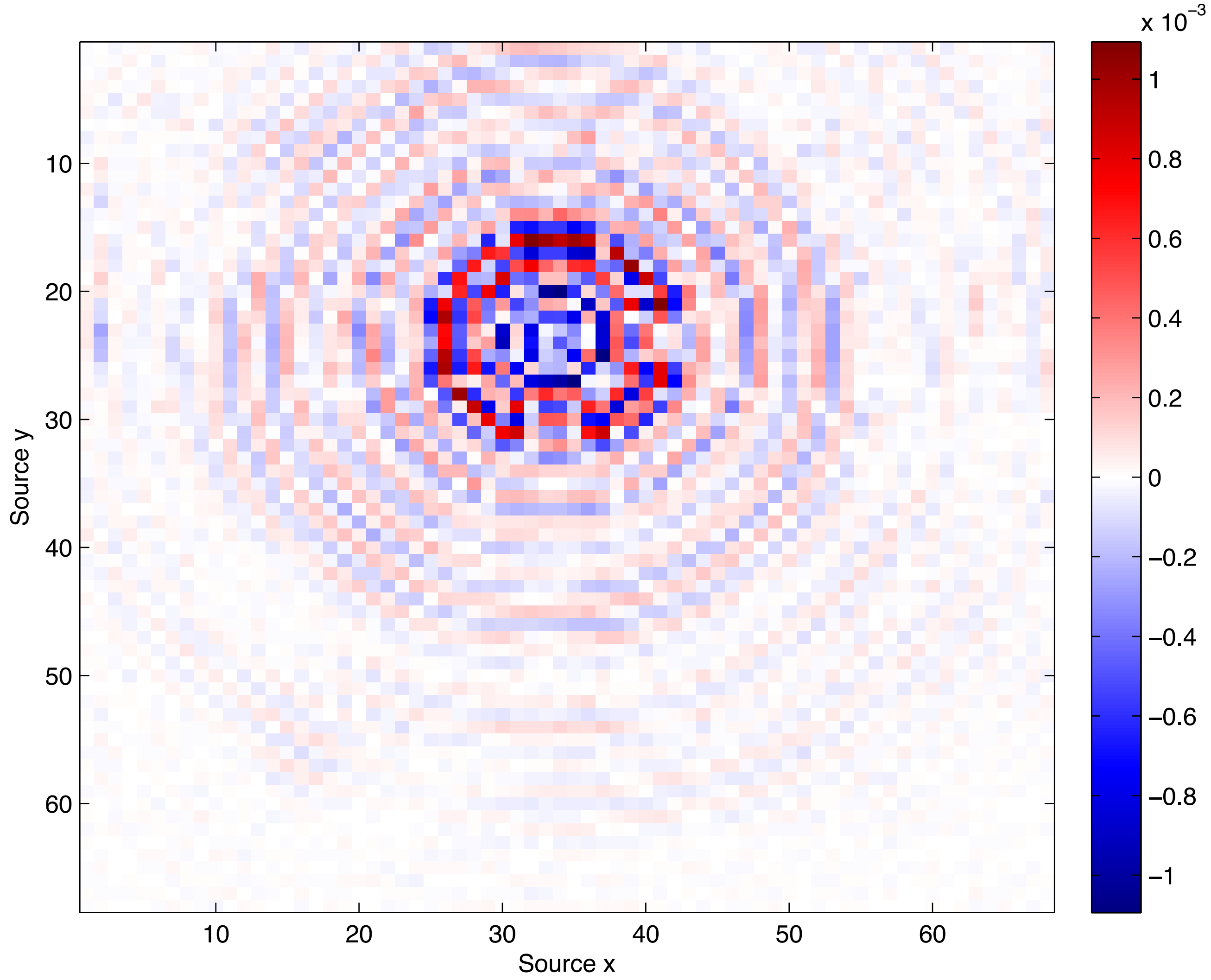
$(x_{rec}, y_{rec}) = (35,50)$

Subsampled Data

7.34 Hz - 75% missing sources

Common receiver gather





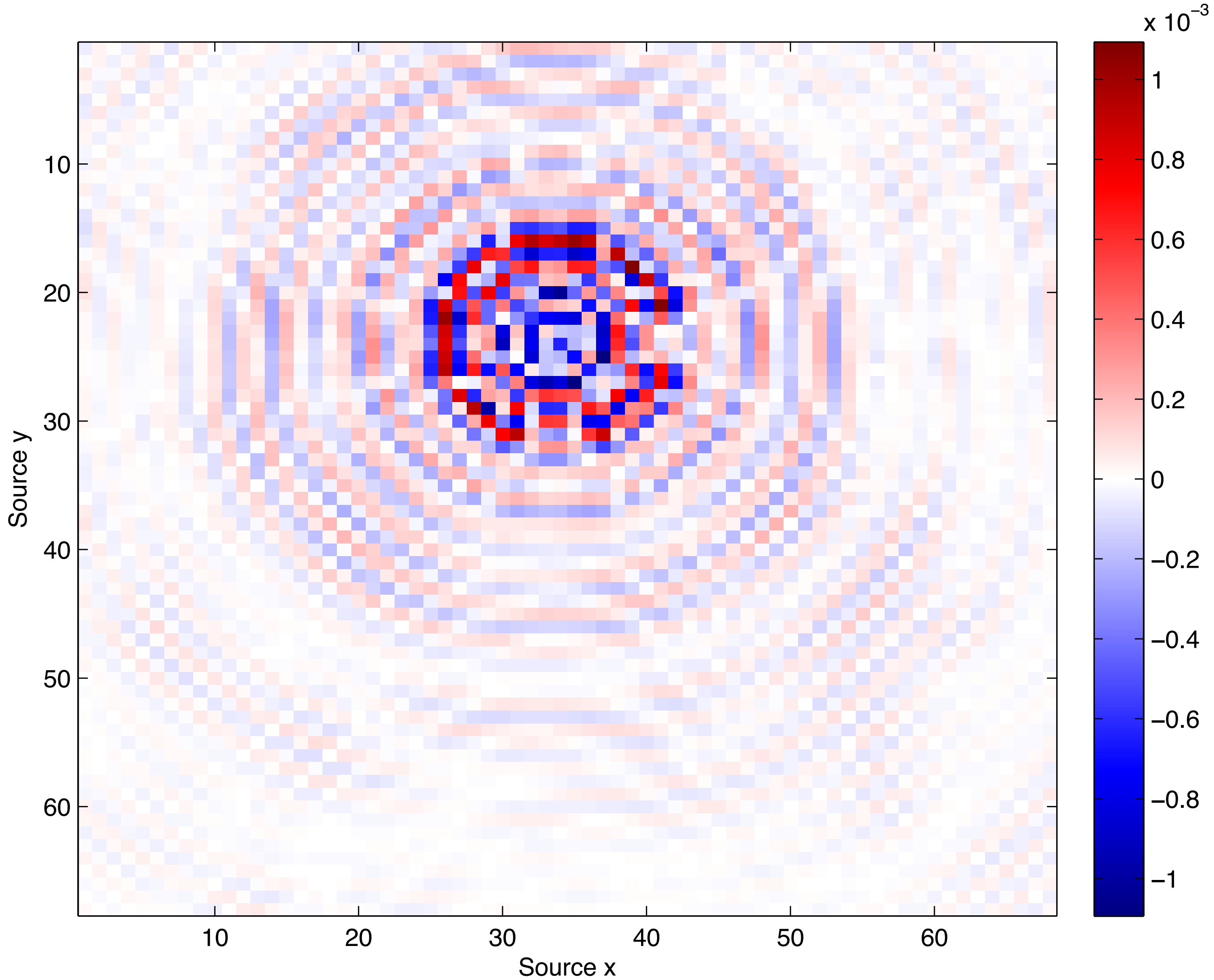
$(x_{rec}, y_{rec}) = (35,50)$

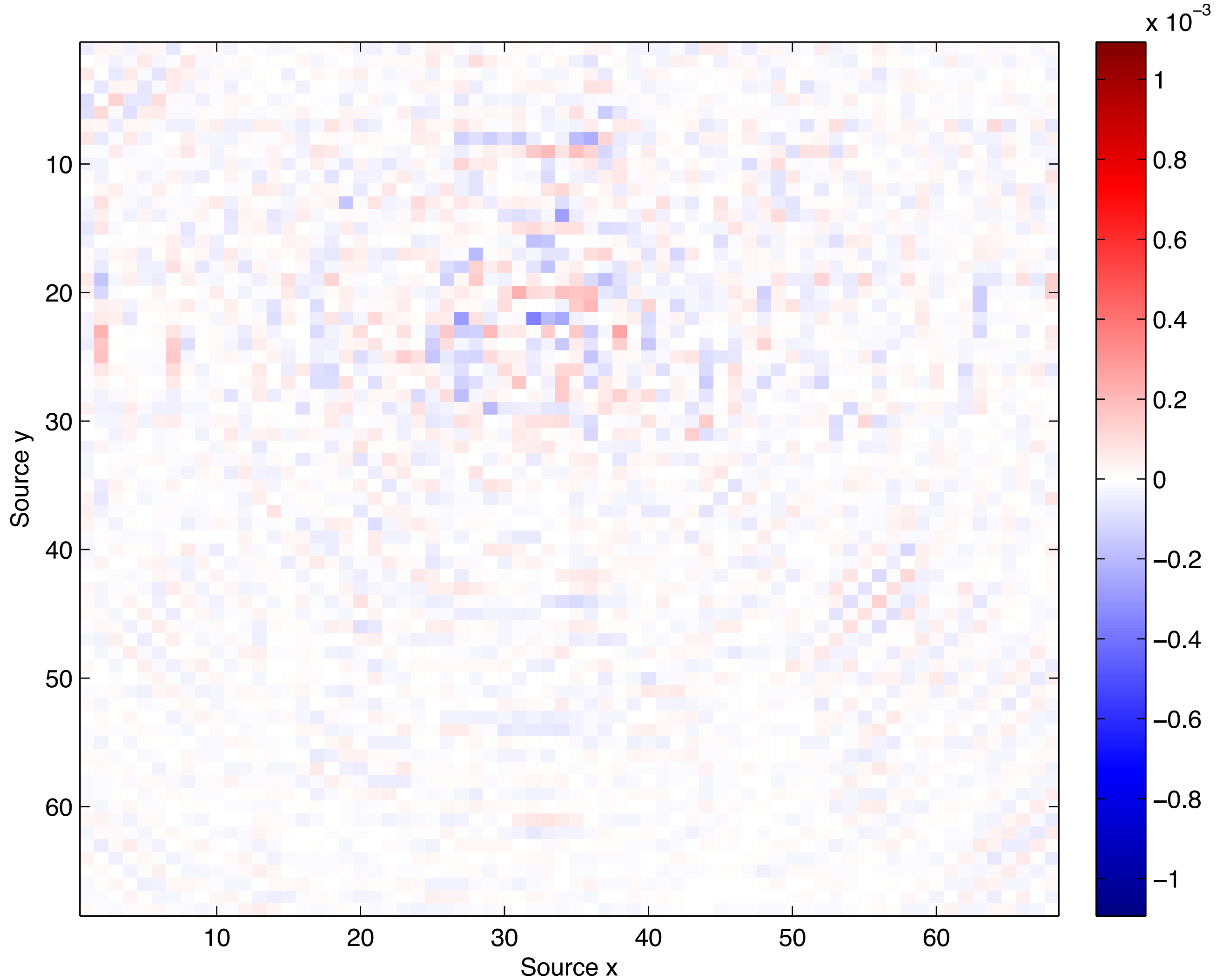
Interpolated Data

SNR 12.1 dB

7.34 Hz - 75% missing sources

Common receiver gather



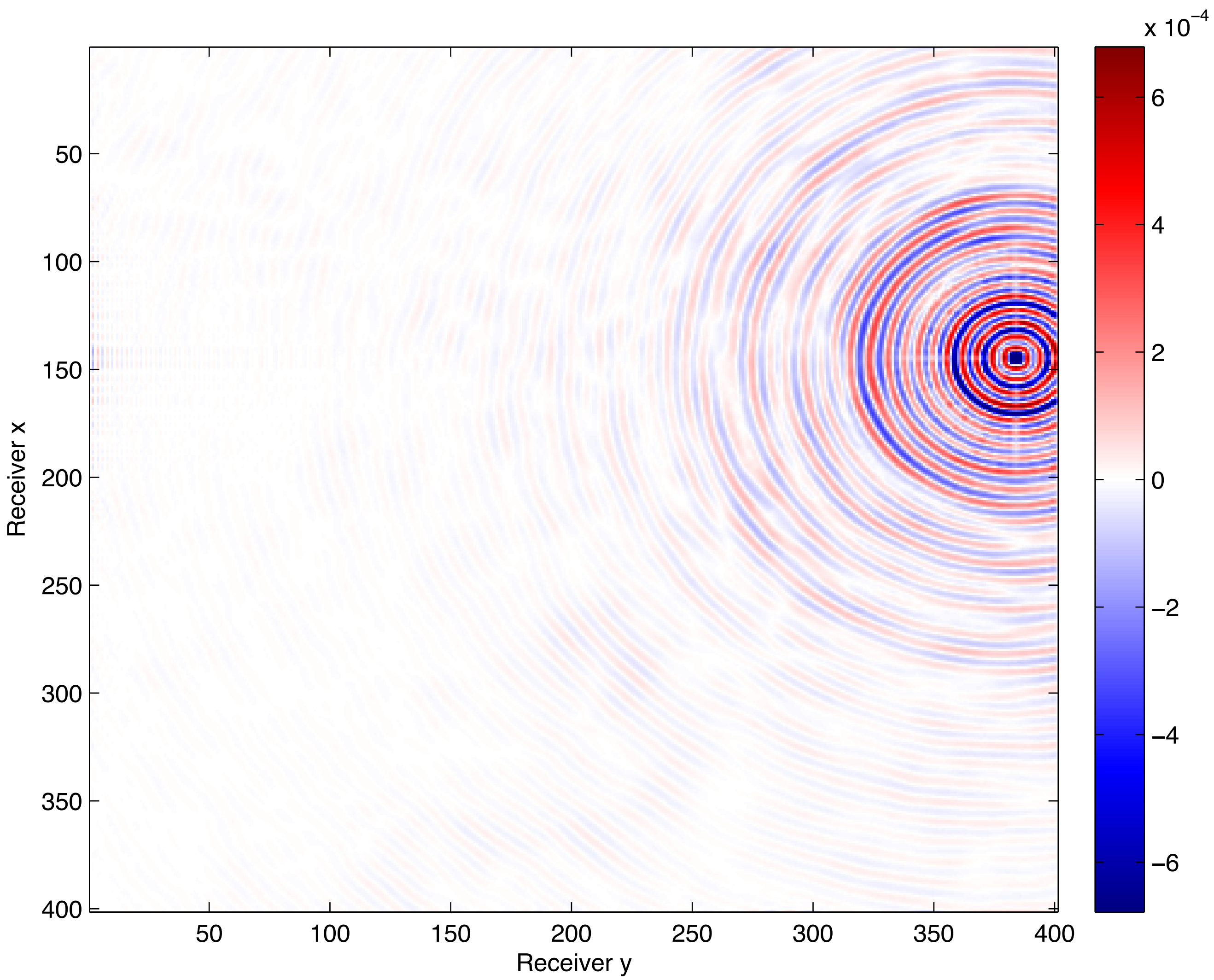


$(x_{rec}, y_{rec}) = (35,50)$

