Using prior support information in approximate message passing algorithm

Navid Ghadermarzy

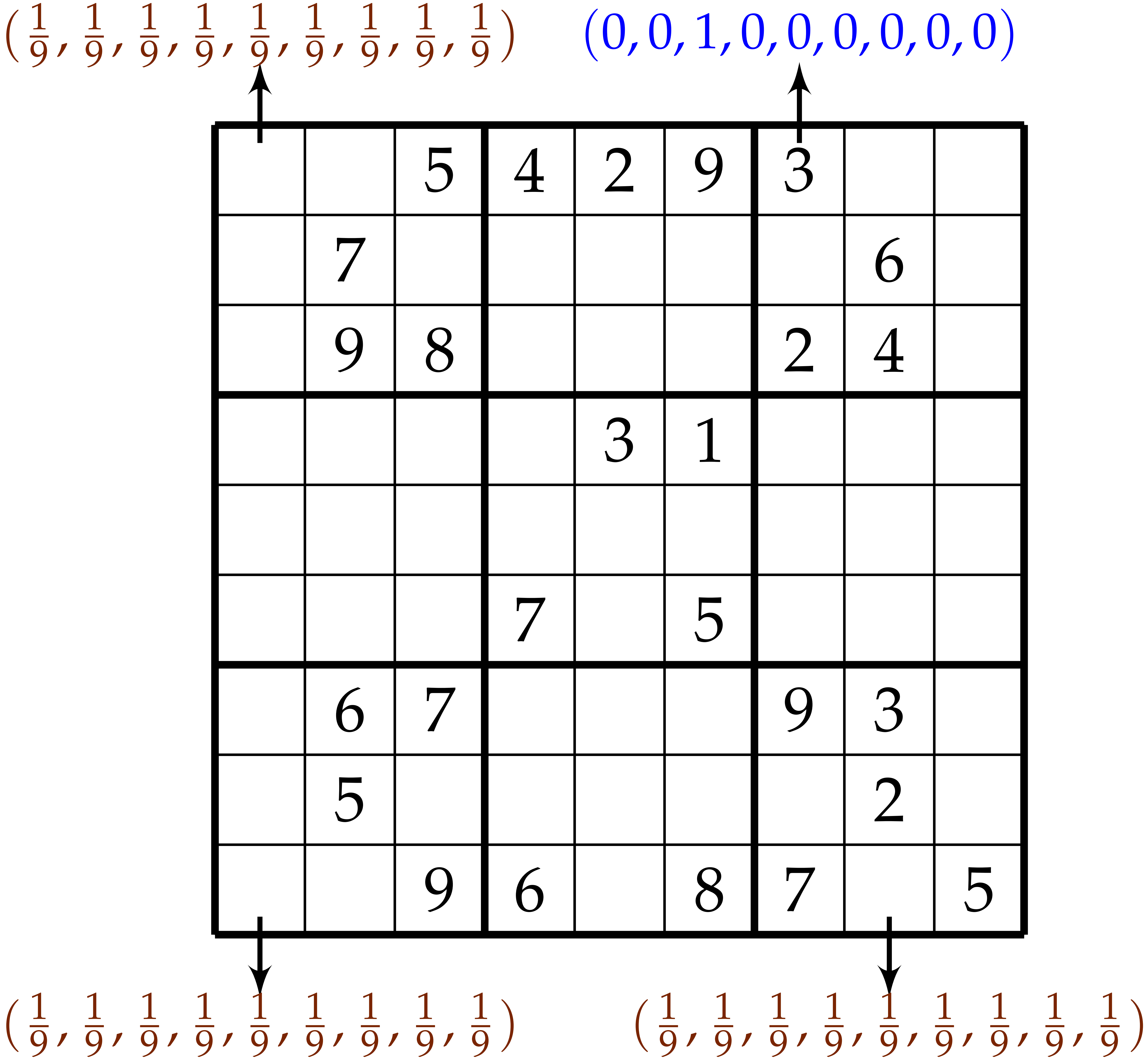
University of British Columbia

December 2, 2013

Main messages

- We incorporate prior support information into the approximate message passing (AMP) algorithm which is a fast iterative algorithm for sparse recovery.
- Using weighted AMP as a pre-processor we can improve the results of seismic trace interpolation via ℓ_1 recovery in terms of both accuracy and convergence time.

A simple example



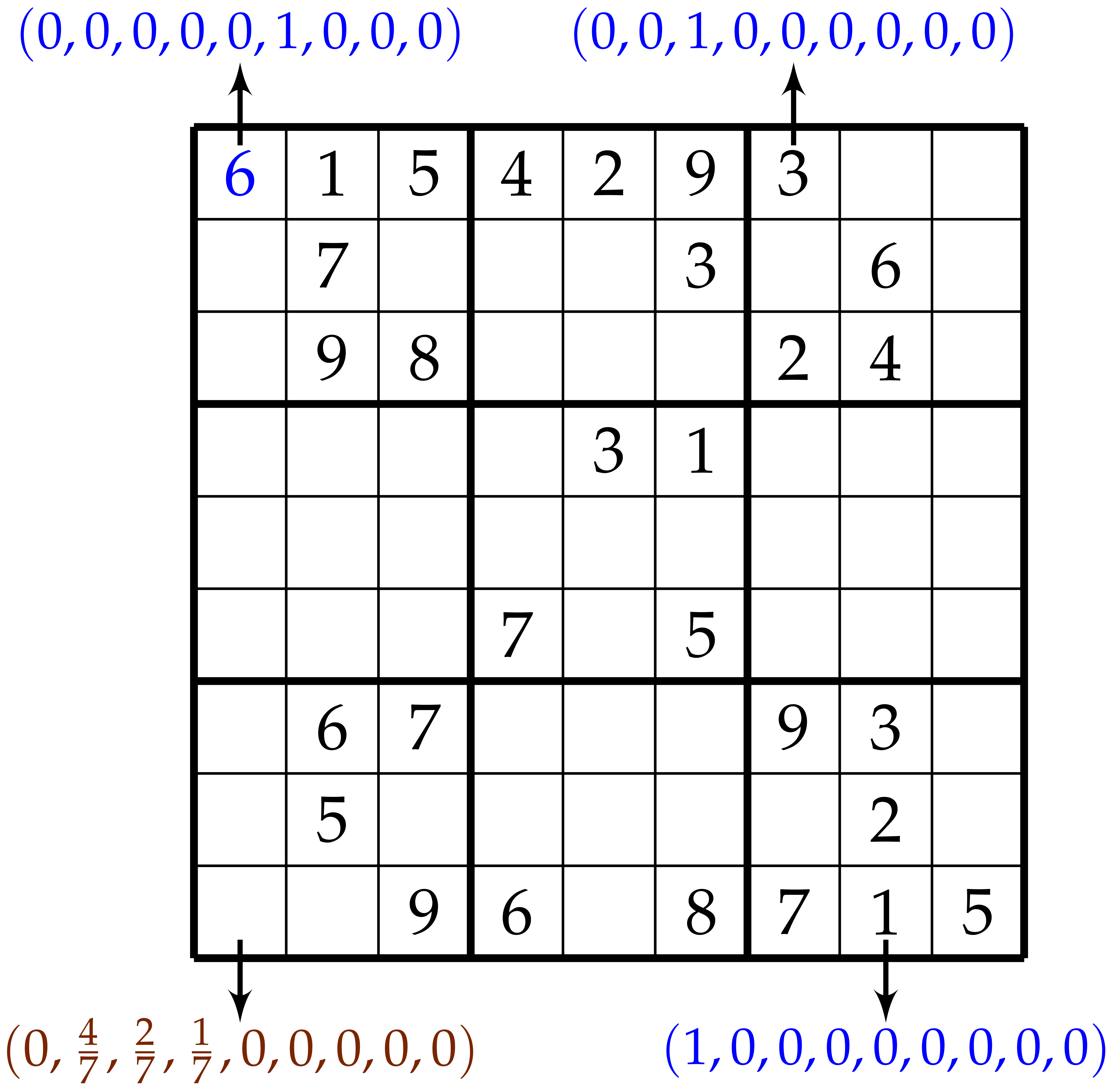
First iteration

$(\frac{1}{2}, 0, 0, 0, 0, \frac{1}{2}, 0, 0, 0)$ $(0, 0, 1, 0, 0, 0, 0, 0, 0)$

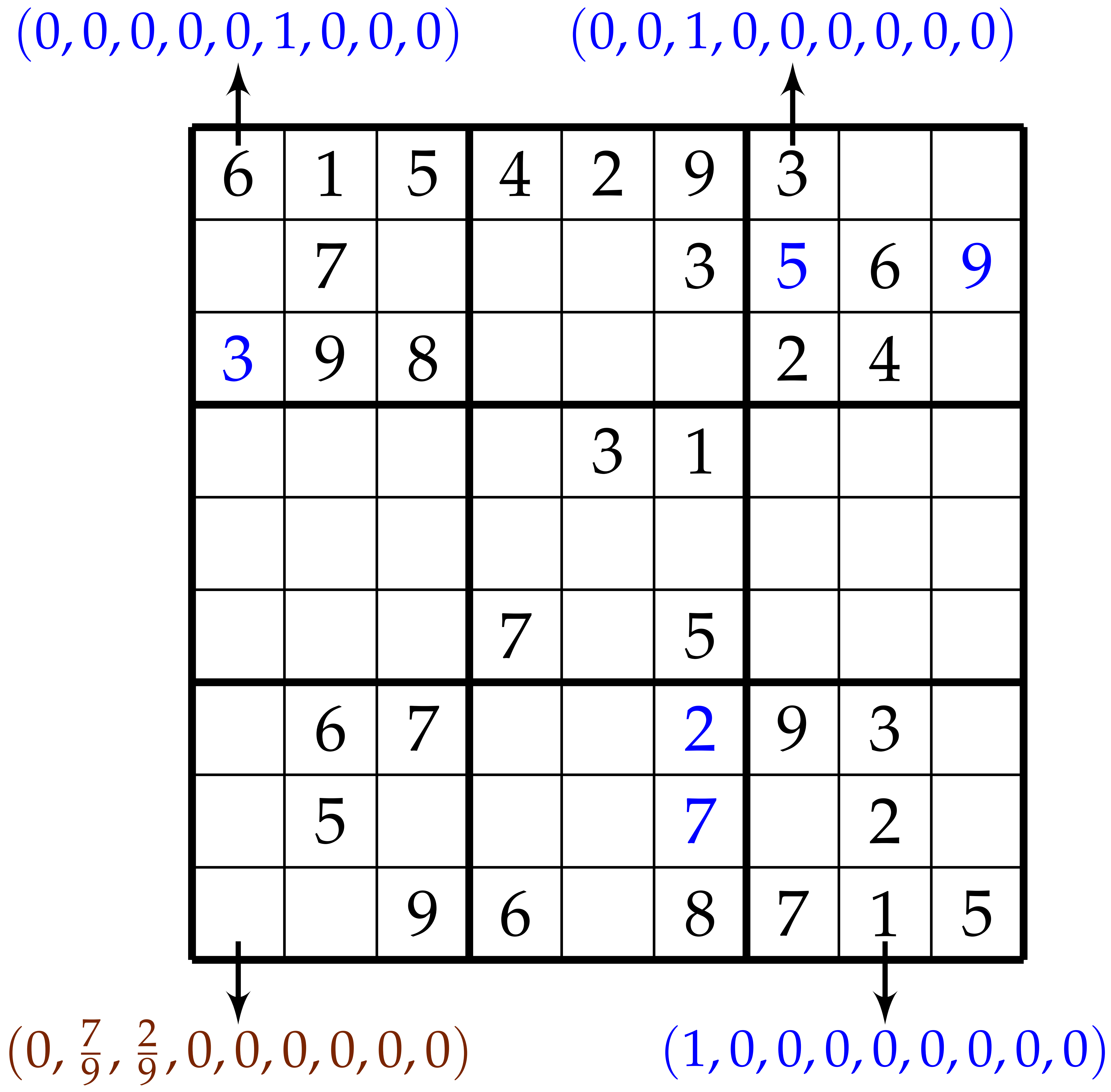
	1	5	4	2	9	3		
	7				3		6	
	9	8				2	4	
				3	1			
			7		5			
	6	7				9	3	
	5						2	
		9	6		8	7	1	5

$(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, 0, 0, 0, 0, 0)$ $(1, 0, 0, 0, 0, 0, 0, 0, 0)$

Second iteration

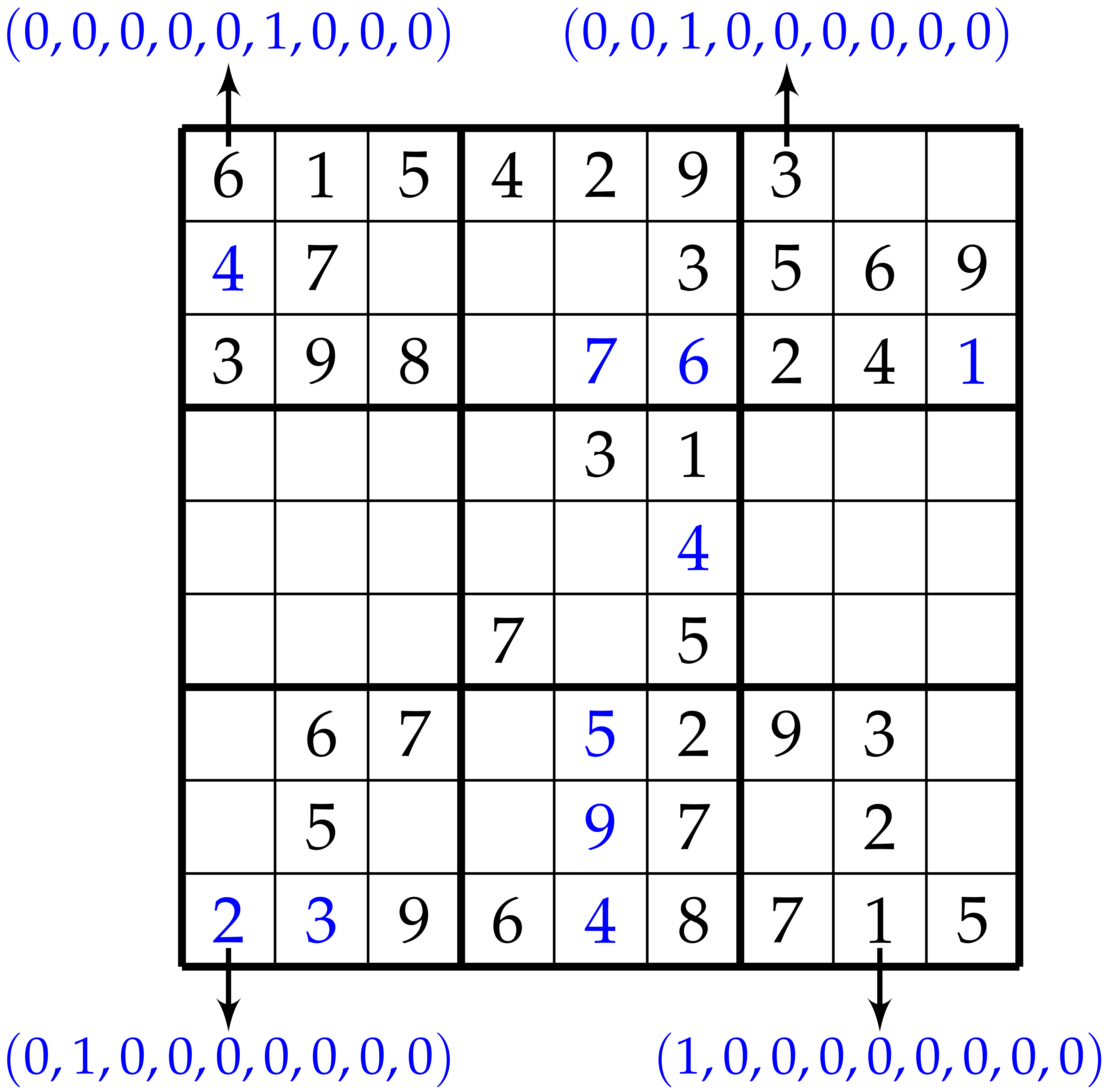


Third iteration



6	1	5	4	2	9	3		
	7				3	5	6	9
3	9	8				2	4	
				3	1			
			7		5			
	6	7			2	9	3	
	5				7		2	
		9	6		8	7	1	5

Fourth iteration



8th iteration

$(0,0,0,0,0,1,0,0,0)$ $(0,0,1,0,0,0,0,0,0)$

6	1	5	4	2	9	3	7	
4	7	2	8	1	3	5	6	9
3	9	8	5	7	6	2	4	1
		6		3	1	4		
					4			
9			7	6	5		8	
8	6	7	1	5	2	9	3	4
1	5	4	3	9	7		2	
2	3	9	6	4	8	7	1	5

$(0,1,0,0,0,0,0,0,0)$ $(1,0,0,0,0,0,0,0,0)$

12th iteration

$(0,0,0,0,0,1,0,0,0)$ $(0,0,1,0,0,0,0,0,0)$

6	1	5	4	2	9	3	7	8
4	7	2	8	1	3	5	6	9
3	9	8	5	7	6	2	4	1
5	8	6	2	3	1	4	9	7
7	2	1	9	8	4	6	5	3
9	4	3	7	6	5	1	8	2
8	6	7	1	5	2	9	3	4
1	5	4	3	9	7	8	2	6
2	3	9	6	4	8	7	1	5

$(0,1,0,0,0,0,0,0,0)$ $(1,0,0,0,0,0,0,0,0)$

Example: Revisiting ℓ_1 for randomized acquisition of seismic lines

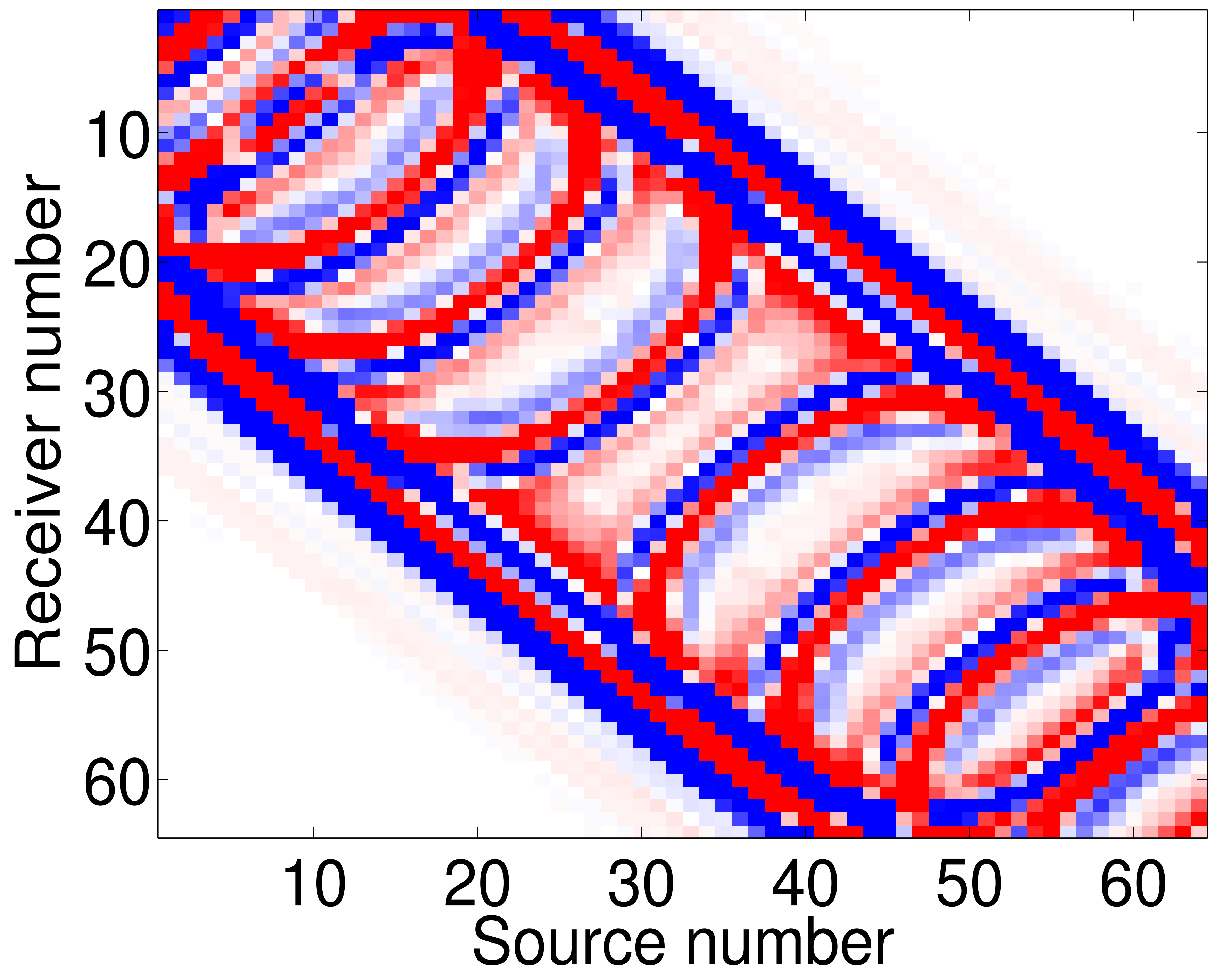
Consider a seismic line with 64 sources, 64 receivers, and 256 time samples.

The receiver spread is randomly subsampled using the mask R.

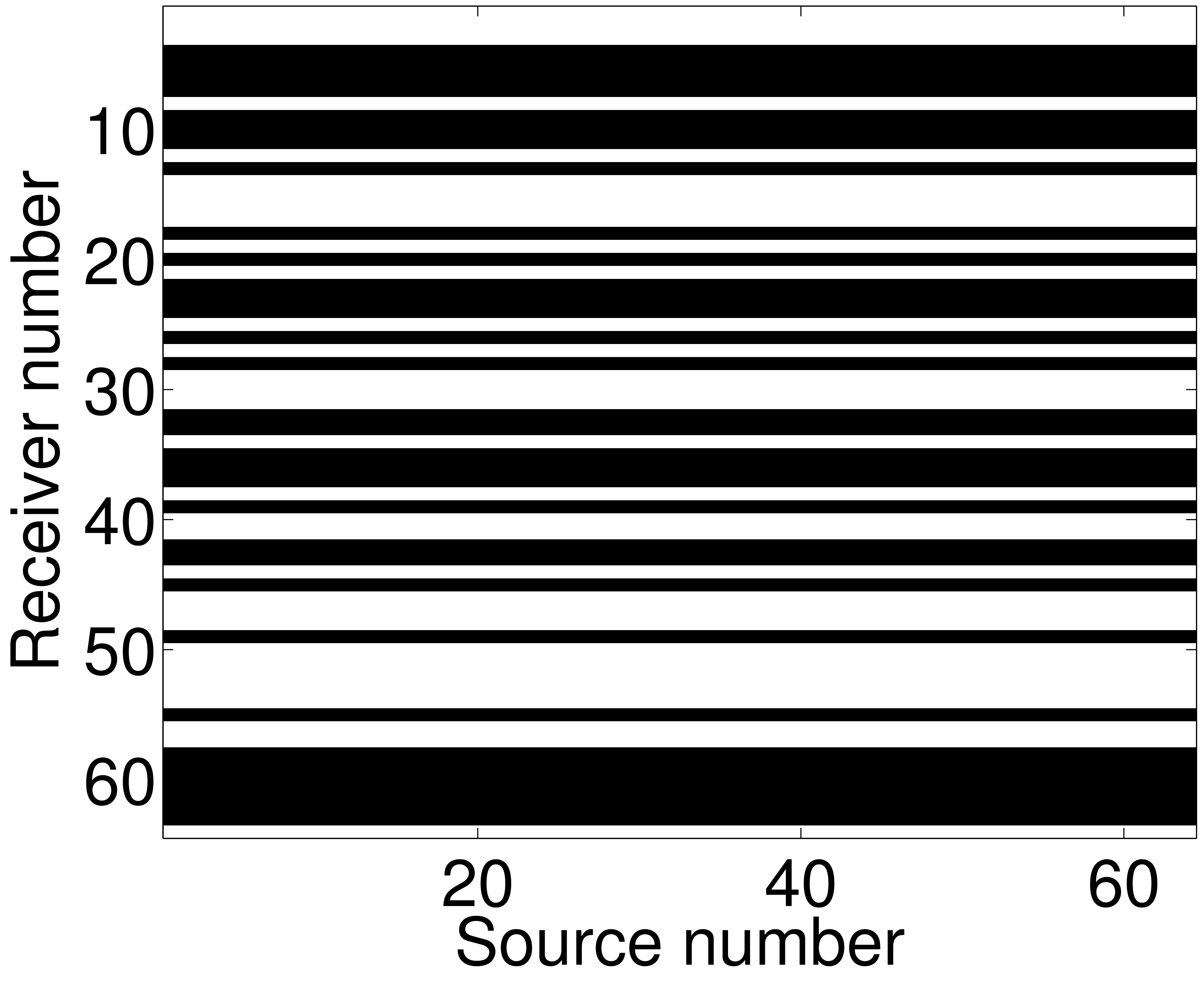
Example: Revisiting ℓ_1 for randomized acquisition of seismic lines

Consider a seismic line with 64 sources, 64 receivers, and 256 time samples.

The receiver spread is randomly subsampled using the mask R.



