Seismic trace interpolation with approximate message passing

Navid Ghadermarzy

University of British Columbia

October 28, 2014

Collaborators

Joint work with:

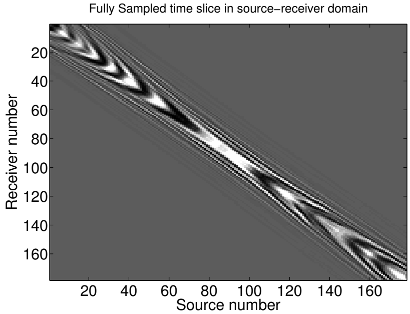
- Felix Herrmann
- Özgür Yılmaz

Main messages and motivation

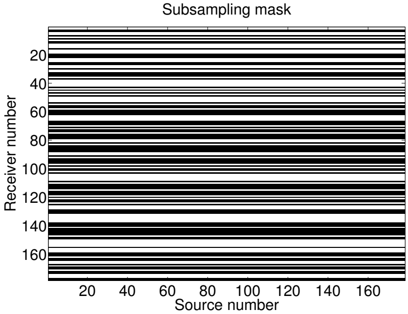
- We do seismic trace interpolation with approximate message passing (AMP).
- We solve the problem in Fourier domain.
- AMP is faster than ℓ_1.
- However, AMP can not be applied to measurements in curvelet domain.
- We use AMP (in Fourier domain) to get a very fast interpolation.
- We use this interpolation as an initial point for ℓ_1 in curvelet domain.
- We get better results in less time by using AMP to find a good initial point.

ℓ_1 for randomized acquisition of seismic lines

- Consider a seismic line from the Gulf of Suez with 178 sources, 178 receivers with a sample distance of 12.5 meters
- 512 time samples collected in a 2s temporal window.
- 50% of the receiver spread is randomly subsampled using the mask R.



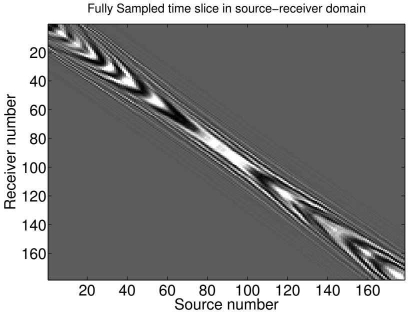
(a) Fully sampled time slice f at $t = 0.3125s$



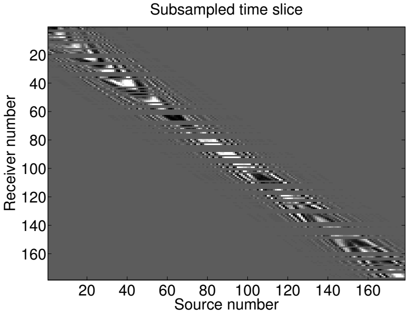
(b) Mask R

ℓ_1 for randomized acquisition of seismic lines

- Consider a seismic line from the Gulf of Suez with 178 sources, 178 receivers with a sample distance of 12.5 meters
- 512 time samples collected in a 2s temporal window.
- 50% of the receiver spread is randomly subsampled using the mask R.



(c) Fully sampled time slice f at $t = 0.3125s$



(d) Subsampled time slice

Using 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 ($S^H S = \mathbb{I}$).

Using 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

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 ($S^H S = \mathbb{I}$).

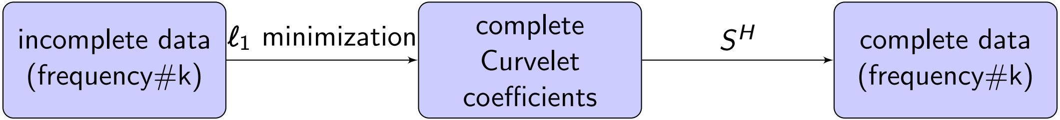
Then $b = RMS^H x$, where x can be recovered by sparse recovery algorithms like ℓ_1 minimization.