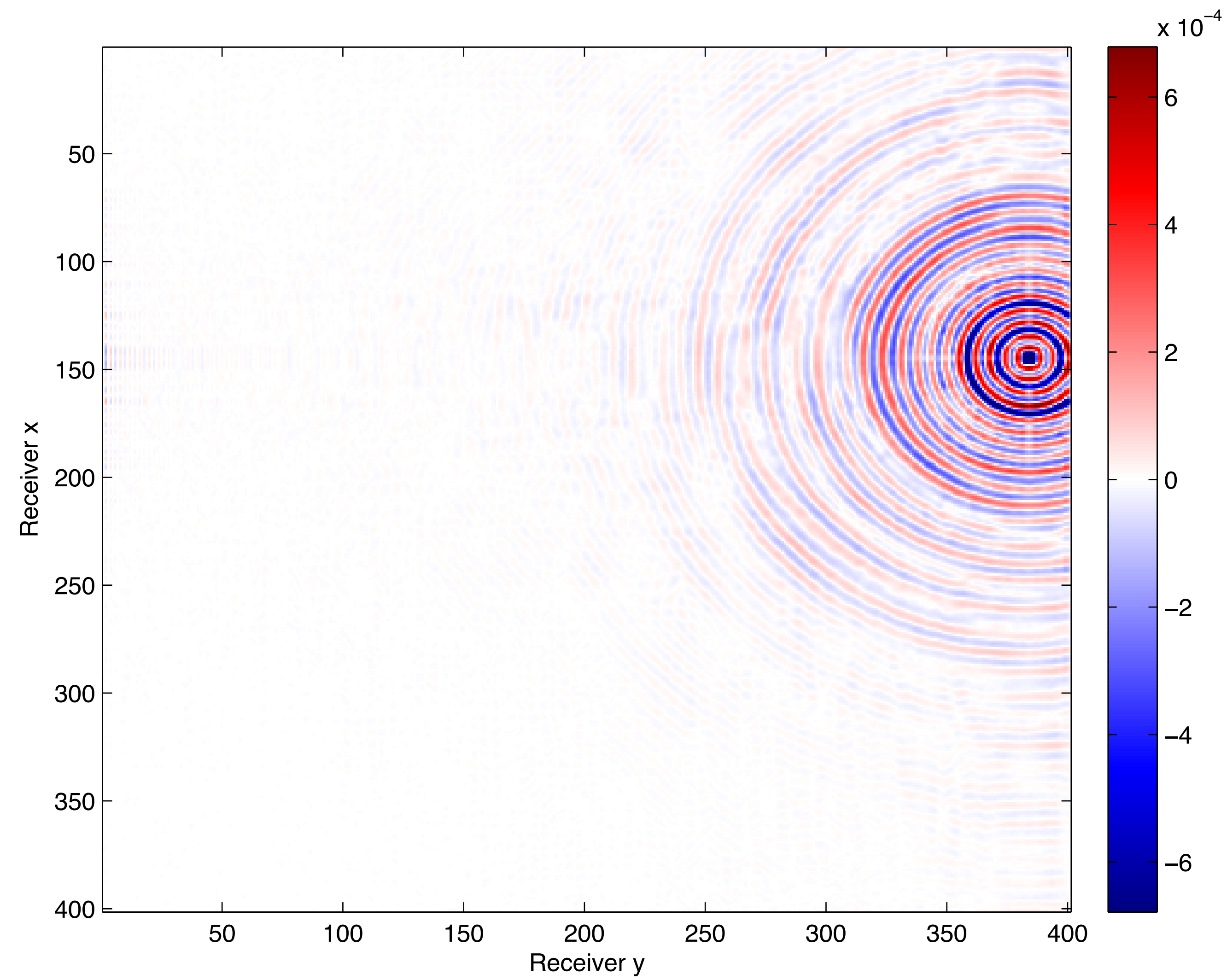
Difference

12.3 Hz - 50% missing sources

Common source gather - no data originally



$(x_{src}, y_{src}) = (25,65)$

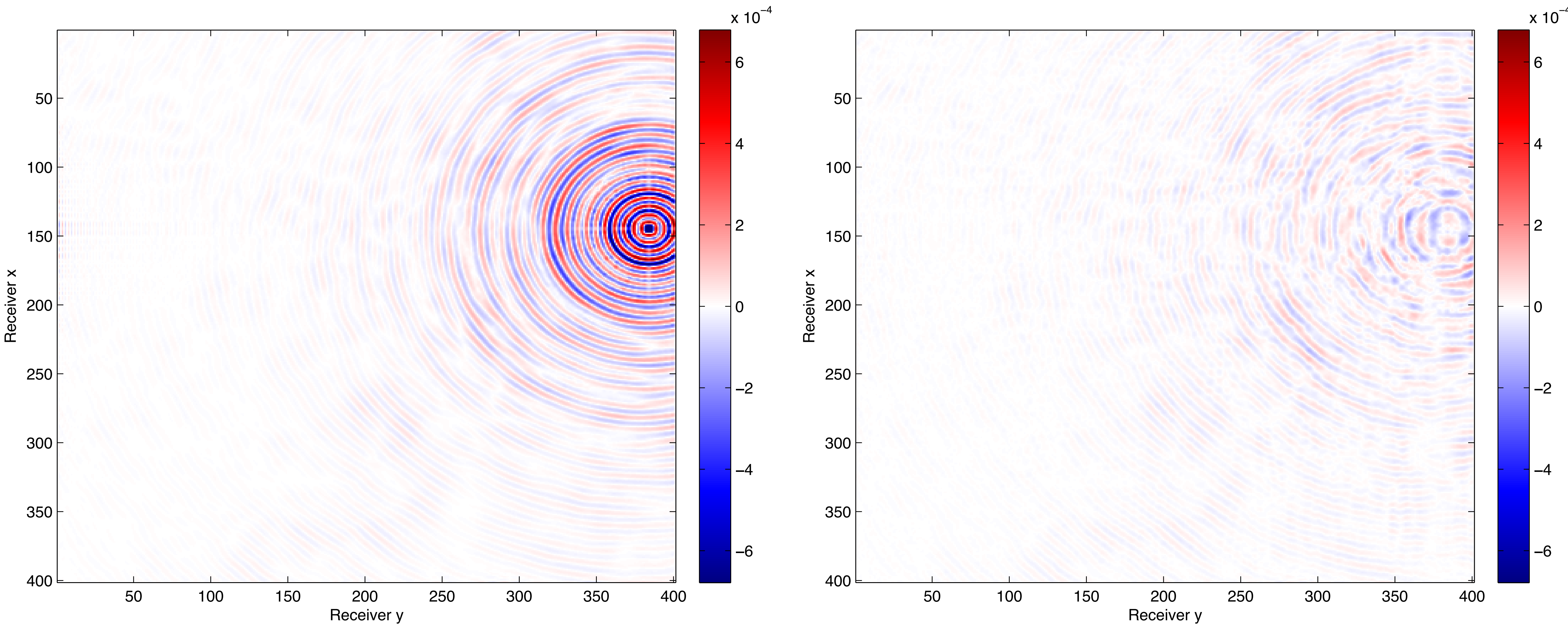


Interpolated Data

SNR 10.1 dB

12.3 Hz - 50% missing sources

Common source gather - no data originally

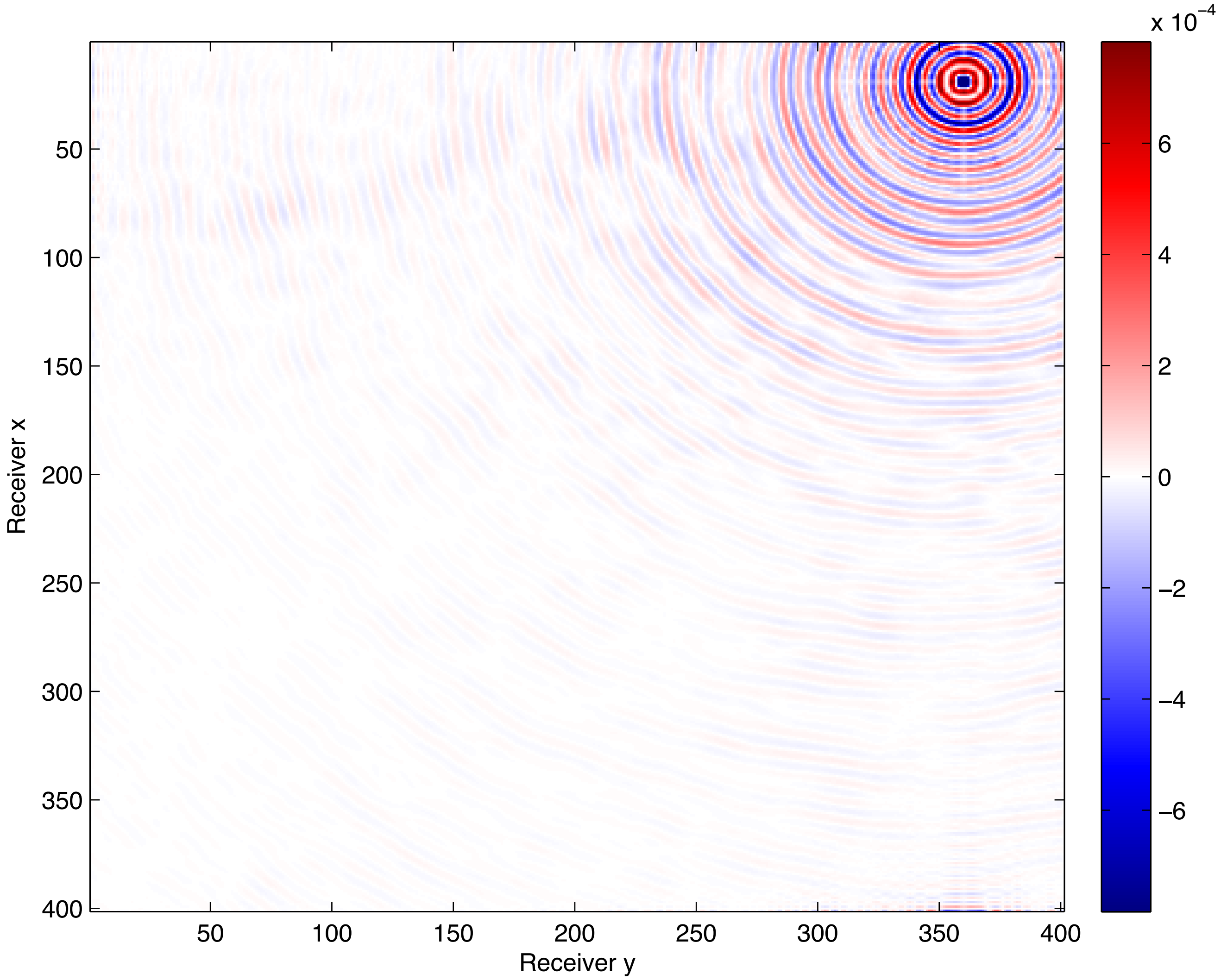


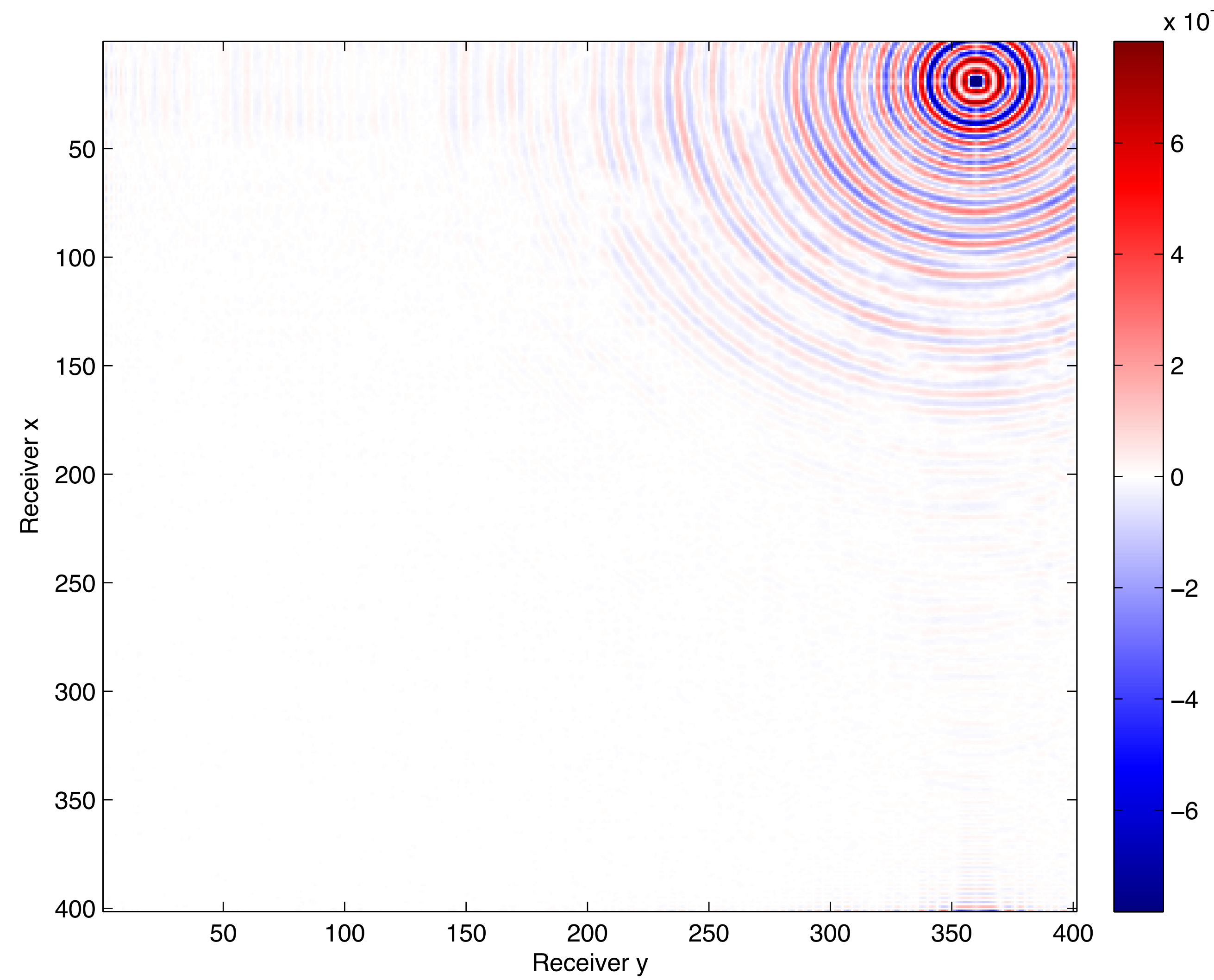
$(x_{src}, y_{src}) = (25,65)$

