Data examples of logarithm Fourier domain bidirectional deconvolution

Qiang Fu and Yi Shen and Jon Claerbout

ABSTRACT

Time domain bidirectional deconvolution methods show a very promising perspective on blind deconvolving a signal that contains a mixed-phase wavelet, such as seismic data, to overcome minimum phase assumption. Howevery usually one time domain method suffers with low convergence speed (slalom method) and the other one is sensitive of initial point or preconditioner (symmetric method). Claerbout perposed a logarithm Fourier domain method to perform bidirectional deconvolution. In this paper, we test the new logarithm Fourier domain method on both synthetic data and field data. The results demonstrate that the new method do a better job than the previous one on both the quality and the stability.

INTRODUCTION

Usually, a seismic data trace d can be decomposed into a convolution of a wavelet w with a reflectivity series r as $d = r * w$. Traditionally, seismic blind deconvolution has two assumptions, whiteness assumption and minimum phase assumption. Whiteness assumption is we assumes the reflectivity series r is white spectrum. And minimum phase assumption assumes we have minimum phase wavelet w in our problem. Recently, some new methods proposed to replace or correct these two assumption in seismic blind deconvolution.

In Zhang and Claerbout (2010a), the authors proposed to use hyperbolic penalty function that introduced in Claerbout (2009) instead of conventional L2 norm penalty function to solve blind deconvolution problem. By using this method a sparseness assumption takes place of traditional whiteness assumption in deconvolution problem. And then Zhang and Claerbout (2010b) proposed a new method called "bidirectional deconvolution" in order to overcome the minimum-phase assumption. If the wavelet w is a mixed-phase wavelet, it can be decomposed into a convolution of two parts: $w = w_a * w_b$, where w_a is a minimum-phase wavelet and w_b is a maximum-phase wavelet. So we use two deconvolution filters a and b to deal with the two wavelets w_a and w_b. In Zhang and Claerbout (2010b), the authors solved the two deconvolution filters a and b alternately, so we call this method slalom method. Shen et al. (2011) proposed another method to solve the same problem. The authors use a linearized approximation to solve the two deconvolution filters simultaneously. We call this

method symmetric method. And Fu et al. (2011) proposed a way to find out the initial solution to relief the local minima problem caused by highly non-linear of blind deconvolution problem. **?** discussed a very important aspect of any inversion problem, the preconditioning and how the preconditioning helps the bi-directional deconvolution problem.

All forementioned methods solve the problem in time domain, on the contrary, Claerbout et al. (2011) proposed a method solve the problem in Fourier domain. We will see in later section this new method has fast speed and not so sensitive with the start point or preconditioner.

METHODOLOGY

Claerbout et al. (2011) has shown the complete derivation of the method and we do not want to repeat it here. We just describe the major steps of this method. As any iterative method, we have two issue to solve in one iteration, first we need to get the update direction and then we need to know the step length of the update. Now let us see how we can solve these two issue in the logarithm Fourier domain method.

As we discussed in the previous section, we can decomposed the seismic arbitrary data d into three parts: the reflectivity series r, the minimum phase wavelet w_a and the maxmum phase wavelet w_b.

$$d = r * (w_a * w_b). \tag{1}$$

The deconvolution filters a and a are the unknowns we want to solve. And they should be the inverse of wavelets w_a and w_b.

$$\begin{cases} w_a * a = \delta(n) \\ w_b * b = \delta(n) \end{cases}, \tag{2}$$

From equation 7, we know a is minimum phase and b is maxmum phase. If we know the deconvolution filters a and a, we can get reflectivity series r very easily.

$$r = d * a * b, \tag{3}$$

Now we just transform our problem into Fourier domain, we use the capital letter to denote variable in Fourier domain, we have

$$R = D\ A\ B, \tag{4}$$

We use U to denote logarithm of the product of A and B,

$$U = \log(A\ B) \tag{5}$$

Our problem becomes

$$R = D\ e^U \tag{6}$$

U has become our new unknown in bi-directional deconvolution. We want to get update U in each iteration. After some derivation (Claerbout et al., 2011), we get in time domain

$$\begin{cases} \Delta u = r^{\circledast}\mathrm{Hyp}'(r) \\ \Delta r = r * \Delta u \end{cases}, \tag{7}$$

where $^{\circledast}$ means cross correlation and $\mathrm{Hyp}(r_i) = \sqrt{r_i^2 + R_0^2} - R_0$ is the hyperbolic penalty function.

By Newton method (using the only first 2 terms of the Taylor expansion), we can calculate the step length α

$$\alpha = \frac{\sum_i \mathrm{Hyp}'(r_i)\Delta r_i}{\sum_i \mathrm{Hyp}''(r_i)\Delta r_i^2} \tag{8}$$

Because we use the Newton method, this step length α calculated above is not the final value. We need another iteration (iteration in iteration or second order iteration) to get the final step length α of each iteration.

$\alpha_j = 0$
Iterate(j)
$\alpha_j = \dfrac{\sum_i \mathrm{Hyp}'(r_i)\Delta r_i}{\sum_i \mathrm{Hyp}''(r_i)\Delta r_i^2}$
$\alpha_{final} = \alpha_{final} + \alpha_j$
$r = r + \alpha_j \Delta r$
$u = u + \alpha_j \Delta u$

So we know the update direction (both for the unknown u and for the residual r) and the step length α of the update, we have everything for an iteration now. We can keep updating the unknown until it is converged or we are satisfied.

Trial and error on step length α

When we use the way above to calculate the step length α for a huge volume data set (say a whole pre-stack survey line), we may get a blow up problem. This caused by the over shot problem. The Newton method requires a convex function, but for a huge volume data set that may be not true. There is a small trick for solving this problem. If we use the way describe above to get step length α, we usually get over shot problem. We use trial and error to avoid a too large step length α. If the hyperbolic penalty function on $r = r + \alpha\Delta r$ is greater than it on r, the step length α

is too large and we get over shot problem, we just reduce the step length α by half.

$\alpha_j = 0$
Iterate(j)
$\alpha_j = \frac{\sum_i \text{Hyp}'(r_i)\Delta r_i}{\sum_i \text{Hyp}''(r_i)\Delta r_i^2}$
Iterate
If Hyp$(r + \alpha_j \Delta r) <=$ Hyp(r) Then Break
$\alpha_j = \alpha_j / 2$
$\alpha_{final} = \alpha_{final} + \alpha_j$
$r = r + \alpha_j \Delta r$
$u = u + \alpha_j \Delta u$

EXAMPLES

In this section we will test the logarithm Fourier domain method with three test cases. One is simple synthetic data and other two are field data. As our prior experience (**?**), the preconditioning is a critical part for the seismic blind deconvolution problem. And here we use Burg PEF as the preconditioner for all tests consistently. And we will use one same filter for all traces unless otherwise specified.

Synthetic 1D example

First of all we want to demonstrate the advantage of the logarithm Fourier domain method over the time domain method on a very simple synthetic data example. Here we just have a synthetic Ricker wavelet as the input data. This Ricker wavelet is generated by approximate approach that apply a second order derivative on a binomial coefficients. The detail about to generated approximate Ricker wavelet discussed in Fu et al. (2011). Figure 1(a) shows a synthetic Ricker wavelet and 1(b) shows the Ricker wavelet after a Burg PEF perconditioning. We use the symmetric method(Shen et al., 2011) to perform time domain bi-directional deconvolution. Figure 1(c) and 1(d) show the comparison of bi-directional deconvolution result of the two different method.

From this very simple 1D synthetic example, we use 0.1 as the threshold of the hyperbolic penalty function for logarithm Fourier domain method and use a quatil 95% of data residual as the threshold for time domain method. And we can find for the logarithm Fourier domain method after about 50 iterations we turn the Ricker wavelet into a spike output. But for the time domain symmetric method, even after 30000 iterations, we can get a major spike and followed by a minor spike plus a few jitters on the begining of the trace.

Another important observation from this synthetic test case is the output location of the bi-directional deconvolution. If we look at the wiggles, we will find the major peak of the both deconvolution results figure 1(c) and 1(d) (which is at the time sample 104) are not the same as the location of the major peak of the input data figure 1(a) (which is at the time sample 100). But they are the same as the major peak location of the preconditioning result figure 1(b). That inspire us that the output spike location of the deconvolution is determined by the preconditioner and we can change the preconditioner to change the output location of the spike of the deconvolution result to the location desired. In our another paper (**?**) we will discuss this interesting topic more specifically.

