CHAPTER 2

Surface-consistent deconvolution

2.1 OVERVIEW

Multi-trace deconvolution, and surface-consistent deconvolution in particular, has three major uses:

1. *Noise reduction* The reliability of filter estimates is increased by using more statistics. By using the redundancy of multichannel seismic data in deconvolution, we combat noise much as we do in stacking. The surface-consistent model tells us where to find the additional statistics. It also tells us what effects we have to unravel to get at these statistics.

2. *Statics estimation* A key assumption behind statics picking is that the cross-correlation of two reflection events is maximized when the events are aligned. Unless the events have similar wavelets attached to them, this will not be the case. Deconvolution usually acts to balance the spectra of seismic traces, improving the similarity of these wavelets. A drawback of deconvolution is that it will usually shift the location of the wavelet at the same time. This can be a problem for surface-consistent statics decomposition, which assumes the trace-to-trace shifts fit a surface-consistent model. By using surface-consistent deconvolution, we obtain spectral balancing without sacrificing the fit of our statics to a surface-consistent model.

3. *Amplitude extraction* Tracking amplitude as a function of shot-to-receiver offset is useful for weeding out strong reflections that are not due to trapped hydrocarbons. Surface-consistent amplitude balancing is used to compensate for amplitude effects that aren't due to changes in reflection coefficients with angle of incidence. This is best done after spectral balancing, i.e. deconvolution. The pitfall is that deconvolution changes the energy of seismic traces. Reductions by 90% are not uncommon for spiking deconvolution. Usually, this is compensated by rescaling each trace after deconvolution to match its energy before deconvolution. Such rebalancing can easily destroy relative amplitudes, however. A solution is to balance spectra with surface-consistent deconvolution in order to keep amplitudes corrections surface-consistent.

In this chapter, I develop a good method for surface-consistent deconvolution that avoids transformation to the log-spectral domain. In my approach, I design deconvolution filters directly in the time-space domain.

I avoid the log-frequency domain for several reasons. The log transformation distorts the statistics of additive noise. The heaviest weighting occurs where the signal spectrum approaches zero; these gaps should have the least influence on filter estimates. These spectral gaps also degrade phase unwrapping, a critical component when phase is not assumed a priori, such as in homomorphic deconvolution. Additionally, smoothing or other constraints must be imposed on the frequency spectrum in order to

control the temporal duration of the estimated filters. The problem cannot simply be solved one frequency at a time; the original appeal of the log-frequency transform is then lost.

A more fundamental problem with the log-frequency transform is that it simplifies only the purely one-dimensional convolutional model. Two-dimensional effects such as array response or wave propagation cannot be incorporated cleanly into such processing; nor can nonconvolutional effects such as attenuation and, as noted earlier, additive noise.

For these reasons I choose to work directly in the time-space domain in designing inverse filters for surface-consistent deconvolution. I find that the straightforward method of repeated gather-by-gather deconvolution runs into both technical and philosophical problems. I employ methods for solving large, sparse least-squares systems to sort out the difficulties and develop a stable and reasonably efficient surface-consistent analogue of conventional prediction-error filtering. Here, not surprisingly, the chief operations are convolution and correlation. Applying it to field data from the San Joaquin Valley, I obtain results about the same as conventional single-trace deconvolution. This is independent evidence that the surface-consistent model is an apt approximation for this seismic data. I also apply surface-consistent deconvolution to the suppression of marine multiples on a Barents Sea profile recorded over a hard water-bottom. I find, for reasons which I will explain, that the surface-consistent model does not fit these data, contrary to theoretical arguments that have appeared in the literature.

2.2 REPETITION TESTS

Surface-consistent statics solutions are computed iteratively by repeated spatial averaging over common-source, common-receiver, and common-midpoint gathers until convergence is reached. The result satisfies a repetition test: no better answer is obtained by further iteration. For a process to satisfy such a test is both useful and reassuring. If a good result can be obtained, it is obtained; a poor result will not be improved by further iteration. Ideally, we would like our processing to satisfy an even stronger test: no better answer is obtained by reprocessing with a new choice of parameters. In practice, this is unobtainable; the best we can do is to reduce the number of parameters the user needs to select. This is the appeal of estimating residual statics by stack power optimization (Ronen and Claerbout, 1985): a separate time-shift picking step requiring half a dozen user-specified tolerances and thresholds is avoided.

Ideally, deconvolution should also satisfy a repetition test. Claerbout (1984) argues this is especially important for nonlinear, iterative deconvolution schemes. He proposes a stringent repetition test:

> "Take the output of a deconvolution process and use it for the input of a second iteration of the same deconvolution process. The second output should be the same as the first output."

This is too strong a requirement. It would force us to abandon even single-trace predictive deconvolution. What predictive deconvolution does satisfy is an optimization test: the inverse filters it produces really do minimize the prediction error as measured by the same least-squares criterion from which the method is derived. This is a sensible requirement to impose on surface-consistent prediction-error filtering, one that I will show how to attain at reasonable cost.

Prediction-error filtering is not the only deconvolution method that is based on optimization. The maximum-entropy method (Burg, 1975) minimizes the sum of forward and backward prediction error for its reflection coefficient estimates. The minimum-entropy (Wiggins, 1978) and variable-norm (Gray, 1979) deconvolutions minimize some measure of spikiness of the output. While I do not study these here, one should require their surface-consistent extensions to satisfy the optimization test too.

Iterative deconvolution

For deconvolution, an analogue of the surface-consistent statics solution used in practice (e.g. Newman, 1986) is to perform one or more passes over the data deconvolving individual gathers and switching gather types, e.g. common-shot to common-receiver, between iterations. Usually, the iterations are kept to one or two to both limit the cascaded filter length and save computer time. To satisfy Claerbout's repetition test we should iterate until convergence. Here I show by example that neither of these methods can be relied upon in practice to give a good deconvolution.

In Figures 2.1 through 2.3, I show the results of repeated iteration on a seismic line from the Central Valley of California. For these tests I adapted a single trace Wiener-Levinson predictive deconvolution program to process each gather by averaging the autocorrelations of the individual traces, normalized to have zero lag 1, and designing an inverse filter for the average. This filter is convolved with each trace of the gather to produce the deconvolved output. In the early iterations, a stack of the gathers after one pass over shots and receivers (Figure 2.2) looks very similar to the stack of the input (Figure 2.1). Further iteration produces a noiser, less appealing stacked image and after a few more passes over the survey (Figures 2.3) the stack becomes simply awful. None of the results comes anywhere near the quality of the output of single-trace predictive deconvolution (Figure 2.4.) This is not a failure of the surface-consistent model. In the next section I will produce a surface-consistent deconvolution of these data that is comparable to single-trace deconvolution. The fault lies in assuming simple (unweighted) averaging is sufficient to cancel out the "gather-inconsistent" effects in the individual traces.

