Stanford Exploration Project, Report 97, July 8, 1998, pages 183–193

Stanford Exploration Project, Report 97, July 8, 1998, pages 183–193

Decon and interpolation with nonstationary filters

Sean Crawley, Robert Clapp, and Jon Claerbout[1]

keywords: *missing data, interpolation, nonstationary*

ABSTRACT

Building on the notions of time-variable filtering and the helix coordinate system, we develop software for filters that are smoothly variable in multiple dimensions. Multiscale prediction-error filters (PEFs) can estimate dips from recorded data and use the dip information to fill in unrecorded shot or receiver gathers. The data are typically divided into patches with approximately constant dips. Instead, we estimate a set of smoothly varying filters, up to one PEF for every data sample. They are more memory-intensive to estimate, but the smoothly varying filters do give more accurate interpolation results than discrete patches. Finally, we offer an improved method of controlling the smoothness of the filters. We design filters like directional derivatives that we call "flag filters". They destroy dips in easily adjusted directions. We use them in residual space to encourage dips in the specified directions. We develop the notion of "radial-flag filters", i.e., flag filters oriented in the radial direction (lines of constant x/t in (t, x) space). Break a common-midpoint gather into pie shaped regions bounded by various values of x/t. Such a pie-shaped region tends to have constant dip spectrum throughout the region so it is a natural region for smoothing estimates of dip spectra or of gathering statistics (via 2-D PEFs). In this paper we use smoothly variable PEFs to interpolate missing traces, though obviously they have many other uses.

INTRODUCTION

Claerbout (1997) describes a helical coordinate to cast multi-dimensional filtering as one dimensional, enabling the use of some well-developed signal processing theory in applications including missing data interpolation (Fomel et al., 1997) and low-cut filtering (Claerbout, 1998). Missing data interpolation with PEFs is typically done in small patches where dips are approximately stationary (Abma and Claerbout, 1995), to account for nonstationarity in the data. Claerbout (1997) describes a method for estimating smoothly time-varying PEFs without patching. We use the helix to extend the idea of smooth time-variable PEF estimation to smooth time- and space-variable PEF estimation. The new PEFs can perform better at interpolating missing data

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

than PEFs estimated in patches. More applications will come later.

Clapp et al. (1997) show how to use space-variable inverse flag (or "steering") filters to smooth in adjustable directions, and they show how to solve empty-bin problems filling in missing data along the directions of the flags. We use space-variable flag filters to control the direction of smoothness between PEFs. We orient the flag filters radially in a CMP gather to encourage PEFs to have the same dip information along lines of constant x/t, where data tends to have constant dip spectra. In this paper we review the theory for estimating nonstationary PEFs, and show examples of their application to missing data interpolation. Finally, we describe an improvement to filter estimation for CMP gathers using "radial-flag filters."

TIME- AND SPACE-VARYING PEFS

The time dip of seismic data changes rapidly along many axes, so a single PEF can only represent a small amount of data. Often we divide the data into patches, where we assume the data have constant dips. Because seismic data have curvature and may not be well represented by piecewise-constant dips, it is appealing to extend the idea of time-variable filtering to include spatial dimensions as well, and have smoothly varying PEFs to represent curved events.

Instead of one PEF per patch, we estimate a PEF for every output data point; changing the problem from overdetermined to very underdetermined. We can estimate all these filter coefficients by the usual formulation, supplemented with some damping equations, say

$$\begin{aligned} \mathbf{0} &\approx \mathbf{YKa} + \mathbf{r}_0 \\ \mathbf{0} &\approx \epsilon\, \mathbf{Ra} \end{aligned} \tag{1}$$

where $\mathbf{R}$ is a roughening operator, $\mathbf{Y}$ is convolution with the data, and $\mathbf{K}$ is a known filter coefficient mask.

