

October 17, 2011

# Implementation of model space weighing for Fourier domain Bidirectional deconvolution approach

*Qiang Fu*

## ABSTRACT

Last time we see why we should use model space weight to compensate the divergence in bidirectional deconvolution. Now let see the implementation for this in Fourier domain Bidirectional deconvolution approach.

## IMPLEMENTATION OF FOURIER DOMAIN BIDIRECTIONAL DECONVOLUTION

Claerbout et al. (2011) provide the complete derivation of Fourier domain Bidirectional deconvolution approach step by step. We will see what need to be changed if we want to include a weighting function  $c(t)$  in model domain to compensate the divergence in Fourier domain Bidirectional deconvolution approach.

### original implementation without model space weighing

Before we see what need to be changed, let us review the major steps in the implementation of Fourier domain Bidirectional deconvolution. Our goal for Fourier domain Bidirectional deconvolution is to minimize the object function

$$J = \sum_t H(r_t). \quad (1)$$

And  $H(r_t)$  is the hyperbolic penalty function

$$H(r_t) = \sqrt{r_t^2 + R_{thr}^2} - R_{thr}, \quad (2)$$

where  $R_{thr}$  is a constant threshold for determining the behavior the hyperbolic penalty function between  $l1$  and  $l2$  norm.

There are three key formulas for implementation. First of all, we need to know the gradient direction of the object function. From Claerbout et al. (2011), it is

$$\Delta u_\tau = \frac{\partial J}{\partial u_\tau} = \sum_t r_{t+\tau} H'(r_t). \quad (3)$$

And then we want to know the relation between the model gradient direction  $\Delta \mathbf{u}$  and the residual update direction  $\Delta \mathbf{r}$ . It is

$$\Delta \mathbf{r} = \mathbf{r} * \Delta \mathbf{u}. \quad (4)$$

Finally we also need the step length of each updating. In Claerbout et al. (2011), we use a Newton iterative method to solve this step length  $\alpha$ . And in each iteration of this Newton search, the incremental step length  $\alpha_{inc}$  is

$$\alpha_{inc} = - \frac{\sum_t \Delta r_t H'(r_t)}{\sum_t (\Delta r_t)^2 H''(r_t)}. \quad (5)$$

## weighting or weighting plus time variant threshold?

Traditionally, if we want to add weighting on the object function, we should apply a weighting function ( $c_t$ ) on the Hyperbolic penalty function (equation 2) to get a new object function like

$$J_{wt} = \sum_t c_t H(r_t). \quad (6)$$

But in doing this way, we need the  $H(r_t)$  to be a norm, which is some function satisfied

$$C \text{ Norm}(x) = \text{Norm}(Cx), \quad (7)$$

where  $C$  is a constant value independent with variable  $x$ .

However the Hyperbolic penalty function (equation 2), which used to be named as ‘‘Hybrid norm’’ but later people found it is not a real ‘‘norm’’, is not satisfied equation 7 actually.

So we have two choices to apply weighting function. One is

$$J_{wt} = \sum_t c_t H(r_t), \quad (8)$$

and the other is

$$J_{wt} = \sum_t H(c_t r_t). \quad (9)$$

What is the difference between them? The equation 9 can be represented into

$$J_{wt} = \sum_t \sqrt{c_t^2 r_t^2 + R_{thr}^2} - R_{thr} \quad (10)$$

$$= \sum_t c_t \left( \sqrt{r_t^2 + \frac{R_{thr}^2}{c_t^2}} - \frac{R_{thr}}{c_t} \right) \quad (11)$$

$$= \sum_t c_t H_{new}(r_t), \quad (12)$$

where  $H_{new}(r_t) = \sqrt{r_t^2 + \frac{R_{thr}^2}{c_t^2}} - \frac{R_{thr}}{c_t}$  is a hyperbolic penalty function with a new threshold  $R'_{thr} = \frac{R_{thr}}{c_t}$ .

It is very obvious that the equation 9 is a way to add a weighting on the object function plus changing the threshold into time variant in the meanwhile.

## new implementation with new weighing

Now we add the weight function  $c(t)$  to compensate the divergence.

