Wavefield reconstruction via randomized sampling

Navid Ghadermarzy

SINBAD Consortium Meeting, December 8 2014

Outline

- We do seismic trace interpolation by transforming the data to the curvelet domain.
- This problem has a lot of room for improvement.
- In this talk we consider of a few simple techniques that can improve the results significantly.

Outline

- ℓ_1 recovery in curvelet domain

Outline

- ℓ_1 recovery in curvelet domain
- Weighting

Outline

- ℓ_1 recovery in curvelet domain
- Weighting
- Jittered subsampling

Outline

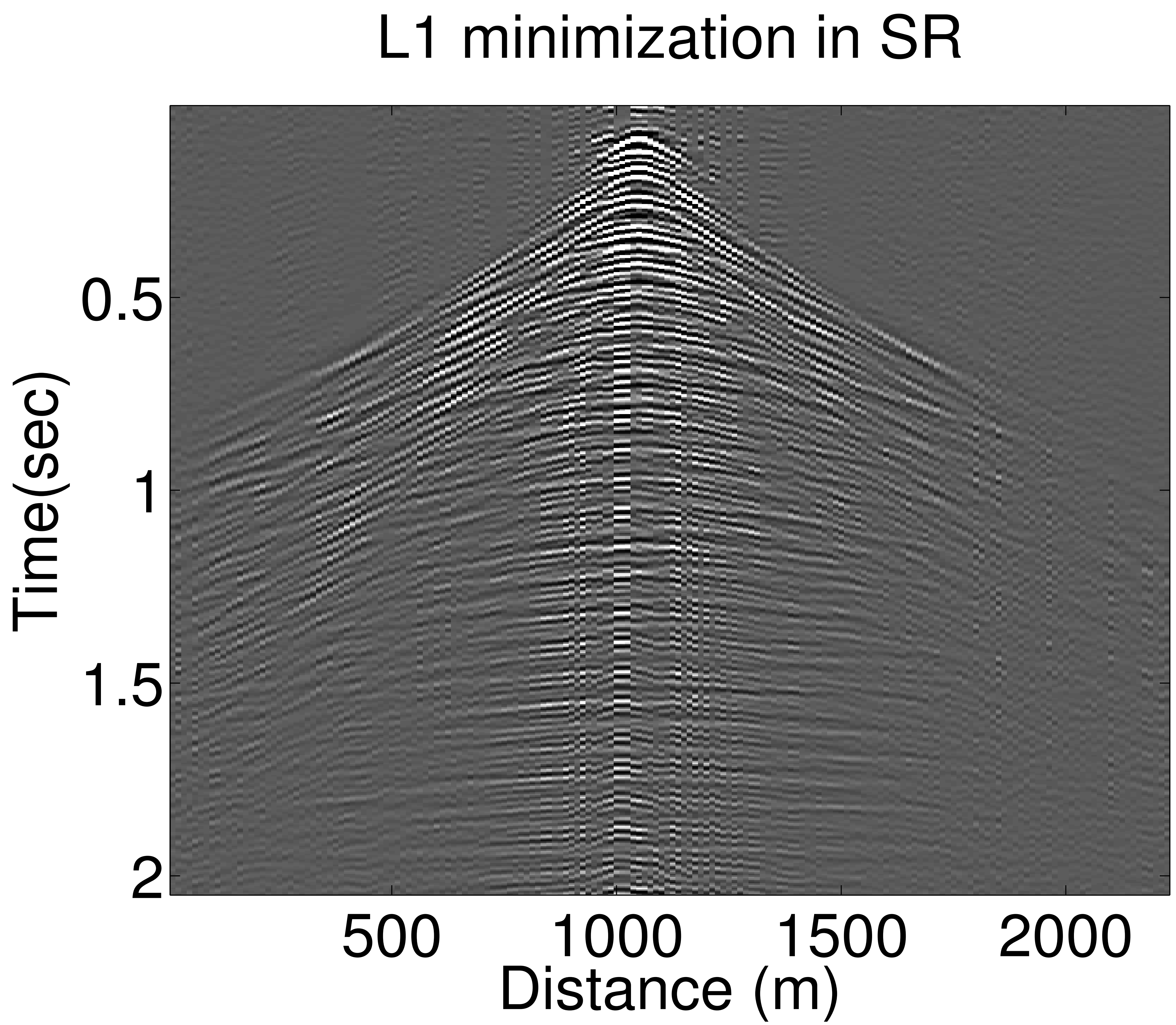
- ℓ_1 recovery in curvelet domain
- Weighting
- Jittered subsampling
- Midpoint-offset domain

Outline

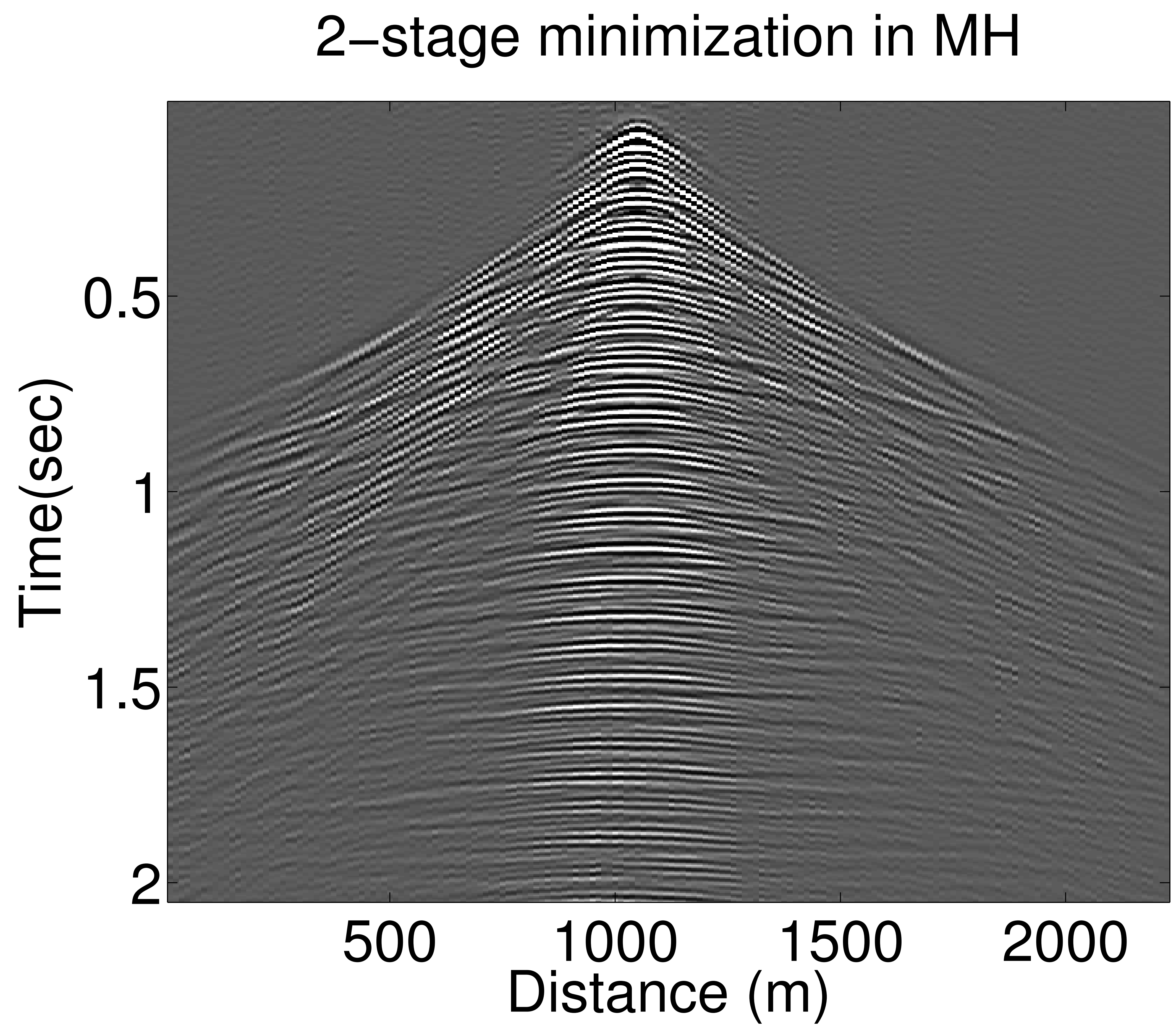
- ℓ_1 recovery in curvelet domain
- Weighting
- Jittered subsampling
- Midpoint-offset domain
- 2-stage weighted algorithm

Main result

Improvement on an example with 70% missing receivers.



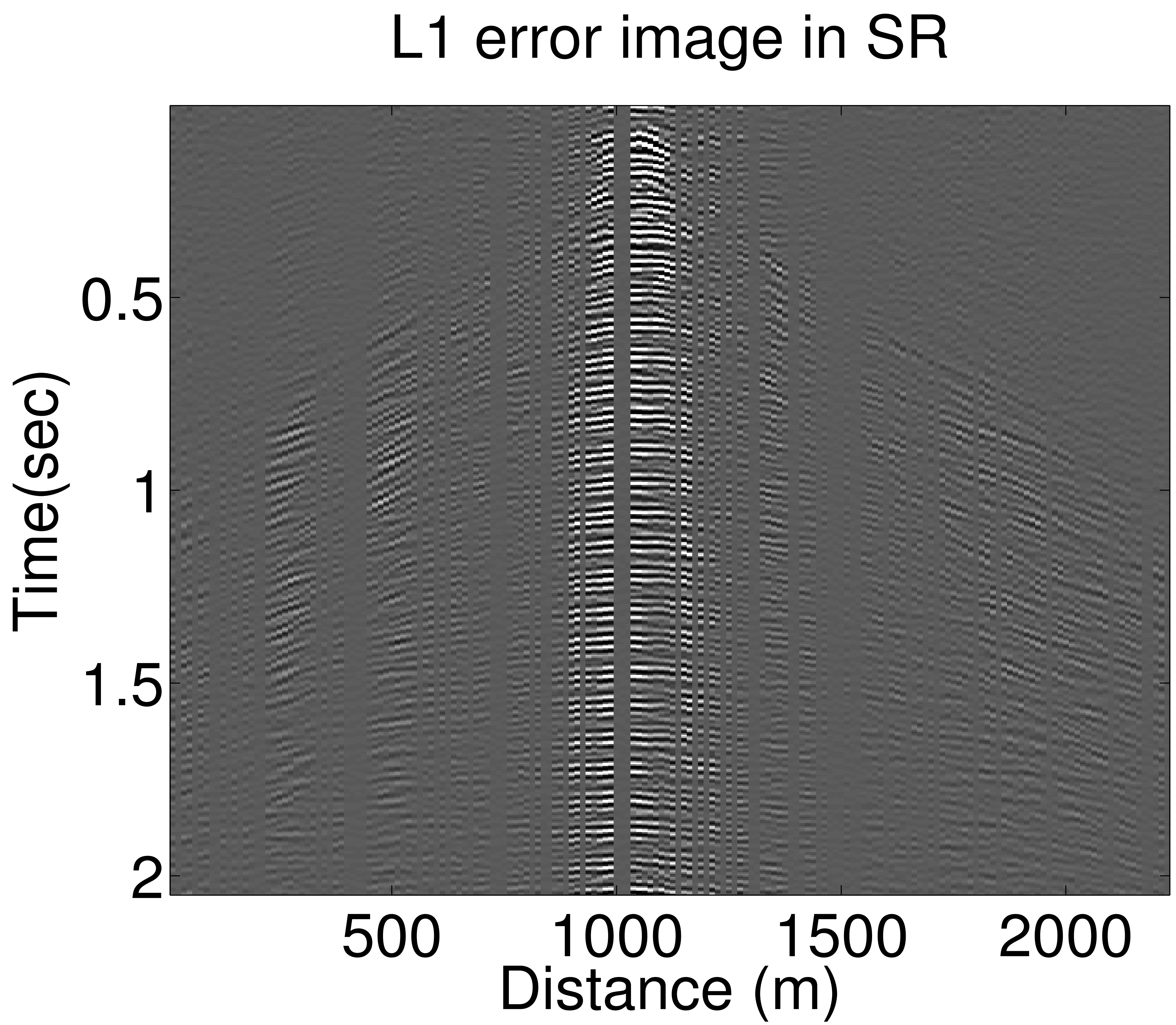
(a) SNR= 5.3 dB



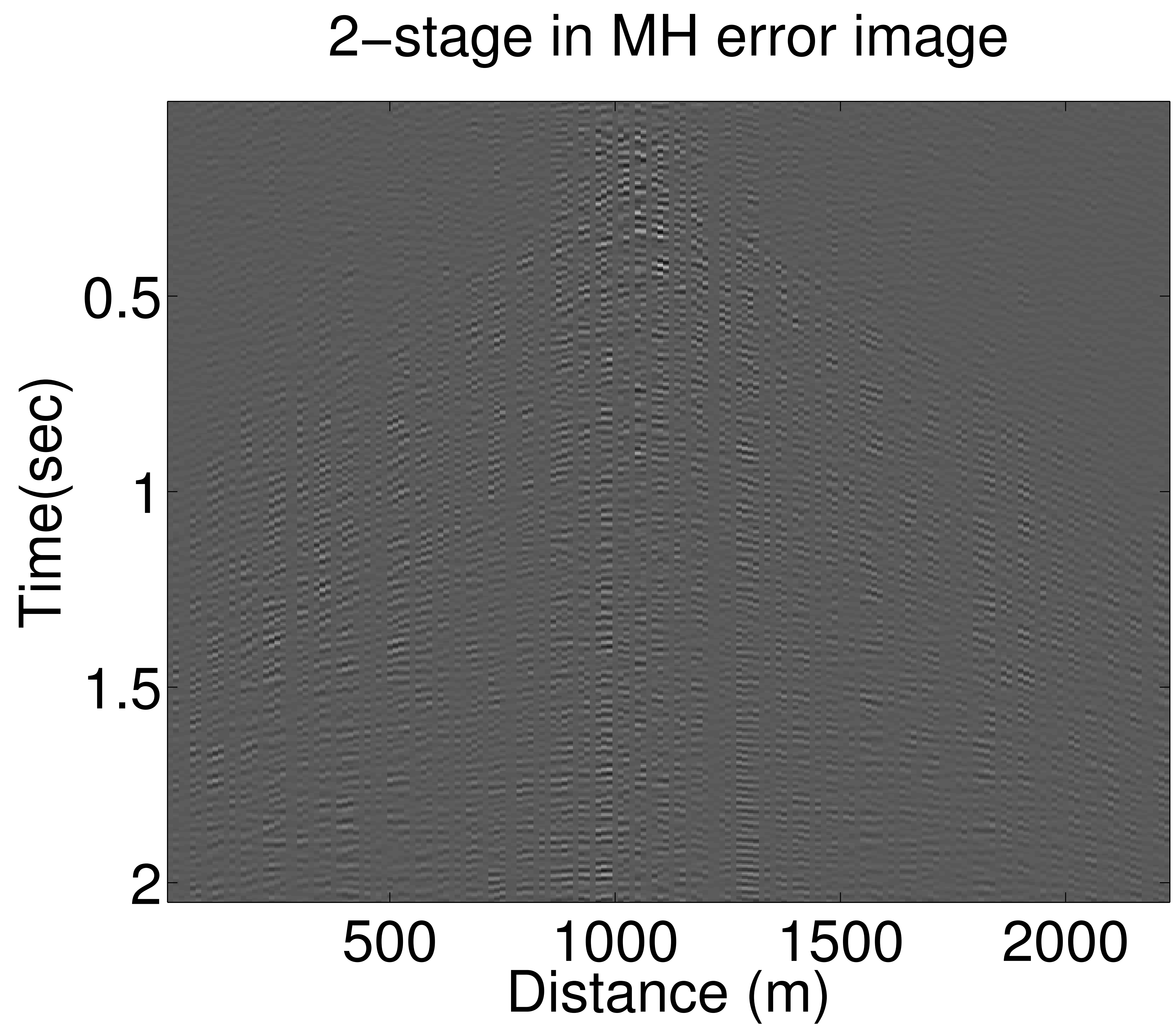
(b) SNR= 14.8 dB

Main result

Improvement on an example with 70% missing receivers.



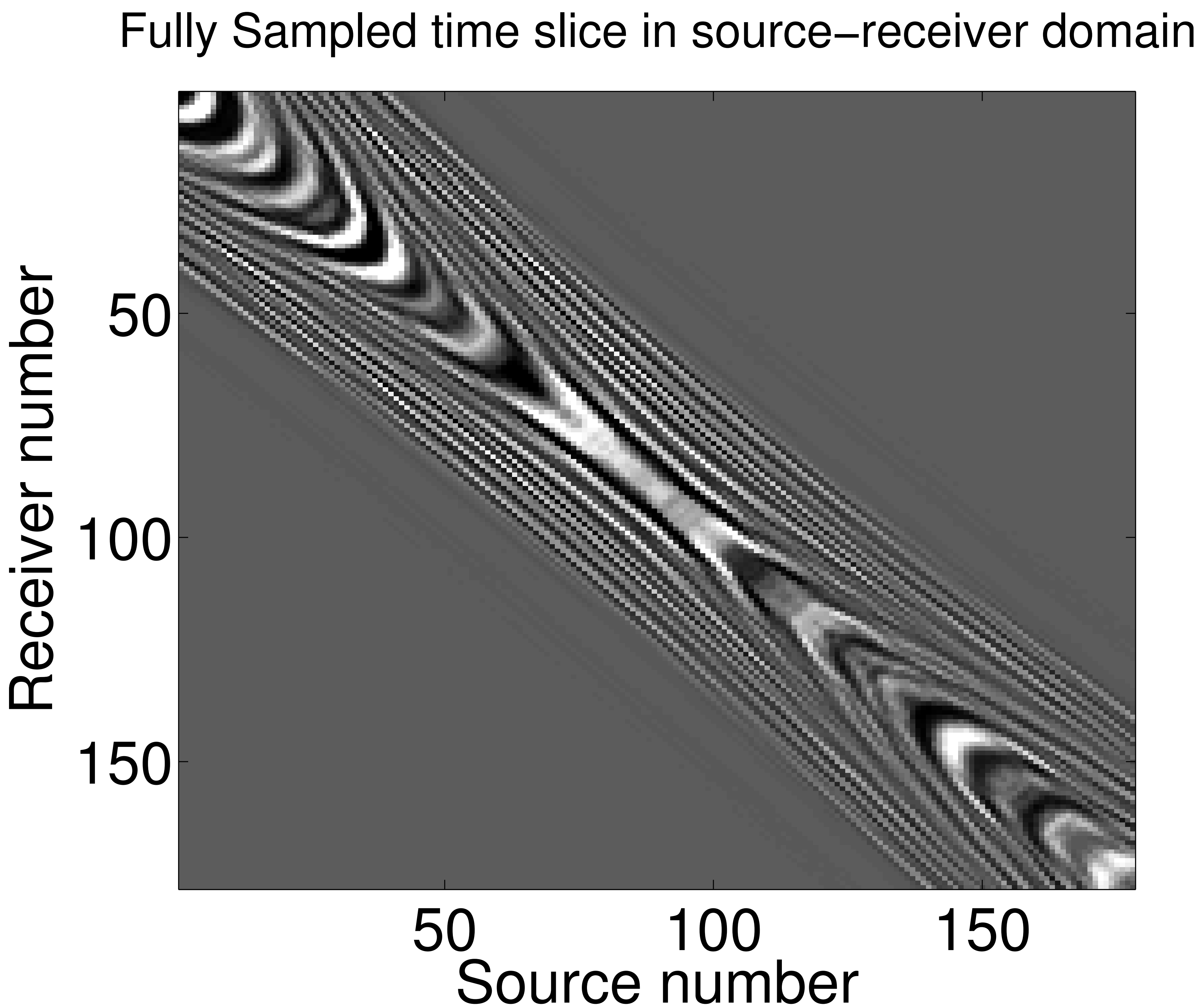
(c) SNR= 5.3 dB

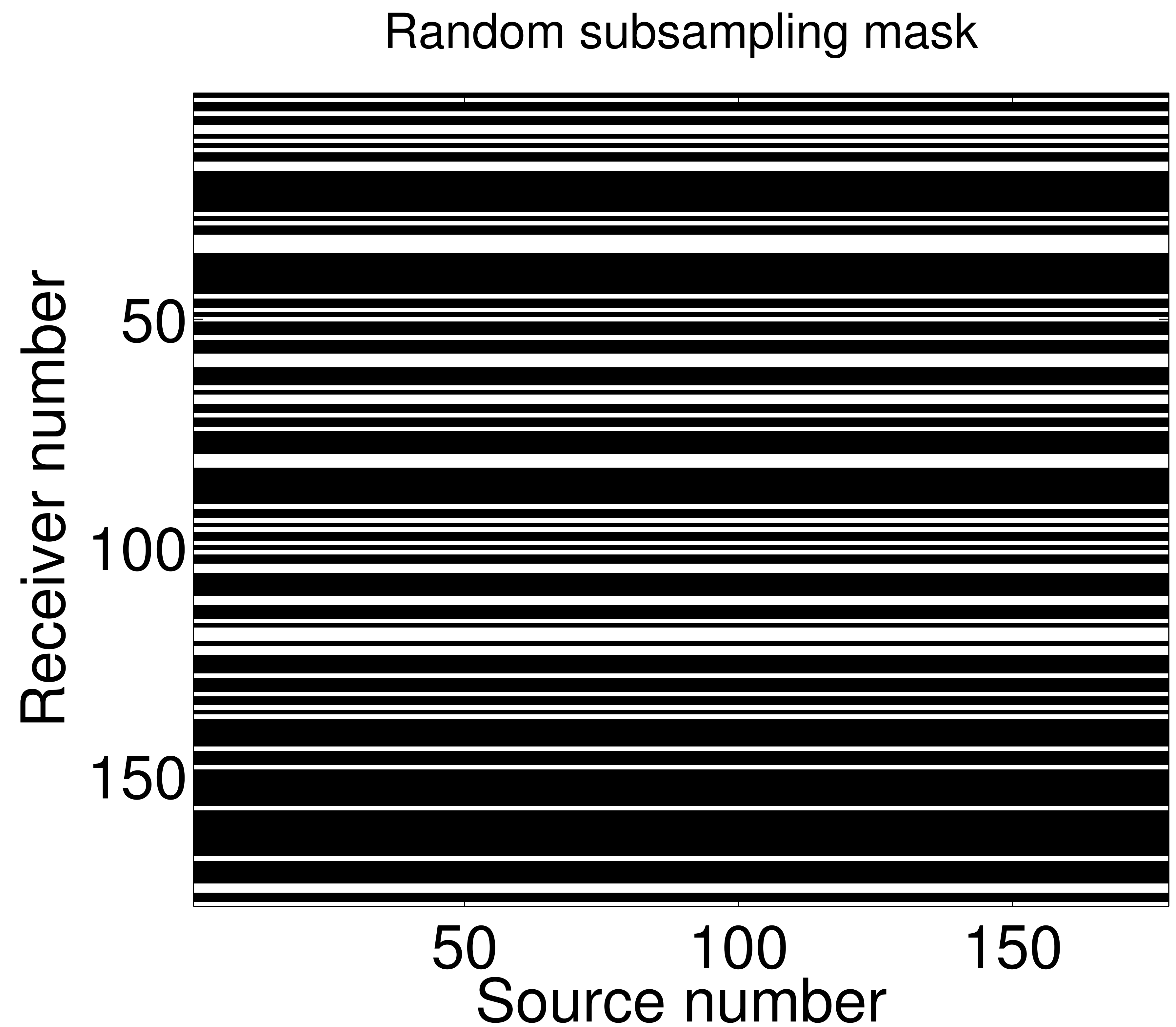


(d) SNR= 14.8 dB

ℓ_1 for randomized acquisition of seismic lines

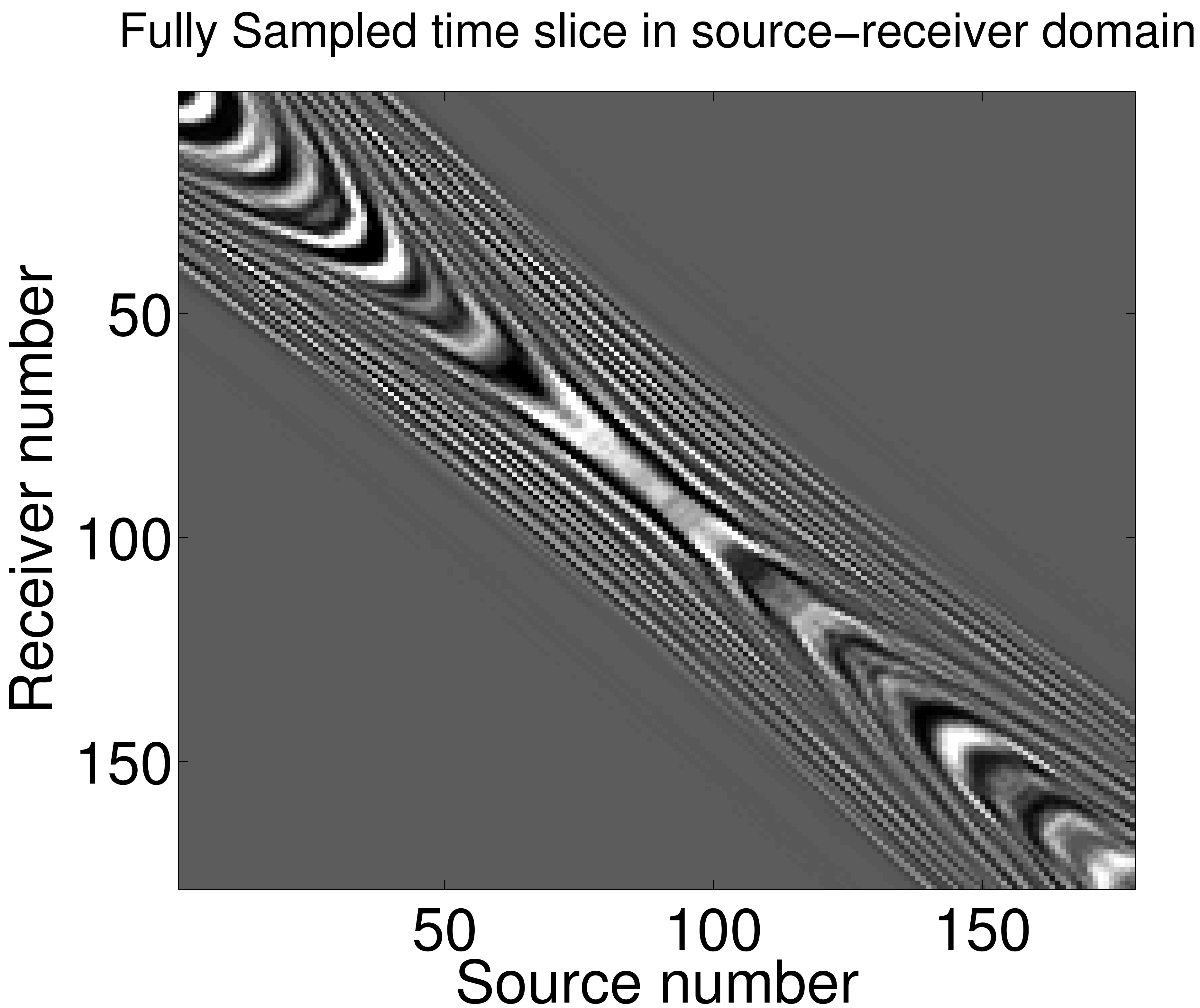
- Consider a seismic line with 178 sources, 178 receivers with a sample interval of 12.5m.
- 512 time samples collected in a 2s temporal window.
- 30% of the receiver spread is randomly subsampled.

