Composite Convex Smooth Optimization with Seismic Data Processing Applications

Curt Da Silva

SLIM

University of British Columbia

Optimization problem

We'll look at techniques for solving

$$\min_{x} h(g(x))$$

where

$h(x)$ - is convex, non-smooth, has an easy projection

$g(x)$ - is a smooth mapping

Applications - Robust Tensor Completion

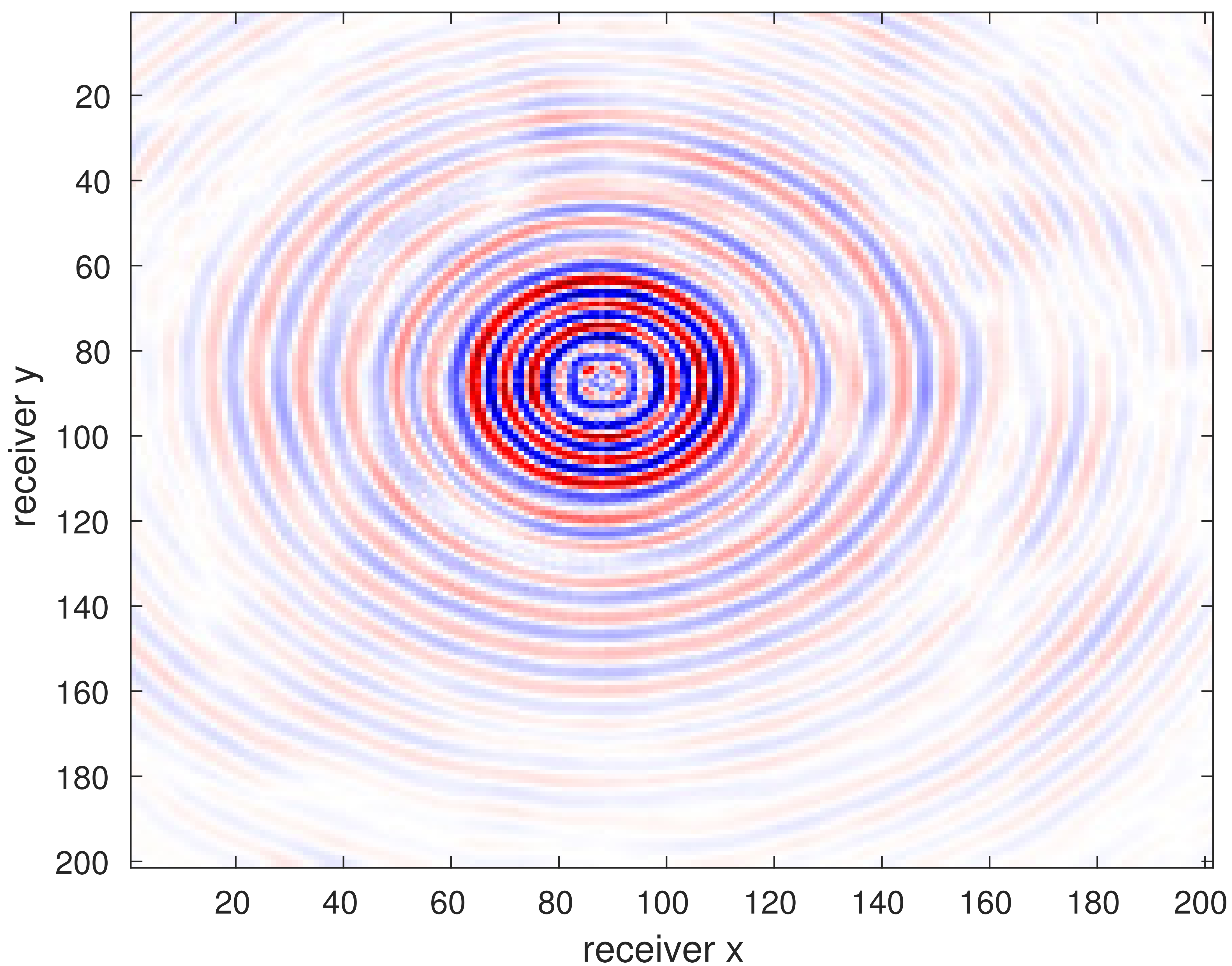
$$\min_x \overbrace{\| \underbrace{A\phi(x) - b}_{g(x)} \|_1}^{h(\cdot)=\|\cdot\|_1}$$

$\mathcal{A}$ - subsampling operator

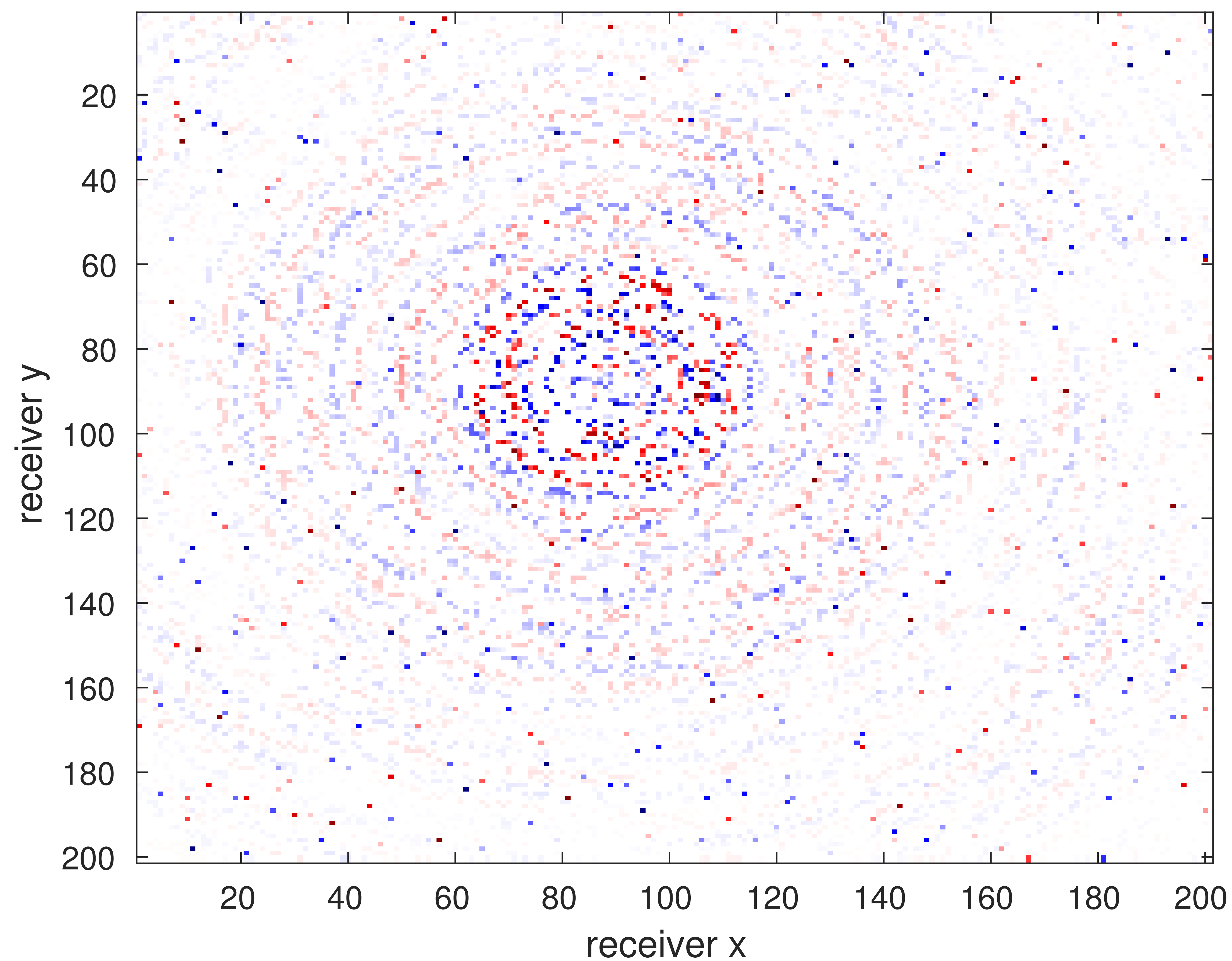
$\phi(x)$ - mapping from tensor parameters -> full tensor

b - measured data contaminated by impulsive noise

Robust Tensor Completion
75% Missing Receivers



True Data



Input Data - SNR 0dB

Optimization problem

Back to this problem

$$\min_x h(g(x))$$

where

$h(x)$ - is convex, non-smooth, has an easy projection

$g(x)$ - is a smooth mapping

Optimization problem

Standard optimization trick, introduce a new variable

$$\min_{x,y} h(y)$$
$$\text{such that} \;\; g(x) = y$$

Problem: This problem is hard to solve (nonlinear programming with a non-smooth objective)

SLIM

Optimization problem

Solution: Look at the associated *value function*

$$v(\tau) = \text{minimize}_{x,y} \quad \frac{1}{2}\|g(x) - y\|_2^2$$
$$\text{such that} \qquad h(y) \leq \tau$$

The smallest τ for which $v(\tau) = 0$ is the optimal value of the original problem (SPGL1 trick)

SLIM

Optimization problem

If we can compute $v(\tau)$ for any τ, we can use the secant method to update τ and find a root $v(\tau) = 0$

$$\tau_{k+1} = \tau_k - v(\tau_k)\frac{\tau_k - \tau_{k-1}}{v(\tau_k) - v(\tau_{k-1})}$$

SLIM

Optimization problem

If $h(x)$ is a gauge (think: nonsmooth norm like $\|\cdot\|_1$), we can upgrade the secant method to Newton's method with

$$v'(\tau) = -h^\circ(z - g(x))$$

$h^\circ(y)$ is the *polar* of h (think dual norm, like $\|\cdot\|_\infty$)

SLIM

Computing the value function

How to solve this problem?

$$v(\tau) = \text{minimize}_{x,y} \quad \frac{1}{2}\|g(x) - y\|_2^2$$
$$\text{such that} \qquad h(y) \leq \tau$$

Objective function is smooth, projection is simple

- might converge slowly

Kaufman, "A variable projection method for solving separable nonlinear least squares problems" BIT Num. Math, 1975
Aravkin, Van Leeuwen, "Estimating nuisance parameters in inverse problems" Inverse Problems, 2012

SLIM

Computing the value function

We'll use *variable projection* - for each fixed x, define

$$y(x) = \arg\min_y \frac{1}{2}\|g(x) - y\|_2^2$$
$$\text{such that } h(y) \leq \tau$$

Simple projection operation

Computing the value function

Plugging this expression back in yields

$$v(\tau) = \arg\min_{x} \frac{1}{2}\|g(x) - y(x)\|_2^2$$

Single-variable, unconstrained optimization

Can be tackled with SD, LBFGS, etc.

Computing the value function

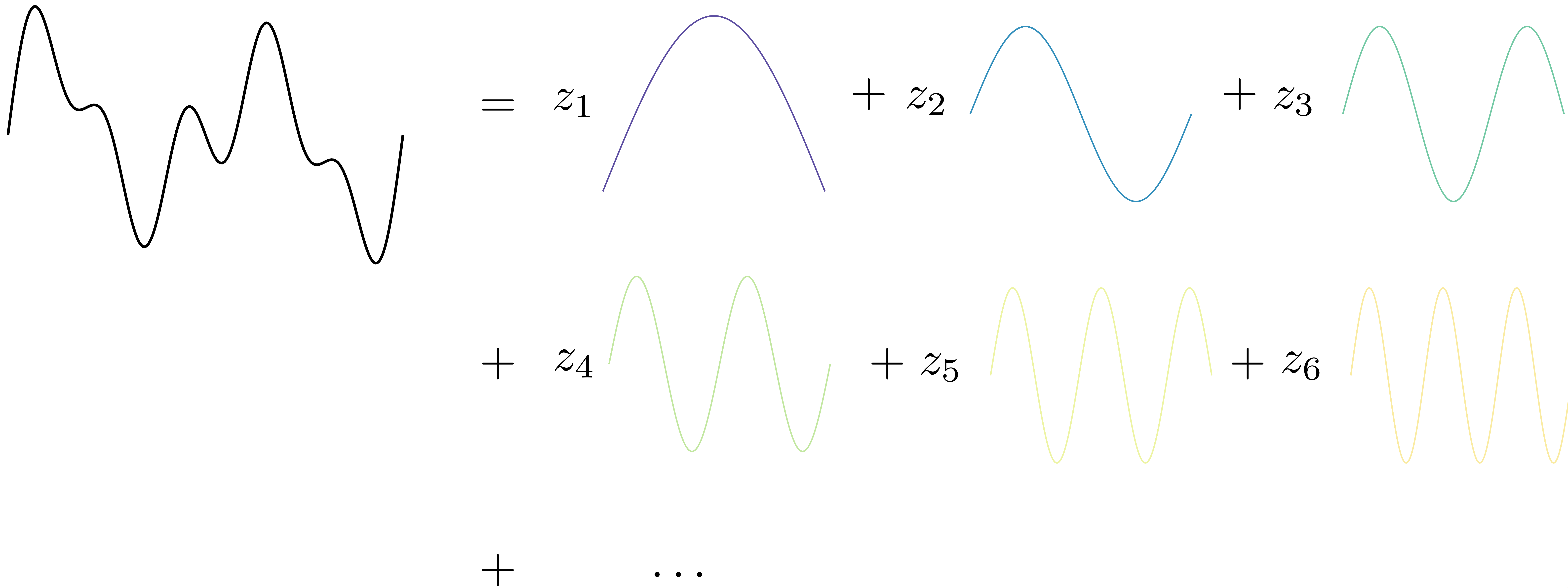
Because of variable projection, derivatives with respect to x don't change

- minimal amount of changes to existing code

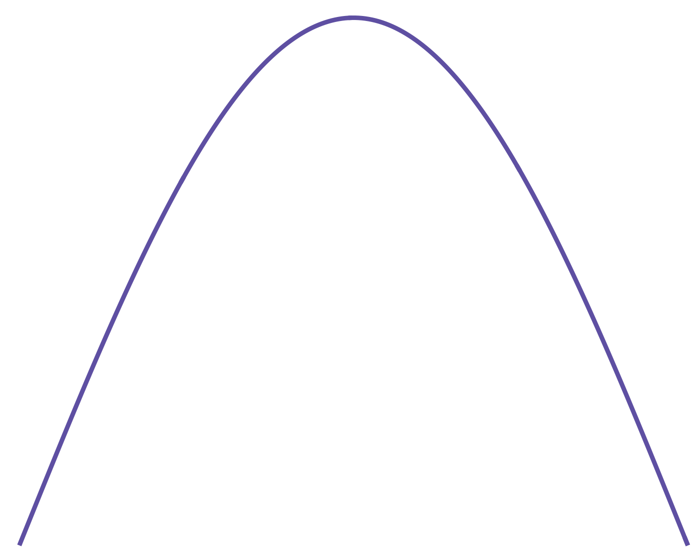
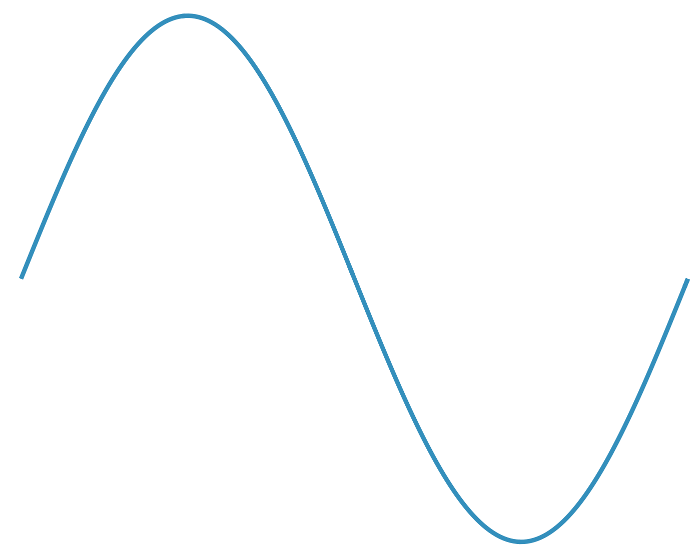
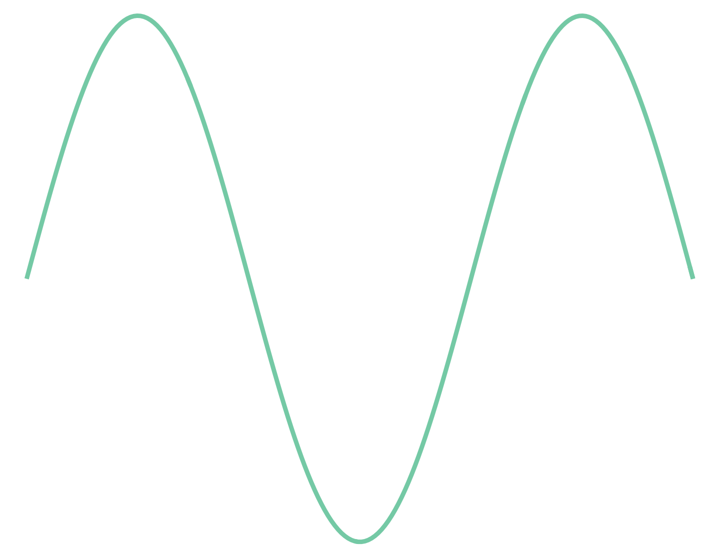
If you code can do non-linear least squares, it can handle convex-composite optimization