Field data common offset gather example

Now we will test the logarithm Fourier domain bidirectional deconvolution with the same 2D marine common offset gather data which is used to test time domain bidirectional deconvolution. This 2D marine common offset gather is very popular in the hyperbolic penalty function based blind deconvolution papers. In Zhang and Claerbout (2010a), Zhang and Claerbout (2010b), Fu et al. (2011), Shen et al. (2011) and **?** the authors tested their methods and theories with this data set as a field data example. Hence this is a good choice that we can compare the result of the new method with the result of previous ones.

Figure 2 shows the 2D marine common offset gather. Figure 3 shows the common offset section after Burg PEF preconditioner. And Figure 3 and 3 compare the results of 2 different method of bi-directional deconvolution.

Figure 6(a) and 6(b) show the comparison of estimated wavelet from two different bi-directional deconvolution methods. The estimation wavelet is in fact the inverse deconvolution filter. We get the inverse filter by doing 1 over the frequency spectrum of the filter in Fourier domain, so the wavelet waveform in fact is periodic. That means the jitter we find at the end of the wavelet is the anti-causal part of the filter.

From the figure 3 and 3, we can see the deconvolution results both are good. But we think the logarithm Fourier domain method (figure 3) is a little better. Within the salt body (in the vicinity of 2.4s to 2.6s), the Fourier domain result looks cleaner than the other one. We do not expect there is any feature within the salt body, and all events we see in this area in the raw data are the air gun bubbles. so that indicate the deconvolution in Fourier domain just handle the air gun bubbles a little better than time domain method. But except of these differences, the rest of the results are quite similar. We can also find the estimated wavelets (Figure 6(a) and 6(b)) are just very similar excepts the tail part. That is consistant with the deconvolution result.

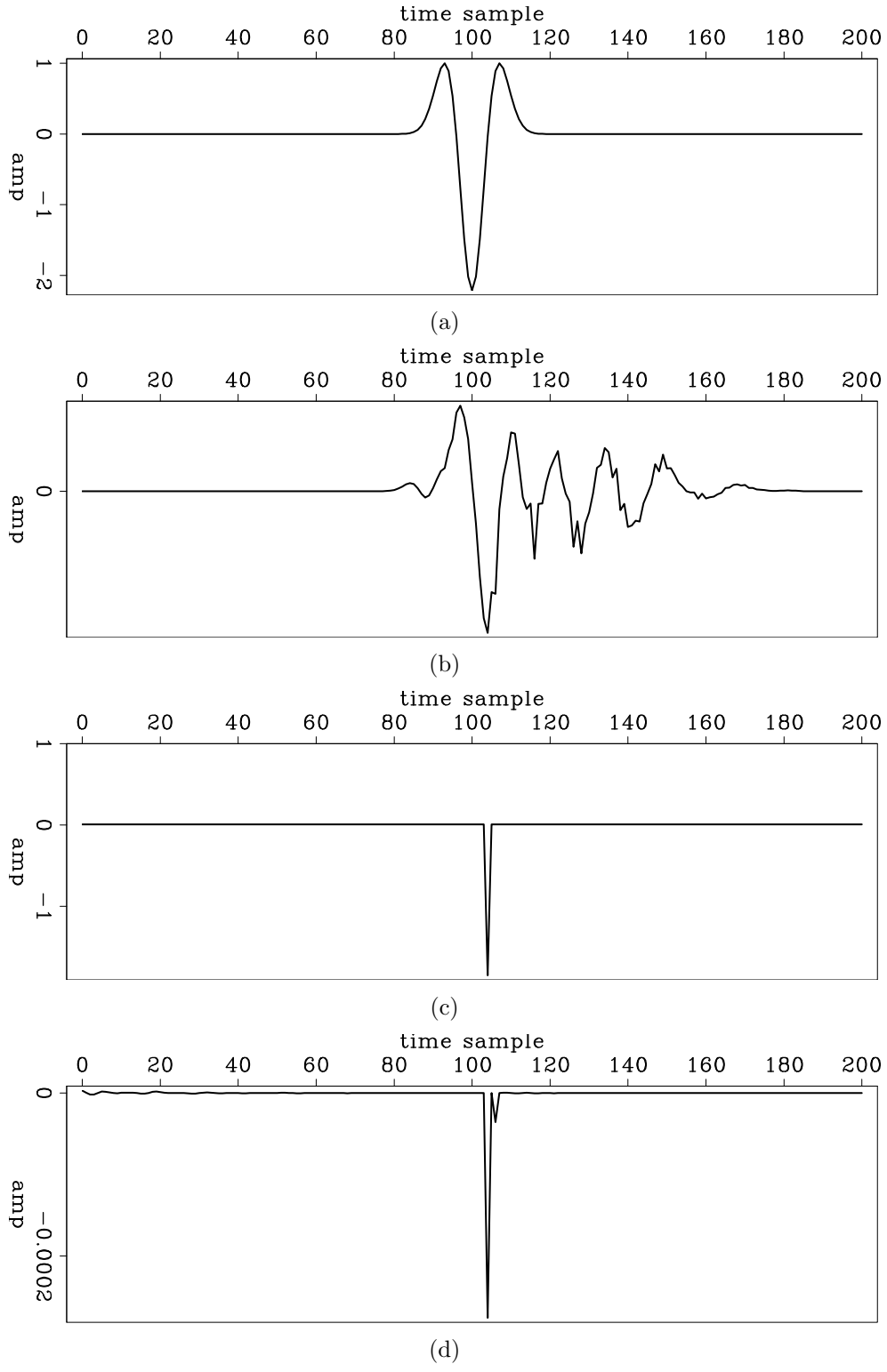

Figure 1: (a)The synthetic Ricker wavelet; (b)The Ricker wavelet after Burg PEF preconditioning; (c)The bi-directional deconvolution result of logarithm Fourier domain method after 50 iterations; (d)The bi-directional deconvolution result of time domain symmetric method after 30000 iterations. [**ER**]

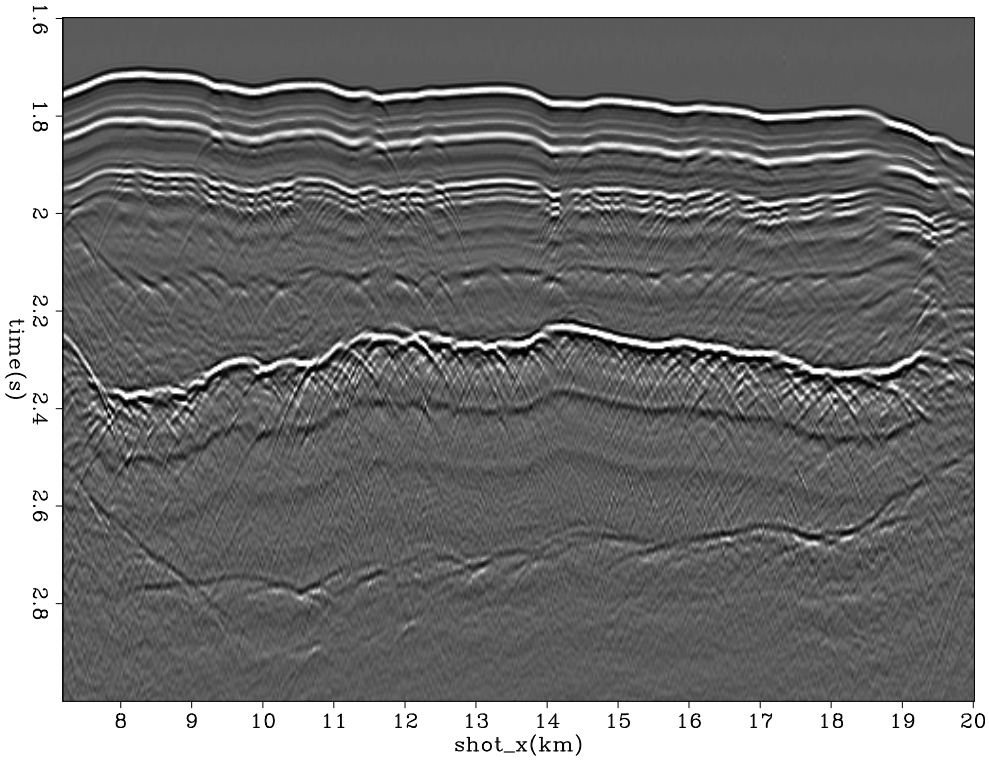

Figure 2: A common offset section of a marine survey field data. [**ER**]