Difference

12.3 Hz - 50% missing sources

Common source gather - no data originally





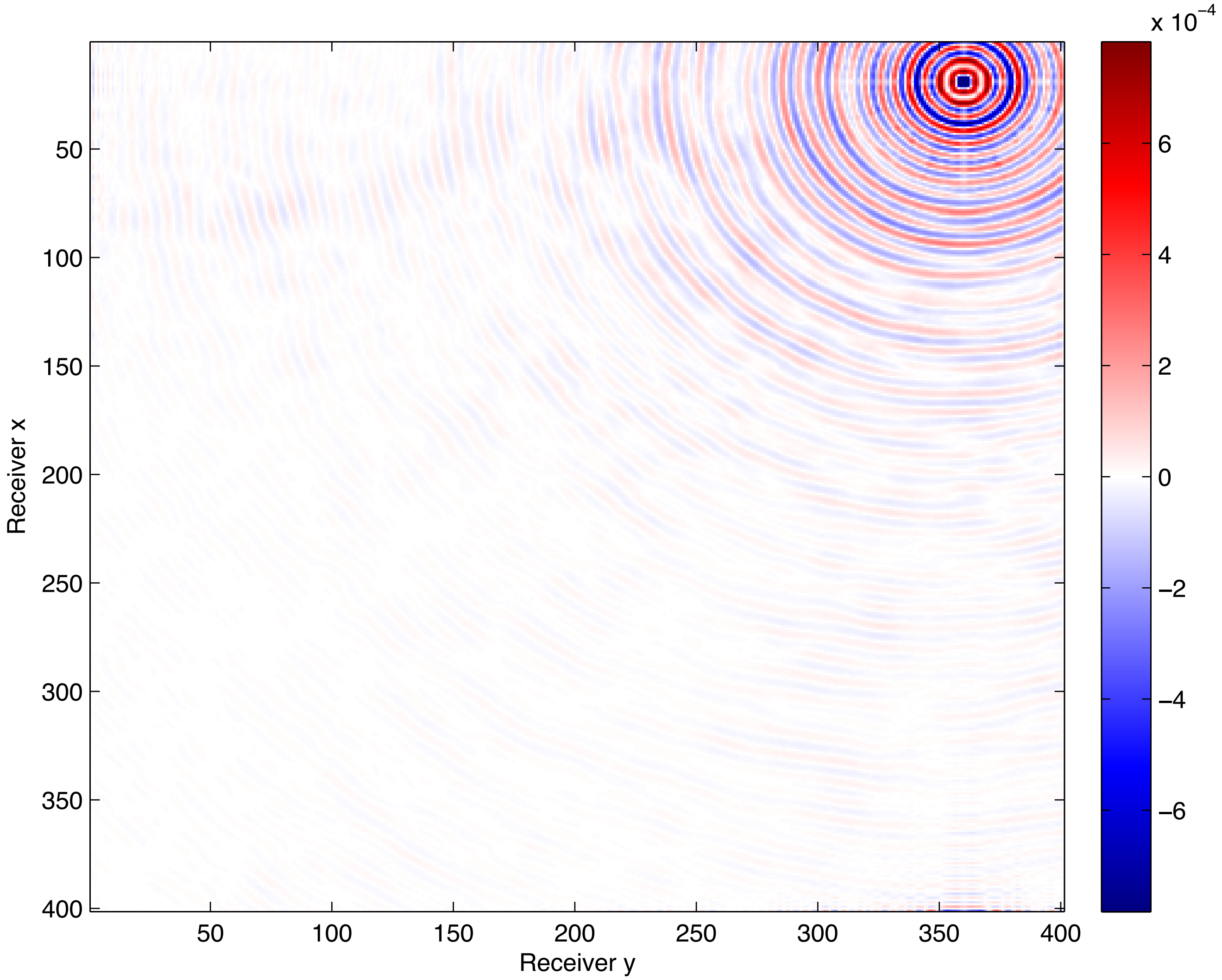
$(x_{src}, y_{src}) = (25,65)$

Interpolated Data

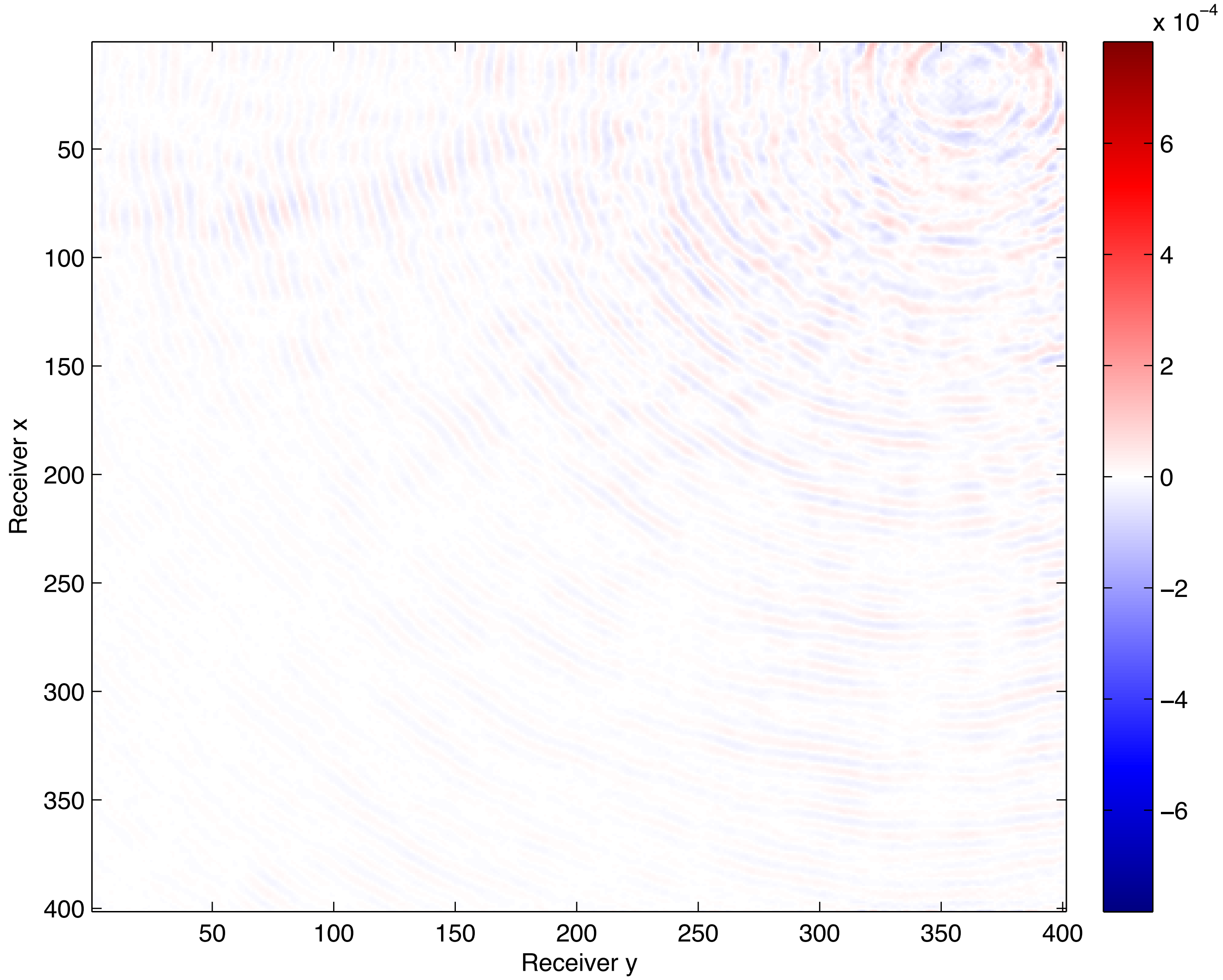
SNR 13 dB

12.3 Hz - 50% missing sources

Common source gather - no data originally



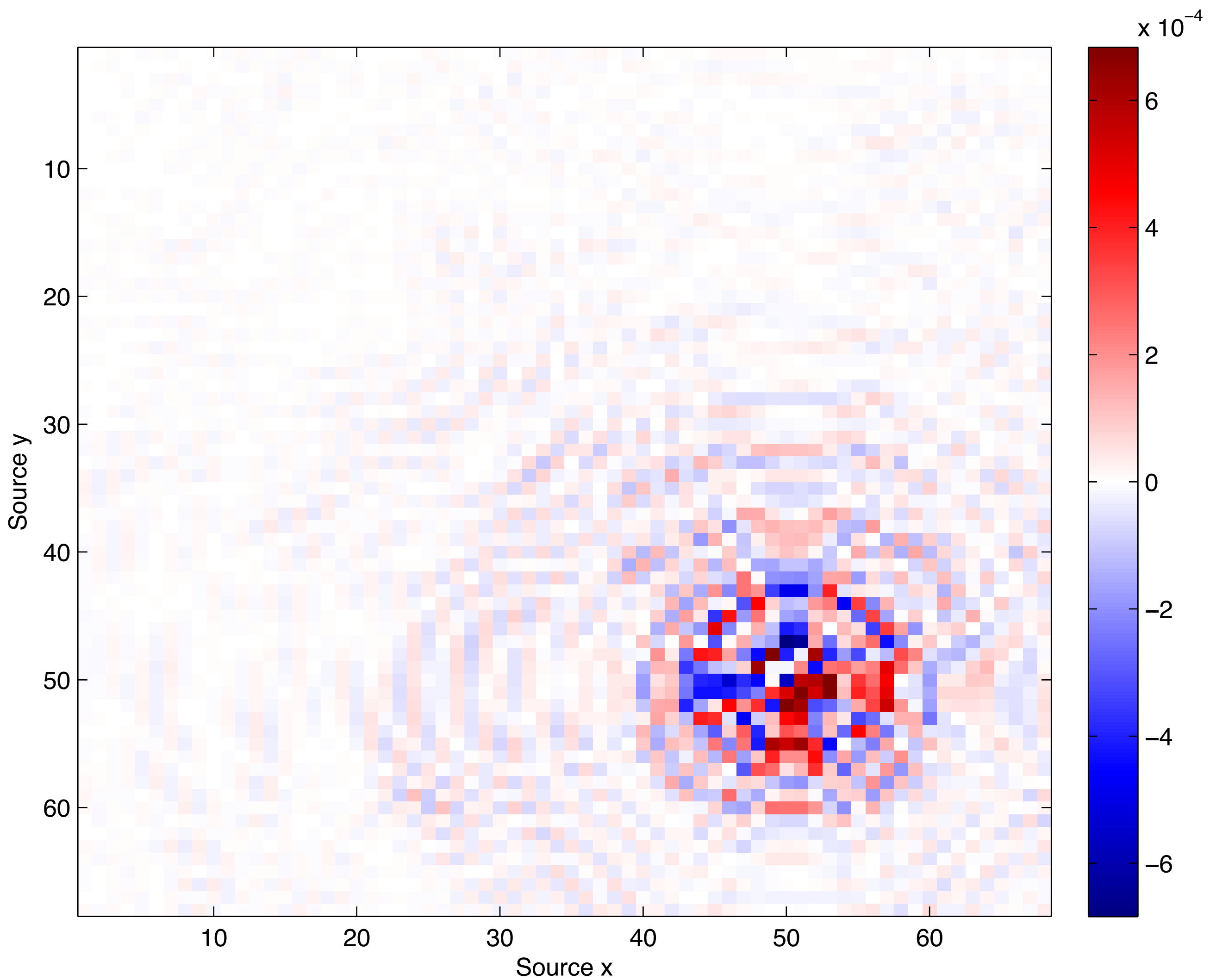
$(x_{src}, y_{src}) = (25,65)$

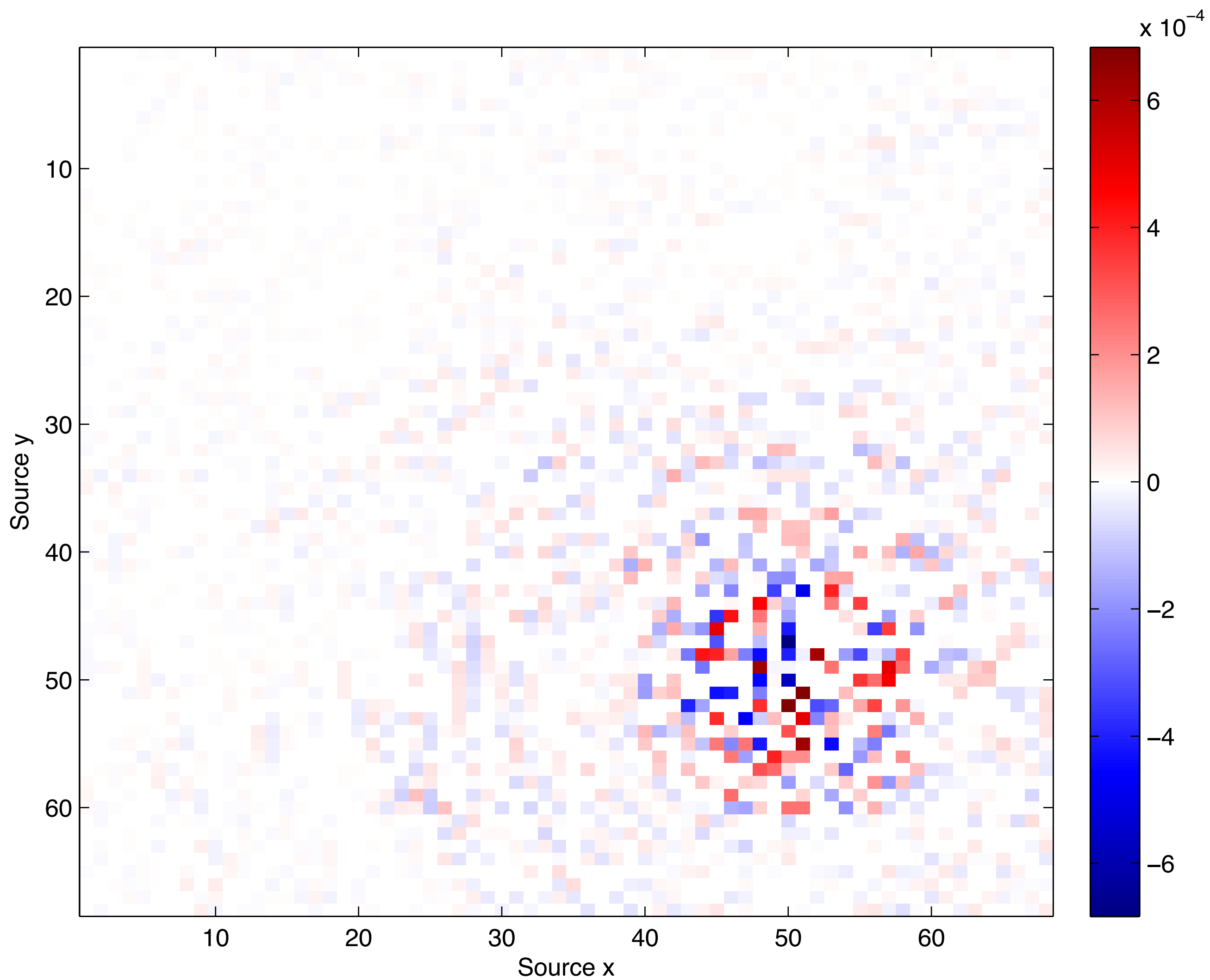