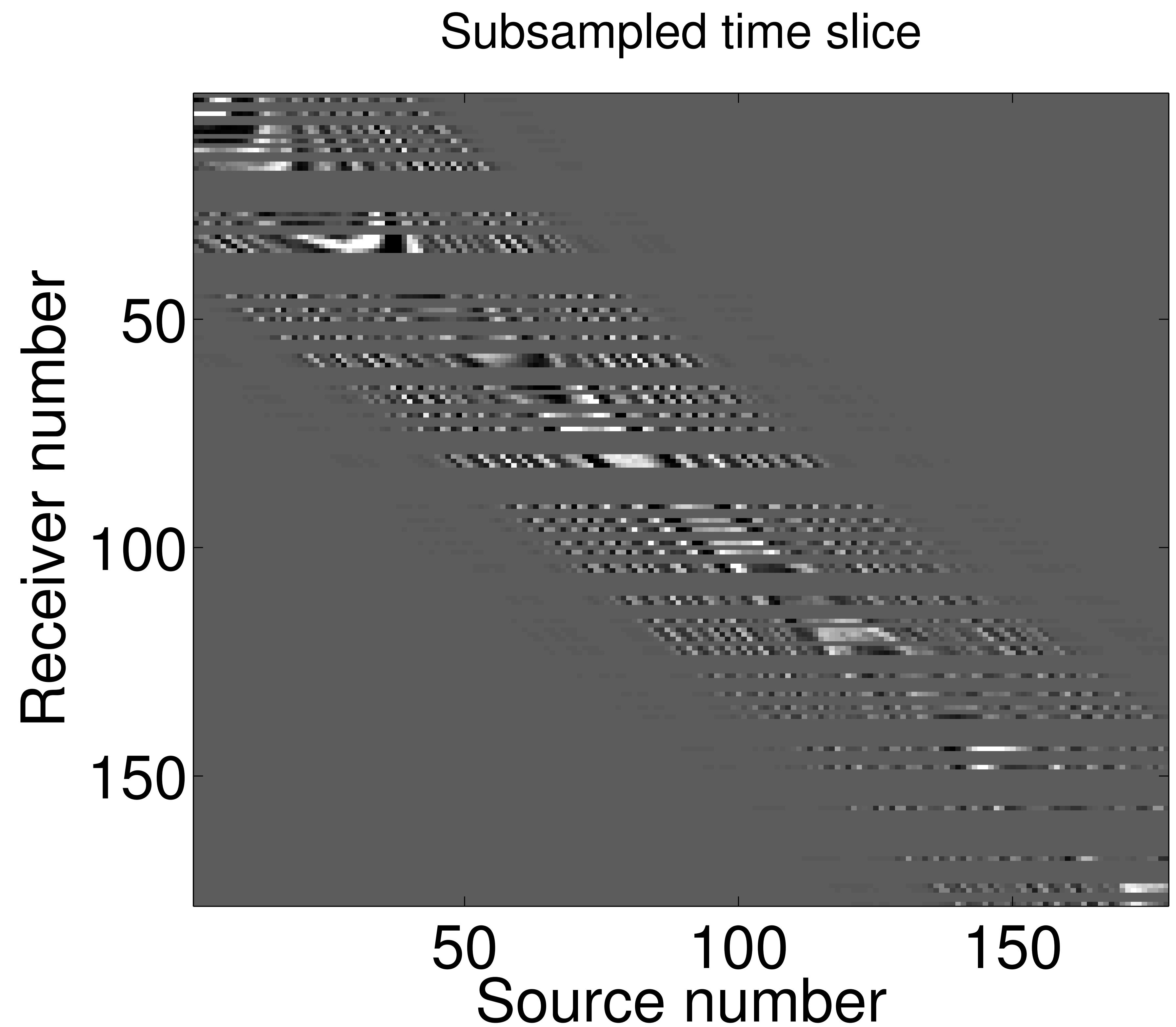




ℓ_1 for randomized acquisition of seismic lines

- Consider a seismic line with 178 sources, 178 receivers with a sample distance of 12.5m.
- 512 time samples collected in a 2s temporal window.
- 30% of the receiver spread is randomly subsampled.





ℓ_1 for seismic trace interpolation

We want to recover f by interpolating between a smaller number of measurements $b = RMf$.

ℓ_1 for seismic trace interpolation

We want to recover f by interpolating between a smaller number of measurements $b = RMf$.

Let $S \in \mathbb{C}^{P \times N}$ with $P > N$ be the redundant curvelet transform ($S^H S = \mathbb{I}$).

ℓ_1 for seismic trace interpolation

We want to recover f by interpolating between a smaller number of measurements $b = RMf$.

Let $S \in \mathbb{C}^{P \times N}$ with $P > N$ be the redundant curvelet transform ($S^H S = \mathbb{I}$).

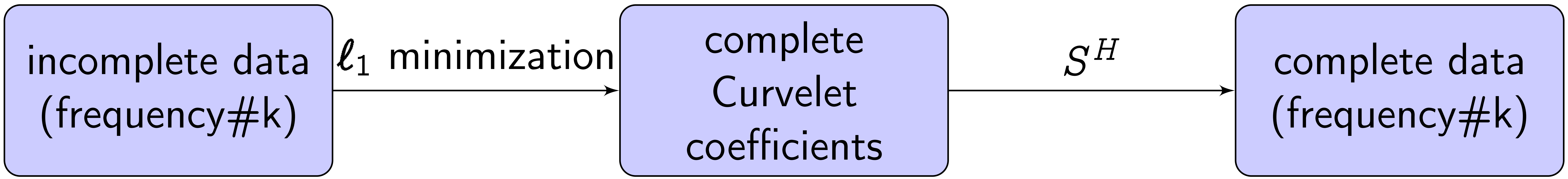
Then $b = RMS^H x$, where x can be recovered by sparse recovery algorithms like ℓ_1 minimization.

Using ℓ_1 for seismic trace interpolation

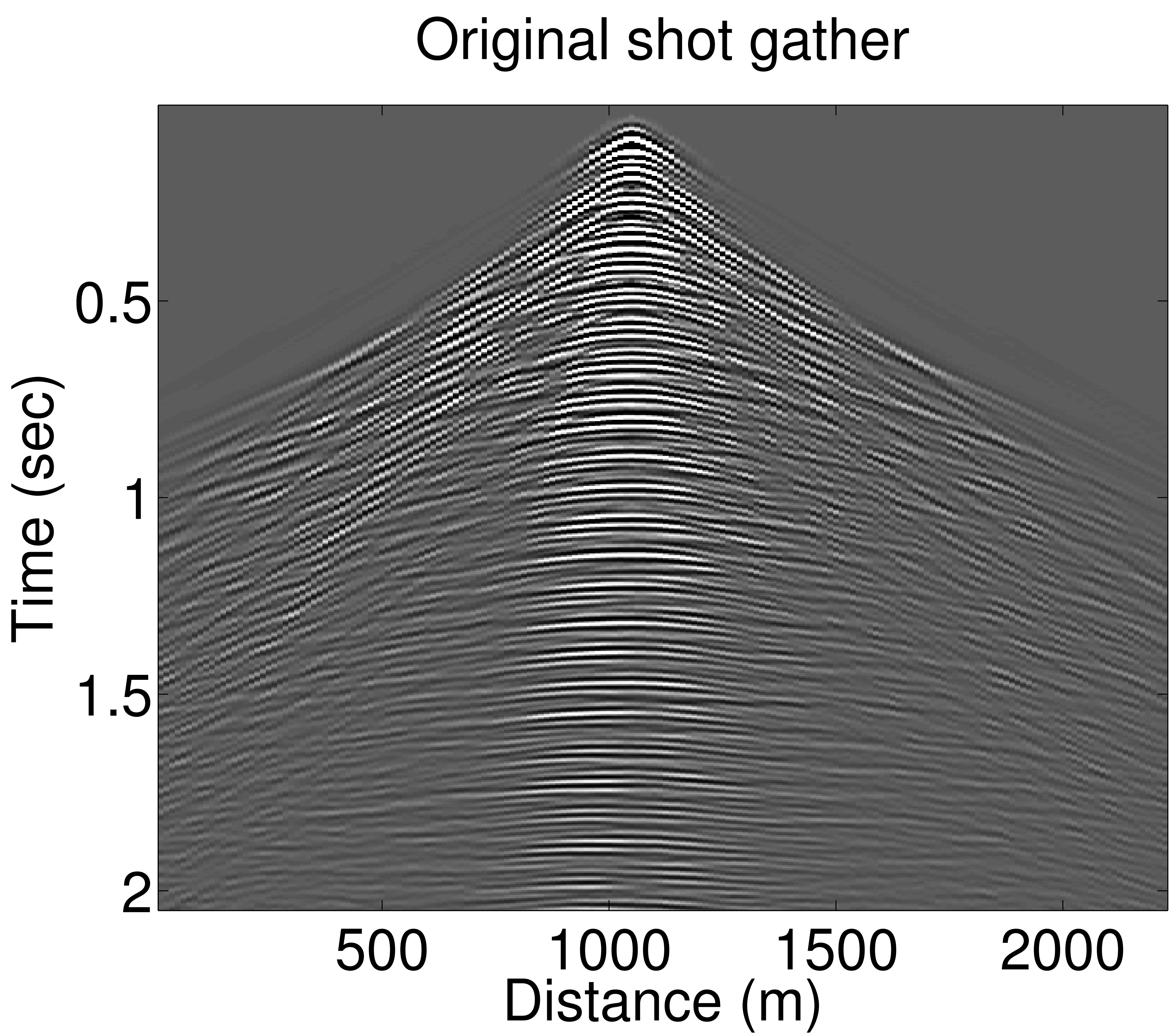
To recover f from the measurements $b = RMS^H x$, we solve the ℓ_1 minimization problem

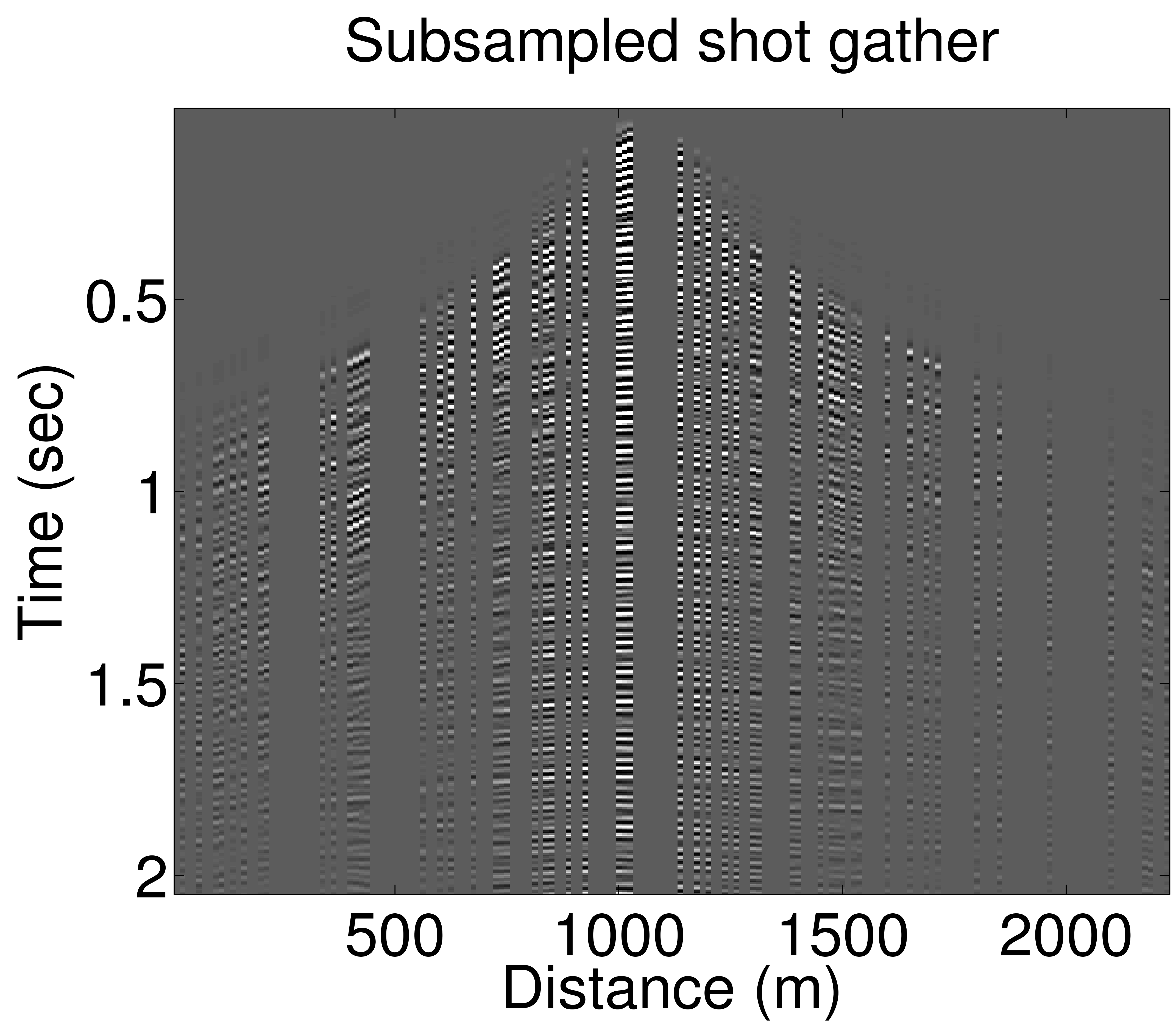
$$x^{\ell_1} := \underset{z \in R^P}{\text{minimize}} ||z||_1 \quad \text{subject to} \quad ||RMS^H z - b||_2 \leq \epsilon,$$

and approximate f by $S^H x^{\ell_1}$.

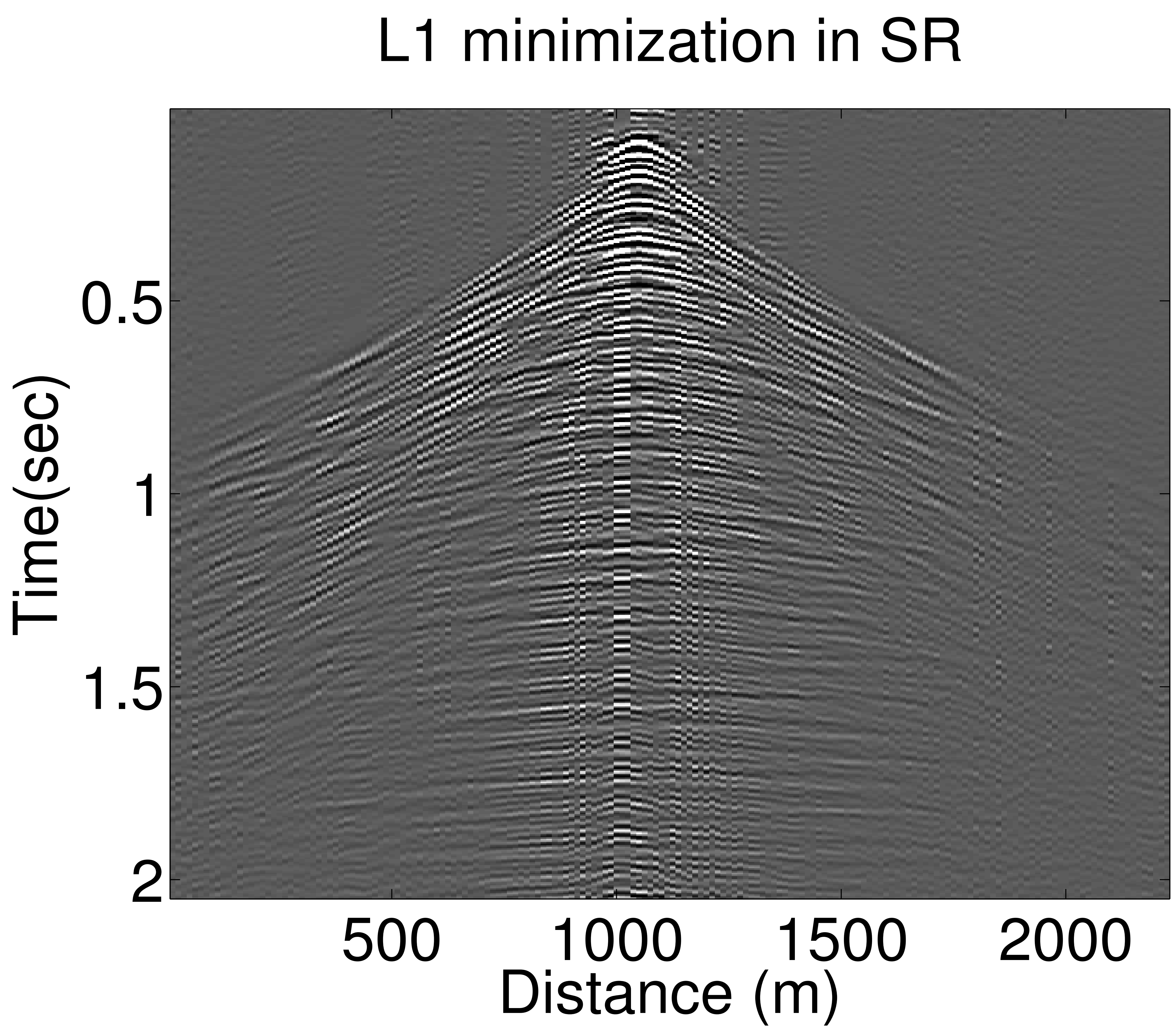


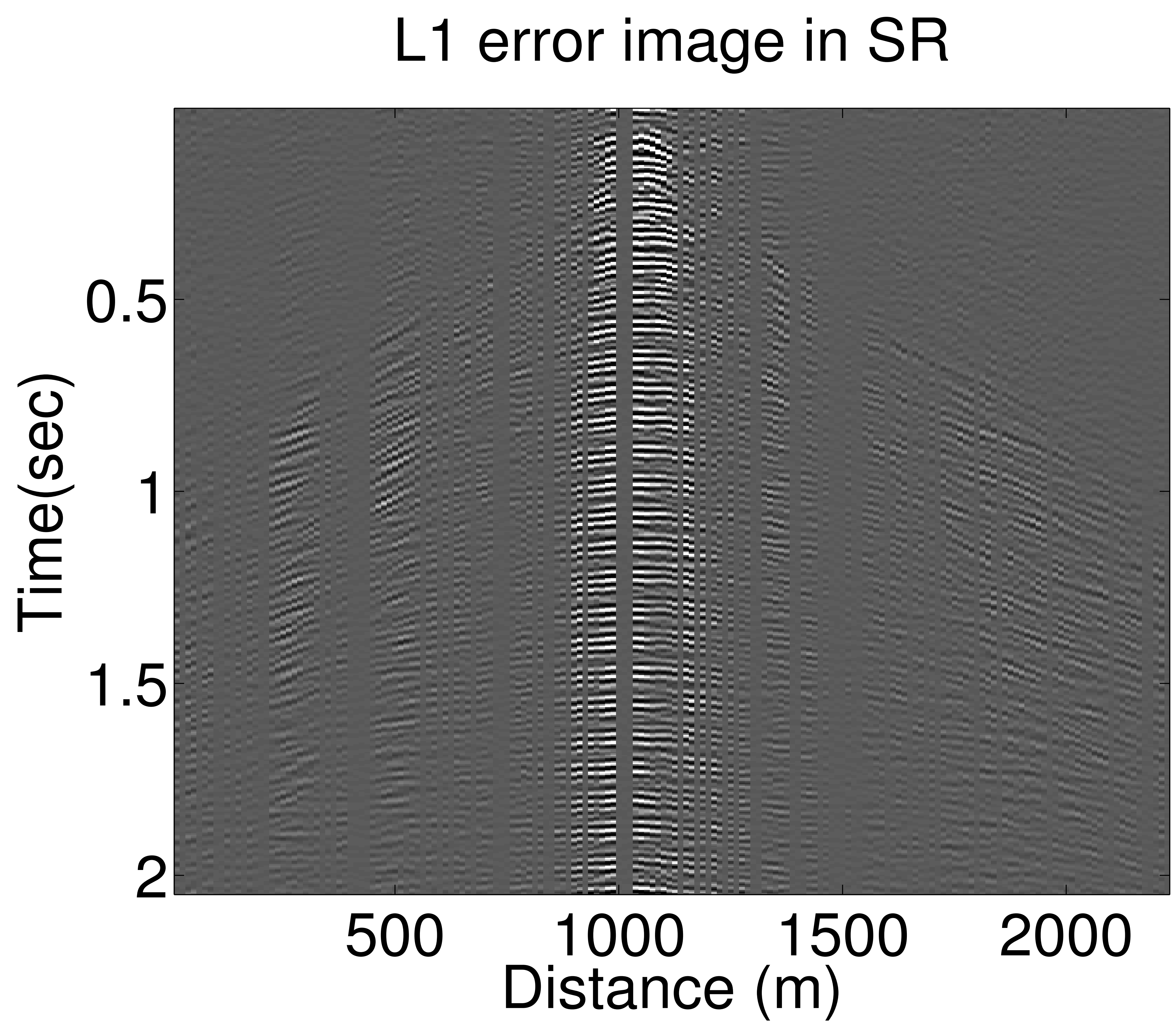
Recovery results (shotgather # 84)