Synthesis VS Analysis Interlude

Synthesis-based reconstruction



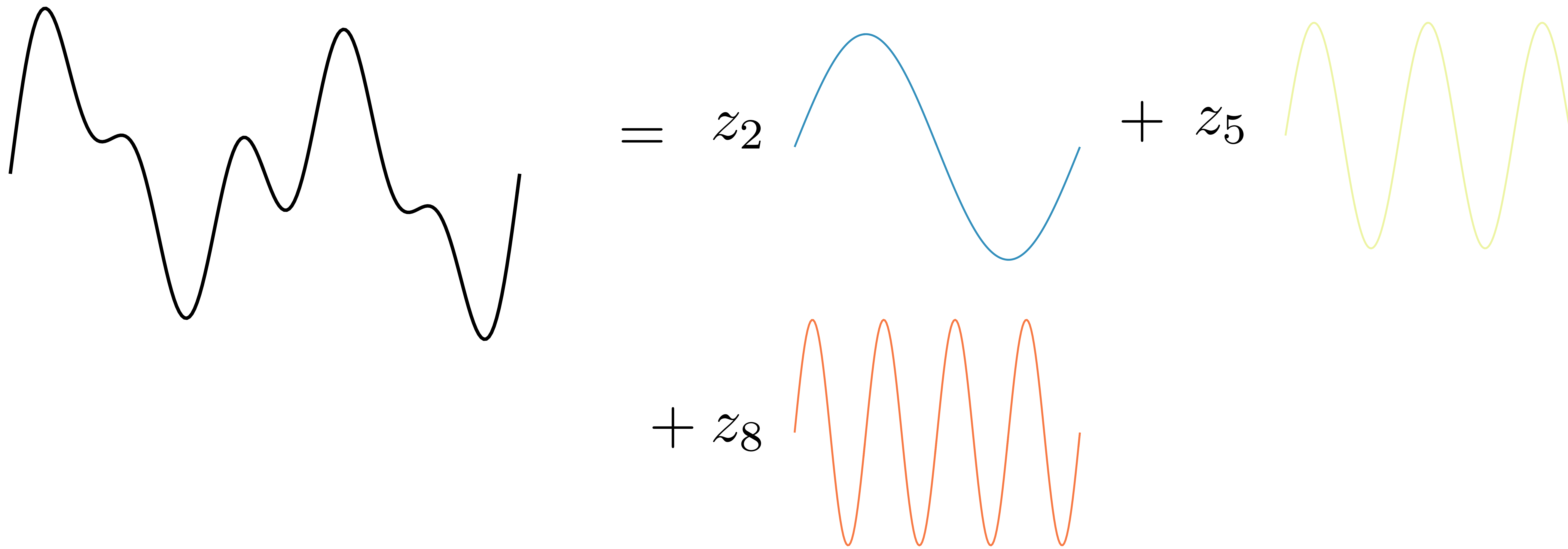
Synthesis-based reconstruction

$=\ 0$  $+\ z_2$  $+\ 0$ 

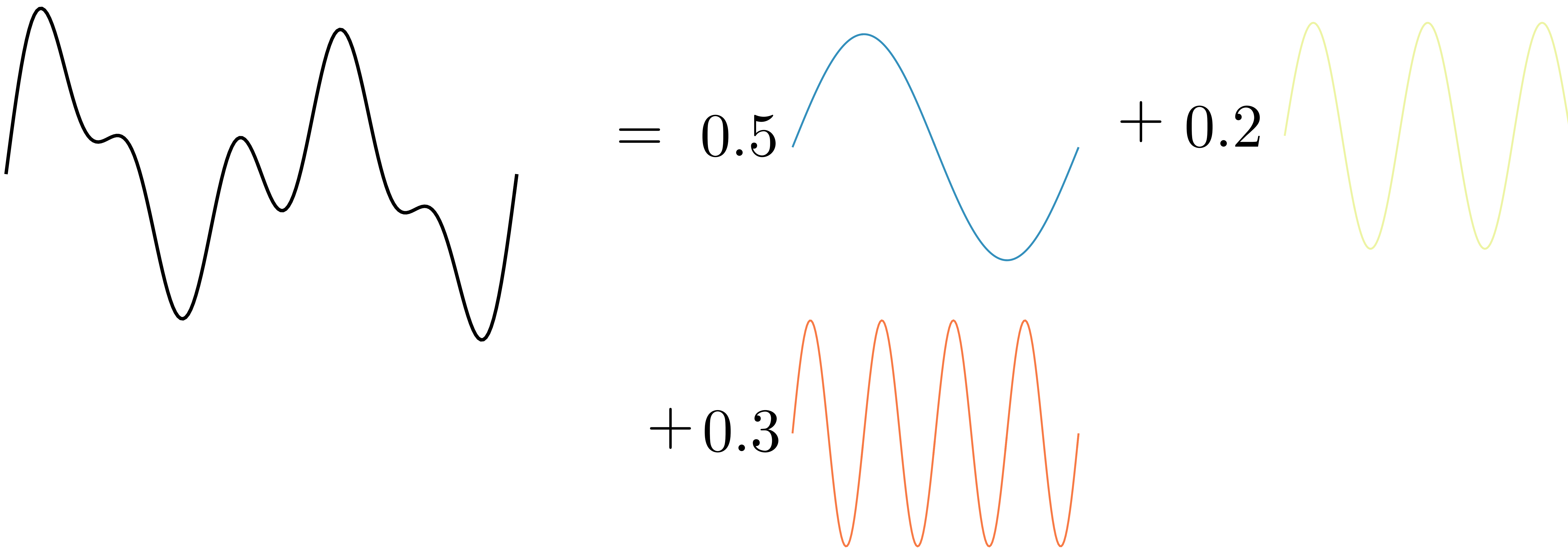
$+\ 0$ $+\ z_5$ $+\ 0$

$+\ \ldots$

Synthesis-based reconstruction

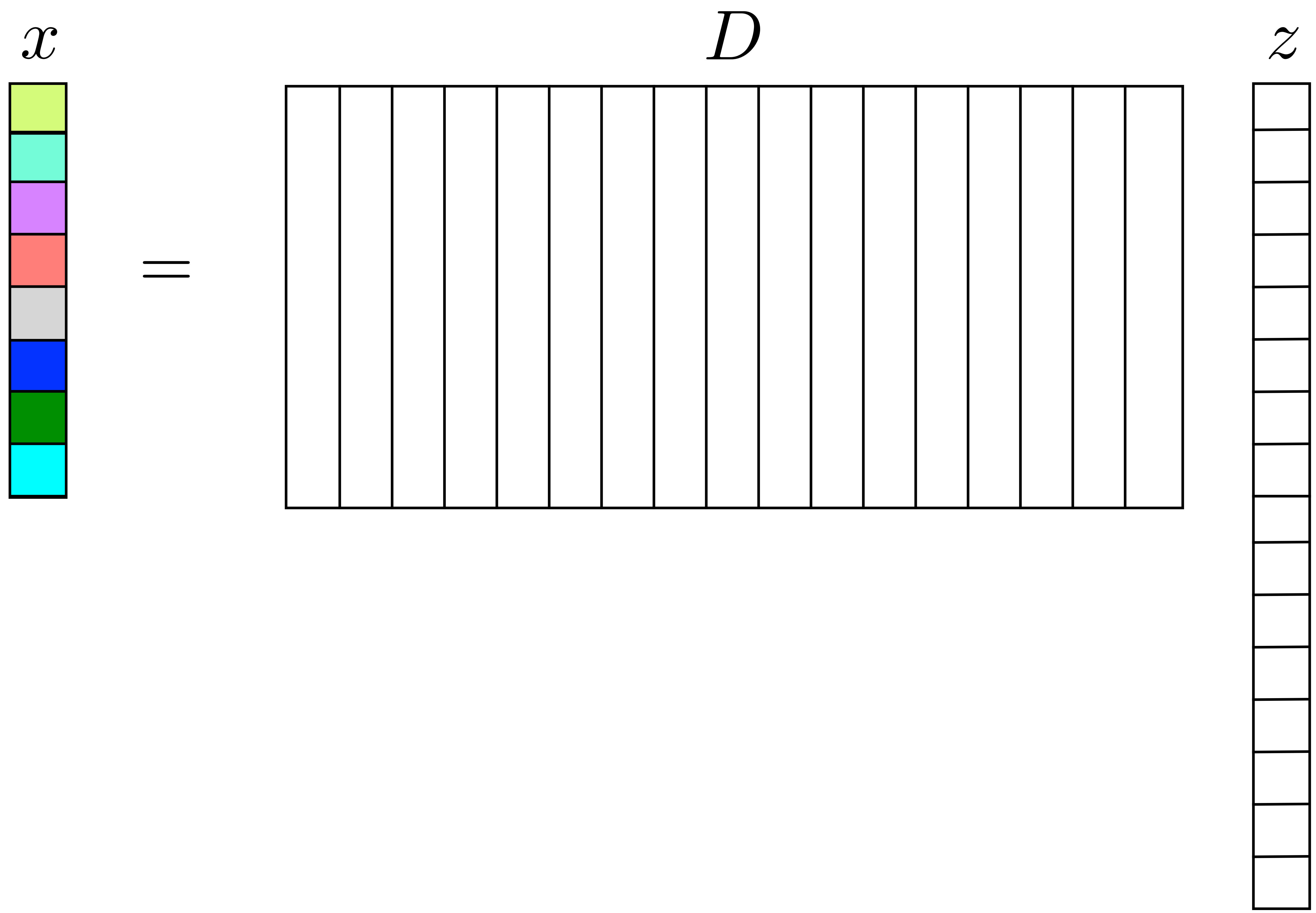


Synthesis-based reconstruction



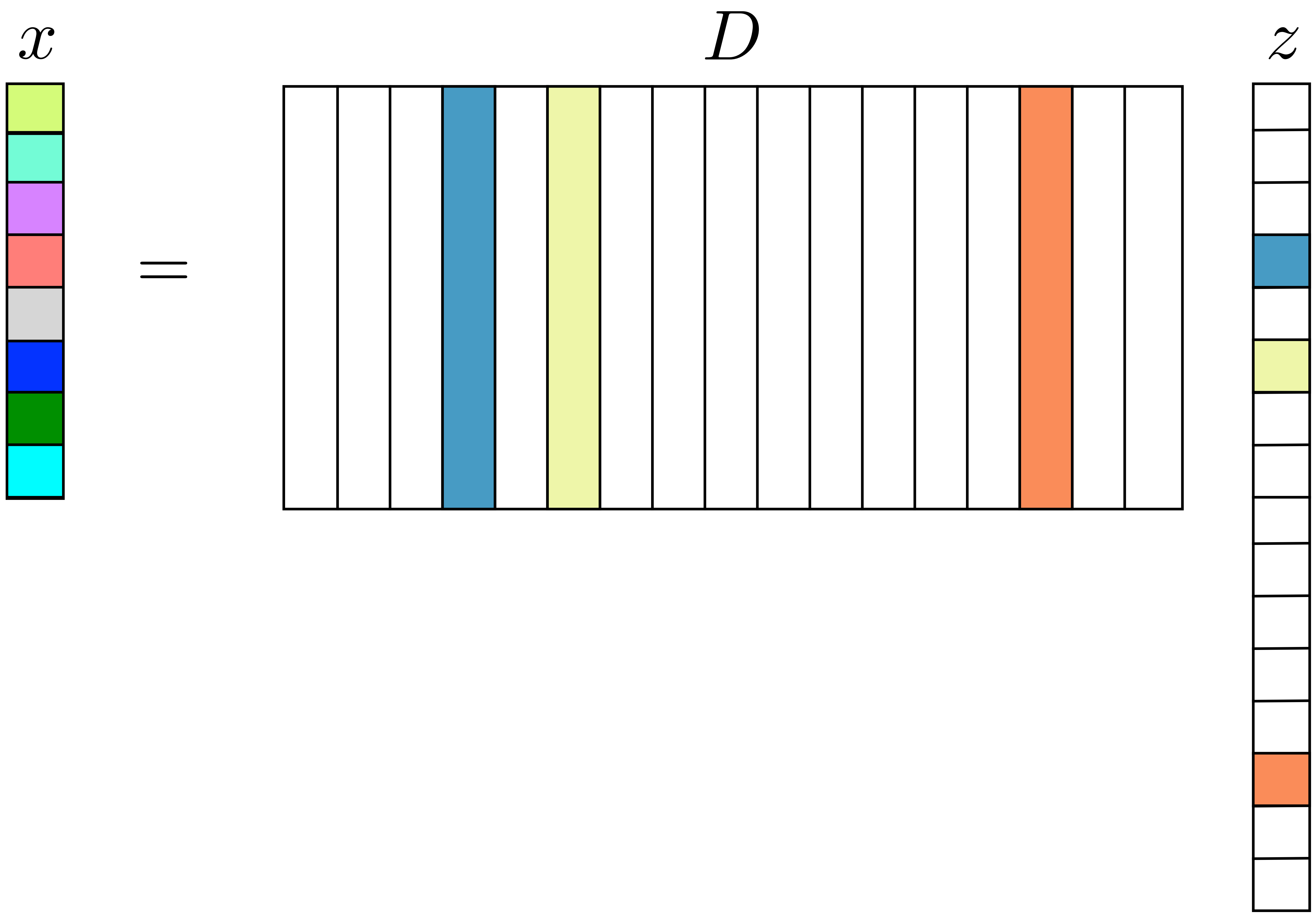
Synthesis-based reconstruction

This is the so-called *synthesis* model of signal reconstruction



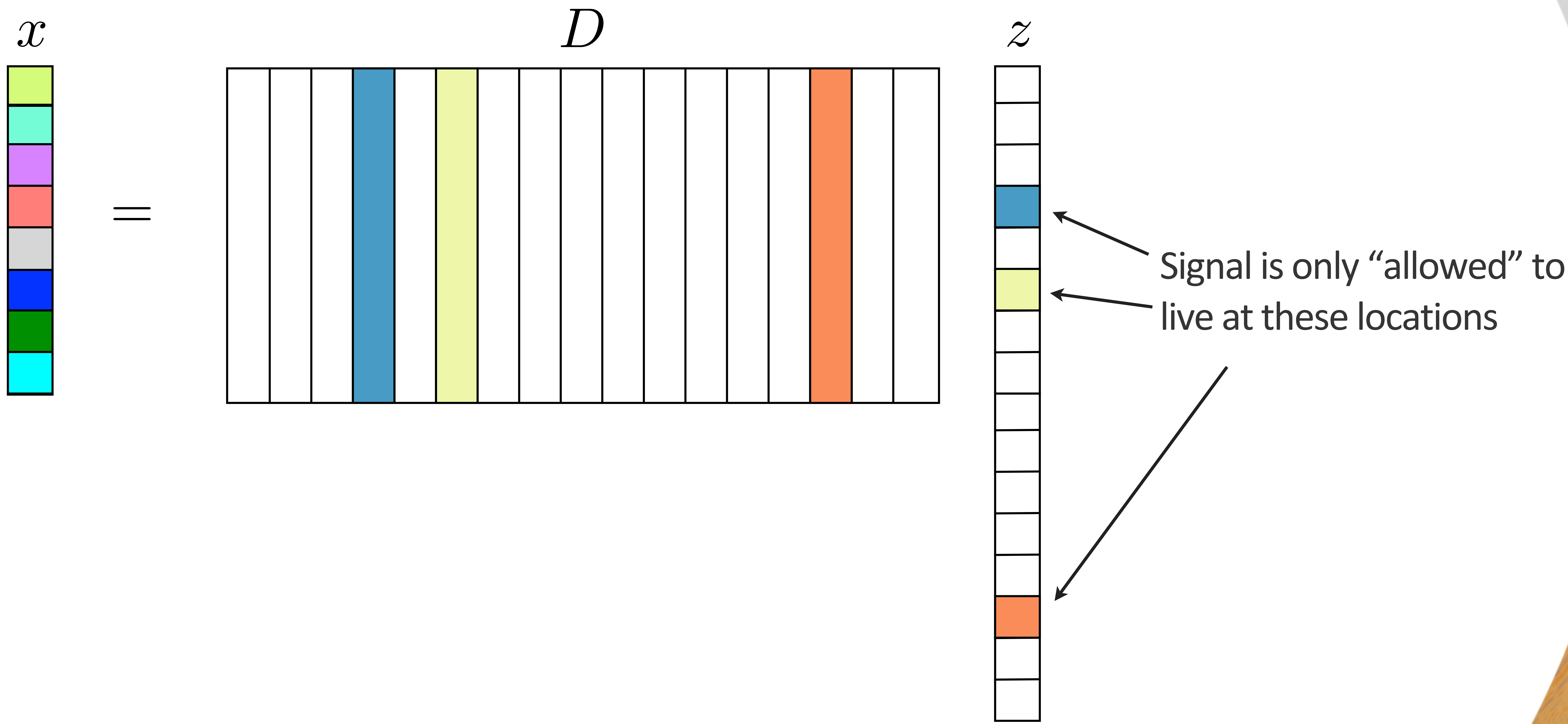
Synthesis-based reconstruction

This is the so-called *synthesis* model of signal reconstruction



Synthesis-based reconstruction

This is the so-called *synthesis* model of signal reconstruction



Synthesis-based reconstruction

Standard sparsity-promoting interpolation

$$\min_{z} \|z\|_1$$
$$\text{such that} \;\; RMz = b$$

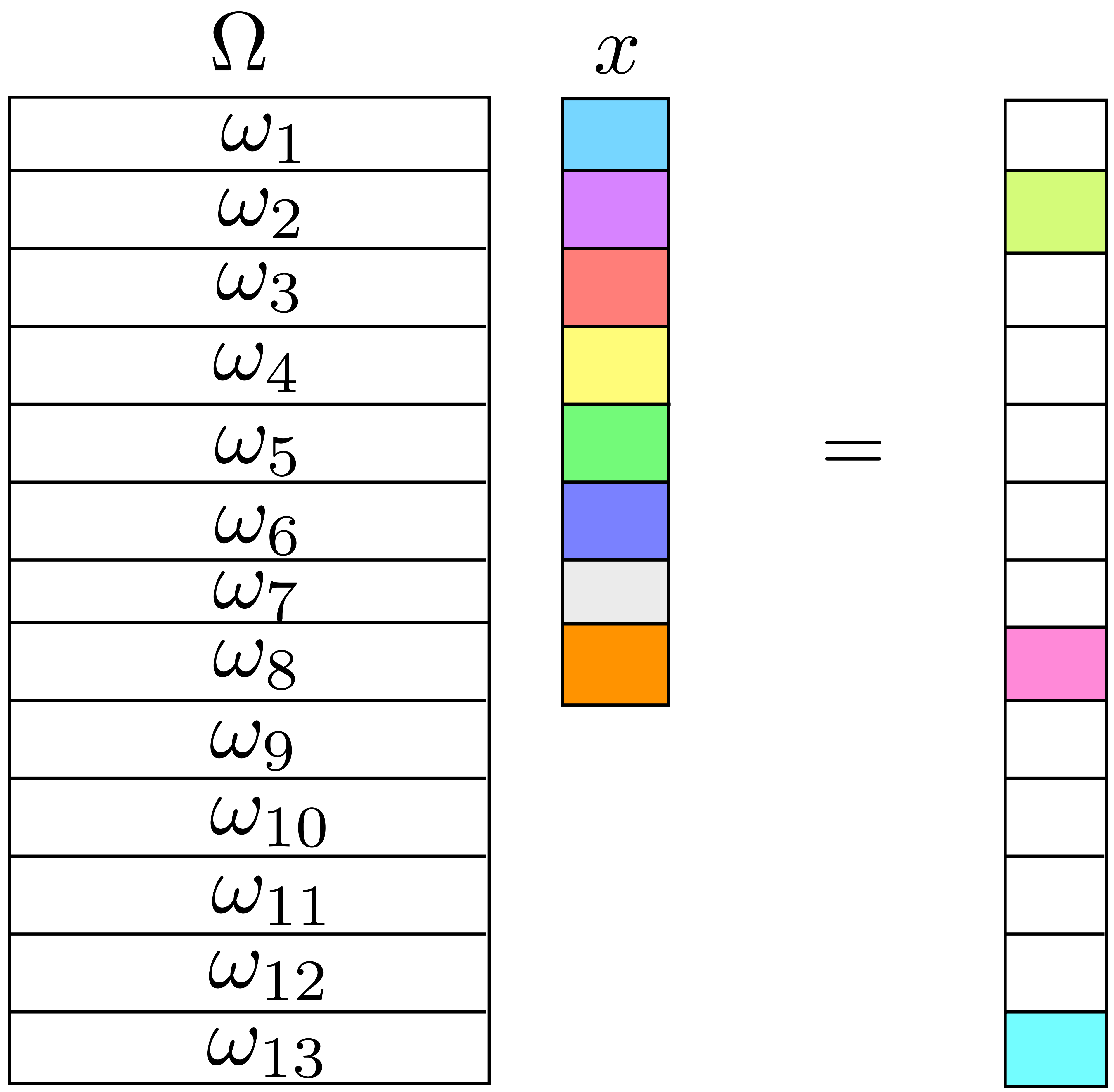
R - Trace restriction operator

M - Measurement operator (adjoint curvelet transform)

