Stanford Exploration Project, Report 100, April 20, 1999, pages 181–196

Interpolation with smoothly nonstationary prediction-error filters

Sean Crawley [1]

keywords: *missing data, interpolation, nonstationary*

ABSTRACT

Building on the notions of time-variable filtering and the helix coordinate system, I develop software for filters that are smoothly variable in multiple dimensions, but that are quantized into large enough regions to be efficient. 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, with the requirement that the patches contain enough data samples to provide a sufficient number of fitting equations to determine all the coefficients of the filter. Each patch of data represents an independent estimation problem. Instead, I estimate a set of smoothly varying filters in much smaller patches, as small as one data sample. They are more work to estimate, but the smoothly varying filters do give more accurate interpolation results than PEFs in independent patches, particularly on complicated data. To control the smoothness of the filters. I use filters like directional derivatives that Clapp et al. (1998) call "steering filters". They destroy dips in easily adjusted directions. I use them in residual space to encourage dips in the specified directions. I describe the notion of "radial-steering filters" (Clapp et al., 1999), i.e., steering 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 I use smoothly variable PEFs to interpolate missing traces, though they may have many other uses. Finally, since noisy data can produce poor interpolation results, I deal with the separation of signal and noise along with missing data.

INTRODUCTION

Claerbout (1998) 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,

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

1998a). To account for nonstationarity in the data, missing data interpolation with PEFs is typically done in patches or gates where dips are assumed to be approximately stationary (**?**Spitz, 1991). Each patch constitutes an independent problem, though they may overlap. The smaller the patch, the more stationary the data is likely to be within the patch; however, there is a lower limit on the patch size, because a patch must contain enough data to provide fitting equations for all the filter coefficients. Claerbout (1997) describes a method for estimating smoothly time-varying PEFs without patching. The helix extends the idea of smooth time-variable PEF estimation to smooth time- and space-variable PEF estimation. The smoothly-variable PEFs can perform better at interpolating missing data than PEFs estimated in independent patches.

Clapp et al. (1998) show how to use space-variable inverse 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 steering. I use space-variable steering filters to control the direction of smoothness between PEFs. I orient the steering 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 I review the theory for estimating smoothly varying PEFs, and show examples of their application to missing data interpolation. I describe an improvement to filter estimation for CMP gathers using "radial-steering filters." Finally, I add the notion of signal and noise separation for interpolating noisy data.

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 it is assumed 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.

I decrease the patch size, to as small as a single data sample, changing the problem from overdetermined to very underdetermined. It is possible to 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. To speed the calculation immensely and make the equations somewhat neater, we can "precondition" the problem. Define a new variable $\mathbf{p}$ by $\mathbf{a} = \mathbf{Sp}$ and insert it into (1) to 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 **RS** by **I** and get

$$\mathbf{0} \approx \mathbf{YKSp} + \mathbf{r}_0 \tag{4}$$
$$\mathbf{0} \approx \epsilon\, \mathbf{Ip} \tag{5}$$

We solve for **p** using conjugate gradients. To see **a**, just use $\mathbf{a} = \mathbf{Sp}$. To simplify things, one could just drop the damping (5) and keep only (4); then to control the null space, start from a zero solution and limit the number of iterations. For **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 later.

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 **K** and an unknown data selector **U**, with $\mathbf{U} + \mathbf{K} = \mathbf{I}$. These multiply by 1 or 0 depending on whether the data were originally recorded or not. With **A** signaling convolution with the PEF and **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 PEF at every sample works well for destroying the data, it is not the best choice for reconstructing it; interpolation with PEFs estimated at every data point gives poor results and requires extravagant memory allocation. One answer is just that zero is not the correct value of ϵ in (5); but we can greatly improve the results and decrease the memory usage without adding equations, by using very small patches, such as $2 \times 2 \times 2$; small enough that the assumption of stationarity within a patch is reasonable. This is similar to putting an extra roughener in the damping equation, in that it is essentially an infinite penalty on variations of **p** between small groups of samples, and it has the important economizing effect of reducing the memory allocation. 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 varying filters effectively reduces the minimum patch size, so that the filter estimation problem can be underdetermined, and still produce useful PEFs.

Radial Smoothing

Clapp et al. (1998) 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. So it makes sense to arrange patches and smooth filter coefficients in the radial direction on a CMP gather, to accelerate convergence and get good results with as few coefficients as possible. Very small patches are desirable where the data has the most curvature, which tends to be at smaller times and offsets. At larger times and offsets, however, events in CMP gathers tend to be near their asymptotes, and much more linear. Smoothing and patching in radial coordinates has the pleasing result that the largest patches fall at far offsets and late times, where they are most appropriate. Figure 1, which comes from Clapp (1999), shows randomly scattered points smoothed with radial-steering filters.