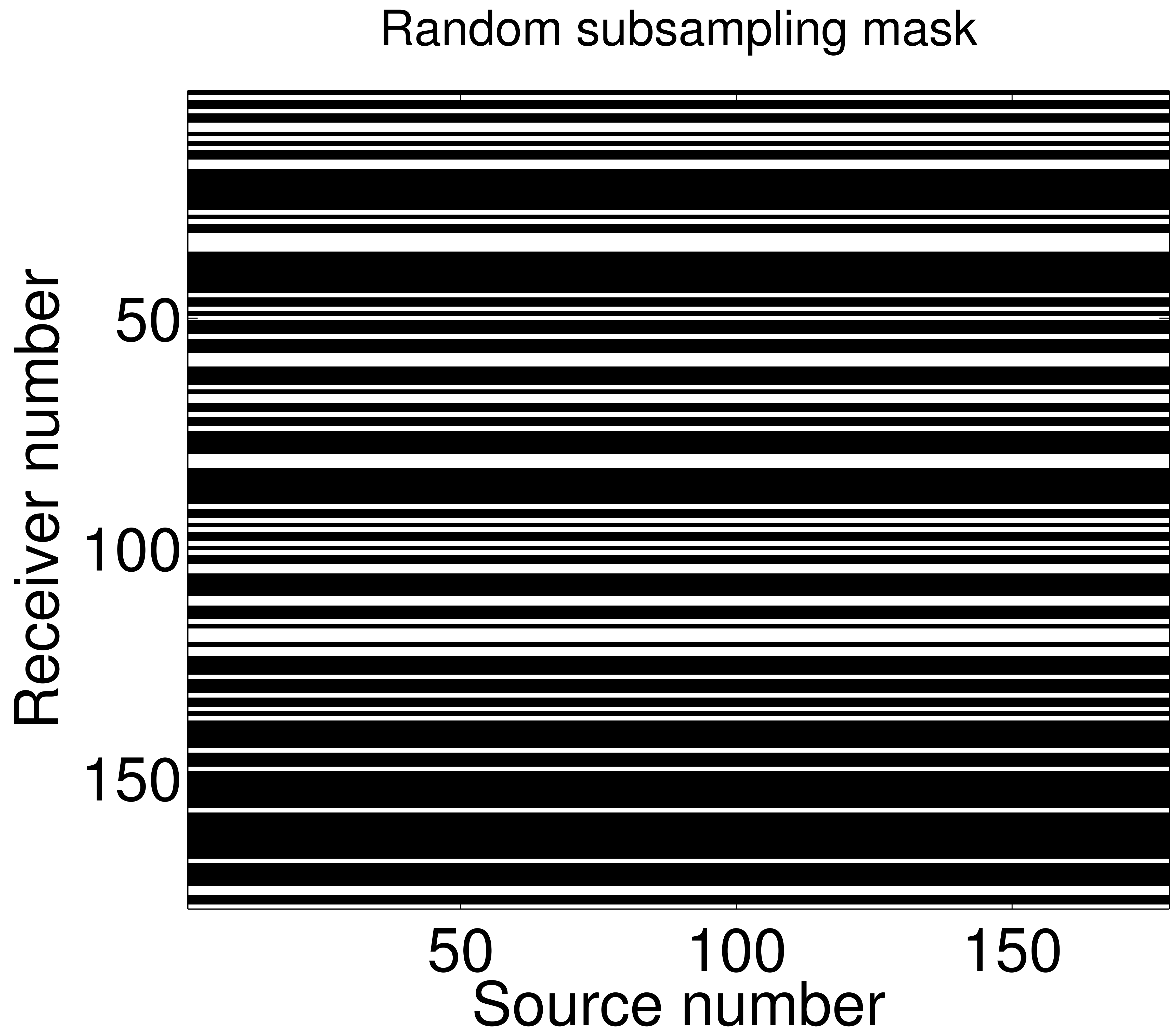


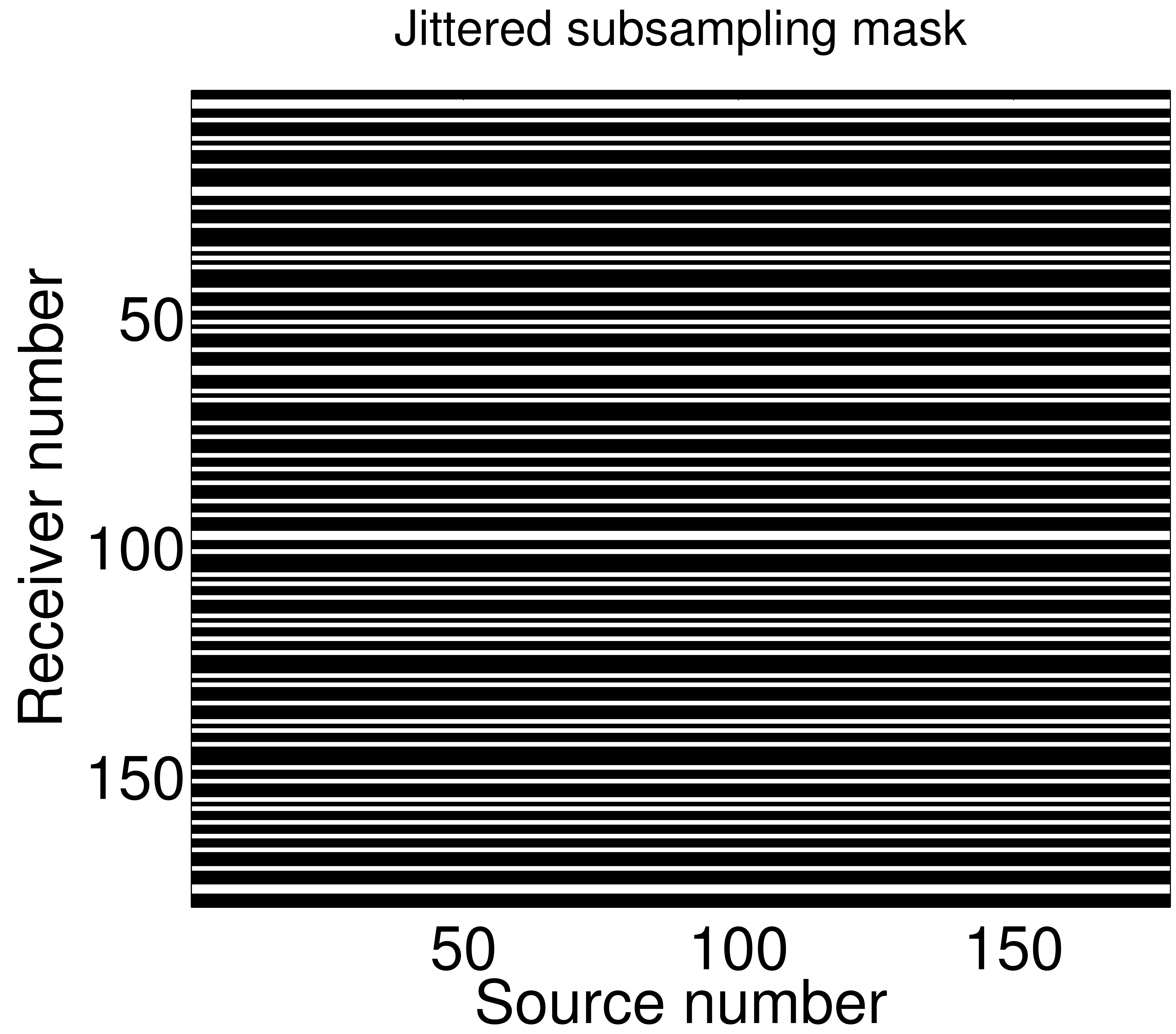
Recovery results: ℓ_1 minimization (5.3 dB)





Jittered subsampling mask





Why does jittering help?

- success of Randomized subsampling depends on destroying the structure of data in some transform domain.

Why does jittering help?

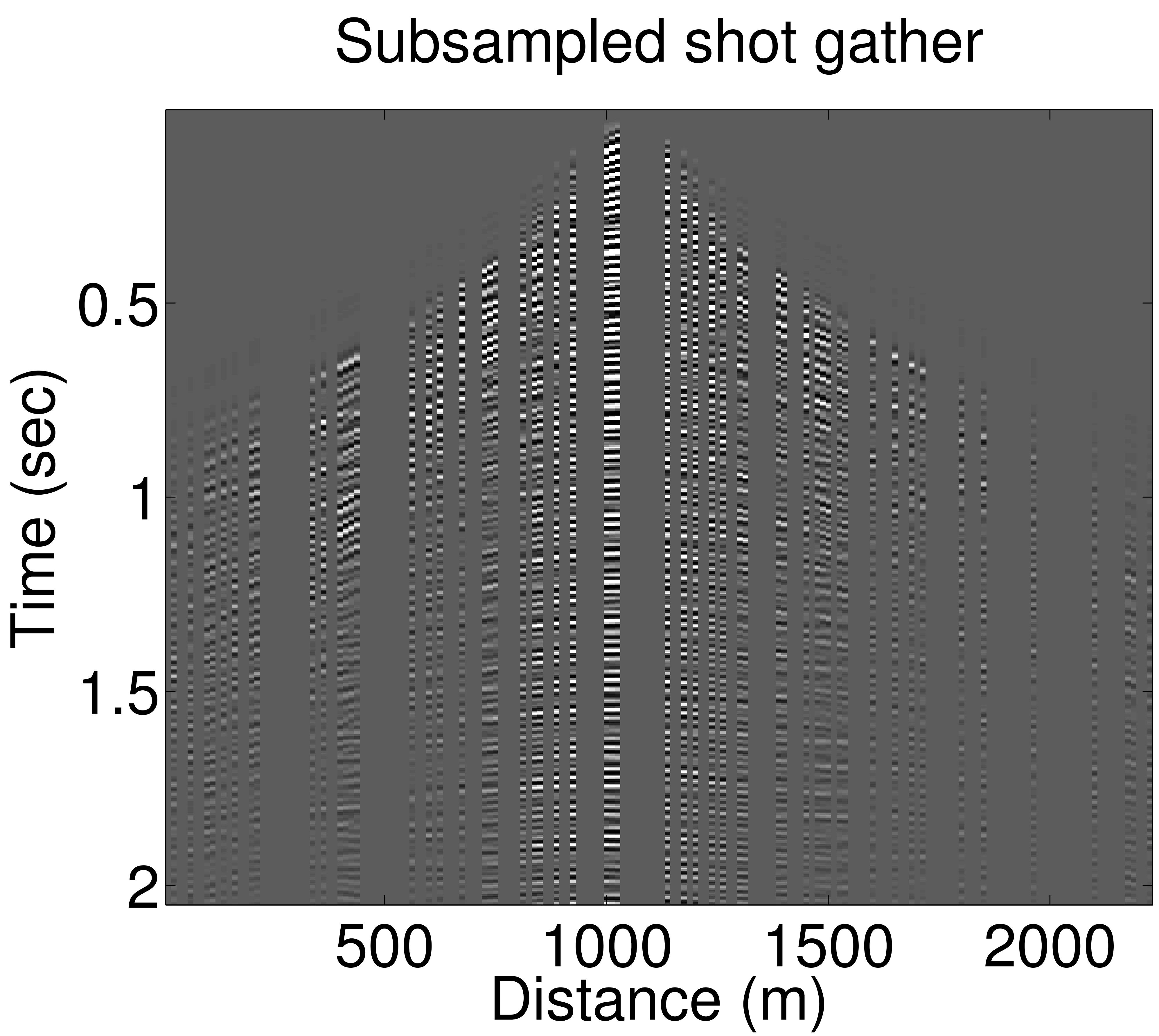
- success of Randomized subsampling depends on destroying the structure of data in some transform domain.
- Uniformly random subsampling might result in large gap in the data.

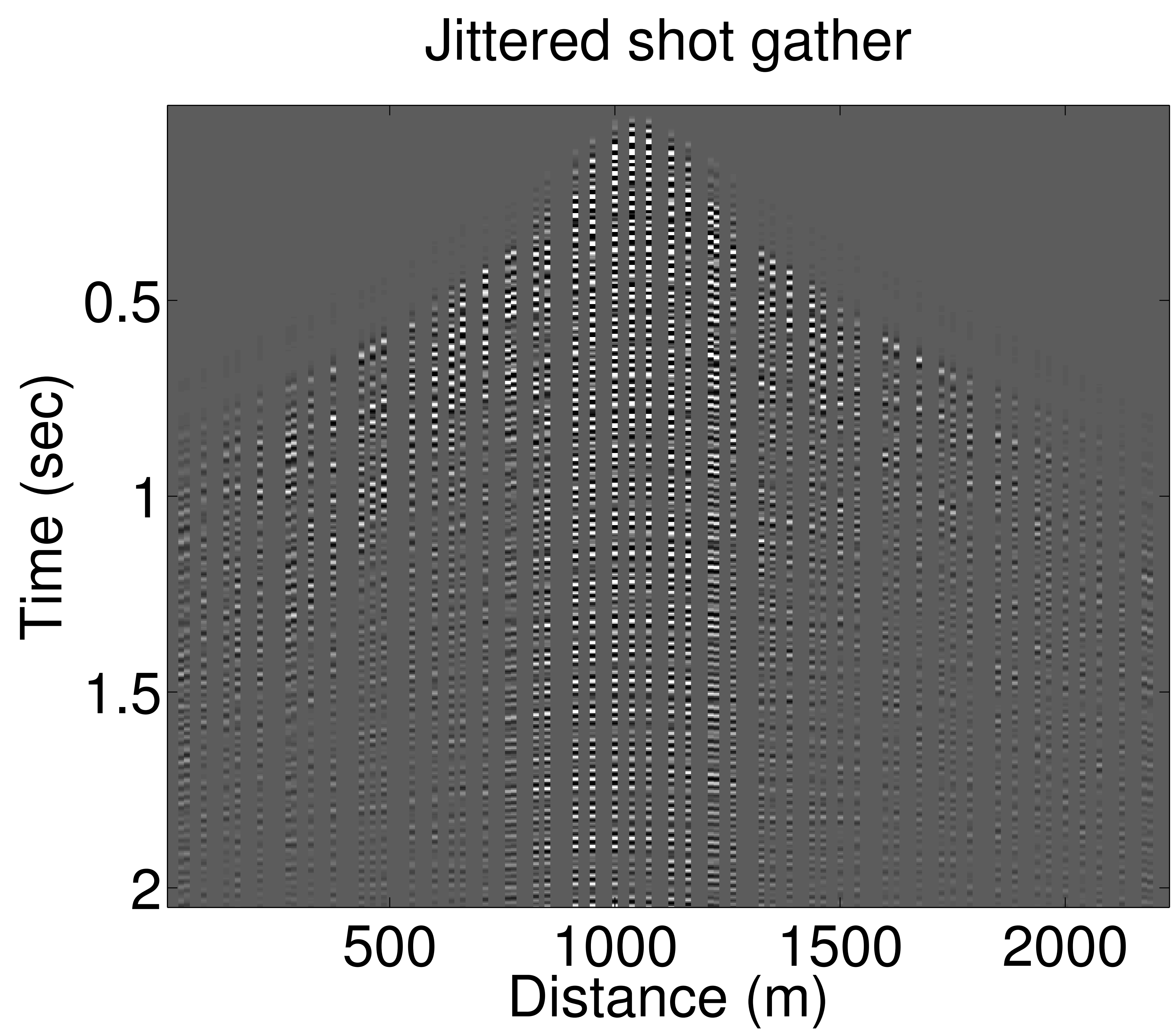
Why does jittering help?

- success of Randomized subsampling depends on destroying the structure of data in some transform domain.
- Uniformly random subsampling might result in large gap in the data.
- Jittering is a safe alternative that doesn't allow large gaps in the data.

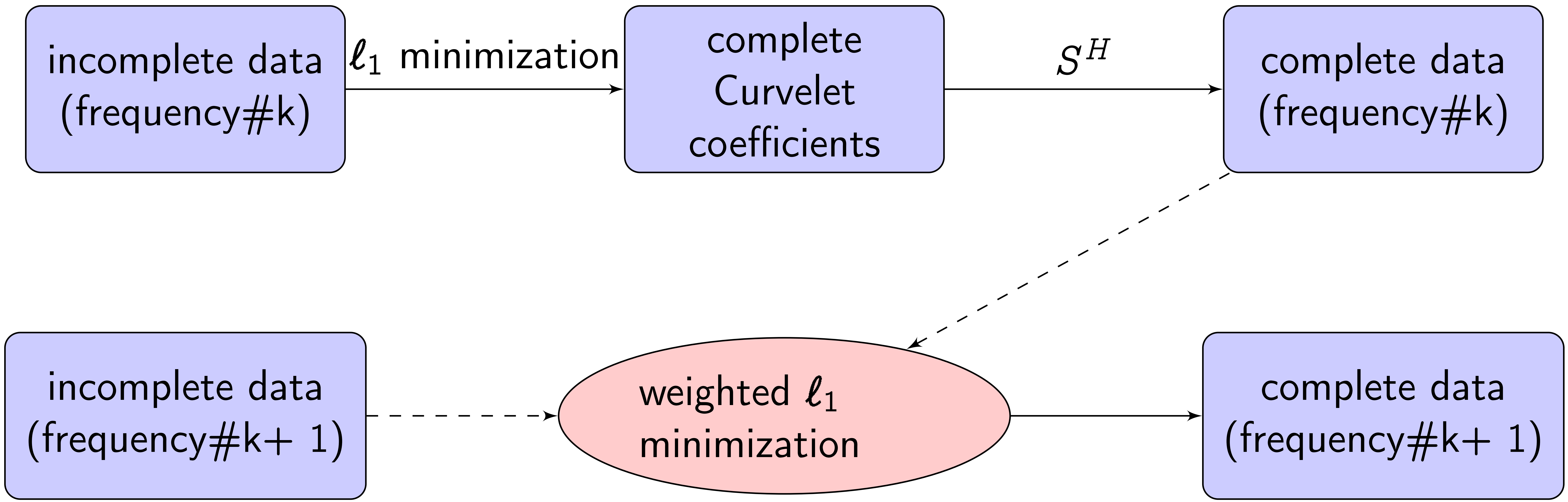
New subsampled data

Jittered sampling controls the average amount of information per row in the transform domain.





Weighted ℓ_1 for seismic trace interpolation

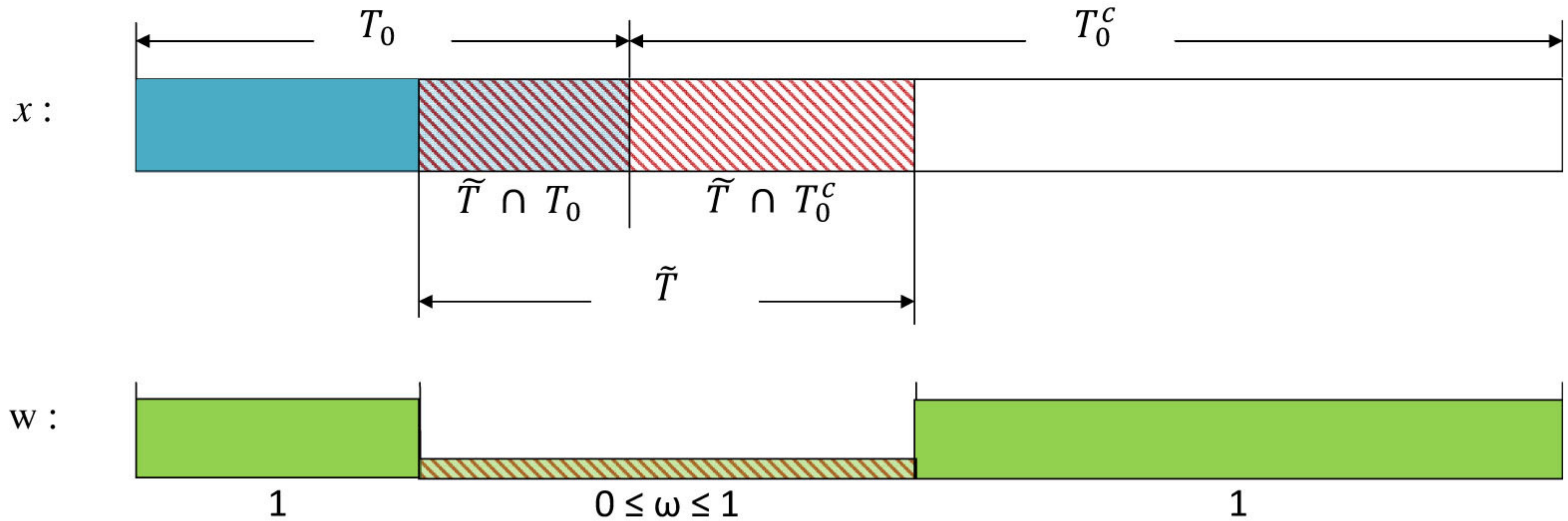


Monday, December 8, 14

Weighted ℓ_1 for seismic trace interpolation

$$x^{w\ell_1} := \underset{z \in R^P}{\text{minimize}} ||z||_{1,\mathrm{w}} \quad \text{subject to} \quad ||RMS^H z - b||_2 \leq \epsilon,$$

- For a vector x, $||x||_{1,\mathrm{w}} := \Sigma_i \mathrm{w}_i |x_i|$ is the weighted ℓ_1 norm of x.
- $\mathrm{w}_i = \omega < 1$ if x_i is in the support estimate. Otherwise $\mathrm{w}_i = 1$.



Why does weighting help?

- We partition the data which reduces the dimension in each slice.

Why does weighting help?

- We partition the data which reduces the dimension in each slice.
- 2-D curvelet transform captures the continuity in each slice. However we lose the continuity along frequency slices.

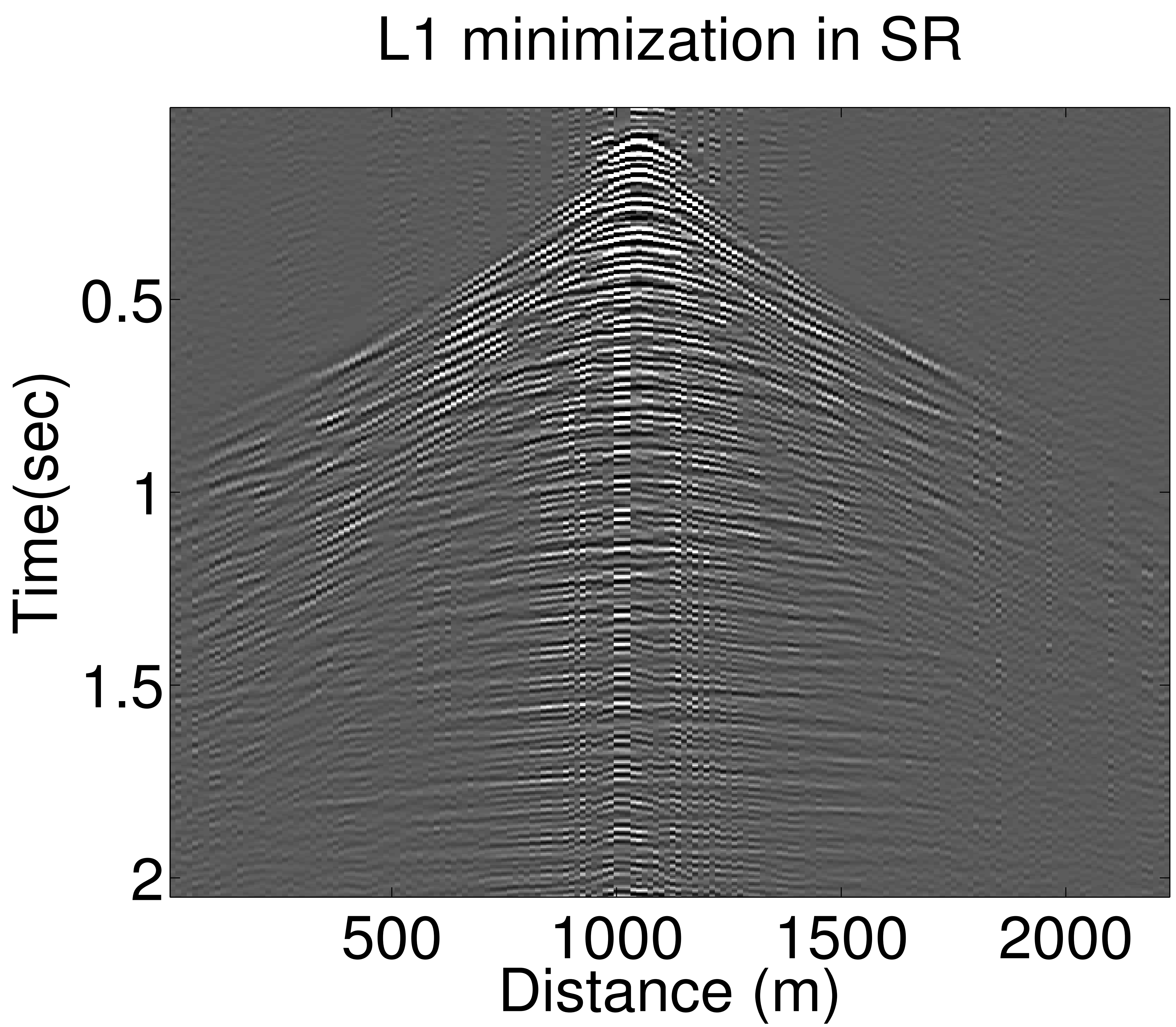
Why does weighting help?

- We partition the data which reduces the dimension in each slice.
- 2-D curvelet transform captures the continuity in each slice. However we lose the continuity along frequency slices.
- We utilize the continuity along adjacent frequency slices by weighting.

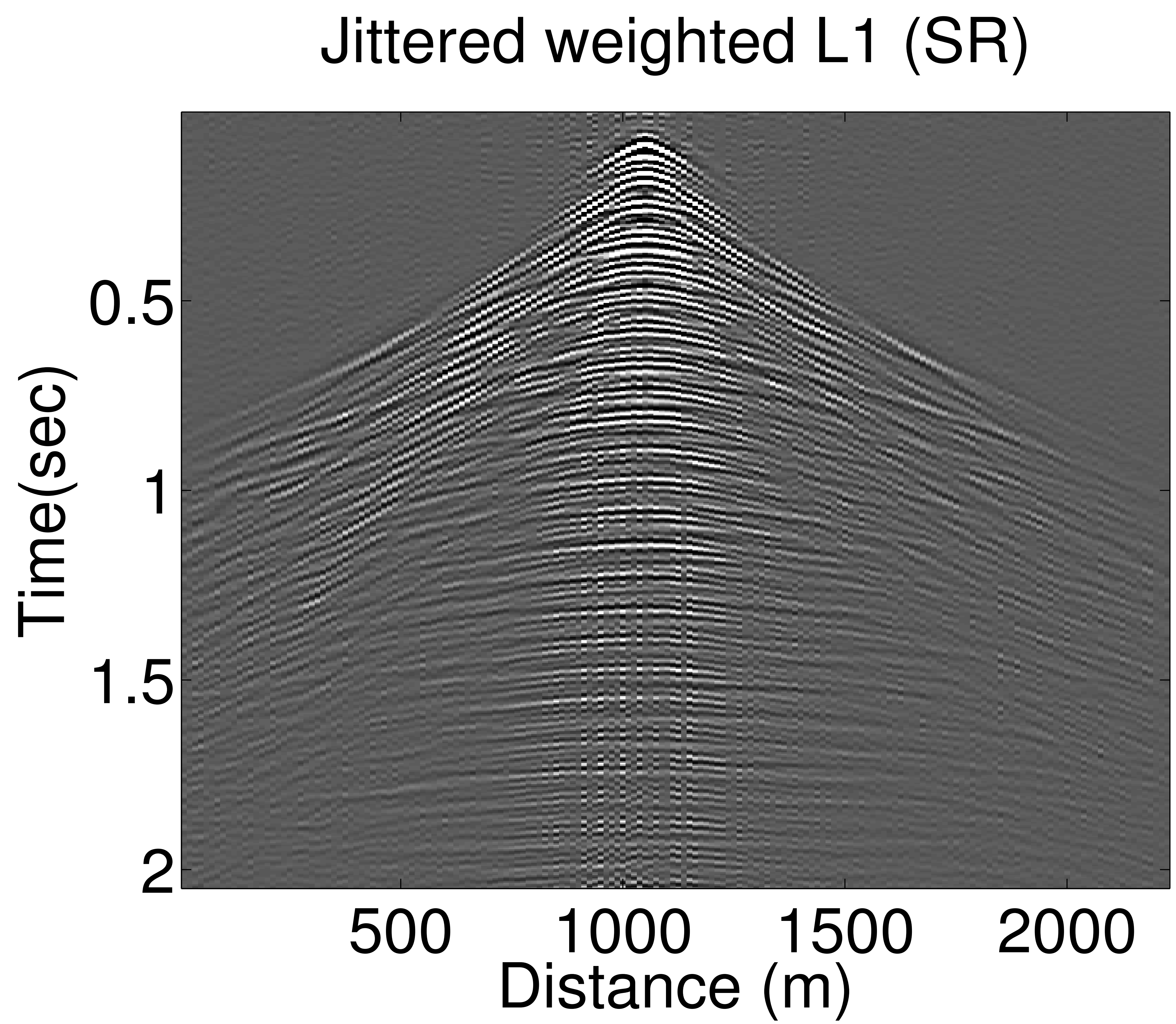
Why does weighting help?