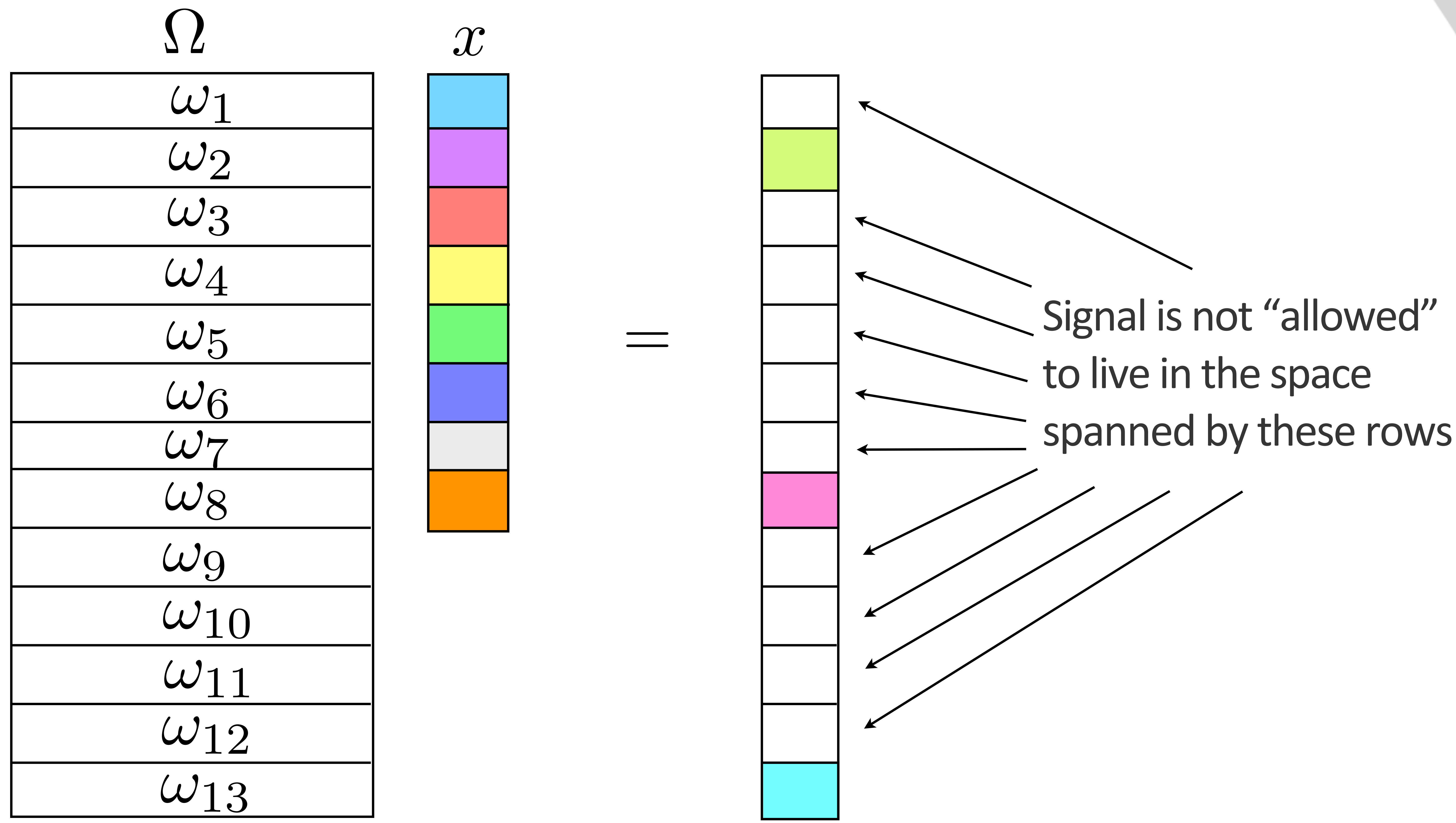
b - Measured data

z - Signal coefficients

Analysis-based reconstruction



Analysis-based reconstruction



Analysis-based reconstruction

$$\min_x \|\Omega x\|_p$$
$$\text{s.t.} Rx = b$$

R - Trace restriction operator

Ω - Cosparsity Dictionary (Curvelet Transform)

b - Measured data

x - Signal

p - 0 or 1

Synthesis VS Analysis

Note that synthesis $\neq$ analysis unless M, Ω are orthonormal bases and $M = \Omega^{-1}$

Synthesis : building up a signal through a small selection of atoms

Analysis : carve away areas of Euclidean space where a signal cannot live (i.e., orthogonal to a large number of atoms)

Example: Analysis-based interpolation

$$\min_x \|\Omega x\|_1$$
$$\text{such that } Ax = b$$

$$v(\tau) = \min_x \frac{1}{2}\|Ax - b\|_2^2 + \frac{1}{2}\|\Omega x - y(x)\|_2^2$$

$$y(x) = \arg\min_y \frac{1}{2}\|\Omega x - y\|_2^2$$
$$\text{such that } \|y\|_1 \le \tau$$

Matlab code - objective evaluation

```
function [f,g] = cosparsity_obj(A,b,x,Omega,tau)
    r = A*x-b;
    z = Omega*x;
    y = NormL1_project(z,tau);

    z = z-y;

    f = 0.5*norm(r)^2 + 0.5*norm(z)^2;
    if nargout >=2
        g = A'*r + Omega'*z;
    end
end
```

Matlab code - outer loop

```
P = @(x,tau) NormL1_project(x,tau);
obj = @(x,tau) cosparsity_obj(A,b,x,Om,tau);
[x,f1] = minFunc(@(x) obj(x,tau1),x,inner_opts);
for i=1:ntau_updates
    c = Om*x;
    df = -norm(P(c,tau1)-c,'inf');
    tau1 = tau1 - f1/df;
    [x,f1] = minFunc(@(x) obj(x,tau1),x,inner_opts);
end
```

Example: Analysis-based interpolation

$$\min_x \|\Omega x\|_0$$

$$\text{such that} Ax = b$$

$$v(\tau) = \min_x \frac{1}{2}\|Ax - b\|_2^2 + \frac{1}{2}\|\Omega x - y(x)\|_2^2$$

$$y(x) = \arg\min_y \frac{1}{2}\|\Omega x - y\|_2^2$$

$$\text{such that } \|y\|_0 \leq \tau$$

Note that τ is now integer-valued - need to round secant method update

Compare to GAP method

Nam, et. al., “The cosparse analysis model and algorithms” (2013)

Start with full index set of rows of $\Omega \in \mathbb{C}^{n \times d}$

$$\Lambda = \{1, \ldots, n\}$$

1. Projection: compute $z = \Omega x_k$
2. Find the largest elements of z
3. Remove the corresponding rows from Λ
4. Update solution estimate

$$x_{k+1} = \arg\min_x \|\Omega_\Lambda x\|_2 \text{ subject to } y = Ax$$

Compare to GAP method

Nam, et. al., “The cosparse analysis model and algorithms” (2013)

SLIM

Start with full index set of rows of $\Omega \in \mathbb{C}^{n \times d}$

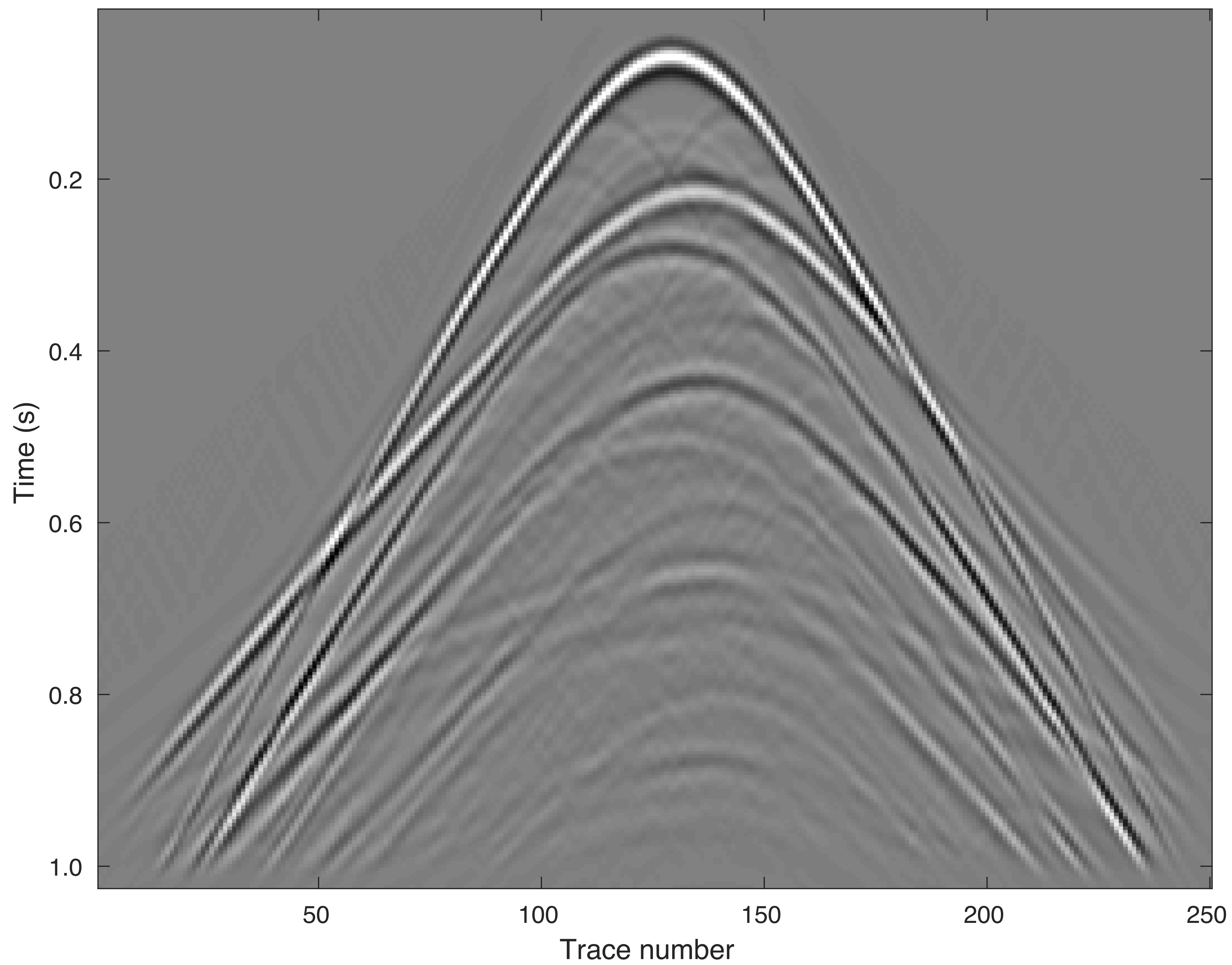
$$\Lambda = \{1, \ldots, n\}$$

1. Projection: compute $z = \Omega x_k$
2. Find the largest elements of z
3. Remove the corresponding rows from Λ
4. Update solution estimate

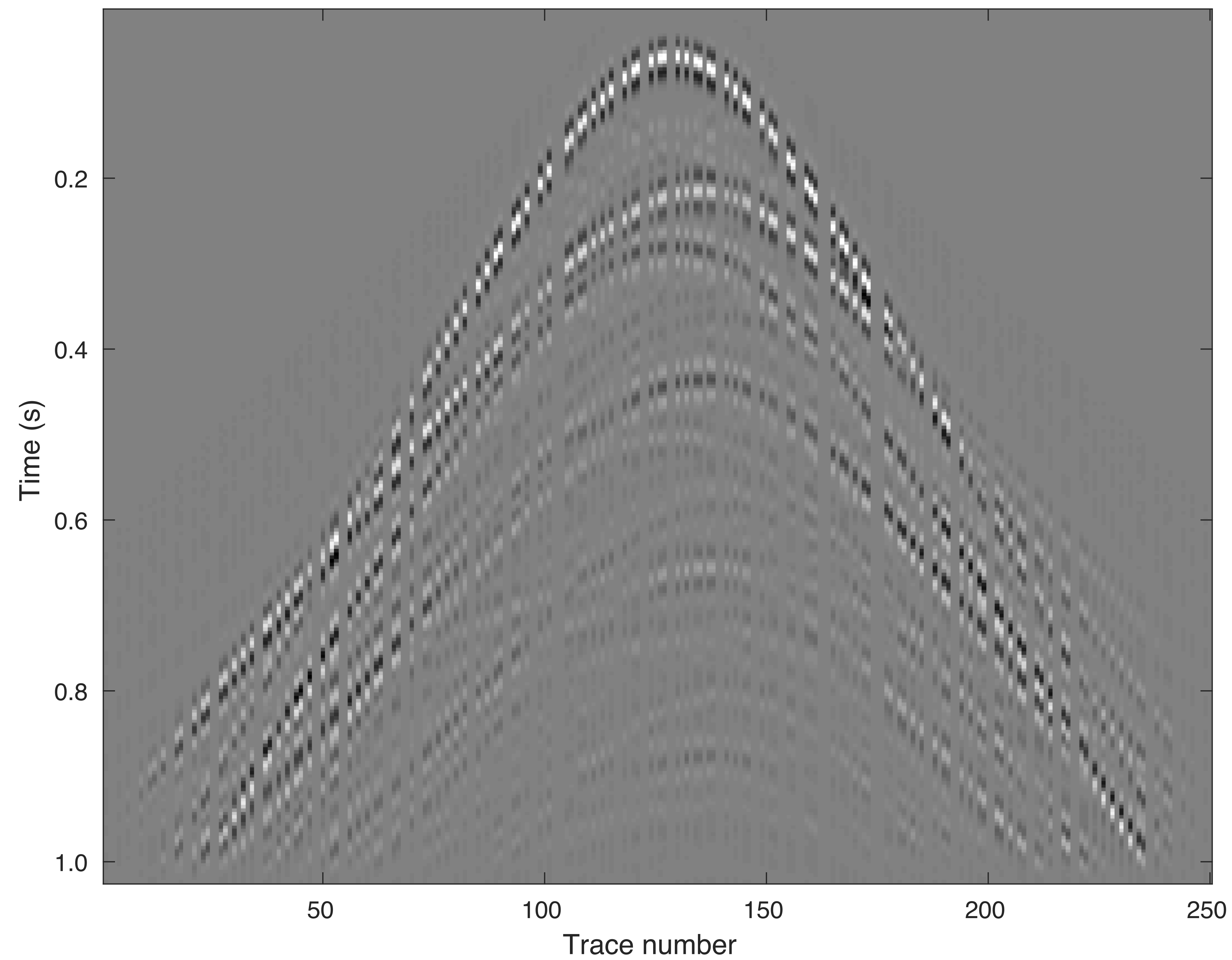
$$x_{k+1} = \arg\min_x \|\Omega_\Lambda x\|_2 \text{ subject to } y = Ax$$