(a) Fully sampled time slice f

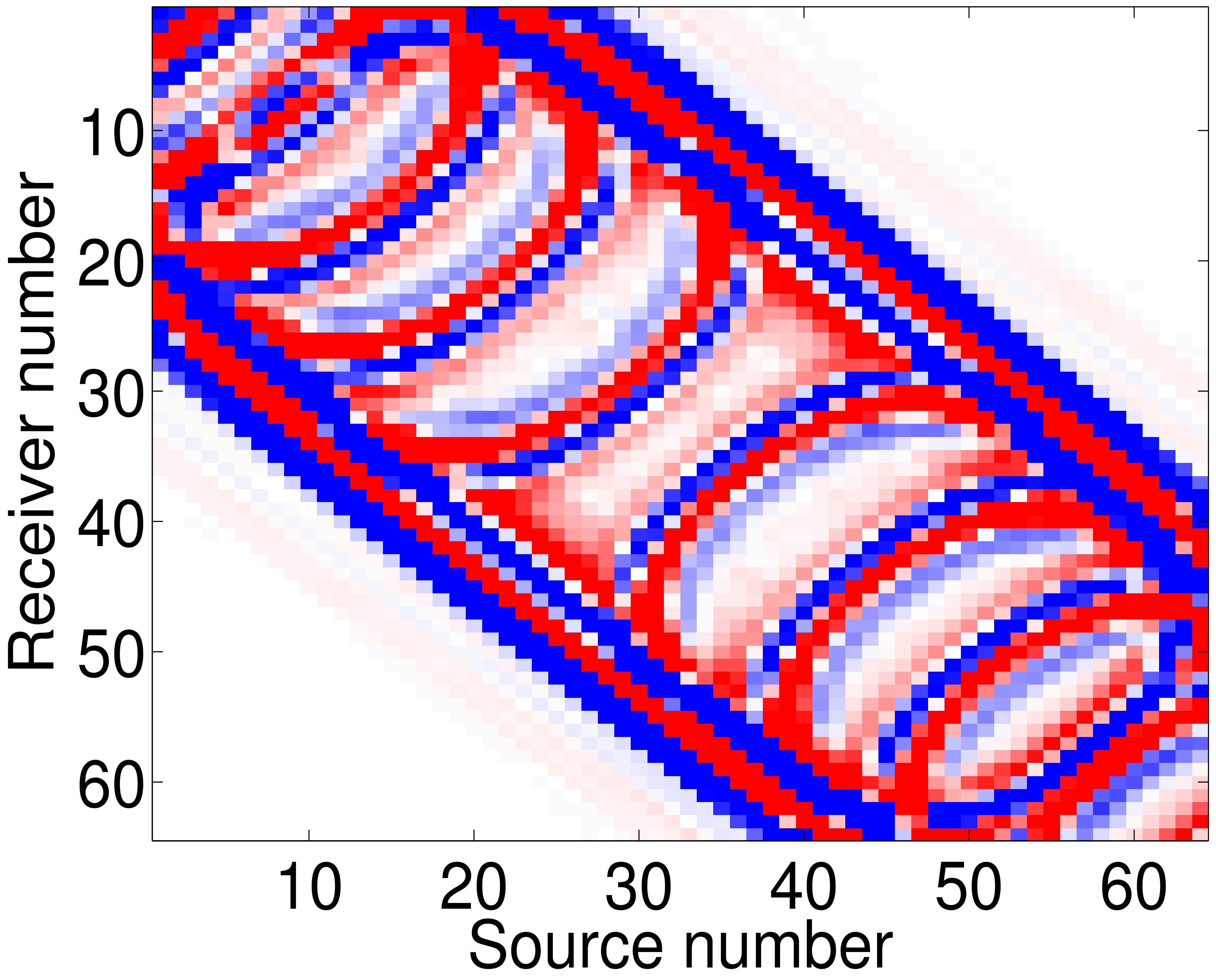


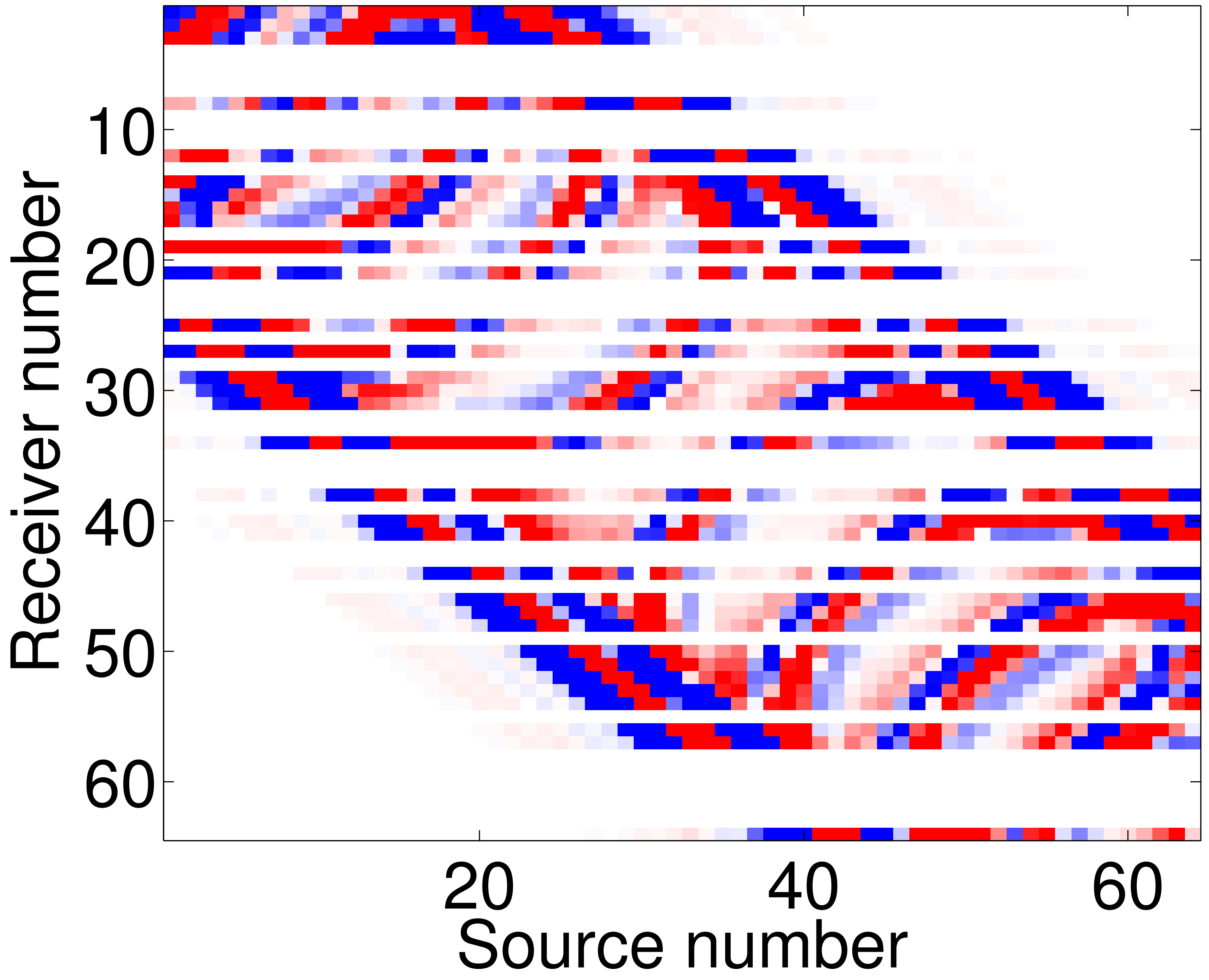
(b) Mask R

Example: Revisiting ℓ_1 for randomized acquisition of seismic lines

Consider a seismic line with 64 sources, 64 receivers, and 256 time samples.

The receiver spread is randomly subsampled using the mask R.





Using ℓ_1 for seismic trace interpolation

We want to recover a high dimensional seismic data volume f by interpolating between a smaller number of measurements $b = RMf$.

Let S be a sparsifying operator that characterizes the transform domain of f, such that $S \in \mathbb{C}^{P \times N}$ with $P > N$. In the case of the redundant curvelet transform $S^H S = \mathbb{I}$.

Using ℓ_1 for seismic trace interpolation

We want to recover a high dimensional seismic data volume f by interpolating between a smaller number of measurements $b = RMf$.

Let S be a sparsifying operator that characterizes the transform domain of f, such that $S \in \mathbb{C}^{P \times N}$ with $P > N$. In the case of 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

We want to recover a high dimensional seismic data volume f by interpolating between a smaller number of measurements $b = RMf$.

Let S be a sparsifying operator that characterizes the transform domain of f, such that $S \in \mathbb{C}^{P \times N}$ with $P > N$. In the case of the redundant curvelet transform $S^H S = \mathbb{I}$.

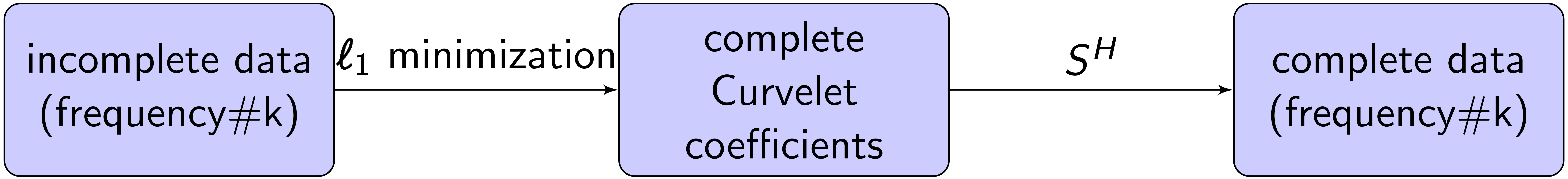
Then $b = RMS^H x$, where x can be recovered by sparse recovery algorithms like ℓ_1 minimization.

Using ℓ_1 for seismic trace interpolation

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

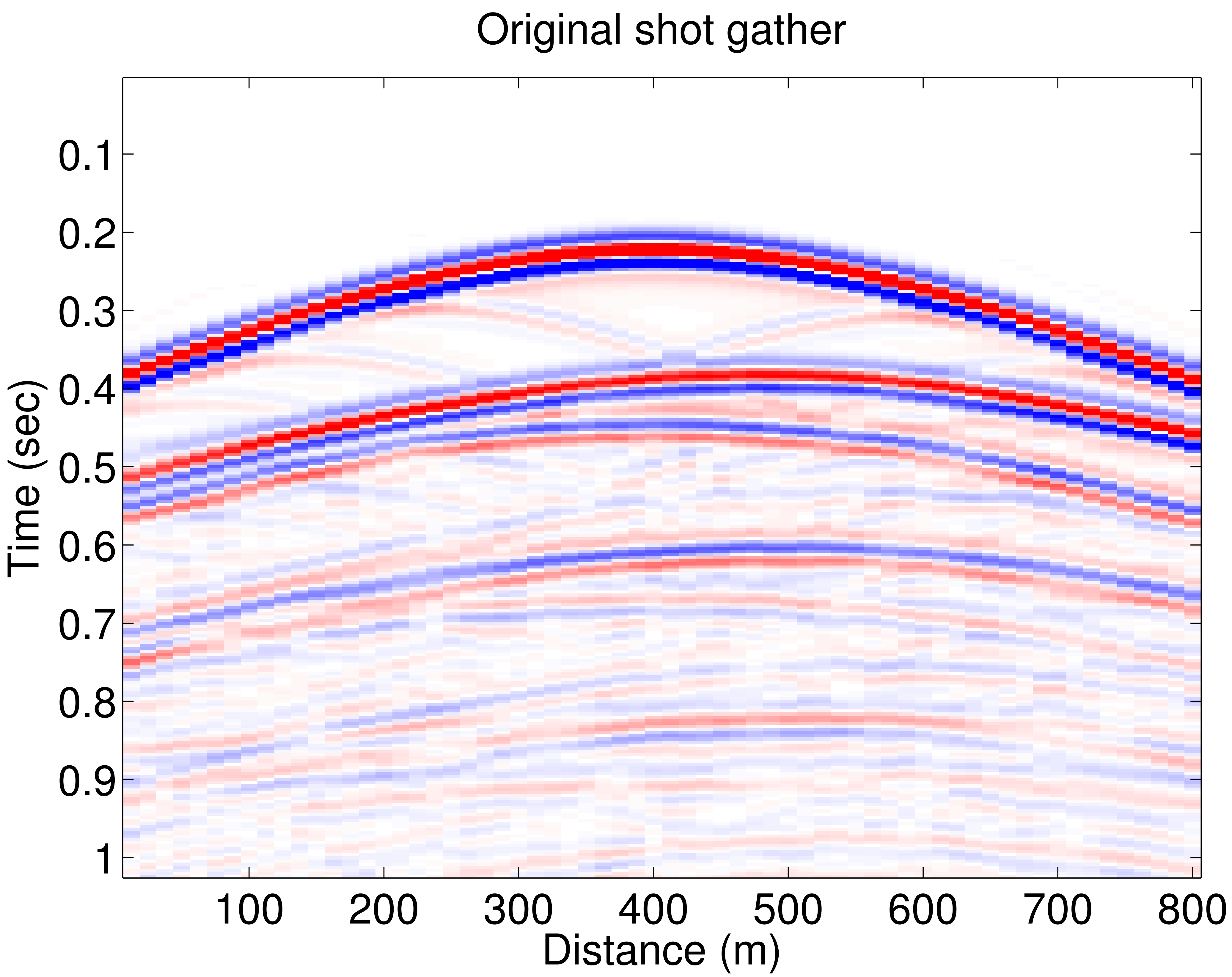
$$x^{\ell_1} := \underset{z \in R^P}{\text{minimize}} ||z||_1 \quad \text{subject to} \quad ||RMS^H z - b||_2 \leq \epsilon,$$

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

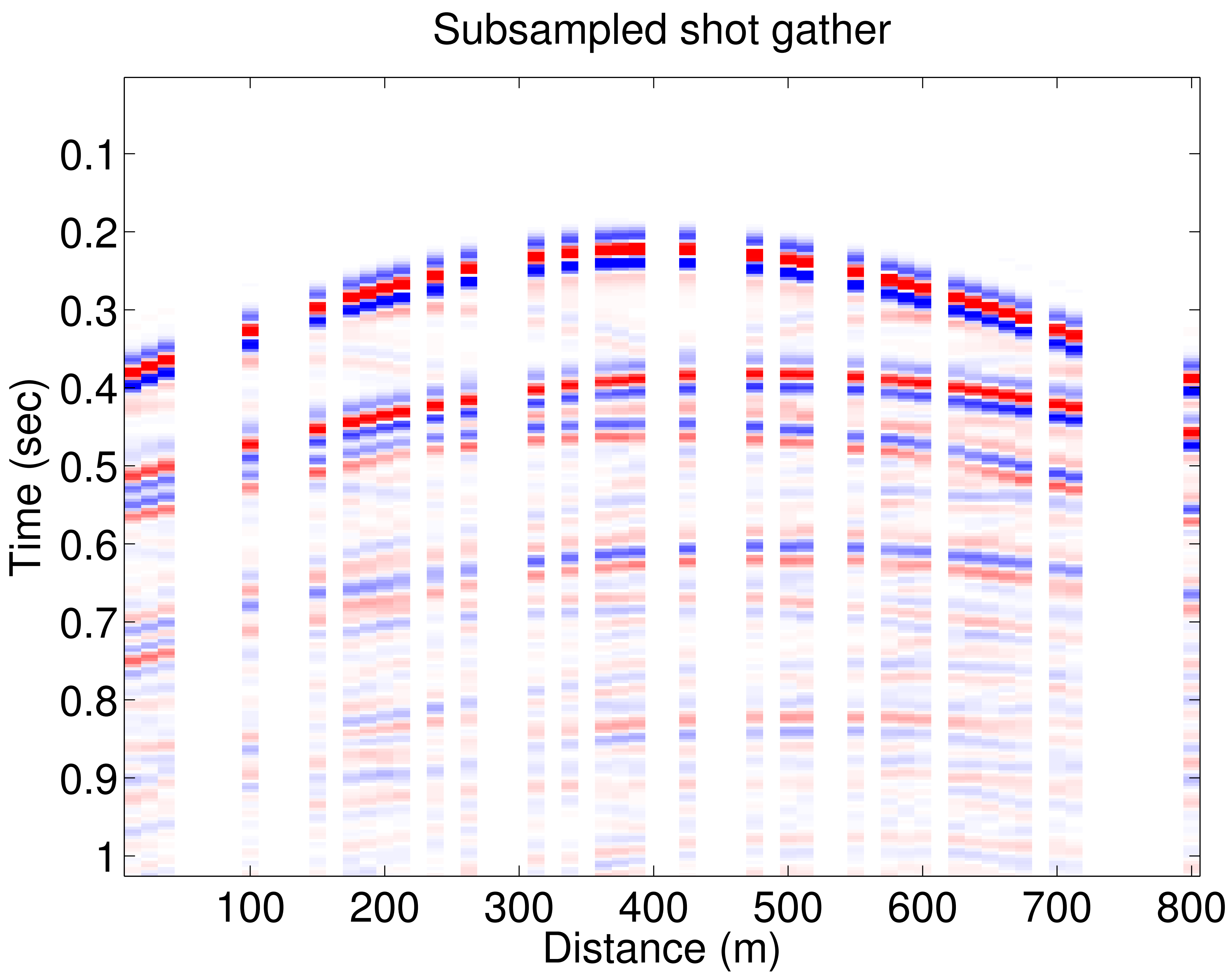


Recovery using ℓ_1 minimization on frequency slices (shotgather # 32)

Shotgather number 32 from the seismic line:

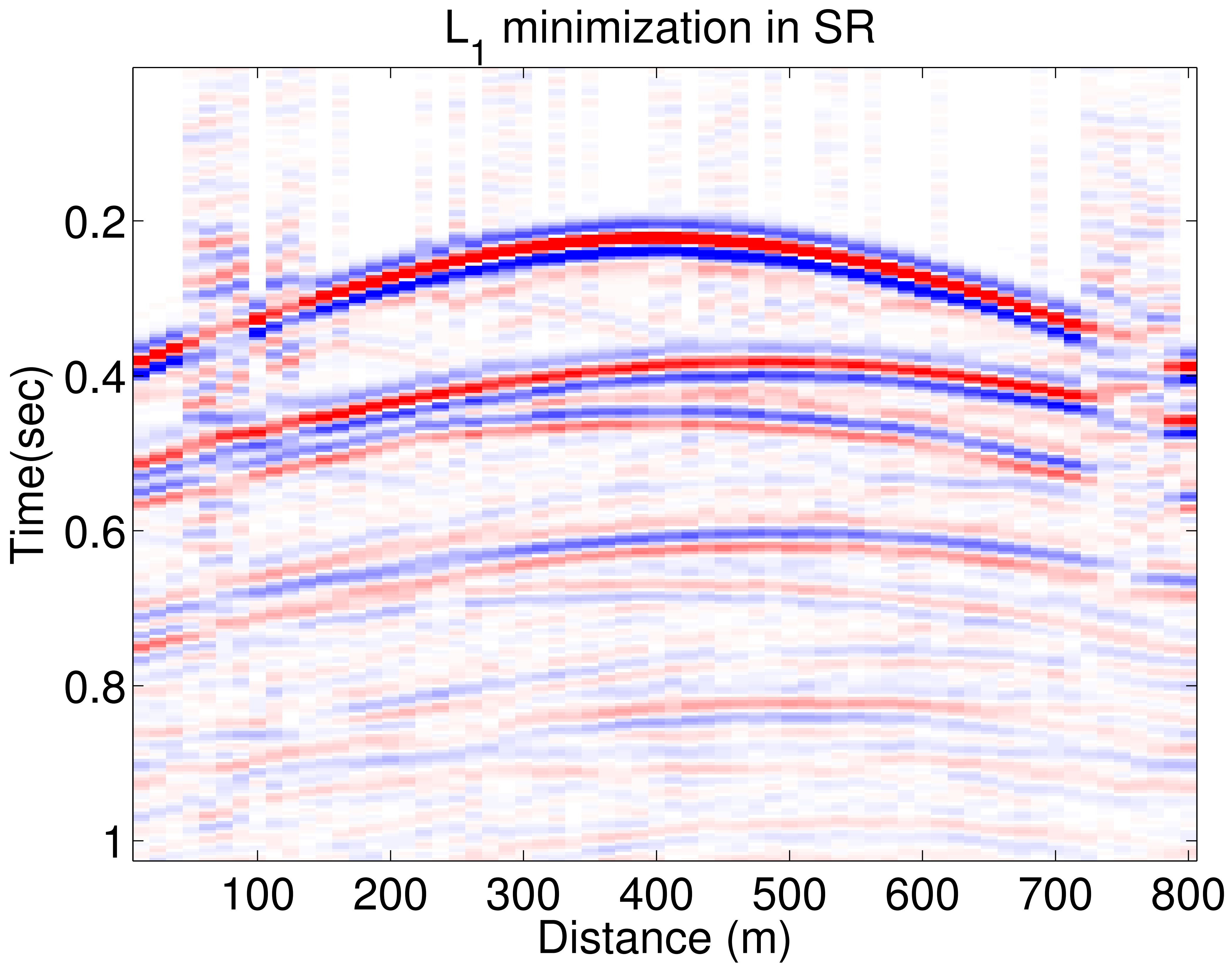


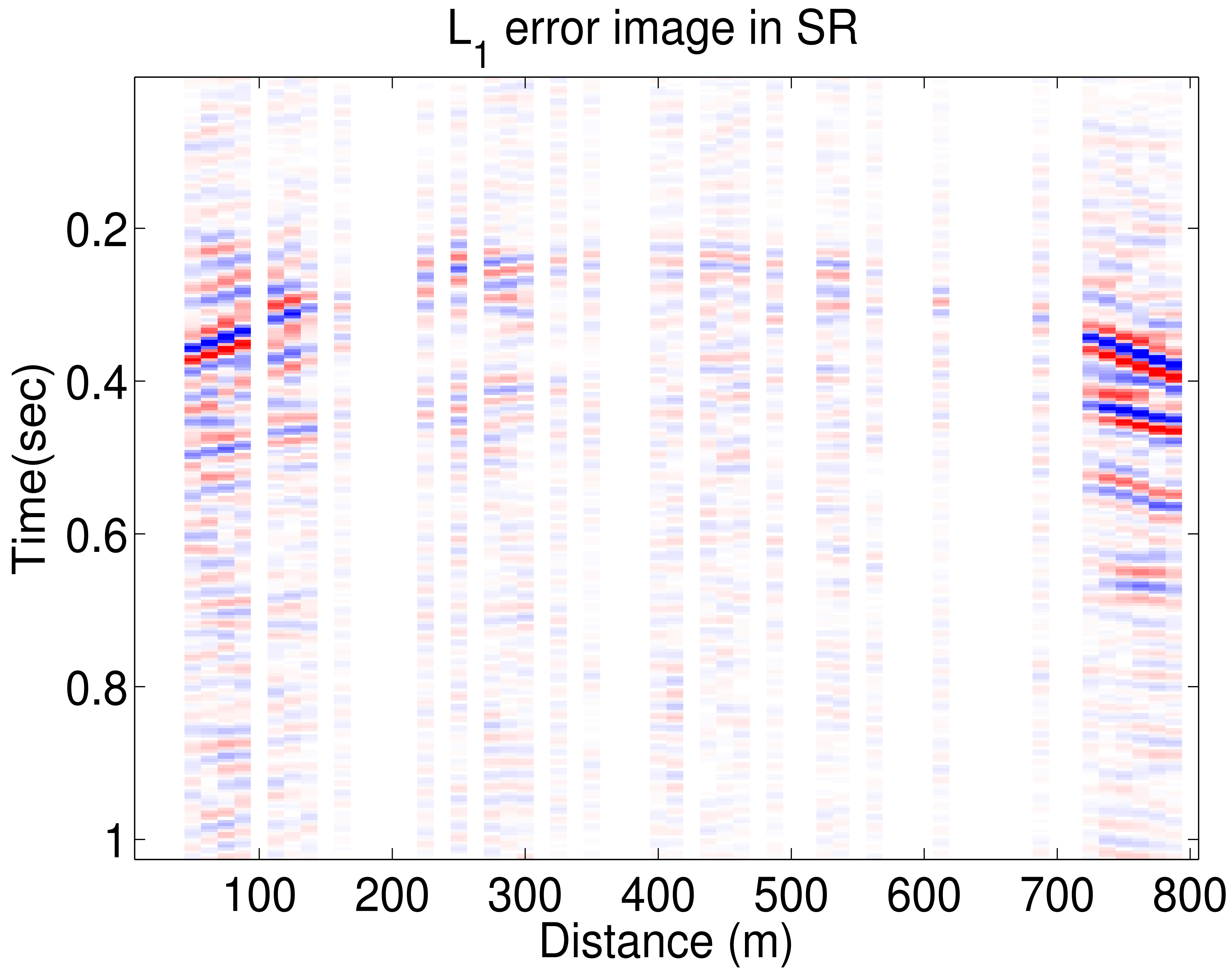
(e)



(f)