Difference

12.3 Hz - 50% missing sources

Common receiver gather



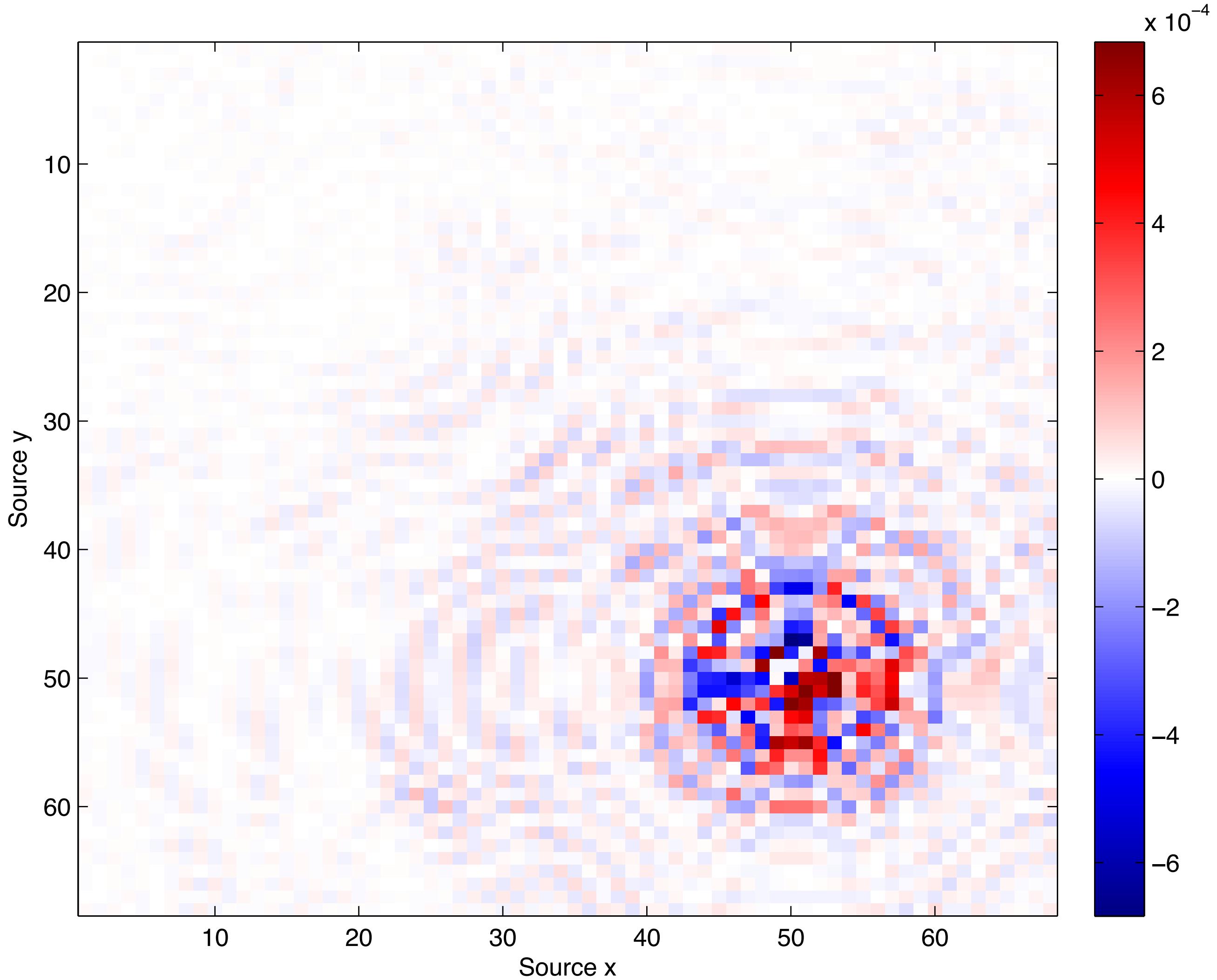


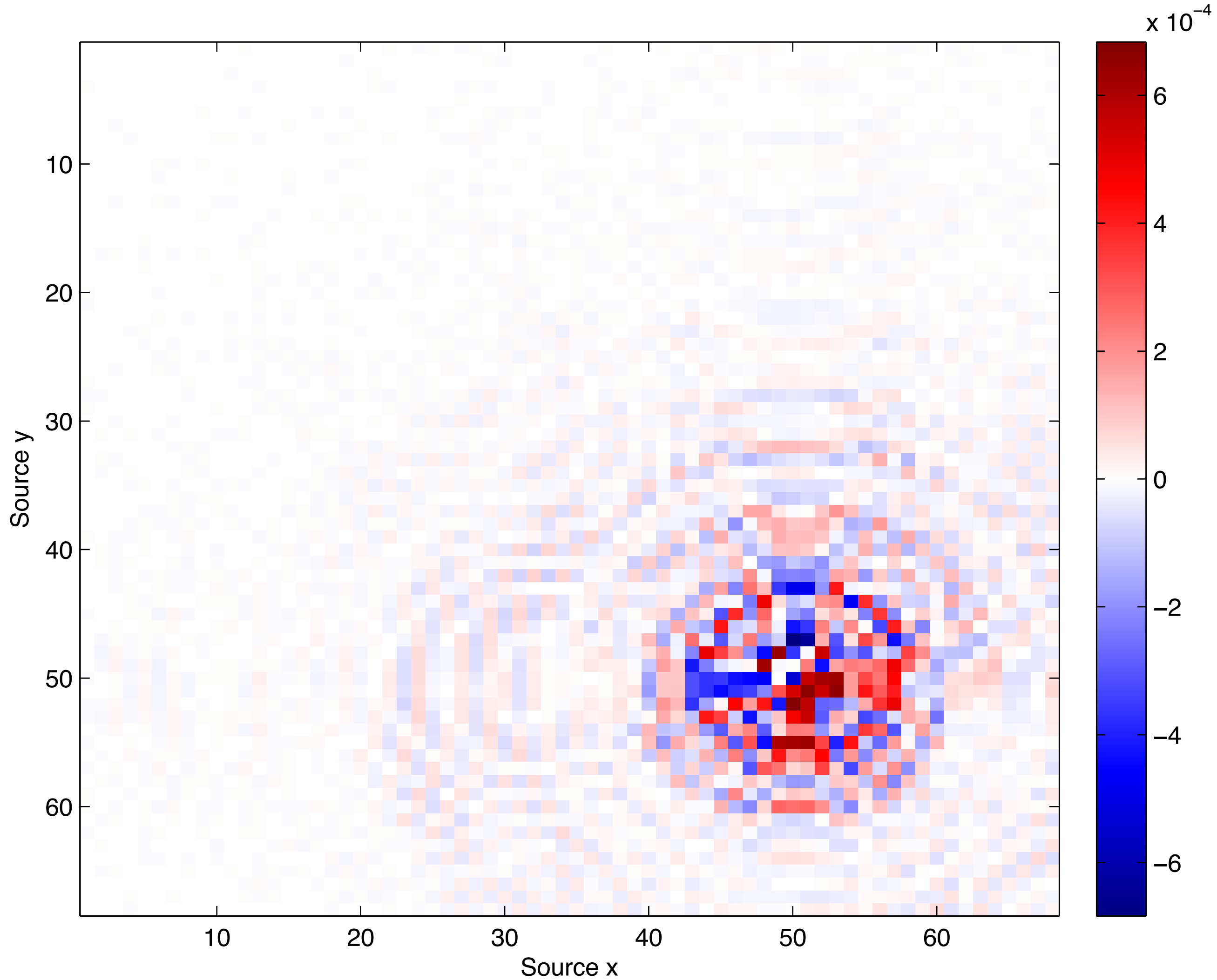
$(x_{rec}, y_{rec}) = (149,149)$

Subsampled Data

12.3 Hz - 50% missing sources

Common receiver gather





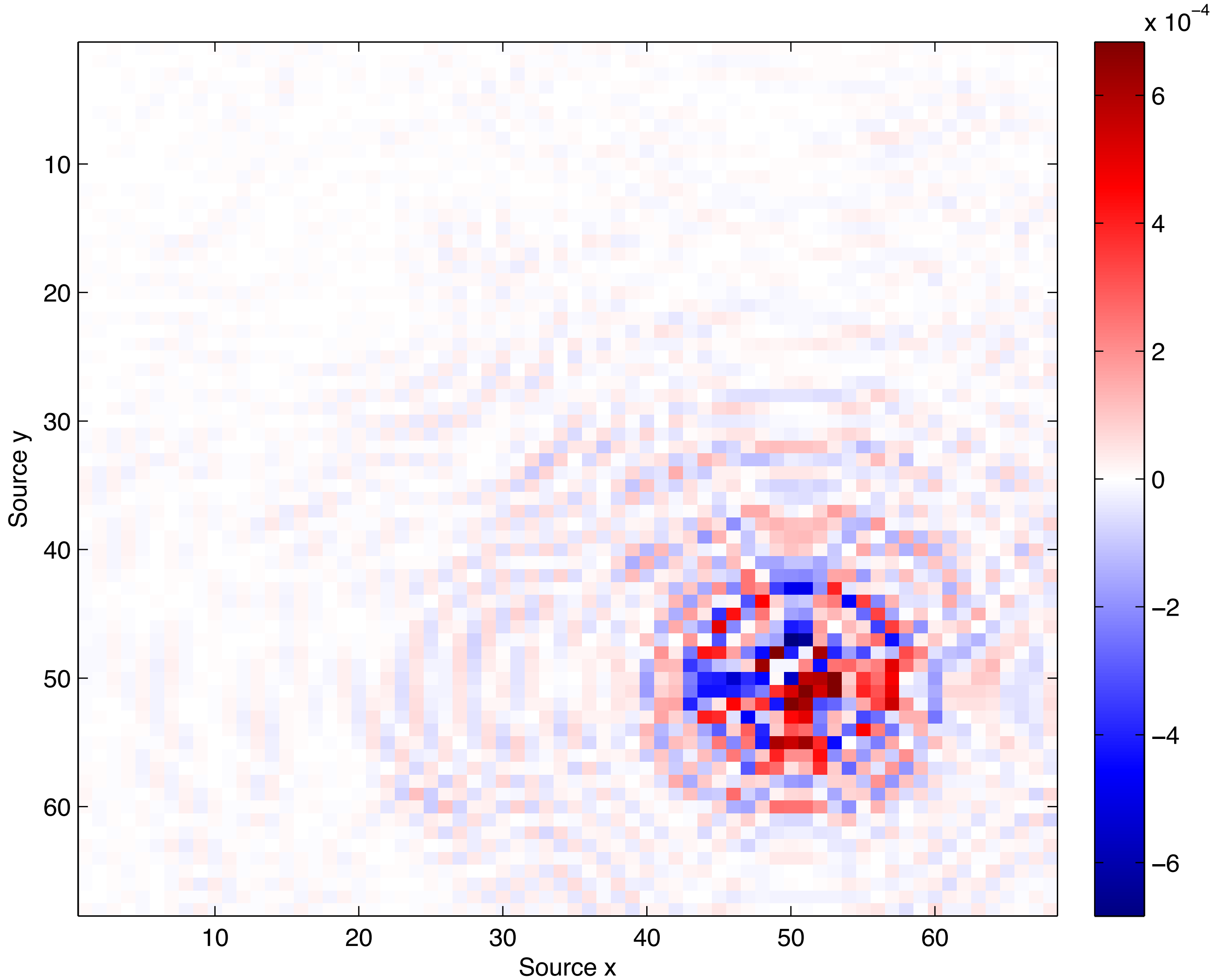
$(x_{rec}, y_{rec}) = (149,149)$

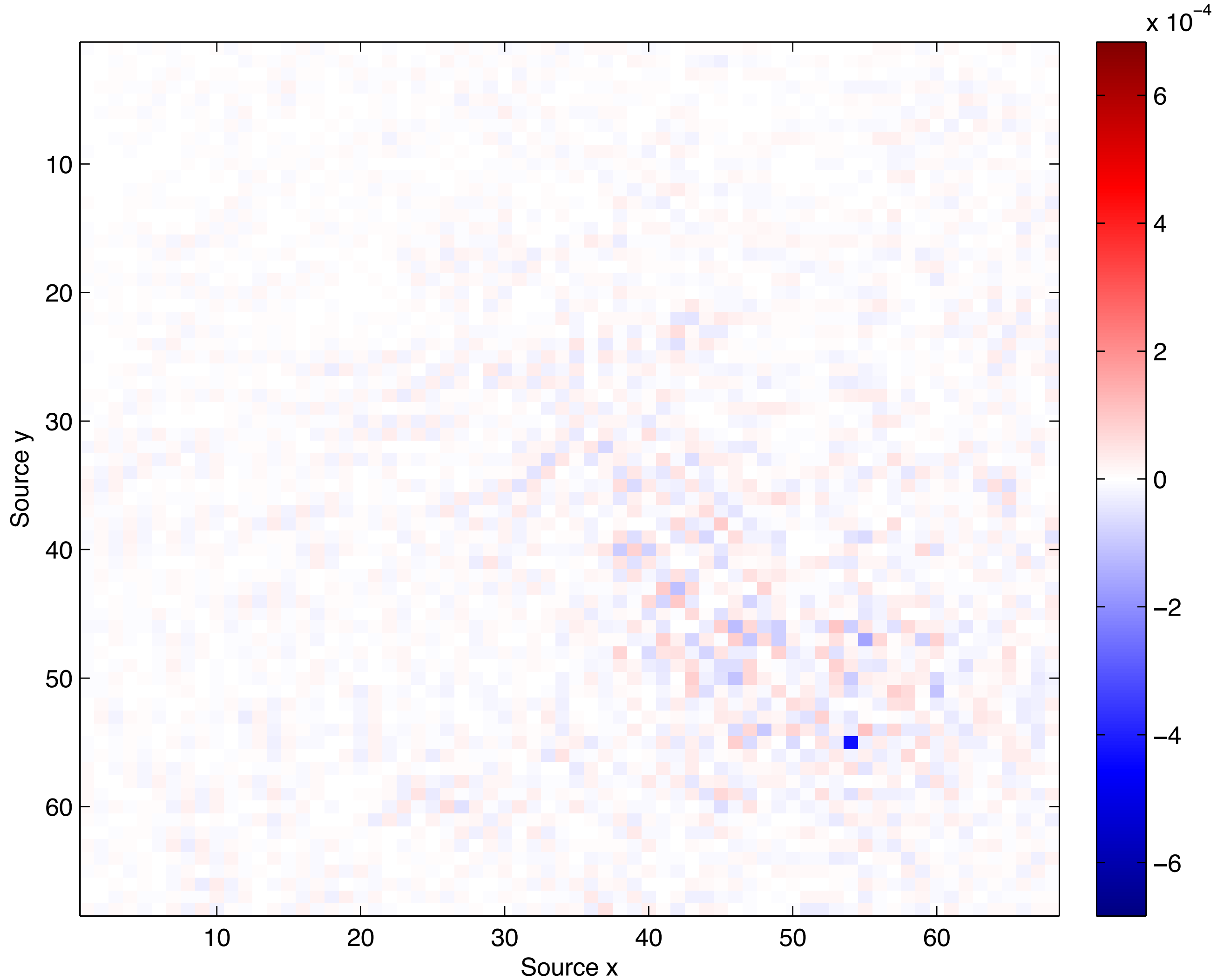
Interpolated Data