Costly/complicated

Cosparsity VS Sparsity 50% Missing sources

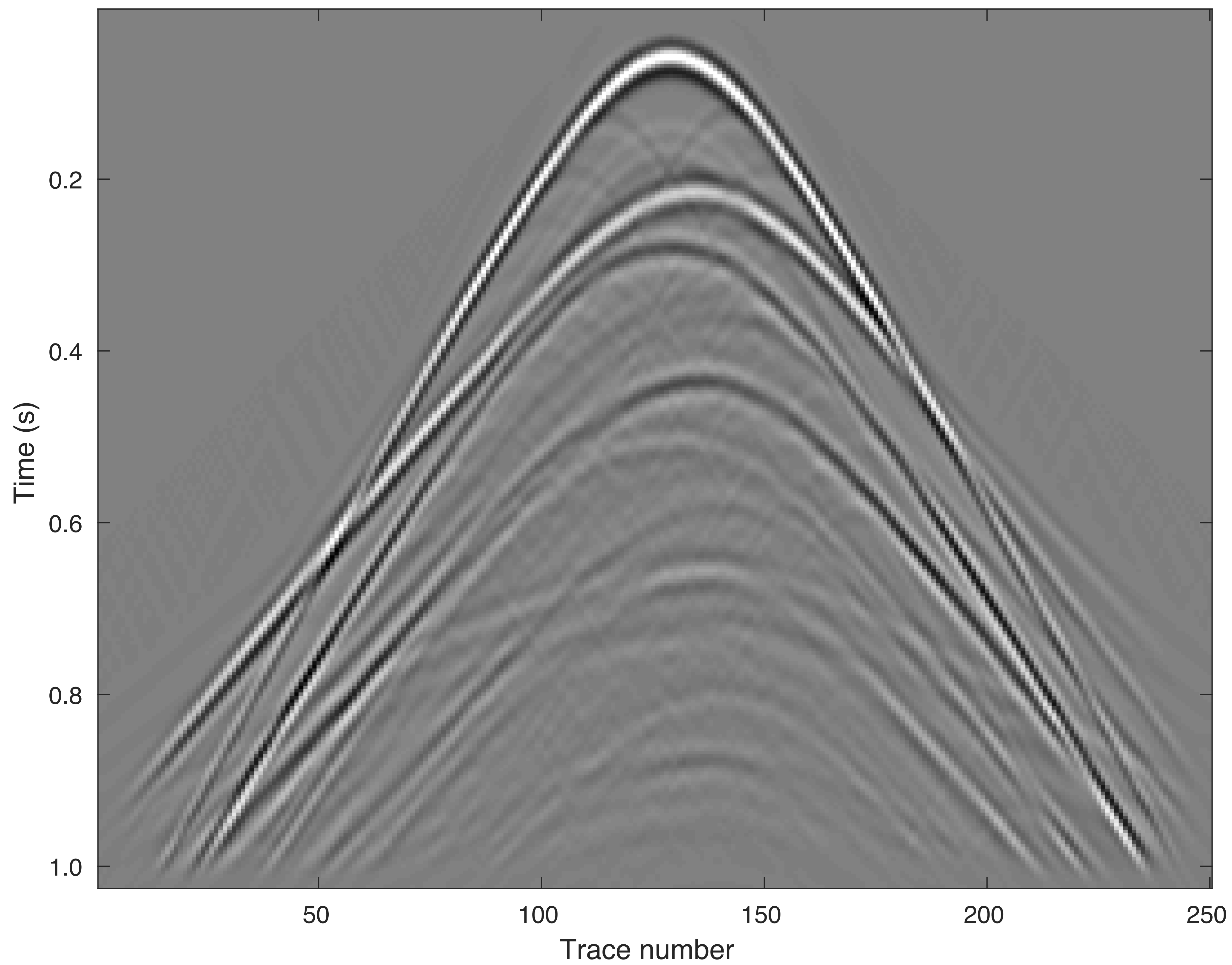


True signal

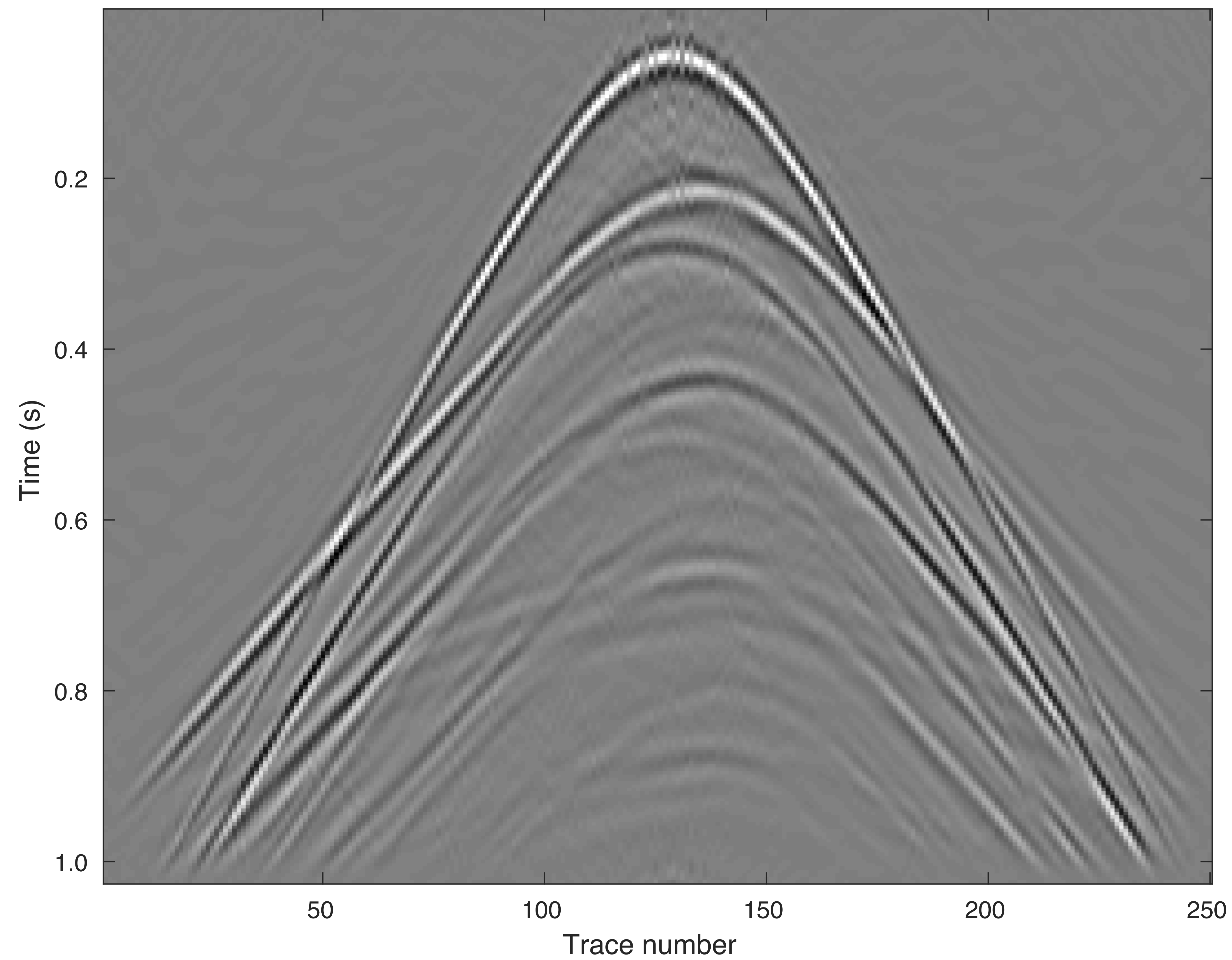


Input Data

Cosparsity VS Sparsity
50% Missing sources



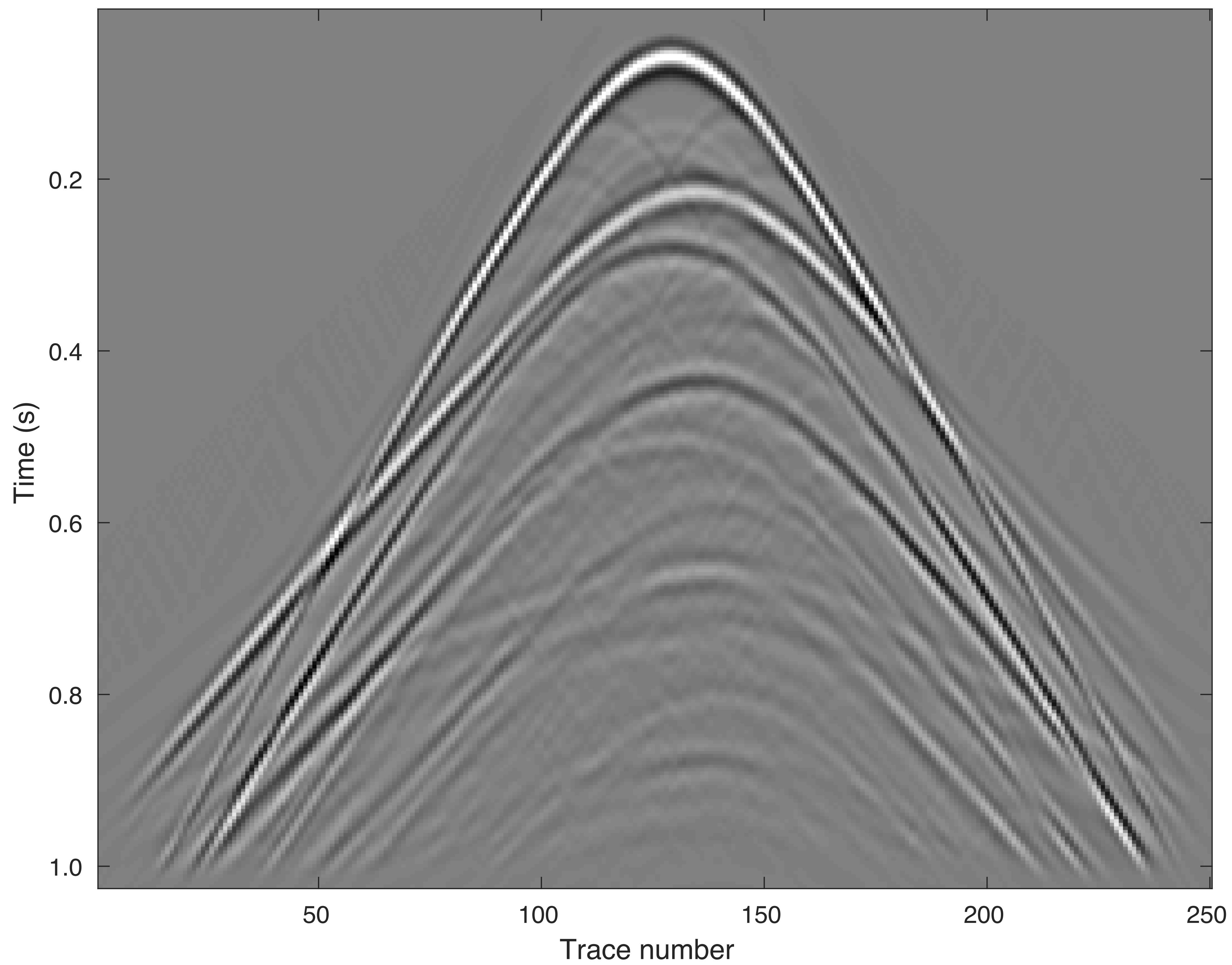
True signal



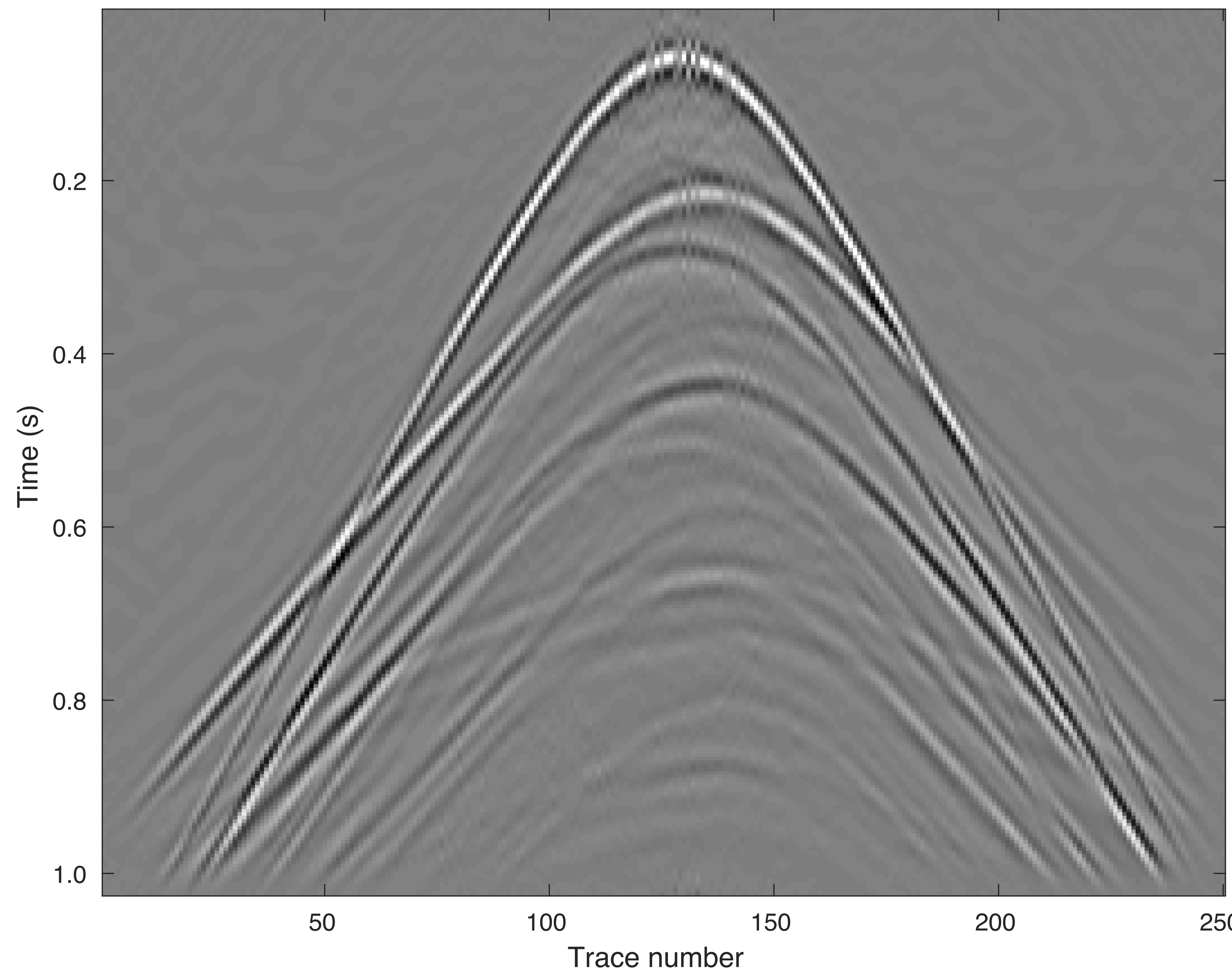
Synthesis L1 (SPGL1) - SNR 14.9 dB

Cosparsity VS Sparsity
50% Missing sources

Cai, Osher, Shen - 'Split Bregman Methods and Frame Based Image Restoration' (2009)