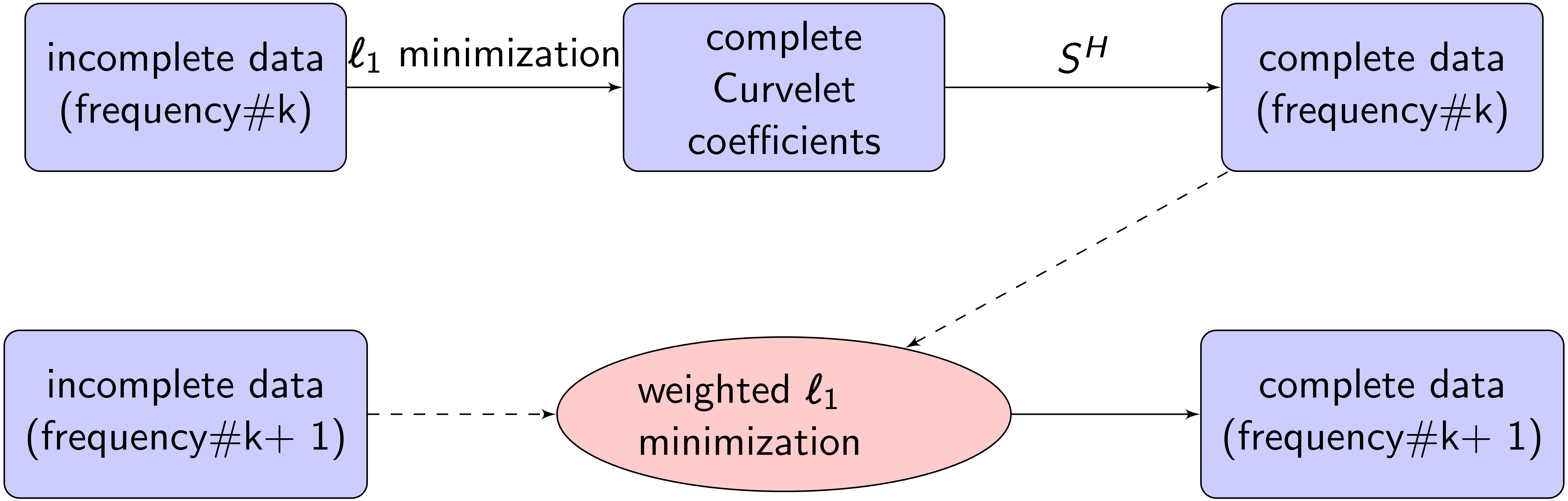
Recovery results: ℓ_1 minimization





Using weighted ℓ_1 for seismic trace interpolation

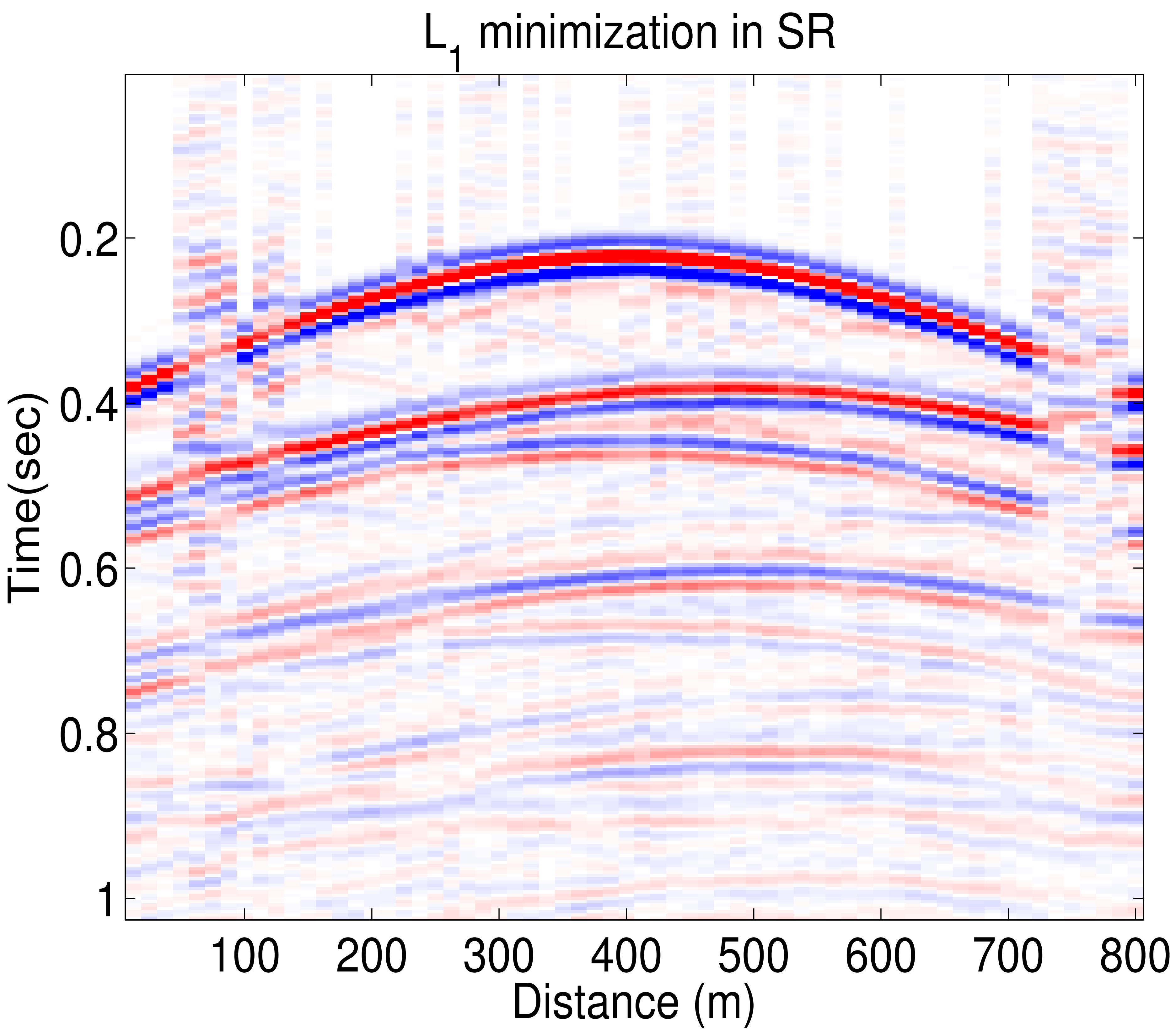
Adjacent frequency slices and have highly correlated curvelet domain support sets. Hence we can use the support set of each slice as an estimate for the support of the next slice.

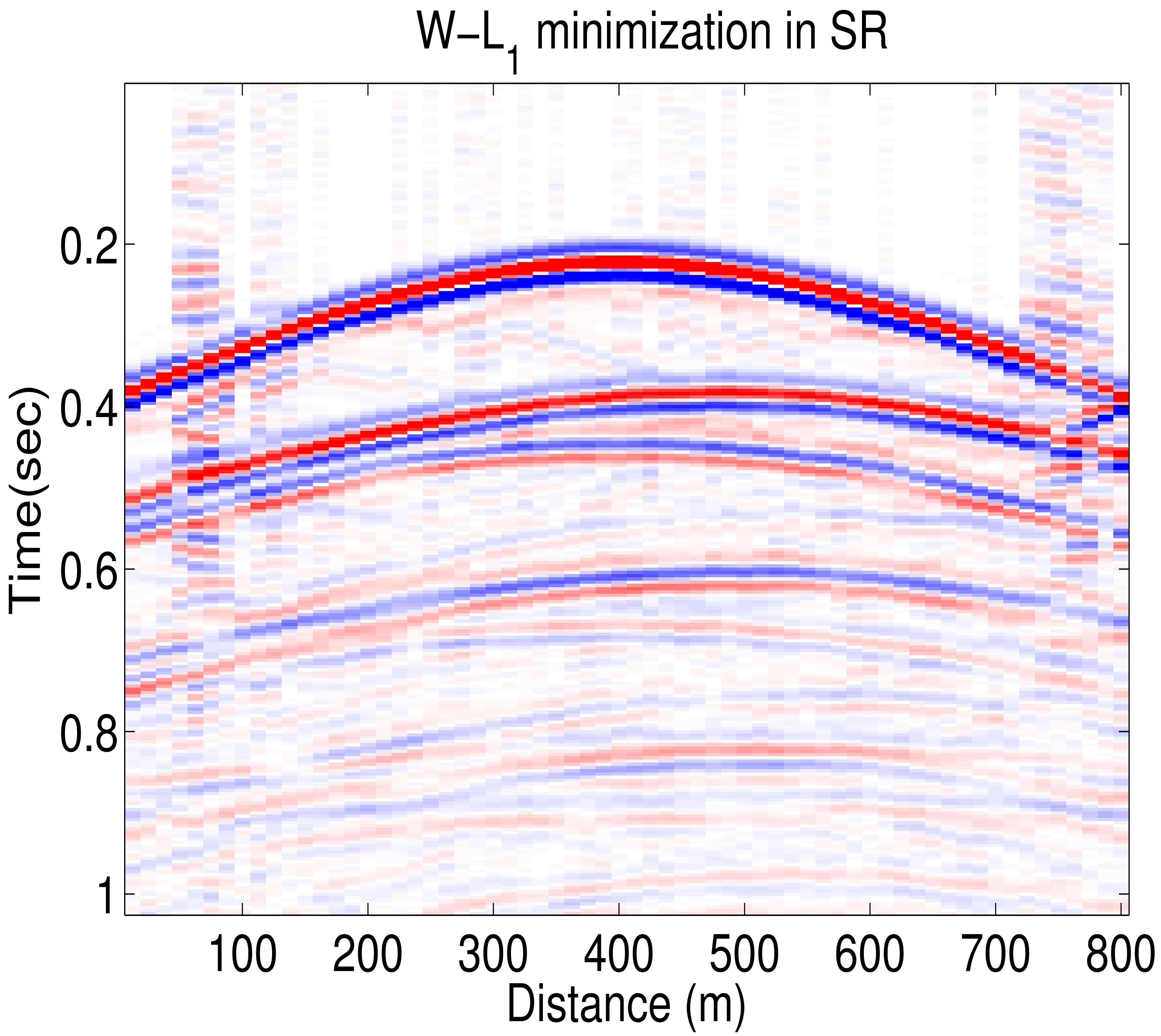


Weighted ℓ_1 for seismic trace interpolation

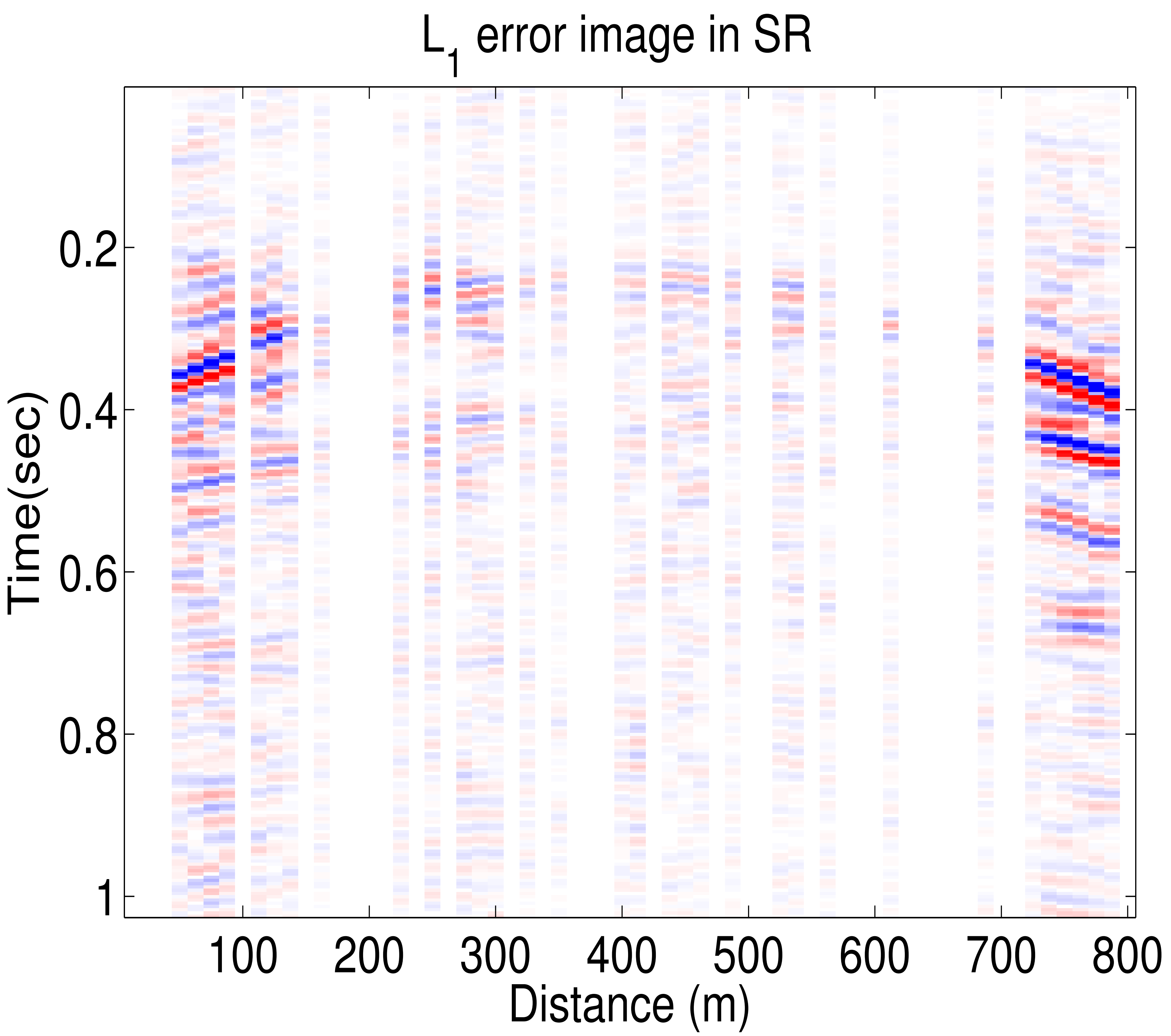
$$x^{w\ell_1} := \underset{z \in R^P}{\text{minimize}} ||z||_{1,\mathtt{w}} \quad \text{subject to} \quad ||RMS^H z - b||_2 \leq \epsilon,$$

$\mathtt{w} \in \{\omega, 1\}^N$ is the weight vector and $||z||_{1,\mathtt{w}} := \Sigma_i \mathtt{w}_i |z_i|$ is the weighted ℓ_1 norm.

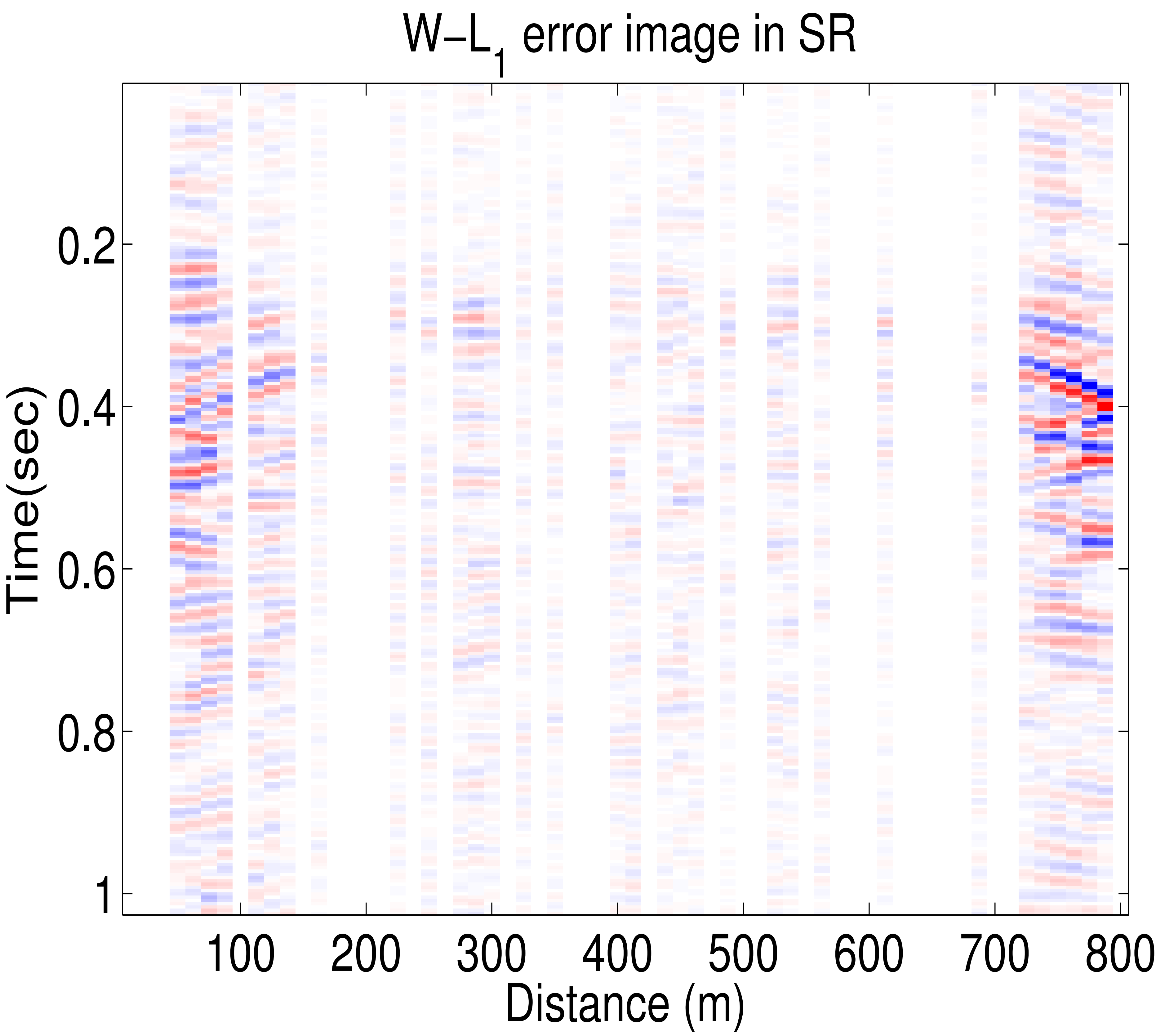




Recovery error: ℓ_1 vs weighted ℓ_1



(k) SNR=8.3 dB



(l) SNR= 10.7 dB

Iterative thresholding algorithms

- The ℓ_1 minimization problem is perhaps the most common approach to recover x and can be solved by linear programming algorithms.

- ℓ_1 has high computational complexity.

Iterative thresholding algorithms

- The ℓ_1 minimization problem is perhaps the most common approach to recover x and can be solved by linear programming algorithms.

- ℓ_1 has high computational complexity.

- **Iterative thresholding** algorithms is an appealing alternative

$$\begin{aligned} x^{t+1} &= \eta(x^t + A^* z^t, \hat{\tau}^t), \\ z^t &= y - Ax^t. \end{aligned} \tag{1}$$

Iterative thresholding algorithms

- The ℓ_1 minimization problem is perhaps the most common approach to recover x and can be solved by linear programming algorithms.
- ℓ_1 has high computational complexity.
- **Iterative thresholding** algorithms is an appealing alternative

$$\begin{aligned} x^{t+1} &= \eta(x^t + A^* z^t, \hat{\tau}^t), \\ z^t &= y - A x^t. \end{aligned} \tag{1}$$

Approximate message passing (AMP)

- Donoho, Maleki and Montanari proposed a new iterative thresholding algorithm which is referred to as approximate message passing (AMP).

- This algorithm enjoys both the low complexity of IT algorithms and the superior recovery conditions of BP.

Approximate message passing (AMP)

- Donoho, Maleki and Montanari proposed a new iterative thresholding algorithm which is referred to as approximate message passing (AMP).

- This algorithm enjoys both the low complexity of IT algorithms and the superior recovery conditions of BP.

- The AMP algorithm starts from an initial x^0 and an initial threshold $\hat{\tau}^0 = 1$ and iteratively goes by

$$\begin{aligned} x^{t+1} &= \eta(x^t + A^* z^t; \tau^t), \\ z^t &= y - Ax^t + \delta^{-1} z^{t-1} \langle \eta'(x^{t-1} + A^* z^{t-1}; \tau^{t-1}) \rangle. \end{aligned} \tag{2}$$

Approximate message passing (AMP)

- Donoho, Maleki and Montanari proposed a new iterative thresholding algorithm which is referred to as approximate message passing (AMP).

- This algorithm enjoys both the low complexity of IT algorithms and the superior recovery conditions of BP.

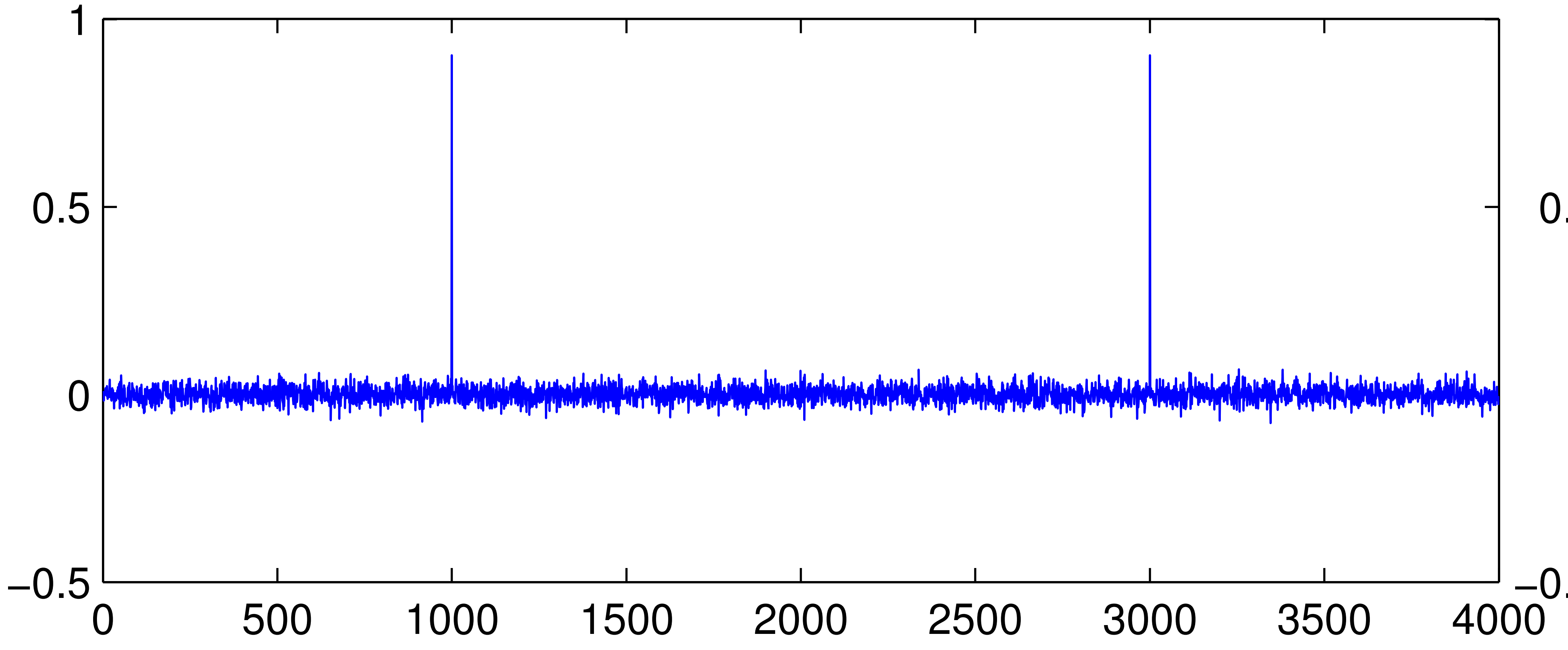
- The AMP algorithm starts from an initial x^0 and an initial threshold $\hat{\tau}^0 = 1$ and iteratively goes by

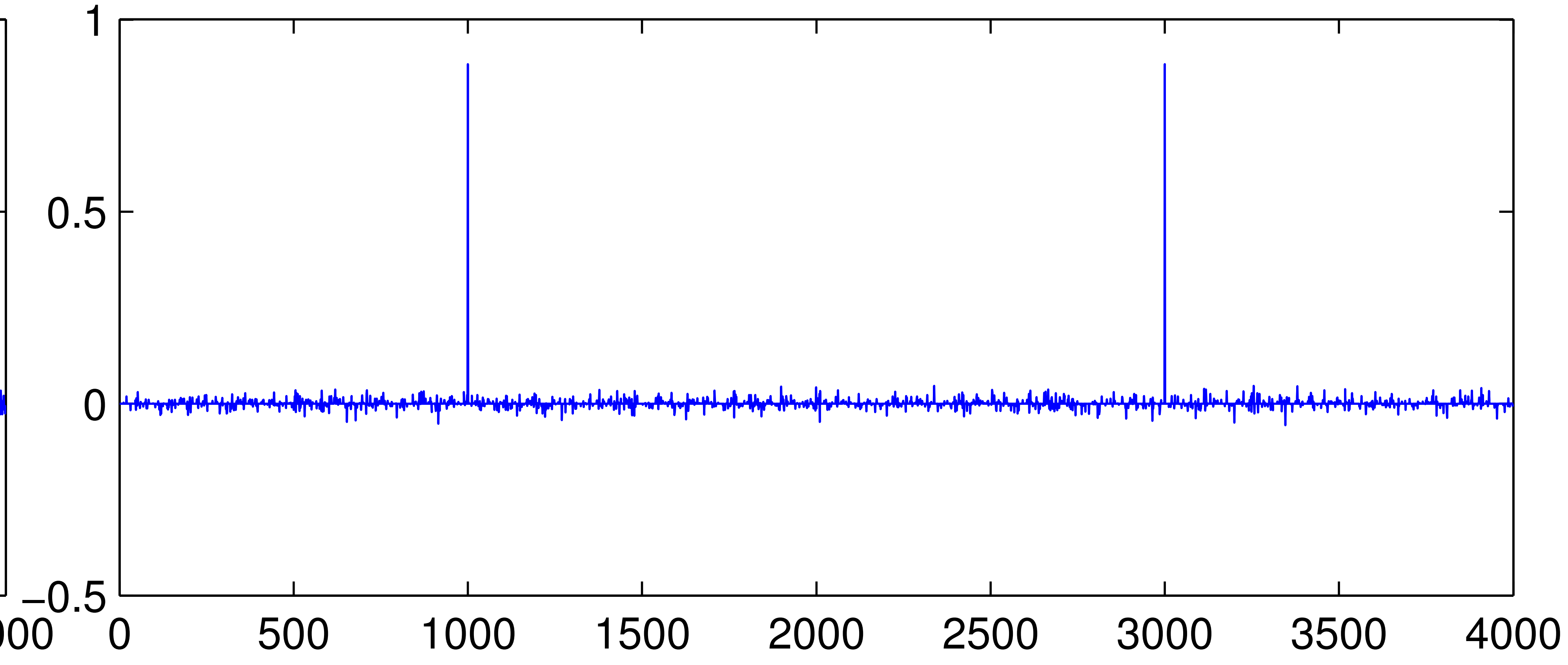
$$\begin{aligned} x^{t+1} &= \eta(x^t + A^* z^t; \tau^t), \\ z^t &= y - Ax^t + \delta^{-1} z^{t-1} \langle \eta'(x^{t-1} + A^* z^{t-1}; \tau^{t-1}) \rangle. \end{aligned} \tag{2}$$