In summary, I've shown that a good surface-consistent deconvolution cannot be expected if only gather-by-gather deconvolutions are cascaded. If we anticipate that the convolutional effects on our data are mostly due to surface-consistent effects, we

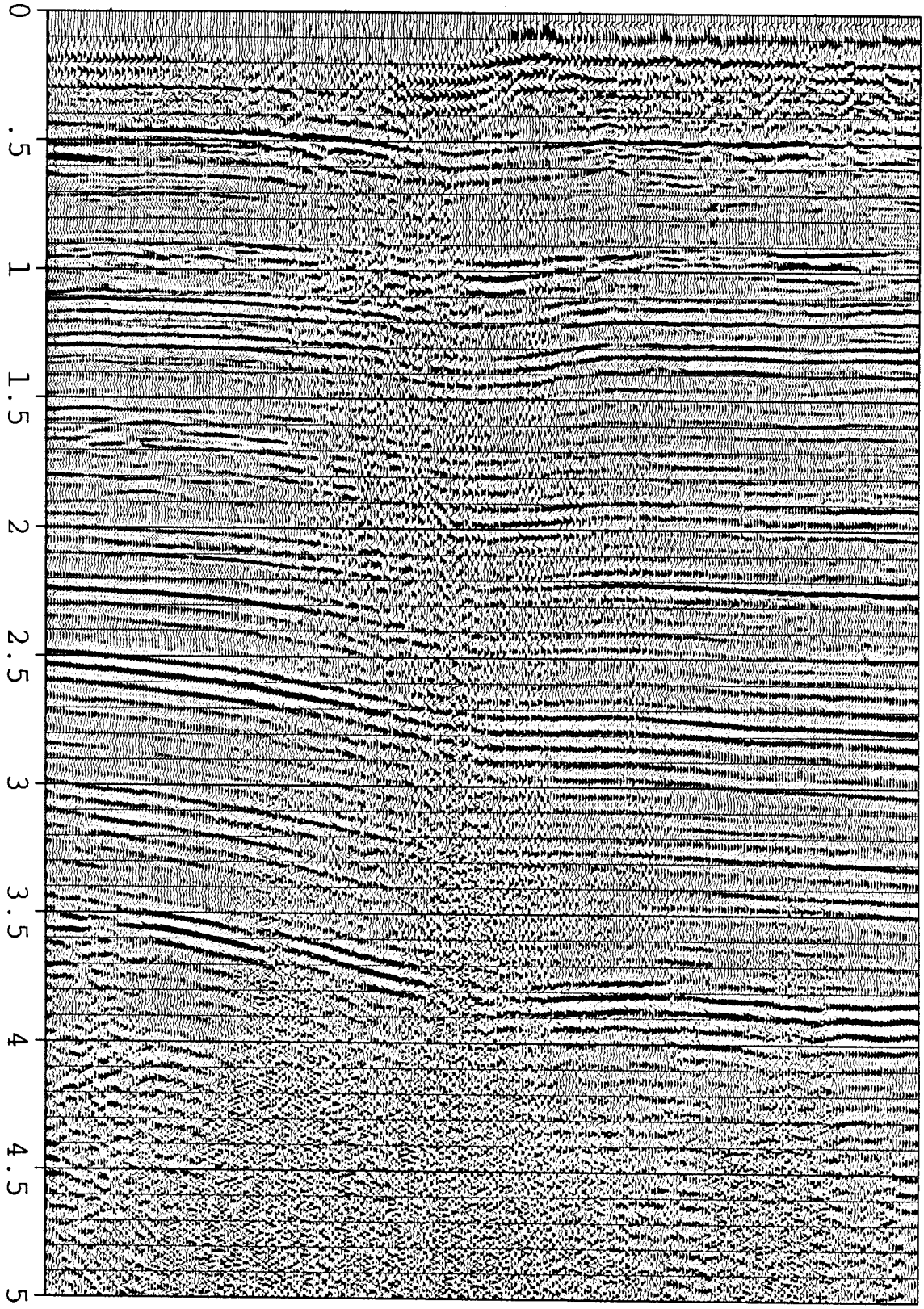


FIG. 2.1. Stack of undeconvolved input to surface-consistent deconvolution. There are 275 28-fold common midpoint (CMP) gathers in this dataset. Trace length is 5 seconds sampled at 4 msec. These data are from the Central Valley of California. The sag in the center of the section is due to a low-velocity near-surface anomaly.