Using ℓ_1 for seismic trace interpolation

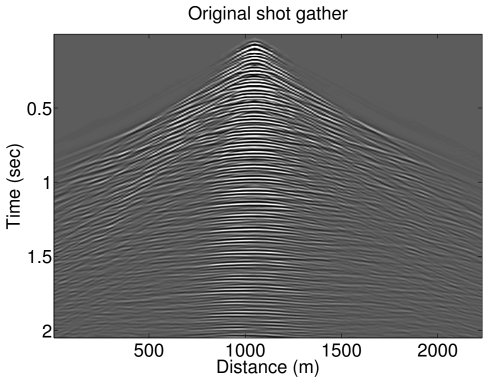
To recover f from the measurements $b = RMS^H x$, we solve the ℓ_1 minimization problem

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

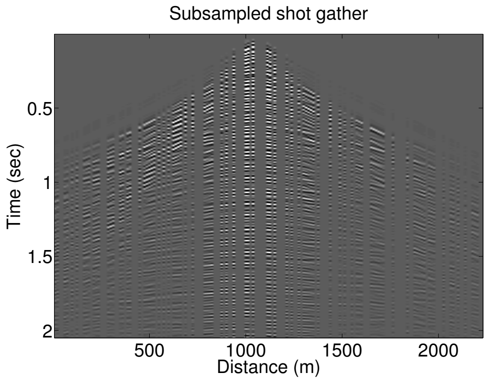
and approximate f by $S^H x^{\ell_1}$.



Recovery using ℓ_1 minimization on frequency slices (shotgather # 84)

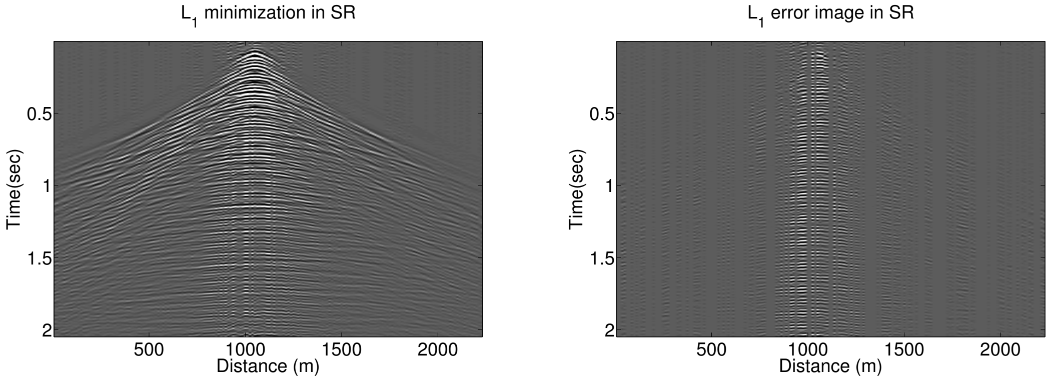


(e)

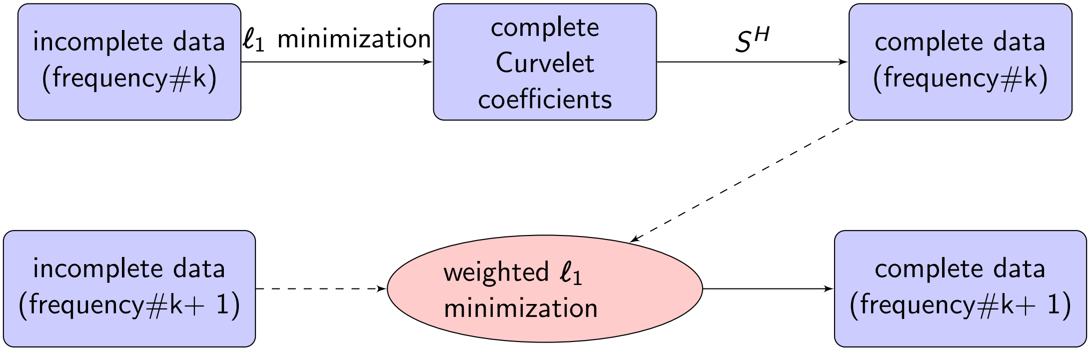


(f)

Recovery results: ℓ_1 minimization (8.4 dB)