SNR 12.6 dB

12.3 Hz - 50% missing sources

Common receiver gather





$(x_{rec}, y_{rec}) = (149,149)$

Difference

SLIM

Summary - SNR

	% Missing srcs	SNR Recovered (dB)
4.86 Hz	25%	18.9
	50%	17.0
	75%	16.2

Summary - SNR

	% Missing srcs	SNR Recovered (dB)
7.34 Hz	25%	14.4
	50%	14.1
	75%	11.9

Summary - SNR

	% Missing srcs	SNR Recovered (DB)
12.3 Hz	25%	12.1
	50%	12
	75%	9.3

Conclusion

- 3D seismic data has an underlying structure that we can exploit for interpolation (Hierarchical Tucker format)
- Different schemes for organizing data - important for recovery

Conclusion

- We can interpolate HT tensors with missing entries using the *Riemannian manifold* structure of the HT format
  - Efficient optimization framework
- Achieve good results from largely subsampled data (75% missing sources)

Future Work

- Using the Hierarchical Tucker format (or other tensor formats) to speed up multidimensional convolution
- Adapt this approach to higher frequency data, which tend to be higher rank

Future Work

- Impose regularization based on an understanding the physics underlying these data sets
  - E.g. use knowledge of the behaviour of wave-propagation in 3D to prevent spurious artifacts

Acknowledgements

Thank you for your attention

SINBAD

NSERC CRSNG

This work was in part financially supported by the Natural Sciences and Engineering Research Council of Canada Discovery Grant (22R81254) and the Collaborative Research and Development Grant DNOISE II (375142-08). This research was carried out as part of the SINBAD II project with support from the following organizations: BG Group, BGP, BP, Chevron, ConocoPhillips, Petrobras, PGS, Total SA, and WesternGeco.