Simple example with a 2-sparse signali Iteration $t = 1$

Soft thresholding:

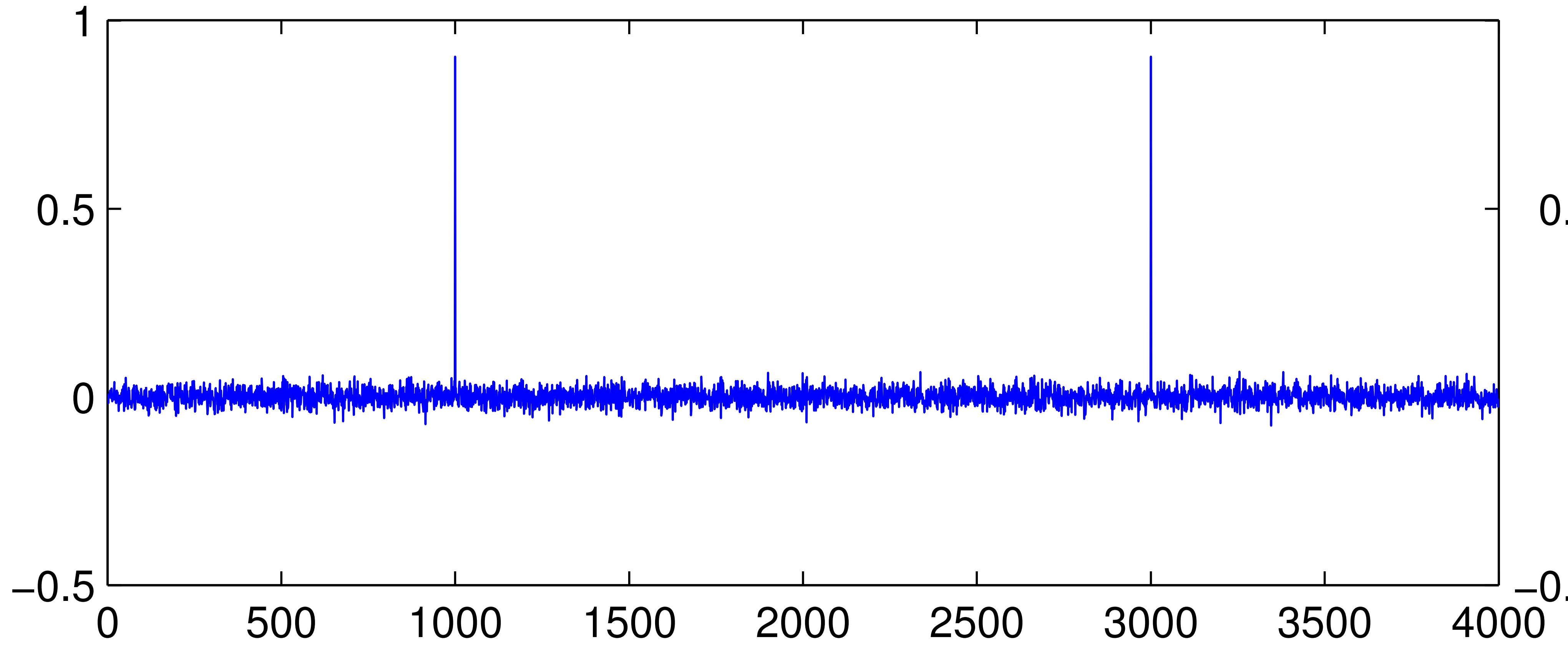
$x^t + A^* z^t$ $\qquad$ $\eta(x^t + A^* z^t; \tau^t)$

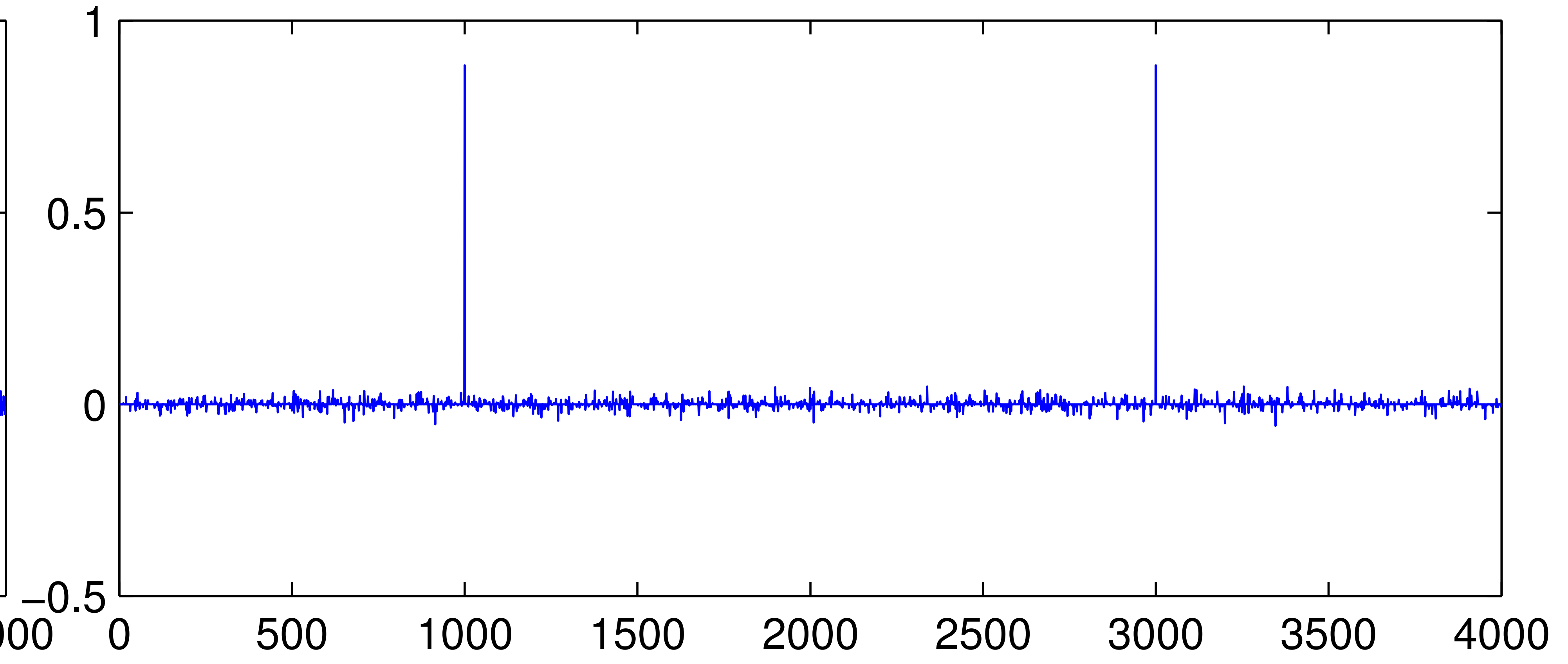




AMP:

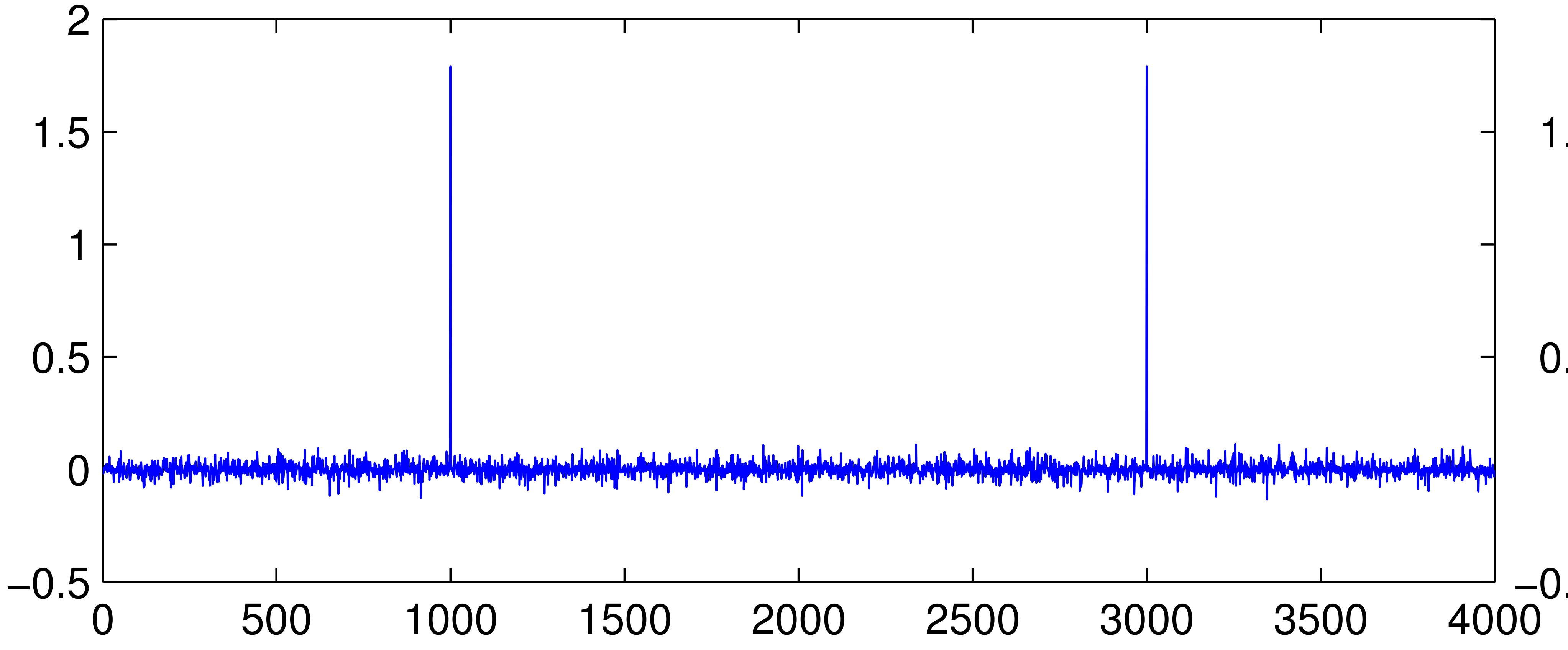
$x^t + A^*(z^t + \alpha^t z^{t-1})$ $\qquad$ $\eta(x^t + A^*(z^t + \alpha^t z^{t-1}); \tau^t)$

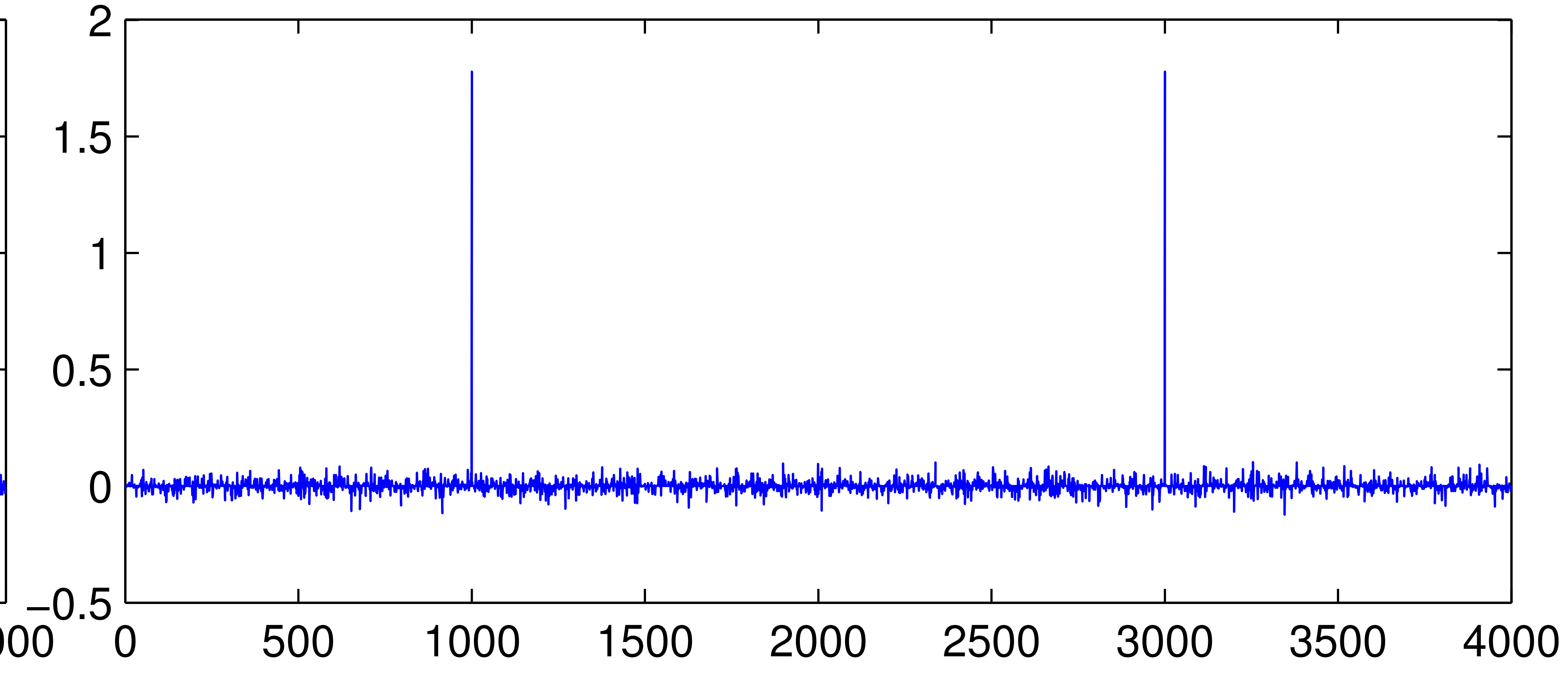




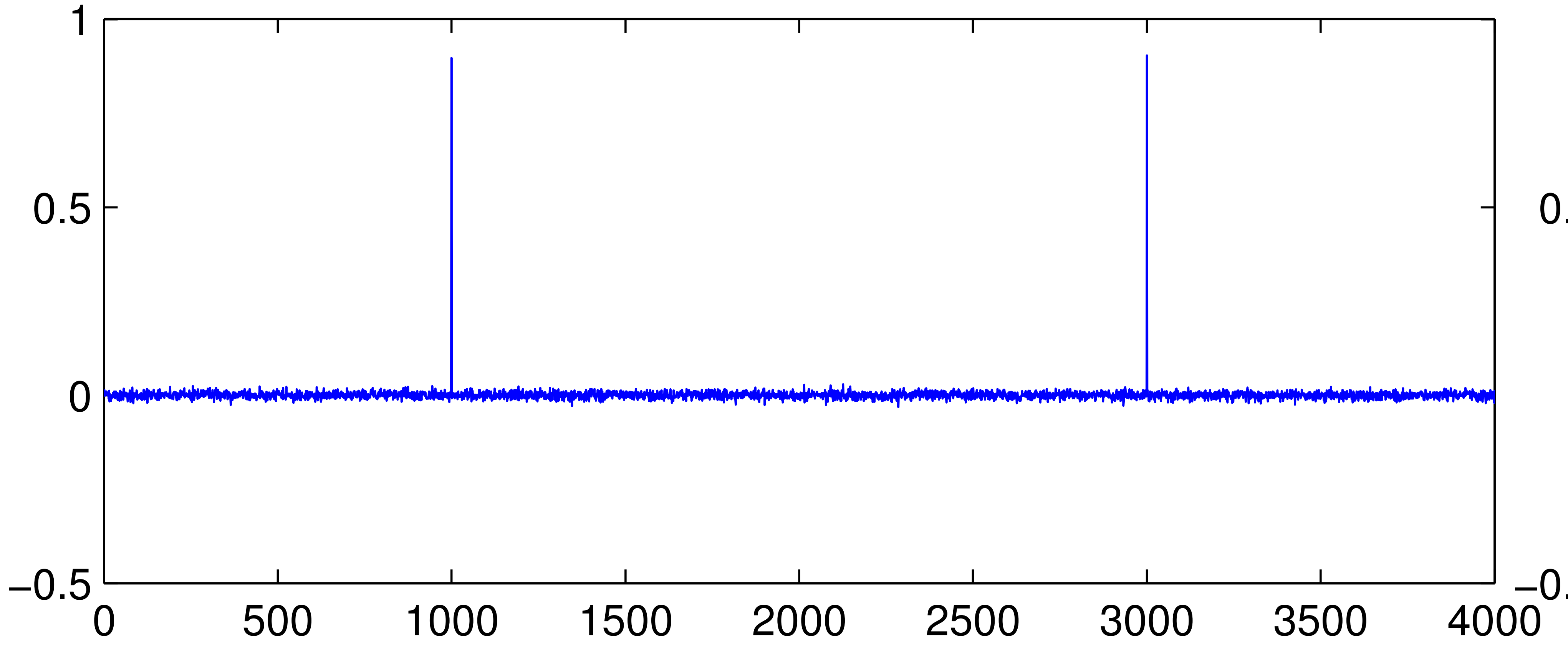
Iteration $t = 2$

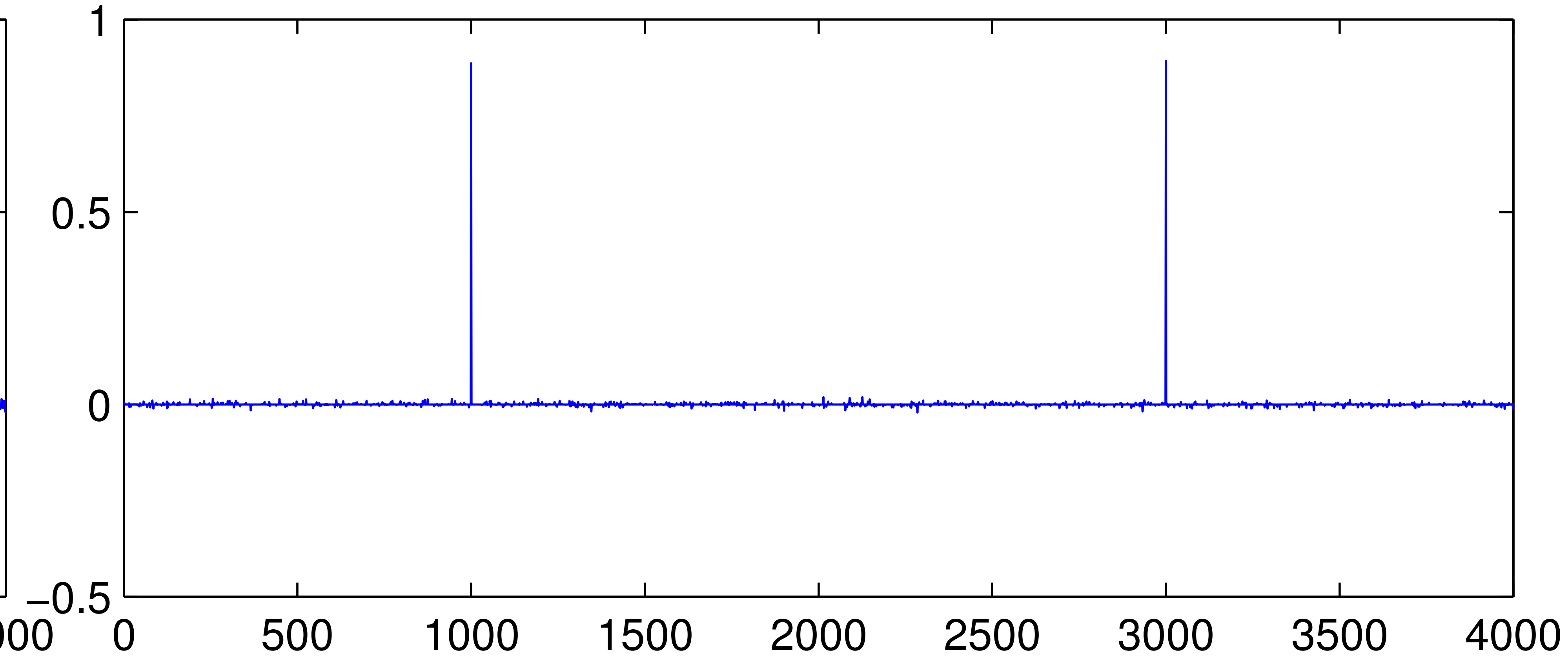
Soft thresholding: $x^t + A^* z^t$ $\qquad$ $\eta(x^t + A^* z^t; \tau^t)$





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

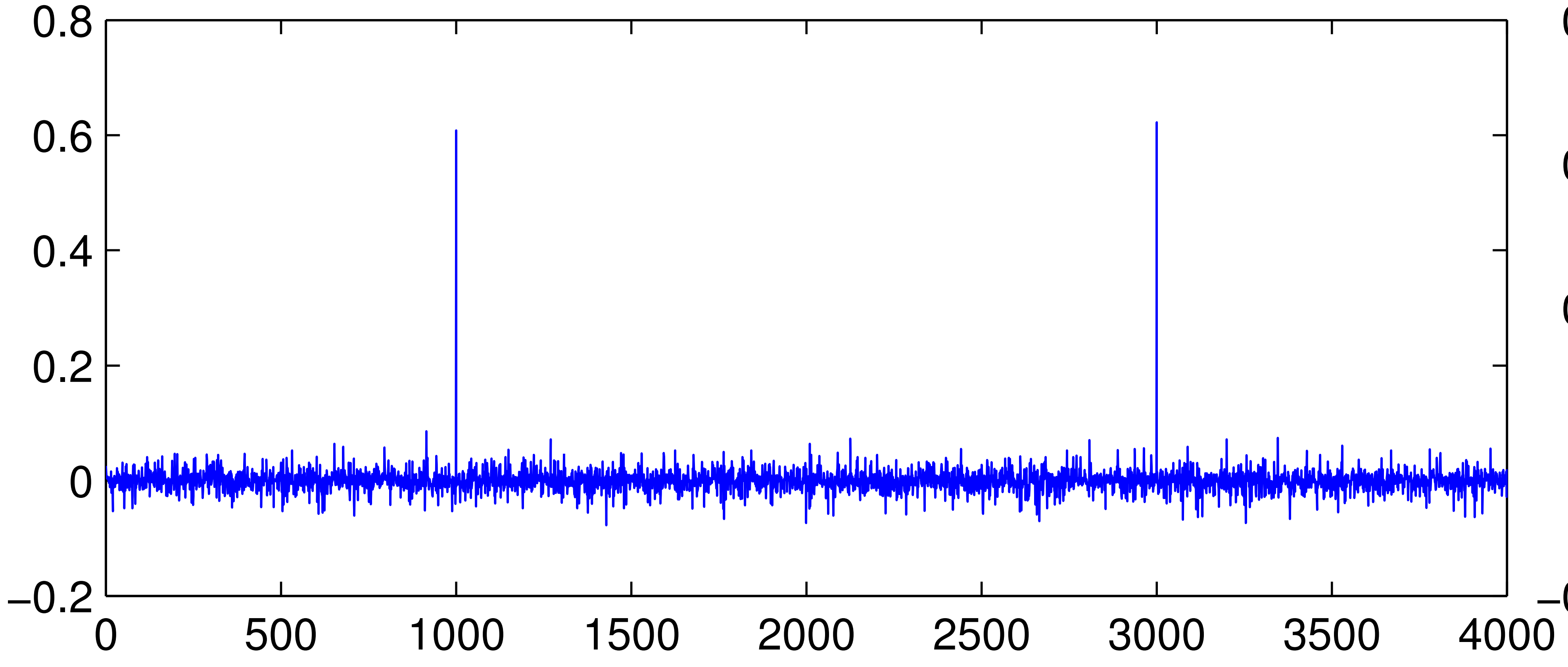




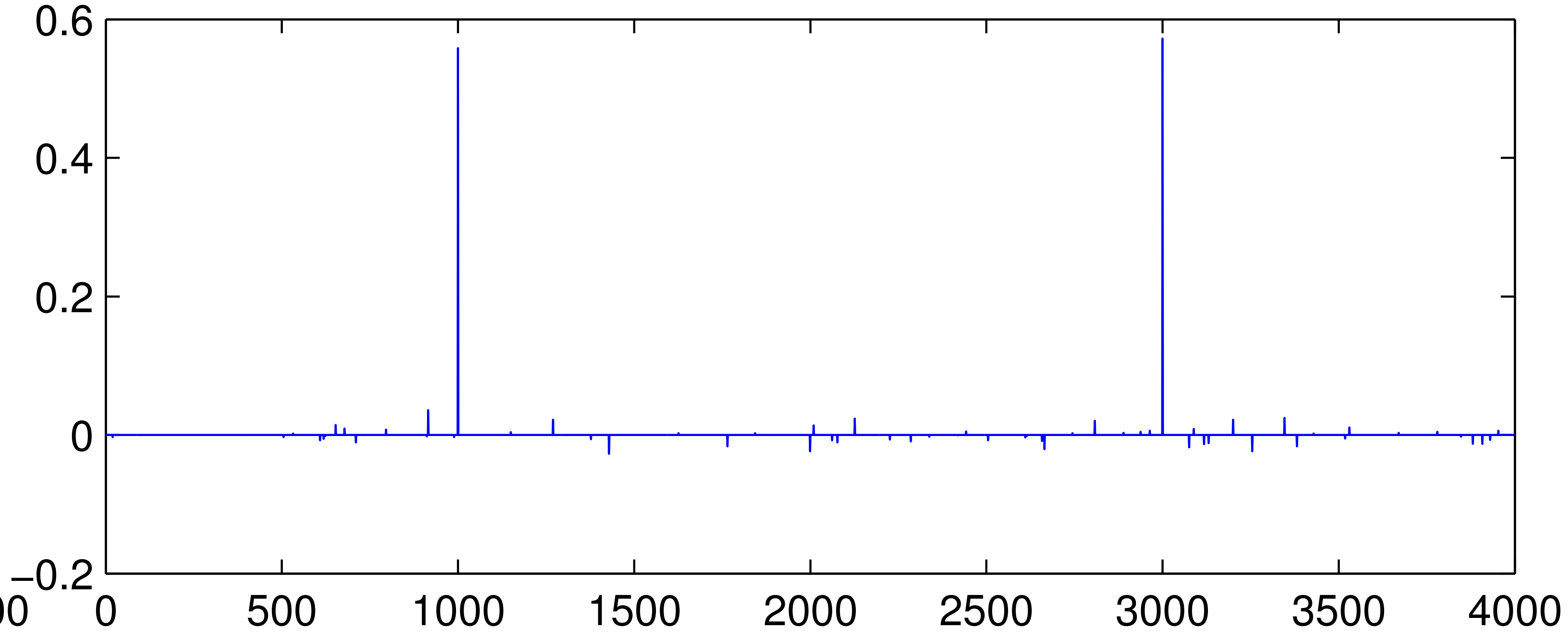
Iteration $t = 3$

Soft thresholding:

