Stanford Exploration Project, Report 117, October 23, 2004, pages 129–140

Estimating a 2D stationary PEF on sparse data

Jesse Lomask[1]

ABSTRACT

A stationary 2D PEF and missing data are simultaneously estimated on sparse data where the 2D PEF is never fully on known data. This PEF is estimated using non-linear conjugate gradients. A weight is applied to the residual to use only fitting equations where a prescribed minimum number of the PEF coefficients are on known data. The minimum parameter is then reduced and a new 2D PEF is estimated using the previous PEF as a starting solution. This process is repeated and the 2D PEF is gradually built up. This method is tested on the Madagascar satellite data. Using increasingly sparse data, the sparse 2D PEF compares favorably to the 2D PEF estimated on the dense data even when 67 percent of the data is unknown.

INTRODUCTION

Prediction error filters (PEFs) are commonly used to interpolate data. This is typically carried out in a two step process. First the PEFs are estimated on areas where all of the filter coefficients land on known data. Then the PEFs are used to fill in the missing data. However, sometimes the data is so sparse that PEFs estimated on entirely known data do not adequately capture the nature of the data. In Curry (2003), a multi-scale approach estimates a non-stationary PEF on sparse data. This requires that the gaps in the data range continuously in size. If the data has only a few scales of gaps then this multi-scale approach fails.

I present an approach to estimate a stationary PEF on sparse data using non-linear conjugate gradients (Claerbout, 1999) where both the filter and the missing data are estimated simultaneously. I add a weight to use only fitting equations where a prescribed minimum number of filter coefficients are on known data. Then, I reduce the minimum number of filter coefficients and solve again. I repeat this bootstrapping process until all of the data is used. A variant of this method was first suggested by Jon Claerbout and described in Lomask (2002). In this paper, this method proves capable of estimating a 2D PEF on datasets where the known data occurs systematically as in the Madagascar satellite data (Ecker and Berlioux, 1995; Lomask, 1998, 2002; Curry, 2004b) or possibly the data used in Curry (2004a). If the data is not too radically non-stationary, this stationary PEF can, in principle, be used as a starting solution for tackling the non-stationary problem.

In the following paper, I first present the methodology for estimating the stationary PEF

[1]**email:** lomask@sep.stanford.edu

on sparse data. Then I illustrate its effectiveness on the Madagascar satellite data.

METHODOLOGY

Equation (1) is taken from Chapter 6. of Claerbout (1999).

$$\mathbf{0} \quad \approx \quad \begin{bmatrix} \mathbf{AJ} & \mathbf{YK} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{y} \\ \Delta\mathbf{a} \end{bmatrix} + \bar{\mathbf{r}}. \tag{1}$$

This fitting goal estimates both missing data and a filter simultaneously. **A** and **Y** are the convolutional matrix algebraic notations for the filter and the data, respectively. $\Delta\mathbf{a}$ and $\Delta\mathbf{y}$ are perturbation vectors for the filter and the data. The free-mask matrix for missing data is denoted **J** and that for the PEF is **K**. The original residual is defined as $\bar{\mathbf{r}} = \mathbf{Ay}$.

Next, a diagonal weight matrix **W** can be applied to the residual as:

$$\mathbf{0} \quad \approx \quad \mathbf{W}\left[\begin{bmatrix} \mathbf{AJ} & \mathbf{YK} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{y} \\ \Delta\mathbf{a} \end{bmatrix} + \bar{\mathbf{r}}\right]. \tag{2}$$

This weight is used to remove any fitting equations in which the filter has fewer than a specified number of coefficients on known data. The number of coefficients that are on known data at each point in the model can be thought of as the coefficient fold. The range of coefficient fold values is broken up into several steps. Equation (2) is first solved using fitting equations with coefficient fold greater than the highest step. Then the resulting PEF and missing data are used as the initial solution for the next step. This is repeated and the 2D PEF is gradually built up.