First the object function need to include the new weight function. We use the way described in 9, so the equation 1 changes into

$$J = \sum_t H(c_t r_t). \quad (13)$$

Consequently, the new gradient direction equation 3 becomes

$$\Delta u_\tau = \frac{\partial J}{\partial u_\tau} = \sum_t c_t r_{t+\tau} H'(c_t r_t). \quad (14)$$

Here is exactly the place where I made mistake at the first time. I use the notation  $\mathbf{a} \otimes \mathbf{b}$  to represent the element-by-element product of two vectors  $\mathbf{a}$  and  $\mathbf{b}$ , and I thought the Fourier domain update should be

$$\Delta \mathbf{U} = \overline{\text{FT}(\mathbf{c} \otimes \mathbf{r})} \text{FT}(\mathbf{H}'(\mathbf{c} \otimes \mathbf{r})). \quad (15)$$

But in fact it should be

$$\Delta \mathbf{U} = \overline{\text{FT}(\mathbf{r})} \text{FT}(\mathbf{c} \otimes \mathbf{H}'(\mathbf{c} \otimes \mathbf{r})). \quad (16)$$

Or in words, the weighting function should be applied on the second softclip function instead of on the residual function.

We can clearly see this in equation 14. Notice in the term  $c_t r_{t+\tau} H'(c_t r_t)$ , the index of the weighting  $c$  and the softclip function  $H'$  are the same ( $t$ ), but the index of the  $t$  is different ( $t+\tau$ ).

Now we continue to check the second major step, the link between model gradient direction and the residual update direction. Due to this link has nothing to do with the new object function, it stays unchanged. In another word, equation 4 is still valid in the new implementation with weighing function.

The incremental step length  $\alpha_{inc}$  in Newton search for updating step length  $\alpha$  depends on the object function, hence it also has to be changed into

$$\alpha_{inc} = -\frac{\sum_t \Delta(c_t^2 r_t H'(c_t r_t))}{\sum_t c_t^4 (\Delta r_t)^2 H''(c_t r_t)}. \quad (17)$$

Now we have all we need to implement model space weight in Fourier domain Bidirectional deconvolution, if I did all the derivation above correctly

## SYNTHETIC DATA EXAMPLE

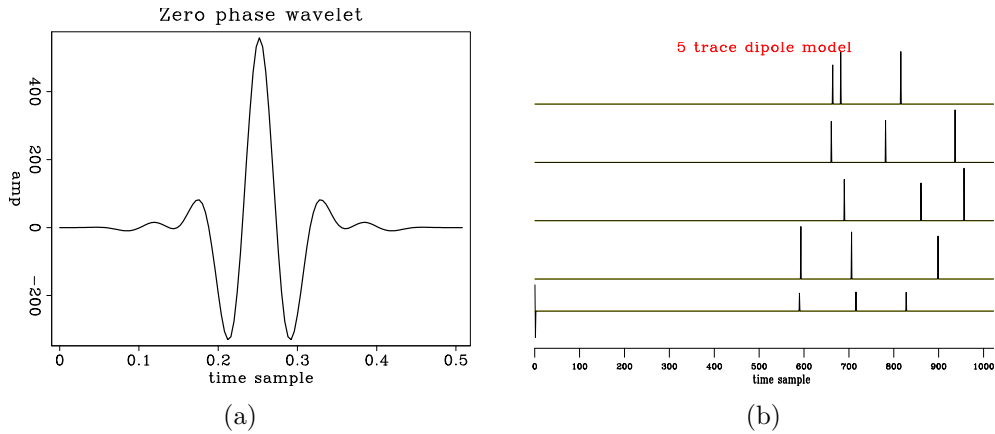


Figure 1: Wavelet and five traces model used for synthetic example.