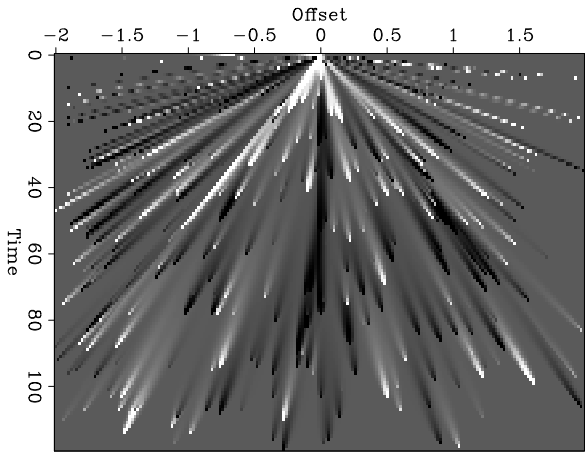

Figure 1: Radial smoothing. Panel shows result of smoothing random scattering of dots with the adjoint radial steering operator. The forward operator points out from the origin. Figure borrowed from Clapp (1999). sean1-bob [NR]

Data Example

Some data from a 3-D marine survey are shown in Figure 2. The top left panel shows part of a shot gather, and the top right panel shows a time slice through several shots from the same source and the same streamer. Marine 3-D data may be broken up into a set of 2-D surveys or treated as a single 3-D survey; more on that later. In this particular case, the data are fairly routine, but they can serve to illustrate the method. The bottom half shows the output of removing half the shots and then interpolating them.

More than two sources

In the previous example I interpolated data collected with two alternating sources. Typical marine geometries have two alternating sources, both placed near the center of the crossline spread, which gives the survey efficient crossline midpoint coverage. It is possible to imagine desiring more than two sources, distributed more or less evenly across the crossline spread. If the geology is complex (salt bodies), and there are genuinely 3-D multiples, then even the best 2-D multiple suppression method will leave multiple events behind. Sources only near the center of the crossline spread prevents the reciprocity necessary for a method like the Delft SRME (van Borselen et al., 1991) method, or multiple prediction by upward continuation, to predict the multiple energy moving in the crossline direction. Using more

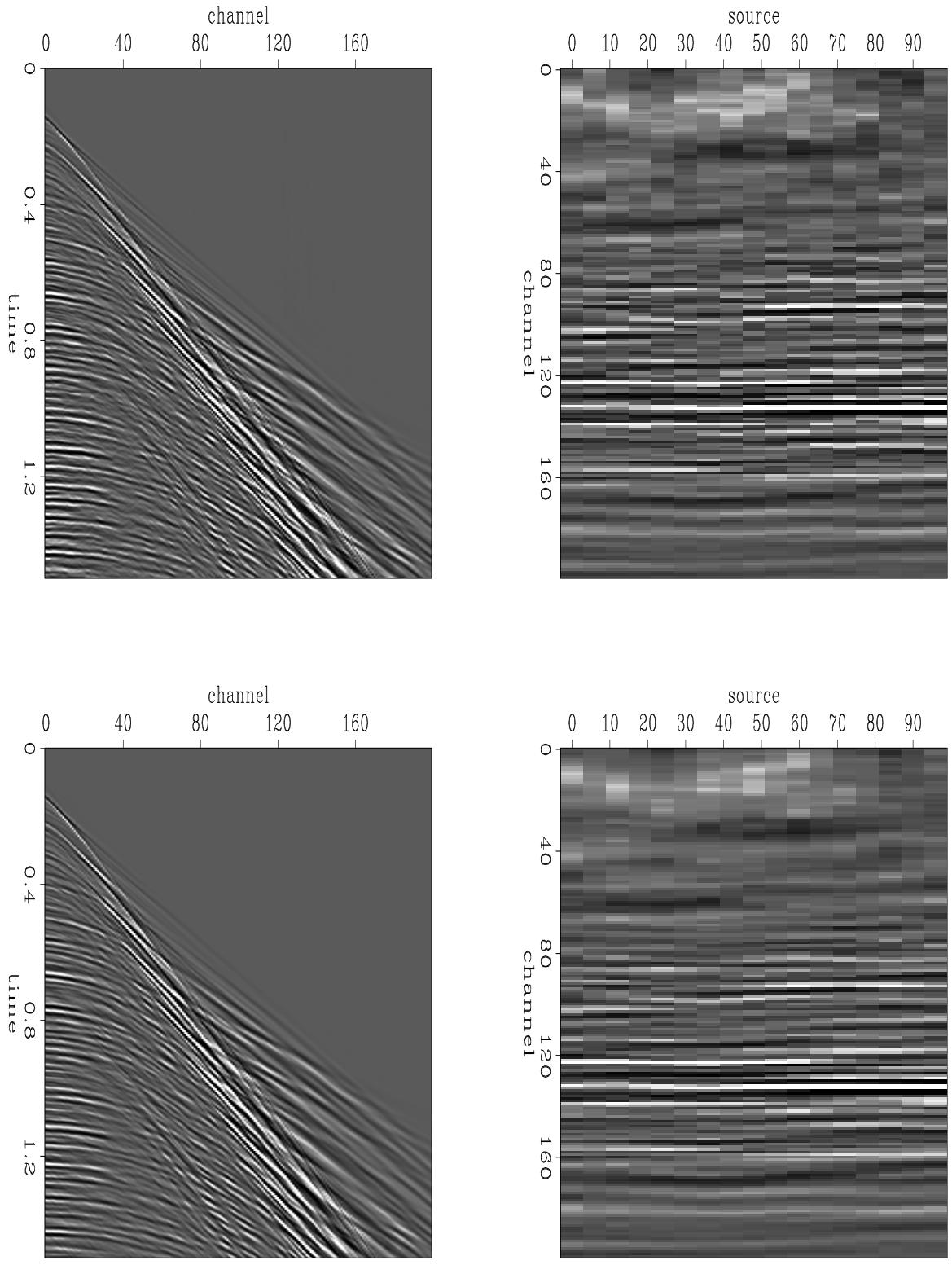

Figure 2: Example input data. The top left panel is part of a shot gather from a 3-D survey, the top right panel is a time slice through several shots from the same source into the same streamer. Bottom panels are the same, after zeroing every other gather and interpolating. sean1-curtFig1 [CR]