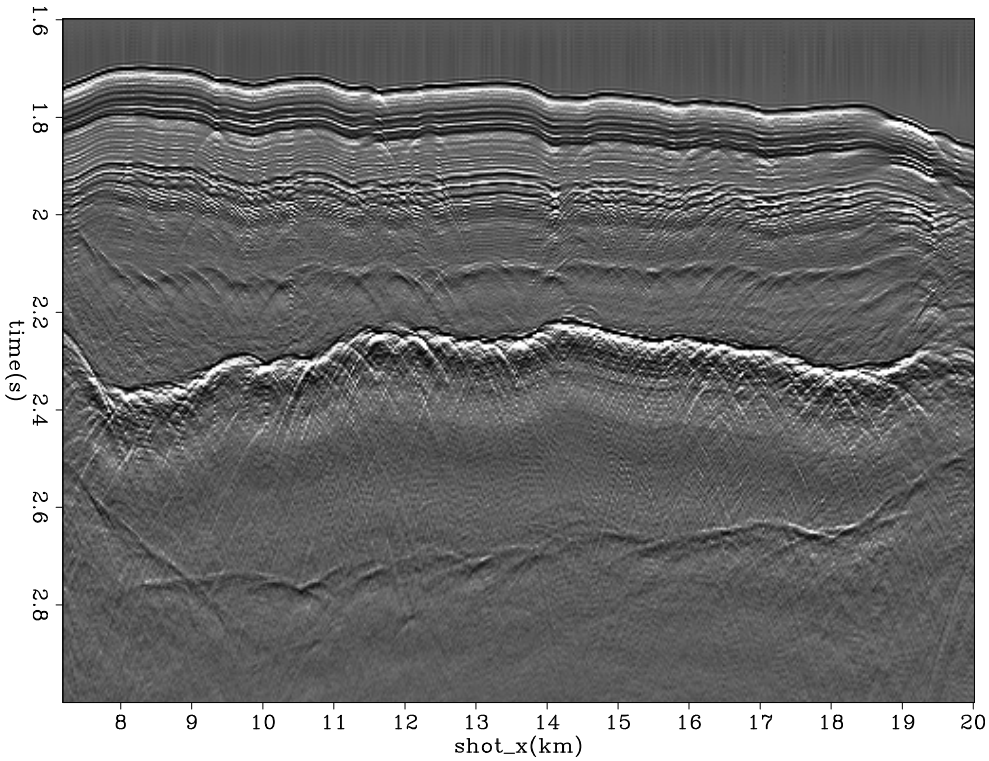

Figure 3: The common offset section after Burg PEF preconditioner. [**ER**]

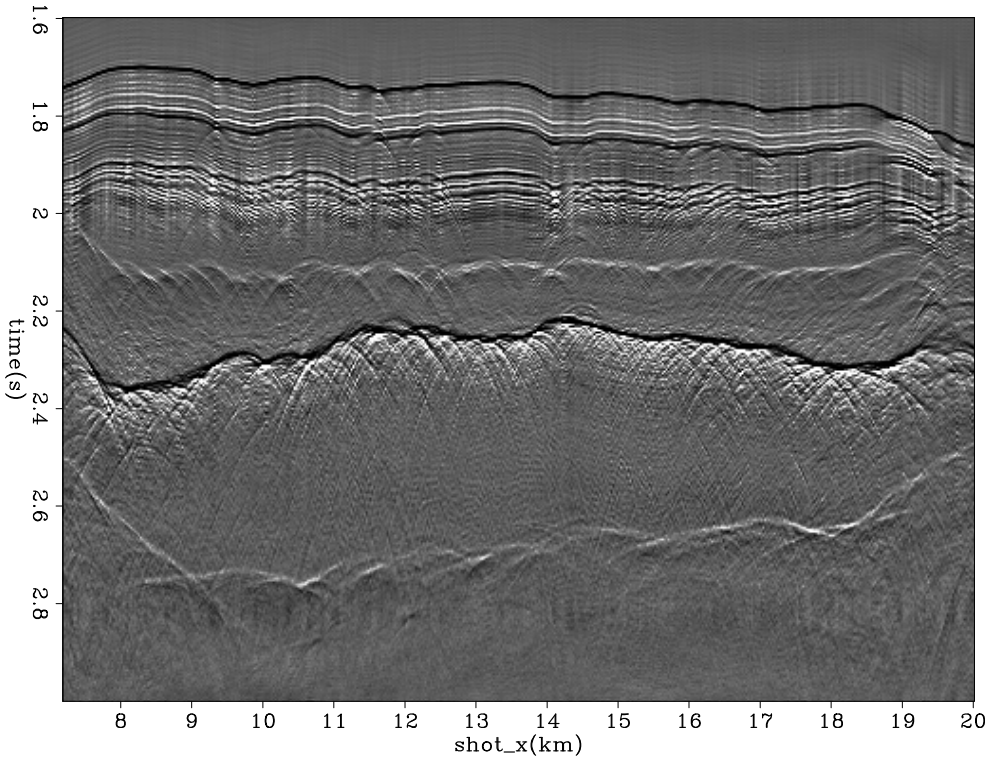

Figure 4: Logarithm Fourier domain bidirectional deconvolution result. [**ER**]

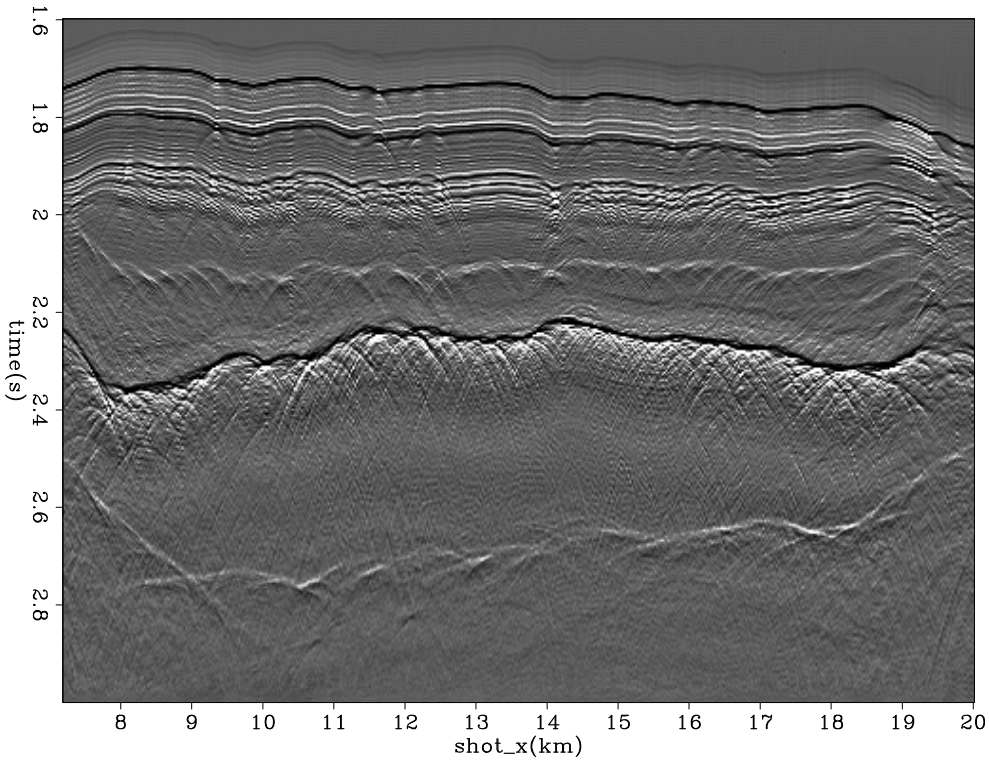

Figure 5: Time domain (symmetric) bidirectional deconvolution result. [**ER**]