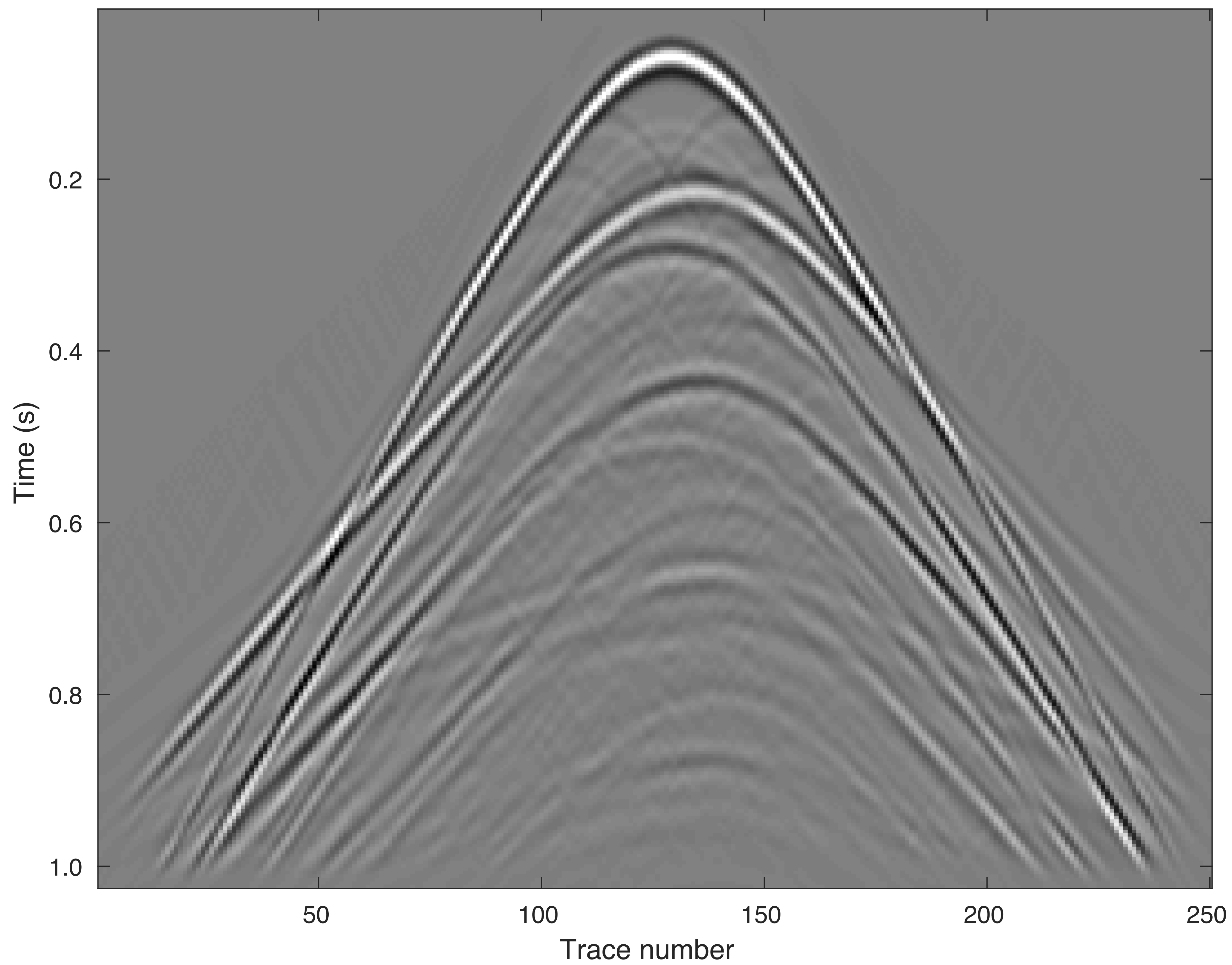


True signal

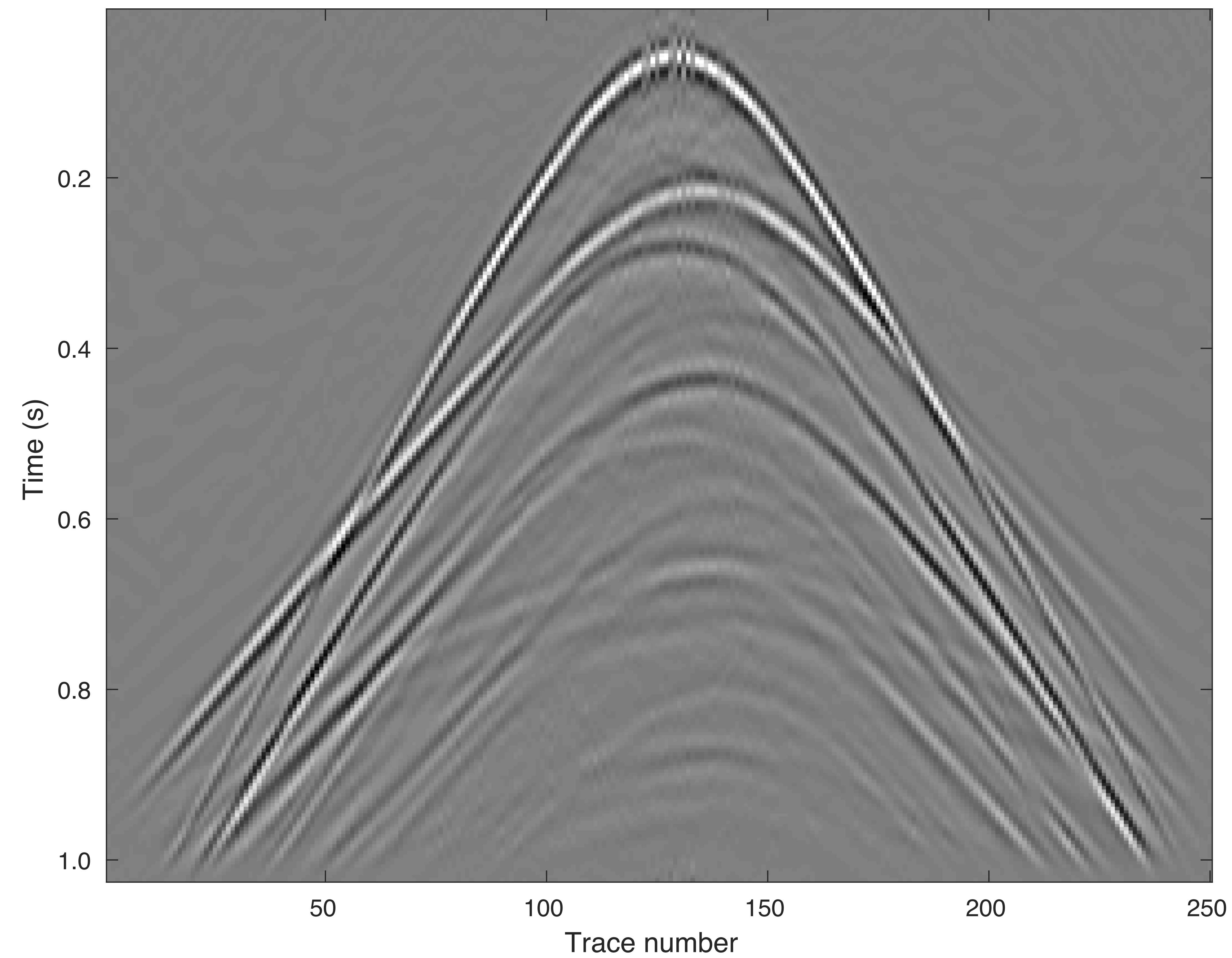


Analysis L1 (Linearized Bregman) - SNR 14.3 dB

Cosparsity VS Sparsity
50% Missing sources

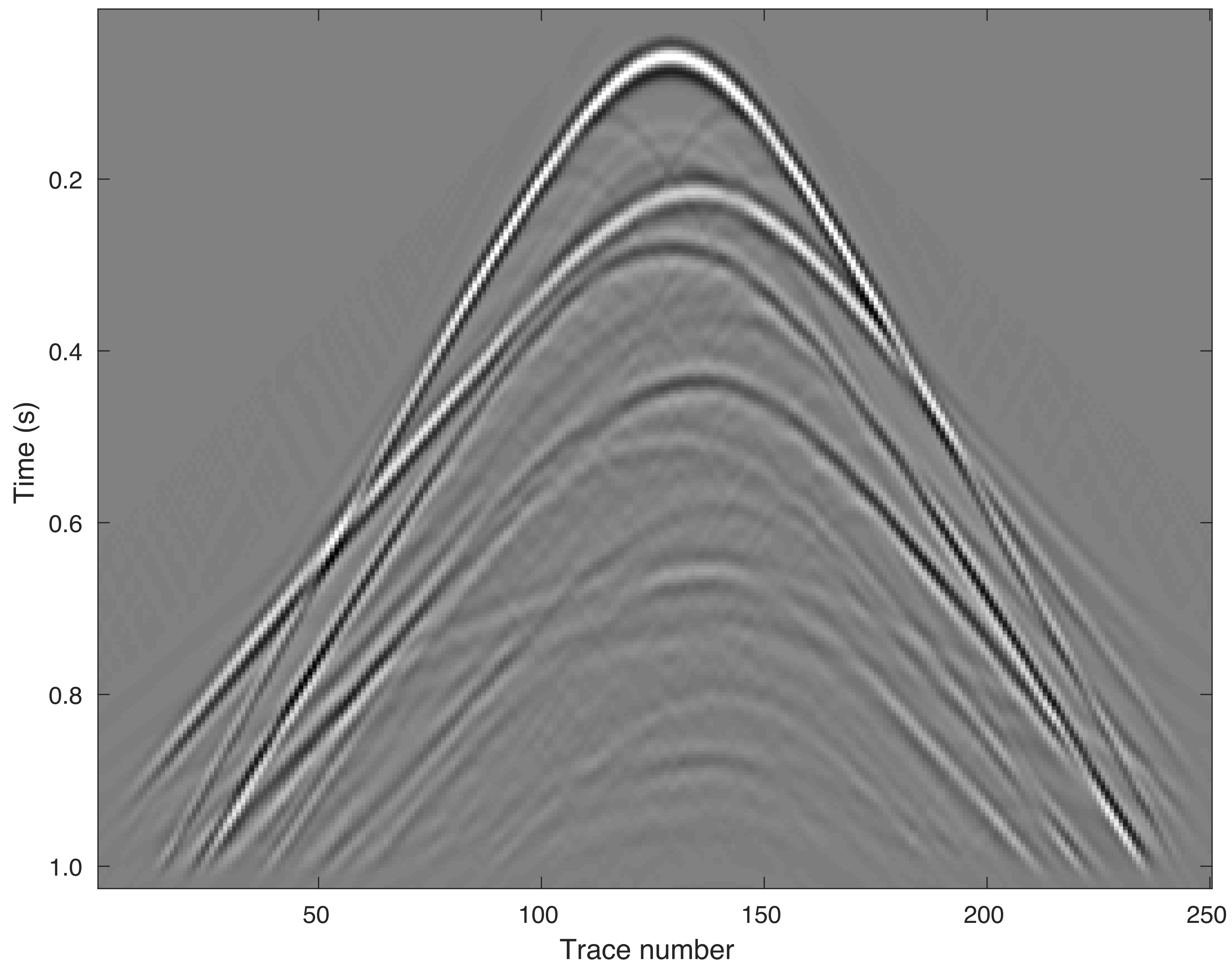


True signal

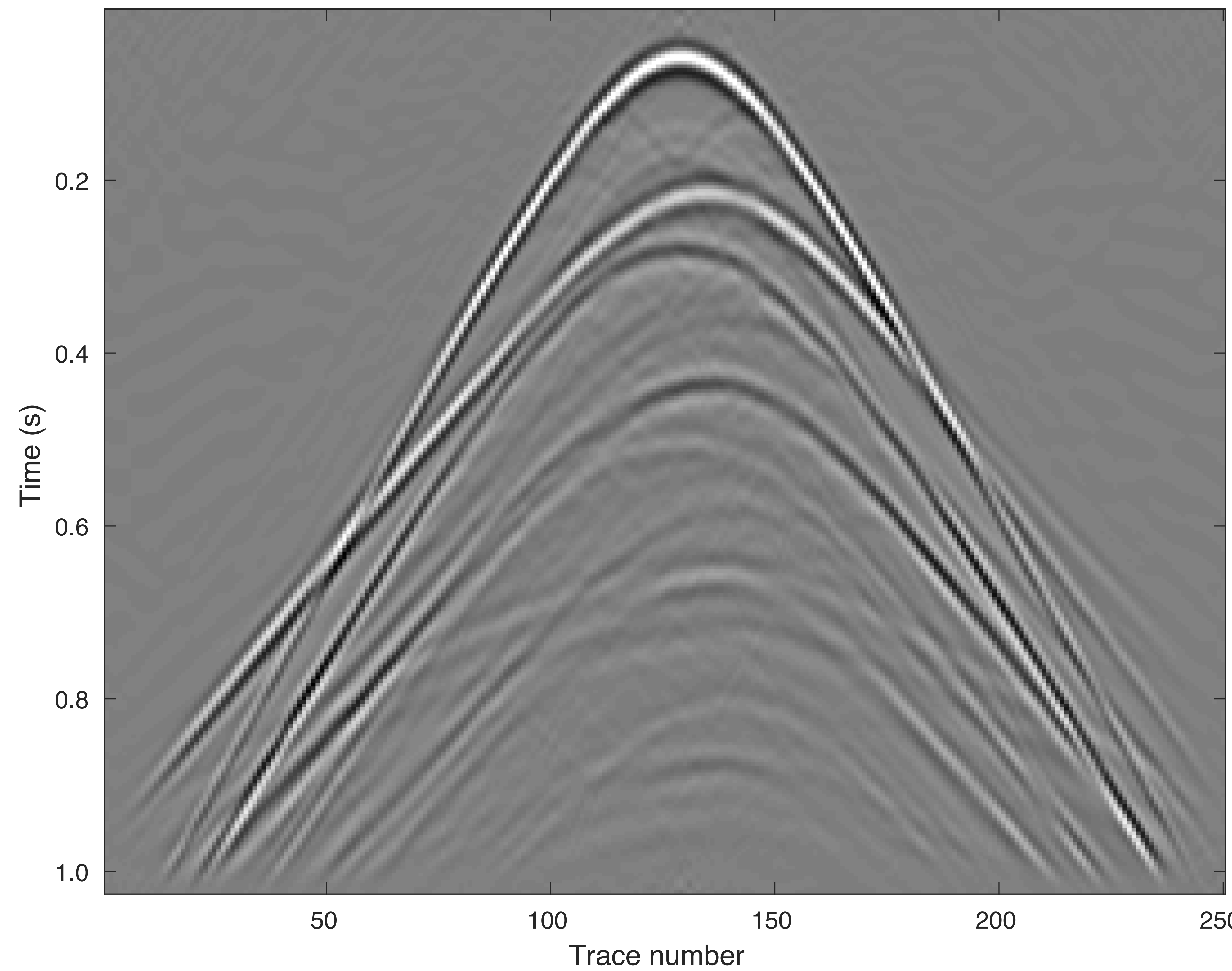


Analysis L1 (Ours) - SNR 15.0 dB

Cosparsity VS Sparsity
50% Missing sources



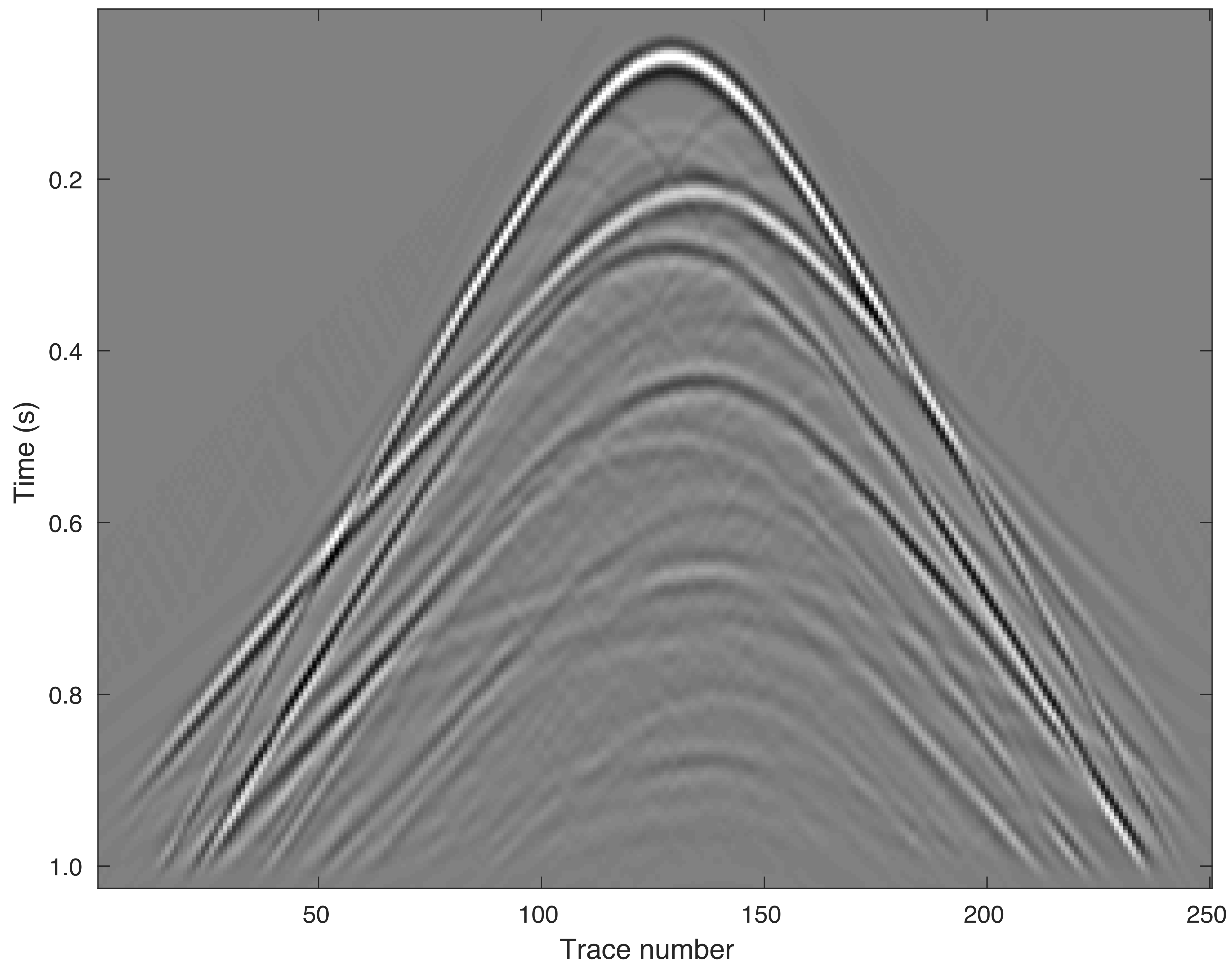
True signal



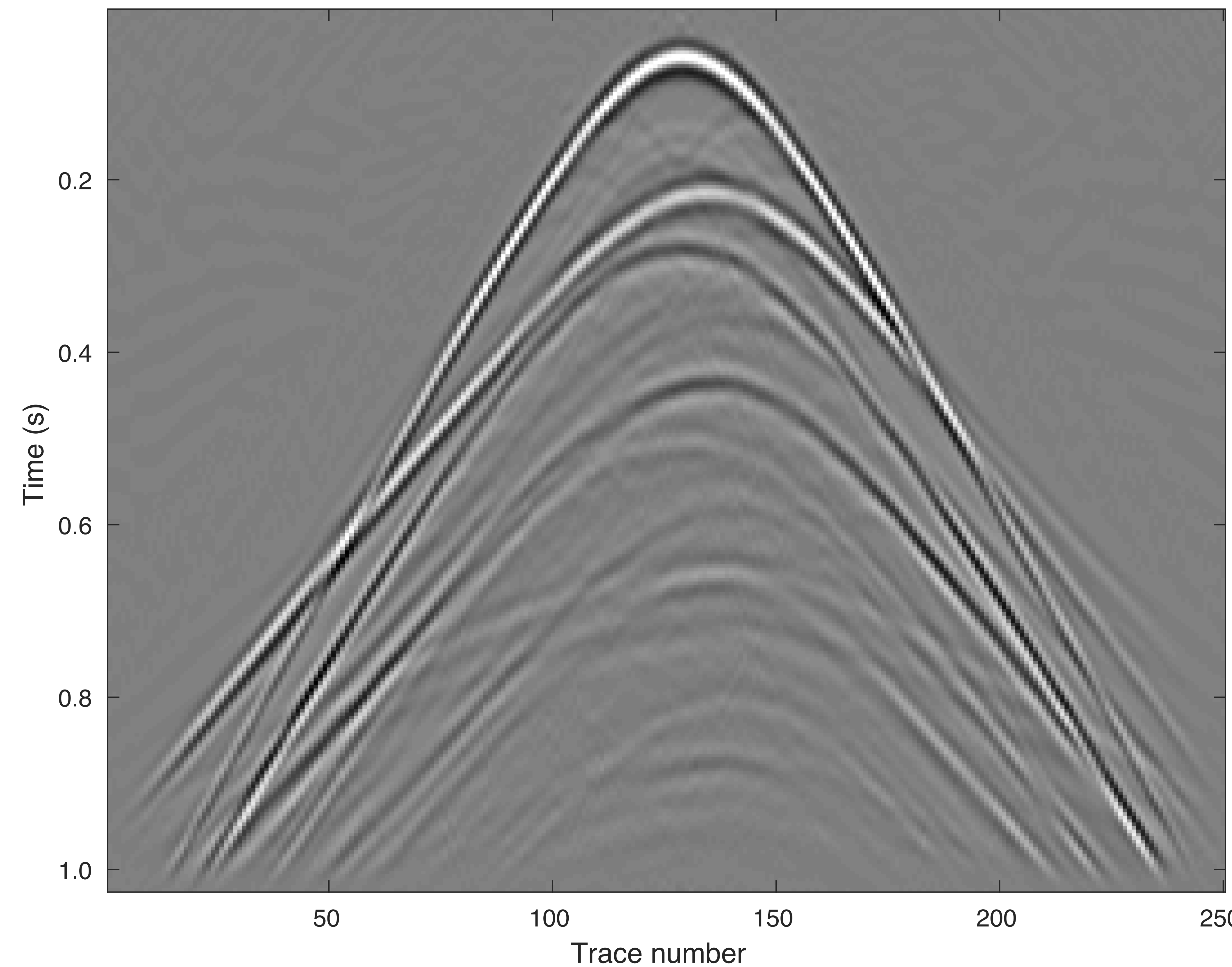
Analysis L0 (Ours) - SNR 23.7 dB

Cosparsity VS Sparsity
50% Missing sources

Nam, et. al., "The cosparse analysis model and algorithms" (2013)