$x^t + A^* z^t$

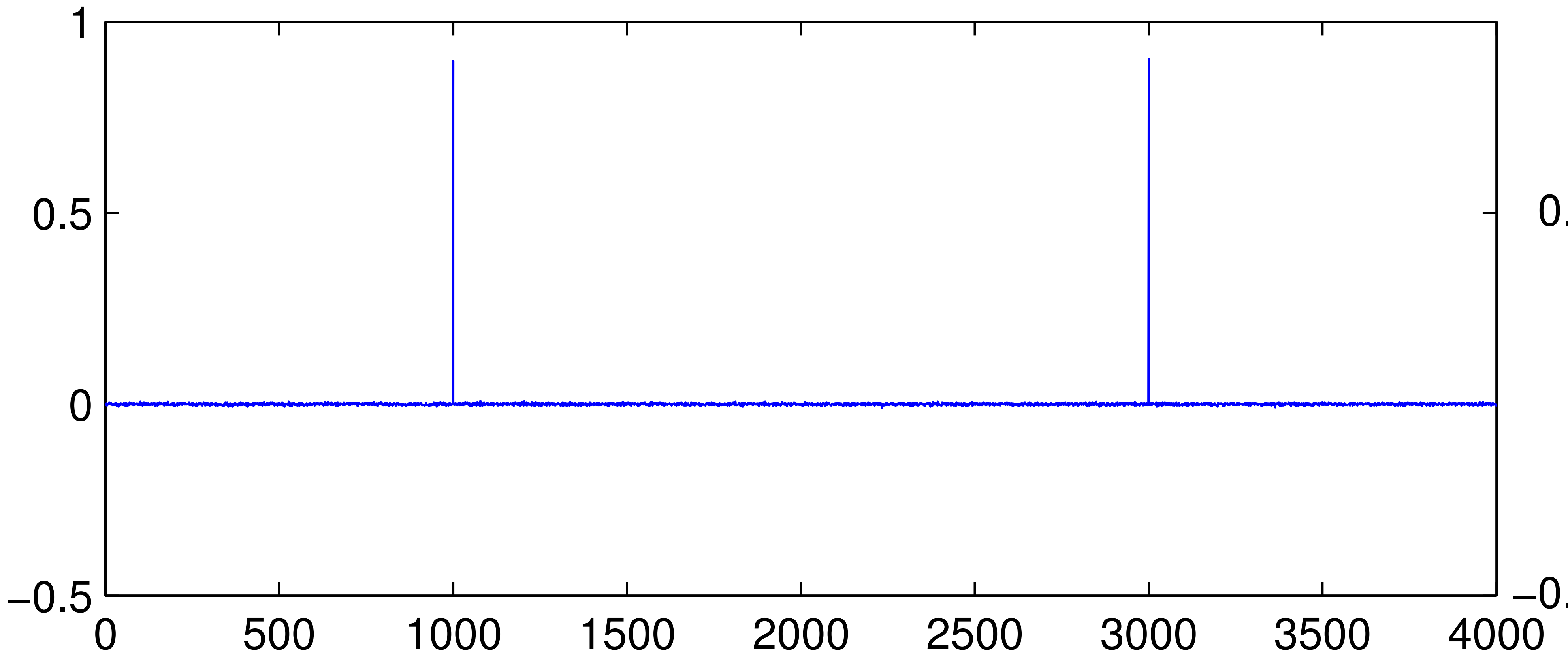


$\eta(x^t + A^* z^t; \tau^t)$

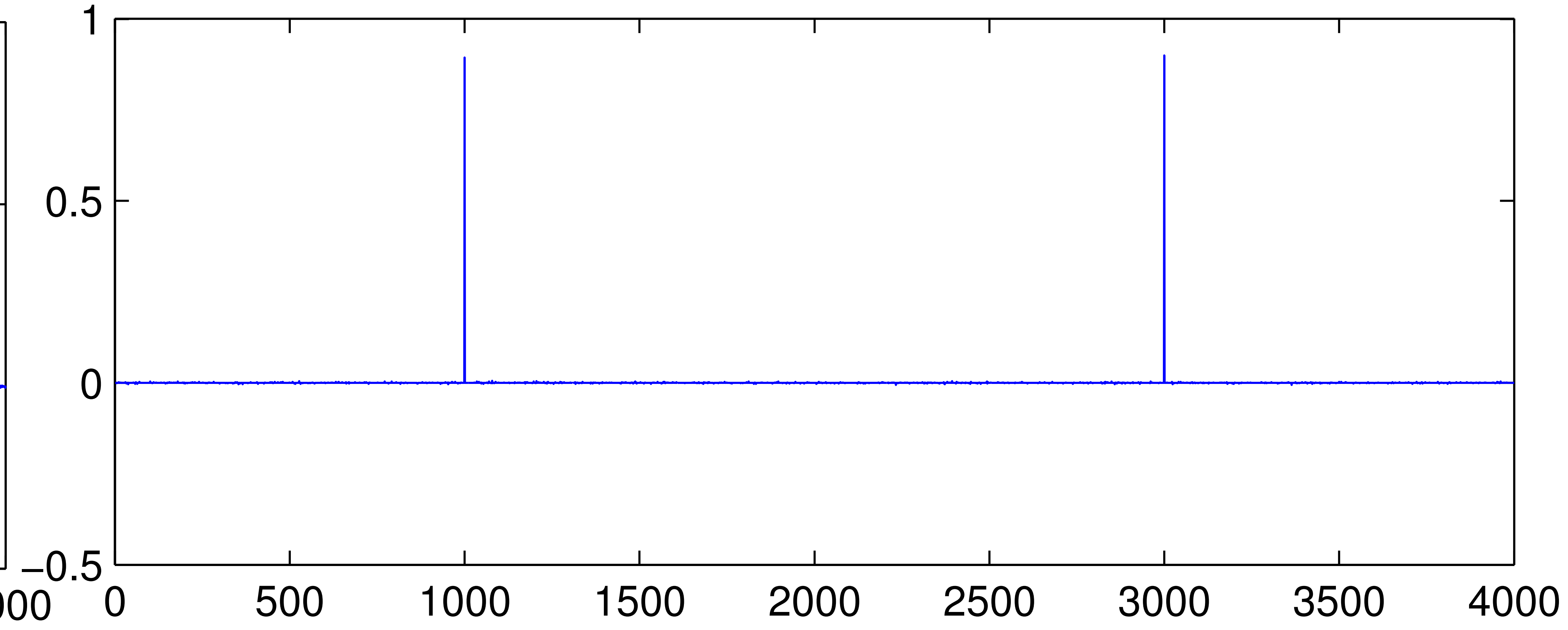


AMP:

$x^t + A^*(z^t + \alpha^t z^{t-1})$



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



Weighted AMP

Given a support estimate $\widetilde{T} \subseteq \{1, ..., N\}$, assume $\mathtt{w}_j = \omega < 1$ for $j \in \widetilde{T}$ and $\mathtt{w}_j = 1$ for $j \notin \widetilde{T}$.

We incorporate this information into the AMP algorithm by the following weighted AMP algorithm:

$$\begin{aligned} x^{t+1} &= \eta(x^t + A^* z^t; \hat{\tau}^t \mathtt{w}), \\ z^t &= y - Ax^t + \delta^{-1} z^{t-1} \langle \eta'(x^{t-1} + A^* z^{t-1}; \hat{\tau}^{t-1} \mathtt{w}) \rangle, \\ \hat{\tau}^t &= \frac{\hat{\tau}^{t-1}}{\delta} \langle \eta'(x^{t-1} + A^* z^{t-1}; \hat{\tau}^{t-1} \mathtt{w}) \rangle. \end{aligned} \tag{3}$$

Weighted AMP

Given a support estimate $\widetilde{T} \subseteq \{1, ..., N\}$, assume $\mathtt{w}_j = \omega < 1$ for $j \in \widetilde{T}$ and $\mathtt{w}_\mathtt{j} = 1$ for $j \notin \widetilde{T}$.

We incorporate this information into the AMP algorithm by the following weighted AMP algorithm:

$$\begin{aligned} x^{t+1} &= \eta(x^t + A^* z^t; \hat{\tau}^t \mathtt{w}), \\ z^t &= y - Ax^t + \delta^{-1} z^{t-1} \langle \eta'(x^{t-1} + A^* z^{t-1}; \hat{\tau}^{t-1} \mathtt{w}) \rangle, \\ \hat{\tau}^t &= \frac{\hat{\tau}^{t-1}}{\delta} \langle \eta'(x^{t-1} + A^* z^{t-1}; \hat{\tau}^{t-1} \mathtt{w}) \rangle. \end{aligned} \tag{3}$$

How does AMP work?

Assume $[\mathrm{w}_1, ..., \mathrm{w}_\mathrm{N}]^\mathrm{T}$ are the weights we use for the coefficients of the signal s.

Consider the following distribution over variables $s_1, \ s_2, \ \ldots, \ s_N$:

$$\mu(ds) = \frac{1}{Z} \prod_{i=1}^{N} exp(-\beta \mathrm{w}_i |s_i|) \prod_{a=1}^{m} \delta_{\{y_a = (As)_a\}}, \tag{4}$$

where $\delta_{\{y_a=(As)_a\}}$ denotes a Dirac distribution on the hyperplane $y_a = (As)_a$.

How does AMP work?

Assume $[\mathtt{w}_1, ..., \mathtt{w}_\mathrm{N}]^\mathrm{T}$ are the weights we use for the coefficients of the signal s.

Consider the following distribution over variables $s_1,\ s_2,\ \ldots,\ s_N$:

$$\mu(ds) = \frac{1}{Z} \prod_{i=1}^{N} exp(-\beta \mathtt{w}_i |s_i|) \prod_{a=1}^{m} \delta_{\{y_a=(As)_a\}}, \tag{4}$$

where $\delta_{\{y_a=(As)_a\}}$ denotes a Dirac distribution on the hyperplane $y_a = (As)_a$.

As $\beta \to \infty$ the mass of this distribution concentrates around the solutions of $y = As$ with smaller ℓ_1 norm.

How does AMP work?

Assume $[\mathtt{w}_1, ..., \mathtt{w}_\mathrm{N}]^\mathrm{T}$ are the weights we use for the coefficients of the signal s.

Consider the following distribution over variables $s_1, s_2, \ldots, s_N$:

$$\mu(ds) = \frac{1}{Z} \prod_{i=1}^{N} exp(-\beta \mathtt{w}_i |s_i|) \prod_{a=1}^{m} \delta_{\{y_a=(As)_a\}}, \tag{4}$$

where $\delta_{\{y_a=(As)_a\}}$ denotes a Dirac distribution on the hyperplane $y_a = (As)_a$.

As $\beta \to \infty$ the mass of this distribution concentrates around the solutions of $y = As$ with smaller ℓ_1 norm.

AMP and WAMP for seismic trace interpolation

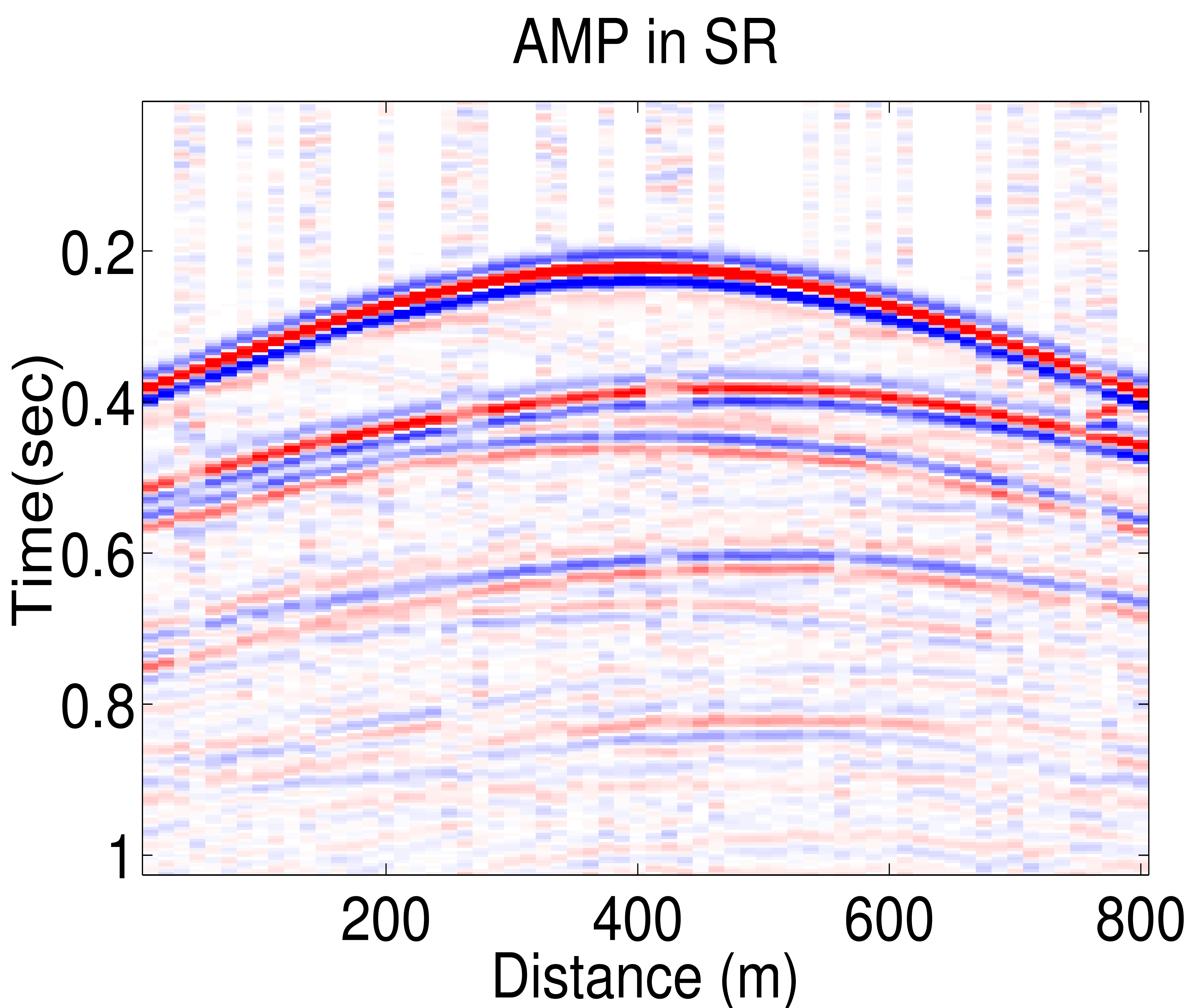
In order to use AMP and WAMP for seismic trace interpolation we use a 2-D DFT matrix as our sparsifying matrix.

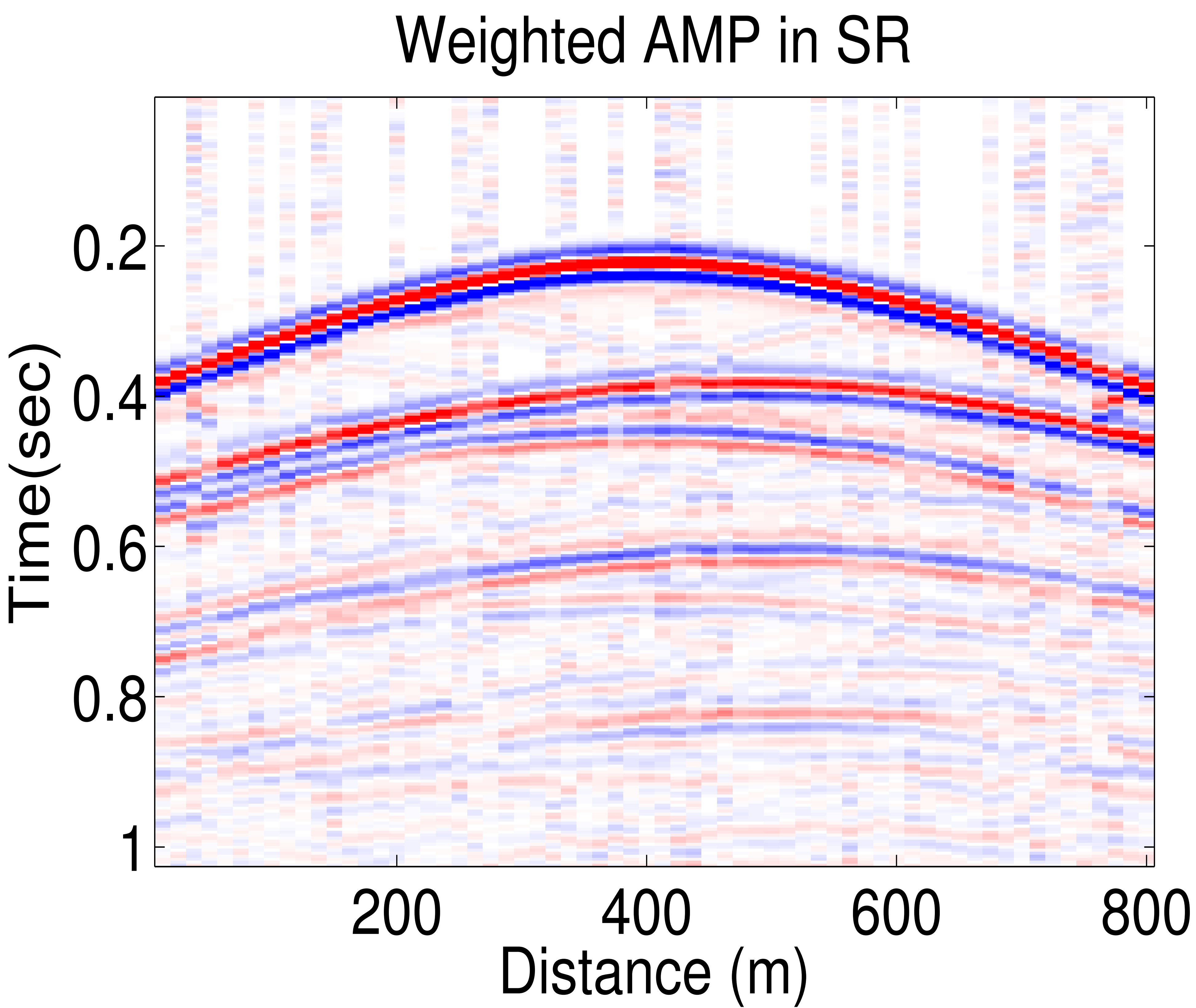
Then $b = RMF_s^H F_s f$, where F_s is a 2-D DFT matrix in the source-receiver domain.

AMP and WAMP for seismic trace interpolation

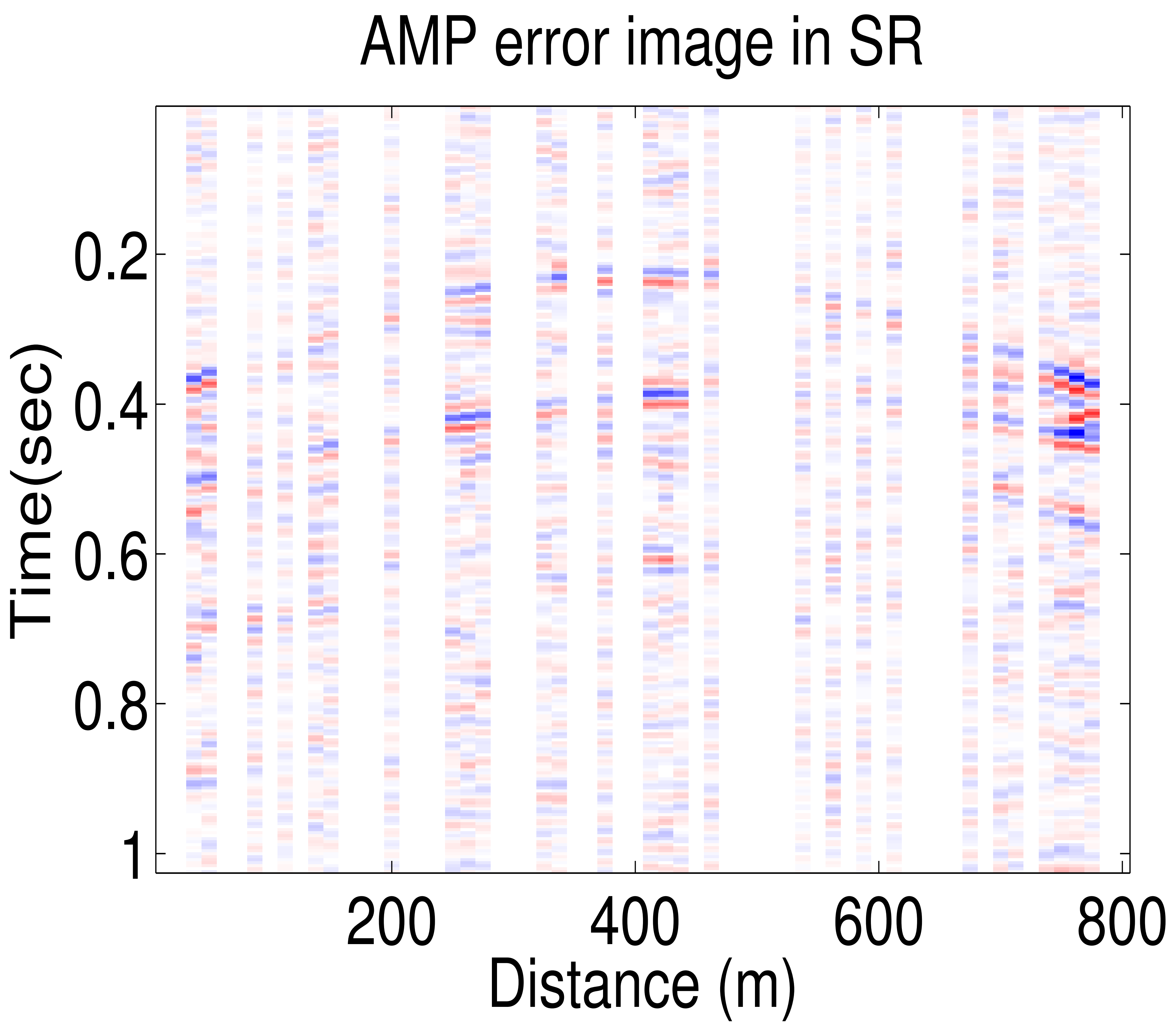
In order to use AMP and WAMP for seismic trace interpolation we use a 2-D DFT matrix as our sparsifying matrix.

Then $b = RMF_s^H F_s f$, where F_s is a 2-D DFT matrix in the source-receiver domain.