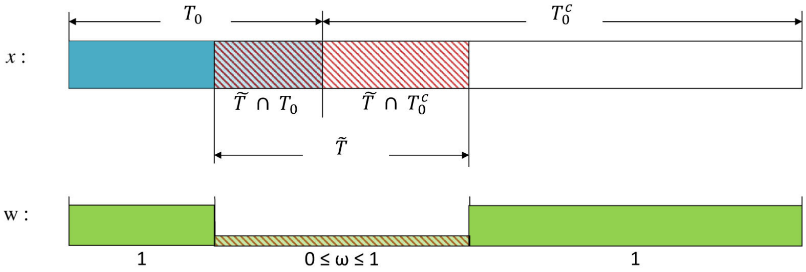
Weighted ℓ_1 for seismic trace interpolation



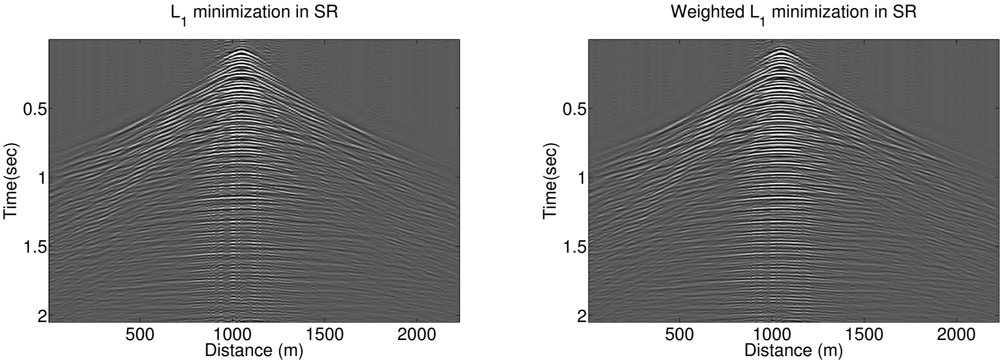
Weighted ℓ_1 for seismic trace interpolation

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

- 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 $\mathrm{w}_i = 1$.



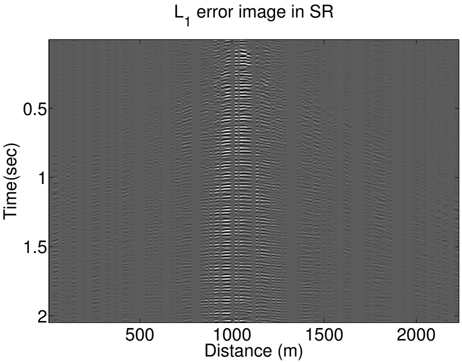
Weighted ℓ_1 for seismic trace interpolation



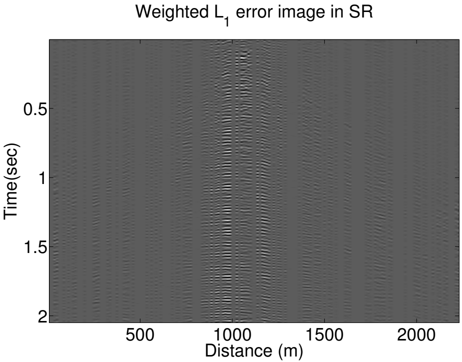
(i) SNR=8.4 dB

(j) SNR= 12.21 dB

Recovery error: ℓ_1 vs weighted ℓ_1



(k) SNR=8.4 dB



(l) SNR= 12.21 dB

Iterative thresholding algorithms

- ℓ_1 has high computational complexity.

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

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

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

Iterative thresholding algorithms

- ℓ_1 has high computational complexity.
- **Iterative thresholding** algorithms is an appealing alternative

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

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

Approximate message passing (AMP)

- Donoho, Maleki and Montanari proposed approximate message passing (AMP).
- AMP has low complexity of iterative thresholding algorithms.

Approximate message passing (AMP)

- Donoho, Maleki and Montanari proposed approximate message passing (AMP).
- AMP has low complexity of iterative thresholding algorithms.
- In certain cases AMP has the superior recovery conditions of BP.