sources, spread more widely across the crossline, means the recorded data would contain more crossline information that could be used to model 3-D multiple reflections. On the other hand, cycling through more than two sources naturally creates larger gaps in the acquisition and more opportunities for the data to be aliased. In the previous example, I could throw away lots of data and restore it pretty faithfully, because the data is very predictable. In time slice view, it is almost entirely straight lines. The geology in that example is flat, so it is probably not necessary to think about multiples in three dimensions. The next example is from a marine survey over much more complex geology, where multiple energy moving in three dimensions could be a genuine concern. It is just a 2-D survey; I simulate one inline from a survey with four sources by throwing away three out of four shot gathers. I then attempt to reconstruct the data. Some results are shown in Figures 3, 4, and 5. In this example, the data are reduced to every fourth shot gather and interpolated back to every second shot gather. If that works, then getting from every second shot to every shot is fairly certain. There is no conceptual reason not to go directly from every fourth shot to every shot in a single step, it just means scaling the axes by four rather than two. There is a practical issue that arises, however, in that filter coefficients are smoothed by a variant of leaky integration, and there tends to be a little too much leakage. Figures 3 and 4 show shot gathers that were removed in the left panels, and the interpolated versions of those shot gathers in the right panels. The result are not bad, but not perfect, either. Surprisingly little energy dipping in towards zero offset is lost; given the particular filter smoothing strategy I use, I expected those events to be lost. However, several events close to zero offset are poorly reconstructed. Figure 5 shows a time slice from the correct (recorded) data in the left panel, the input to the interpolation in the middle, and the interpolation on the right. Again the output is not bad, but not flawless either. The spatial frequencies are too low near zero offset, for instance.

2-D or 3-D

For marine data, there is very little difference between interpolating shots in 2-D survey and interpolating shots in a 3-D survey, except that 3-D data provides some interesting choices. We can treat a single source and a single streamer as a 2-D survey (giving us several 3-D input cubes), or separate the sources but leave the streamers together (4-D input), or just use the whole 5-D input. Results get better with more dimensions, because there are more directions for events to be predictable in; but they only get marginally better when we add the crossline directions in marine data, because there are only a handful of crossline points. The cost is large because a few points worth of zero padding are necessary. Padding the inline offset axis by a few points is a small increase in the data volume, but padding the crossline offset axis by a few points may double the data volume. Newer boats tow more streamers, so it may be worth using the extra dimensions on newer data.