Within the solver, the following equations are iterated over (Claerbout, 1999):

$$\mathbf{r} \quad \longleftarrow \quad \mathbf{WAy} \tag{3}$$

$$\begin{bmatrix} \Delta\mathbf{y} \\ \Delta\mathbf{a} \end{bmatrix} \quad \longleftarrow \quad \begin{bmatrix} \mathbf{J'A'} \\ \mathbf{K'Y'} \end{bmatrix} \mathbf{W'r} \tag{4}$$

$$\Delta\mathbf{r} \quad \longleftarrow \quad \mathbf{W}\begin{bmatrix} \mathbf{AJ} & \mathbf{YK} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{y} \\ \Delta\mathbf{a} \end{bmatrix} \tag{5}$$

$$\mathbf{y} \quad \longleftarrow \quad \text{cgstep}(\mathbf{y}, \Delta\mathbf{y}) \tag{6}$$

$$\mathbf{a} \quad \longleftarrow \quad \text{cgstep}(\mathbf{a}, \Delta\mathbf{a}) \tag{7}$$

Balancing the unknown data and the filter

When solving an inversion problem, it can be useful to look at the residual for a couple of reasons. The first is to see that it is white. The second reason, important in the case of regularized inversion, is to see that the fitting goals have been properly balanced.

In equation (2), we are finding both a filter and missing data at the same time. If the solver focuses entirely on finding the filter, it will not change the missing data and create a filter that is estimated on incorrect data. On the other hand, if the solver focuses entirely on missing data, it does not change the filter and creates incorrect interpolation results.

I found it useful to inspect movies of the gradient to balance the data and filter. Since the individual residuals of $\mathbf{WAJ}\Delta\mathbf{y}$ and $\mathbf{WYK}\Delta\mathbf{a}$ are summed, it does not make sense to look at the total residual to see how well the filter and missing data are balanced. Alternatively, the individual residuals could be looked at before they are summed. However, I found it useful to make movies of each step in the solver of the gradient, equation (4).

By observing the relative sizes of the perturbation of the filter and missing data in movies of the gradient, the two can be easily balanced by scaling the input data, $\alpha\mathbf{y}$. Conceptually, it may sound better to implement a model weight to balance the filter and the data, but I got better results by merely scaling the input data.

TEST RESULTS

I tested this method on the Madagascar satellite data set with increasingly sparse missing data. I found that this method estimated a 2D PEF that is comparable to a 2D PEF estimated on dense data when the sparse data was 67 percent unknown. For input data that was more sparse than that, the method failed.

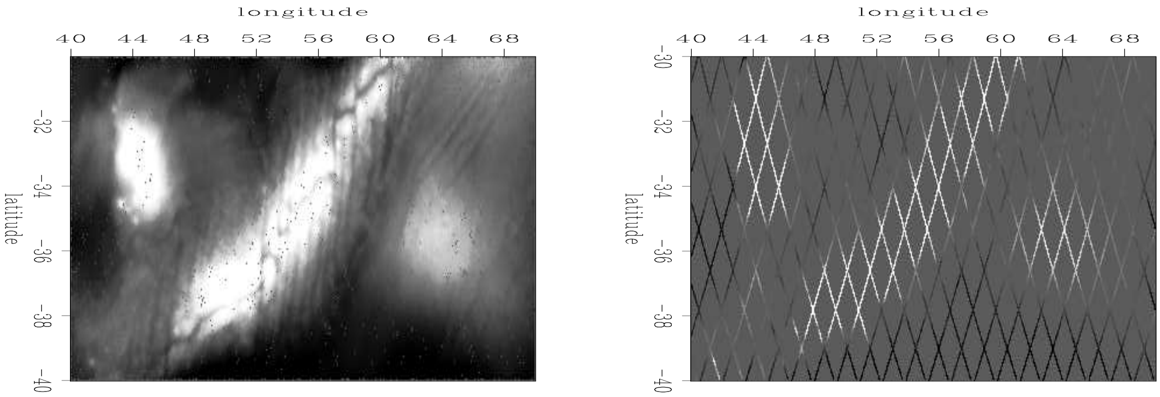


Figure 1: Left is the dense data in the overlapping region. Right is the sparse data. jesse1-dense_sparse [ER]

The data consists of two overlapping datasets. One is densely sampled and one is sparsely sampled. I chose to work in the overlapping region. On the left side of Figure 1 is the dense data in the southern half of the Madagascar data. The right of the figure is the sparse data overlapping the dense data.

Figure 2 is the dense data after its few empty bins have been filled with a laplacian. The right side has been roughened with the helical derivative.