When the roughening operator $\mathbf{R}$ is a differential operator, the number of iterations can be large. We can speed the calculation immensely and make the equations somewhat neater by "preconditioning". When we define a new variable $\mathbf{p}$ by $\mathbf{a} = \mathbf{Sp}$ and insert it into (1) we get

$$\mathbf{0} \approx \mathbf{YKSp} + \mathbf{r}_0 \tag{2}$$

$$\mathbf{0} \approx \epsilon\, \mathbf{RSp} \tag{3}$$

Now, because the smoothing and roughening operators are somewhat arbitrary, we may as well replace $\mathbf{RS}$ by $\mathbf{I}$ and get

$$\mathbf{0} \approx \mathbf{YKSp} + \mathbf{r}_0 \tag{4}$$

$$\mathbf{0} \approx \epsilon\, \mathbf{Ip} \tag{5}$$

We solve for $\mathbf{p}$ using conjugate gradients. To see $\mathbf{a}$, we just use $\mathbf{a} = \mathbf{Sp}$.

To reduce clutter, we could drop the damping (5) and keep only (2); then to control the null space, we need only to start from a zero solution and limit the number of iterations. As a practical matter, without (5) we must find a good number of iterations and with it we must find a good value for ϵ.

For $\mathbf{S}$ we can use polynomial division by a Laplacian or by filters with a preferred direction. If the data are CMP gathers, it is attractive to use radial filters, which are explained further down.

INTERPOLATING MISSING TRACES

We estimate missing data in two steps of linear least squares (Claerbout, 1992). The first step is estimation of PEFs. After the PEFs have been estimated they are used to fill in the empty trace bins. This is the second step of least squares. We want the recorded and estimated data to have the same dips. Since the dip information is now carried in the PEFs, this is once again specifying that the convolution of the filter and data should give the minimum output, except that now the filters are known and the data is unknown. We constrain the data by specifying that the originally recorded data cannot change. To separate the known and unknown data we have a known data selector $\mathbf{K}$ and an unknown data selector $\mathbf{U}$, with $\mathbf{U} + \mathbf{K} = \mathbf{I}$. These multiply by 1 or 0 depending on whether the data was originally recorded or not. With $\mathbf{A}$ signaling convolution with the PEF and $\mathbf{y}$ the vector of data, the regression is $0 \approx \mathbf{A}(\mathbf{U} + \mathbf{K})\mathbf{y}$, or $\mathbf{AUy} \approx -\mathbf{AKy}$.

While a complete set of PEFs works well for destroying the data, I have found that it is not the best choice for reconstructing it; interpolation with PEFs estimated at every data point gives poor results. Probably that just means I have not found the right value of ϵ in (5), or that I have not found the right roughener. At any rate, it is easy to just reduce the number of filters slightly, from every data point to perhaps every fifth. This is similar to putting an extra roughener in the damping equation, in that it is essentially an infinite penalty on variations of $\mathbf{p}$ between small groups of samples, and it has the important economizing effect of reducing the memory allocation. A full set of $5 \times 3 \times 2$ PEFs is like allocating an extra 22 copies of the data in memory, a slightly subsampled set is only a few extra copies. This is a similar to using very small patches. In the method where the patches are independent (Crawley, 1998), the number of filter coefficients puts a lower bound on patch size; the problem has to stay well overdetermined to produce a useful PEF. Using smoothly nonstationary filters effectively reduces the minimum patch size, so that the filter estimation problem can be underdetermined.

Figure 1 illustrates the filter subsampling with a simple synthetic. A random 1-D sequence, uniformly shifted along the horizontal axis, is used to estimate a set of PEFs. A single PEF can predict this panel perfectly with a -1 value in the correct place. The right side of the figure graphs the individual filter coefficient that should be equal to -1, with the horizontal axis tracking different PEFs. Predictably, when

the PEFs are estimated at every point (the upper graph), the values have noticeably greater variance and decreased energy than when the PEFs are estimated at every fifth point (the lower graph). Not shown, the coefficients at other lags in the upper case have more variance and more energy than in the lower case. They upper values still destroy the data perfectly, but they cannot reconstruct it.