- We partition the data which reduces the dimension in each slice.
- 2-D curvelet transform captures the continuity in each slice. However we lose the continuity along frequency slices.
- We utilize the continuity along adjacent frequency slices by weighting.
- If the support estimate is at least 50% accurate, we get better results.

Recovery results: ℓ_1 vs weighted ℓ_1

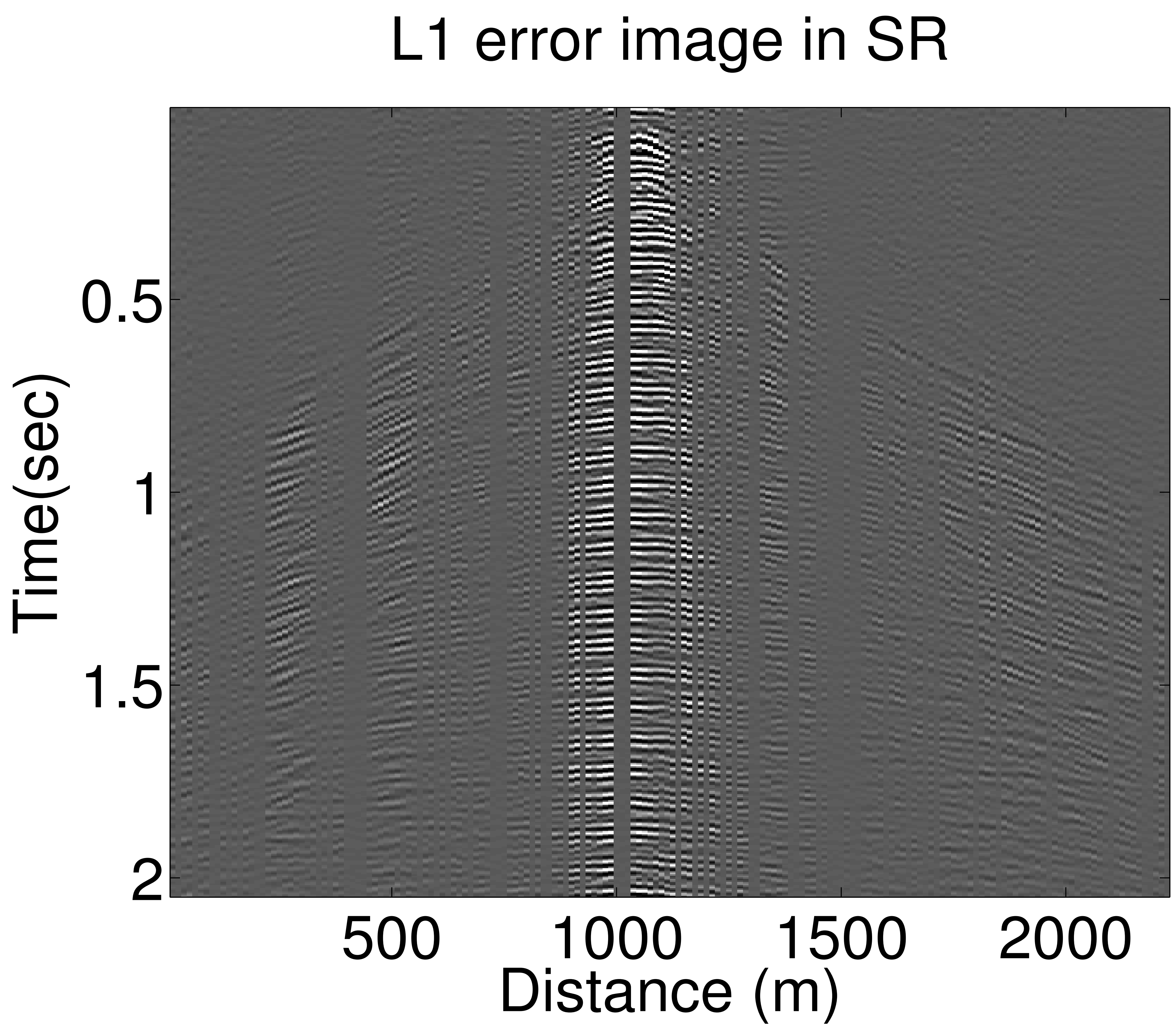


(k) SNR=5.4 dB

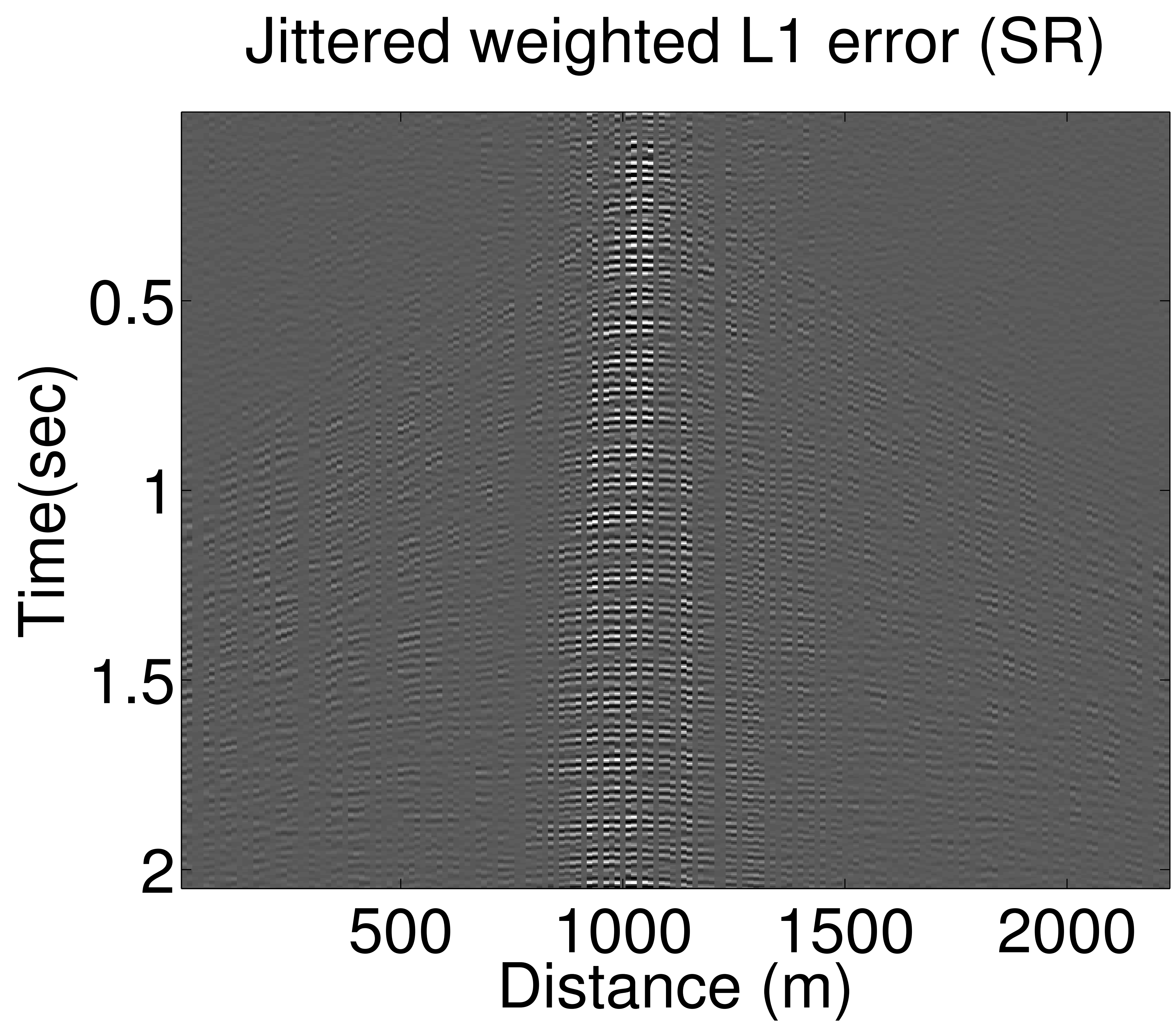


(l) SNR= 8.8 dB

Recovery error: ℓ_1 vs weighted ℓ_1

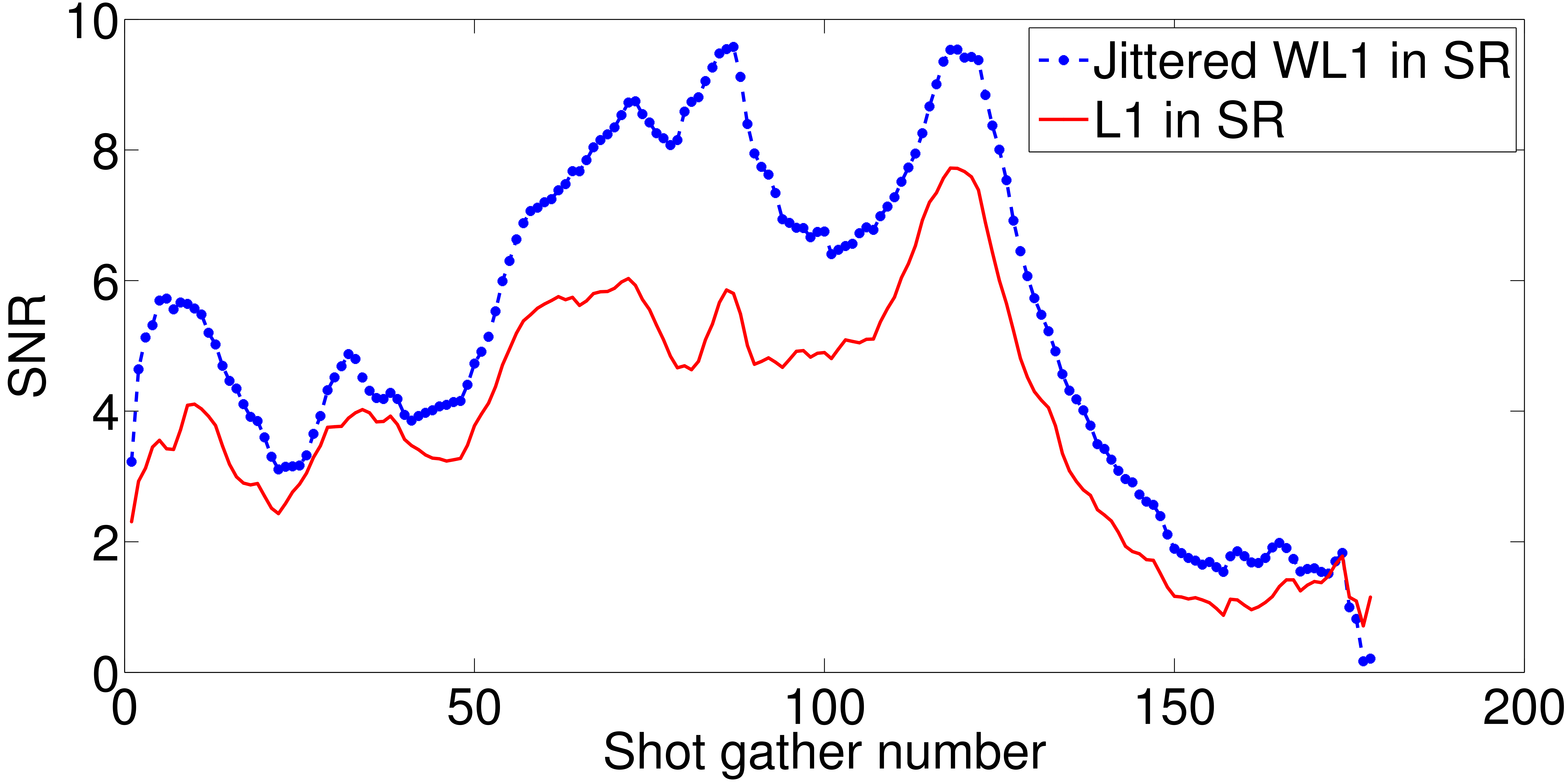


(m) SNR=5.4 dB



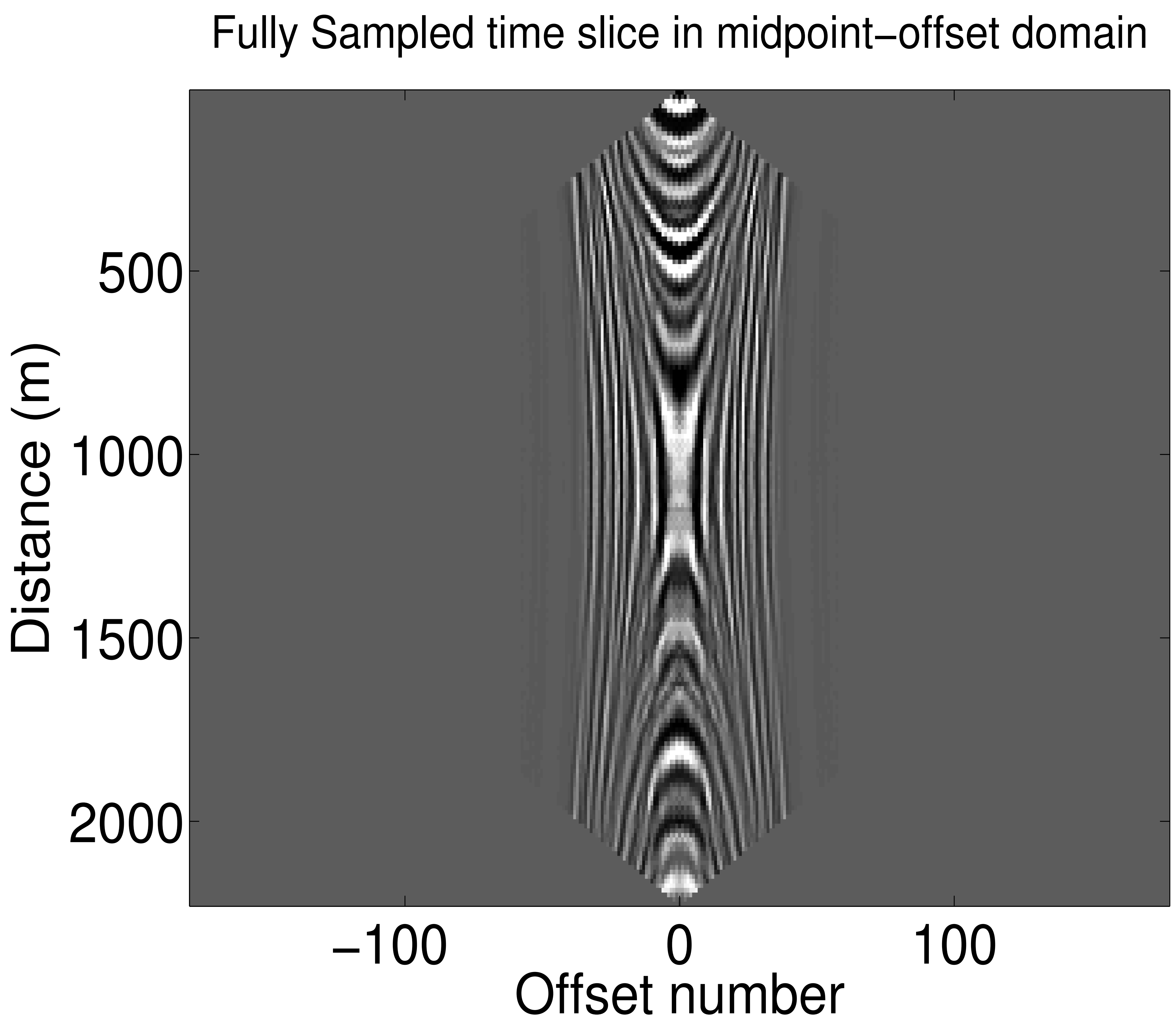
(n) SNR= 8.8 dB

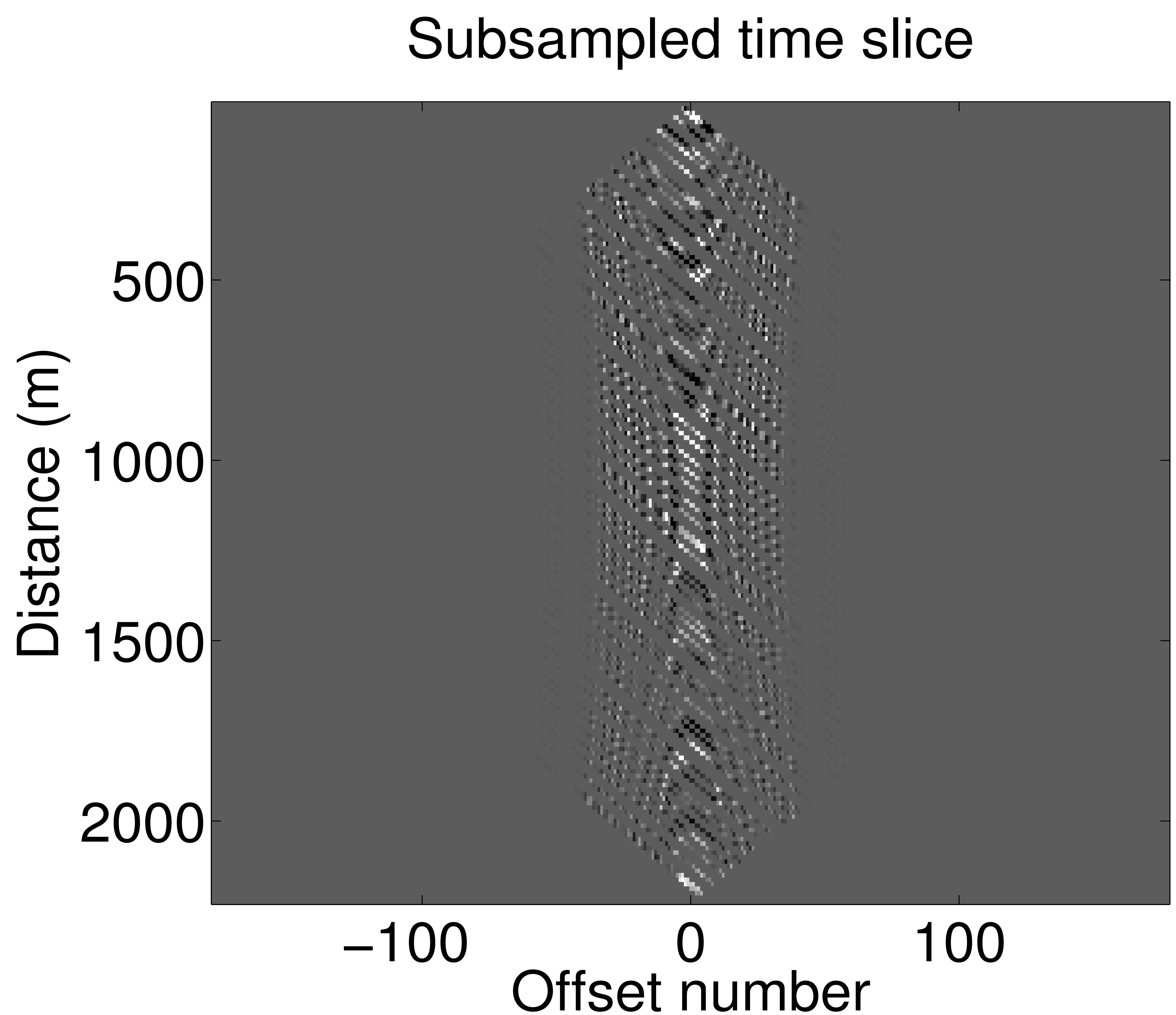
Shot gathers: ℓ_1 vs weighted ℓ_1



Recovery in the midpoint-offset domain

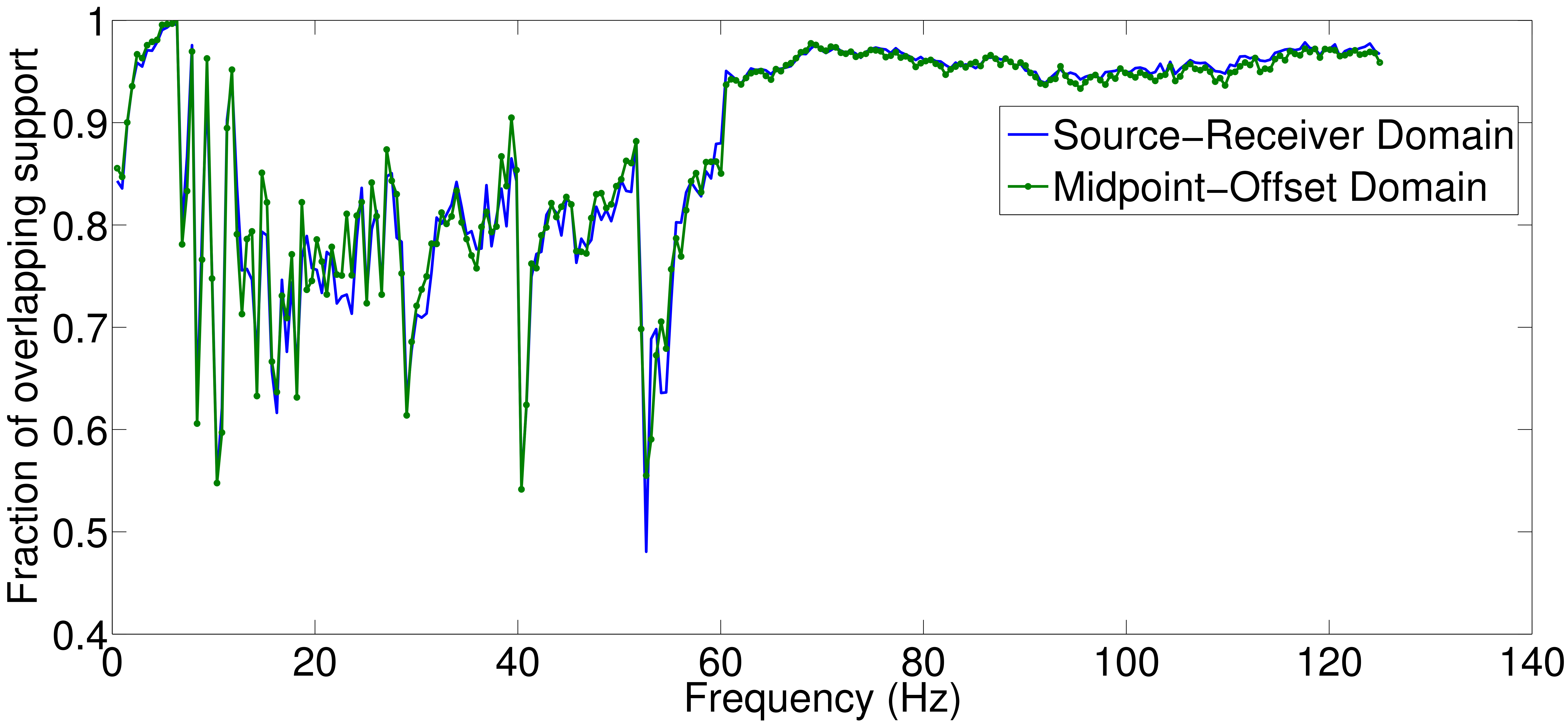
We transform the seismic line into the frequency-midpoint-offset (MH) domain.