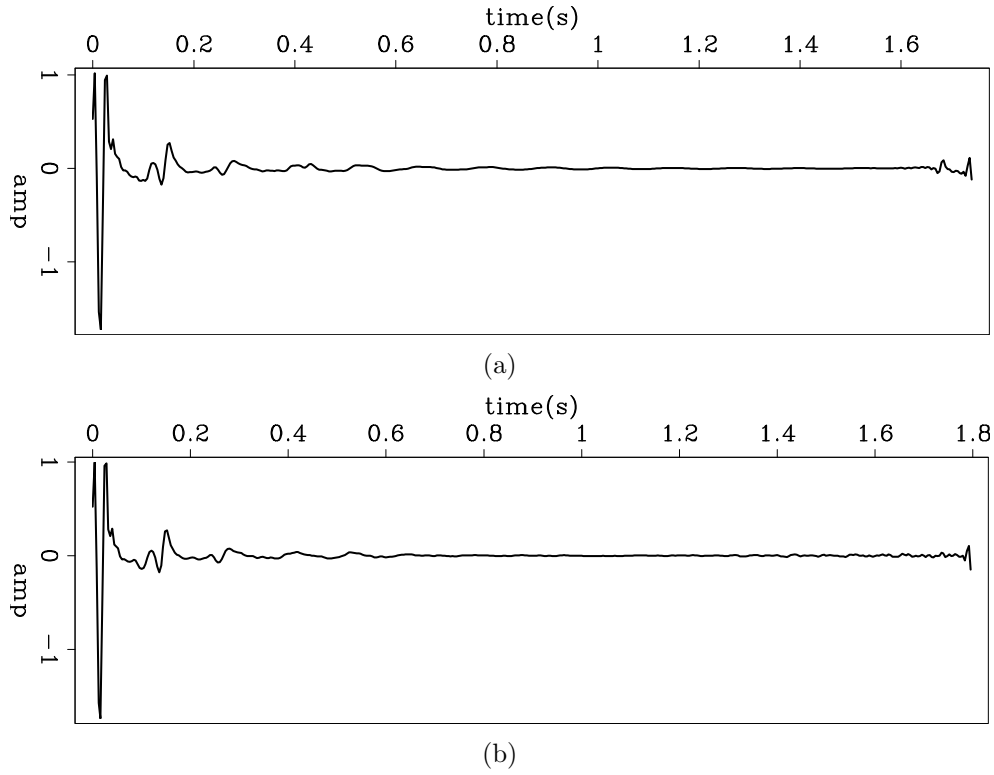

Figure 6: Estimated wavelet from (a)Logarithm Fourier domain and (b)Time domain (symmetric) bidirectional deconvolution. The estimation wavelets are the inverse deconvolution filters. It is calculated by doing 1 over the frequency spectrum of the filter in Fourier domain. So the wavelet waveform is periodic. That means the jitter we find at the end of the wavelet is the anti-causal part of the filter.

Field data pre-stack shot gathers example

The previous common offset data set is extracted from a pre stack survey line. Figure 7 shows the pre-stack shot gathers for the whole survey line. In order to see clearly the deep feature, we use a gain function of $t^{2.5}$ on the figure 7. This gain function is only applied here for display purpose and will never be used in latter deconvolution procedure. Figure 8 shows two shot gather in a marine survey. There is no any gain applied on these two shot gathers.

We can find out the previous common offset gather not only windowed the data in space but also windowed the data with a time range. We want to get the data within the same time window of the previous common offset gather. But now we are working on the prestack gathers, and simply cut a horizontal window within 1.6s to 3s will lose the far offset part of the deep event due to the moveout. We do not preform a NMO to correct the moveout because we do not want the stretch caused by NMO to damage the shot waveform. Instead, we just perform the LMO which shift the each trace with a constant time shift. We use a major event in the desire time window, which is the reflection from the top of the salt body, as a reference to calculate the shift time. Figure 9 shows the time shift function to flatten the gather and two gathers after flattening.

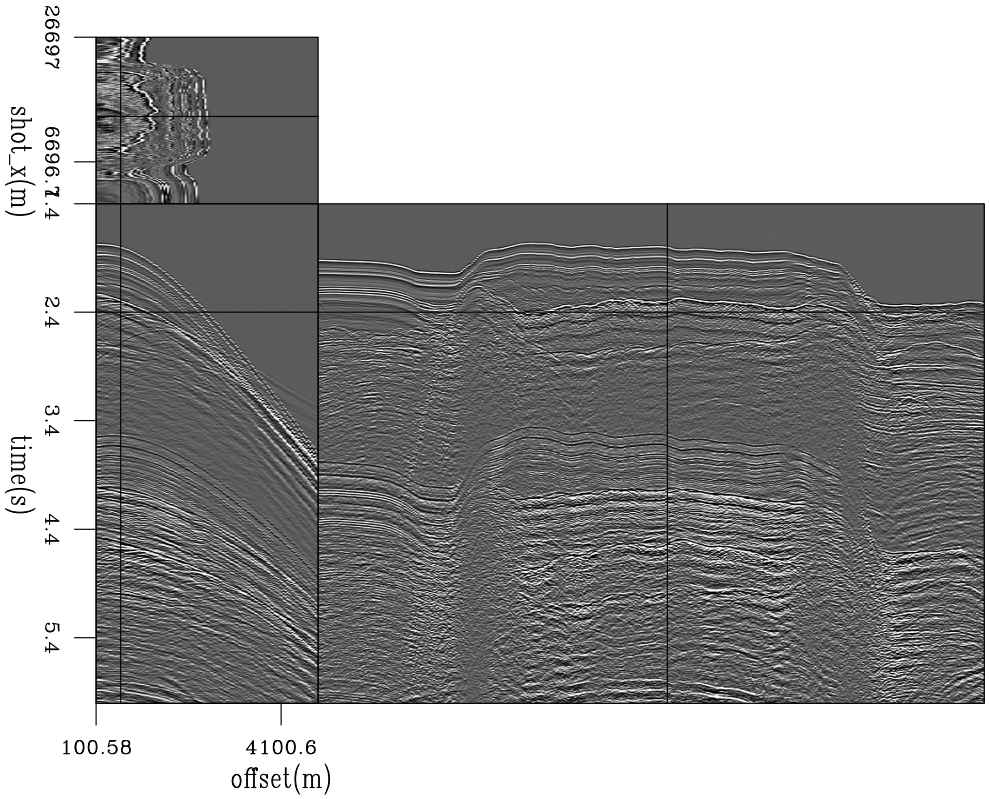

Figure 7: Whole marine pre-stack survey line. [**ER**]

After we get the two windowed shot gathers, we just apply the Burg PEF as the

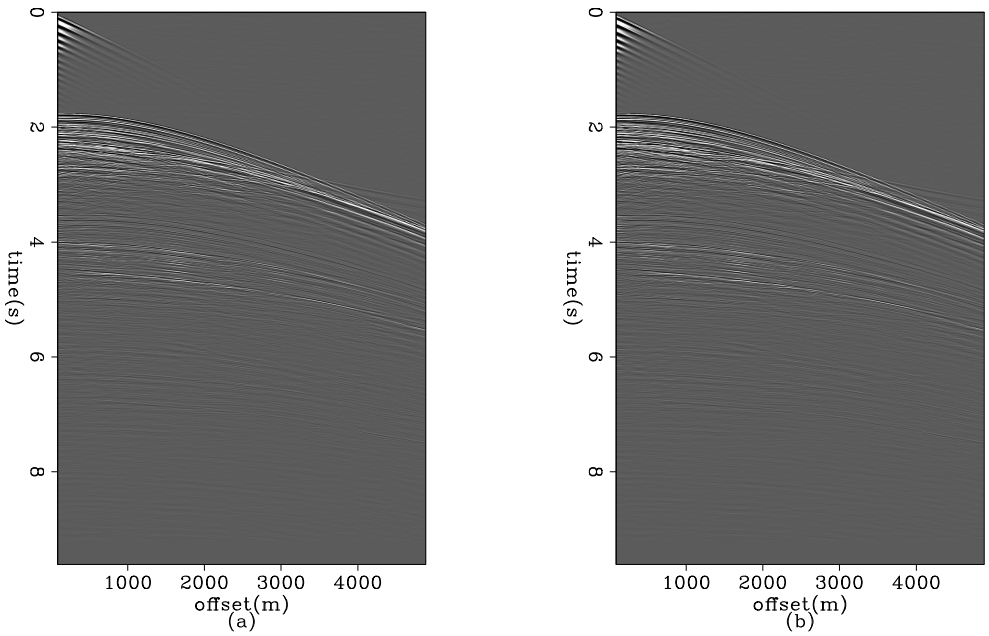

Figure 8: Two shot gathers. [**ER**]