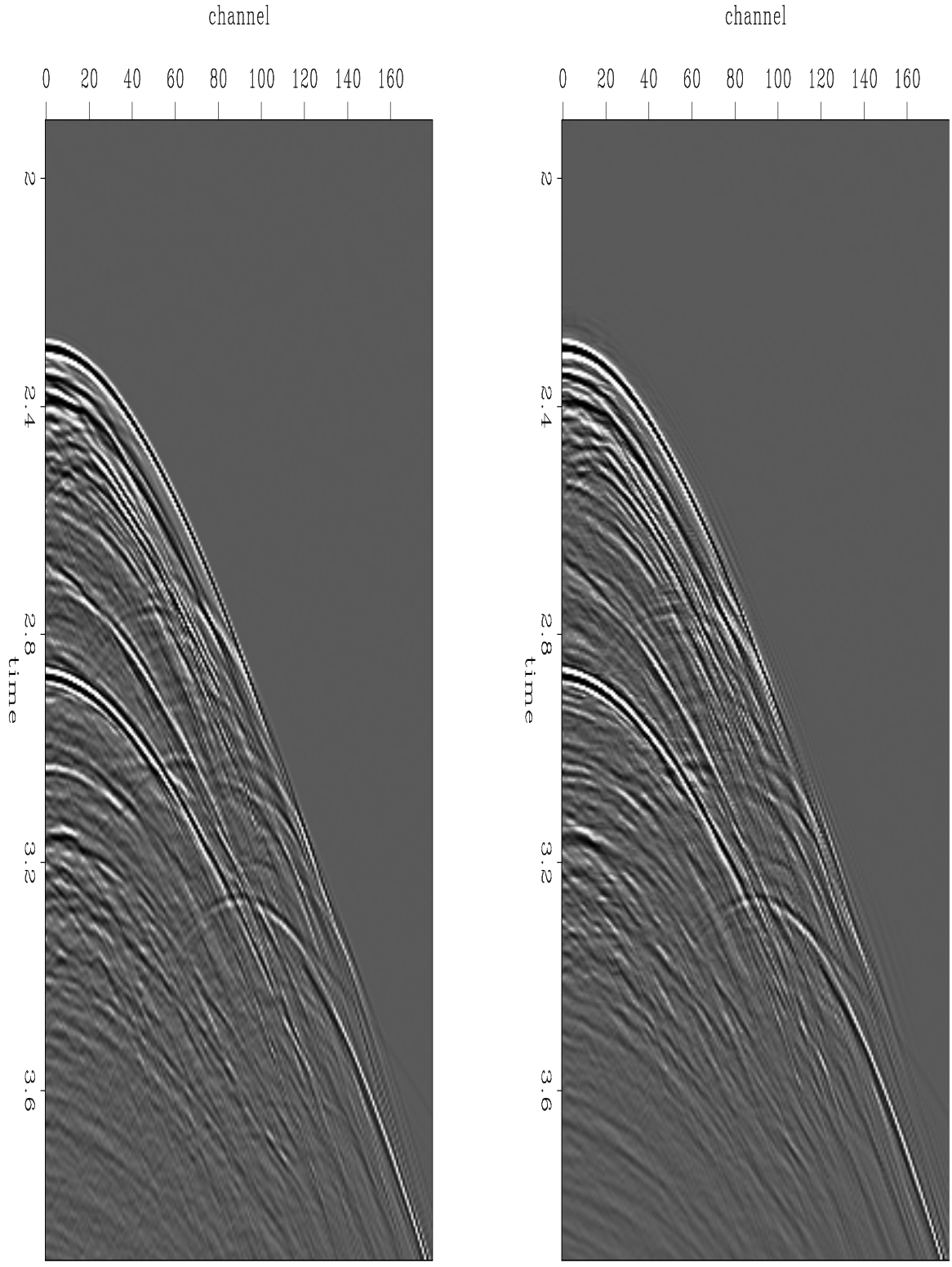

Figure 3: Interpolation test. Left panel is a shot gather that was removed from the input data. Right panel is the interpolated version of that shot gather. sean1-355Fig2b [CR]

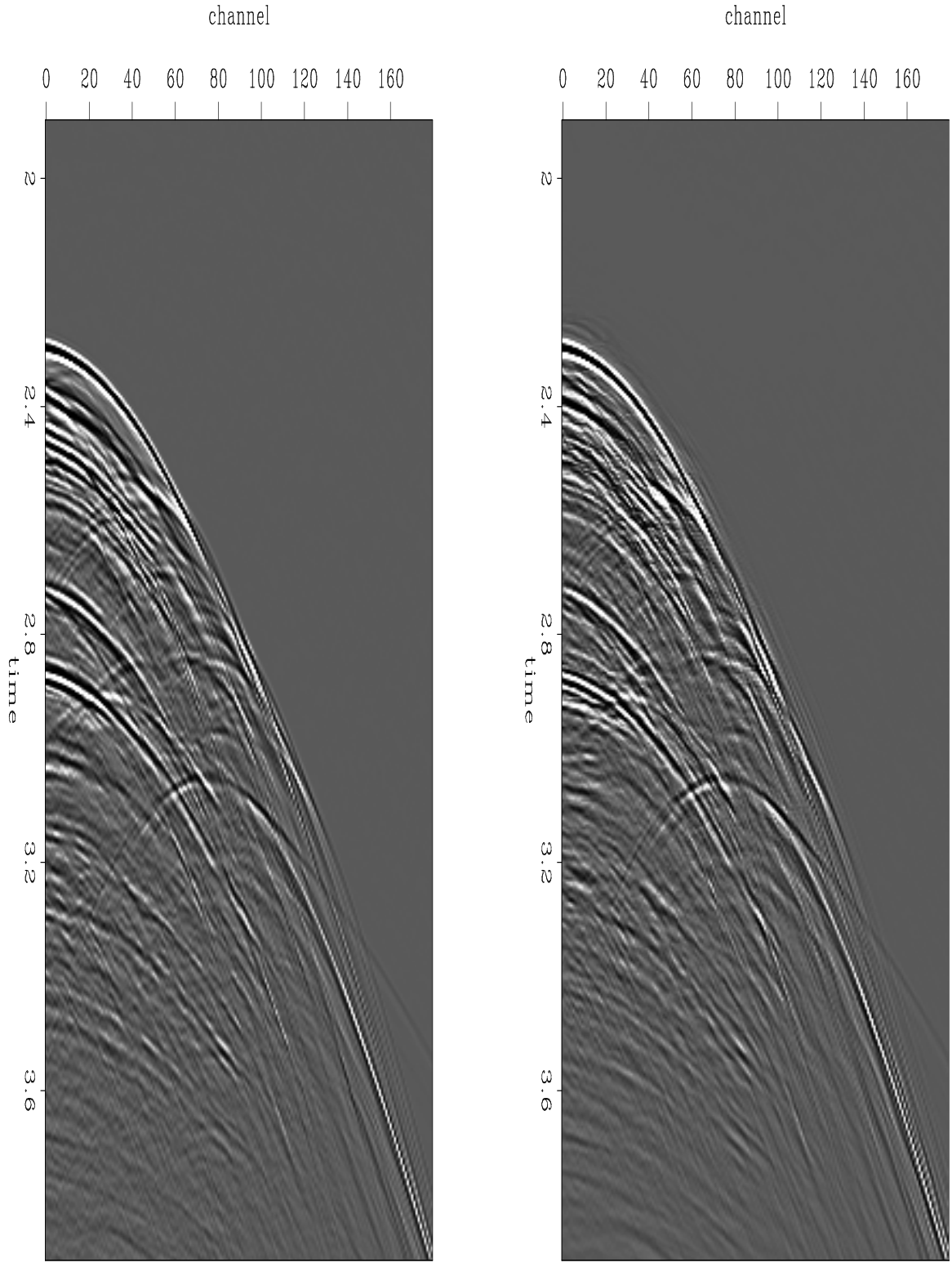

Figure 4: Interpolation test. Left panel is a shot gather that was removed from the input data. Right panel is the interpolated version of that shot gather. sean1-355Fig2 [CR]