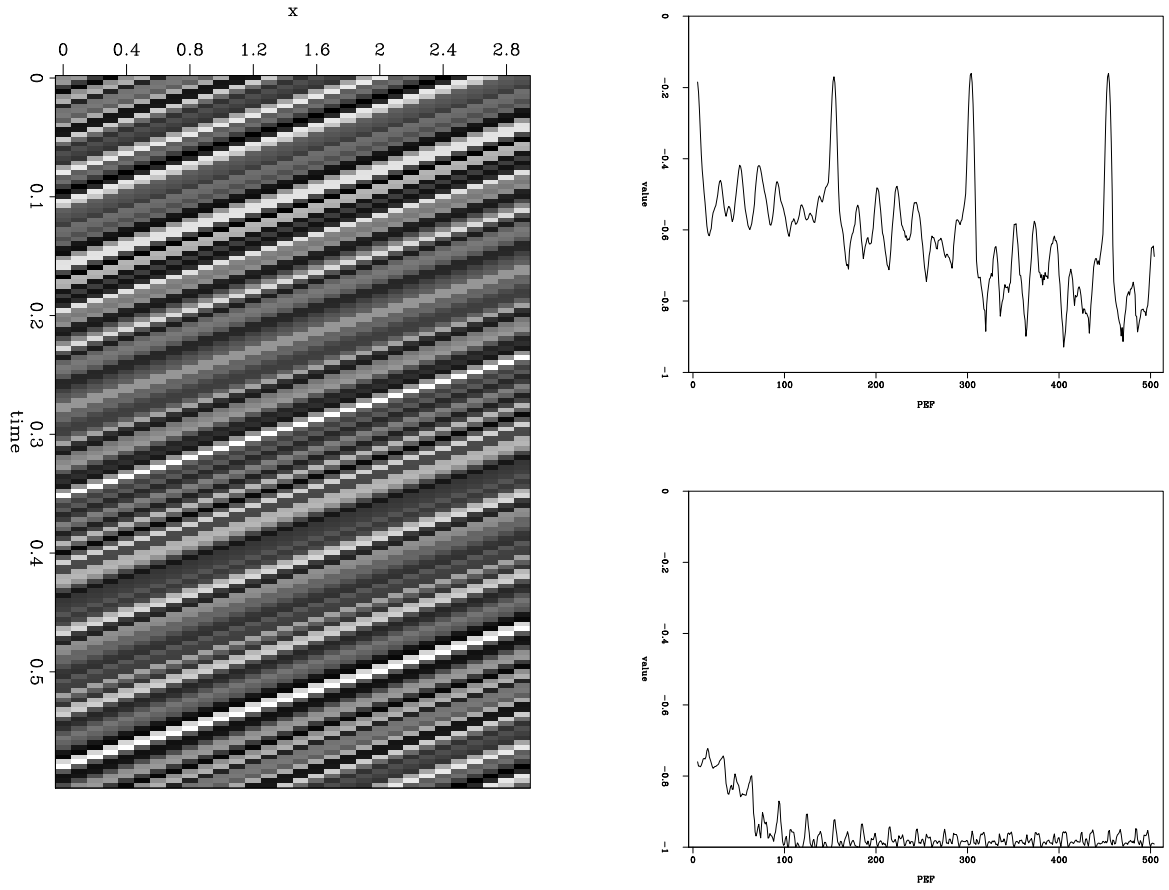

Figure 1: Effect of slightly subsampling the filter estimation. The left shows a very predictable synthetic. The right shows two sets of coefficients that should be equal to -1. The upper graph is the coefficients estimated at every data sample, the lower graph shows the coefficients obtained when PEFs are only estimated at every fifth data point. sean2-coeffs [CR]

Example

Figure 2 shows a cube of input data. It is a window from several shot gathers, reversed and added to itself to produce data with both seismic wavefield character and a complex set of conflicting dips. The upper panel shows the original cube, the lower panel shows the same cube after zeroing every other trace.