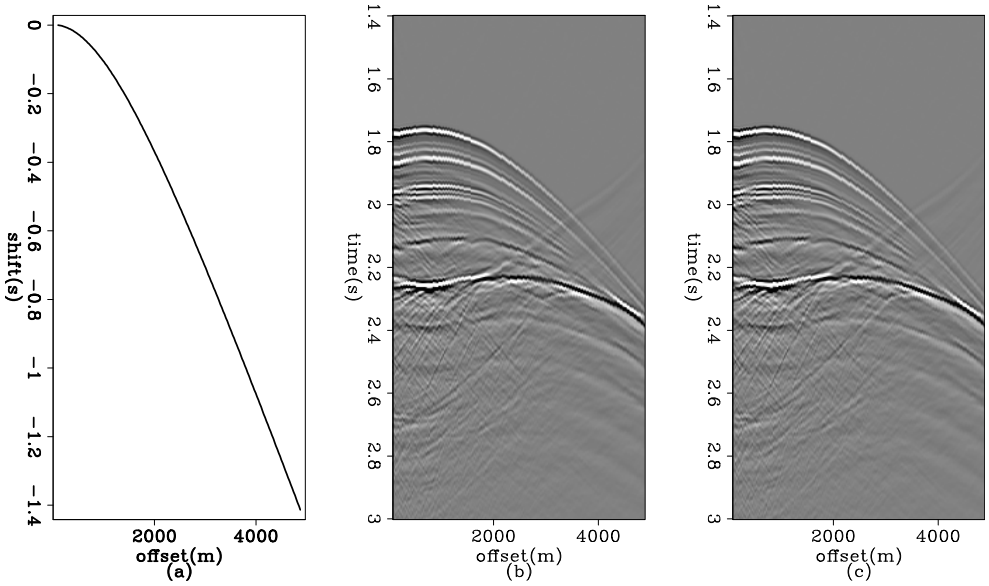

Figure 9: (a)The time shift function to flaten the gather;(b) and (c) two shifted shot gathers. [**ER**]

preconditioner and then preform the logarithm Fourier domain bi-directional deconvolution on the two windowed shot gathers. Figure 10 shows the preconditioner results and figure 11 shows the deconvolution results for the two shot gathers respectively. And the process of the two shot gathers are independently, means we use different Burg PEFs and different deconvolution filters on the two shot gathers.

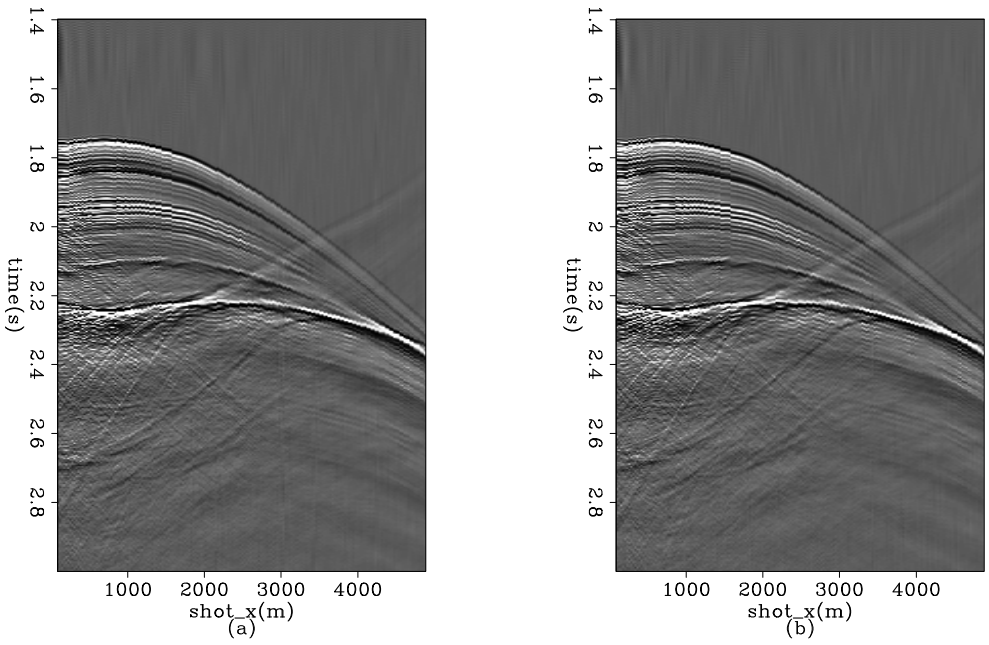

Figure 10: Two shot gathers after Burg PEF preconditioning. Different PEF filters are estimated and applied respectively. [**ER**]

We can find the major event (in the vicinity of 2.2s) which is the top of the salt body has a phase shift versus the offset increasing. In figure 11, the near offset part of this event is black and then it turn into white after about offset 1500. And the head wave is start at the same offset 1500. This is not error. This is caused by the physics that after the critical angle the reflection will has a 90° phase shift.

Figure 12(a) and 12(b) show the estimated wavelets of these two shot gathers. We can see they are quite similar. that shows the shot waveforms do not change too much in these two shots.

From the deconvolution result of the shot gathers, we find the wavelet looks very different in the near offset and far offset. We decided to test different offset range within one shot gather separately. Figure 13 shows three offset panels of different offset range. We just divided the whole offset range by three equally to get them. The head wave of the salt body top event is heppen to start at the boundary between near and middle offset panels.

Figure 14 shows three offset range panels after Burg PEF preconditioning. The different PEF filters is estimated from each panel and apply on it respectively. We combined the three panel together to display here but can still see the boundaries

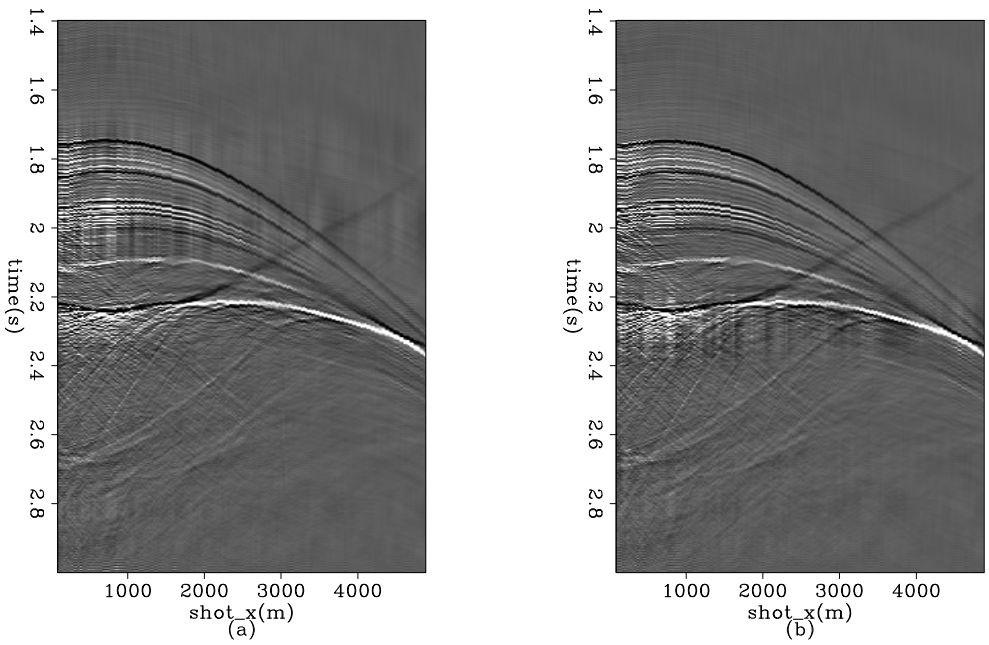

Figure 11: Two shot gathers after after logarithm Fourier domain bi-directional deconvolution. The deconvolution is applied respectively for each gather. [**ER**]