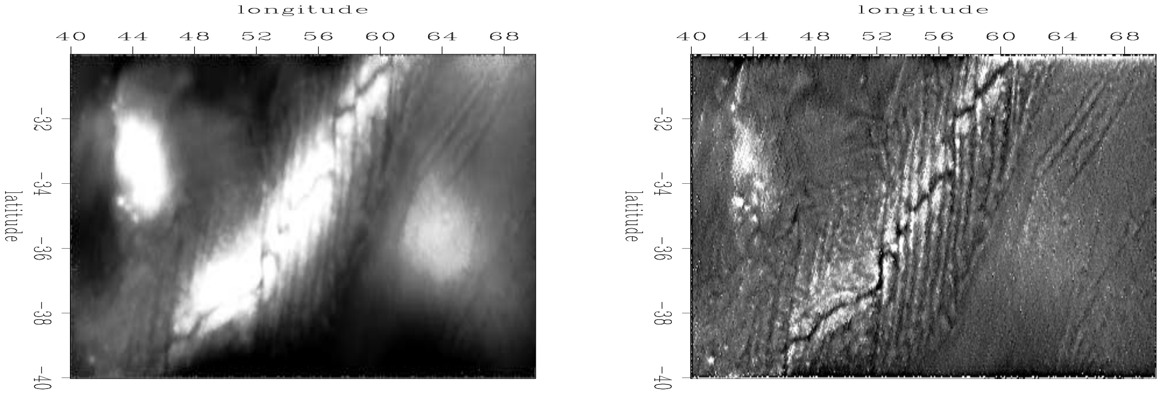


Figure 2: Left is the dense data with it few empty bins filled with a laplacian. Right is the same data roughened with the helical derivative. jesse1-dense_fill [ER]

Figure 3: Left is the impulse response of the inverse full 2D PEF estimated on southern dense region after the few remaining bins have been filled using a 2D gradient as regularization. Right is the impulse response of the inverse helical derivative. This is the starting solution for the sparse 2D PEF estimation. jesse1-comp_imp [ER]

I began with the densely sampled data and gradually threw away data. As I threw out data, I compared the impulse response of the inverse 10 by 10 PEF estimated on the dense data to the impulse response of the inverse PEF estimated using the approach described in the paper. The left side of Figure 3 is the impulse response wrapped on the helix of the inverse PEF estimated on the dense data in Figure 2. Notice that it has a similar character to data it is trying to emulate. This is the result that the sparse 2D PEF is hoping to achieve.

For nonlinear problems, the initial solution is extremely important. Thus far, I have used only the helical derivative (Claerbout, 1999). In the future, other starting solutions may improve the results, such as the solution used in Curry (2004b). In the right side of Figure 3, is the impulse response of the inverse helical derivative.

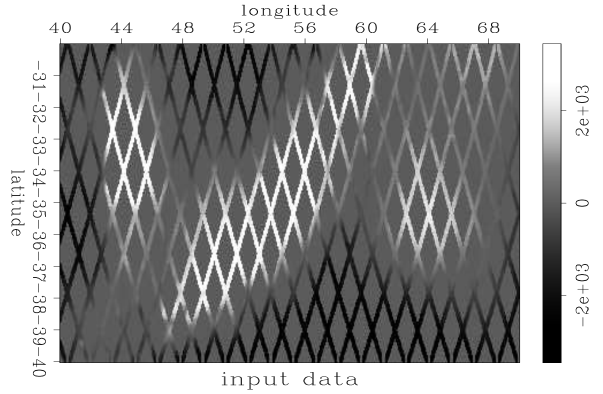


Figure 4: This is the map to be interpolated. The tracks have been made thicker therefore it is less sparse than the original sparse tracks. jesse1-data_mask.4.wide [ER]

Figure 4 shows thicker tracks than the original sparse data shown on the right side of Figure 1. This has 2.6 times as many non-zero values as the original sparse data.

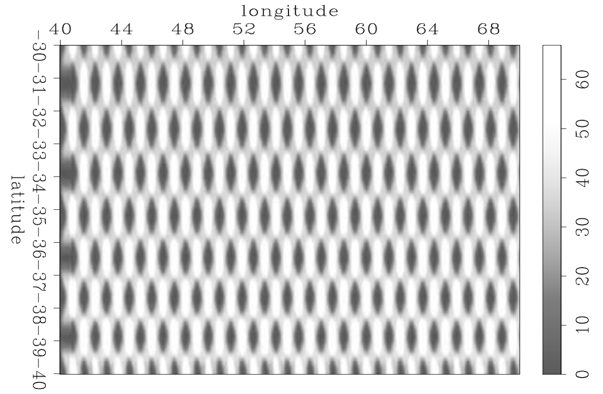


Figure 5: A coefficient fold map for the tracks in Figure 4 with a 10 by 10 filter. The value indicates the number of coefficients of the filter that are on known data at each location in the model. jesse1-coef_fold.4.wide [ER]