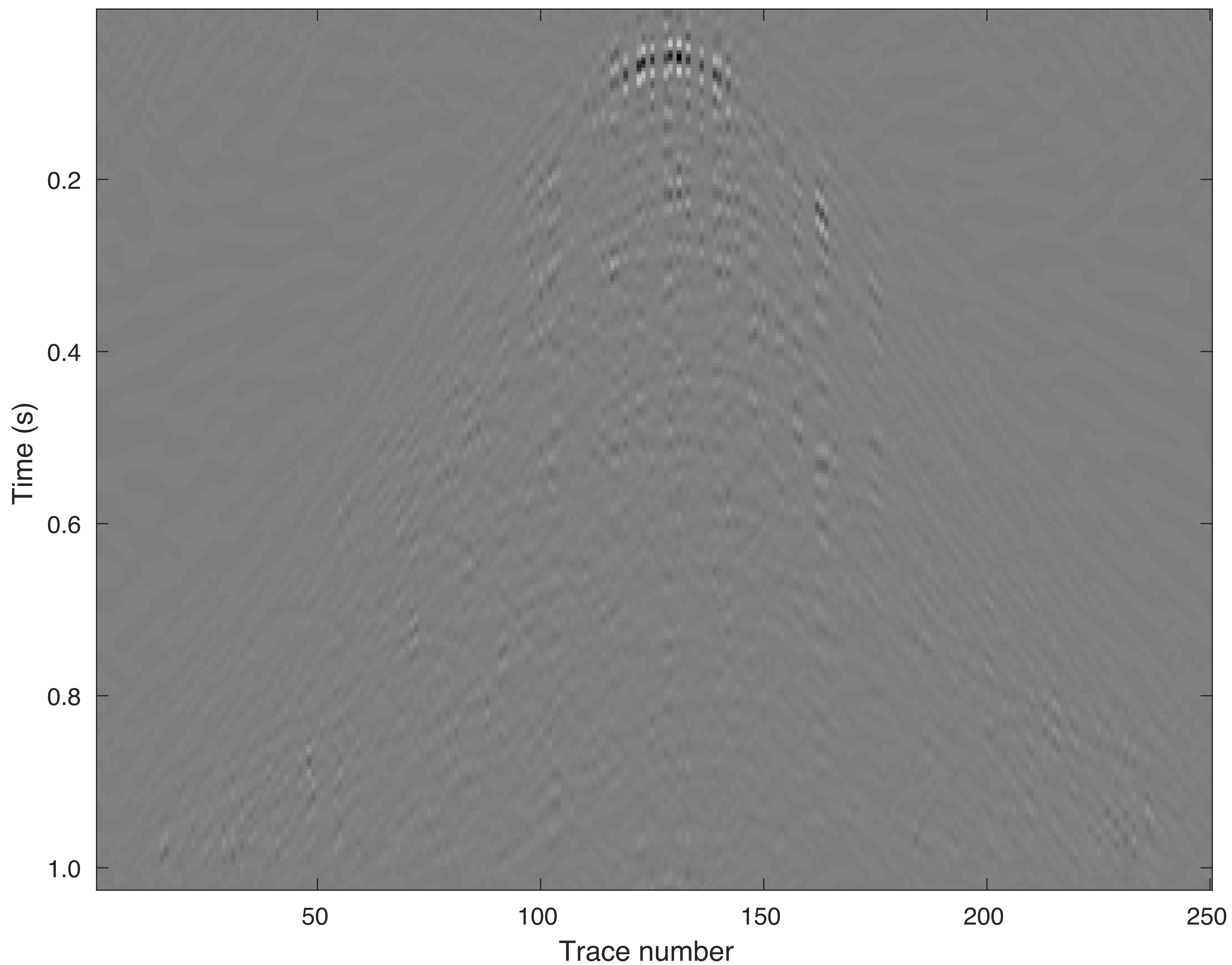


True signal

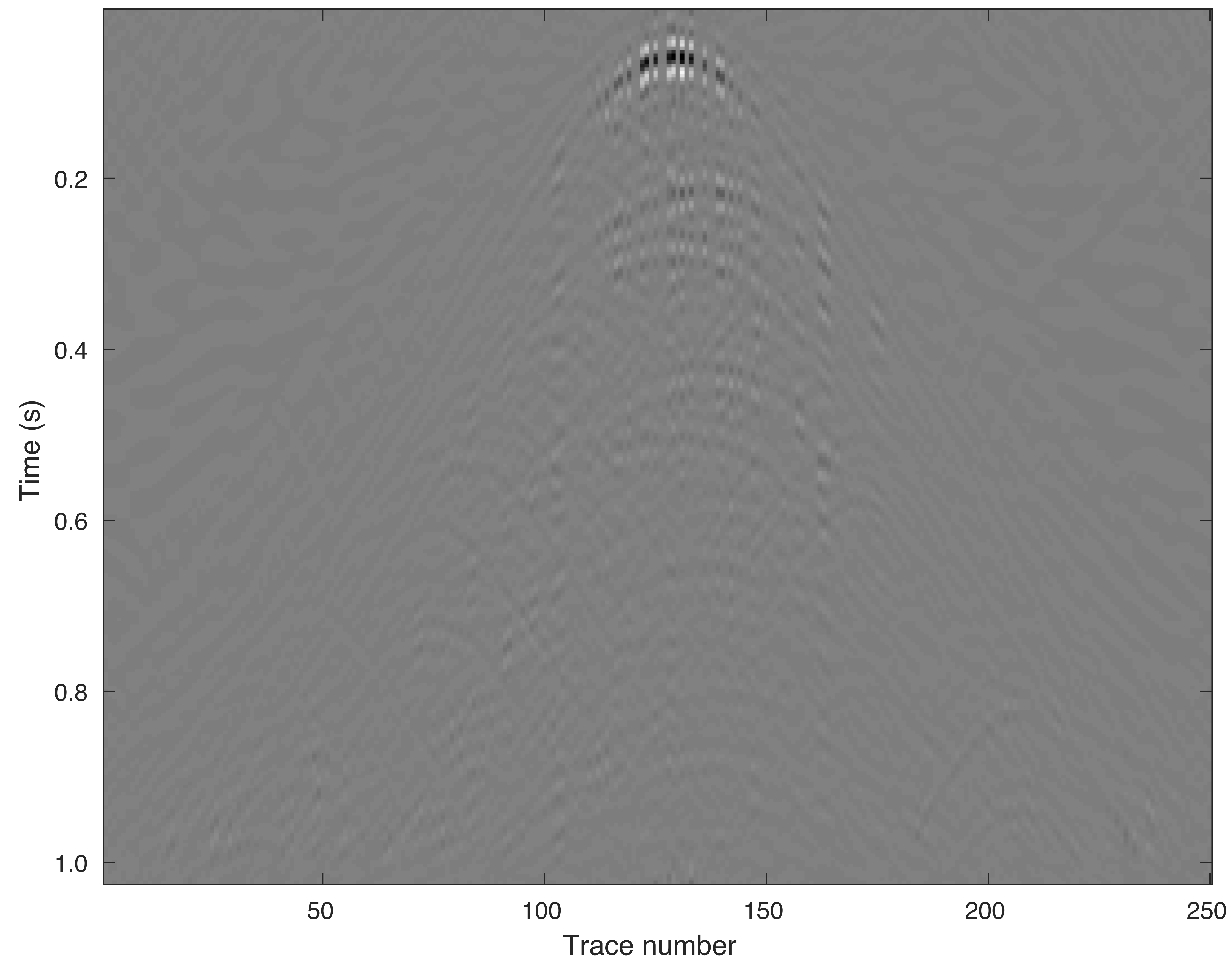


Analysis L0 - GAP - SNR 23.4 dB

Cosparsity VS Sparsity
50% Missing sources

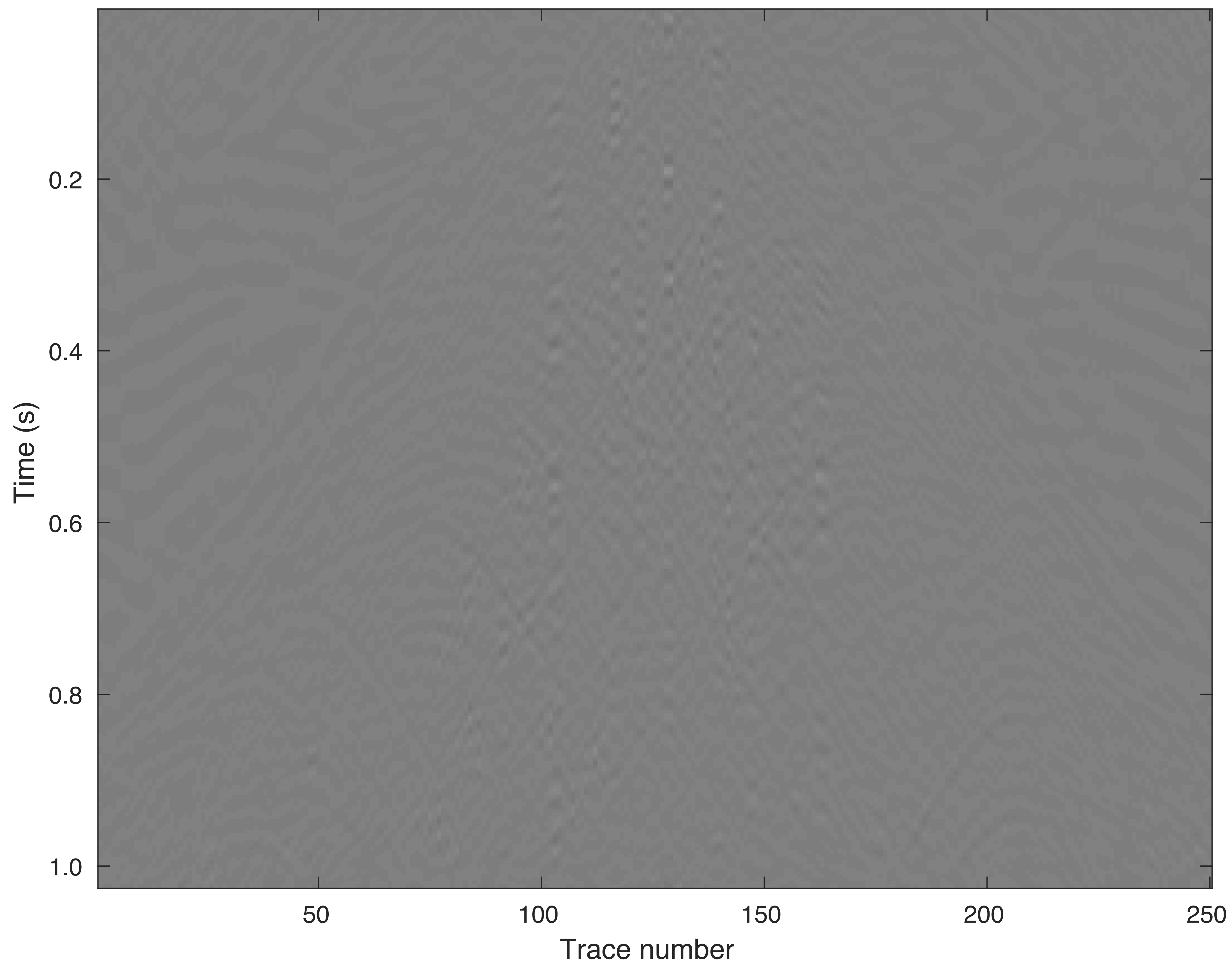


Difference - Synthesis L1

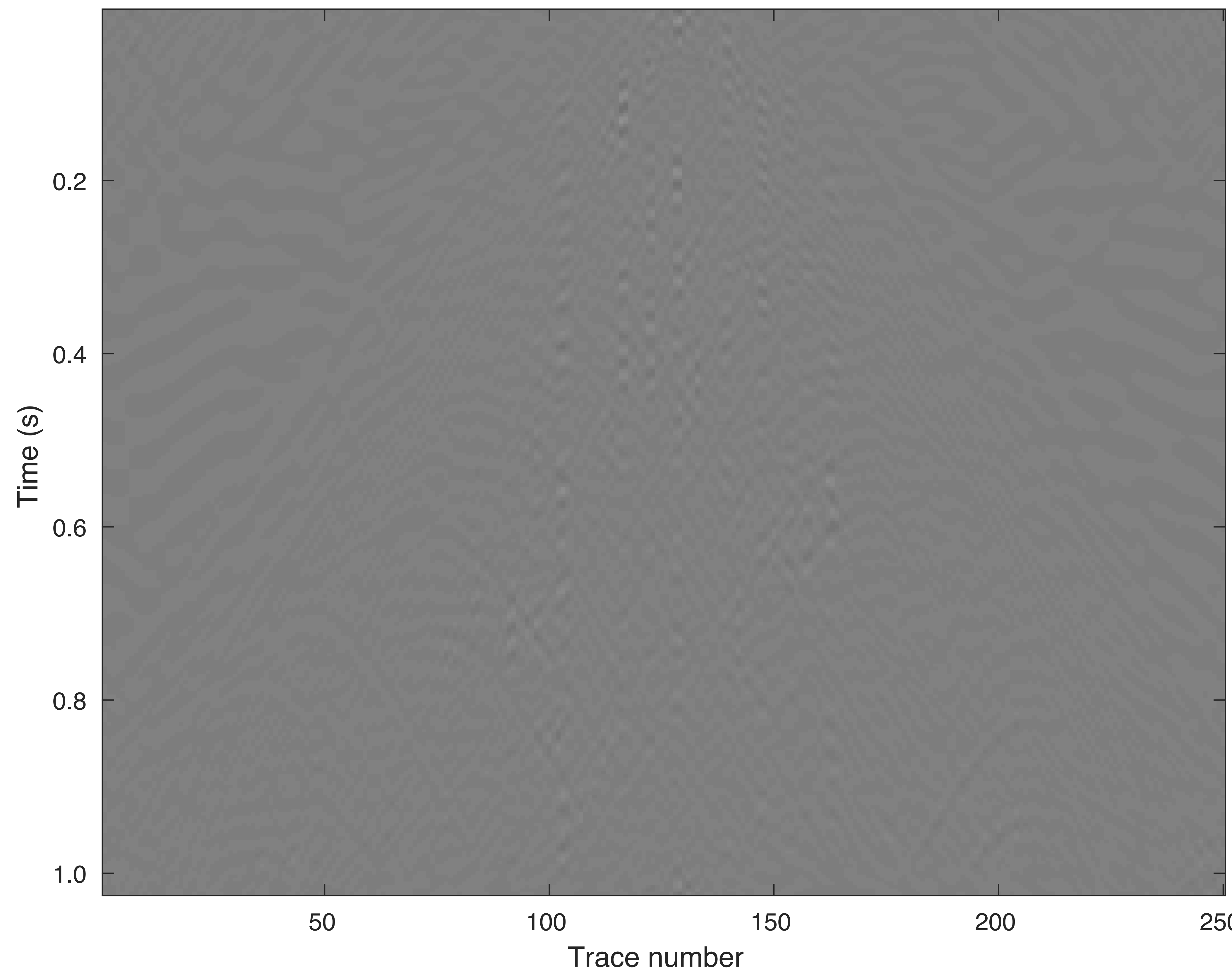


Difference - Analysis L1

Cosparsity VS Sparsity
50% Missing sources



Difference - GAP L0



Difference - Ours L0

L1 Summary

	SNR (dB)	Time (s)
Synthesis (SPGL1)	14.9	112
Analysis (Ours)	15.0	77.2
Linearized Bregman	14.3	188

L0 Summary

	SNR (dB)	Time (s)
Ours	23.7	75
GAP	23.4	118

Extensions

Can be easily extended to handle

- data-side noise
- signal-side constraints

Applications - Robust Tensor Completion

$$\min_x \|\mathcal{A}\phi(x) - b\|_1$$

$\mathcal{A}$ - subsampling operator

$\phi(x)$ - mapping from tensor parameters -> full tensor

b - measured data contaminated by impulsive noise

Example: Robust Tensor Completion

$$\min_x \|\mathcal{A}\phi(x) - b\|_1$$

$$\downarrow$$

$$v(\tau) = \min_x \frac{1}{2}\|\mathcal{A}(\phi(x)) - b - r(x)\|_2^2$$

$$r(x) = \arg\min_r \frac{1}{2}\|\mathcal{A}(\phi(x)) - b - r\|_2^2$$

$$\text{s.t.} \|r\|_1 \leq \tau$$

Example:

BG Data Set

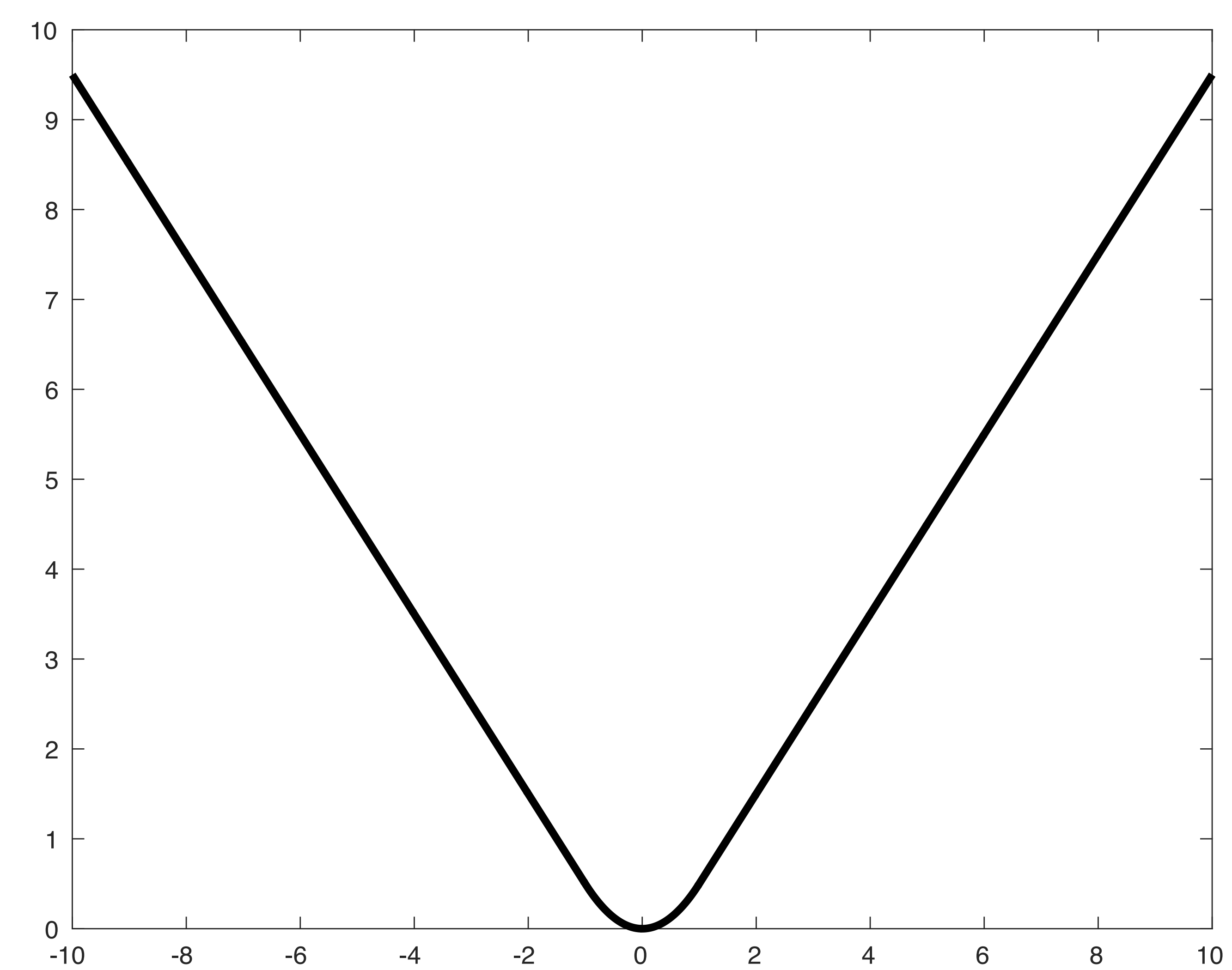
- 68 x 68 sources on a 150m grid, 201 x 201 receivers on a 50m grid, ocean bottom setup
- 75% receivers decimated randomly
- 5% of remaining receivers corrupted with noise = energy of decimated signal
- Hierarchical Tucker interpolation with previous L1 formulation

Example:

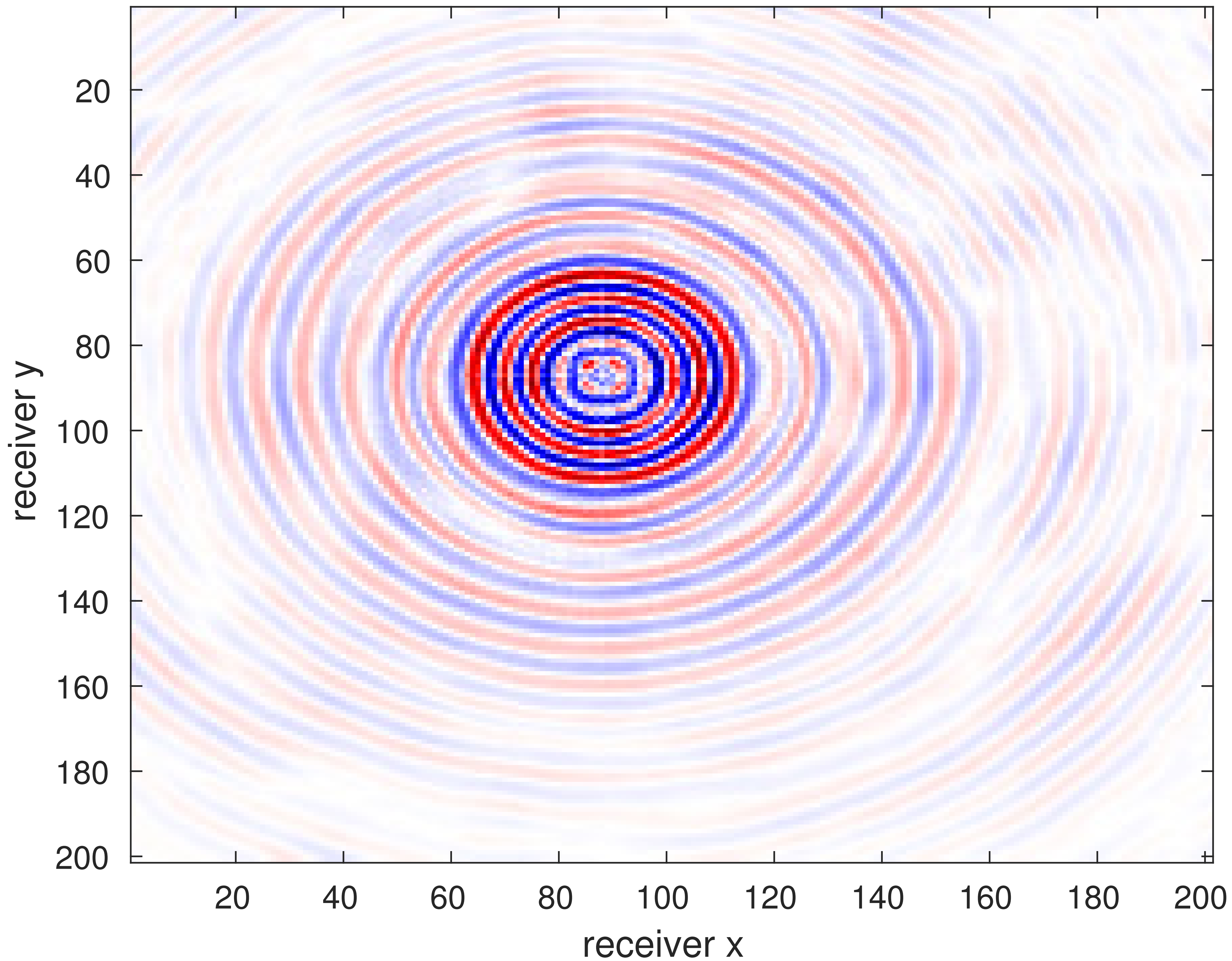
We compare to

- L2 misfit - original HT tensor completion
- Huber misfit - smoothed L1

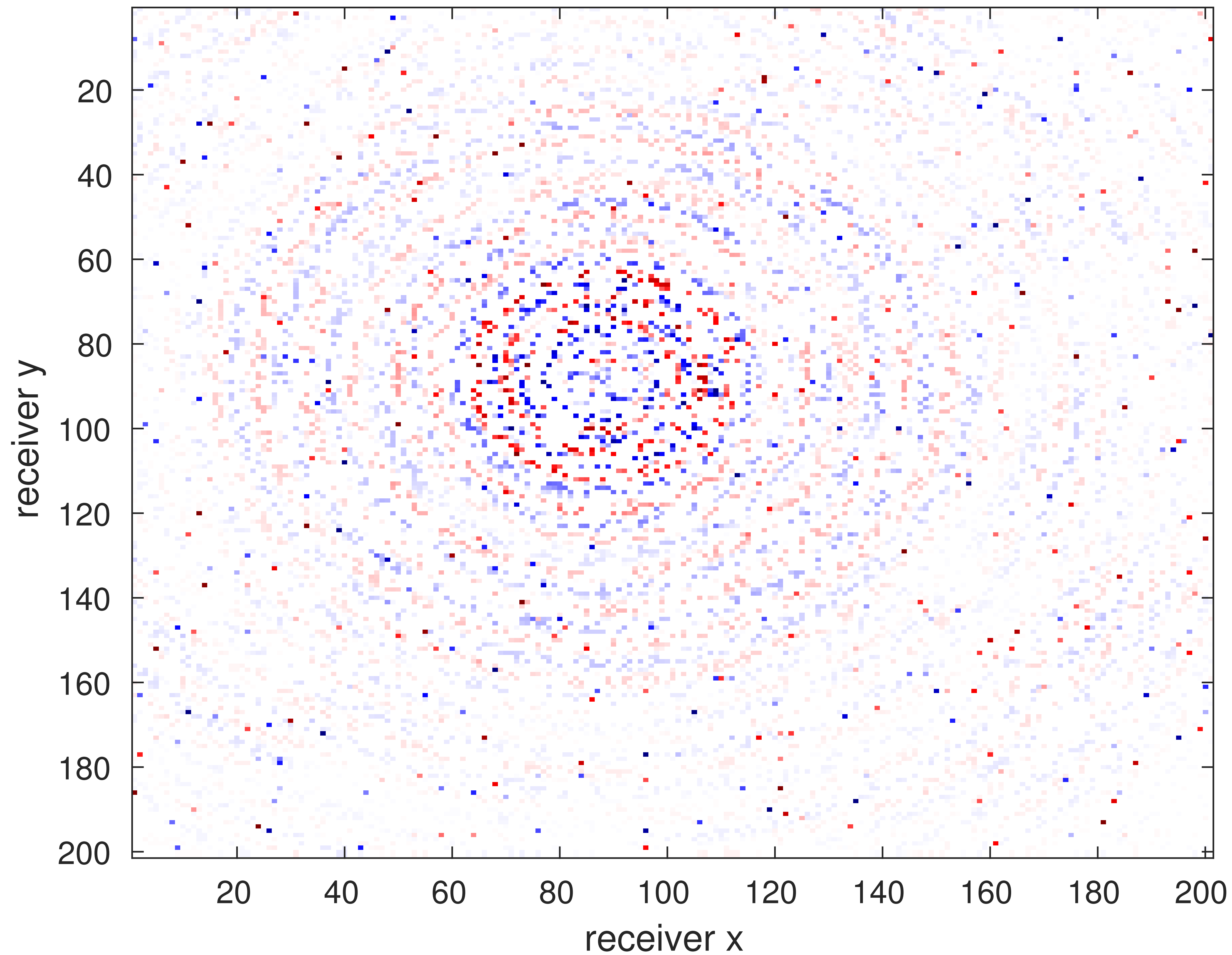
$$H_\delta(x) = \begin{cases} x^2 & \text{if } |x| \leq \delta \\ 2\delta|x| - \delta^2 & \text{if } |x| \geq \delta \end{cases}$$