Approximate message passing (AMP)

- Donoho, Maleki and Montanari proposed approximate message passing (AMP).
- AMP has low complexity of iterative thresholding algorithms.
- In certain cases AMP has the superior recovery conditions of BP.

- The AMP starts from an initial x^0 and iteratively goes by

$$\begin{aligned} x^{t+1} &= \eta(x^t + A^* z^t; \tau^t), \\ z^t &= y - Ax^t + \delta^{-1} z^{t-1} \left(\frac{\#\{|x^{t-1} + A^* z^{t-1}| > \tau^{t-1}\}}{N} \right). \end{aligned} \tag{2}$$

Approximate message passing (AMP)

- Donoho, Maleki and Montanari proposed approximate message passing (AMP).
- AMP has low complexity of iterative thresholding algorithms.
- In certain cases AMP has the superior recovery conditions of BP.

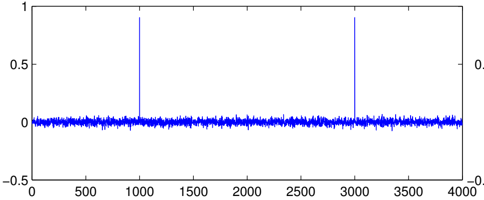
- The AMP starts from an initial x^0 and iteratively goes by

$$\begin{aligned} x^{t+1} &= \eta(x^t + A^* z^t; \tau^t), \\ z^t &= y - Ax^t + \delta^{-1} z^{t-1} \left(\frac{\#\{|x^{t-1} + A^* z^{t-1}| > \tau^{t-1}\}}{N} \right). \end{aligned} \tag{2}$$

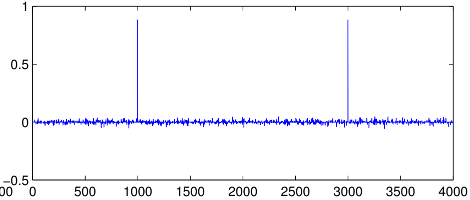
Simple example with a 2-sparse signal, iteration $t = 1$

Soft thresholding:

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

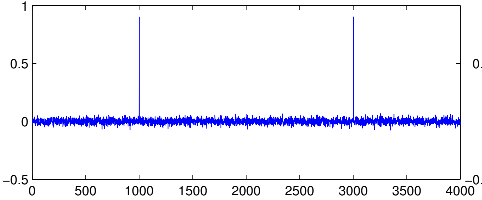


$\eta(x^t + A^* z^t; \tau^t)$

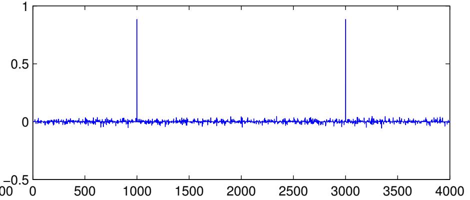


AMP:

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



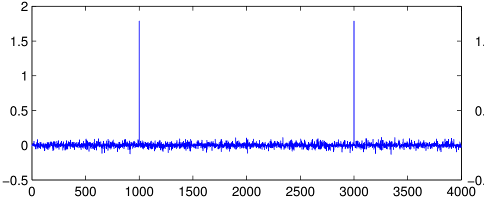
$\eta(x^t + A^*(z^t + \alpha^t z^{t-1}); \tau^t)$



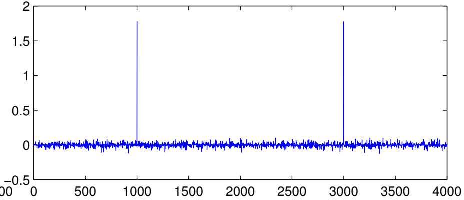
Iteration $t = 2$

Soft thresholding:

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

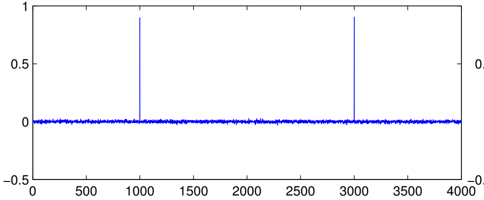


$\eta(x^t + A^* z^t; \tau^t)$

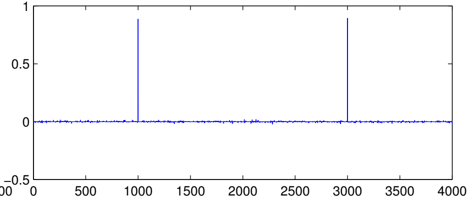


AMP:

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



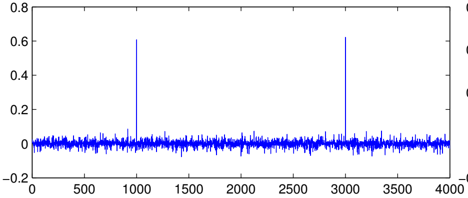
$\eta(x^t + A^*(z^t + \alpha^t z^{t-1}); \tau^t)$



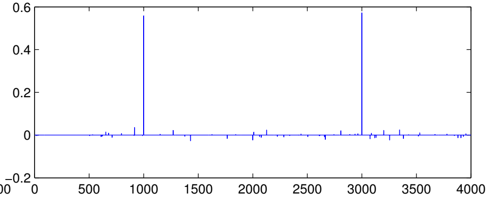
Iteration $t = 3$

Soft thresholding:

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

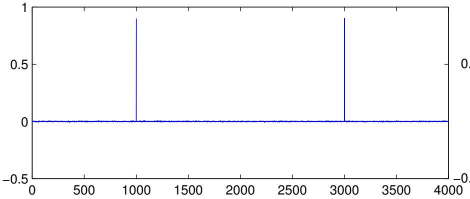


$\eta(x^t + A^* z^t; \tau^t)$

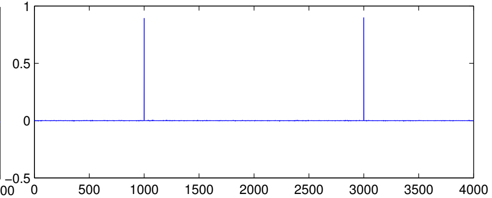


AMP:

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



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



Comparison of AMP, OMP, & SPGL1

An idealized small example with $A \in \mathbb{R}^{800 \times 5000}$. k is the sparsity level.

Average recovery times of AMP, OMP and SPGL1			
recovery method	AMP	OMP	SPGL1
$k = 200$	0.75 s	1.36 s	4.16 s
$k = 250$	0.82 s	1.43 s	7.79 s
$k = 300$	0.94 s	1.62 s	8.56 s

Weighted AMP (WAMP)

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

The weighted AMP algorithm just changes the thresholds accordingly:

$$\begin{aligned} x^{t+1} &= \eta(x^t + A^* z^t; \hat{\tau}^t \mathrm{w}), \\ z^t &= y - Ax^t + \delta^{-1} z^{t-1} \left(\frac{\#\{|x^{t-1} + A^* z^{t-1}| > \hat{\tau}^t \mathrm{w}\}}{N} \right). \end{aligned} \tag{3}$$

Weighted AMP (WAMP)

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

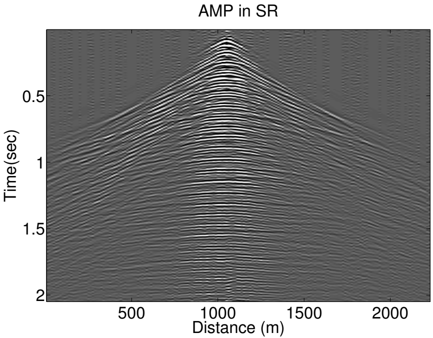
The weighted AMP algorithm just changes the thresholds accordingly:

$$\begin{aligned} x^{t+1} &= \eta(x^t + A^* z^t; \hat{\tau}^t \mathrm{w}), \\ z^t &= y - Ax^t + \delta^{-1} z^{t-1} \left(\frac{\#\{|x^{t-1} + A^* z^{t-1}| > \hat{\tau}^t \mathrm{w}\}}{N} \right). \end{aligned} \tag{3}$$

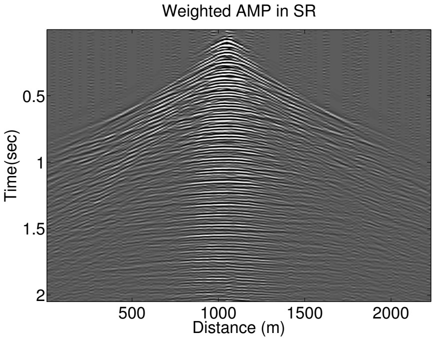
AMP and WAMP for seismic trace interpolation

- 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.