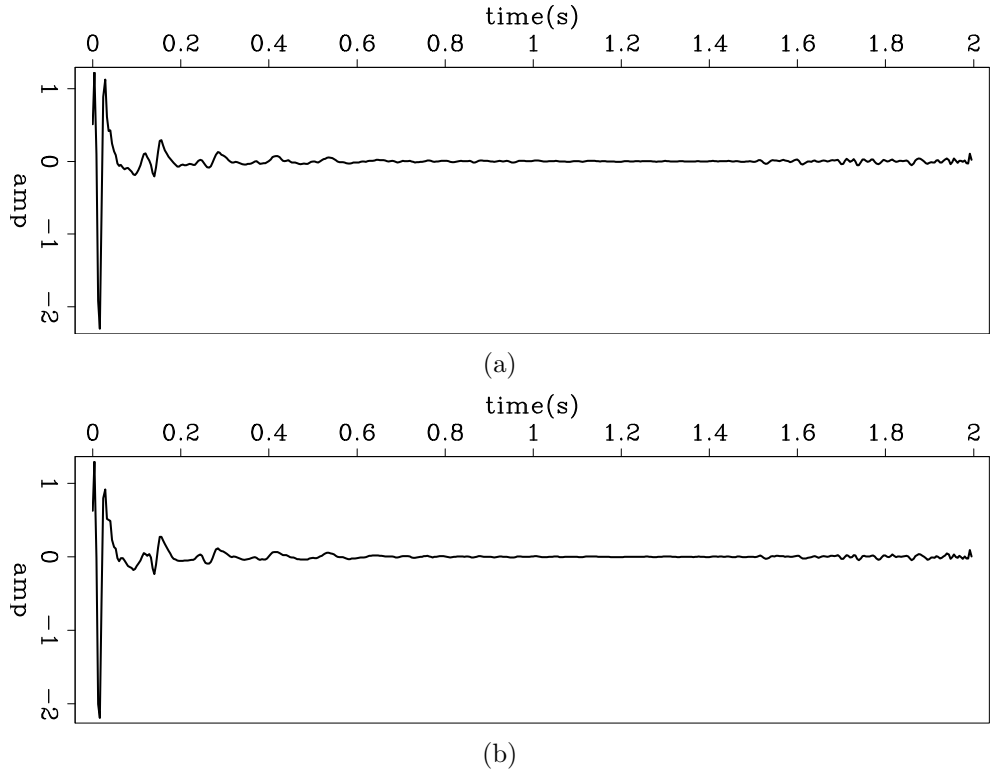

Figure 12: The estimated wavelets by logarithm Fourier domain bi-directional deconvolution.(a)Shot gather 14000 ;(b)shot gather 14028.

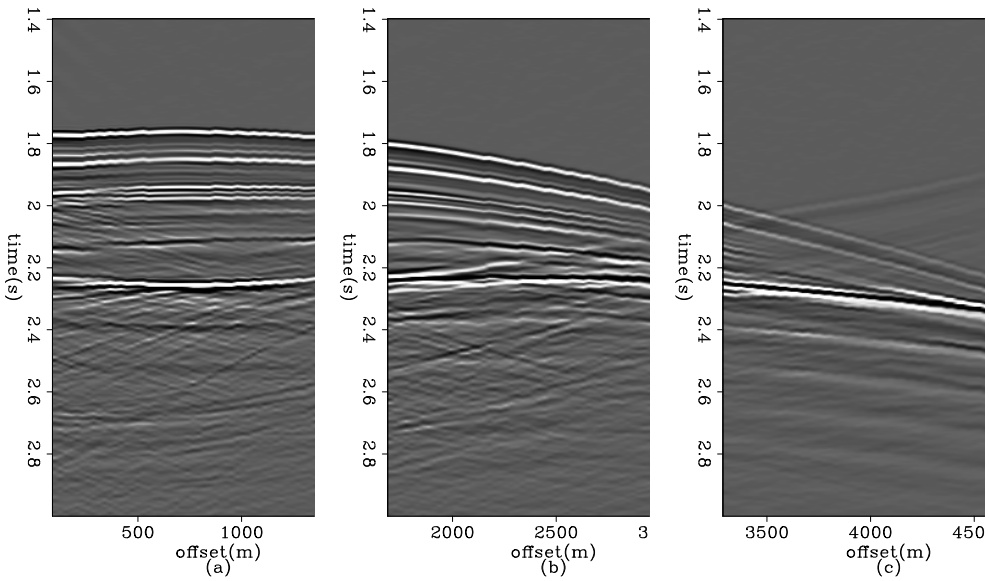

Figure 13: Three offset range panels of shot gather 14000.(a)near offset;(b)middle offset;(c)far offset. [**ER**]

among them. That means the wavelet of the three panels are different.

Figure 15 shows the deconvolution result of the three offset panels. We can easily find there seems to be a time shift of the major salt body top event from near offset panel to middle offset panel. This caused by the 90° phase shift we discussed before. And because we estimated the deconvolution filters respectively of the three offset panels, the wavelet just has a 90° phase shift and make the output deconvolution spike has a time shift.

Figure 16(a), 16(b), 16(c) and 16(d) show the different estimated wavelet from different offset range. Figure 16(a) is the wavelet from the whole shot gathers, which is just the same one as 12(a). And Figure 16(b), 16(c) and 16(d) are the wavelet estimated by near, middle and far offset respectively. We will find the wavelet from the whole gather is similar to the wavelet of the near offset. And the wavelet of the middle and far offset is quite similar each other, which are very different from the ones estimated from near offset. And the wavelet has almost 90° phase shift from near offset panel to middle offset panel. That is consistant with our observation of the raw data, as we discussed before, the major event has a 90° degree phase shift from the near offset panel to middle offset panel.

We also tried multi-shot gathers with one filter. Figure 17 and 18 show deconvolution result for 39 and 451 shot gathers. And figure 19(a), 19(b) and 19(c) show a comparison among estimated wavelet of single shot gather and multi-shot gathers. We can find basically they are very similar and from the (a) to (c), the more shot gathers we used, the less jitter we have. Those tell us the shot waveforms do not change to much from shot to shot. That is consistant with our observation in the

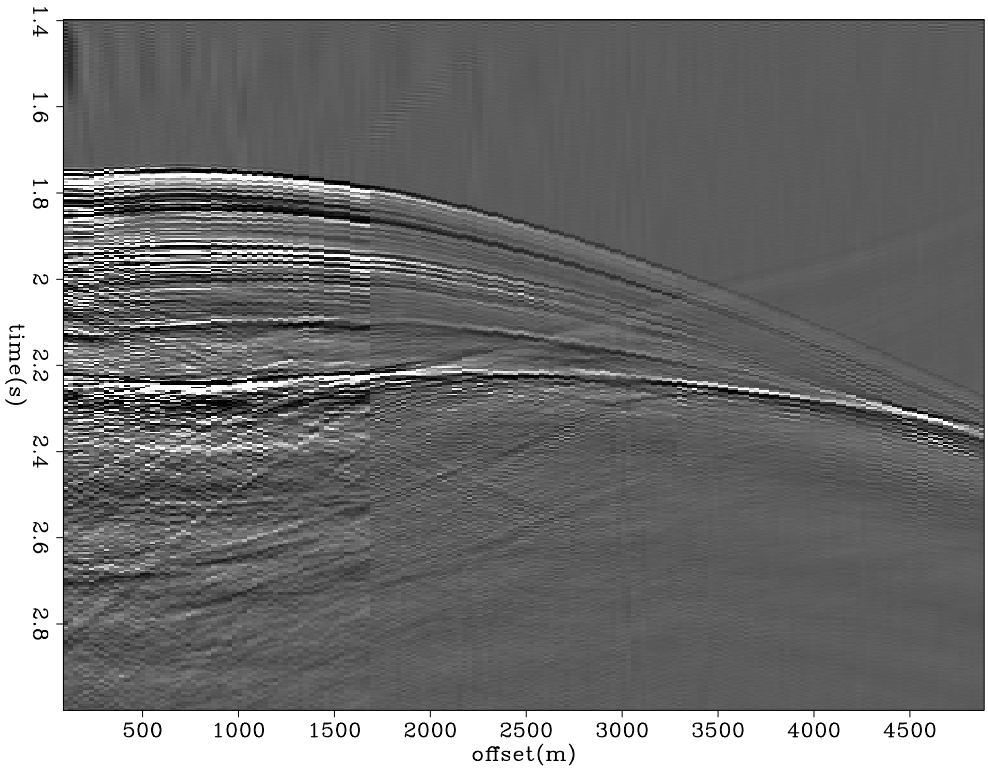

Figure 14: Three offset range panels (combined together) after Burg PEF preconditioning. Different PEF filters are estimated and applied respectively. [**ER**]