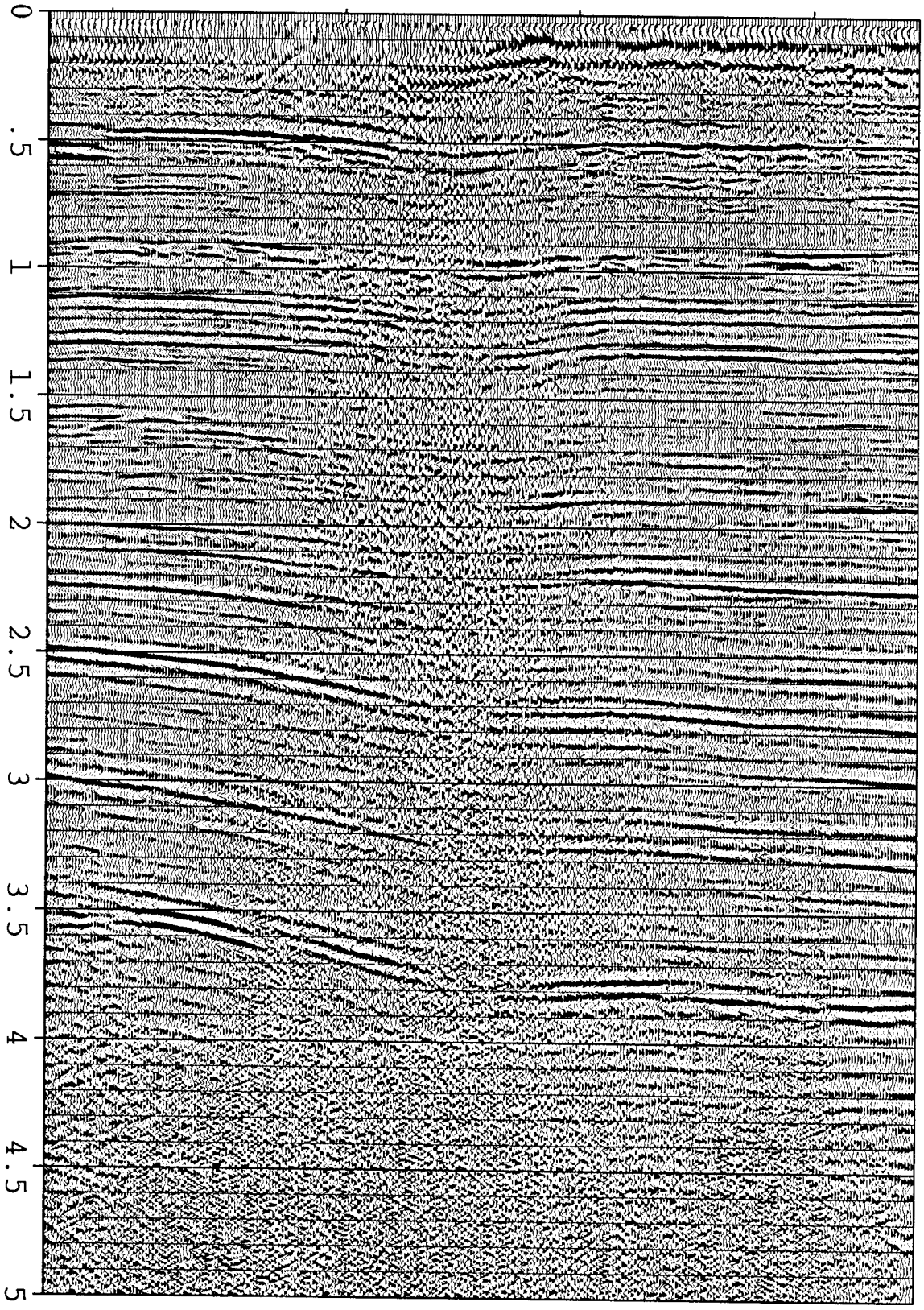


FIG. 2.2. Stack after one iteration through common-shot gathers and common-geophone gathers. This is one industry method for surface-consistent deconvolution. In this example, the shallow reflectors have been broken up and there has been only a small amount of wavelet compression on the deeper horizons.

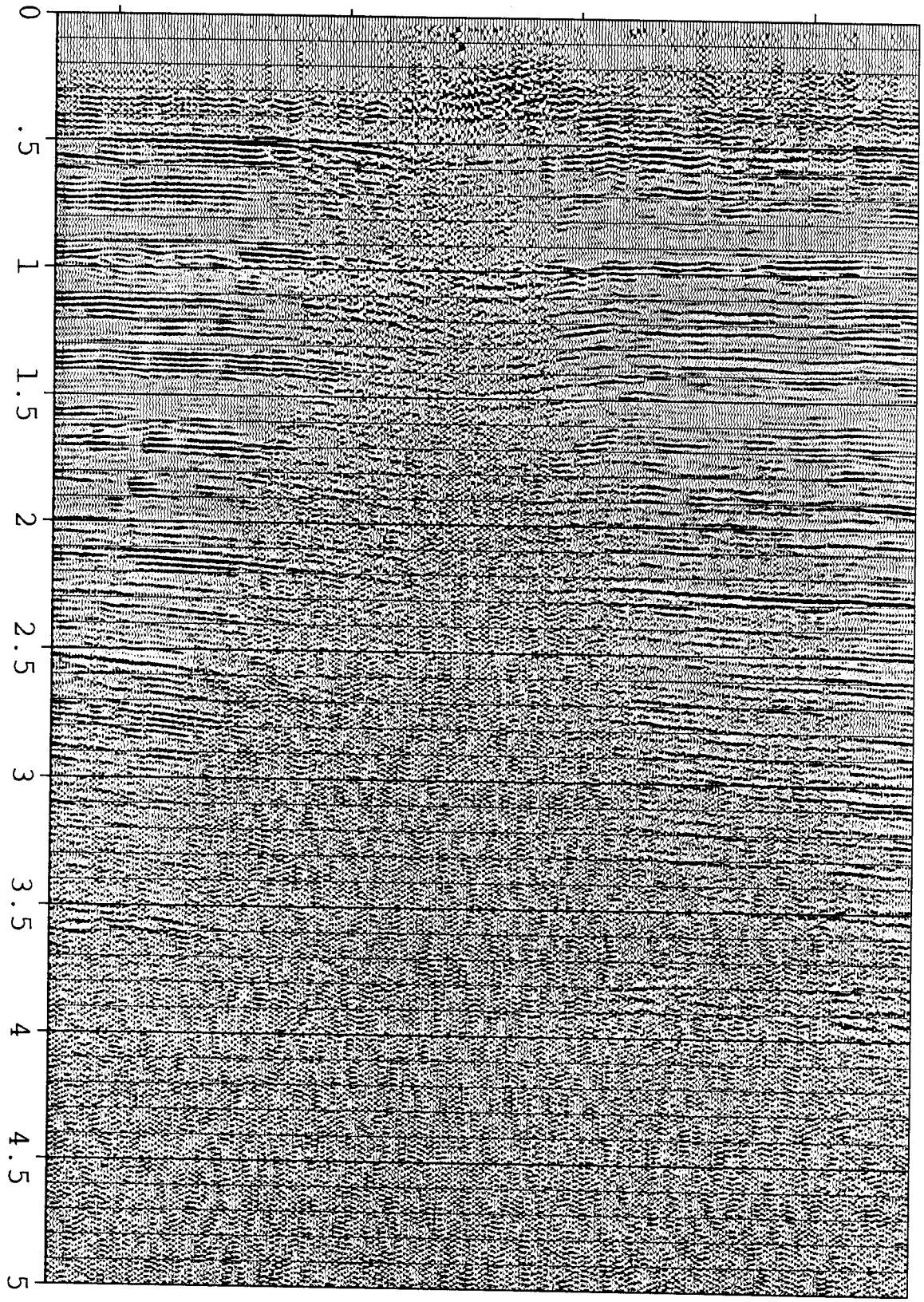


FIG. 2.3. Stack after six iterations of "surface-consistent" gather-by-gather deconvolution on the data of Figure 2.1. Almost nothing interpretable remains on the stack.