## REFERENCES

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

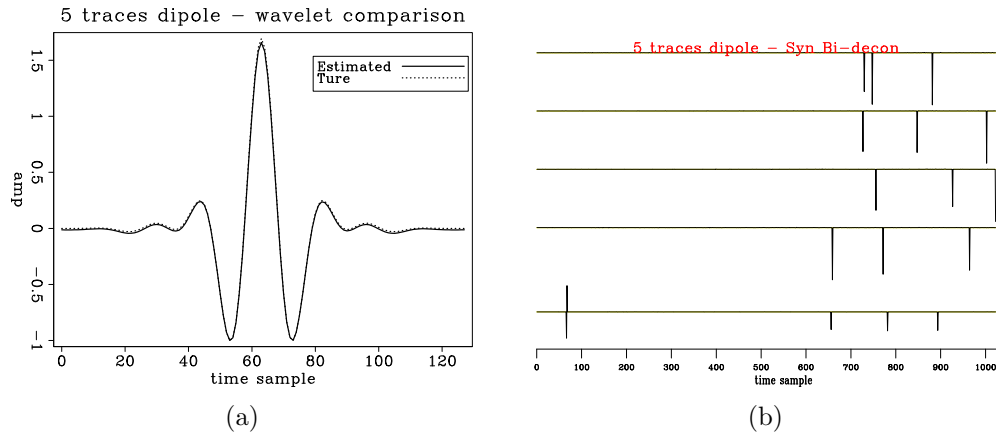


Figure 2: Estimated wavelet and recovered result.

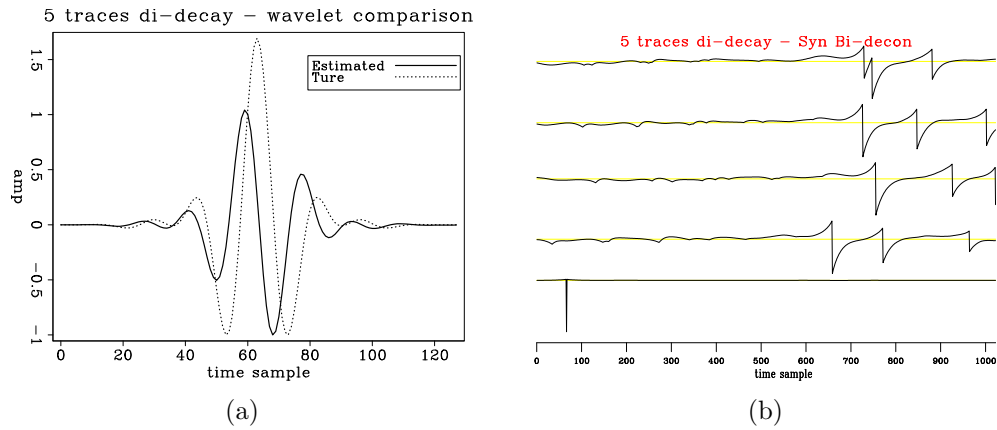


Figure 3: Estimated wavelet and recovered result of decay data without weighting.

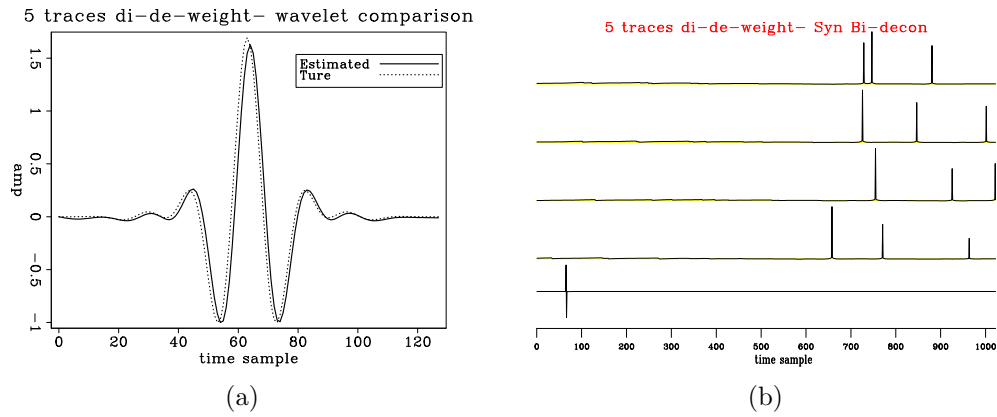


Figure 4: Estimated wavelet and recovered result of decay data with weighting.