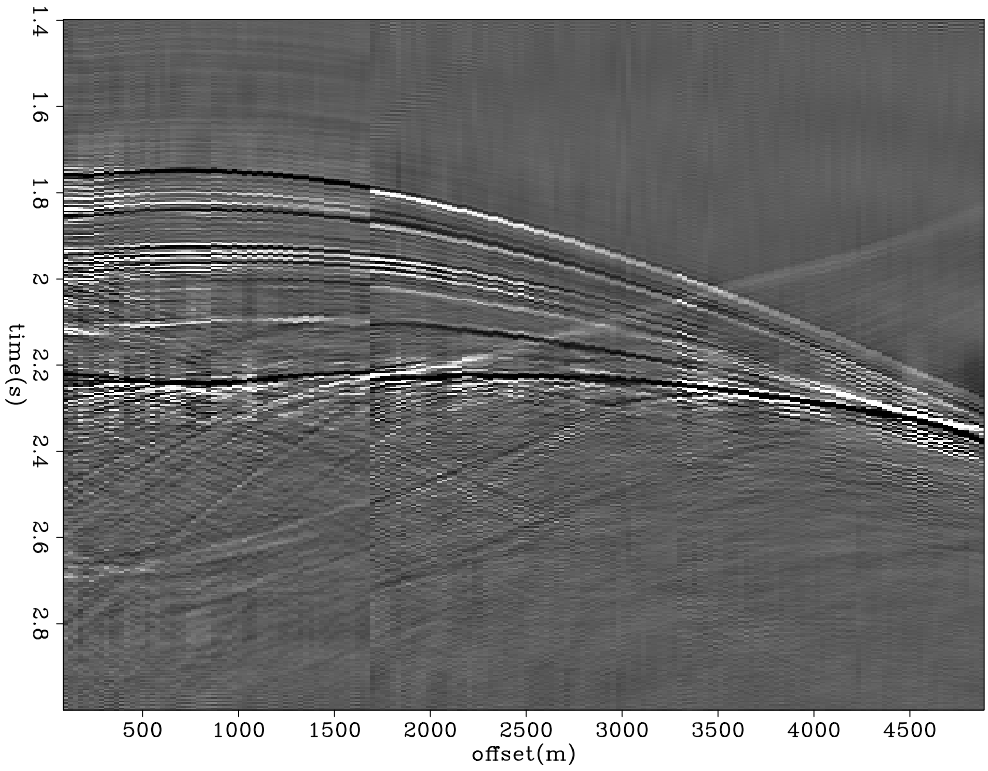

Figure 15: Three offset range panels (combined together) after logarithm Fourier domain bi-directional deconvolution. The deconvolution is applied respectively for each portion. [**ER**]

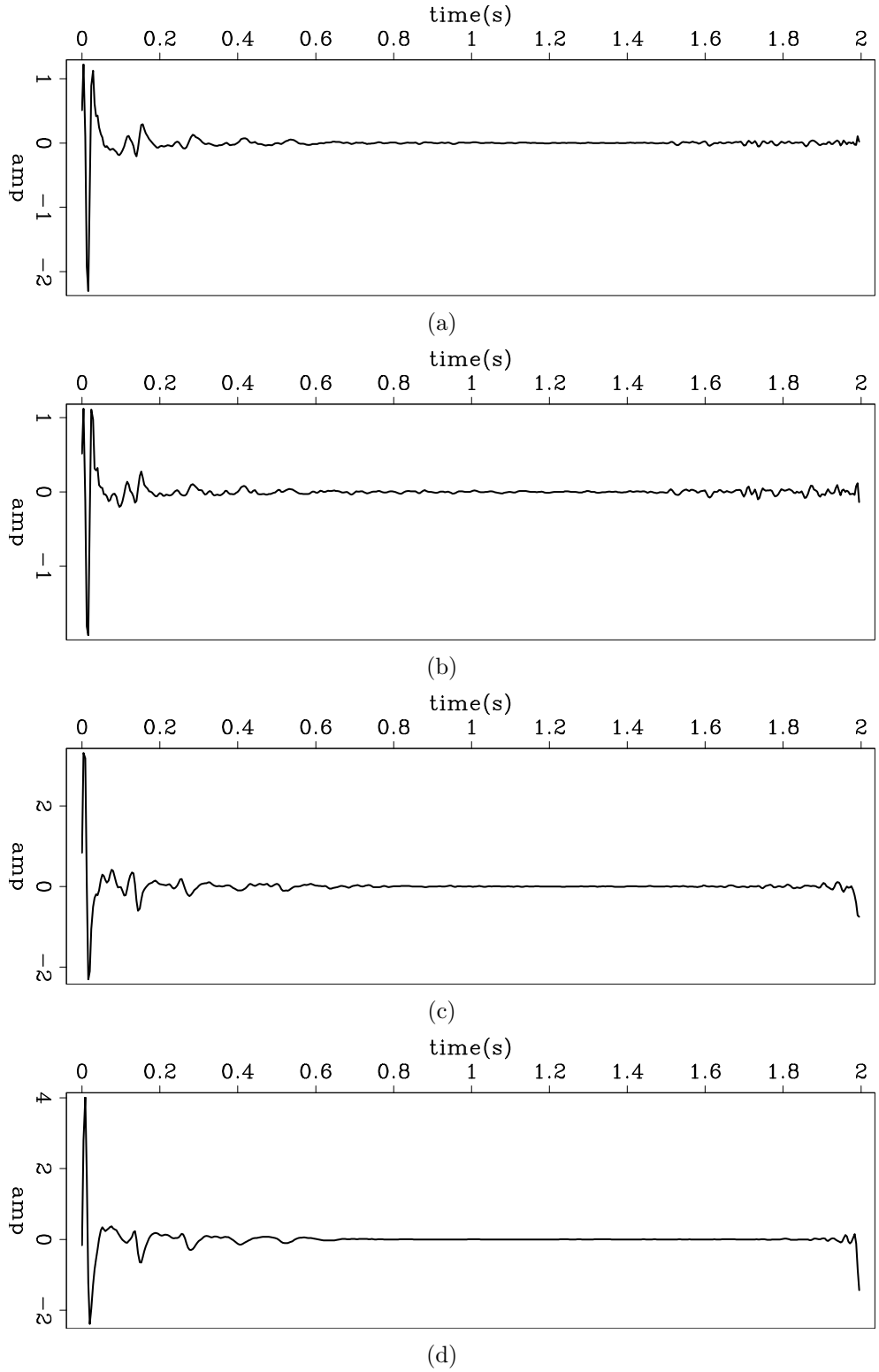

Figure 16: The wavelets estimated from (a)the whole offset range (just same as 12(a));(b)near offset;(c)middle offset;(d)far offset. [**ER**]

previously two shot gathers part.

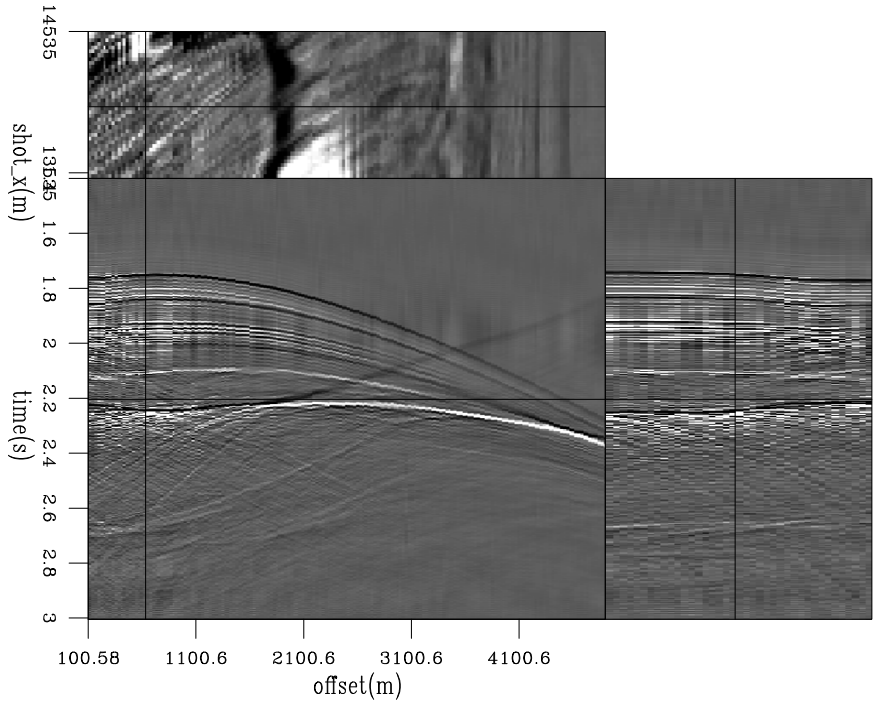

Figure 17: Logarithm Fourier domain bi-directional deconvolution result on 39 shot gathers. [**ER**]

CONCLUSIONS

we test the logarithm Fourier domain method several test cases including both synthetic data and field data. The results comfirm that this Fourier domain method has advantage than previous time domain methods on both convergence speed and result quality.