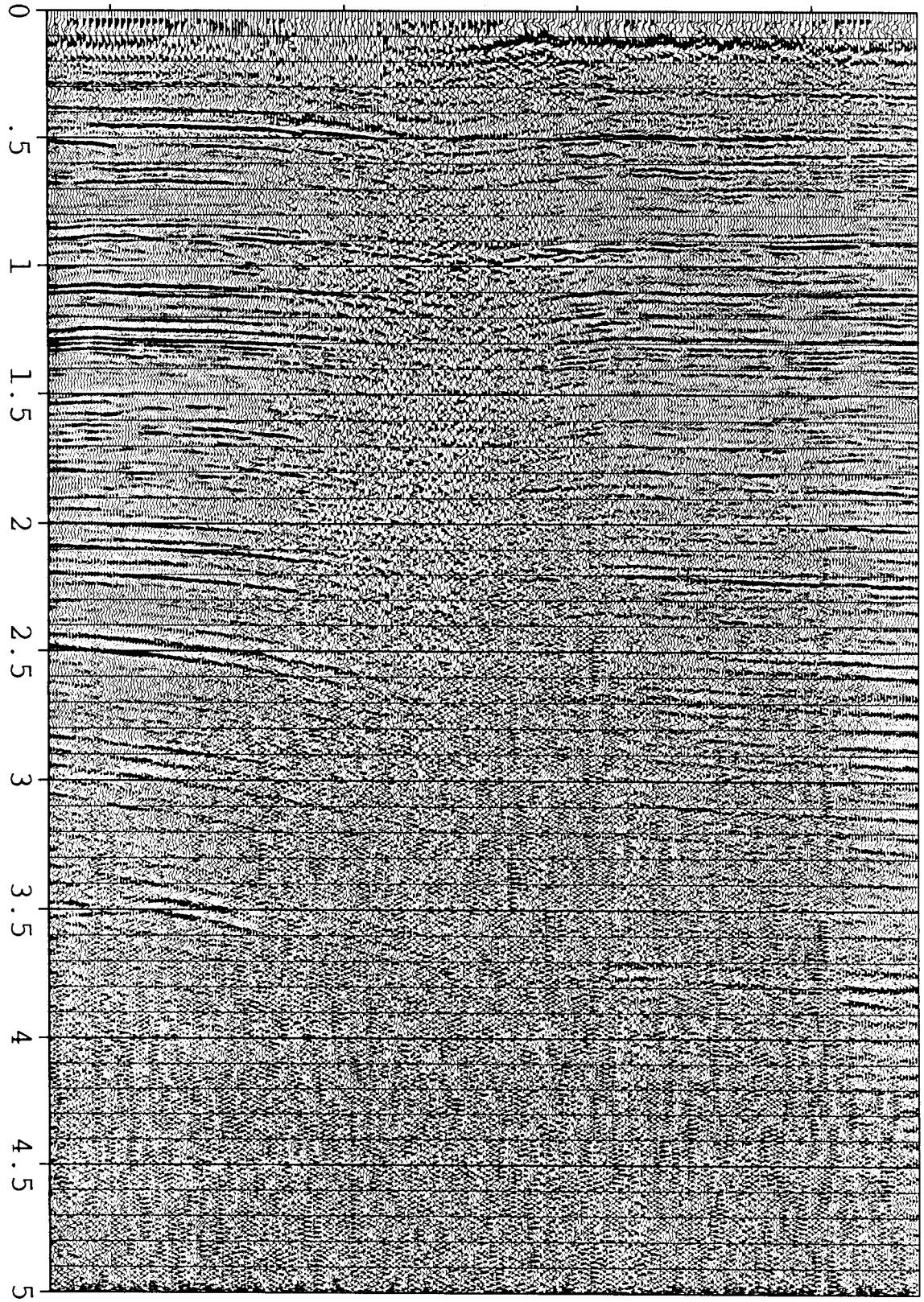


FIG. 2.4. Stack of original gathers after conventional single-trace spiking deconvolution. The deconvolution has done a good job despite the fact that the source was Vibroseis®.

®Trade and service mark of Conoco, Inc.

should demand that surface-consistent deconvolution produce results at least comparable to single-trace (surface-inconsistent) deconvolution. In the next section I pose the problem of designing surface-consistent deconvolution filters as an optimization problem. Applying methods for solving large, least-squares systems, I obtain a proper gather-by-gather iteration that produces the required deconvolution filters simultaneously.

2.3 LINEARIZATIONS FOR SIMULTANEOUS FILTER DESIGN

The previous section argues for simultaneous rather than cascaded surface-consistent design of deconvolution filters. A similar conclusion is reached by Claerbout (1986) for the problem of deconvolving the source signature before normal moveout correction without interfering with multiple reverberations, which should be handled after divergence correction and normal moveout. His solution is to design debubble and dereverberation filters simultaneously by linearizing the combined process about some starting guess and using least-squares to estimate perturbations that reduce prediction error. I too will apply nonlinear least-squares methods to design surface-consistent filters simultaneously; however, I will not incorporate moveout corrections into the process. While it is not difficult to do so, this is not a promising avenue. First, in land acquisition the near-surface is weathered and has a very low acoustic velocity. Reverberations are often generated within this zone, say between the surface and the water table, which have raypaths that are nearly vertical and have a periodicity nearly independent of the angle of deflection of the rays below the weathered zone. Second, there is the velocity dilemma – overlapping of primary events and reverberation trains supplies us with conflicting choices for moveout velocity. Indeed this is a motivation for Claerbout's later work (1986) that advocates replacing normal-moveout correction with wave extrapolation.

Following Dennis (1977), let me briefly review some classic techniques for nonlinear least-squares to minimize a sum of squares of the form

$$1/2 \quad || f \; ||^2 \quad , \tag{2.1}$$

for a vector function f . Equating its derivative to zero produces

$$J^T f \;\; = \; 0 \quad , \tag{2.2}$$

where J is the matrix of partial derivatives of f . The second derivative of the sum of squares may be written

$$J^T J + K \tag{2.3}$$

where

$$K \;\; = \; \sum_i f_i \nabla^2 f_i \quad . \tag{2.4}$$

When f is a linear function, $\nabla^2 f$ is identically zero. Thus K is a measure of the

departure from linearity. The Newton equations

$$(J^T J + K)\,\delta x \;=\; -J^T f \tag{2.5}$$

determine the location of a zero of the quadratic approximation to the sum of squares obtained by truncating a Taylor expansion at second order.