Figure 5 shows the coefficient fold map for a 10 by 10 filter on the sparse data in Figure 4. The number of coefficients that are on known data at each point in the model can be thought of as the coefficient fold. A coefficient fold map can easily be created by packing the filter with all ones and setting the known data in the model to ones and the unknown to data to zeros. The output of the convolution is a coefficient fold map.

In Figure 6 is a histogram of the convolution map in Figure 5. For this 10 by 10 PEF, there are 98 coefficients. It can be seen in the histogram that the maximum coefficient fold is approximately 67. This means that at those locations 31 filter coefficients are on unknown data.

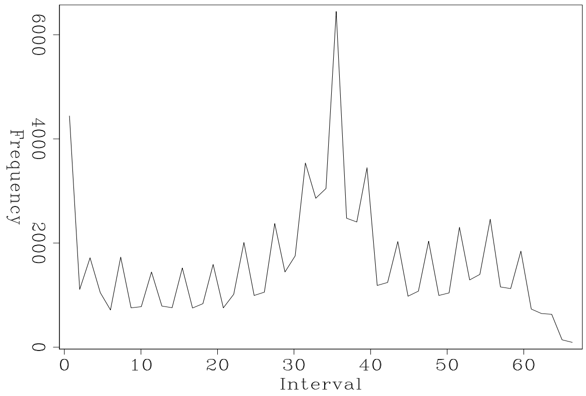


Figure 6: A histogram of the values in the coefficient fold map in Figure 5. Notice that most of the filter locations have about 33 out of about 98 coefficients on known data. The areas with the highest (50-65) are located where the tracks cross. jesse1-histo.4 [ER]

Also, most of the filter locations have about 33 coefficients on known data, 65 coefficients on unknown data. In this case, I set the first minimum fold parameter to 60, solved for the filter and missing data. Then set the minimum fold parameter to 50 and repeated until all of the data was used.

Figure 7: Left is the impulse response of the inverse 2D PEF estimated on the dense data in left of Figure 2. Right is the impulse response of the inverse 2D PEF estimated on the sparse tracks in Figure 4. jesse1-comp_imp2.4 [ER]

Figure 7 compares the impulse responses of the inverse PEF of the full 2D PEF on the left to the 2D PEF estimated on the sparse tracks from Figure 4 on the right. Although they are not exactly the same, they are similar. The sail shape possibly results from the PEF's ability to capture the narrow ridges trending almost north and the entire submerged mountain range trending west and its inability to capture the central spreading ridge trending northeast.

Another comparison to evaluate the quality of the result is to inspect the filled data. Figure 8 is the sparse tracks in Figure 4 filled with the full 2D PEF estimated on the dense data. Figure 9 is the sparse tracks in Figure 4 filled with the 2D PEF estimated on the sparse tracks themselves. In general, this method adequately fills the missing data, only showing some weakness in capturing the main spreading ridge feature.

If the data is too sparse, this method fails. Estimating a PEF on the original sparse tracks on the right side of Figure 1 illustrates this. First, we recalculate a new coefficient fold map shown in Figure 10. Analysis of the histogram in Figure 11 reveals that this data is indeed