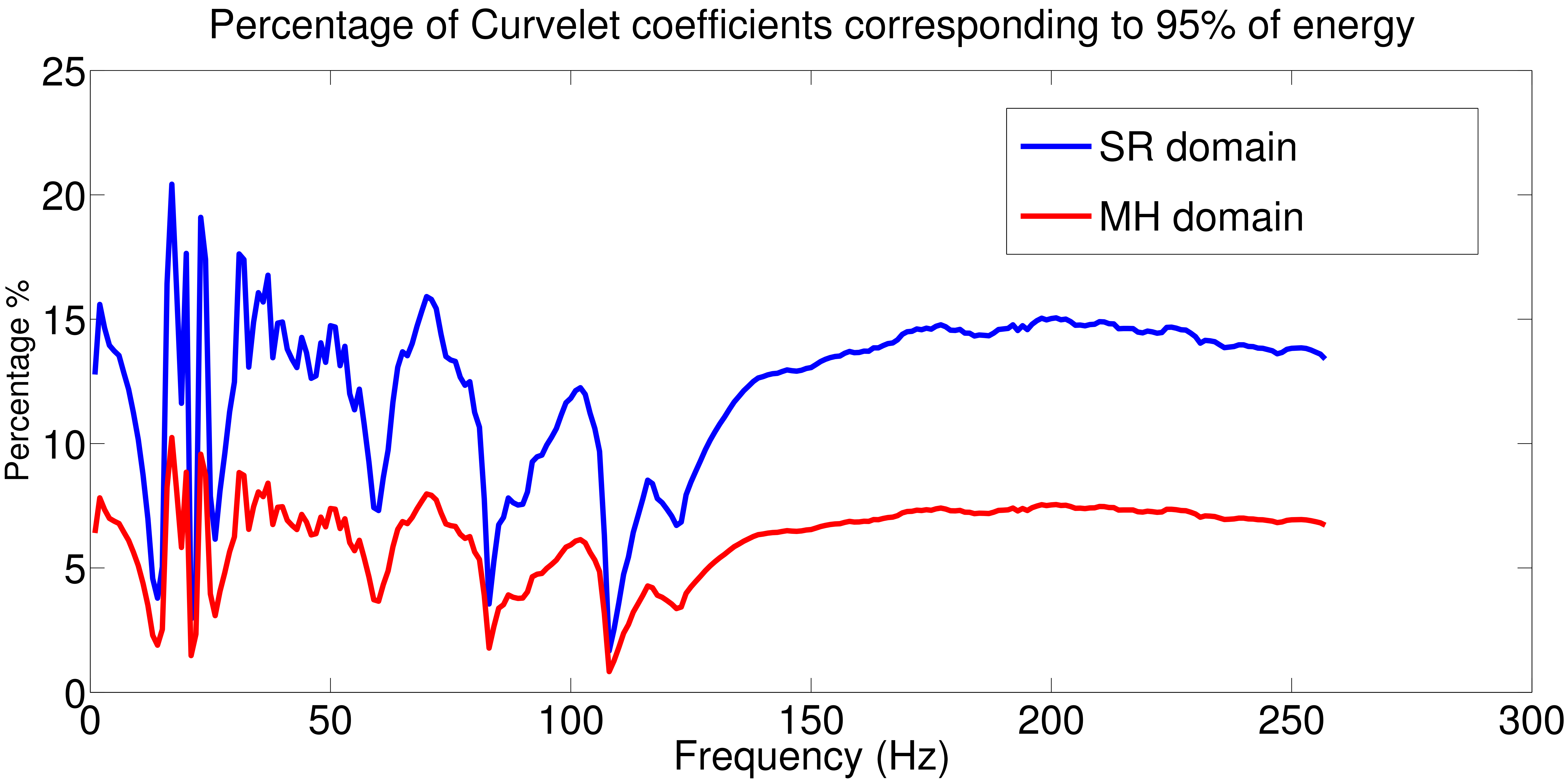


Weighting in MH and SR domain

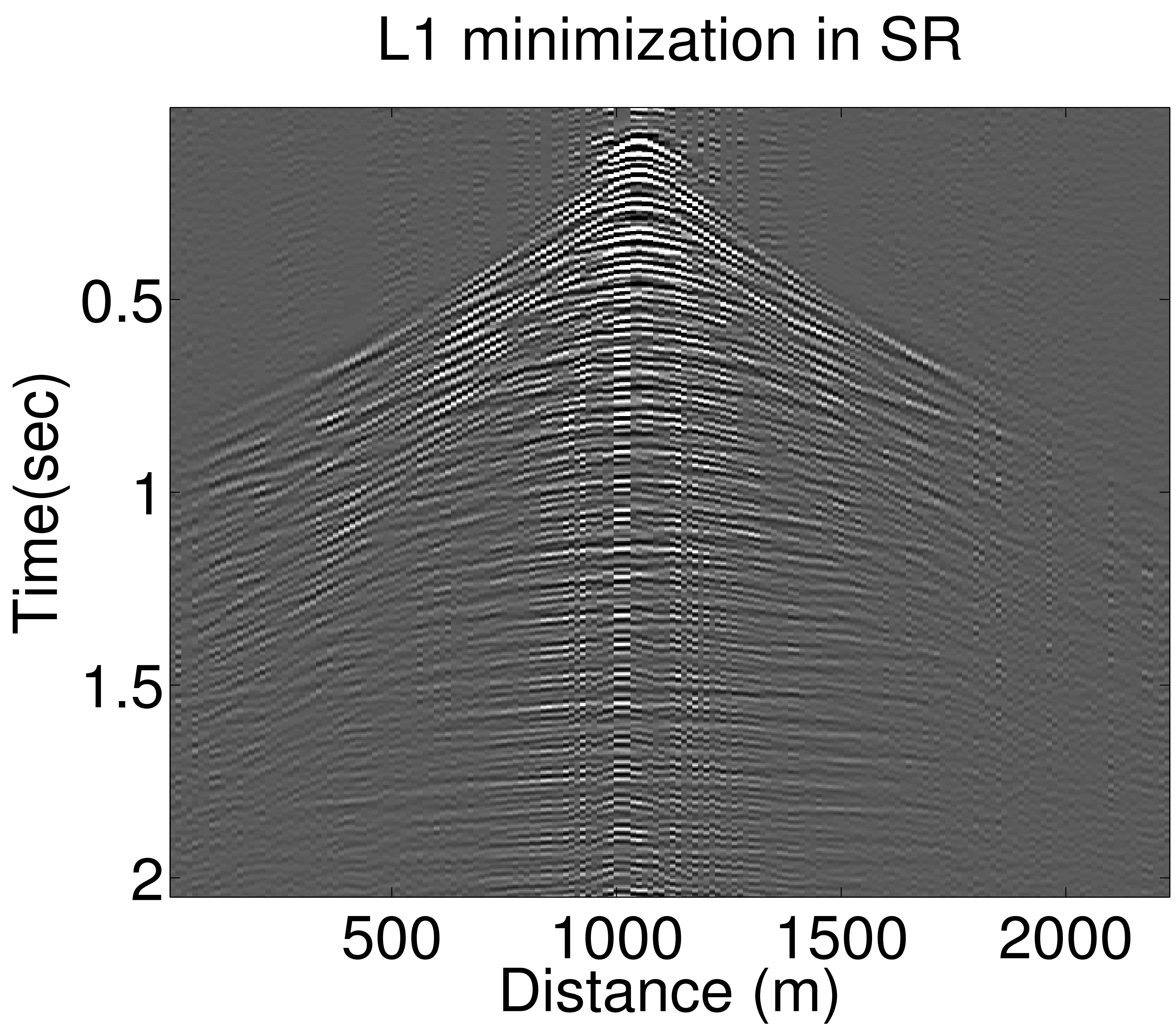
- Similar to the SR domain we do the recovery by utilizing frequency slices.
- The adjacent frequency slices have overlapping support in both cases.



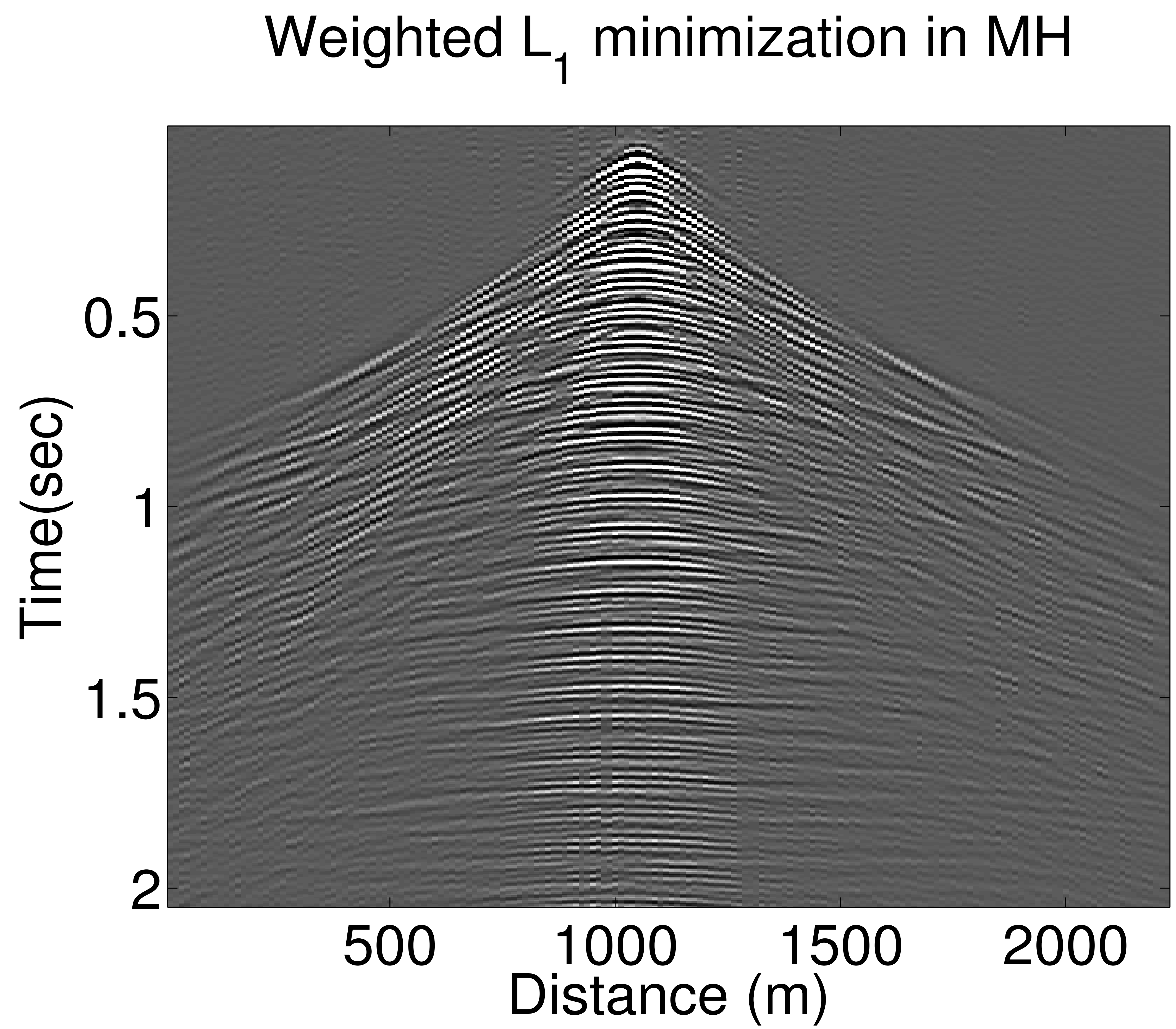
Why does transforming to MH domain help?



Recovery results: ℓ_1 in SR vs weighted ℓ_1 in MH

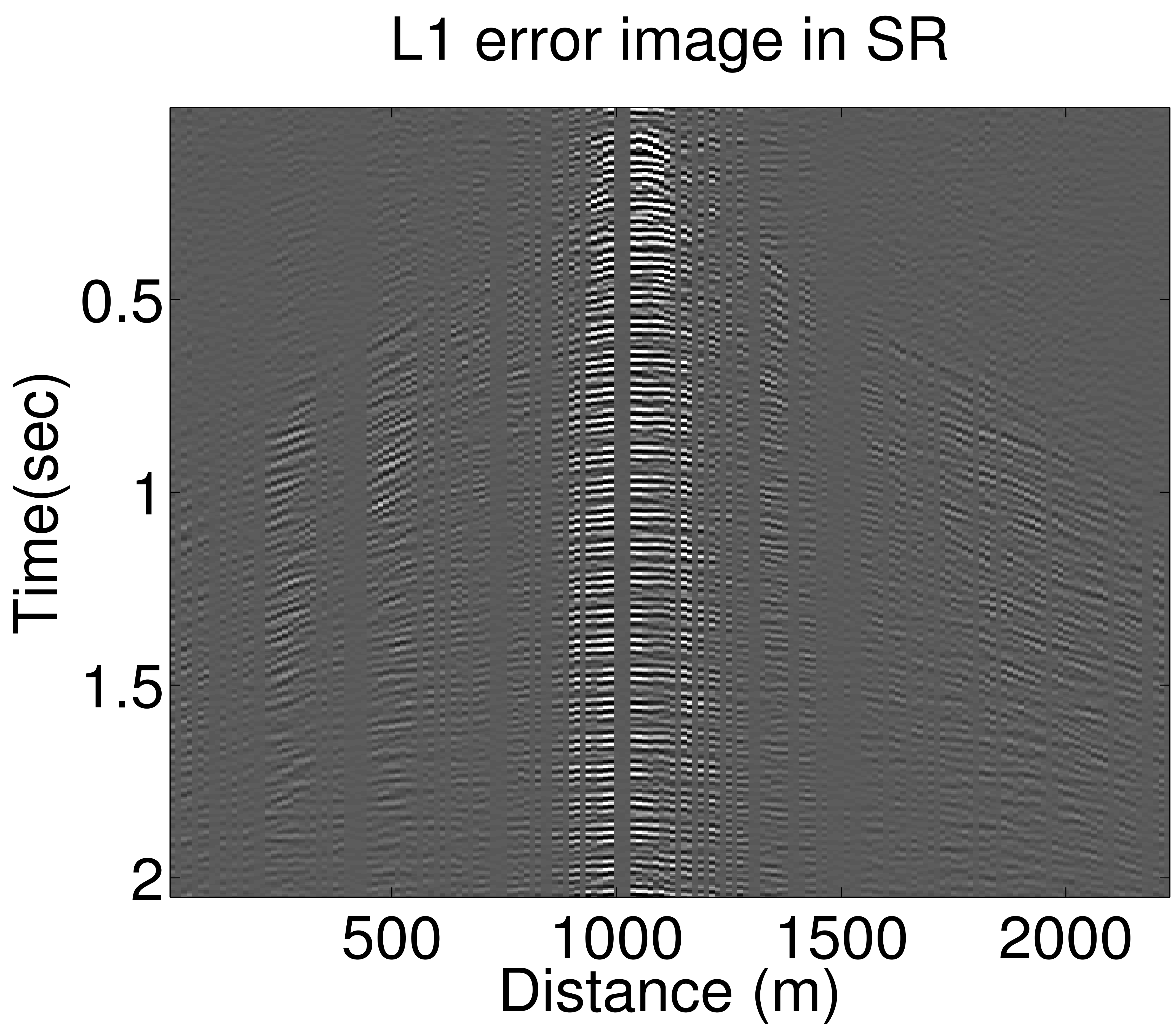


(q) SNR=5.4 dB

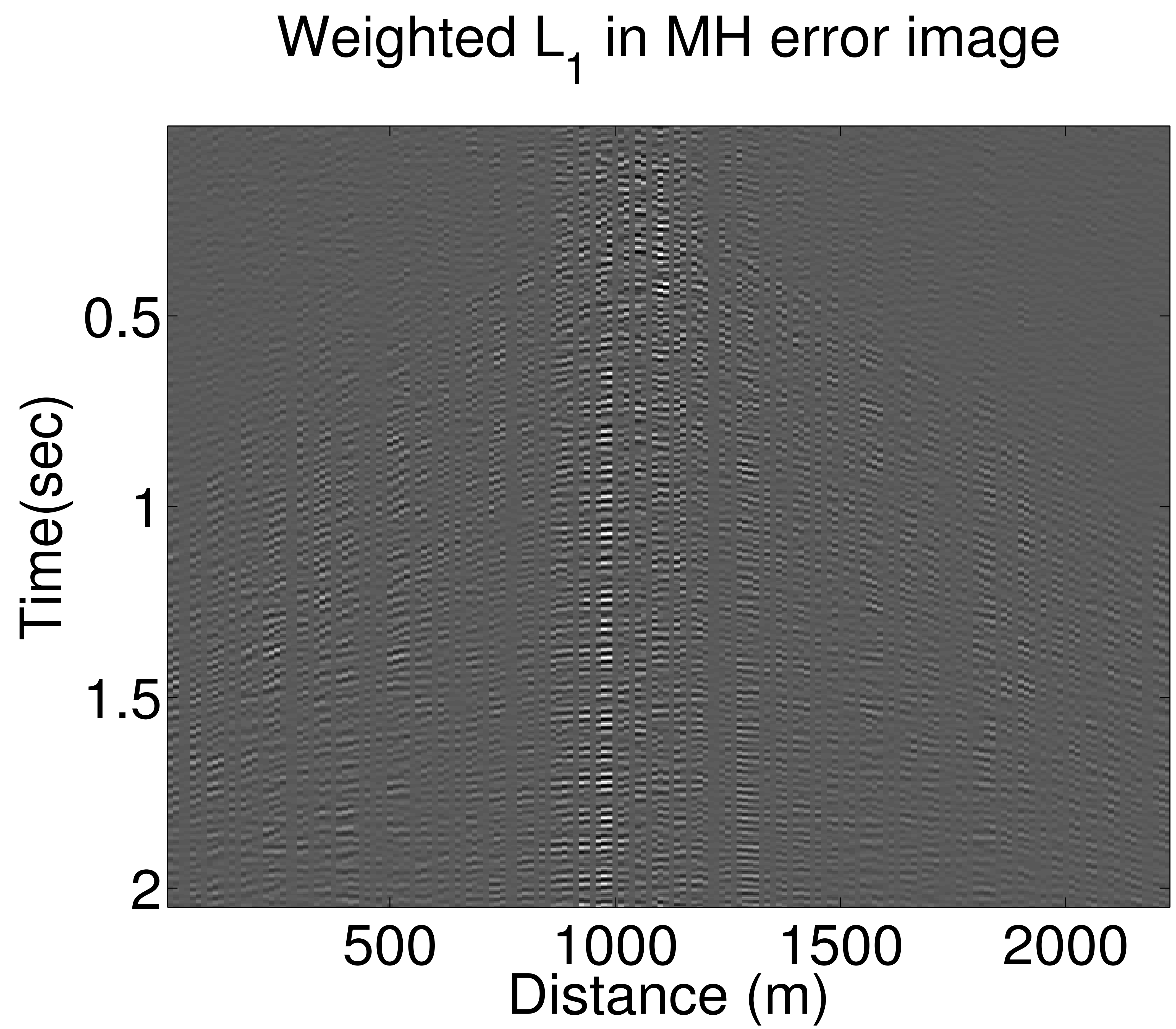


(r) SNR= 12.8 dB

Recovery error: ℓ_1 in SR vs weighted ℓ_1 in MH

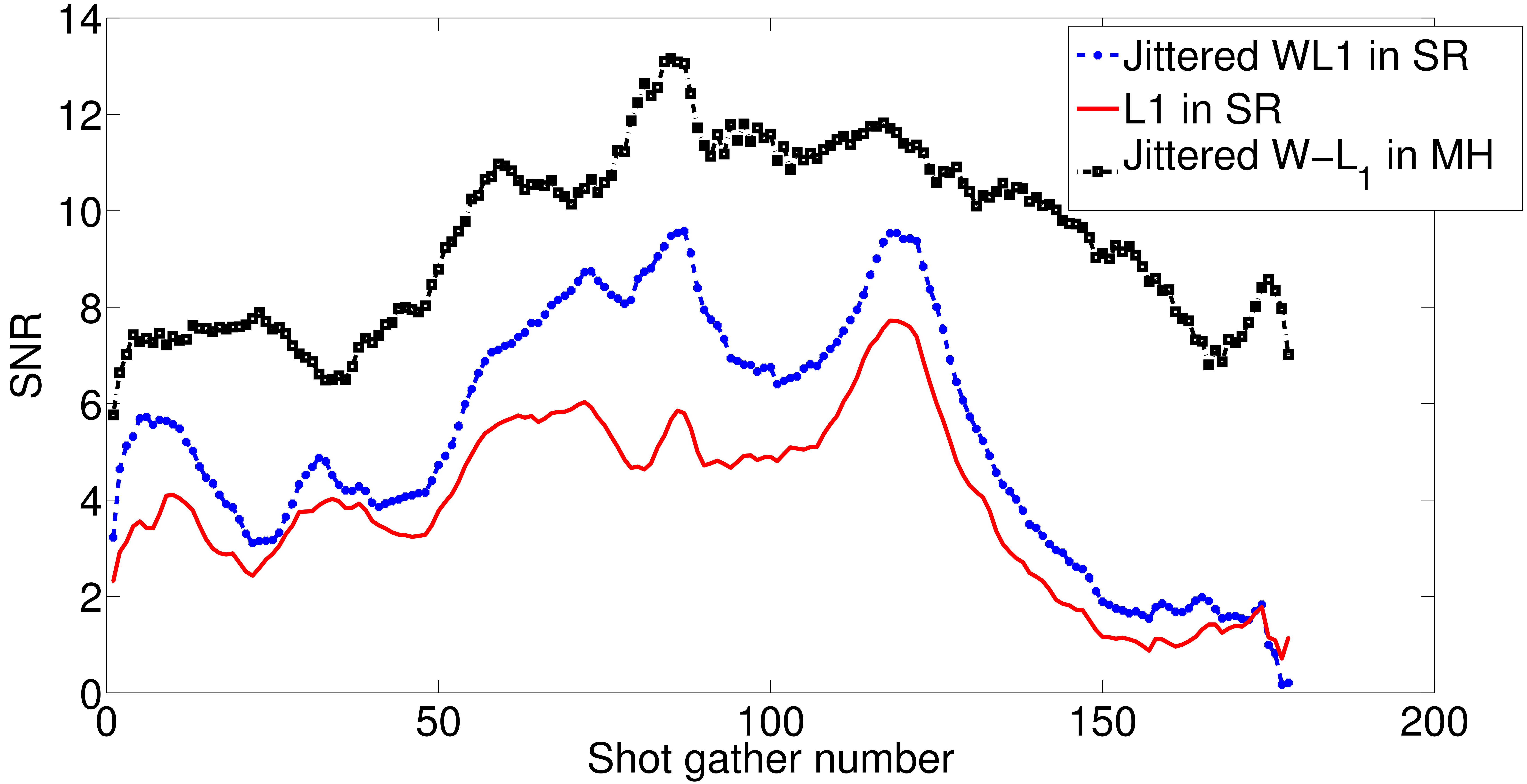


(s) SNR=5.4 dB



(t) SNR= 12.8 dB

Shot gathers results



Monday, December 8, 14

2-stage algorithm

- For each frequency slice first, use a fast algorithm to get a rough estimate of the data in the time domain.

2-stage algorithm

- For each frequency slice first, use a fast algorithm to get a rough estimate of the data in the time domain.
- Transform the estimation to the curvelet domain.

2-stage algorithm

- For each frequency slice first, use a fast algorithm to get a rough estimate of the data in the time domain.
- Transform the estimation to the curvelet domain.
- Use the estimate in curvelet domain to improve the recovery results.

Approximate message passing (AMP)

- We use AMP in Fourier domain for the first stage.

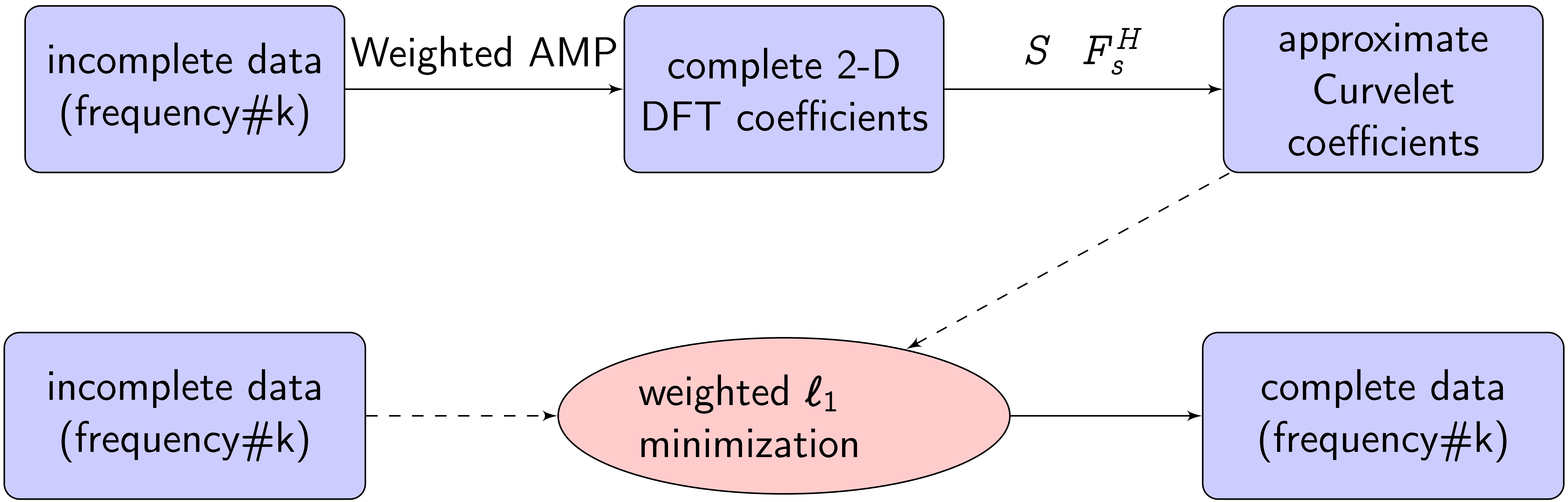
Approximate message passing (AMP)

- We use AMP in Fourier domain for the first stage.
- AMP starts from an initial x^0 and iteratively goes by

$$\begin{aligned} x^{t+1} &= \eta(x^t + A^* z^t; \tau^t), \\ z^t &= y - Ax^t + \delta^{-1} z^{t-1} \left(\frac{\#\{|x^{t-1} + A^* z^{t-1}| > \tau^{t-1}\}}{N} \right). \end{aligned} \tag{1}$$