In the Gauss-Newton method one assumes that K is small compared to $J^T J$, i.e., that f is locally linear, and looks instead at the equations

$$J^T J\;\delta x \;=\; -J^T f \tag{2.6}$$

which are the normal equations for the classic linear least-squares problem

$$J\;\delta x \;\approx\; -f \qquad . \tag{2.7}$$

A myriad of numerical methods exist for handling equations (2.6) or (2.7). In this thesis, I used two canned conjugate-gradient subroutines from Paige and Saunders (1975, 1982) – SYMMLQ for which deals with $J^T J$ in the normal equations (2.6) and LSQR which works directly with J in the least-squares system (2.7).

Deconvolution methods based upon the principle of minimum prediction-error fall into this least-squares framework. The following three examples make this concrete and lead naturally into surface-consistent prediction-error filtering, the goal of this chapter.

Single-channel deconvolution

Here the vector f is given by

$$f \;=\; d \;-\; \mathbf{SH}(d)\,b \tag{2.8}$$

where d is the data trace, b is the unknown prediction filter, and $\mathbf{SH}(d)$ is a matrix of shifted copies of d:

$$\begin{pmatrix}
0 & 0 & 0 & . \\
d_0 & 0 & 0 & . \\
d_1 & d_0 & 0 & . \\
d_2 & d_1 & d_0 & . \\
. & d_2 & d_1 & . \\
. & . & d_2 & . \\
. & . & . & . \\
0 & . & . & . \\
0 & 0 & . & . \\
0 & 0 & 0 & .
\end{pmatrix}$$

This matrix representation emphasizes that the data traces are the known quantities and the filter is unknown. In other applications the filter can be the known quantity and the time series the unknown. Then it is more useful to represent convolution as $\mathbf{SH}(b)\,d$.

The gradient equation (2.2) may be written

$$0 \;=\; -\mathbf{SH}(d)^T \;(\; d \;-\; \mathbf{SH}(d)b \;) \tag{2.9}$$

which are the normal equations for the least squares problem

$$d \;\approx\; \mathbf{SH}(d)b \qquad . \tag{2.10}$$

The second derivative Hessian matrix is

$$\mathbf{SH}(d)^T \;\mathbf{SH}(d) \tag{2.11}$$

which is $J^T\; J$; K is zero because this is a linear problem. From (2.11) we see the Newton step (2.5) satisfies

$$\mathbf{SH}(d)^T \;\mathbf{SH}(d)\;\delta b \;\;=\;\; \mathbf{SH}(d)^T \;(d \;-\; \mathbf{SH}(d)b\,) \qquad , \tag{2.12}$$

again normal equations for

$$\mathbf{SH}(d)\delta b \;\;\approx\;\; d \;-\; \mathbf{SH}(d)b \qquad , \tag{2.13}$$

which, for a starting guess of b =0, is identical to (2.10), since the problem was linear. For the same reason the Gauss-Newton equations are identical to the Newton equations.

Multichannel deconvolution

By this I mean designing a single filter for a gather of traces. Concatenate the data traces d_i and the convolutional matrices $\mathbf{SH}(d_i)$ and the single channel derivation above now applies. The matrix J partitions into blocks with one block for each data trace. This structuring arises because each trace can be filtered separately once the common filter is obtained.

Actually, one should not simply concatenate. To avoid bias, each data trace should be scaled to the same RMS amplitude. This is one form of weighting. Divergence correction is another example of weighting that removes bias. In addition to avoiding bias, weighting may also be useful as a preconditioner for improving accuracy and convergence. Prewhitening is one example of this. It is used to reduce the influence of noise outside the useful signal band on the inverse filter design.

Simultaneous pre- and post-NMO deconvolution

Following Claerbout (1986), let **bub** be convolution with the unknown bubble prediction filter *bub* , **rev** be convolution with the unknown multiple dereverberation filter *rev* , and **NMO t** represent spherical and normal-moveout correction. Denoting the identity operation by **I**, we have then that

$$f \;\;=\;\; (\,\mathbf{I} - \mathbf{rev})\;\mathbf{NMO\;t}\;(\,\mathbf{I} - \mathbf{bub})\;data \qquad , \tag{2.14}$$

and J is given by

$$-\Big[\;\; \mathbf{SH}_{rev}\,(\;\mathbf{NMO\;t}\;(\,\mathbf{I} - \mathbf{bub})\;data\;)\;,\;(\,\mathbf{I} - \mathbf{rev})\;\mathbf{NMO\;t}\;\mathbf{SH}_{bub}\,(data\,)\;\;\Big] \tag{2.15}$$

The Gauss-Newton step for the filter perturbations δ_{bub} and δ_{rev} is computed from

$$\mathbf{SH}_{rev}\,(\mathbf{NMO\ t}\,(\,\mathbf{I}-\mathbf{bub})\,data\,)\,\delta_{rev}\;+$$

$$(\mathbf{I}-\mathbf{rev})\,\mathbf{NMO\ t}\,\mathbf{SH}_{bub}\,(data\,)\,\delta_{bub}\;\approx\;(\,\mathbf{I}-\mathbf{rev})\,\mathbf{NMO\ t}\,(\,\mathbf{I}-\mathbf{bub})\,data\quad, \tag{2.16}$$

which is more loosely written as

$$\delta_{rev}\,\mathbf{NMO\ t}\,(\,\mathbf{I}-\mathbf{bub})\,data\;+$$

$$(\,\mathbf{I}-\mathbf{rev})\,\mathbf{NMO\ t}\,\delta_{bub}\,data\;\approx\;(\,\mathbf{I}-\mathbf{rev})\,\mathbf{NMO\ t}\,(\,\mathbf{I}-\mathbf{bub})\,data\quad . \tag{2.17}$$

With initial guess $bub = rev = 0$ this reduces to Claerbout's form:

$$\delta_{rev}\,\mathbf{NMO\ t}\,data\;+\,\mathbf{NMO\ t}\,\delta_{bub}\,data\;\approx\;\mathbf{NMO\ t}\,data\quad .$$

In that investigation he did not to proceed beyond the first Gauss-Newton step, electing instead to indicate how one might proceed by relinearizing with (2.16).