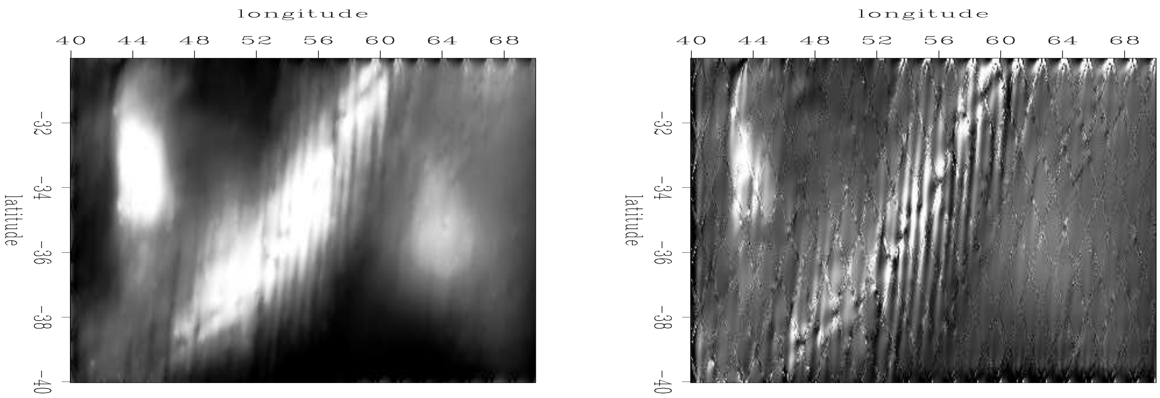


Figure 8: Left is the result of filling the sparse tracks in Figure 4 with a full 2D PEF estimated on the dense data in left side of Figure 2. Right is roughened with the helical derivative. jesse1-hel.2d.4.fill [ER,M]

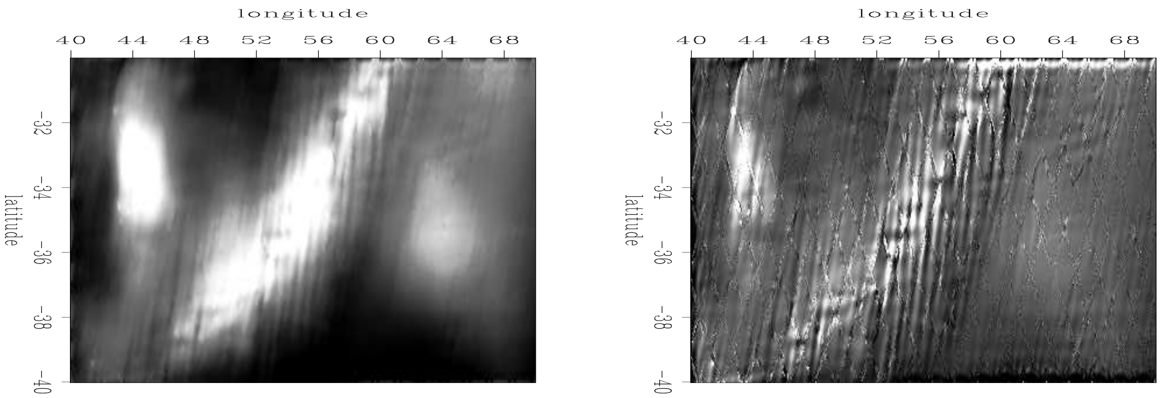
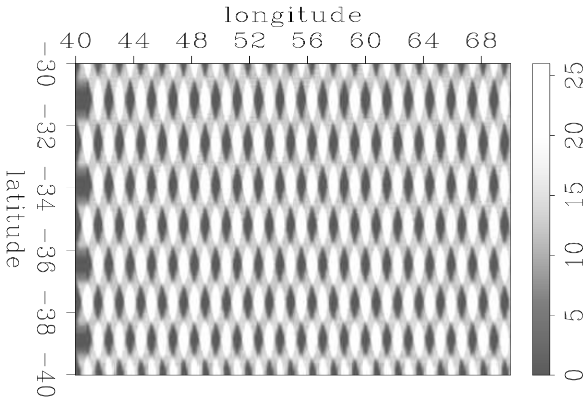


Figure 9: Left is the result of estimating a sparse 2D PEF on the sparse tracks in Figure 4 and filling in the same data. Right is roughened with the helical derivative. Notice the similarity to Figure 8. jesse1-recursive_pef.4.fill [ER,M]

Figure 10: A coefficient fold map for the sparse tracks with a 10 by 10 filter. The value indicates the number of coefficients of the filter that are on known data at each location in the model. jesse1-coef_fold [ER]



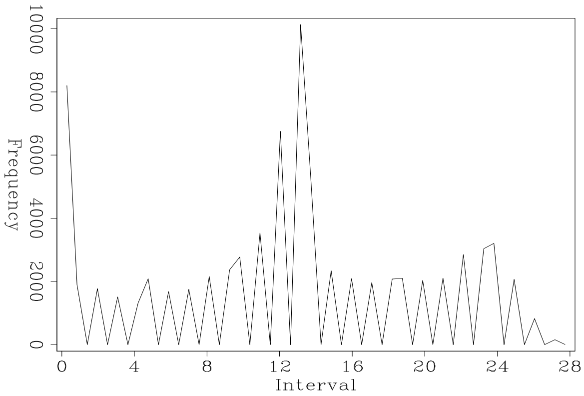


Figure 11: A histogram of the values in the coefficient fold map in Figure 10. Notice that most of the filter locations have about 13 out of about 98 coefficients on known data. jesse1-histo.1 [ER]