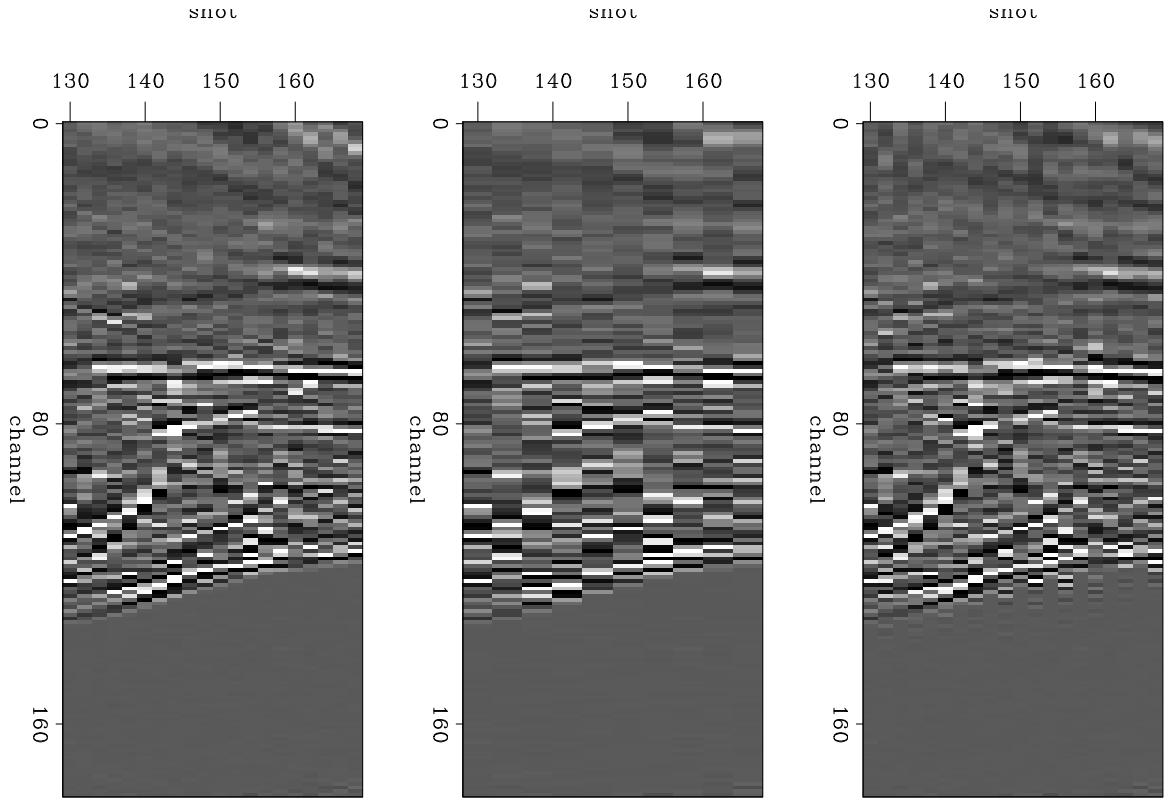

Figure 5: Interpolation test. Left panel shows the correct (recorded) time slice, center panel shows the subsampled input to the interpolation, right panel shows the output. sean1-355Fig1 [CR]

NOISY DATA

It is appealing to think of interpolating land data, because land data can be so expensive to acquire. Land data is more difficult to interpolate than marine data, however, because it tends to be much more noisy. We can guess that the PEFs will find the predictable part of the data, and that it will be the signal. However, even assuming we do estimate the correct PEF (one that captures the dips of the signal rather than the noise), energy from the noise will be carried along those dips to nearby interpolated traces. As an alternative, we can attempt to separate the signal from the noise while interpolating the missing traces. Taking the theory from Claerbout (1997), data $\mathbf{d}$ can be decomposed into known plus missing parts, $\mathbf{d} = \mathbf{k} + \mathbf{m}$. With known and unknown data selectors $\mathbf{K}$ and $\mathbf{M}$, write the data as

$$\mathbf{d} = \mathbf{Kd} + \mathbf{Mm} \tag{6}$$

where $\mathbf{Kd}$ is zero-padded known data and all the components of $\mathbf{m}$ are freely adjustable. The goal is to fill in the appropriate parts of $\mathbf{m}$.

The data can also be decomposed into signal plus noise, $\mathbf{d} = \mathbf{s} + \mathbf{n}$. Thus

$$\mathbf{s} + \mathbf{n} = \mathbf{Kd} + \mathbf{Mm} \tag{7}$$

If we write fitting goals for signal and noise and then get rid of the noise with equation (7) we have

$$\mathbf{0} \approx \mathbf{Nn} = \mathbf{N}(\mathbf{Kd} + \mathbf{Mm} - \mathbf{s}) \tag{8}$$

$$\mathbf{0} \approx \mathbf{Ss} = \mathbf{Ss} \tag{9}$$

Solving (8) and (9) gives estimates for the signal component $\mathbf{s}$ of both the known and missing data, as well as an estimate of the noisy missing traces, $\mathbf{Mm}$. This requires a signal predictor and a noise predictor. For the signal predictor I use the PEFs estimated from the data. In the example to follow I will throw out alternating midpoint gathers and attempt to interpolate them back, so I define noise as whatever is incoherent across midpoints and choose the noise predictor to be an average in a small window along the midpoint axis.