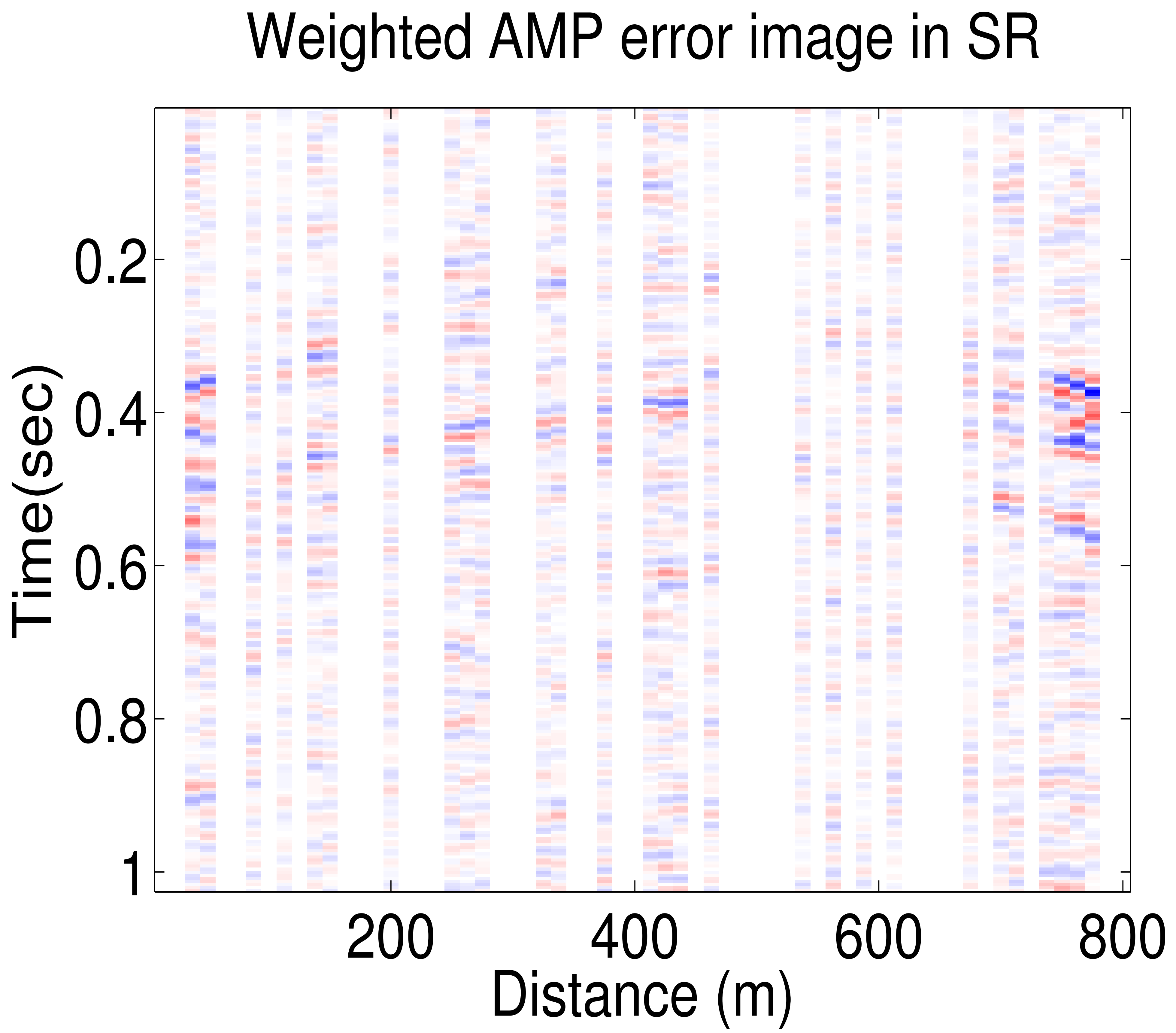




Recovery error: AMP vs weighted AMP



(o) SNR=10.2 dB



(p) SNR= 11.5 dB

A 2-stage algorithm for seismic trace interpolation

- By using simple calculations we can use the fast AMP algorithm to improve the recovery results of ℓ_1 minimization.

- $b = RMF_s^H F_s f$ is the measurements used for AMP algorithms

A 2-stage algorithm for seismic trace interpolation

- By using simple calculations we can use the fast AMP algorithm to improve the recovery results of ℓ_1 minimization.

- $b = RMF_s^H F_s f$ is the measurements used for AMP algorithms
- $b = RMS^H Sf$ are the measurements used for ℓ_1 minimization.

A 2-stage algorithm for seismic trace interpolation

- By using simple calculations we can use the fast AMP algorithm to improve the recovery results of ℓ_1 minimization.

- $b = RMF_s^H F_s f$ is the measurements used for AMP algorithms
- $b = RMS^H Sf$ are the measurements used for ℓ_1 minimization.

- Let x^* be the approximation obtained by solving the AMP algorithm then we have:

$$
\begin{aligned}
x^* &\approx F_s f \\
F_s^H x^* &\approx f \\
SF_s^H x^* &\approx Sf
\end{aligned}
\tag{5}
$$

Hence we can use $SF_s^H x^*$ as an approximation for the ℓ_1 solver.

A 2-stage algorithm for seismic trace interpolation

- By using simple calculations we can use the fast AMP algorithm to improve the recovery results of ℓ_1 minimization.

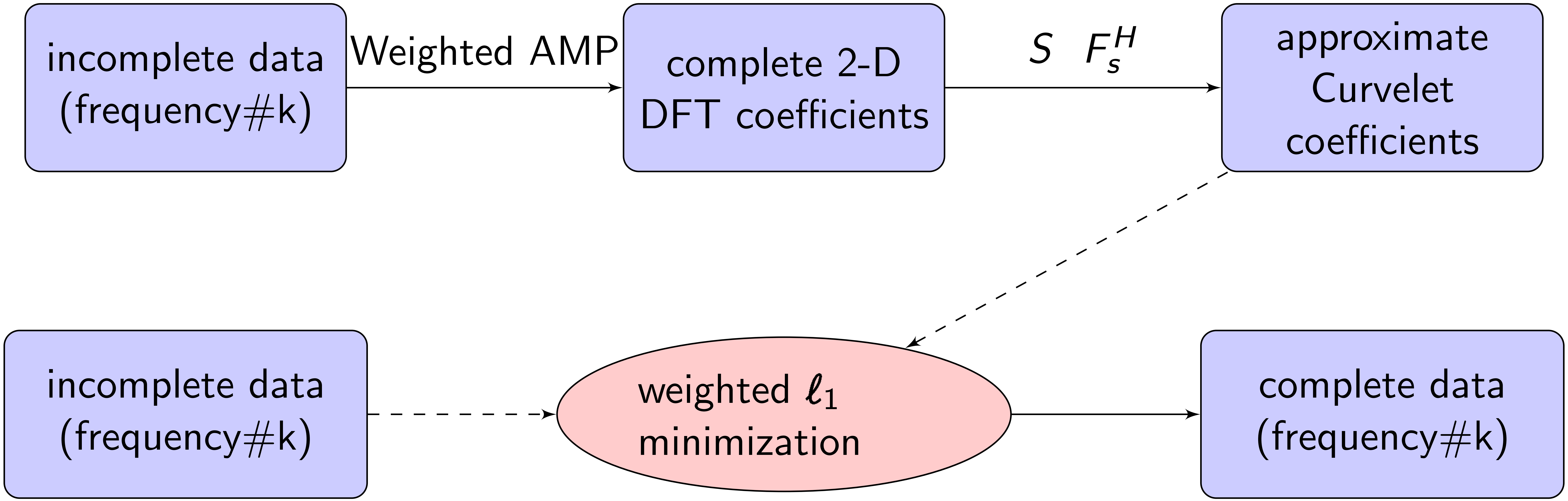
- $b = RMF_s^H F_s f$ is the measurements used for AMP algorithms
- $b = RMS^H Sf$ are the measurements used for ℓ_1 minimization.

- Let x^* be the approximation obtained by solving the AMP algorithm then we have:

$$\begin{aligned} x^* &\approx F_s f \\ F_s^H x^* &\approx f \\ SF_s^H x^* &\approx Sf \end{aligned} \tag{5}$$

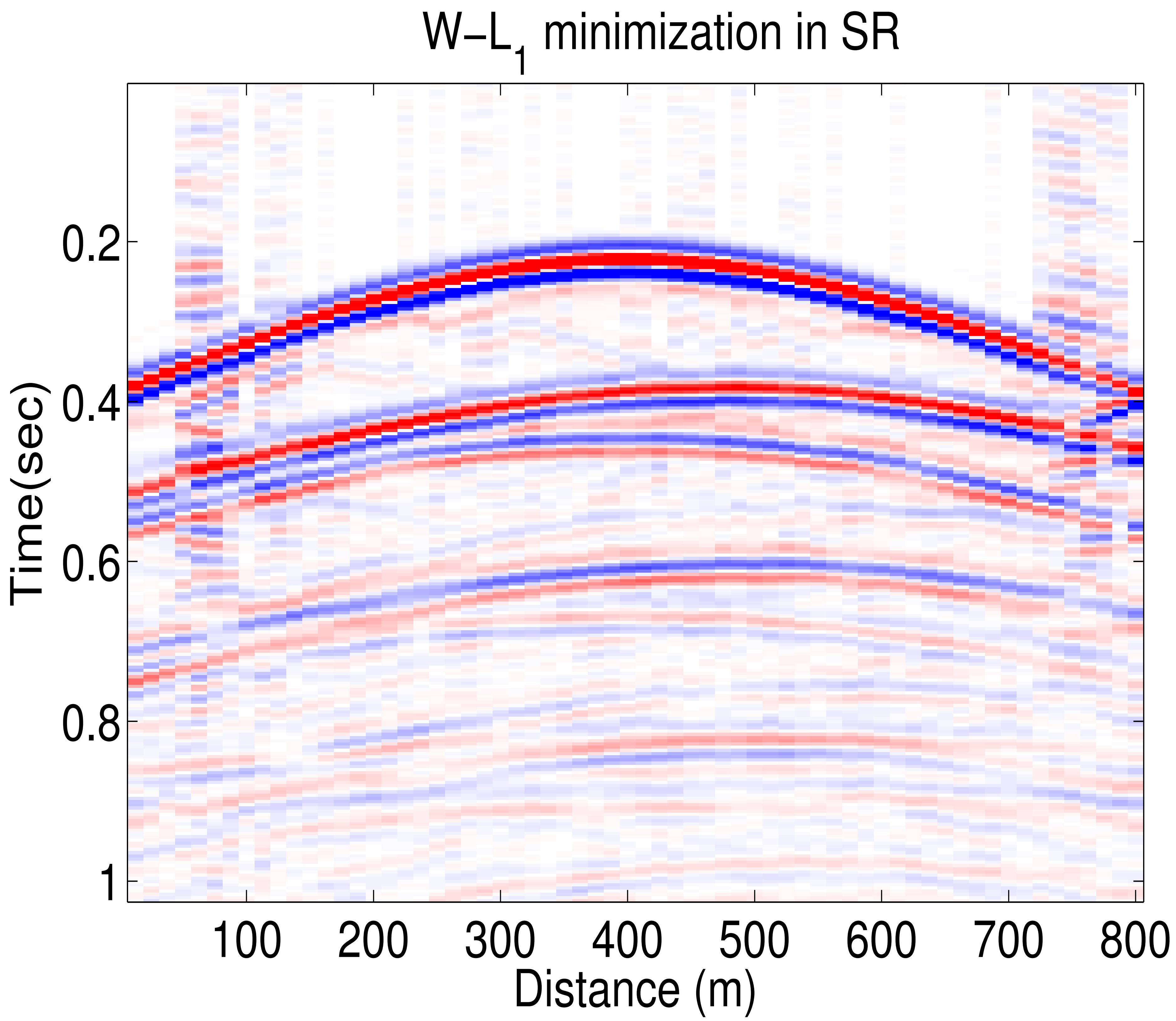
Hence we can use $SF_s^H x^*$ as an approximation for the ℓ_1 solver.

Flowchart of the 2-stage algorithm WAMP+weighted ℓ_1

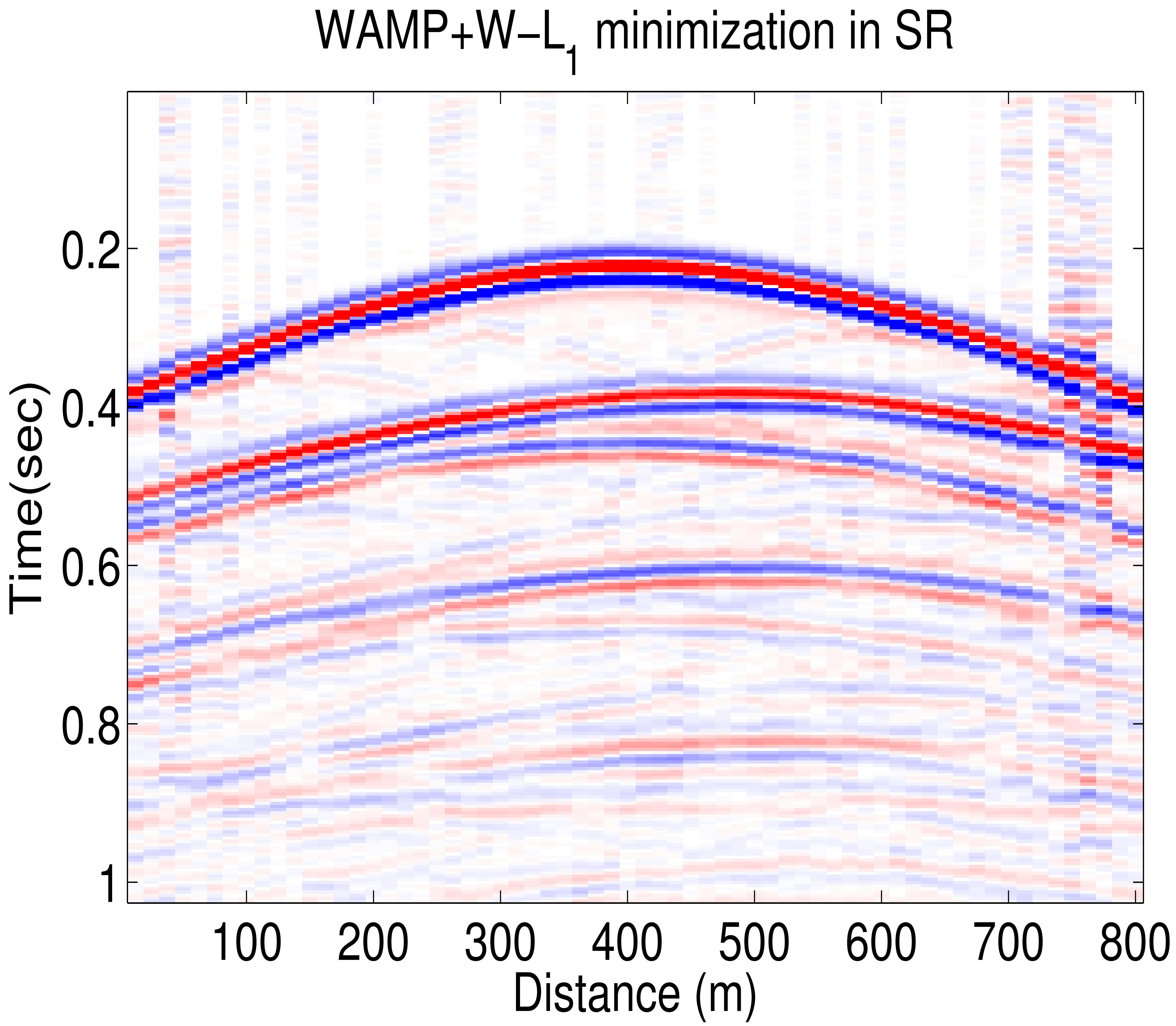


Weighted ℓ_1 vs 2-stage WAMP+weighted ℓ_1

In the 2-stage algorithm, for each frequency slice we first apply a fast WAMP algorithm and use the result to derive new weights for weighted ℓ_1 minimizer with Curvelet coefficients.

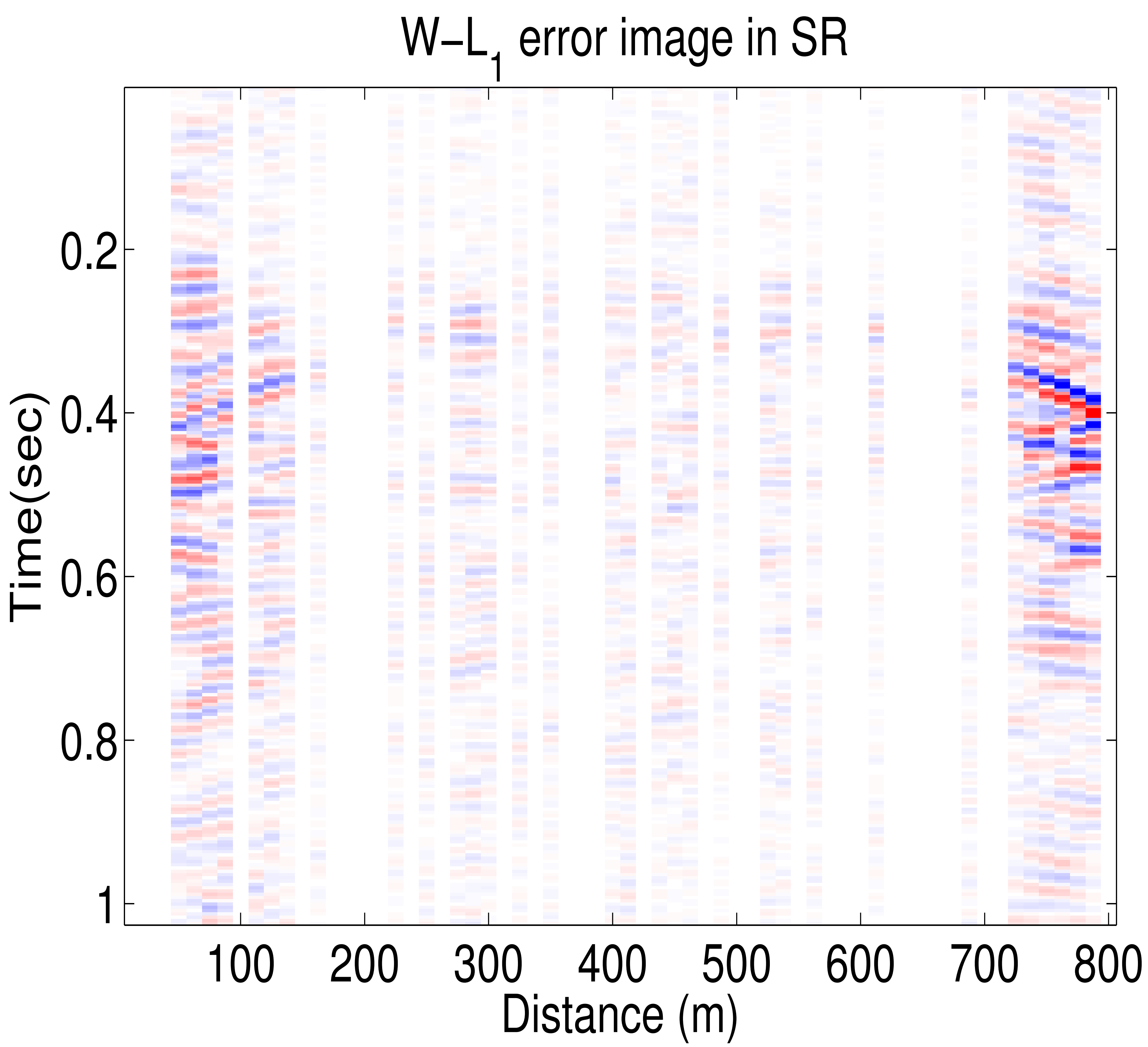


(q)

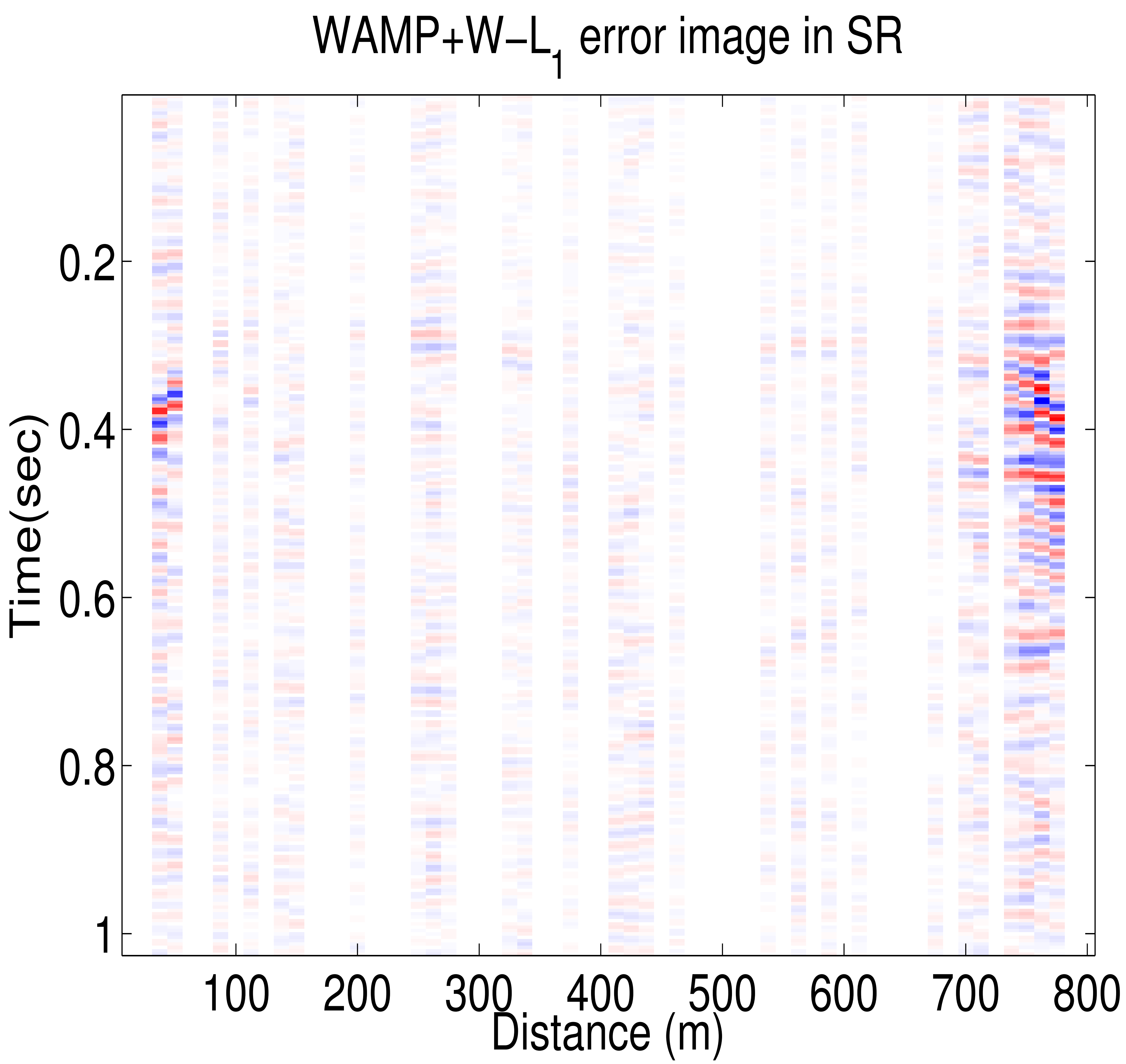


(r)