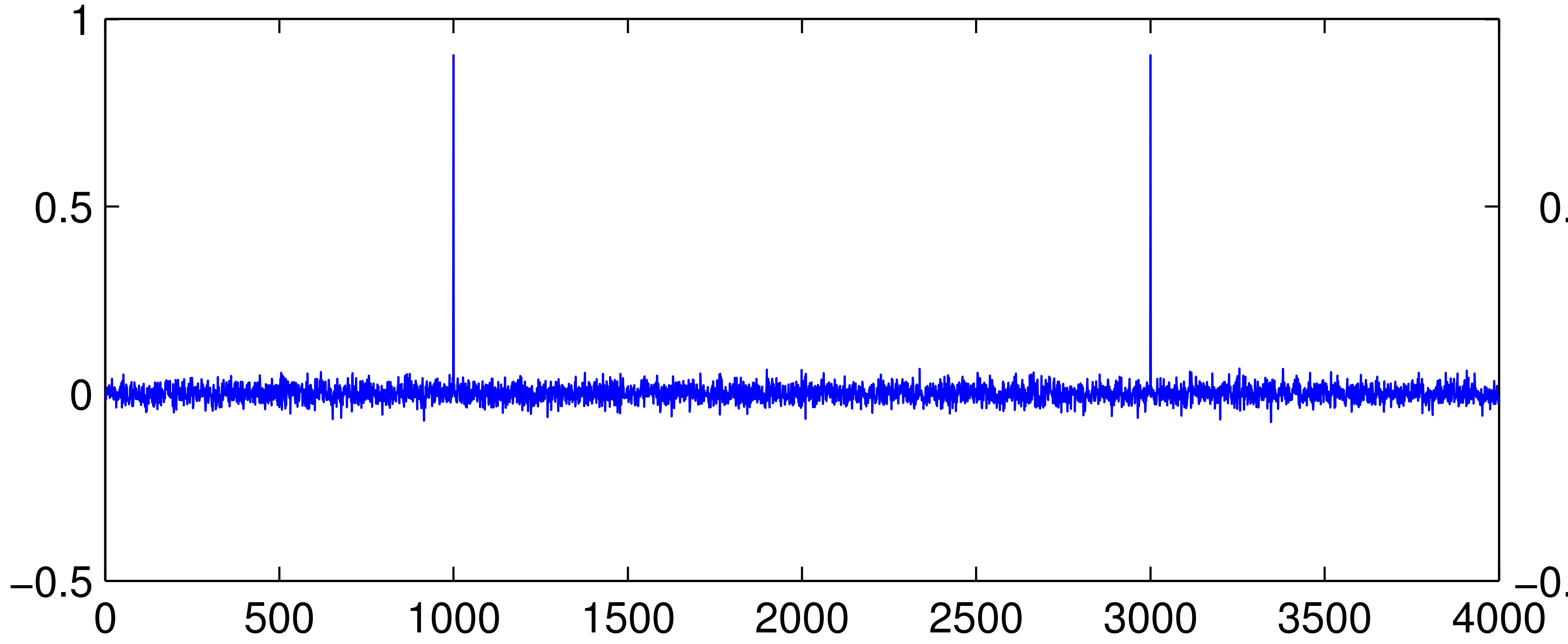
- η is the soft thresholding function $[\eta(x; s)]_j = sign(x_j)(|x_j| - s)_+$

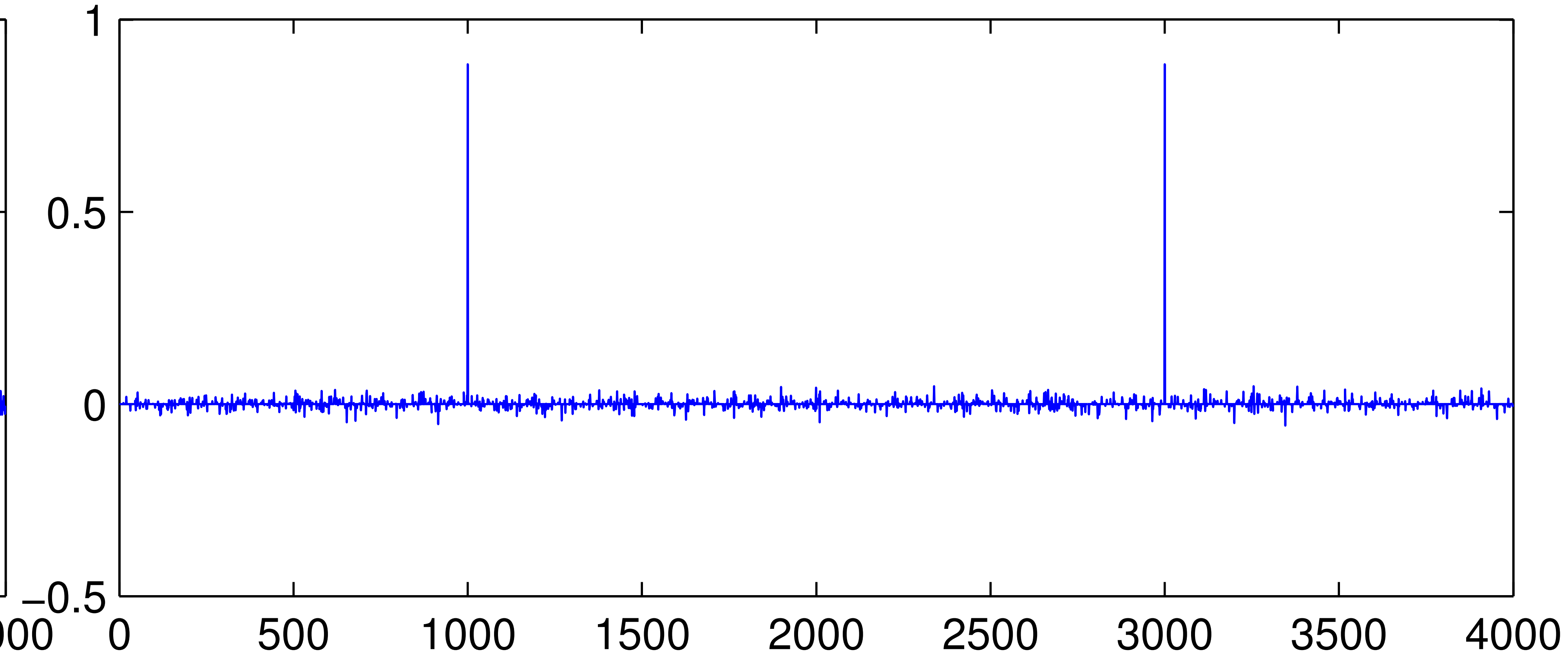
The 2-stage algorithm WAMP+weighted ℓ_1



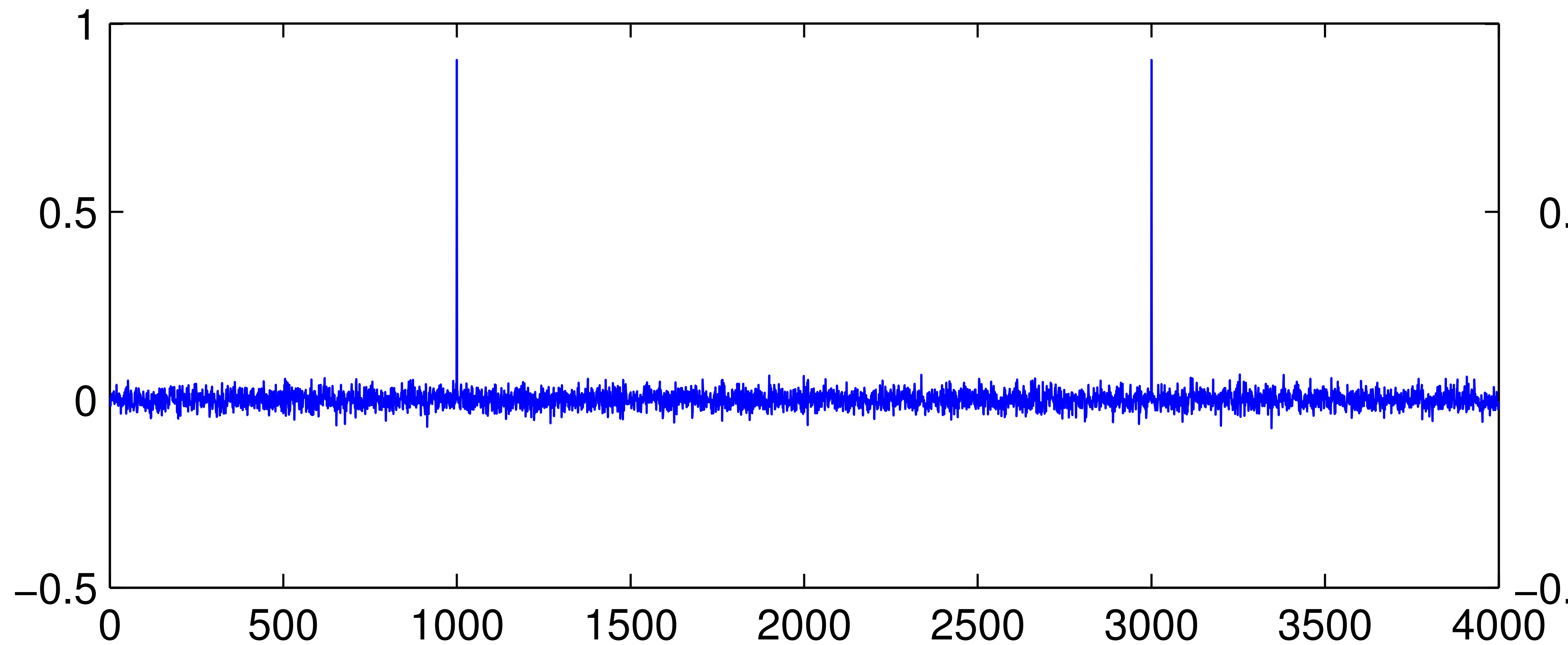
Simple example with a 2-sparse signal, iteration $t = 1$

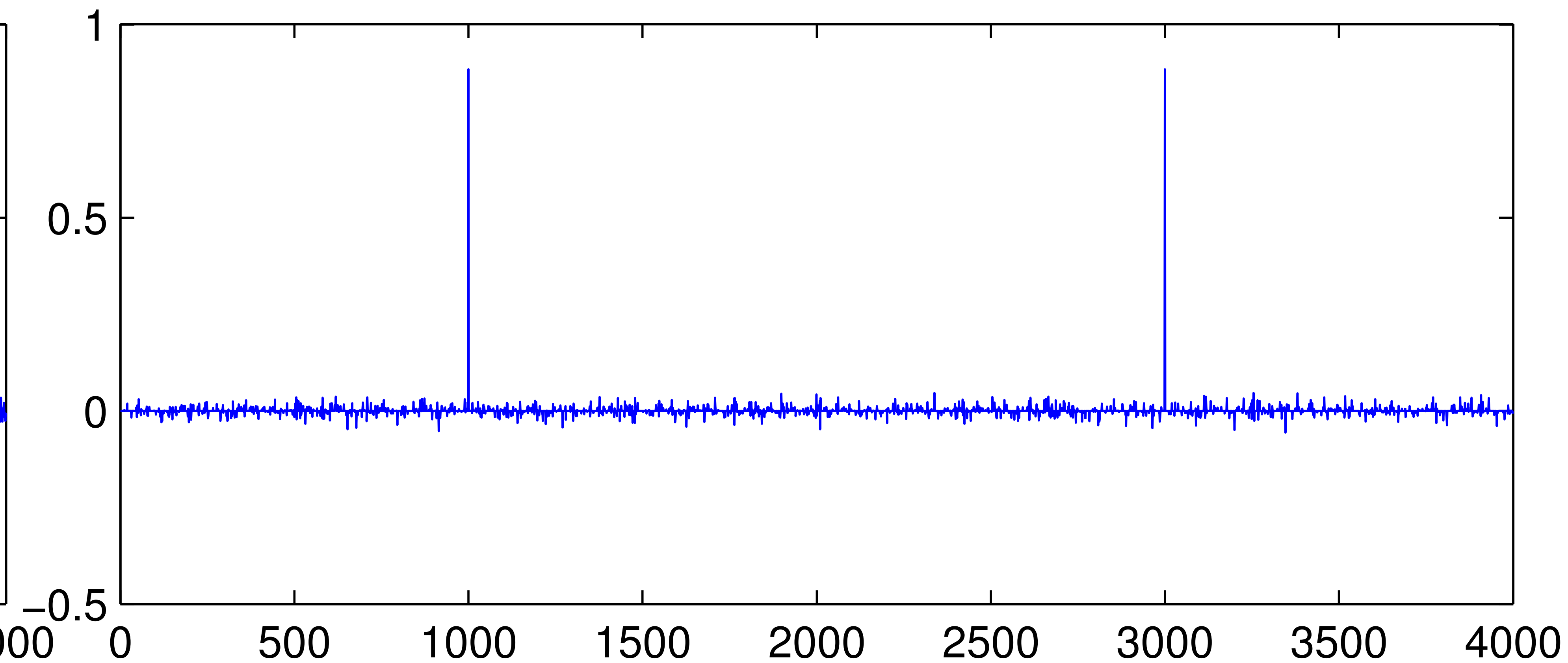
Soft thresholding: $x^t + A^* z^t$ $\qquad$ $\eta(x^t + A^* z^t; \tau^t)$





AMP: $x^t + A^*(z^t + \alpha^t z^{t-1})$ $\qquad$ $\eta(x^t + A^*(z^t + \alpha^t z^{t-1}); \tau^t)$

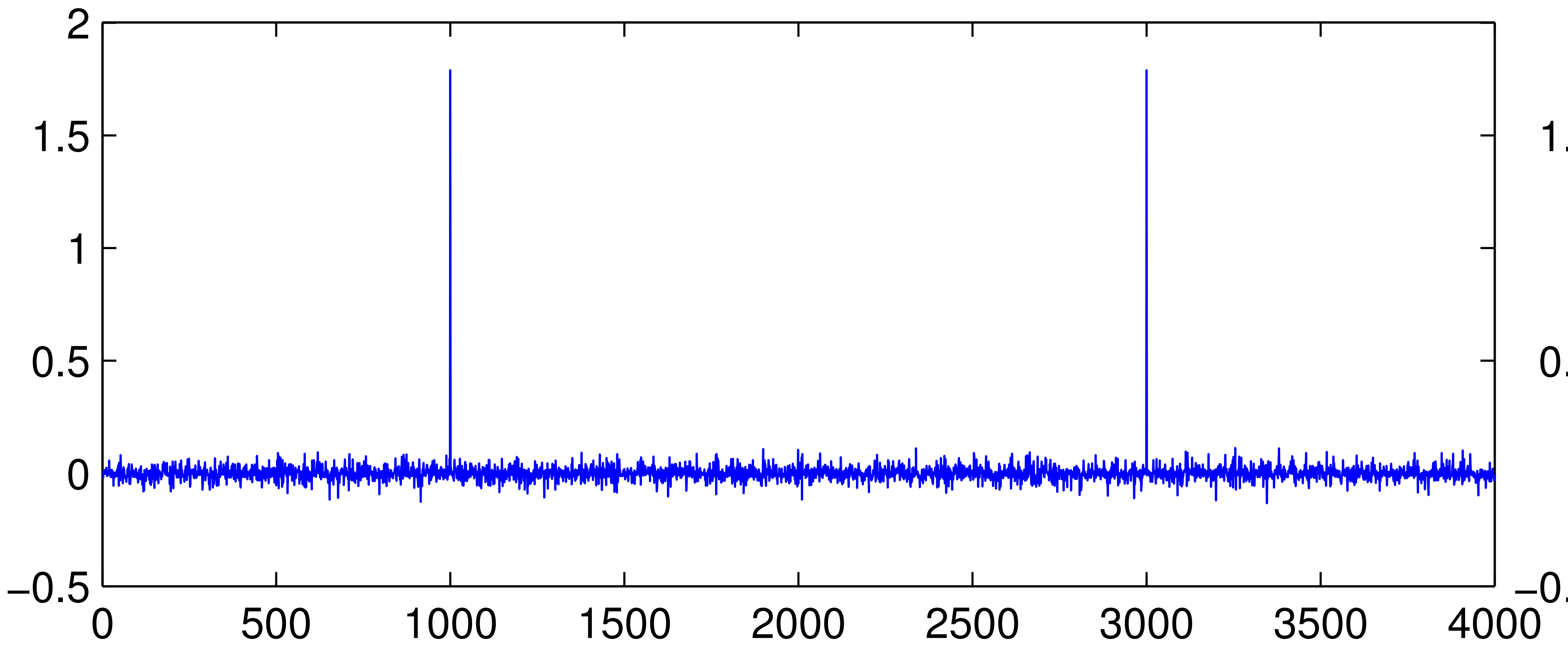


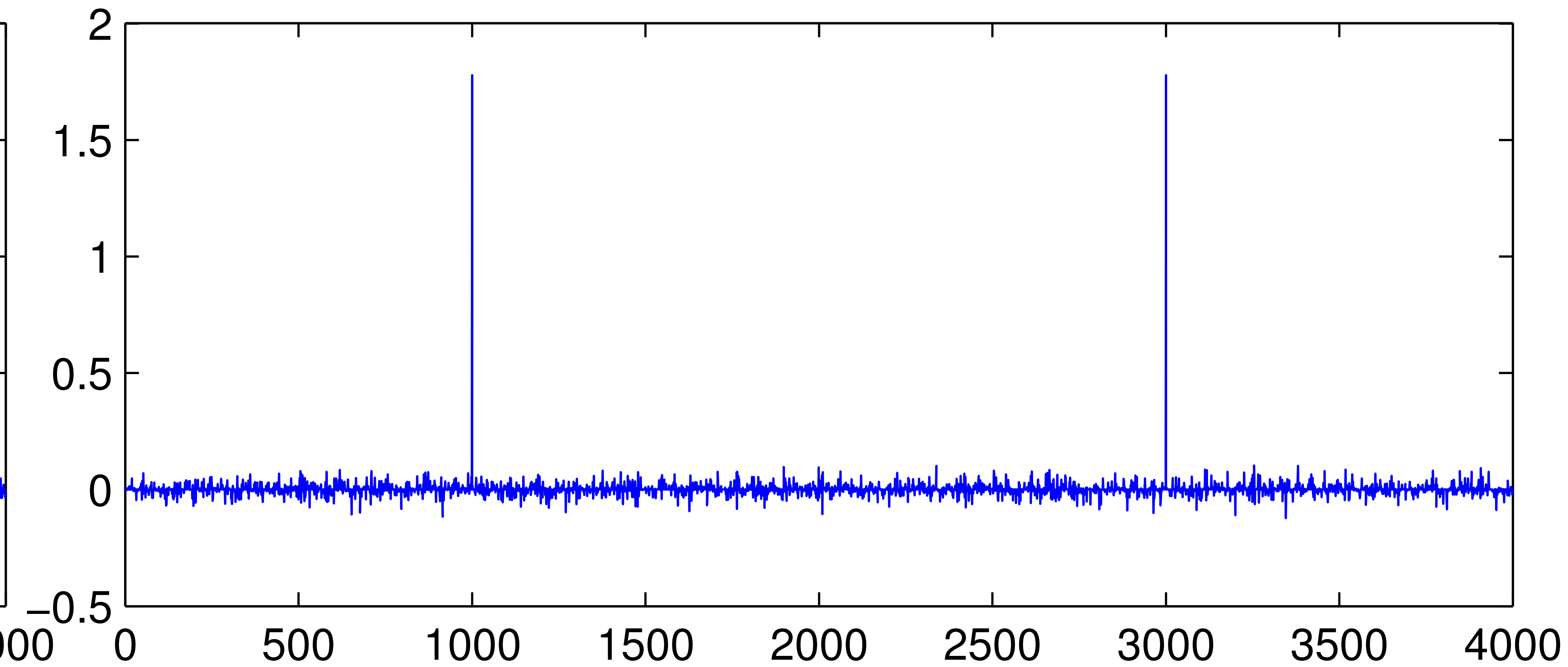


Monday, December 8, 14

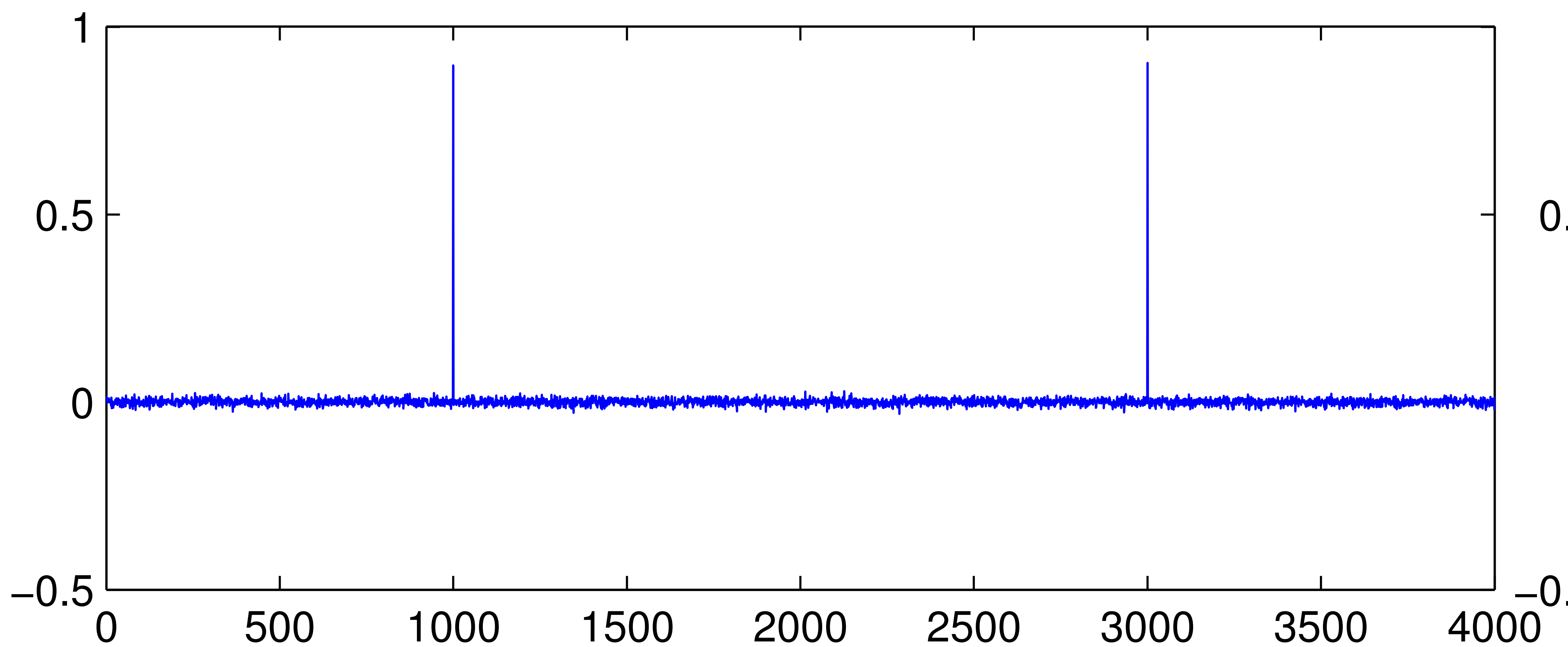
Iteration $t = 2$

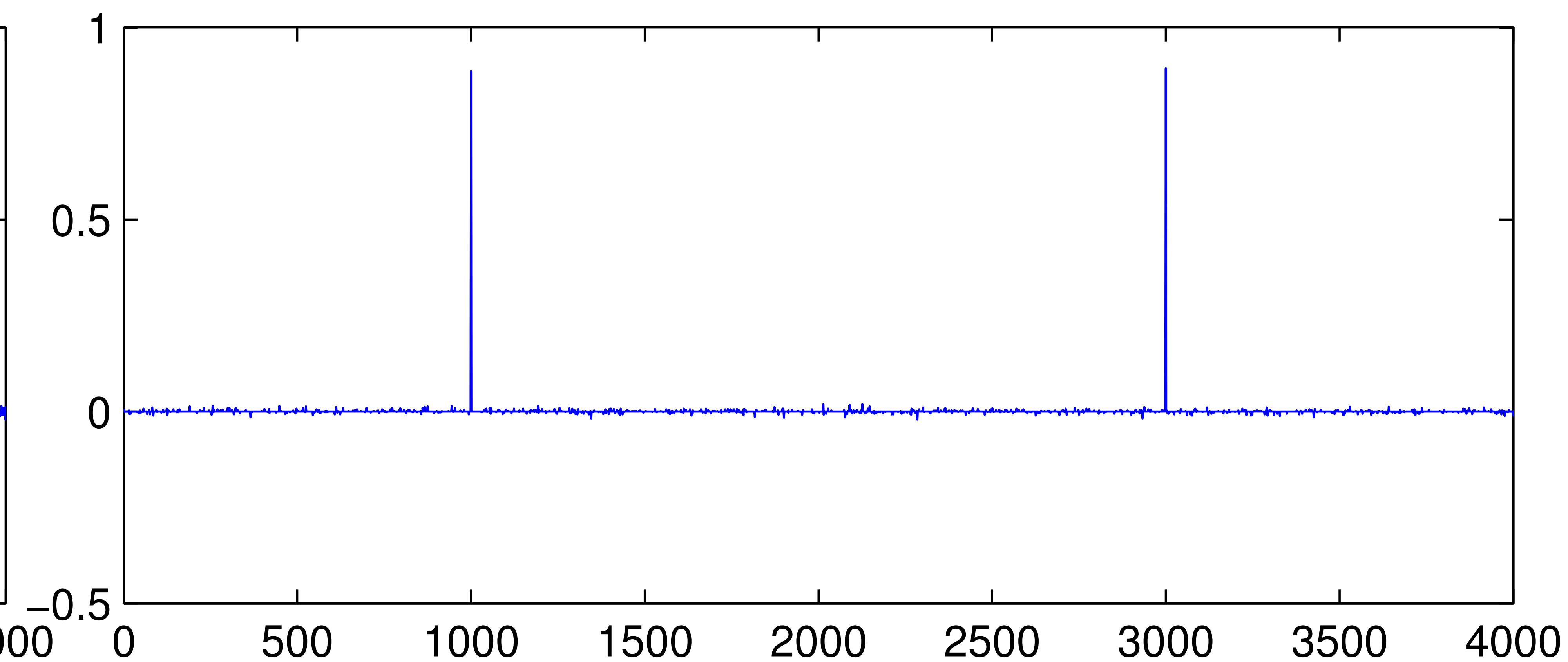
Soft thresholding: $x^t + A^* z^t$ $\qquad$ $\eta(x^t + A^* z^t; \tau^t)$