Example

Figure 6 shows a CMP gather from a land seismic survey, and a time slice through several CMP gathers. There is a visible "noise cone" defined by some low velocity. Inside the cone there is very little coherency along the midpoint axis. As an experiment with noisy data interpolation, I reduced the data to every second CMP, then performed NMO and stack to produce the top left panel of Figure 7. Then I further reduced the data to every fourth CMP and interpolated with and without noise estimation, and again did NMO and stack to try to reproduce the stack in the top left of Figure 7. The top right panel, the result without any noise estimation, has significant problems with reflector continuity. In particular, strong reflectors in the center of the section, at 1.2 seconds and at .9 seconds, have undesirable gaps

in the midpoint direction. The bottom left stack, which is really a stack of the interpolated and extracted signal $\mathbf{s}$, is a much more coherent stack, but is slightly lower in temporal frequency content. The bottom right stack is the same, after matched filtering to restore the temporal spectrum. The spectra of the original (top left) stack and the interpolated stack without matched filtering (bottom left) are shown in the left side of Figure 9. After matched filtering, the interpolated, noise suppressed stack has the same temporal frequencies as the stack of the original data; without the filtering step, it is boosted at low frequencies and suppressed at high frequencies. The right side shows the spectra of the original (top left) stack and the interpolated, matched filtered stack (bottom right). The effects of signal and noise separation are shown in Figure 8. The left panel shows the signal component estimated from one of the input CMP gathers, the right shows signal component for an interpolated CMP gather.

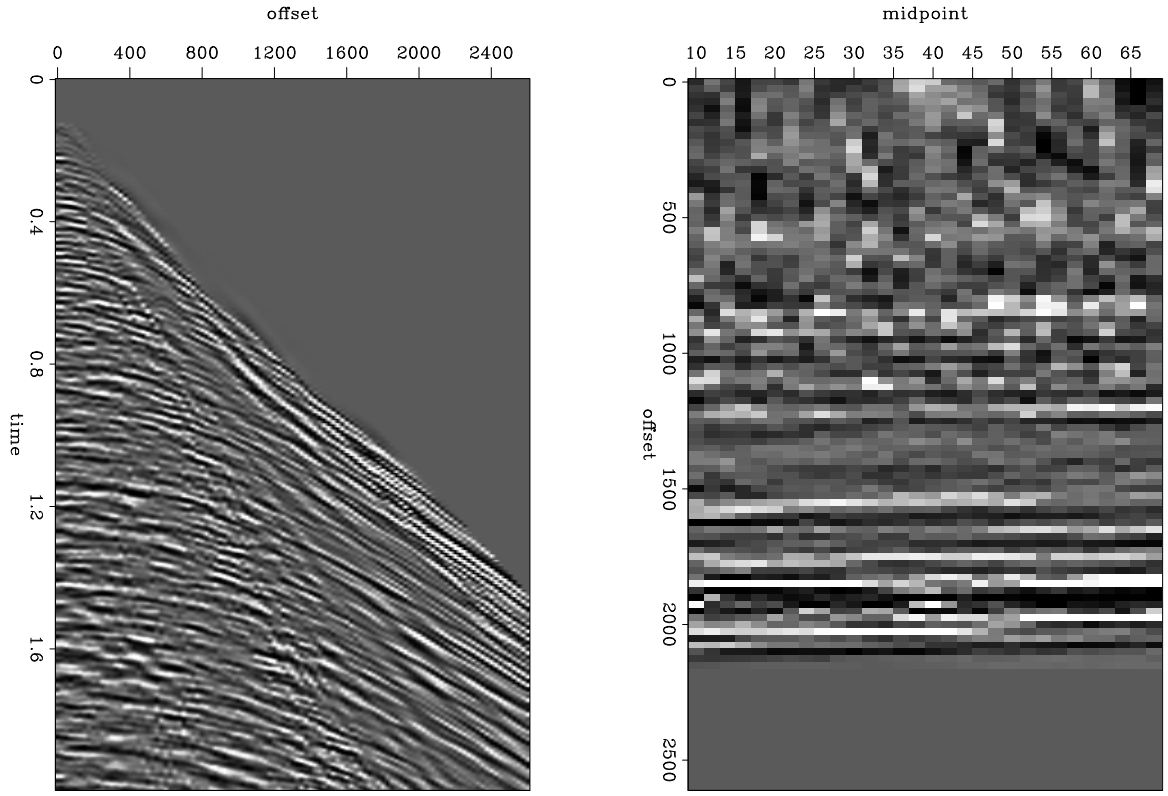

Figure 6: Land seismic data. Left panel shows a CMP gather, right shows a time slice through several CMP gathers. The data are incoherent along the midpoint axis, close to zero offset. sean1-landCmp [CR]

CONCLUSIONS

I describe a method for interpolating missing data with smoothly varying PEFs. Forcing the PEFs to vary smoothly enables to us to use very small patches, even a single data sample,