AMP and WAMP results

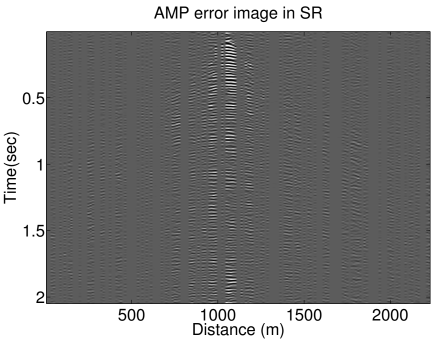


(m) SNR=8.46 dB

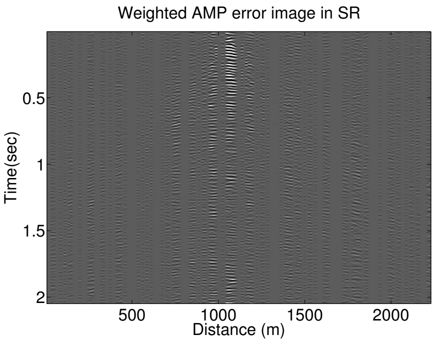


(n) SNR= 10.3 dB

Recovery error: AMP vs weighted AMP

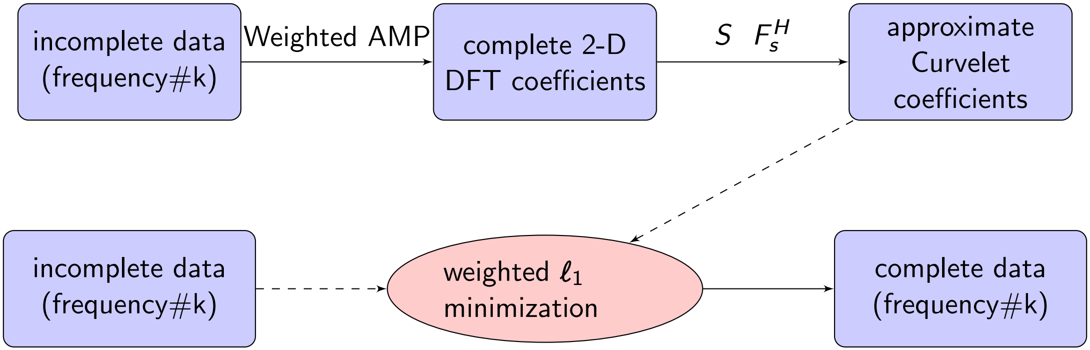


(o) SNR=8.46 dB

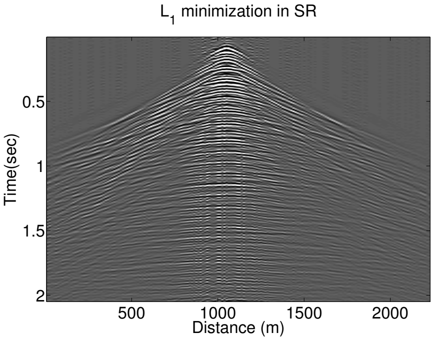


(p) SNR= 10.3 dB

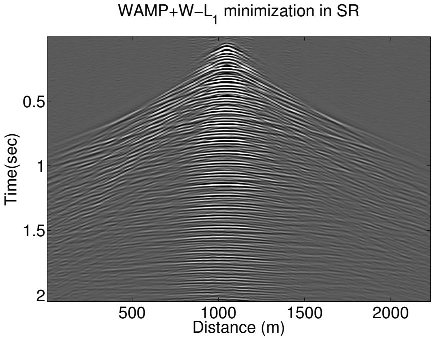
Flowchart of the 2-stage algorithm WAMP+weighted ℓ_1