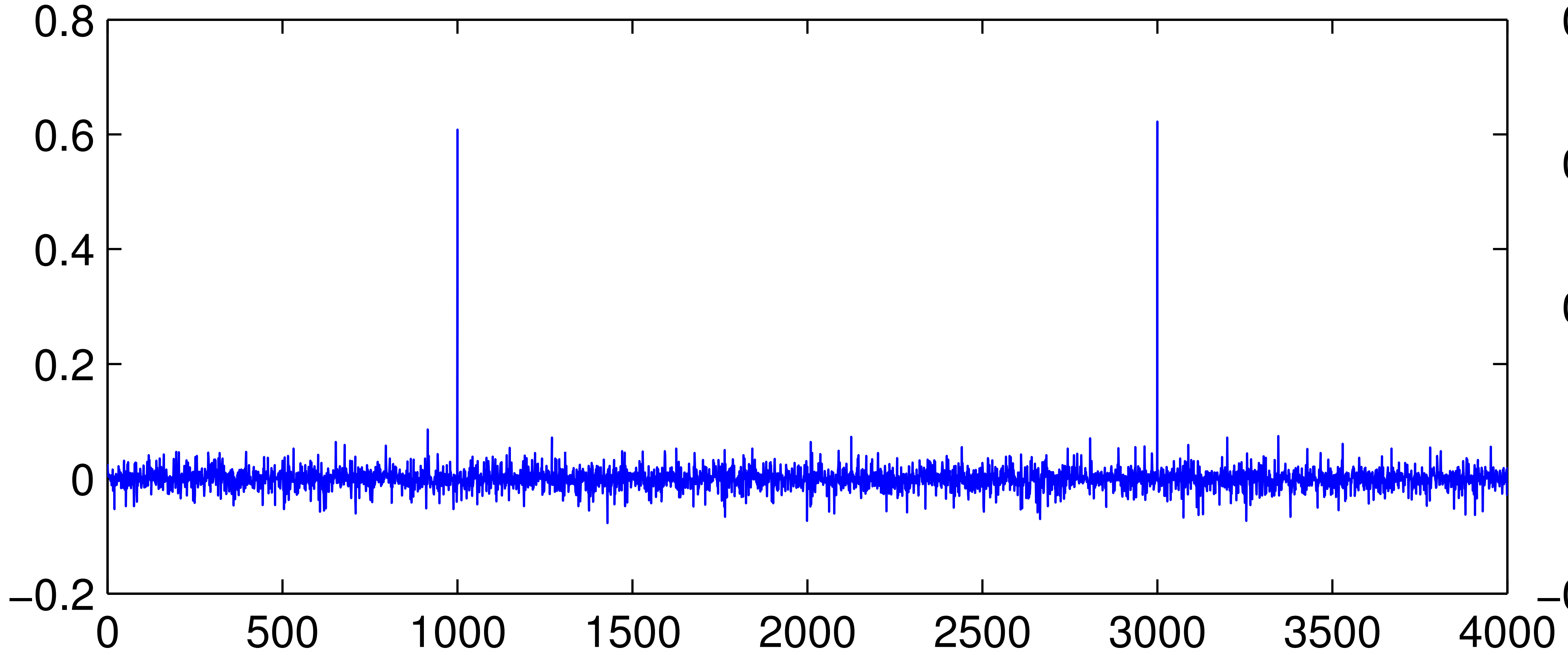
AMP: $x^t + A^*(z^t + \alpha^t z^{t-1})$ $\qquad$ $\eta(x^t + A^*(z^t + \alpha^t z^{t-1}); \tau^t)$

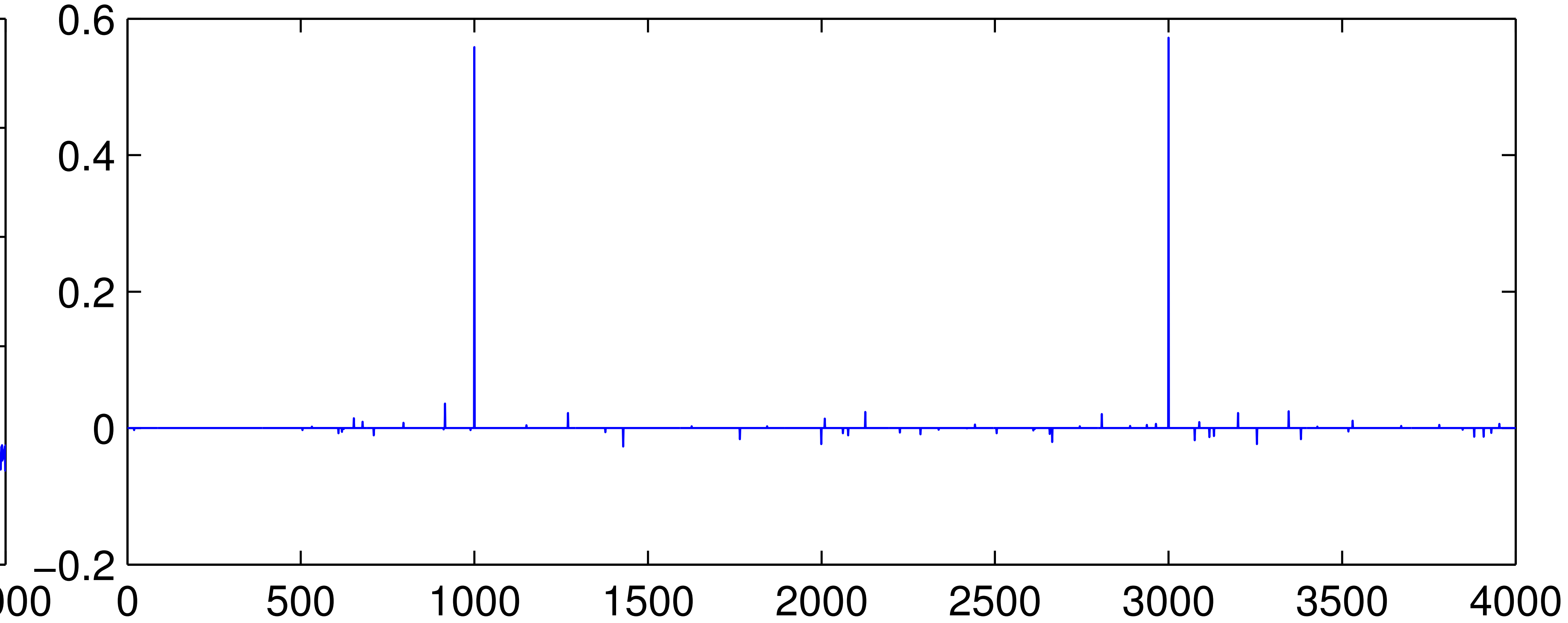




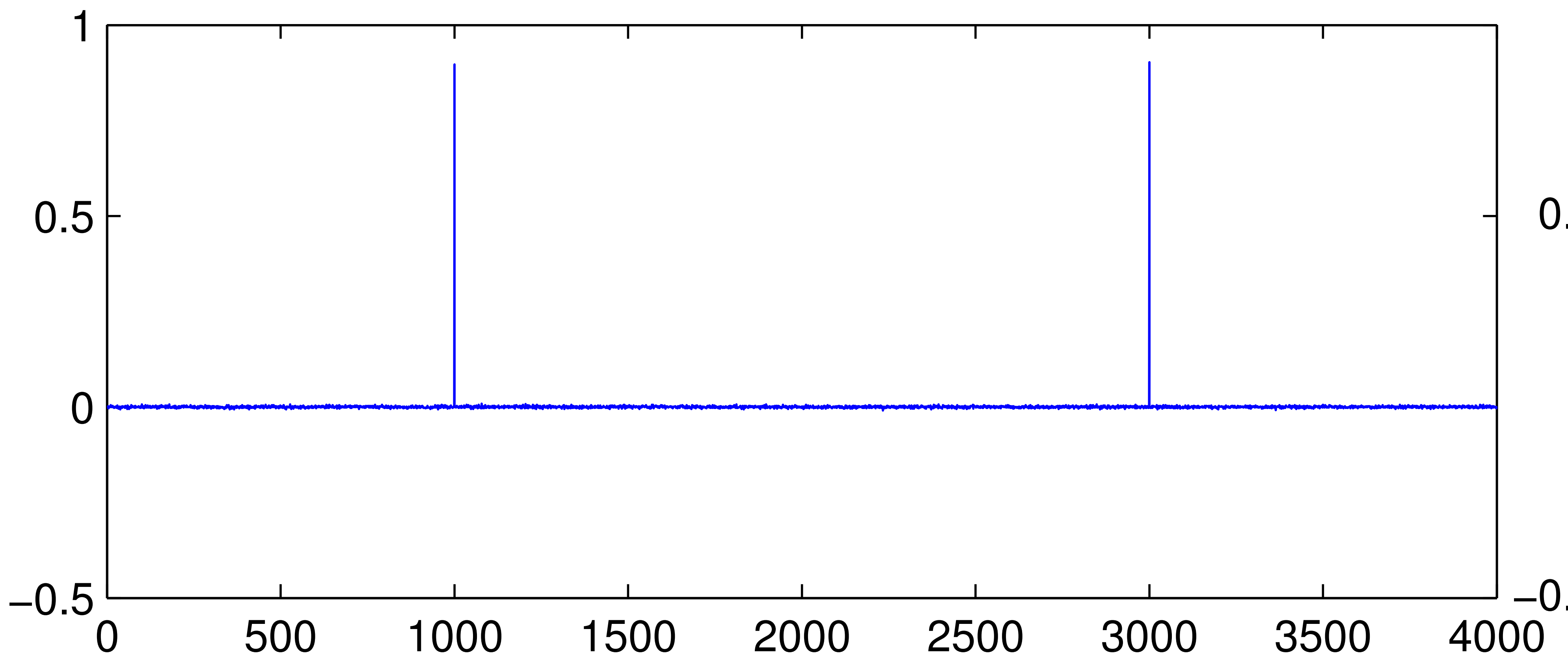
Iteration $t = 3$

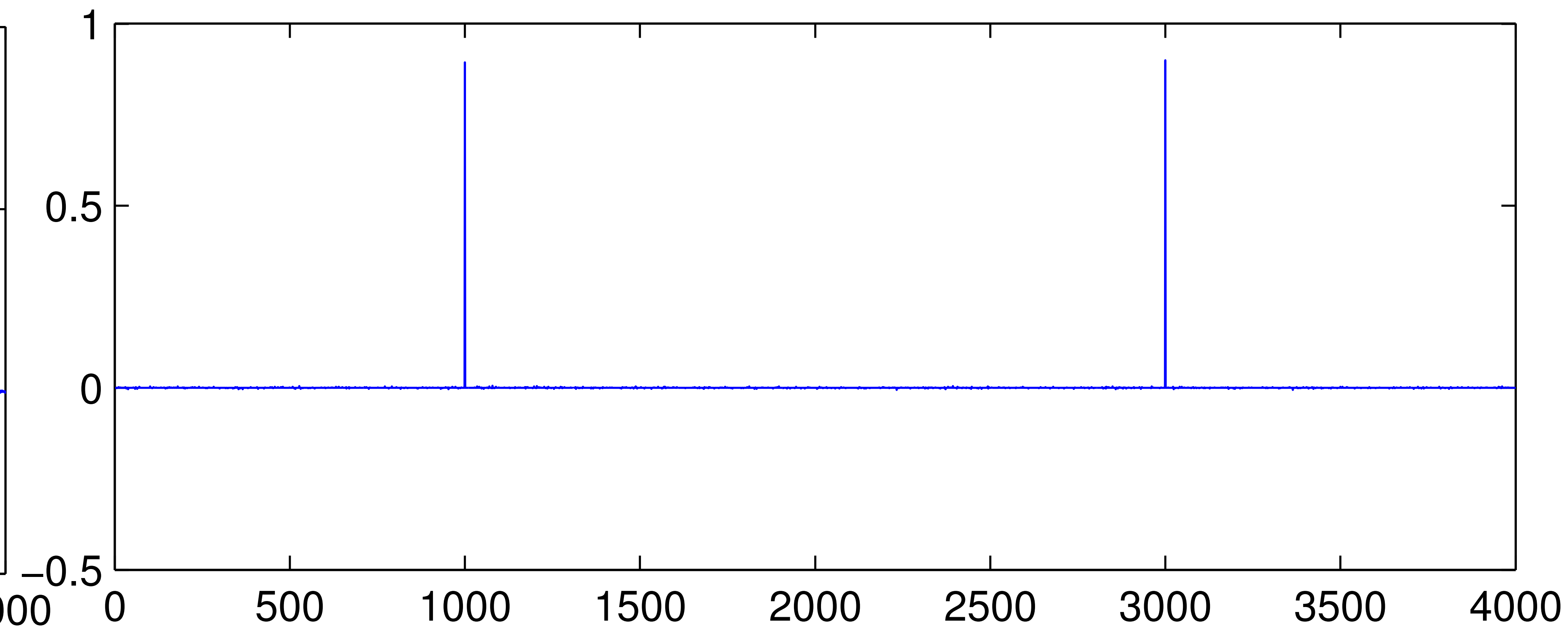
Soft thresholding: $x^t + A^* z^t$ $\qquad$ $\eta(x^t + A^* z^t; \tau^t)$





AMP: $x^t + A^*(z^t + \alpha^t z^{t-1})$ $\qquad$ $\eta(x^t + A^*(z^t + \alpha^t z^{t-1}); \tau^t)$

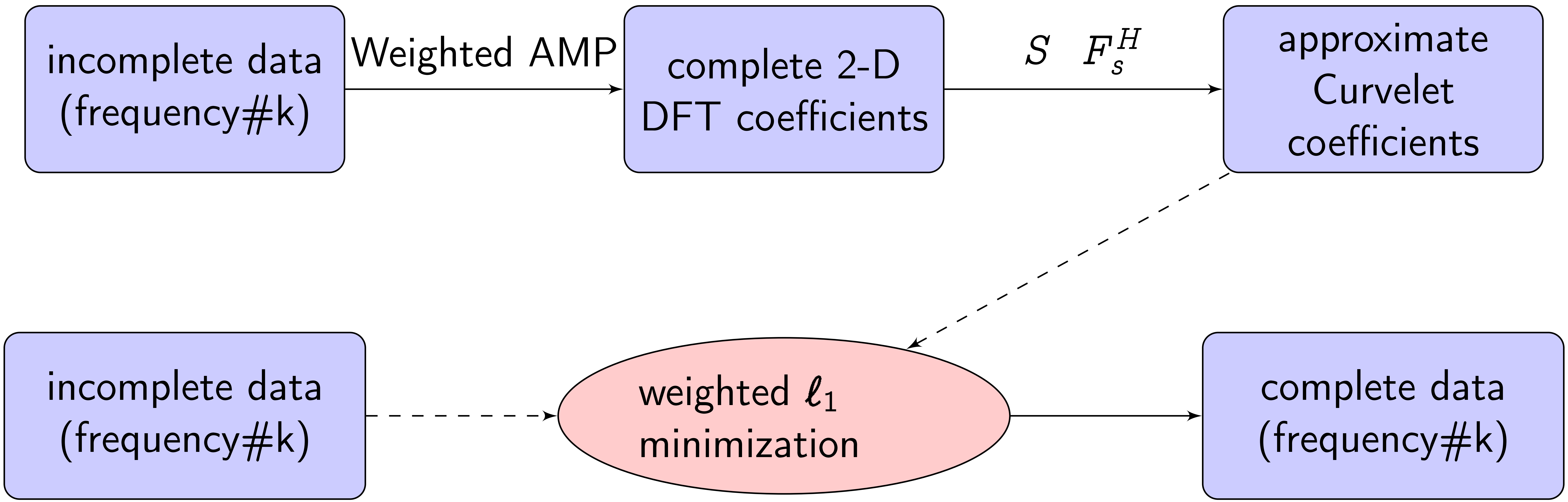




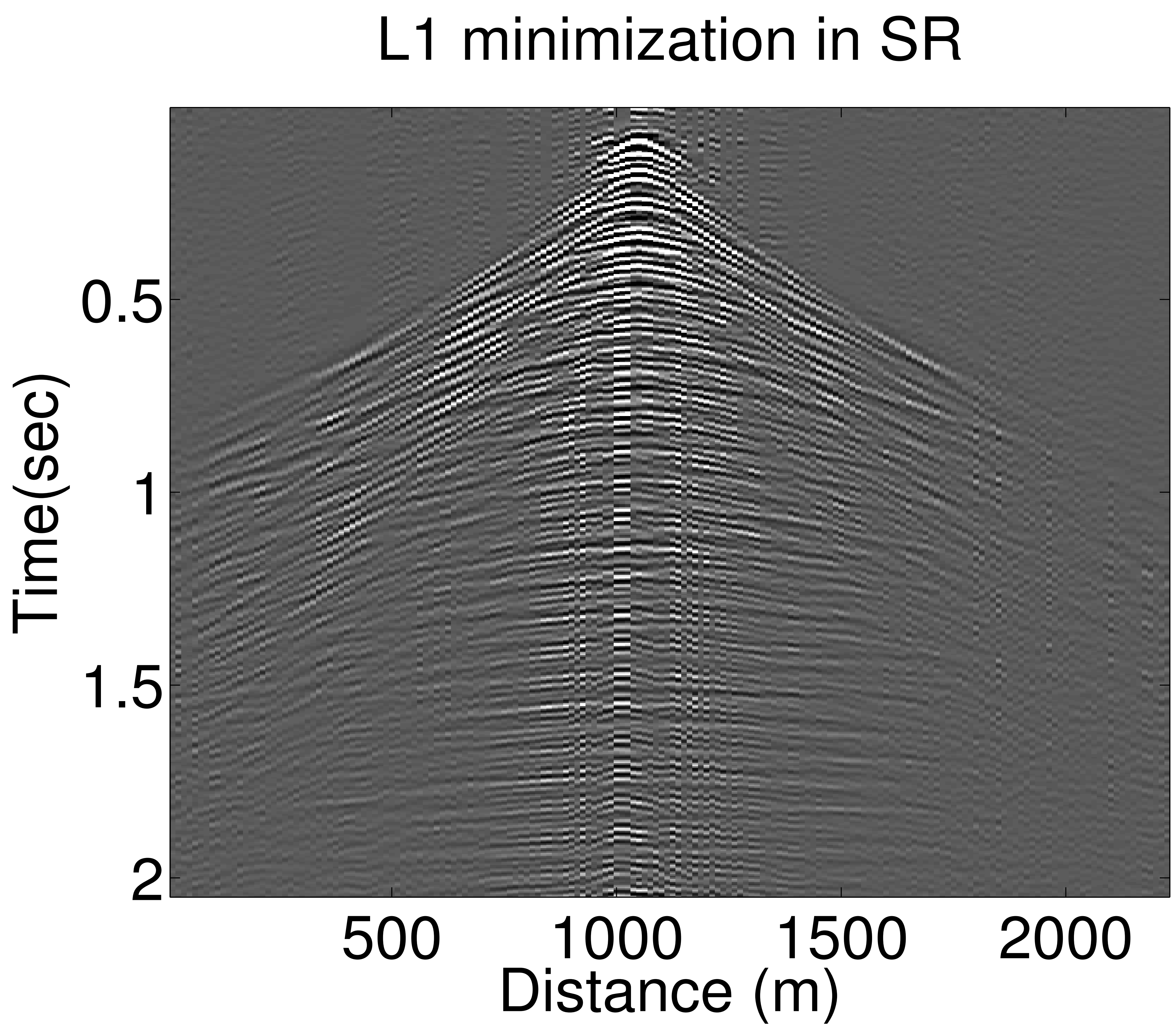
AMP and WAMP for seismic trace interpolation

- AMP is a delicate algorithm that just works for certain types of measurements.
- We can't use curvelets with AMP.
- Instead we use 2-D DFT matrix in the source-receiver domain.
- Then $b = RMF_s^H F_s f$, where F_s is a 2-D DFT matrix.

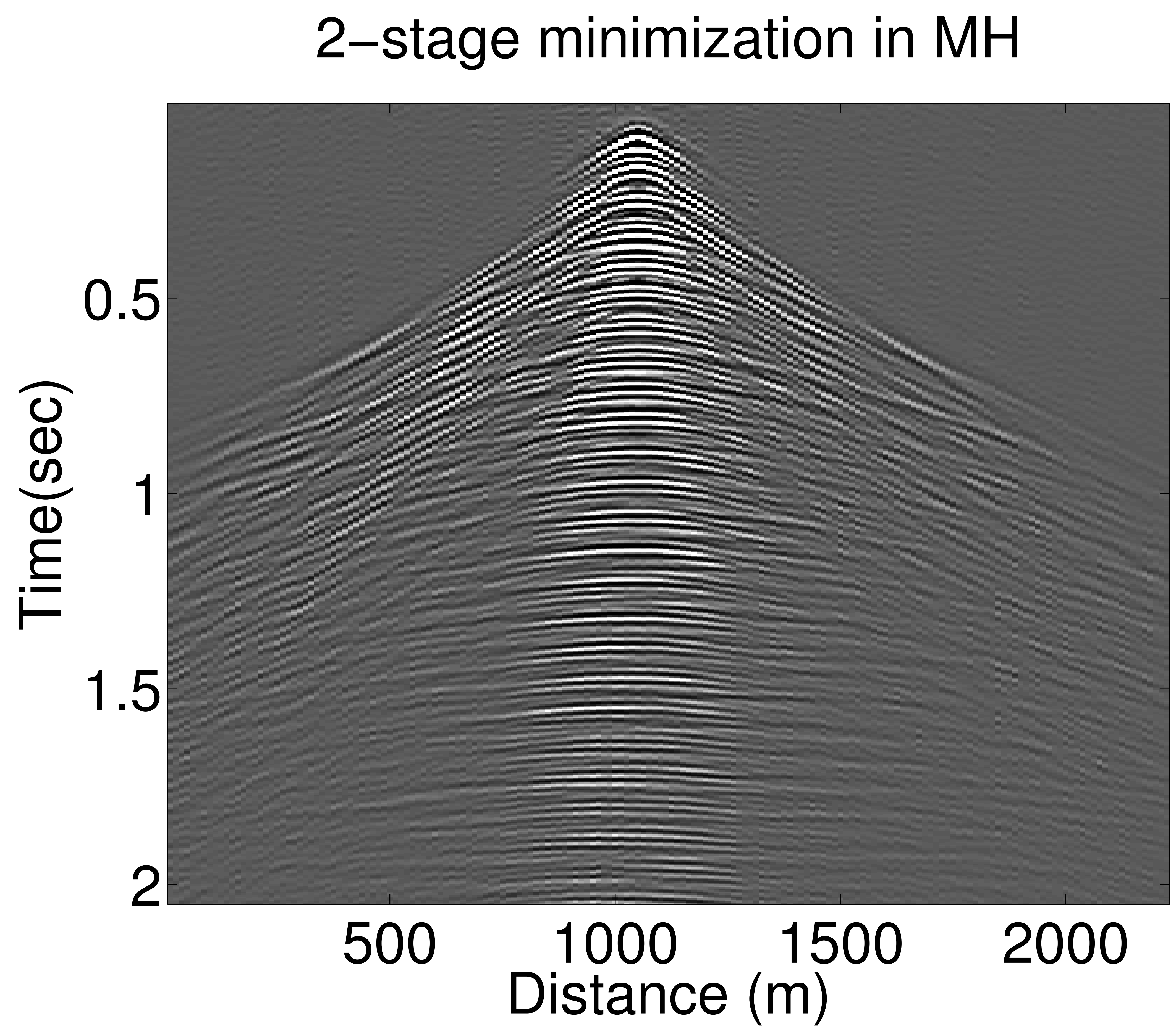
Flowchart of the 2-stage algorithm WAMP+weighted ℓ_1