Figure 3 shows the same data after every other trace was zeroed, and then re-

estimated with smoothly varying PEFs. The lower panel of the figure shows the residual. For comparison, Figure 4 shows the same data after interpolation with a patching algorithm, along with the residual in the lower panel. The patching result is not bad, but it does not have the accuracy of the smoothly varying result.

FUTURE WORK

The examples in this paper show that missing data interpolation can benefit from smoothly varying filters. We expect to find several other interesting applications for time- and space-variable deconvolution; for instance as a prewhitening step to reduce edge effects in later processes.

Radial Smoothing

In the examples so far, the smoothing applied to filters has been (approximately) isotropic. Clapp et al. (1997) show how to control the direction of smoothing. In certain cases, it may make sense to specify some preferred direction of filter smoothing. For instance, CMP gathers tend to have approximately constant dip spectra at constant values of x/t, which correspond to radial lines. Figure 5 shows randomly scattered points smoothed with radial-flag filters, in the adjoint and forward directions, and in combination. Not surprisingly, the radial operator needs a divergence correction, to avoid all the energy in the panel collecting at the origin.

Fixed Coefficients

There is an issue dealing with the fixed coefficient of $\mathbf{p}$, that I believe is not a problem, but I place it under "future work" anyway because I expect it to come up occasionally. Our PEFs, $\mathbf{a}$, have a fixed coefficient that is defined to have the value 1. We actually estimate $\mathbf{p}$, which is related by $\mathbf{a} = \mathbf{Sp}$, which means we should load our initial model guess, $\mathbf{p}_0$, with some other value in the fixed coefficient, that will be integrated by $\mathbf{S}$ to give 1's. This is a hassle because our software has the value 1 built in. Luckily, the problem seems to disappear. Wherever the forward operator is applied, it looks like $\mathbf{YKSp}$, which is the same as $\mathbf{YKa}$. The adjoint is used to calculate the change in the roughened model variable $\mathbf{p}$: $\mathbf{\Delta p} = \mathbf{S'K'Y'r}$, where $\mathbf{r}$ is the residual. Whatever change is made to the fixed coefficient is zeroed by $\mathbf{K'}$. In other words, it seems we never need to know the fixed coefficient of $\mathbf{p}$.

CONCLUSIONS

We describe a method for estimating multi-dimensional, smoothly varying PEFs. In this paper we use them to interpolate missing data, but we expect to find more