Forming $J^T J$ involves auto- and cross-correlation of data partially deconvolved by either *rev* or *bub* but not both. The second derivative is more complex than $J^T J$ because the cascade of *rev* and *bub* filters is nonlinear. There is a component of the nonlinear term K that is independent of the current estimates of the filters *rev* and *bub* , however. The entries of K are the dot products of the various derivatives of J with the vector f . Examining formula (2.15) for J above, we can see immediately that the *rev* derivatives of the first half of J and the *bub* derivatives of the second half of J are identically zero. The remaining crossterms are

$$\frac{\partial f}{\partial rev\,\partial bub}\;=\;\mathbf{SH}_{rev}\,(\,\mathbf{NMO\ t}\,\mathbf{SH}_{bub}\,(data\,)\,) \tag{2.18}$$

Thus, K takes the form of a symmetric 2×2 block matrix with the diagonal blocks zero and the off-diagonal blocks dot products of shifted versions of moveout and gain-corrected delayed copies of the data with the current estimate of the deconvolved data. That this is not just correlation of the input and output of deconvolution reflects Claerbout's underlying observation that neither normal-moveout correction nor gain correction commutes with debubble (or dereverberation) filtering.

2.4 SURFACE-CONSISTENT PREDICTION-ERROR FILTERING

I turn now to the problem at hand: designing one filter for each surface shot and receiver station of a complete seismic survey. Generalizing from the previous examples, I now seek to minimize the norm of f where the components of f are the filtered traces

$$f_{ij}\;=\;(\,\mathbf{I}-\mathbf{s}_i\,)\,(\,\mathbf{I}-\mathbf{g}_j\,)\,d_{ij}\quad . \tag{2.19}$$

The gradient J of f takes the block form

$$-\Big[\;\mathbf{SH}_{s_i}(\,(\,\mathbf{I}-\mathbf{g}_j\,)\,d_{ij}\,)\;,\;(\,\mathbf{I}-\mathbf{s}_i\,)\,\mathbf{SH}_{g_j}(d_{ij}\,)\;\Big]\quad , \tag{2.20}$$

with zero blocks for all other i and j in each row. Invoking the commutative law of convolution, I can rewrite this as

$$-\left[\ \mathbf{SH}_{s_i}(\ (\ \mathbf{I}-\mathbf{g}_j\ )\ d_{ij}\ )\ ,\ \mathbf{SH}_{g_j}(\ (\ \mathbf{I}-\mathbf{s}_i\ )\ d_{ij}\ )\ \right] \quad . \tag{2.21}$$

It is now immediate from (2.7) that the Gauss-Newton step is found by solving the least-squares system

$$\delta s_i\,(\ \mathbf{I}-\mathbf{g}_j\ )\ d_{ij}\ +\ \delta g_j\,(\ \mathbf{I}-\mathbf{s}_i\ )\ d_{ij}\ \approx\ (\ \mathbf{I}-\mathbf{g}_j\ )\,(\ \mathbf{I}-\mathbf{s}_i\ )\ d_{ij} \quad . \tag{2.22}$$

The matrix dimensions for this system are quite large, however. Using the dimensions of the small data example from the previous section, the number of elements in f , and the number of rows in J, is close to 9 million. The number of columns in J is the number of filter coefficients being designed which is several thousand. This means that to work directly on the overdetermined system (2.22), I have to anticipate and manage I/O to and from secondary storage in addition to the computation involved in this sparse matrix multiply. Alternatively, I can sacrifice the superior numerical conditioning associated with working directly with J and use $J^T J$ with, say, SYMMLQ instead to keep all dimensions down to a few thousand. In the latter case, $y = J^T Jx$ looks like Fig. 2.5. This involves external I/O as well, but only to read each trace once during the pass.

```
do itrace=1,ntrace
     strace=conv(1-g(j),trace(itrace))
     gtrace=conv(1-s(i),trace(itrace))
     temptr=conv(xs(i),strace)+conv(xg(j),gtrace)
     ys(i)=ys(i)+xcorr(temptr,strace)
     yg(j)=yg(j)+xcorr(temptr,gtrace)
end do
```

FIG. 2.5. Code fragment to perform $y = J^T Jx$ for surface-consistent prediction-error filter design. Here g and s are initial guesses for the filters and **conv** is convolution, **xcorr** is cross-correlation. In principle, *strace* and *gtrace* could be precomputed. In practice, it costs less to recompute them than to retrieve them from disk.

How much of a sacrifice is it to take the $J^T J$ route? Conventional wisdom is that the number of conjugate-gradient iterations required for convergence increases, and, to a lesser extent, numerical accuracy decreases. If the problem is well conditioned, that is if many of the largest eigenvalues of $J^T J$ are close to each other, convergence is quite rapid. I tested conditioning experimentally for the problem of designing a single filter for the field profile shown in Figure 2.6. I chose to design a 132 msec filter with a 60 msec gap. Figure 2.7 exhibits the filters designed using J and J^T with LSQR and those designed using $J^T J$ with SYMMLQ. Figure 2.8 displays the corresponding deconvolved gathers. LSQR needed seven iterations to generate its filter; SYMMLQ required ten iterations with comparable tolerances. The low frequency content of the two prediction filters is markedly different but the deconvolutions are