Recovery error: AMP vs weighted AMP

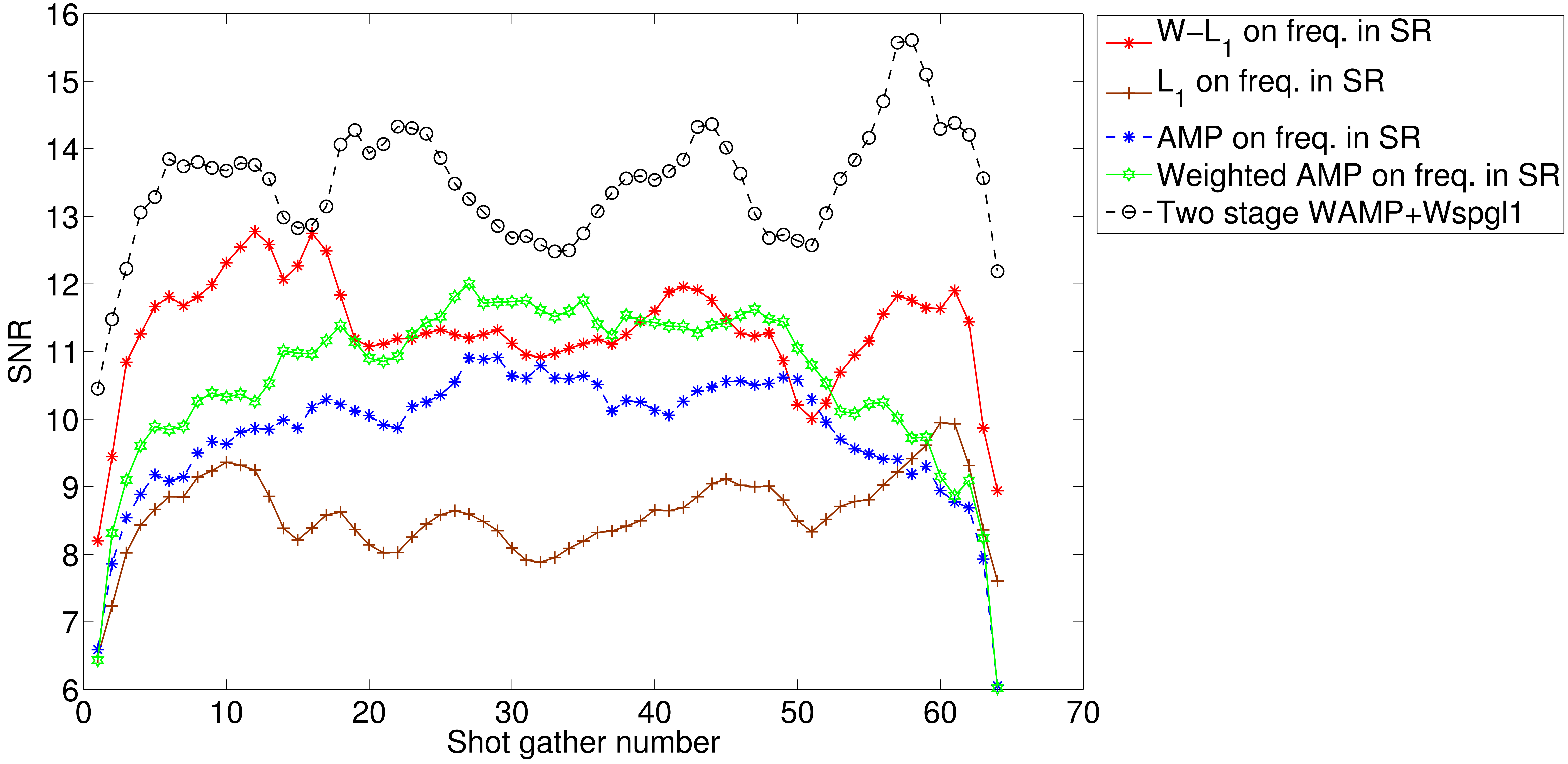


(s) SNR=10.2 dB



(t) SNR= 12.8 dB

Comparison of shotgather SNRs



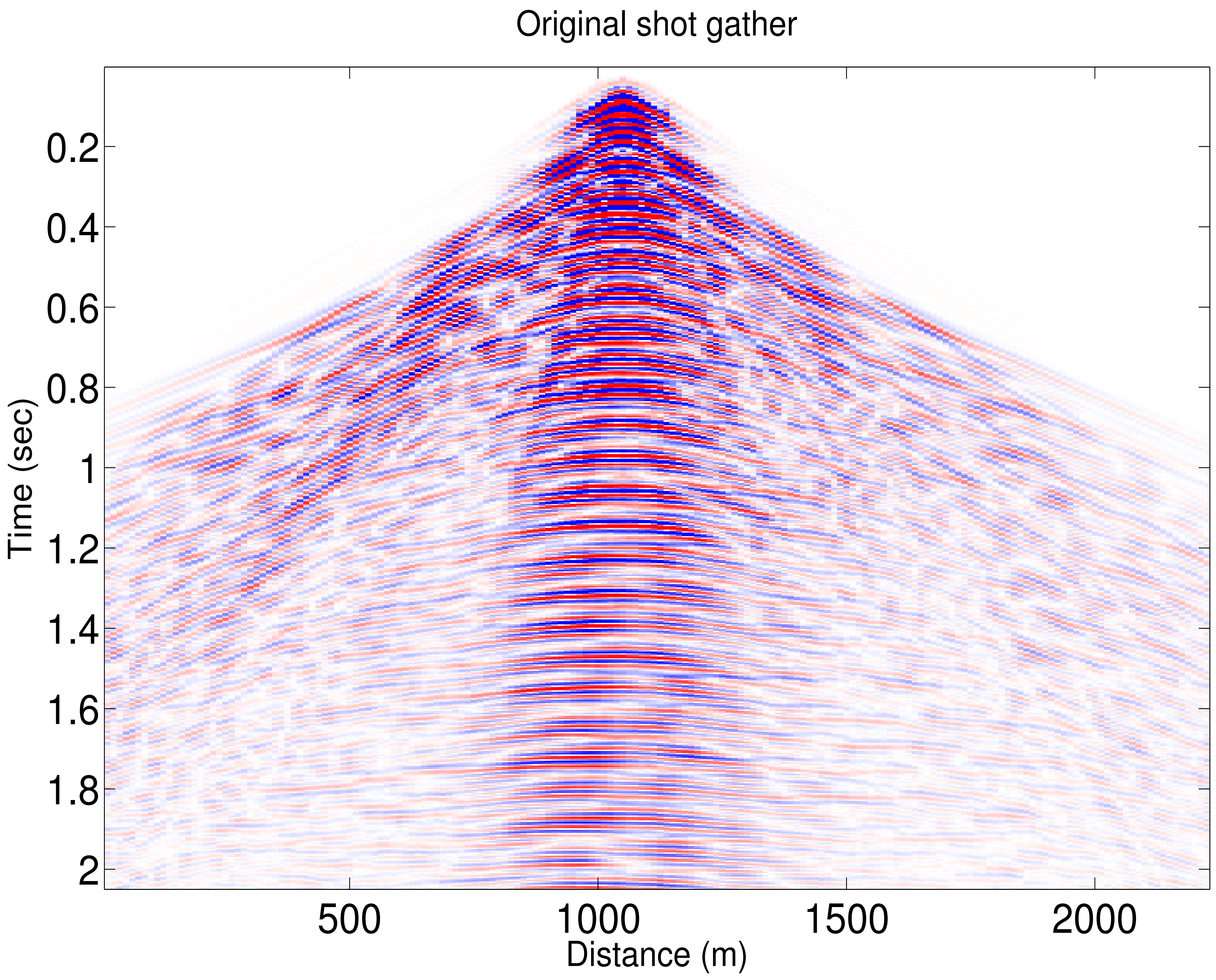
Results on the Gulf of Suez data

In the next slides we show the results of applying these algorithms on a seismic line from the Gulf of Suez.

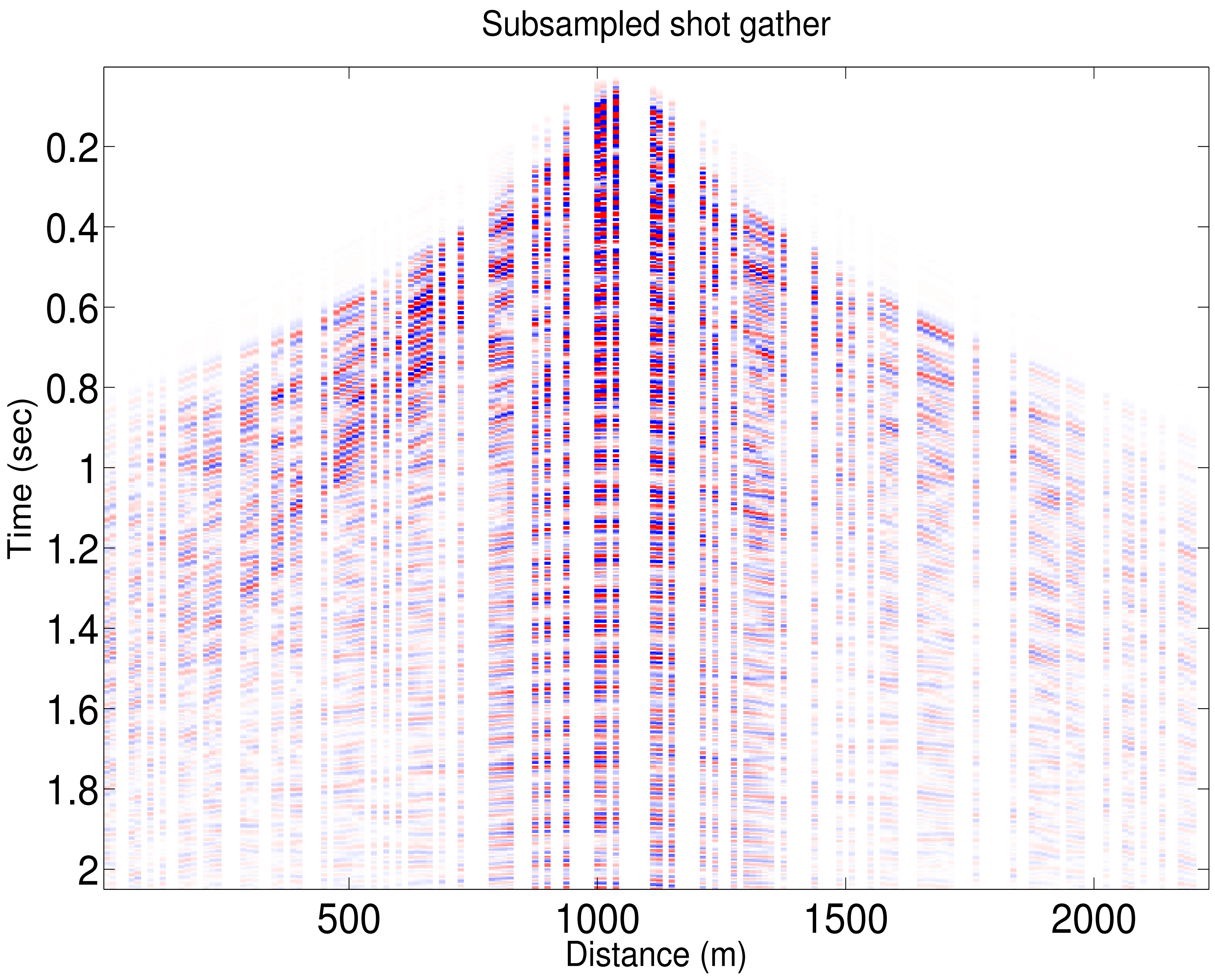
The Seismic line at full resolution has $N_s = 178$ sources, $N_r = 178$ receivers with a sample distance of 12.5 meters, and $N_t = 512$ time samples acquired with a sampling interval of 4 milliseconds. Consequently, the seismic line contains samples collected in a 2s temporal window with a maximum frequency of 125 Hz.

Results on the Gulf of Suez data

Shotgather number 84 from the seismic line:

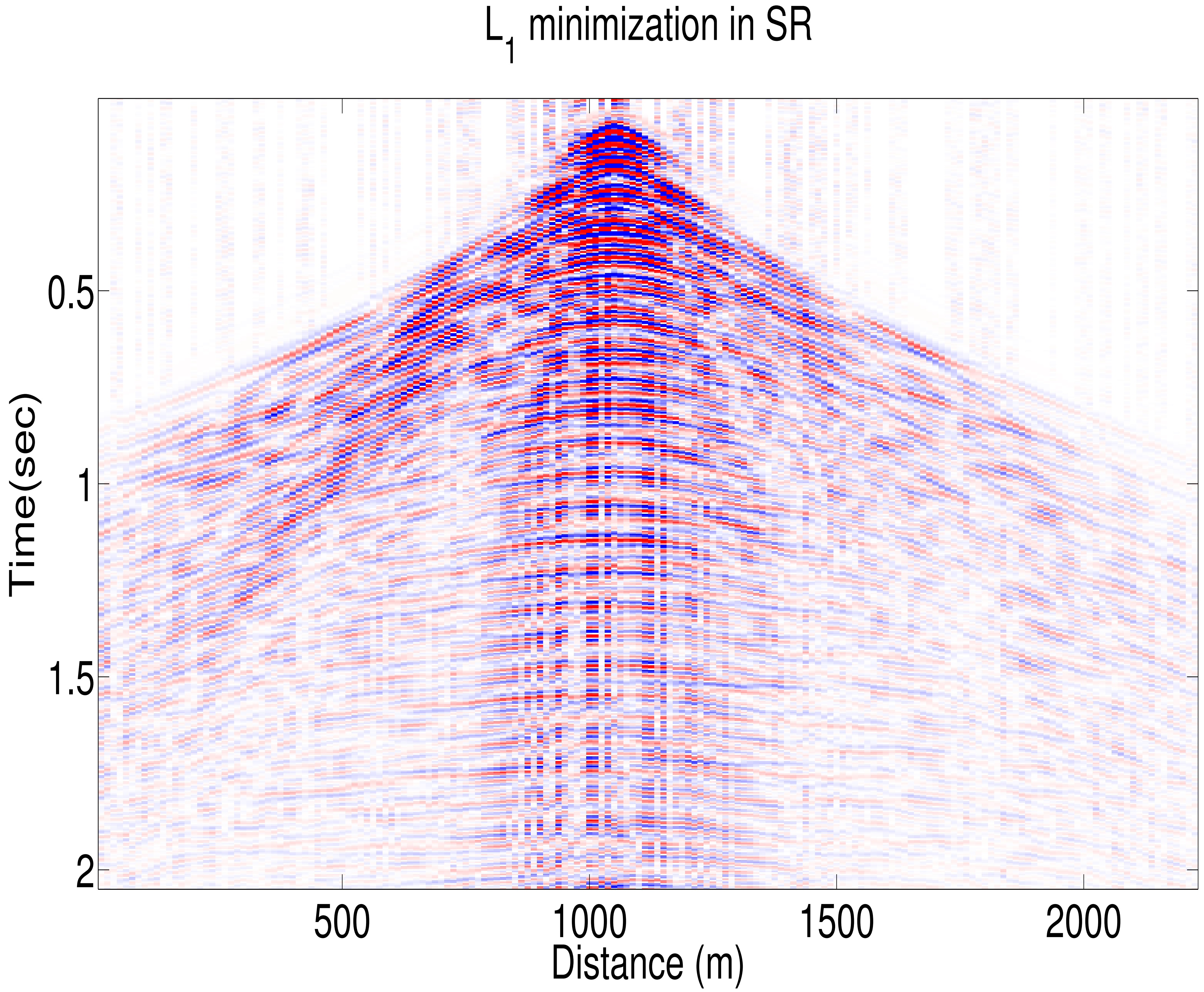


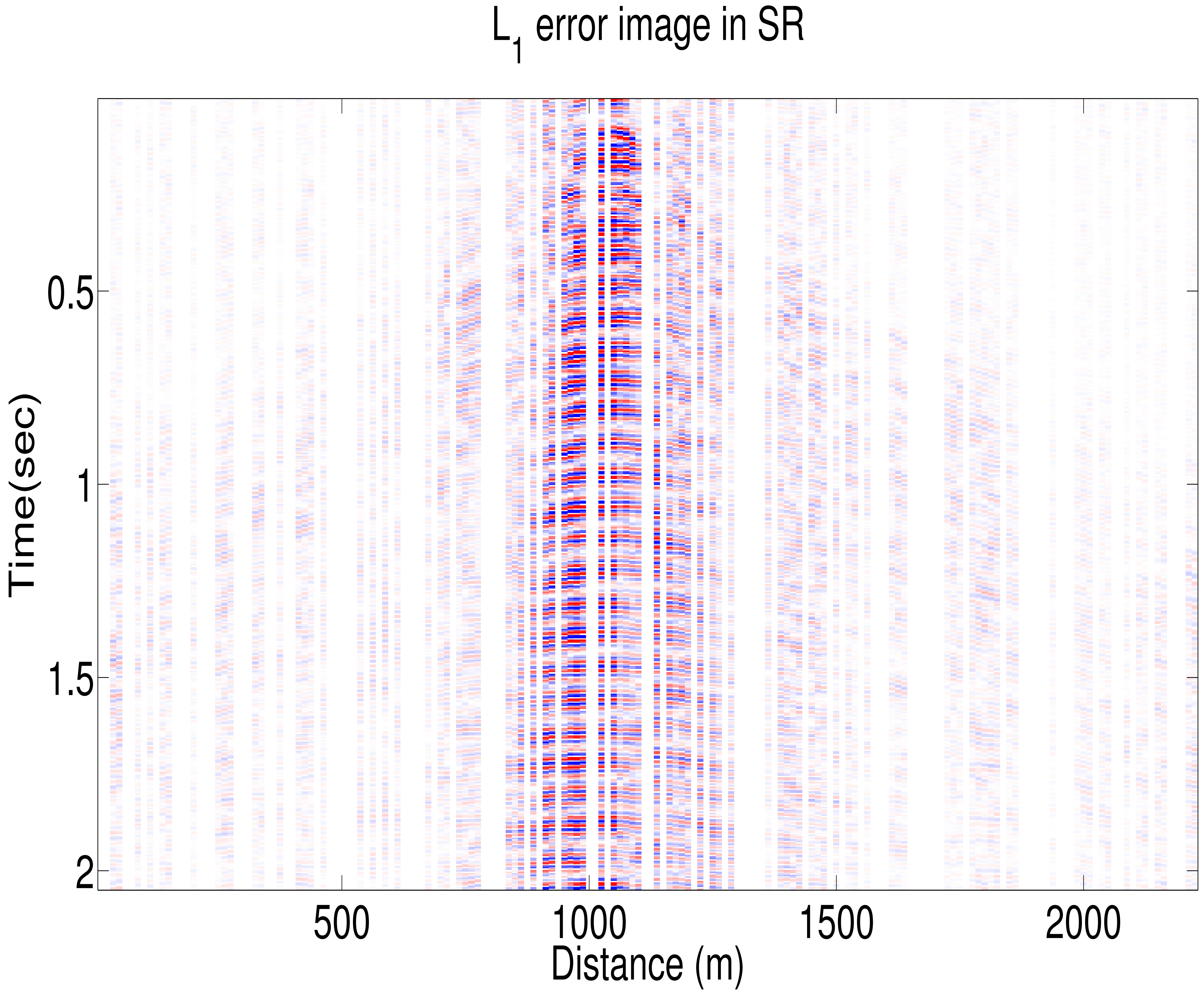
(u)



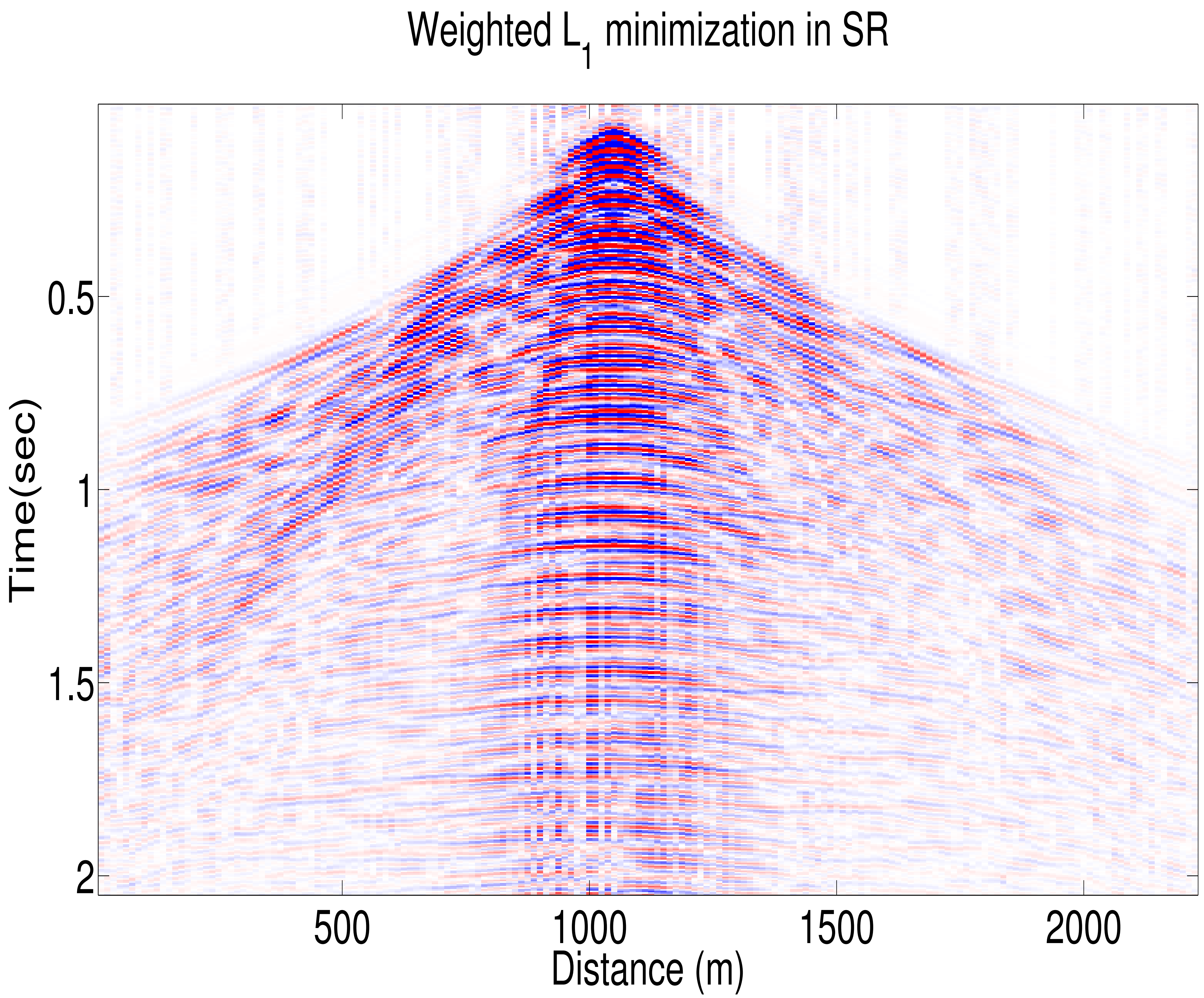
(v)