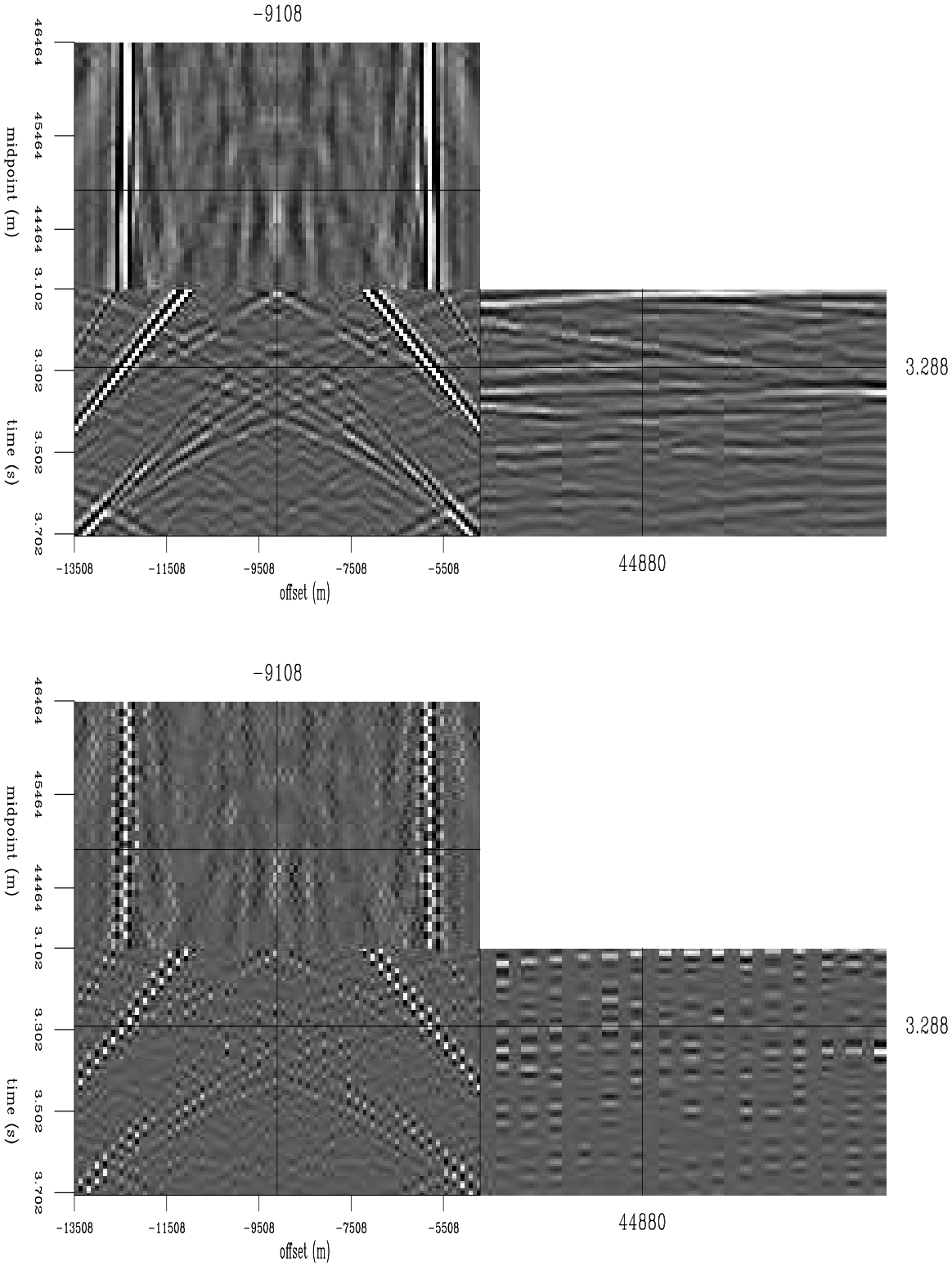

Figure 2: An interpolation test data cube. The top panel is the original data cube; the bottom panel is the same data after every other trace was zeroed. The bottom panel is the input to the interpolation. sean2-interpIn [CR]