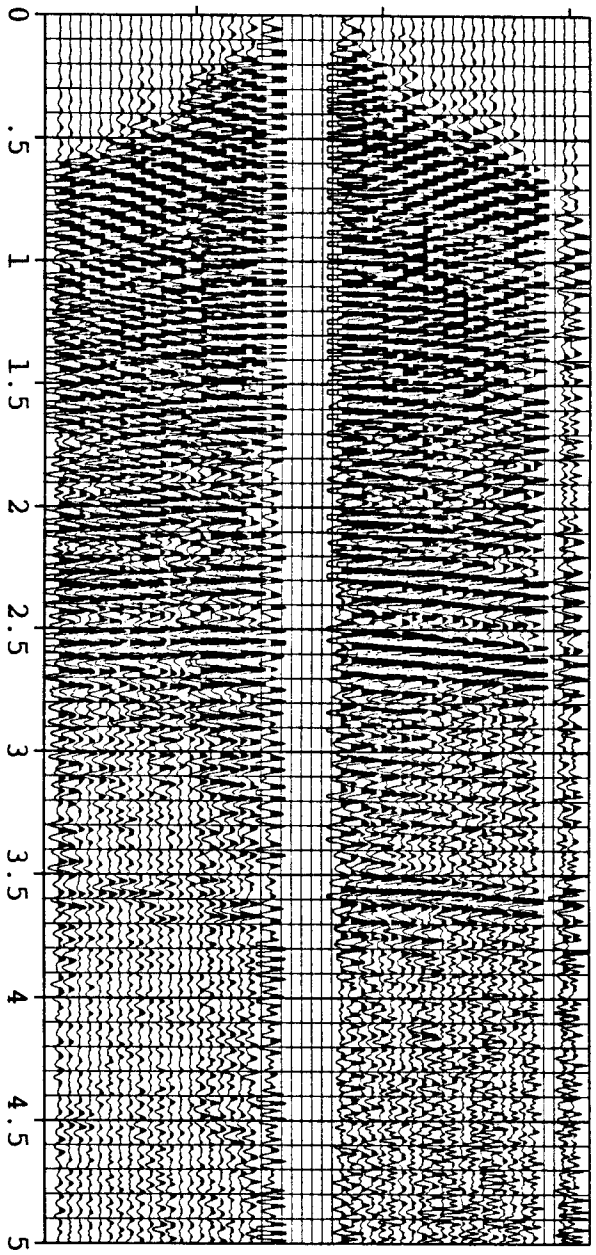
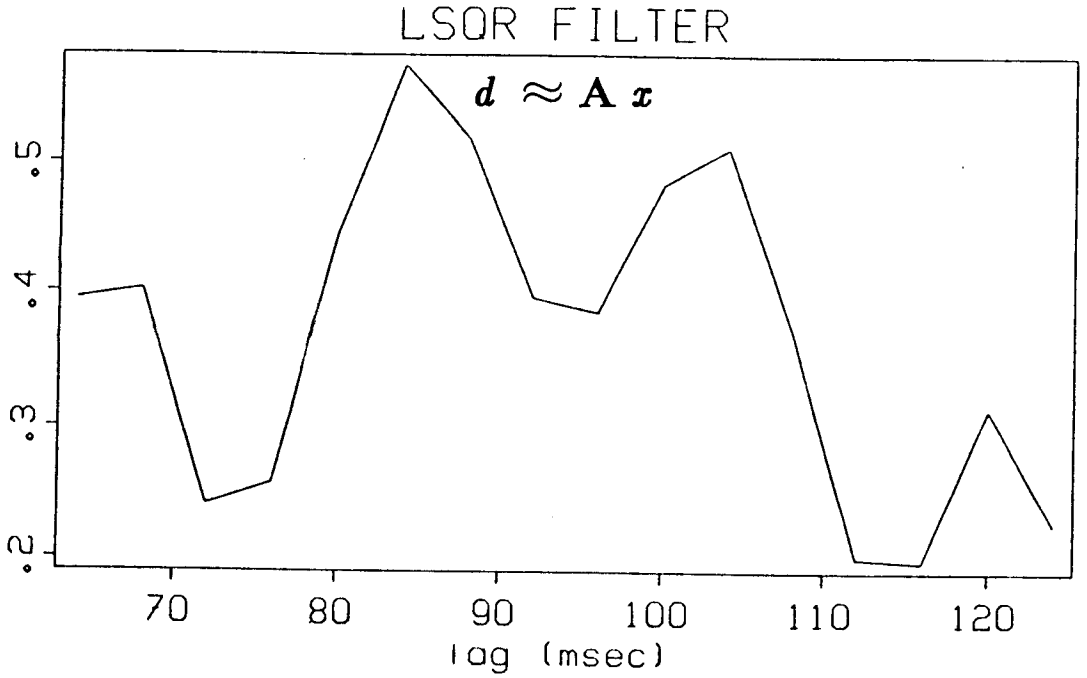


FIG. 2.6. Field gather from the Central Valley of California. I use this gather to compare two prediction-error filter design methods.



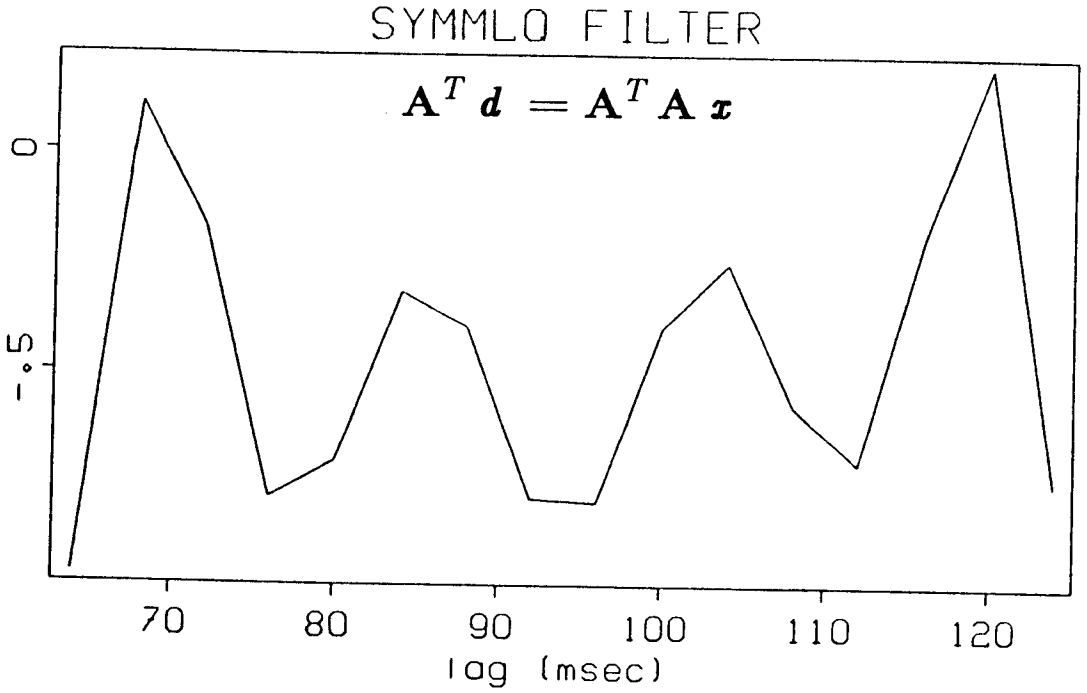


FIG. 2.7. Prediction filters designed to deconvolve the field gather of Fig. 2.6. The LSQR filter is an iterative solution to the overdetermined least-squares system of equation (2.10). The SYMMLQ filter is an iterative solution to the corresponding normal equations (2.9). The filters differ, but produce identical deconvolutions. (See Fig. 2.8.)