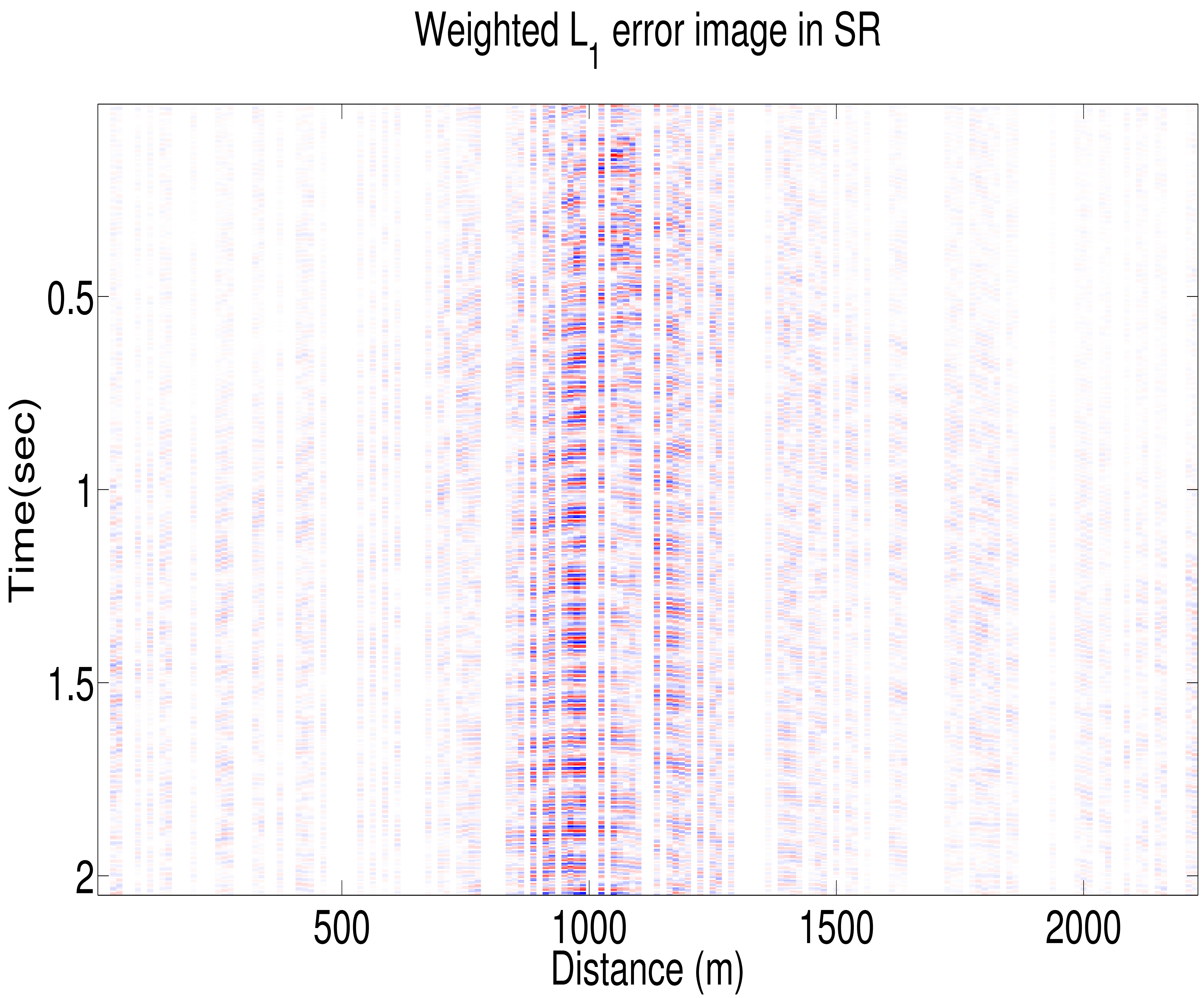
ℓ_1 minimization



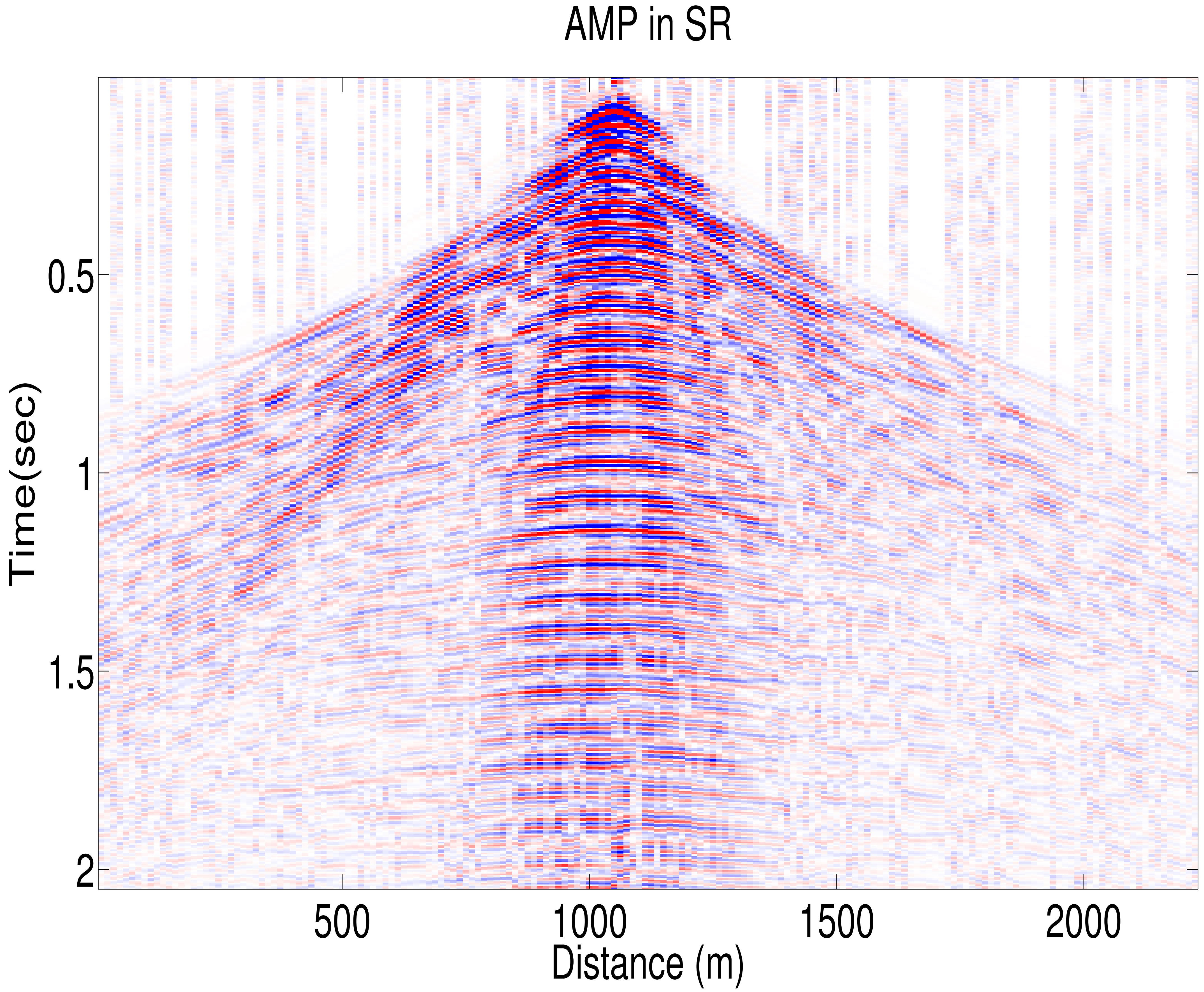


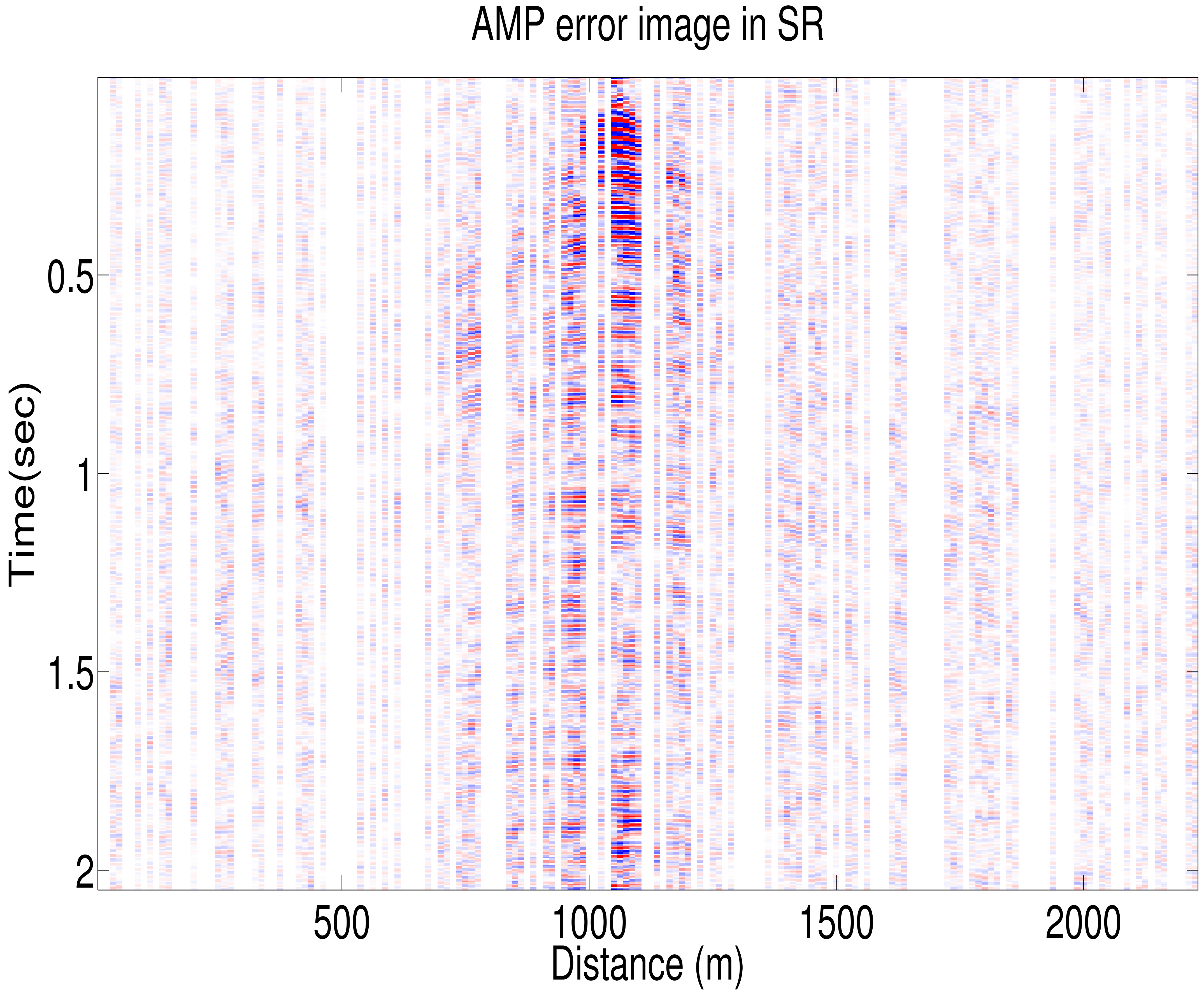
Weighted ℓ_1 minimization



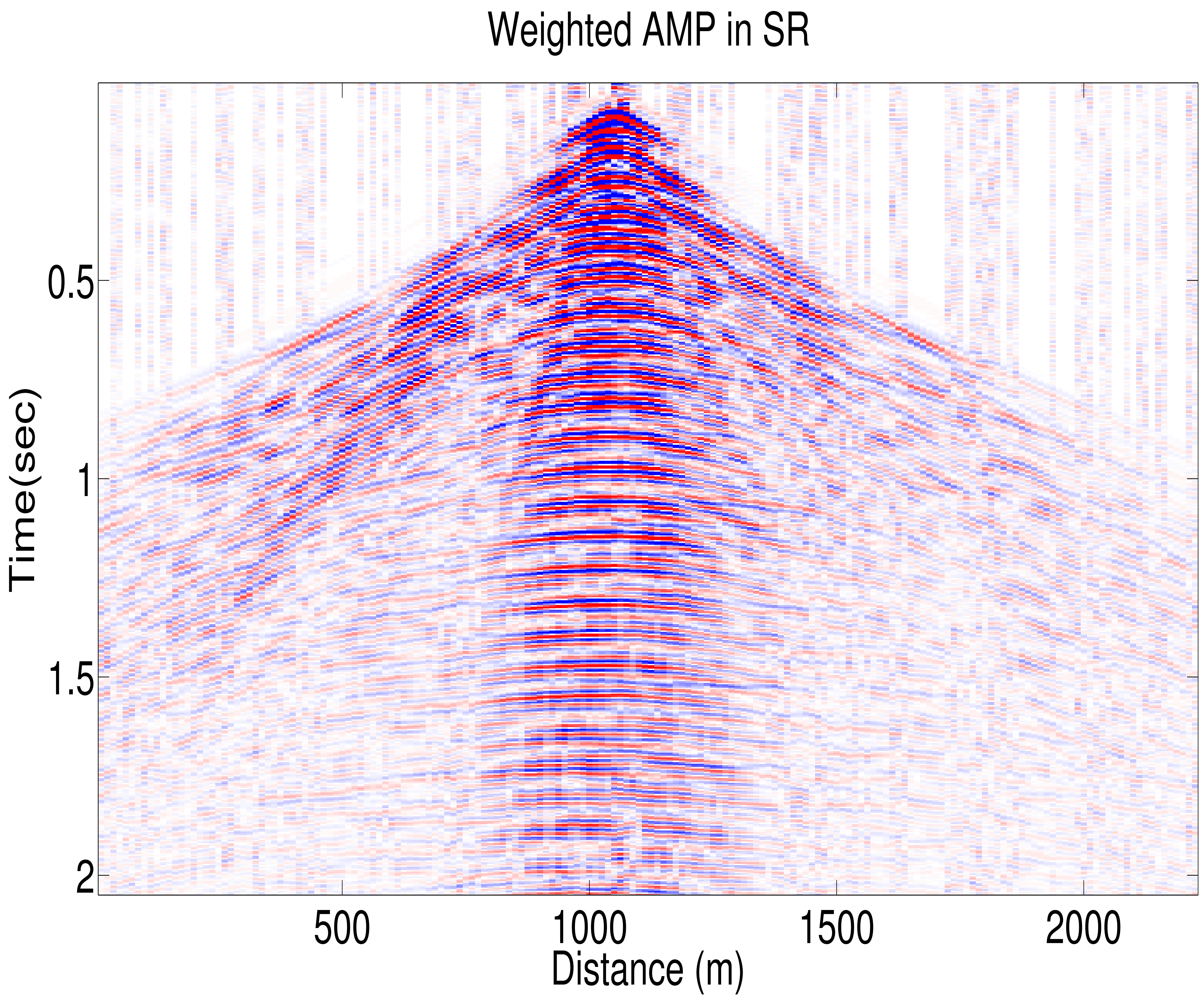


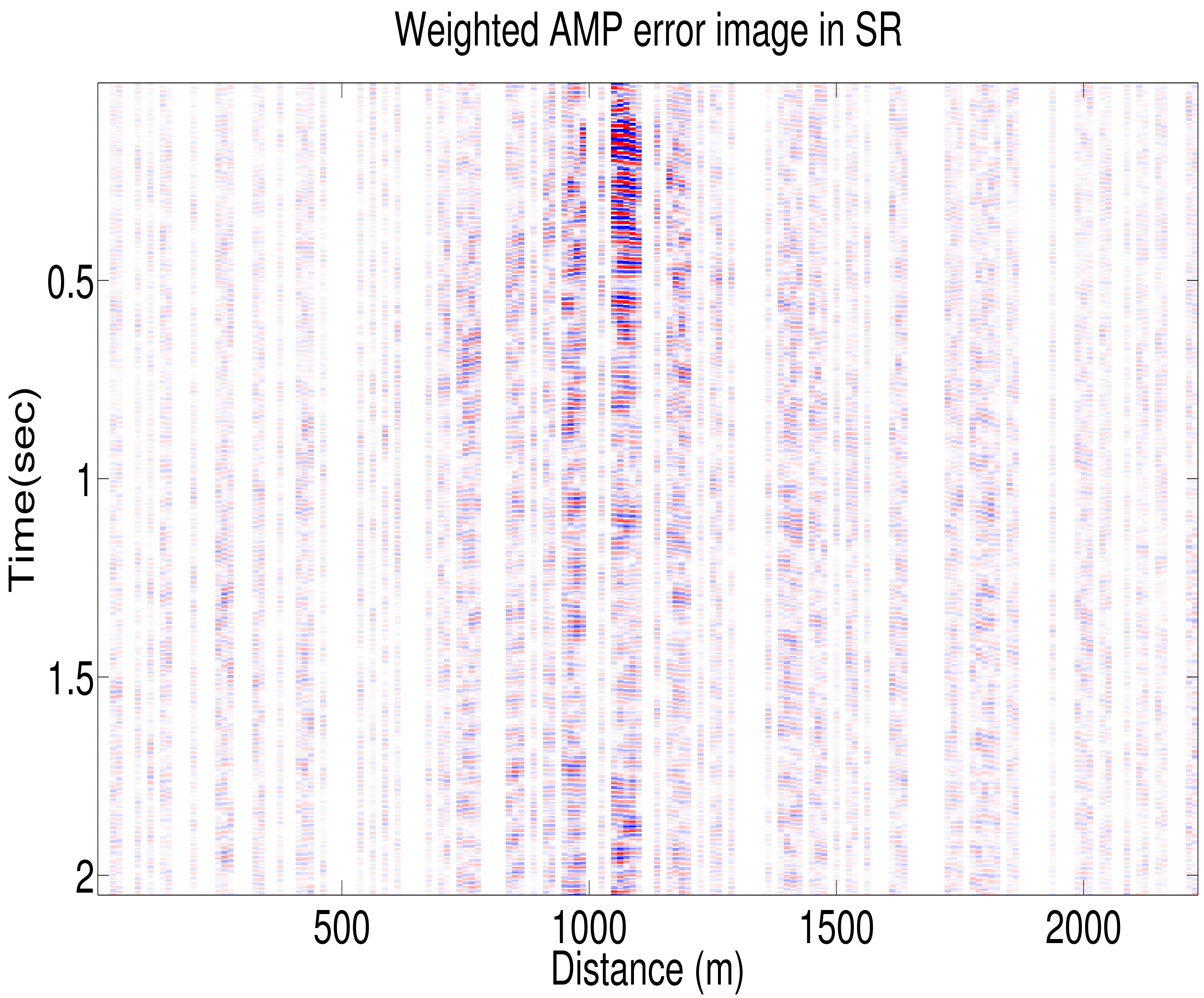
AMP


AMP in SR
Time(sec)
0.5
1
1.5
2
500
1000
1500
2000
Distance (m)

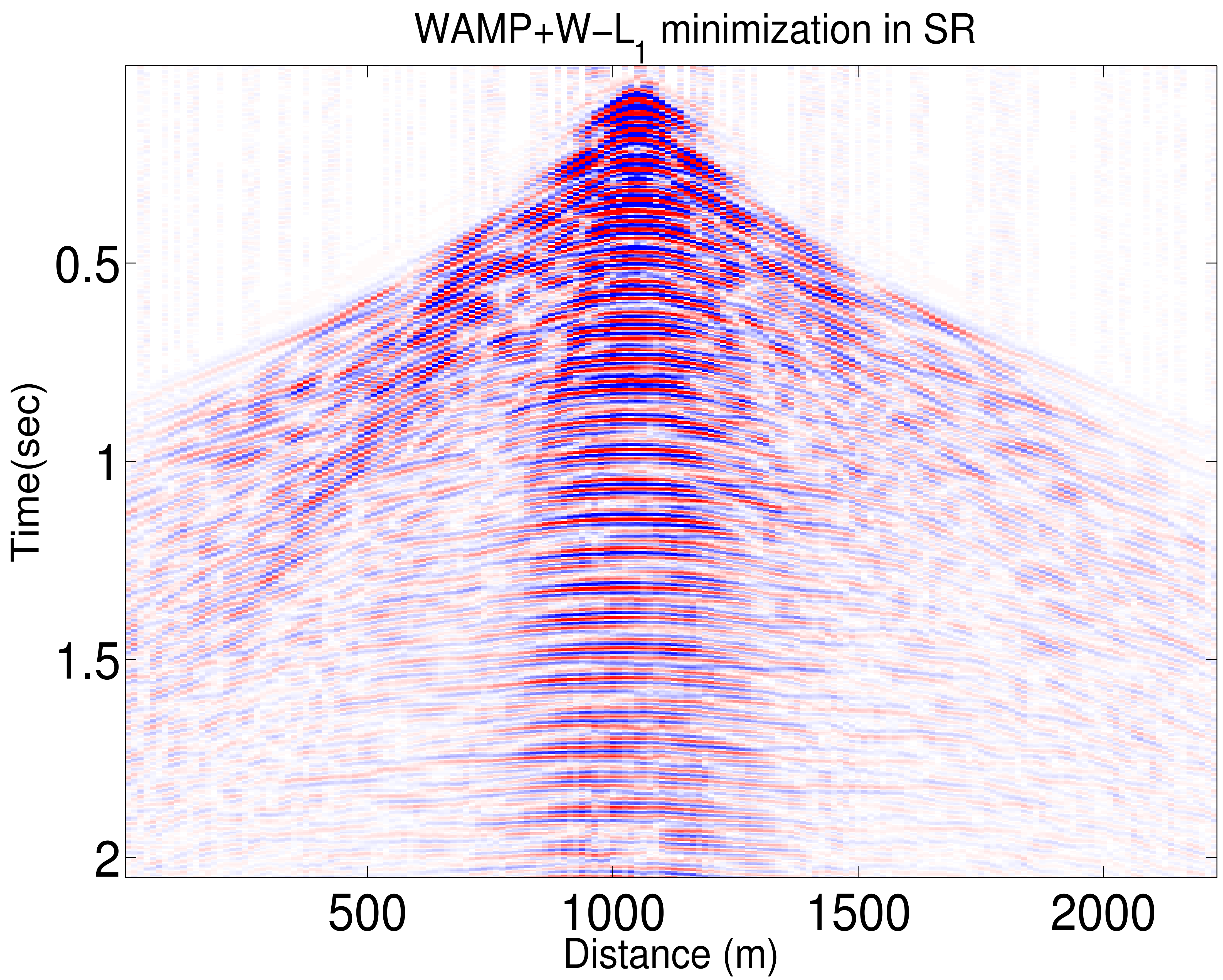

AMP error image in SR
Time(sec)
0.5
1
1.5
2
500
1000
1500
2000
Distance (m)

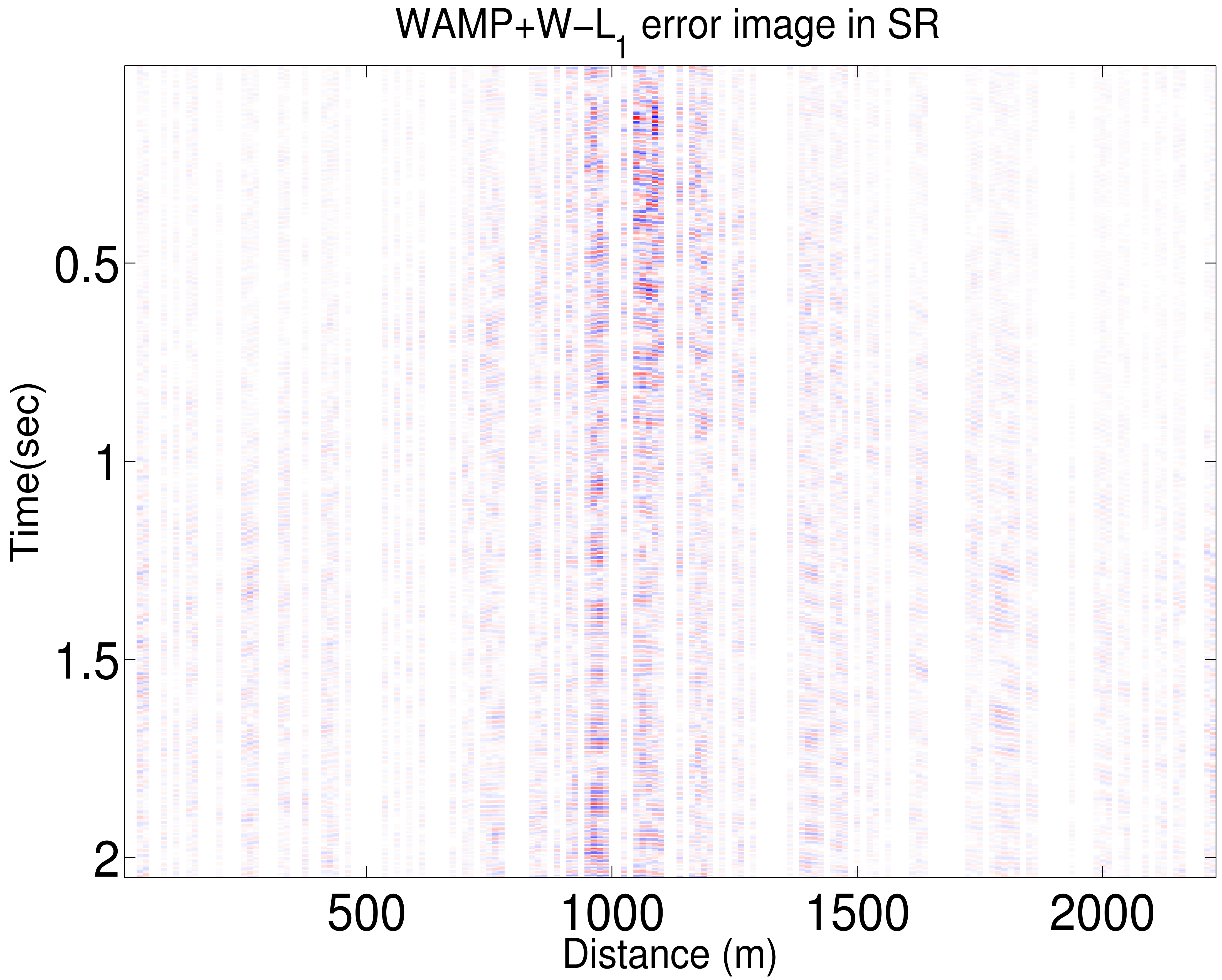
Weighted AMP



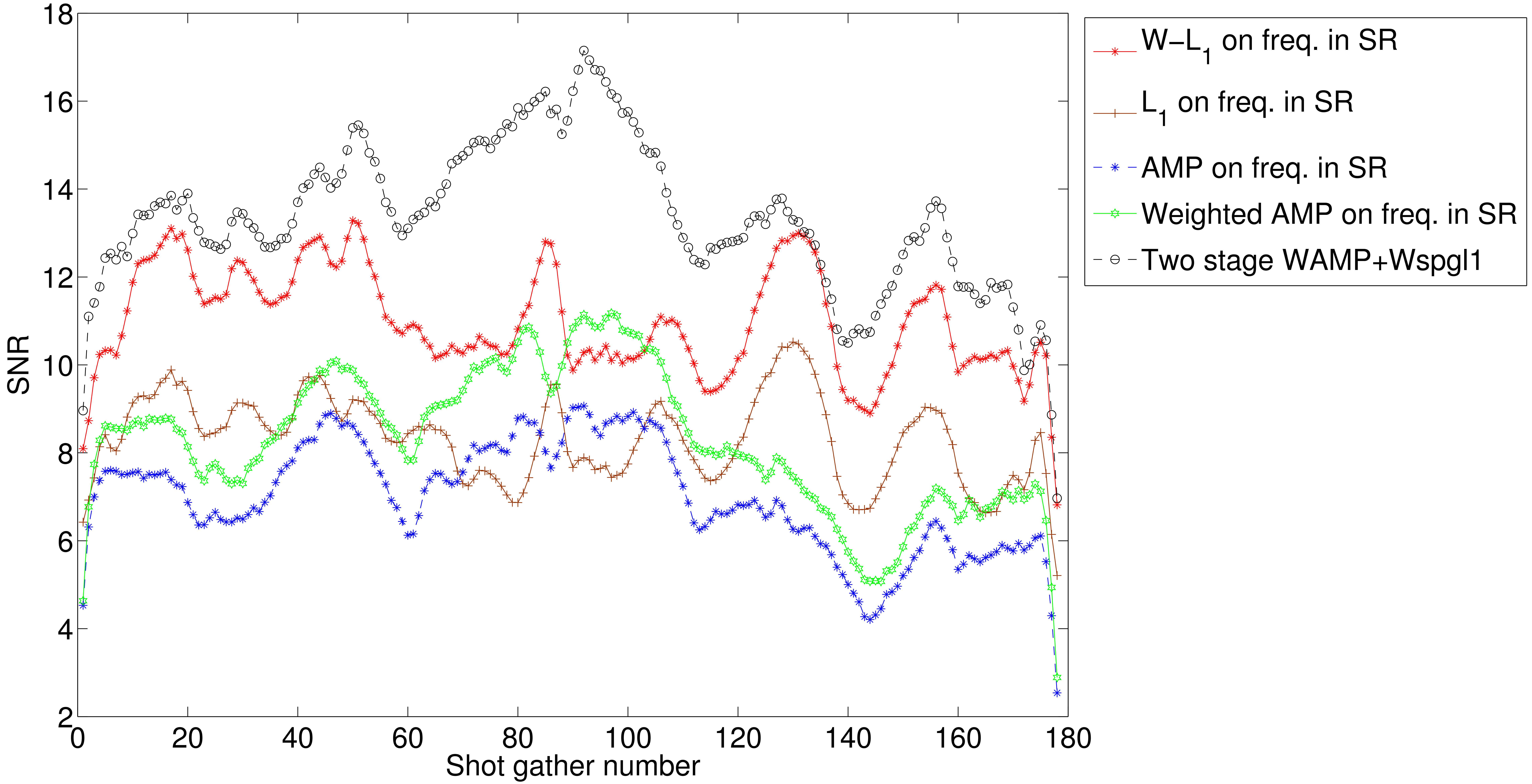


Weighted AMP+Weighted ℓ_1





Shotgather SNRs



SLIM

Acknowledgements

Thank you for your attention !

https://www.slim.eos.ubc.ca/

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, CGG, Chevron, ConocoPhillips, ION, Petrobras, PGS, Statoil, Total SA, WesternGeco, and Woodside.

Acknowledgement

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.