ℓ_1 vs 2-stage WAMP+weighted ℓ_1

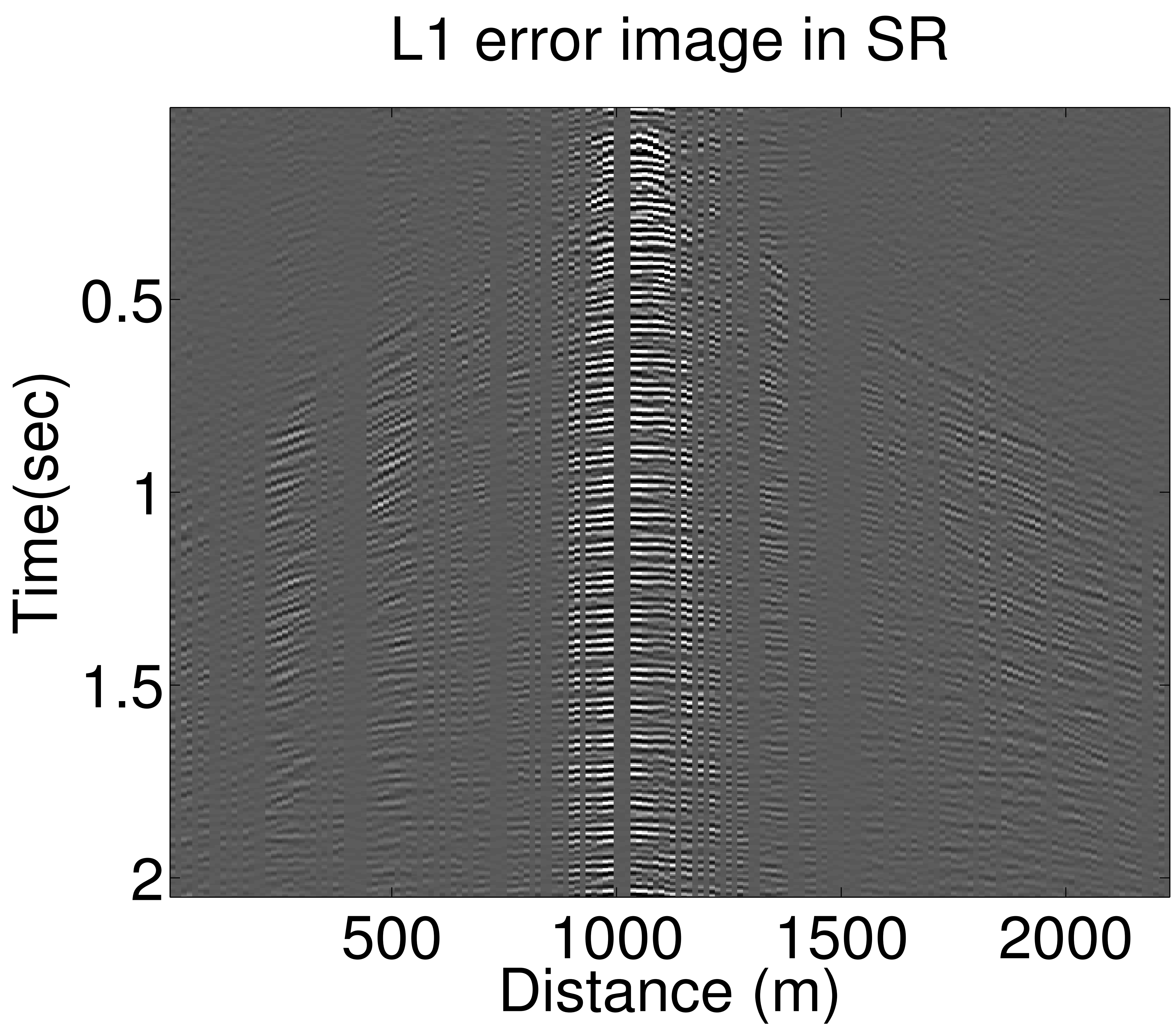


(u) SNR= 5.3 dB

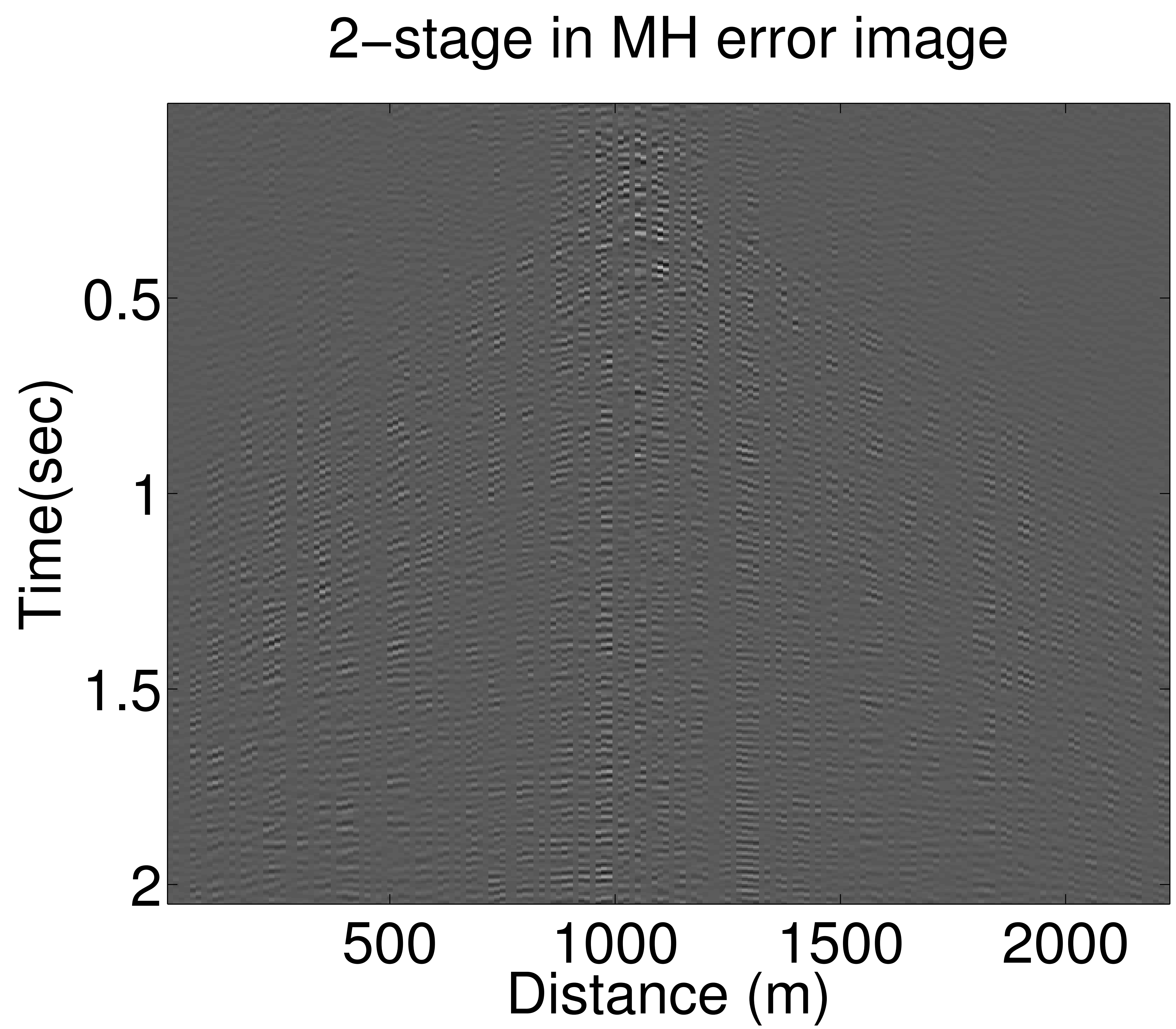


(v) SNR= 14.8 dB

Results of the 2-stage algorithm

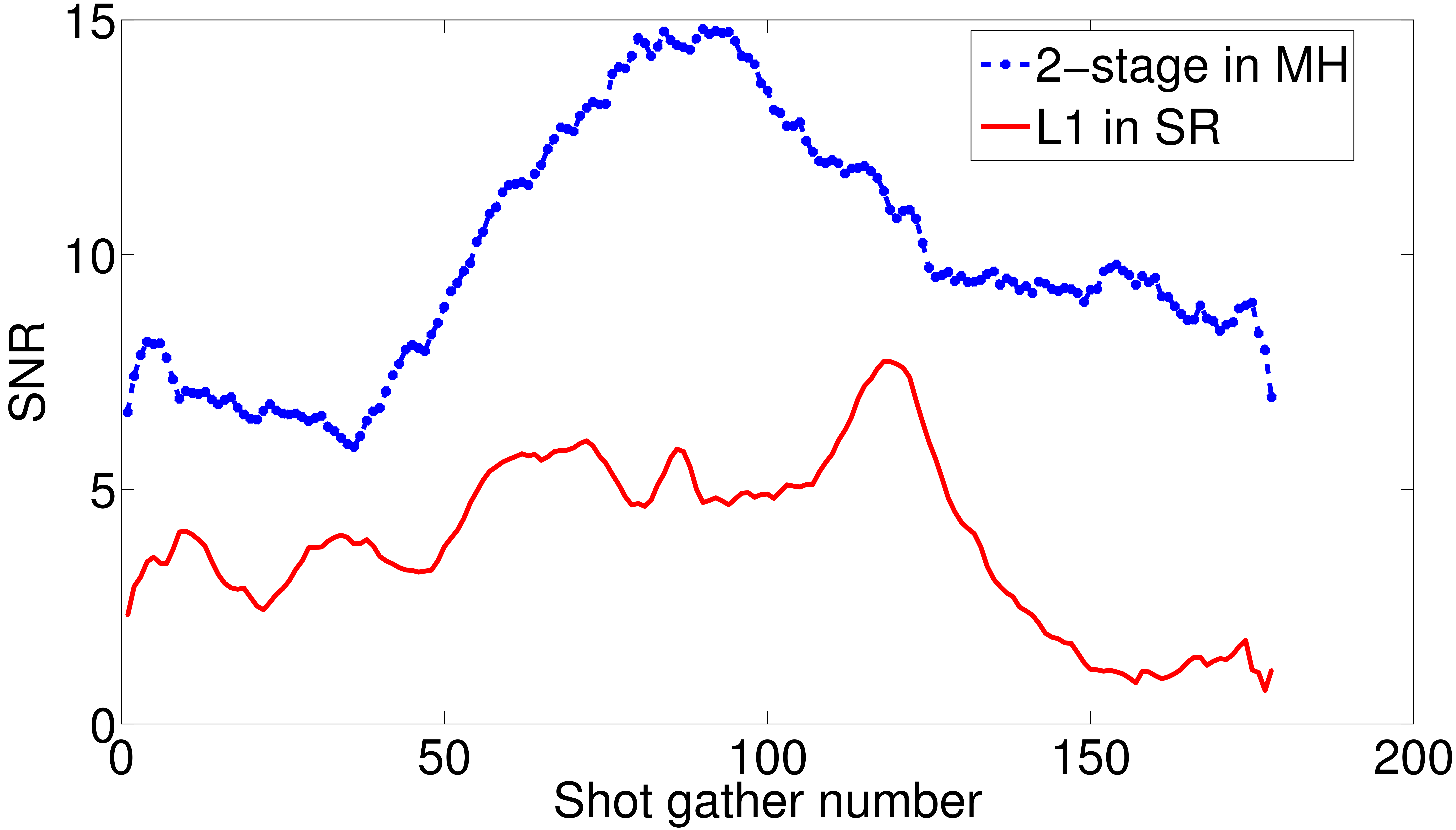


(w) SNR= 5.3 dB

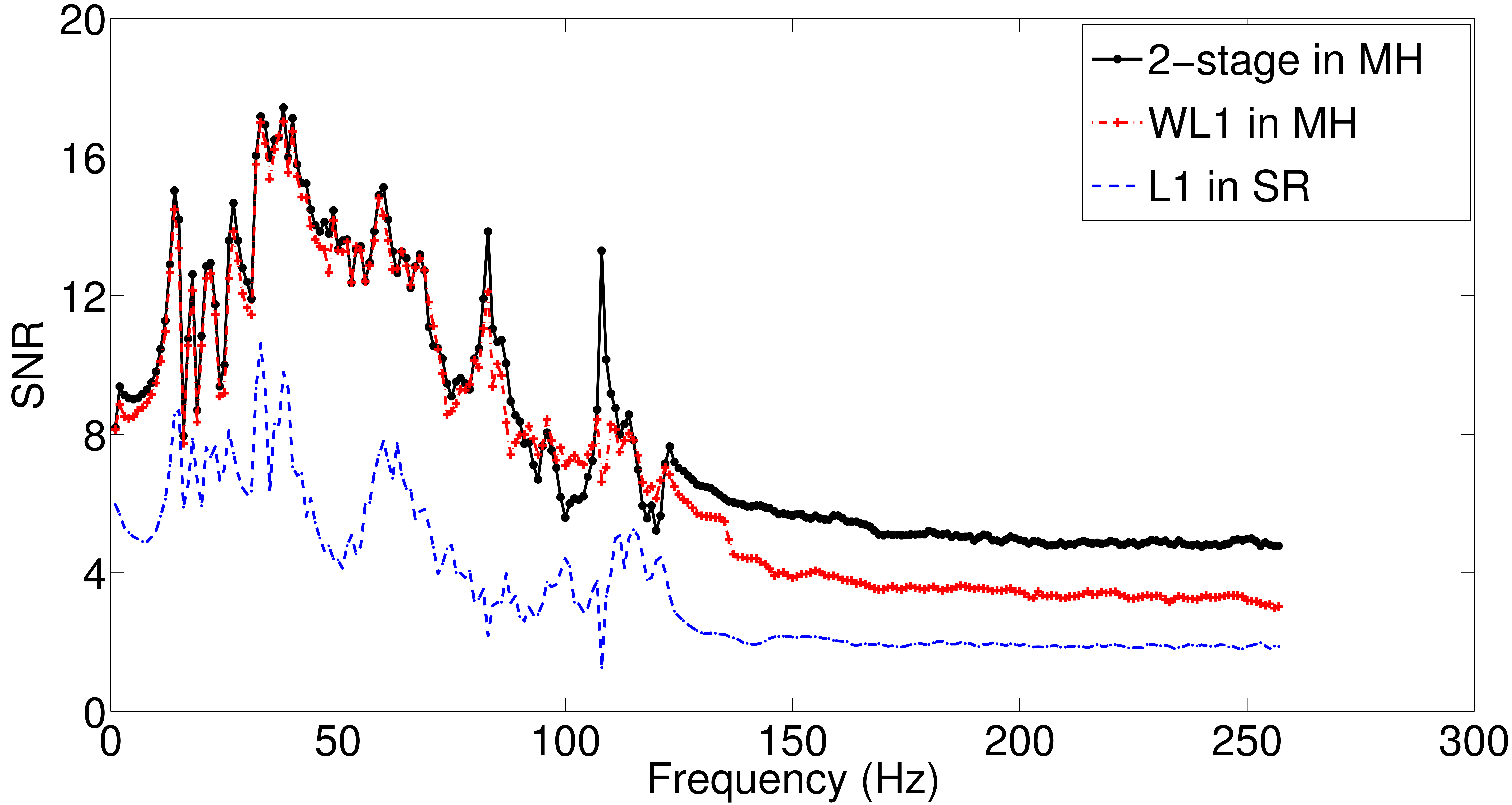


(x) SNR= 14.8 dB

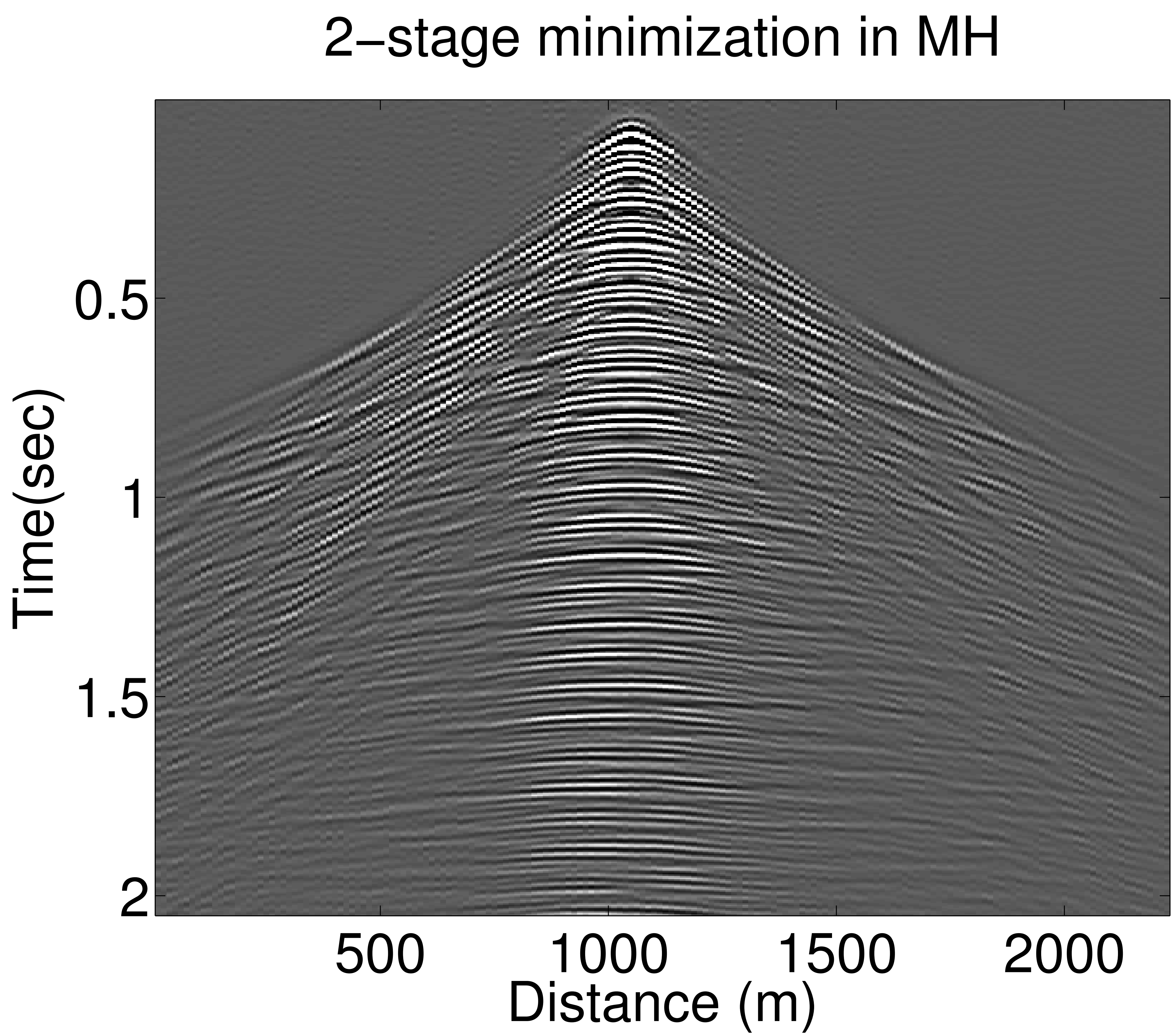
Results of the 2-stage algorithm

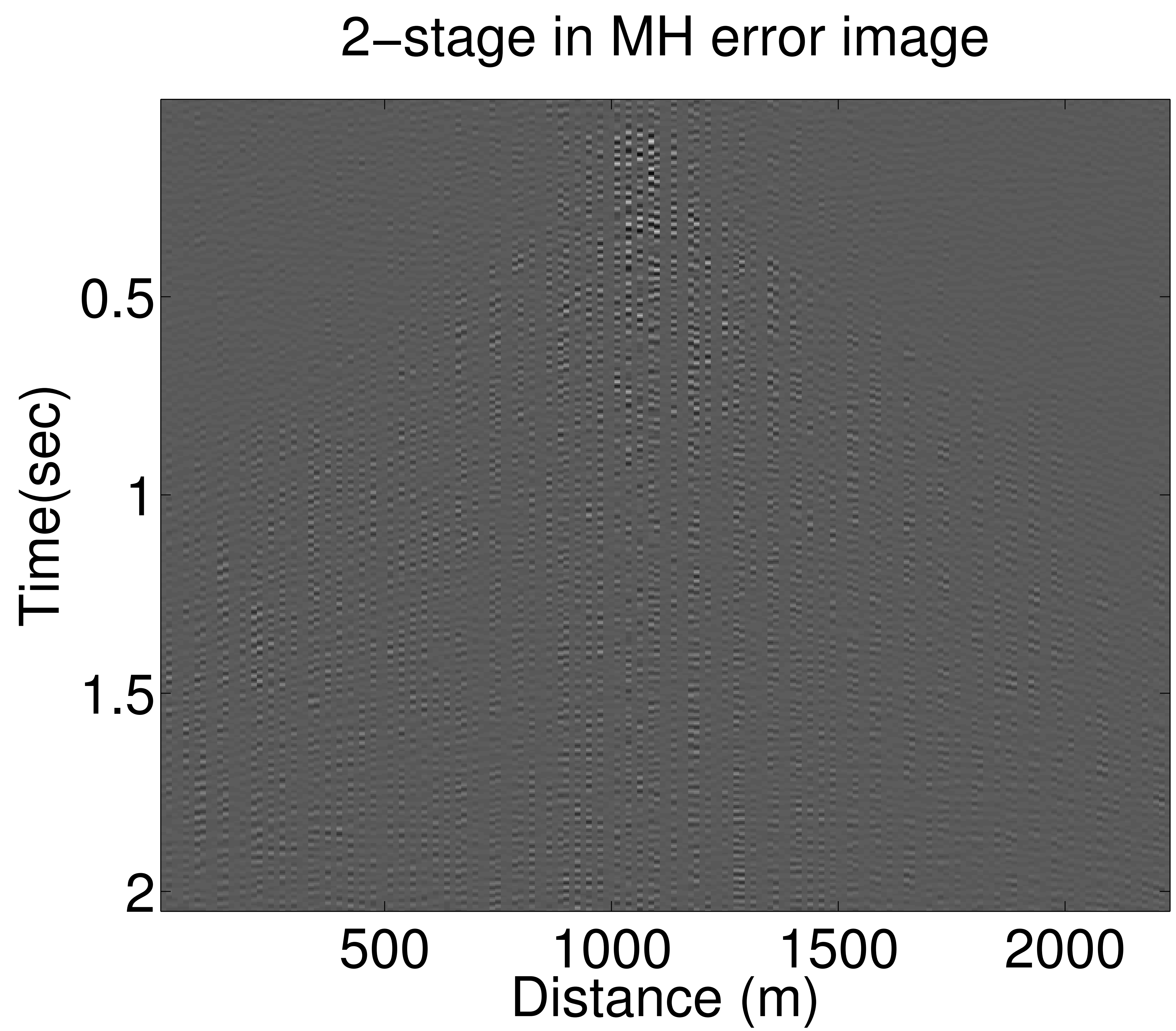


Comparison of recovery results

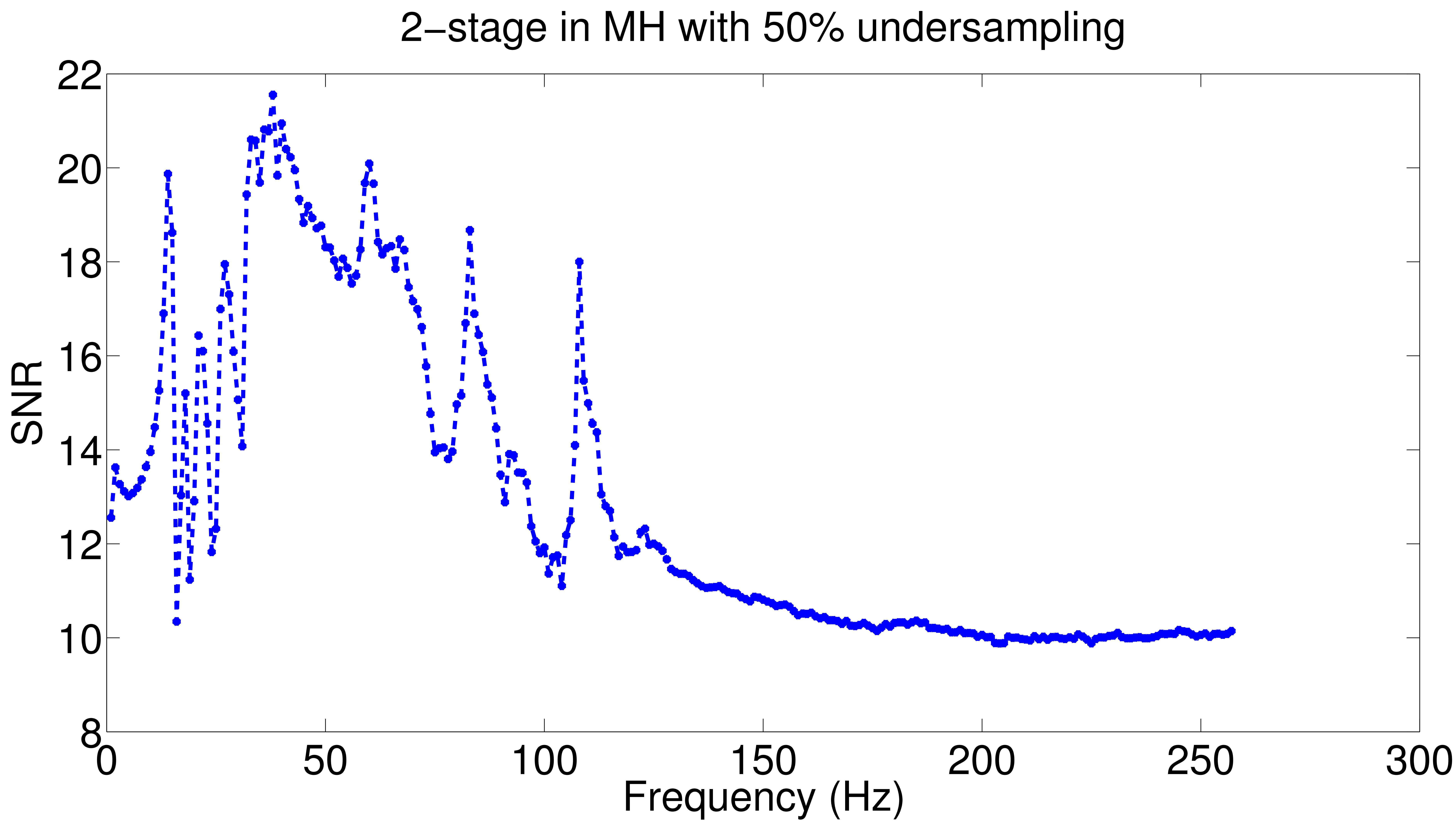


50% undersampling (19.6 dB)





50% undersampling (19.6 dB)



Conclusions

- ℓ_1 recovery for seismic trace interpolation starts failing once we we increase the undersampling rate.

Conclusions

- ℓ_1 recovery for seismic trace interpolation starts failing once we we increase the undersampling rate.
- However, it has a lot of room for improvements.

Conclusions

- ℓ_1 recovery for seismic trace interpolation starts failing once we we increase the undersampling rate.
- However, it has a lot of room for improvements.
- Using low cost techniques, we can have reasonable results in interpolating hugely undersampled data.

Conclusions

- ℓ_1 recovery for seismic trace interpolation starts failing once we we increase the undersampling rate.
- However, it has a lot of room for improvements.
- Using low cost techniques, we can have reasonable results in interpolating hugely undersampled data.
- With jittering we control the maximum gap size in the data.

Conclusions

- ℓ_1 recovery for seismic trace interpolation starts failing once we we increase the undersampling rate.
- However, it has a lot of room for improvements.
- Using low cost techniques, we can have reasonable results in interpolating hugely undersampled data.
- With jittering we control the maximum gap size in the data.
- With weighting we use the continuity of data in all dimension.

Conclusions

- ℓ_1 recovery for seismic trace interpolation starts failing once we we increase the undersampling rate.
- However, it has a lot of room for improvements.
- Using low cost techniques, we can have reasonable results in interpolating hugely undersampled data.
- With jittering we control the maximum gap size in the data.
- With weighting we use the continuity of data in all dimension.
- We get better sparse representation once we go from SR domain to MH domain.

Conclusions

- ℓ_1 recovery for seismic trace interpolation starts failing once we we increase the undersampling rate.
- However, it has a lot of room for improvements.
- Using low cost techniques, we can have reasonable results in interpolating hugely undersampled data.
- With jittering we control the maximum gap size in the data.
- With weighting we use the continuity of data in all dimension.
- We get better sparse representation once we go from SR domain to MH domain.
- Finding a fast estimation of the data in the curvelet domain improves the results significantly. Especially in high frequencies.

Acknowledgement

This work was financially supported in part 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, Chevron, ConocoPhillips, ION, Petrobras, PGS, Statoil, Total SA, WesternGeco, Woodside.