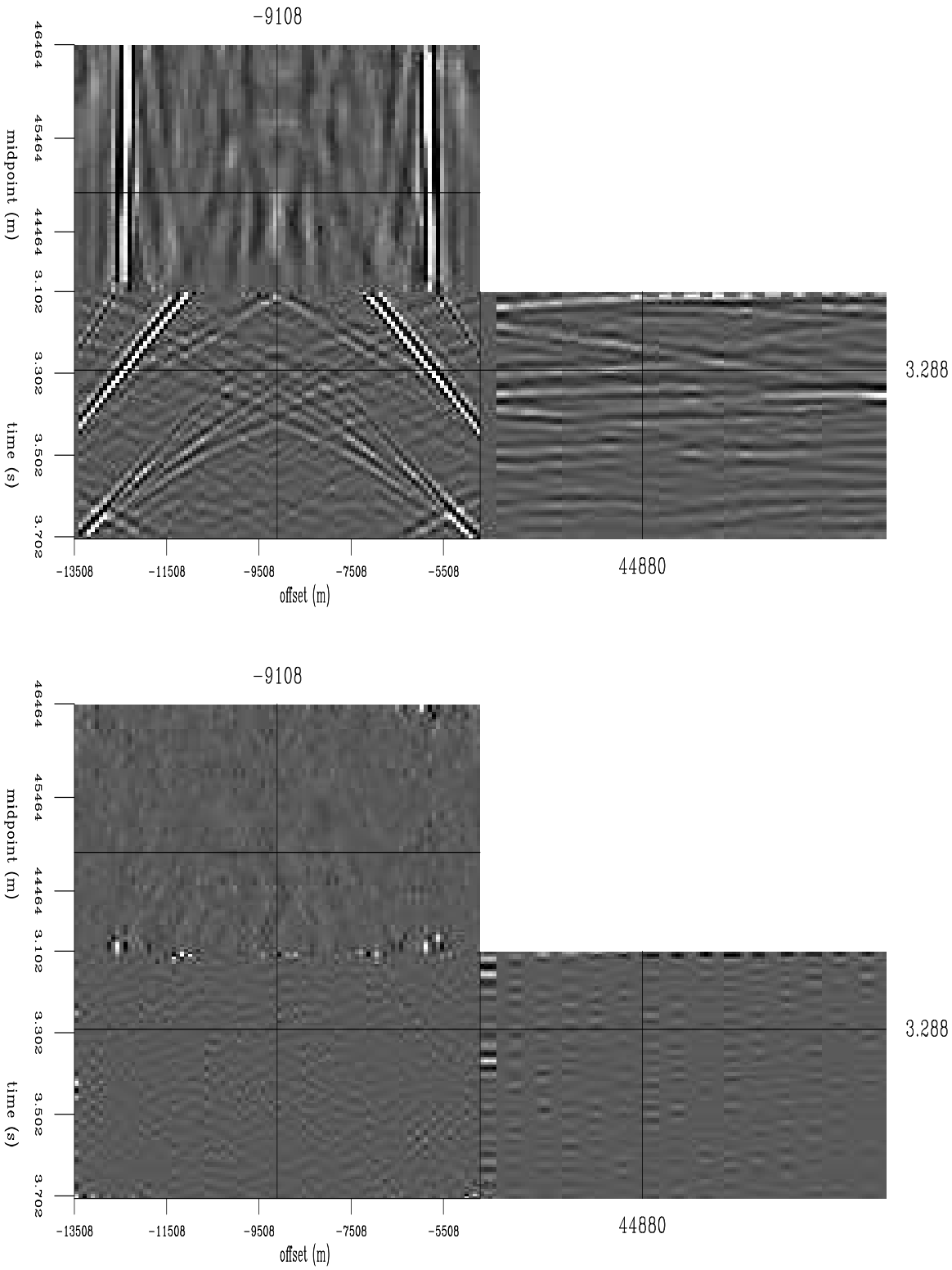

Figure 3: Output of smooth interpolation. Top panel shows interpolated data, bottom panel shows residual. Compared to Figure 4, there is little energy in the residual and it is difficult to tell which traces are interpolated. Clips are set equal. sean2-smoothInterp [CR]