ℓ_1 vs 2-stage WAMP+weighted ℓ_1

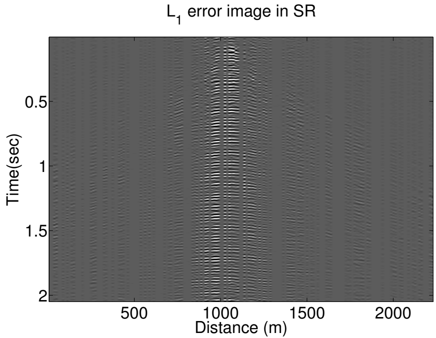


(q) SNR=8.4 dB

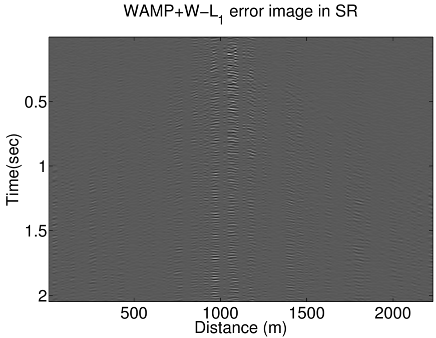


(r) SNR= 16.29 dB

Results of the 2-stage algorithm

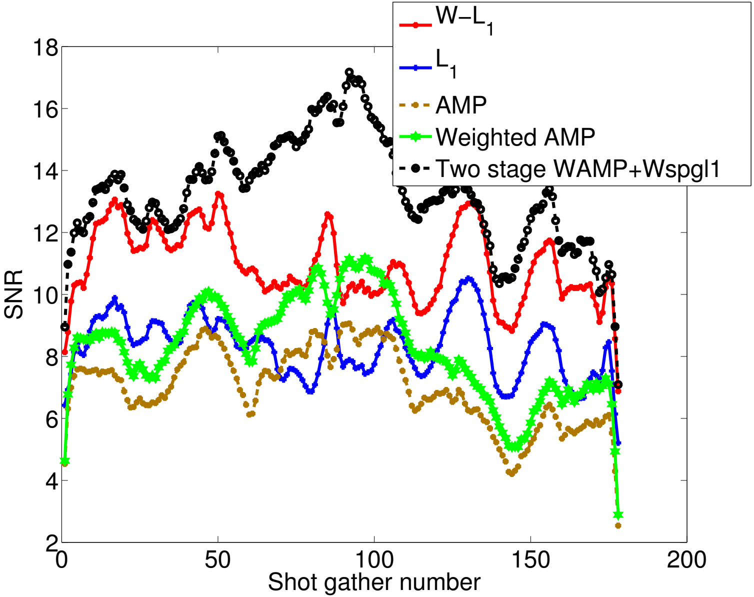


(s) SNR=8.4 dB



(t) SNR= 16.29 dB

Comparison of recovery results



Average computational time for each frequency slice

Comparison of different methods				
recovery method	# of DFT transforms	# of curvelet transforms	average runtime	SNR
weighted ℓ_1	0	1125	924s	12.21
AMP	1000	0	14s	8.46
WAMP	1000	0	14s	10.3
2-stage WAMP+wℓ_1	1000	440	420s	16.29

Conclusion

- We use the fast AMP algorithm and fast DFT transforms to obtain a reasonably good estimate of the solution.
- We use this estimate to get an initial point for the ℓ_1 reconstruction in the curvelet domain.

Conclusion

- We use the fast AMP algorithm and fast DFT transforms to obtain a reasonably good estimate of the solution.
- We use this estimate to get an initial point for the ℓ_1 reconstruction in the curvelet domain.
- We increase both speed and accuracy of ℓ_1 reconstruction in the curvelet domain once we use this well-chosen initial point.

Conclusion

- We use the fast AMP algorithm and fast DFT transforms to obtain a reasonably good estimate of the solution.
- We use this estimate to get an initial point for the ℓ_1 reconstruction in the curvelet domain.
- We increase both speed and accuracy of ℓ_1 reconstruction in the curvelet domain once we use this well-chosen initial point.

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