ACKNOWLEDGMENTS

We would like to thank Yang Zhang, Shuki Ronen, Robert Clapp, Dave Nichols and Antoine Guitton for helpful discussion of our research.

REFERENCES

Claerbout, J., 2009, Blocky models via the l1/l2 hybrid norm: SEP-Report, **139**, 1–10.

Claerbout, J., Q. Fu, and Y. Shen, 2011, A log spectral approach to bidirectional deconvolution: SEP-Report, **143**, 297–300.

Fu, Q., Y. Shen, and J. Claerbout, 2011, An approximation of the inverse ricker wavelet as an initial guess for bidirectional deconvolution: SEP-Report, **143**, 283–296.

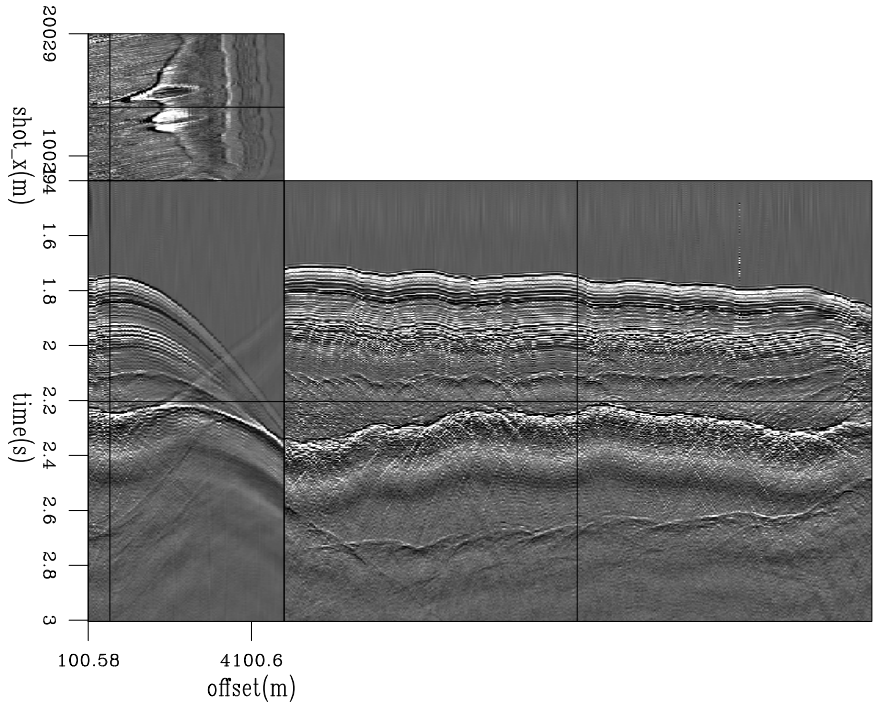

Figure 18: Logarithm Fourier domain bi-directional deconvolution result on 451 shot gathers. [**CR**]

Shen, Y., Q. Fu, and J. Claerbout, 2011, A new algorithm for bidirectional deconvolution: SEP-Report, **143**, 271–282.

Zhang, Y. and J. Claerbout, 2010a, Least-squares imaging and deconvolution using the hb norm conjugate-direction solver: SEP-Report, **140**, 129–142.

———, 2010b, A new bidirectional deconvolution method that overcomes the minimum phase assumption: SEP-Report, **142**, 93–104.

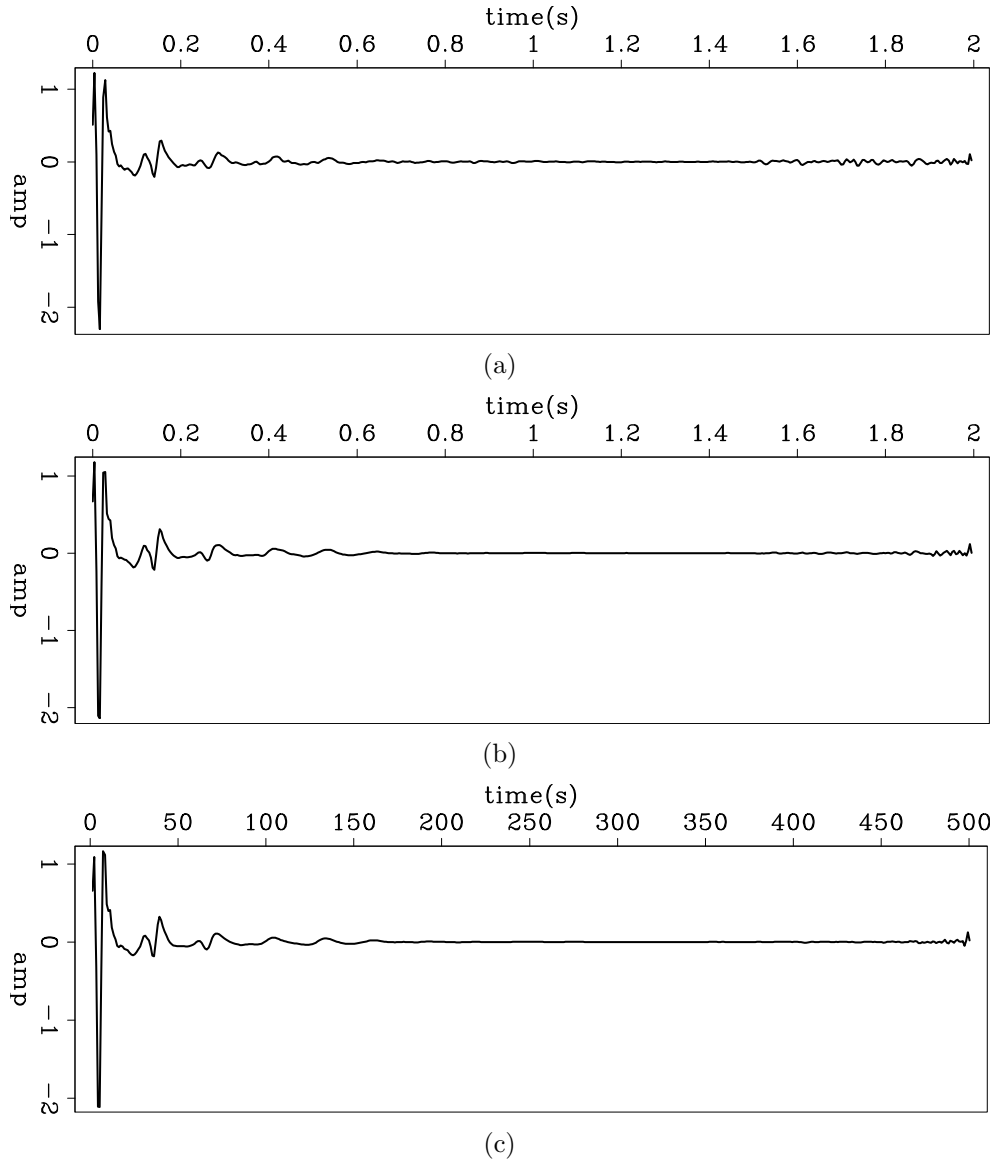

Figure 19: The wavelets estimated from (a)one shot gather (just same as 12(a));(b)39 shot gathers (from shot location 13500 14500);(c)451 shot gathers (from shot location 8000 20000) [**ER**]