Figure 12: Left is the impulse response of the inverse 2D PEF estimated on the dense data in left of Figure 2. Right is the impulse response of the inverse 2D PEF estimated on the sparse tracks in the right of Figure 1. The two impulse responses are noticeably different. jesse1-comp_imp2.1 [ER]

very sparse. Then, Figure 12 compares the impulse responses. The sparse 2D PEF on the right does a very poor job of matching the desired response on the left. Comparison of the filled results in Figures 13 and 14 further demonstrate the sparse 2D PEF inability to capture much of the high frequencies.

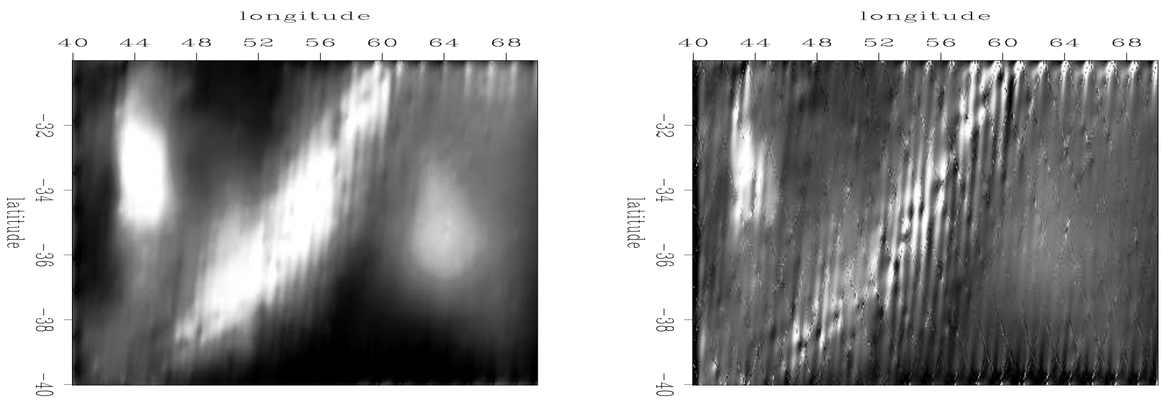


Figure 13: Left is the result of filling the sparse tracks in the right of Figure 1 with a full 2D PEF estimated on the dense data in left side of Figure 2. Right is roughened with the helical derivative. jesse1-hel.2d.1.fill [ER,M]

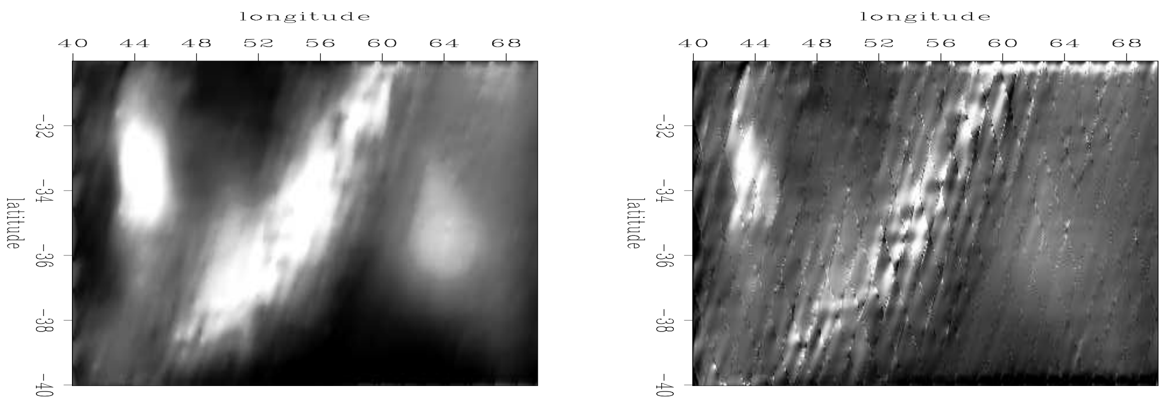


Figure 14: Left is the result of estimating a sparse 2D PEF on the sparse tracks in the right Figure 1. Right is roughened with the helical derivative. Unfortunately, it is not able to capture the higher frequency nature of the data as captured by the full 2D PEF result shown in Figure 13. jesse1-recursive_pef.1.fill [ER,M]