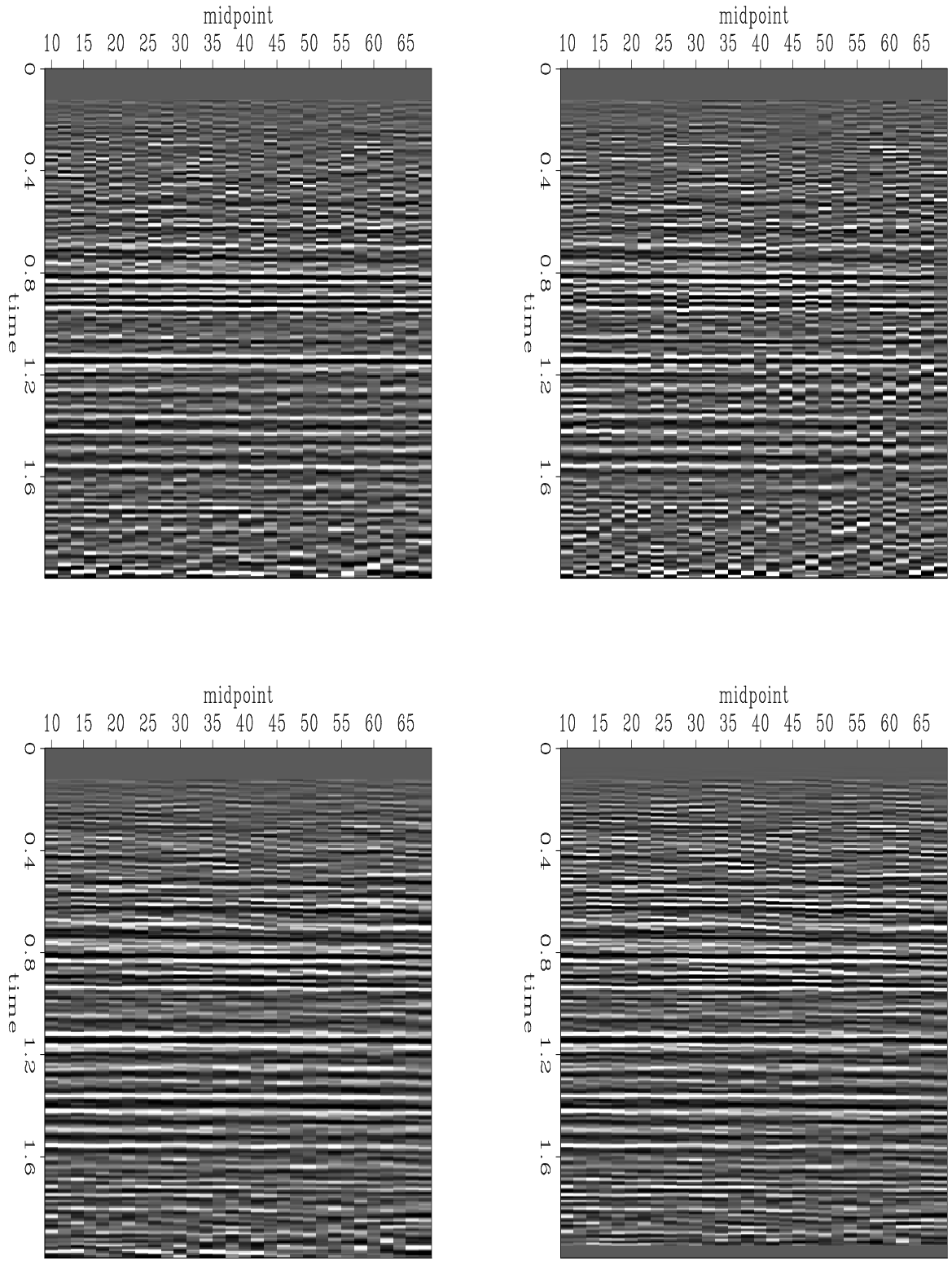

Figure 7: Land data interpolation. Top left panel is a stack of the known data. Top right panel is the stack after throwing away half the CMP gathers and reinterpolating them without attempting to compensate for noise. Bottom left is the stack of the signal component extracted from the known data and interpolated into the missing data. Bottom right is the same stack after matched filtering to get back the temporal frequency content of the known data. sean1-stacks [CR]