Robust Tensor Completion
75% Missing Receivers with 5% impulsive noise

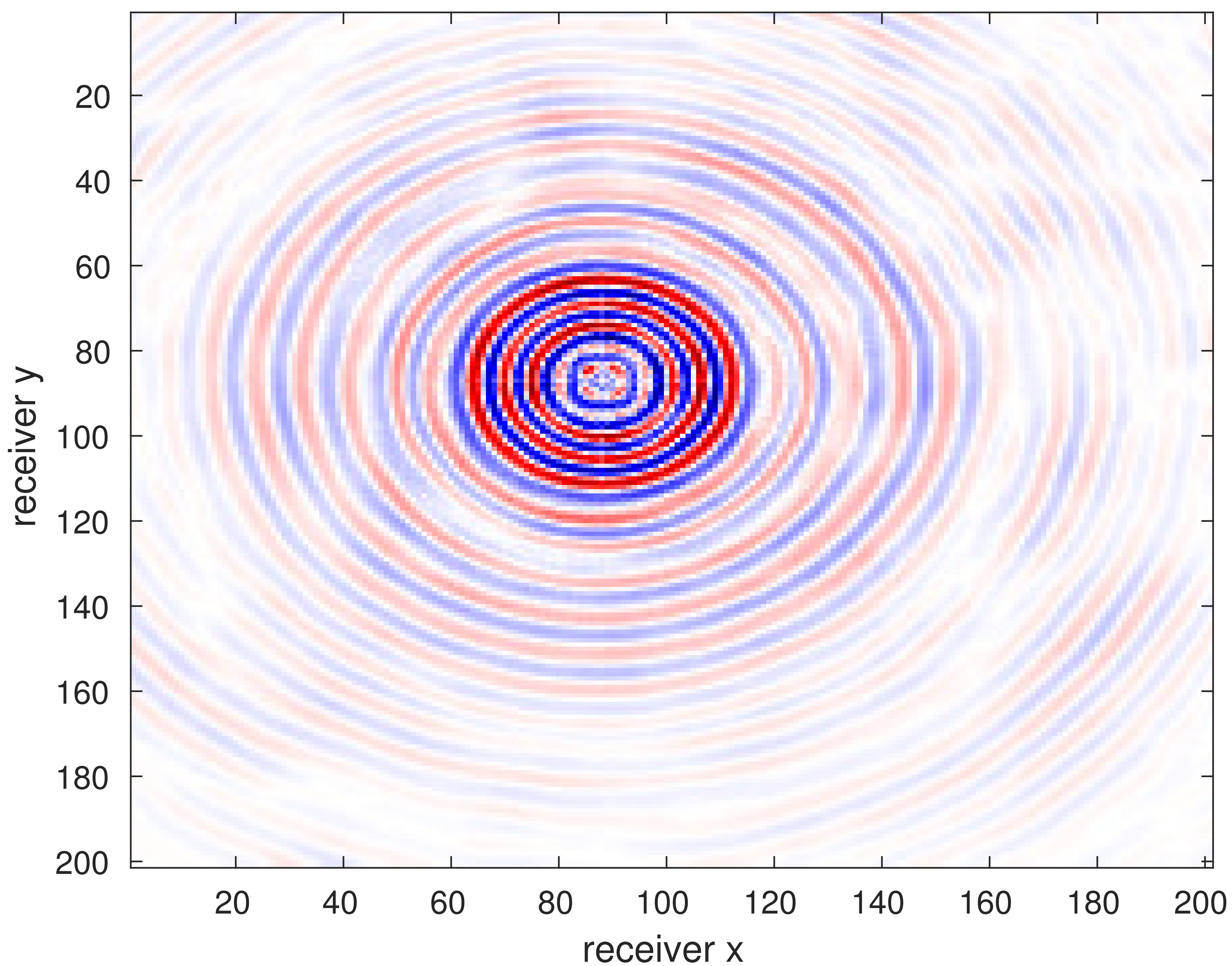


True Data

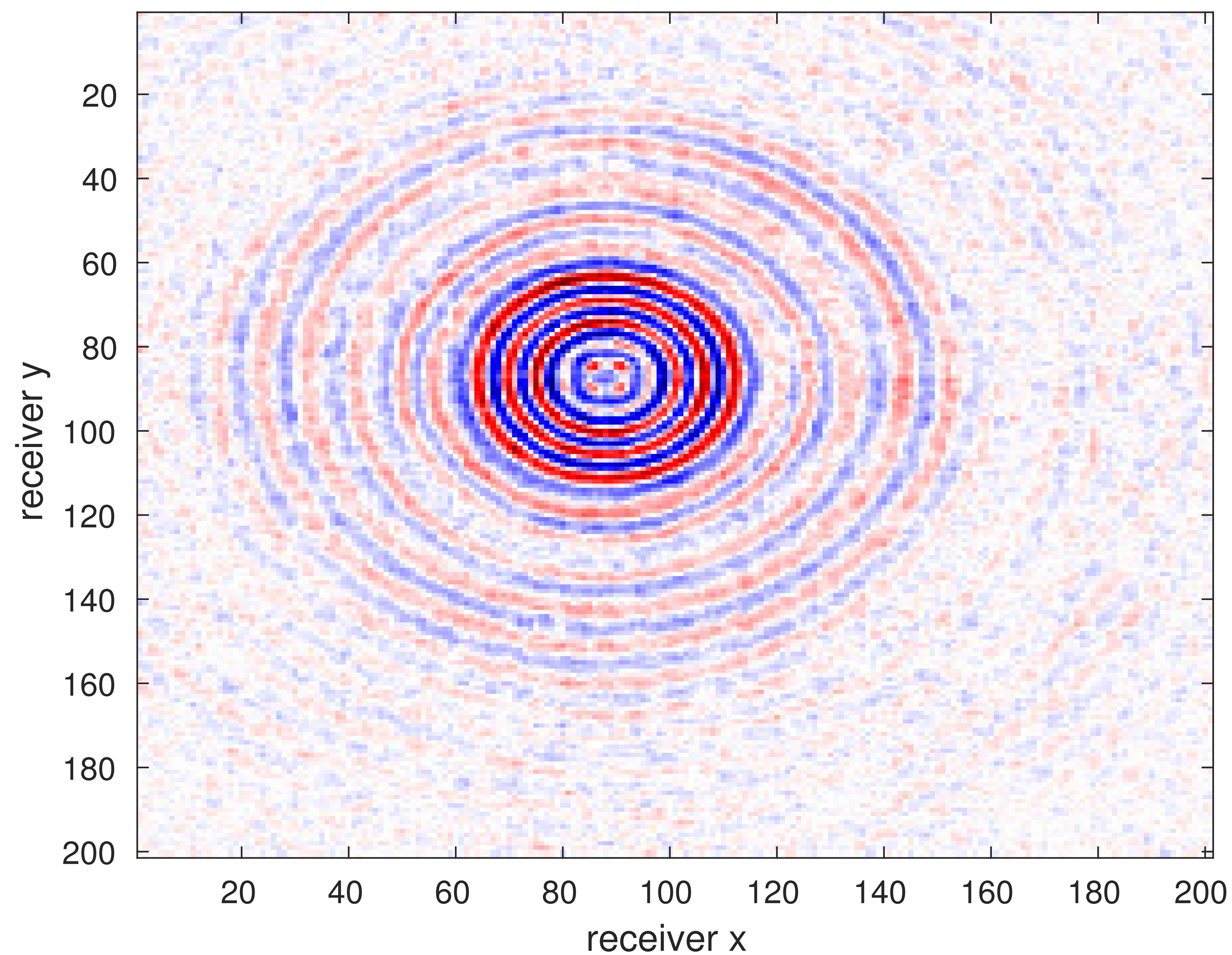


Input Data - SNR 0dB

Robust Tensor Completion
75% Missing Receivers with 5% impulsive noise

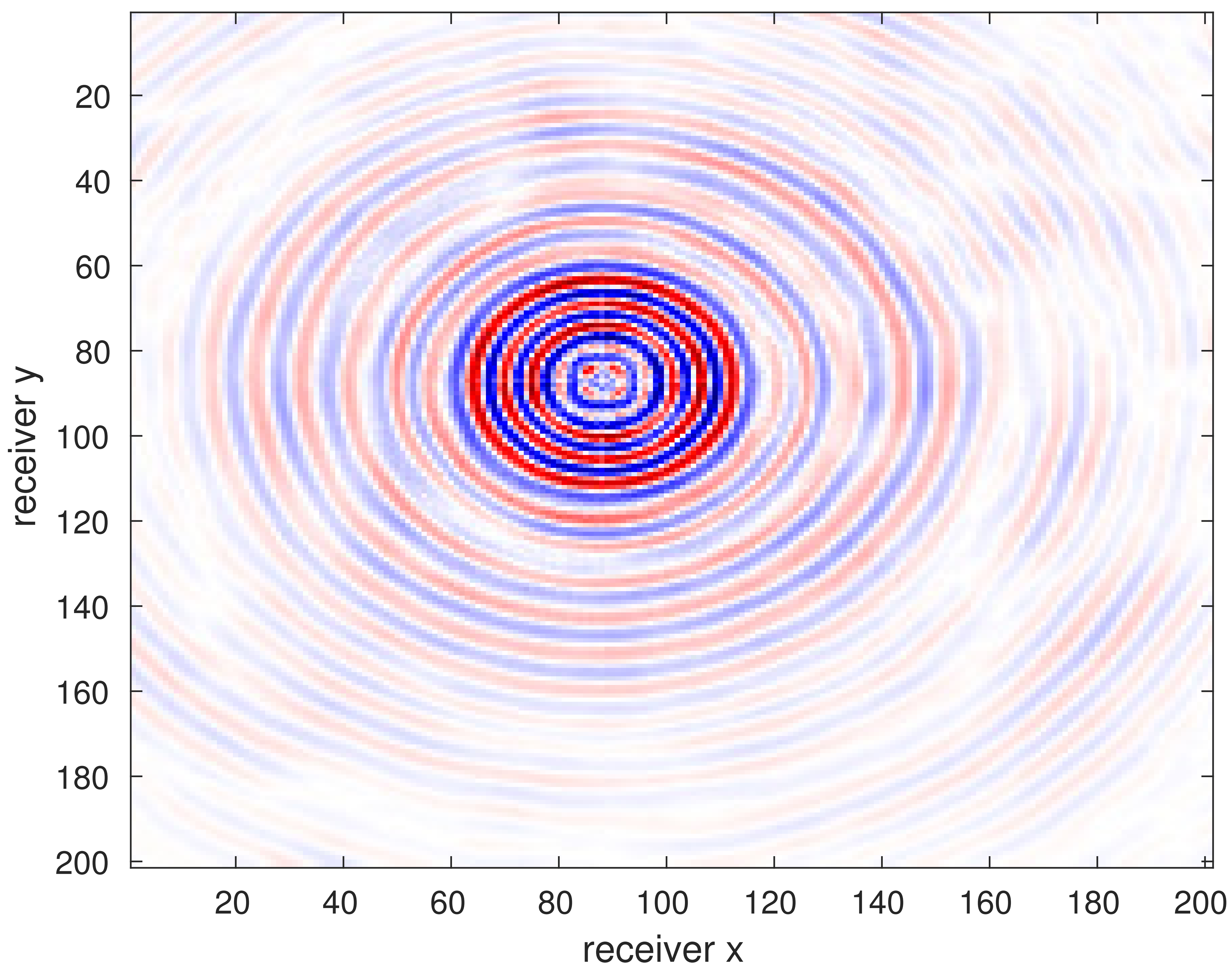


True Data

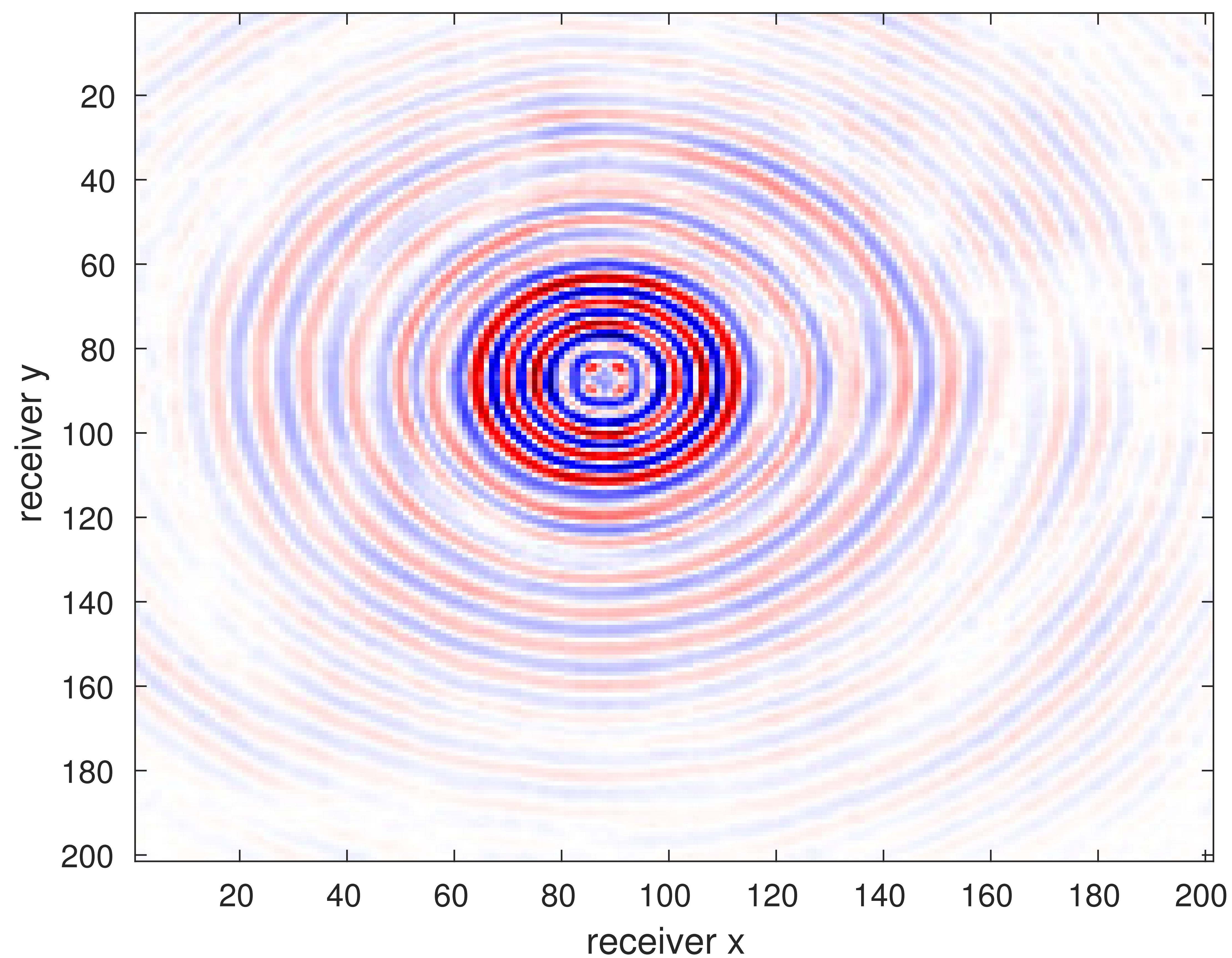


L2 norm - SNR 8.8 dB

Robust Tensor Completion
75% Missing Receivers with 5% impulsive noise

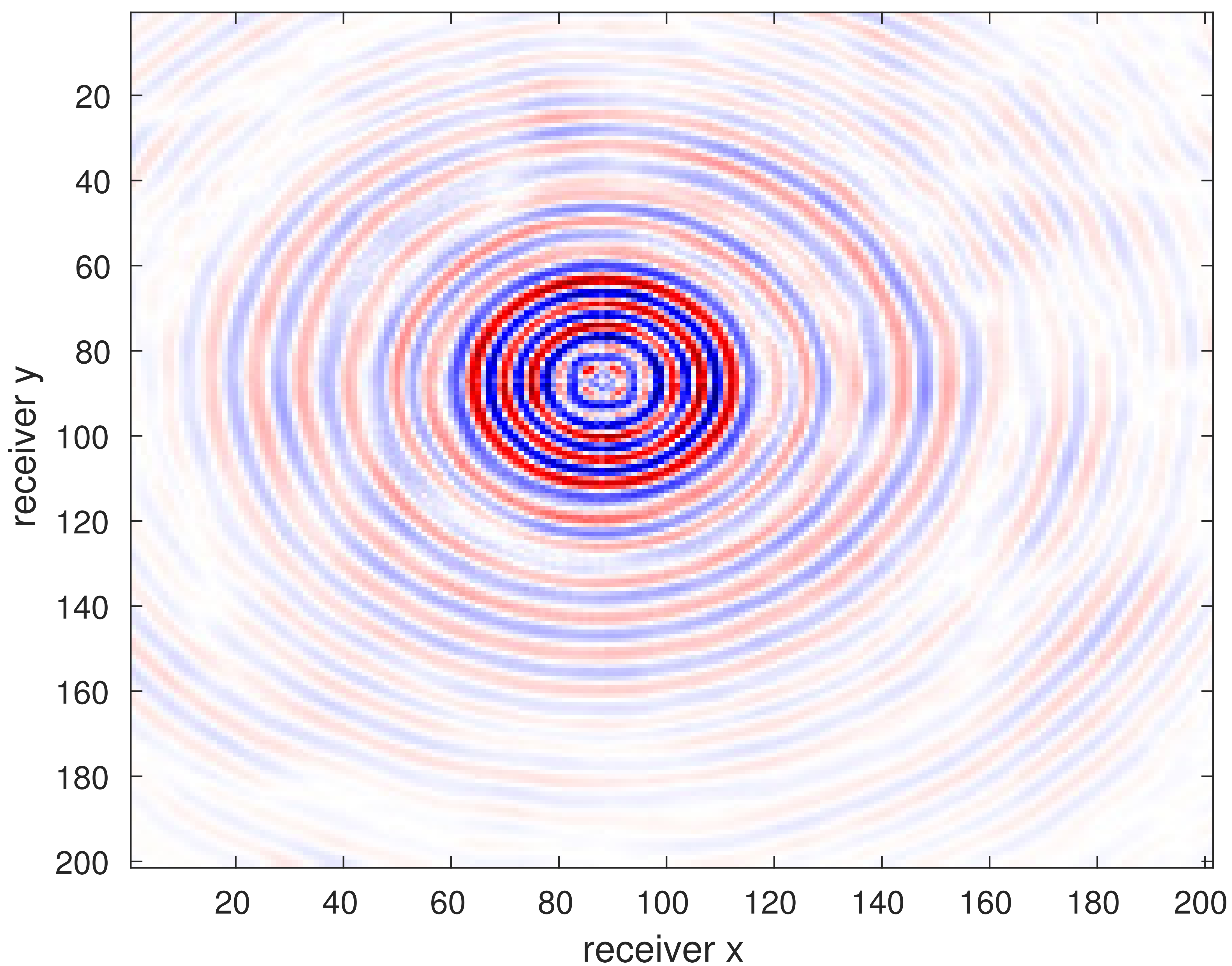


True Data

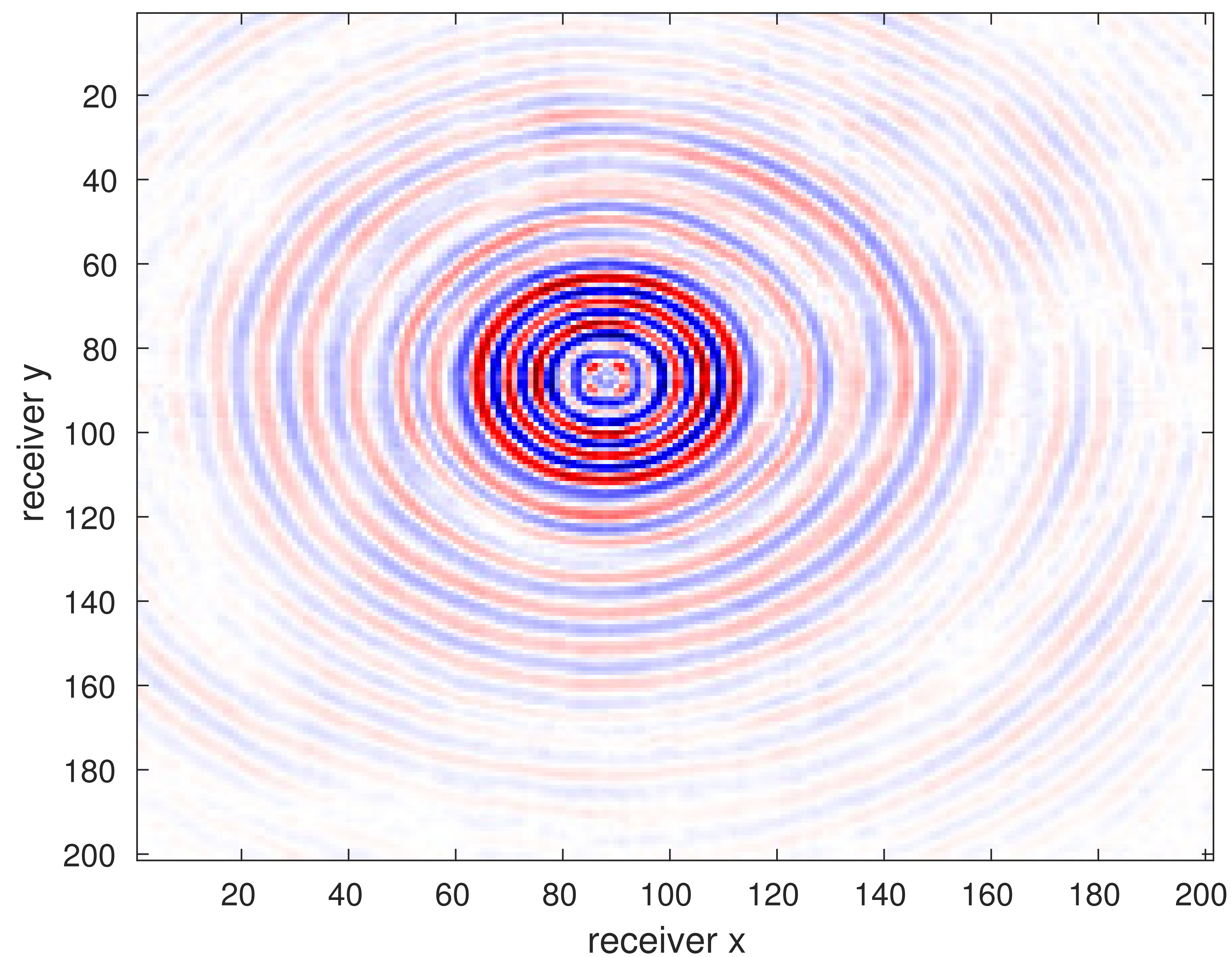


L1 norm - SNR 16.8 dB

Robust Tensor Completion
75% Missing Receivers with 5% impulsive noise

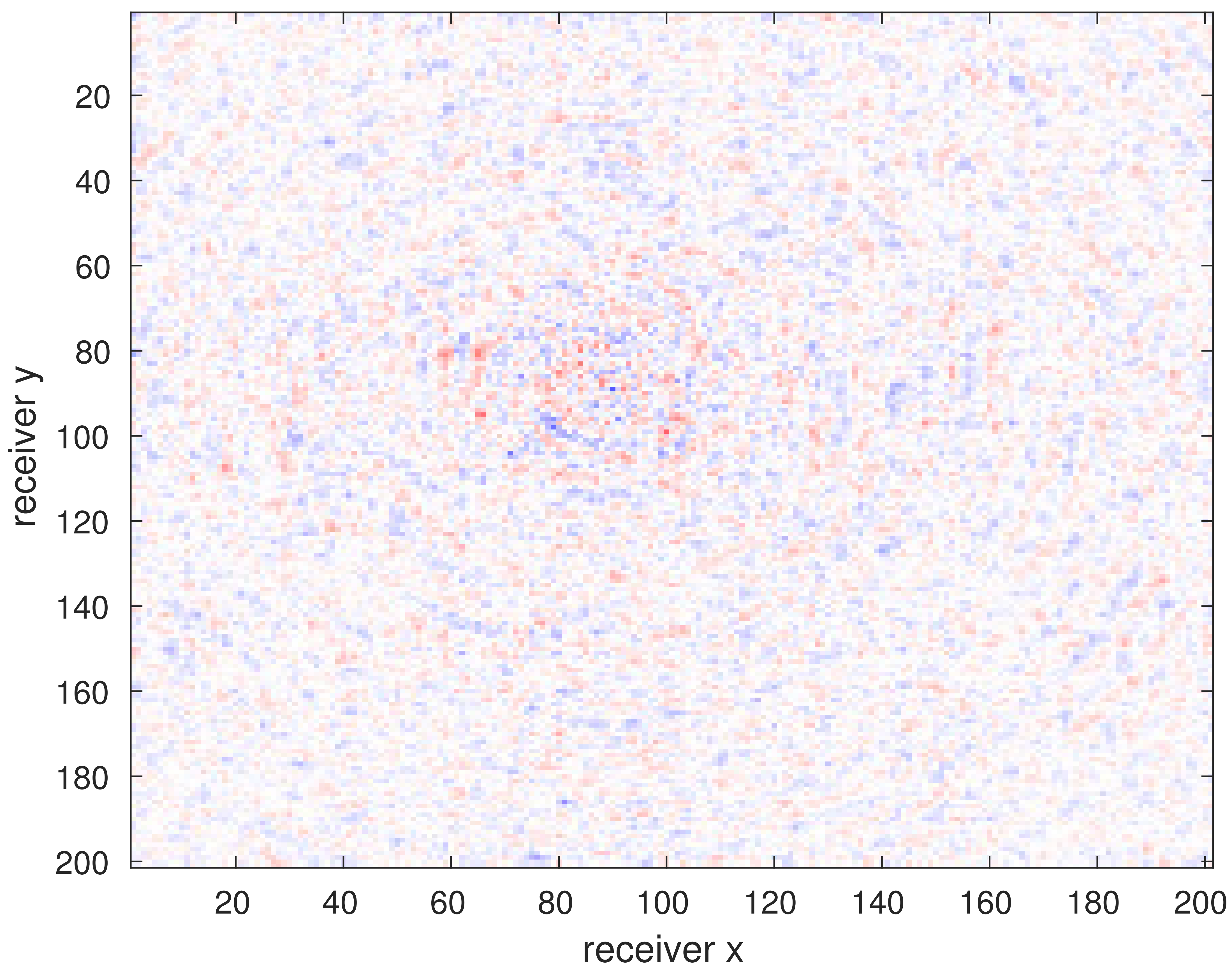


True Data

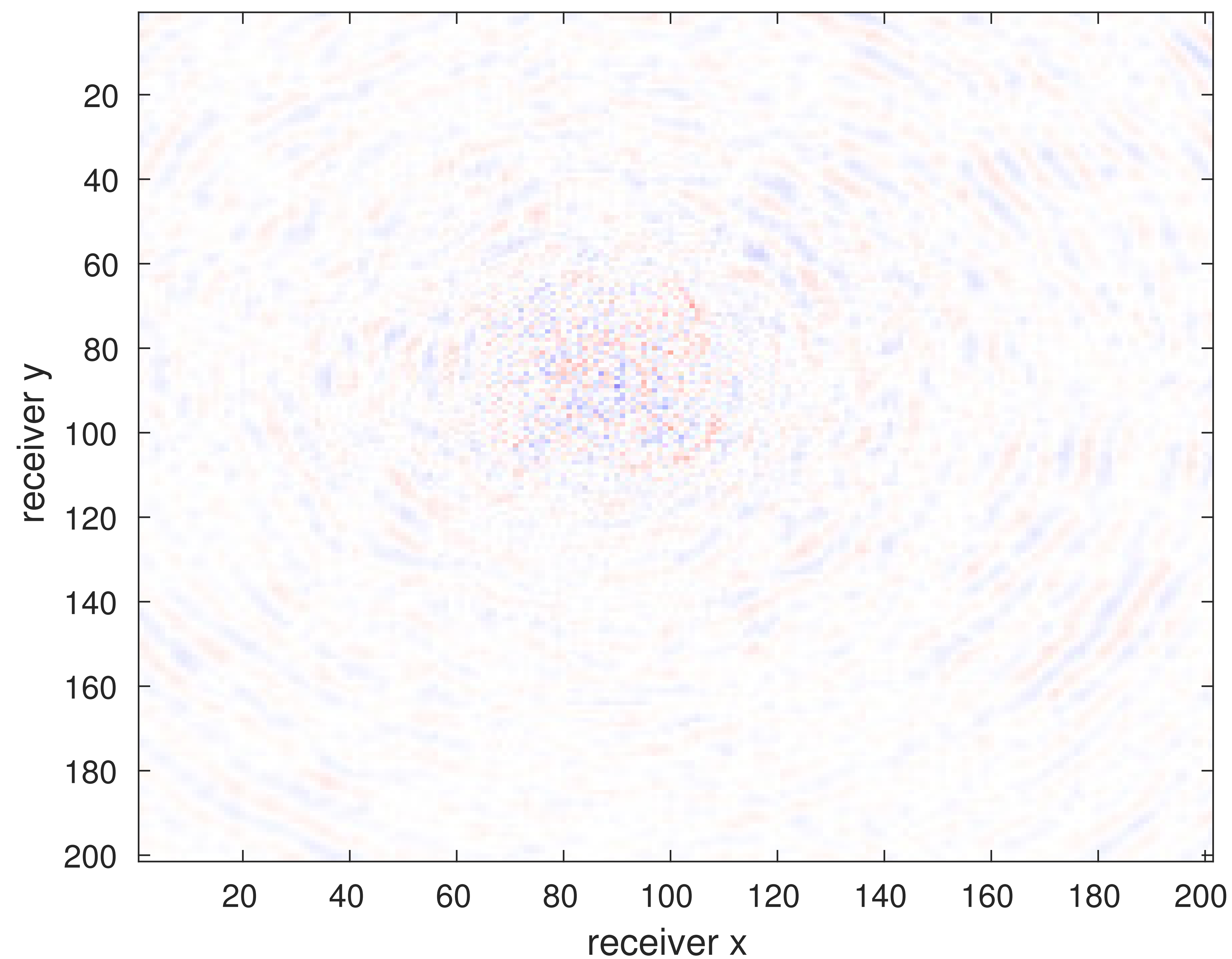


Huber penalty - best parameter - SNR 16.7 dB

Robust Tensor Completion
75% Missing Receivers with 5% impulsive noise



Difference - L2



Difference - L1

Robust Tensor Completion

	Recovery SNR (dB)	Time (s)
$\ell 2$	7.68	632
$\ell 1$	16.2	1072
Huber - best δ	15.9	1003

Huber performance versus δ

	Recovery SNR (dB)	Time (s)
$5 \cdot 10^{-6}$	13.4	1578
$5 \cdot 10^{-5}$	15.9	1003
$5 \cdot 10^{-4}$	8.32	928

SLIM

Summary

Solve problems of the form

$$\min_x h(g(x))$$

with

$h(x)$ - is convex, non-smooth, has an easy projection

$g(x)$ - is a smooth mapping

With level set methods (SPGL1 trick) + variable projection

Summary

Applications to seismic data processing

- Cosparsity-based interpolation
- Robust tensor completion

Competitive compared to other existing sparsity-based methods

SLIM

Future Work

Extensions to non-convex problems

- Full waveform inversion

More seismic examples

- Source localization
- 4D seismic imaging

Acknowledgements

This research was carried out as part of the SINBAD project with the support of the member organizations of the SINBAD Consortium.

SINBAD