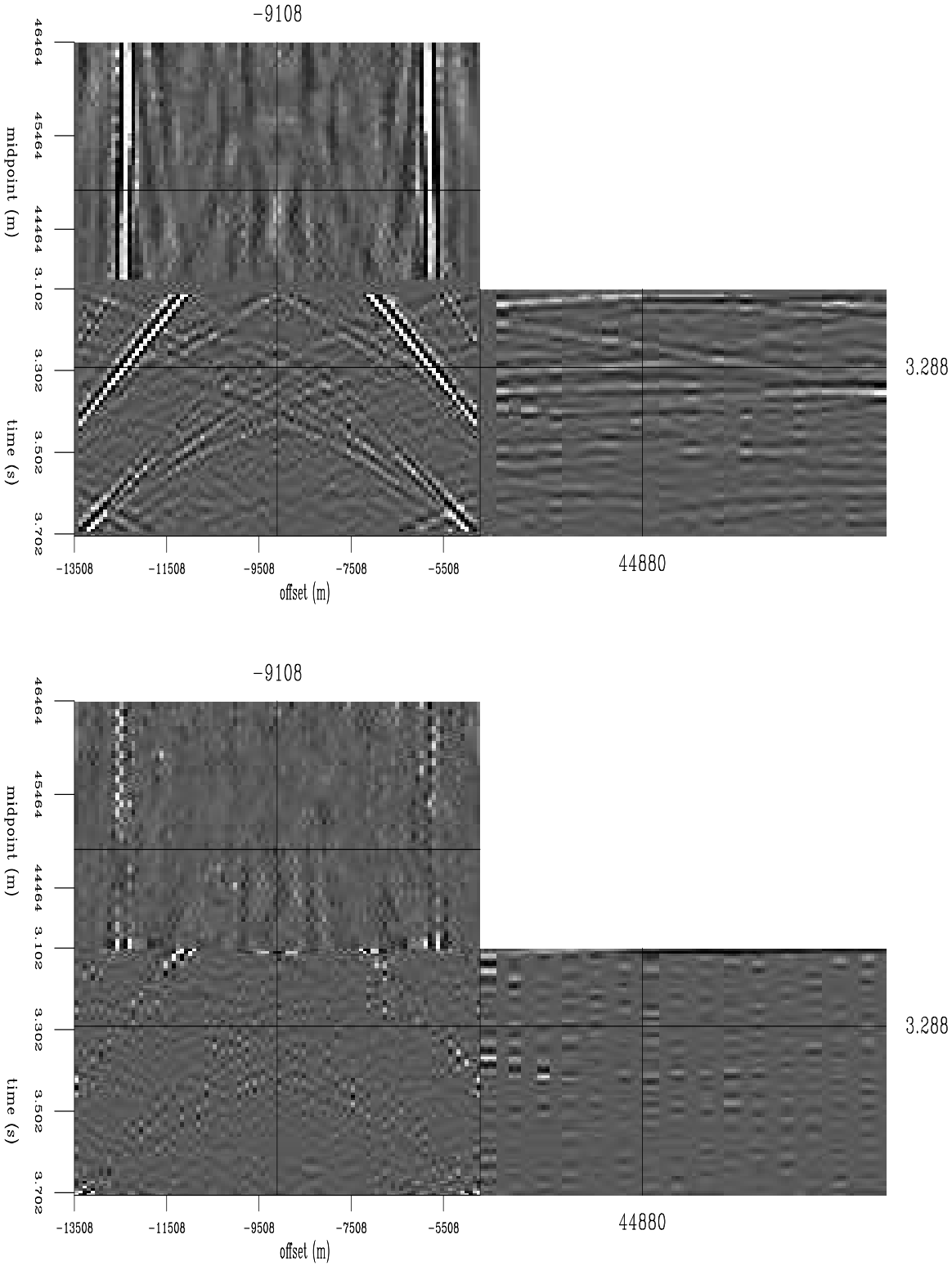

Figure 4: For comparison, output of patched interpolation. Top panel shows interpolated data, bottom panel shows residual. The residual is large compared to Figure 3. Clips are set equal. sean2-patchInterp [CR]