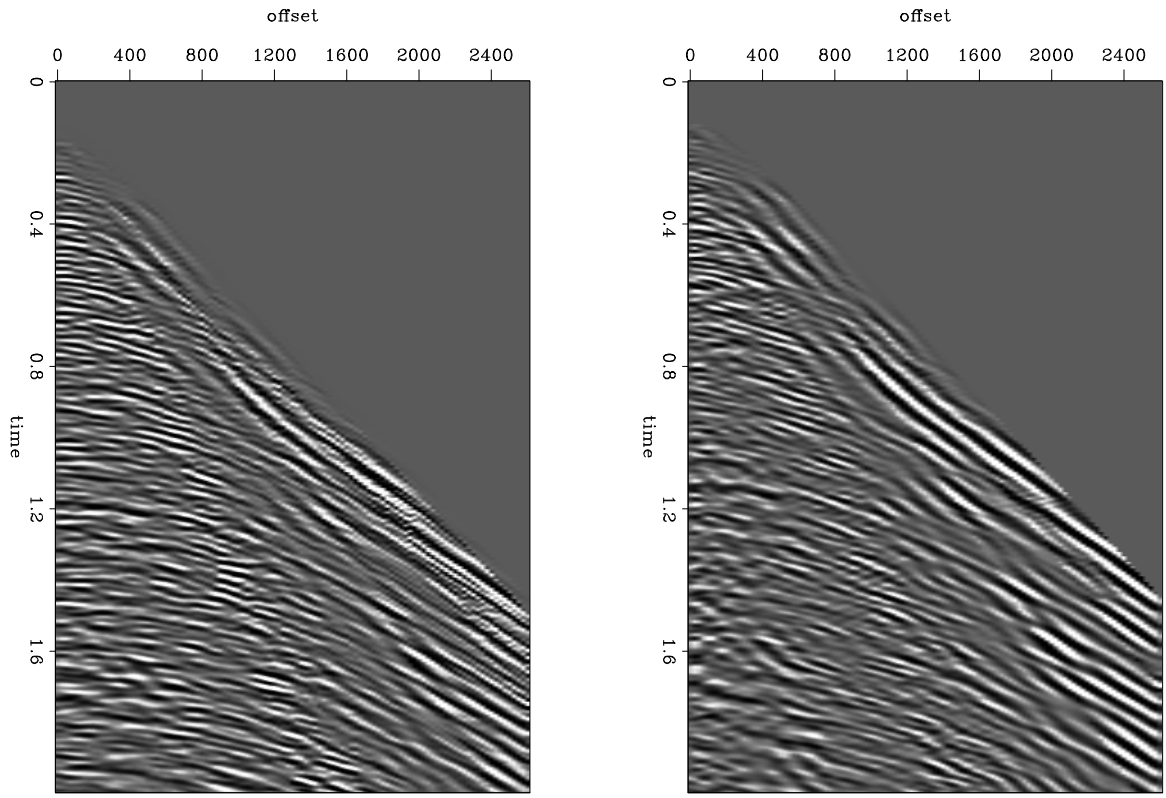

Figure 8: Interpolation and noise removal. Left side shows the estimated signal component of an input CMP gather, right side shows the signal component of an interpolated gather. sean1-outCmp [CR]

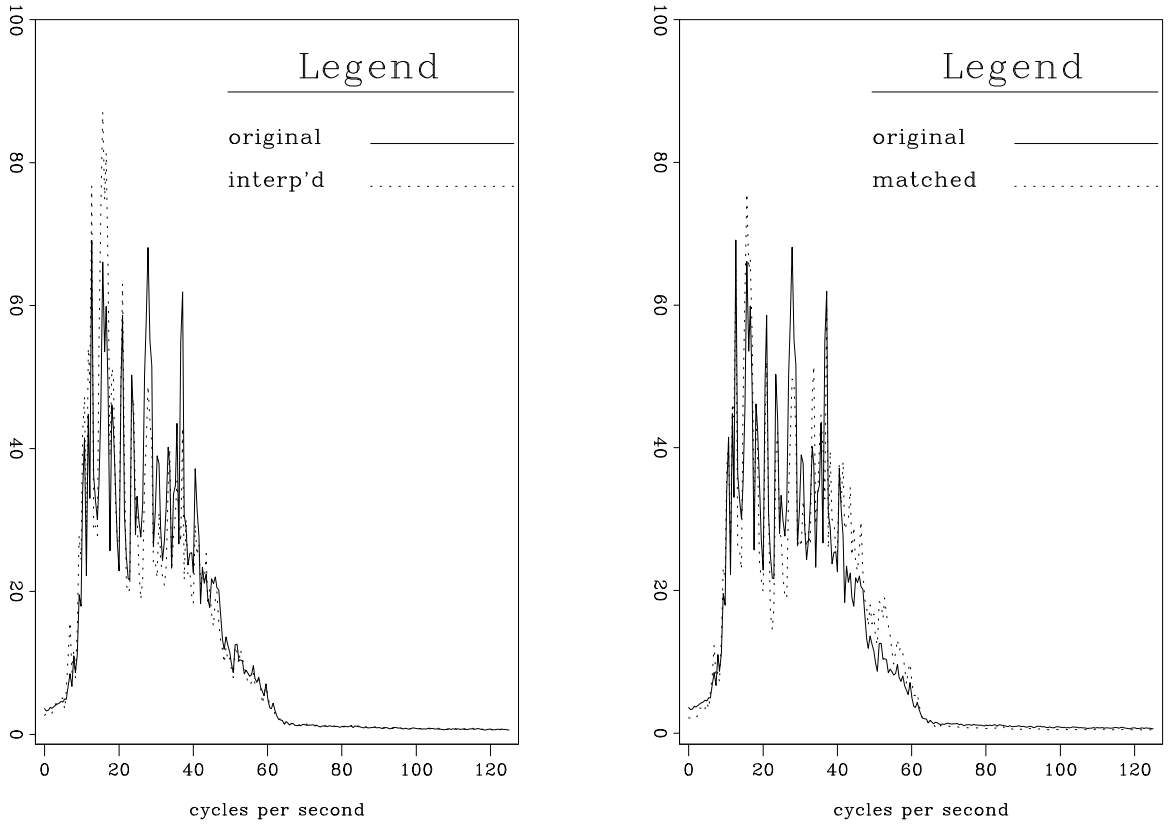

Figure 9: Spectra of stacked data. Left panel shows spectra of the original stack and the stack of the interpolated data, without matched filtering. Right panel shows spectra of the original stack and the matched filtered stack. Solid lines represent original data. sean1-specs [CR]

and produces good interpolation results on complicated data. I also describe a method for separating signal from noise and interpolating just the signal.

ACKNOWLEDGEMENTS

Thanks go to Bob Clapp, Sergey Fomel, and Jon Claerbout for numerous useful discussions.

REFERENCES

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

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

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

Claerbout, J. F., 1998b, Multi-dimensional recursive filtering via the helix: Geophysics, **63**, no. 5, 1532–1541.

Clapp, R. G., Fomel, S., Crawley, S., and Claerbout, J. F., 1999, Directional smoothing of non-stationary filters: SEP–**100**, 197–210.

Clapp, R. G., 1998, Regularizing velocity estimation using geologic dip information: 68th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1851–1854.

Crawley, S., 1998, Shot interpolation for radon multiple suppression: 68th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1238–1241.

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

Spitz, S., 1991, Seismic trace interpolation in the f-x domain: Geophysics, **56**, no. 6, 785–794.

van Borselen, R. G., Thorbecke, J., Fokkema, J. T., and van den Berg, P. M., 1991, Surface related multiple elimination based on reciprocity: Surface related multiple elimination based on reciprocity:, 61st Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1339–1342.