FUTURE WORK

There are still several ways of improving this method to be investigated. For instance, we could start with a smaller PEF and gradually increase its size, iteratively recalculating a new 2D PEF in a similar bootstrapping approach. Also, it would be prudent to come up with a good way of finding the data scale to balance the filter and the missing data. This would probably depend on the relative size of the data and filter. Lastly, for this technique to be useful on seismic data, it has to be able solve the non-stationary problem. We could possibly use this stationary PEF as a starting guess for non-stationary problem. Using code adapted from Clapp (2000), a non-stationary PEF is estimated on the dense data and used to fill in the sparse data in Figure 15. Although this figure was created using the dense data, it illustrates what we hope to achieve with a non-stationary PEF estimated on sparse data tracks.

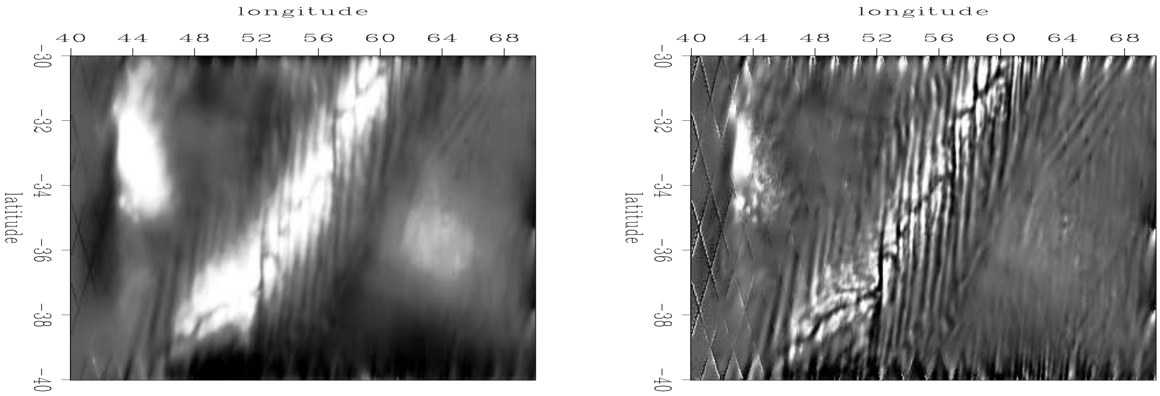


Figure 15: The result of filling the sparse tracks in the Right of Figure 1 with a non-stationary 2D PEF estimated on the dense data in Figure 2. jesse1-non_station.fill [ER,M]

CONCLUSION

The 2D PEF estimated on sparse data compares favorably to the 2D PEF estimated on the dense data in some cases. For the Madagascar data set, when the unknown data exceeds 67 percent, this method fails. However, up to that point, this method is able to estimate a 2D PEF when conventional estimation techniques would certainly fail.

ACKNOWLEDGMENTS

I would like to thank Antoine Guitton, Paul Sava, Alejandro Valenciano, and Bill Curry for useful suggestions.

REFERENCES

Claerbout, J., 1999, Geophysical estimation by example: Environmental soundings image enhancement: Stanford Exploration Project, `http://sepwww.stanford.edu/sep/prof/`.

Clapp, R. G., 2000, Regularizing tomography with non-stationary filters: SEP–**103**, 47–60.

Curry, W., 2003, More fitting equations for PEF estimation on sparse data: SEP–**114**, 171–176.

Curry, W., 2004a, Midpoint-offset vs. source-receiver coordinates for PEF-based interpolation: SEP–**117**.

Curry, W., 2004b, Regularizing madagascar: Pefs from the data space?: SEP–**115**, 347–356.

Ecker, C., and Berlioux, A., 1995, Flying over the ocean southeast of Madagascar: SEP–**84**, 295–306.

Lomask, J., 1998, Madagascar revisited: A missing data problem: SEP–**97**, 207–216.

Lomask, J., 2002, Madagascar satellite data: An inversion test case: SEP–**111**, 337–349.