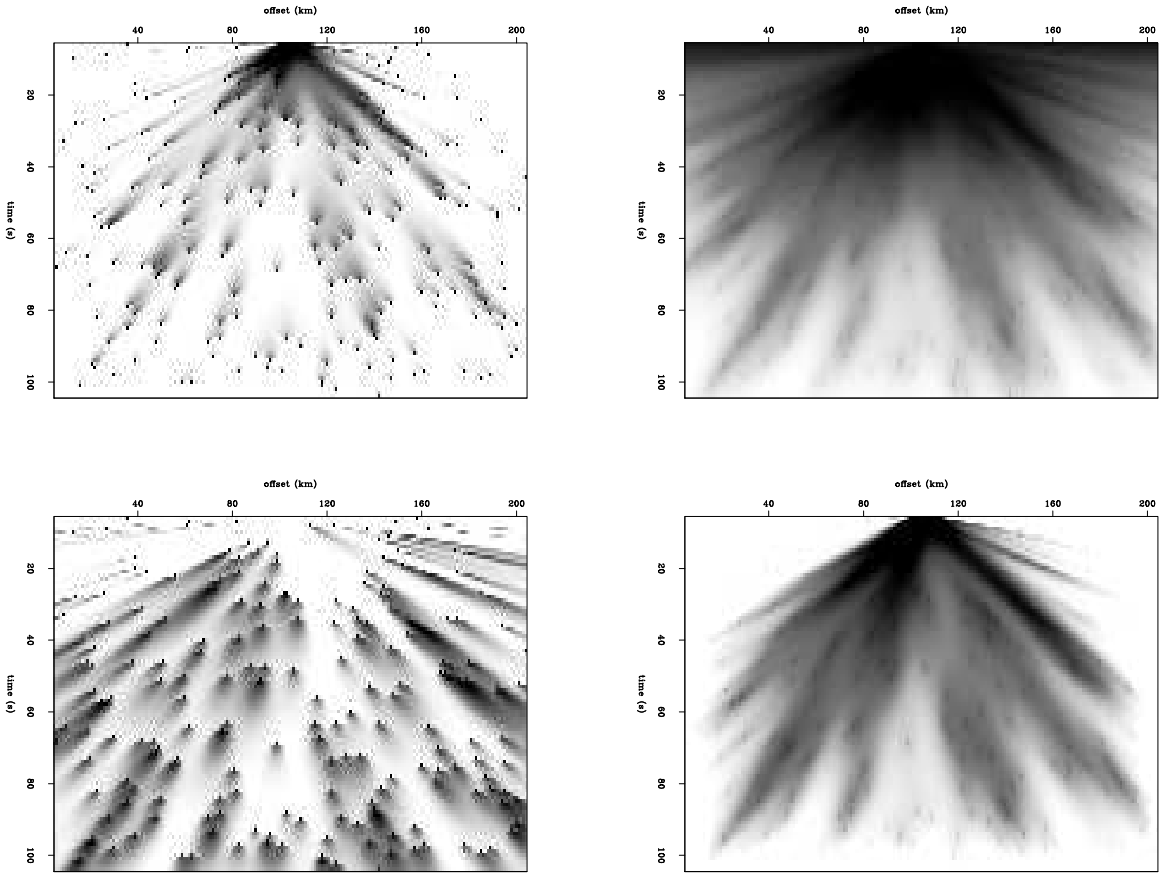

Figure 5: Radial smoothing. Top left panel shows result of smoothing random scattering of dots with the adjoint. Top right panel shows the result of smoothing the top left panel with the forward operator. Bottom left panel shows result of smoothing random dots with the forward. Bottom right show result of smoothing the bottom left with the adjoint. sean2-radial [ER]

applications.

REFERENCES

Abma, R., and Claerbout, J., 1995, Lateral prediction for noise attenuation by t-x and f-x technique s: Geophysics, **60**, no. 6, 1887–1896.

Claerbout, J. F., 1992, Earth Soundings Analysis: Processing Versus Inversion: Blackwell Scientific Publications.

Claerbout, J., 1997a, Multidimensional recursive filters via a helix: SEP–**95**, 1–13.

Claerbout, J. F., 1997b, Geophysical exploration mapping: Environmental soundings image enhancement: Stanford Exploration Project.

Claerbout, J., 1998, Factorization of cross spectra: SEP–**97**, 337–342.

Clapp, R. G., Fomel, S., and Claerbout, J., 1997, Solution steering with space-variant filters: SEP–**95**, 27–42.

Crawley, S., 1998, Shot interpolation for radon multiple suppression: SEP–**97**, 173–182.

Fomel, S., Clapp, R., and Claerbout, J., 1997, Missing data interpolation by recursive filter preconditioning: SEP–**95**, 15–25.