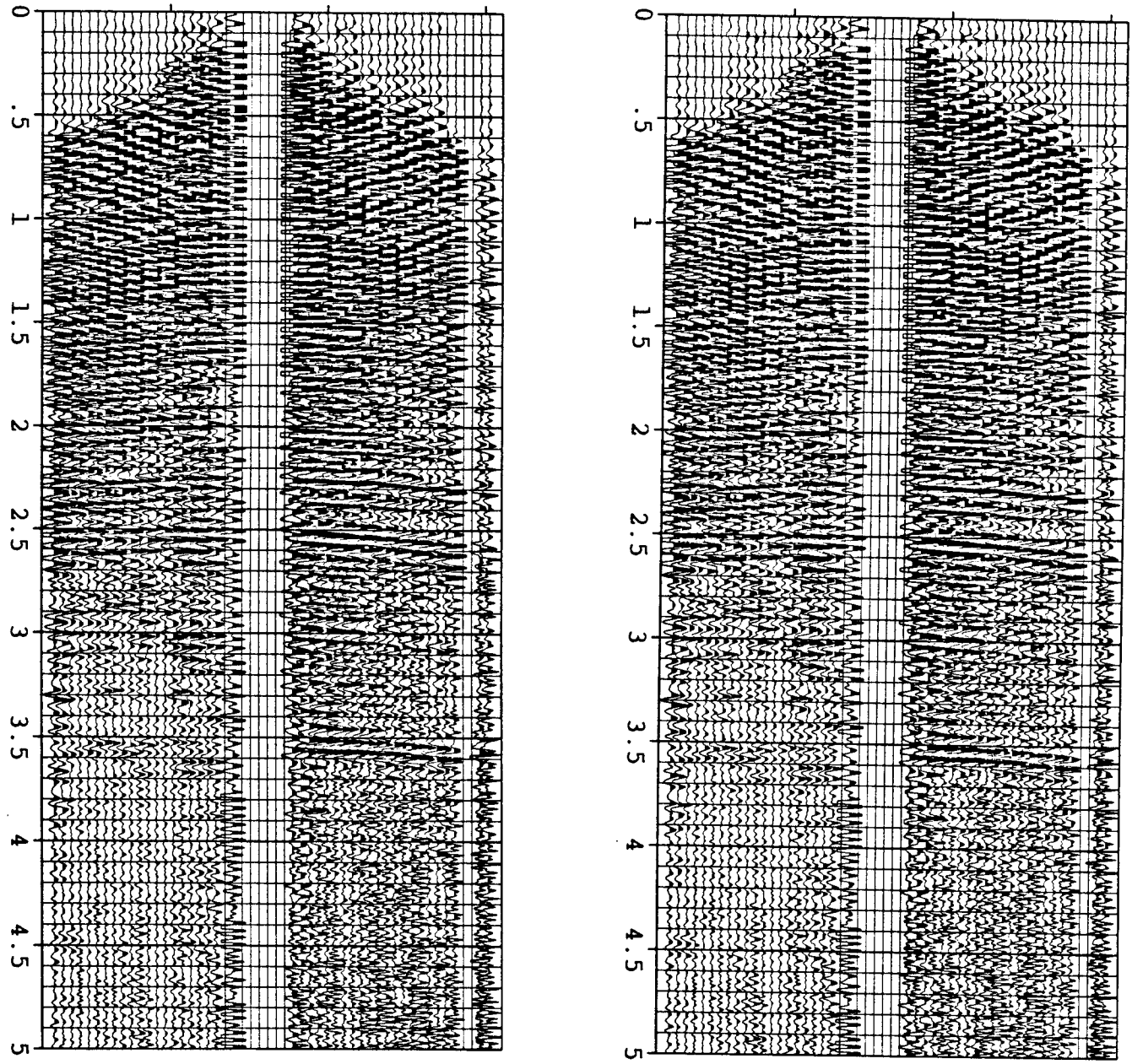


FIG. 2.8. Field gather of Figure 2.6 deconvolved with (a) the LSQR prediction-error filter, and (b) the SYMMLQ prediction-error filter. The results are the same; I can use either method of filter design. For surface-consistent prediction-error filtering, I opt for SYMMLQ because it greatly simplifies program design and I/O management.

nearly identical. This reflects the lack of corresponding low frequencies in the data; my filter design was unconstrained outside the bandwidth of the data. Conclusions: 1) I don't require the additional numerical accuracy of LSQR in order to obtain a reasonable deconvolution of these data, 2) the moderate increase in the number of iterations required by the SYMMLQ design is a modest price to pay for the corresponding simplification in program design and I/O management.

Designing surface-consistent filters for the whole survey is more involved than designing a single filter for a single gather. The surface-consistent constraints complicate the discussion of conditioning. This is, after all, where nonlinearity arises. Even if deconvolution of individual gathers is a well-conditioned process, unraveling the interrelation between the gathers might be poorly conditioned. My previous experience with surface-consistent trace balancing (Levin, 1985) raises this possibility. There I needed several dozen iterations to achieve convergence. This was quite different from my experience with surface-consistent residual statics solutions (Biondi and Levin, 1985). There we found the decomposition of time shifts into surface-consistent components to be remarkably well-conditioned. Which applies to surface-consistent predictive deconvolution?

If I were to simply compute prediction-error filters $1+\epsilon_t$ for each trace, surface-consistent decomposition would try to fit these with $(1+\epsilon_s)(1+\epsilon_g)$. Canceling the unity at zero lag, we get $\epsilon_t \approx \epsilon_s + \epsilon_g + \epsilon_s \epsilon_g$. If the ϵ's are small, the nonlinear cross-term is small and the linear fit, i.e. the well-conditioned residual statics solution, dominates. If the ϵ's are large, the nonlinear term dominates and we are in the troublesome trace-balancing regime. Physical processes that reduce energy, such as attenuation and transmission loss, will make ϵ small. This argues that the surface-consistent unraveling will be well conditioned.

In a larger sense, it is the criterion of minimum prediction error that is really working in our favor. In tackling equations (2.22) what we seek is not highly accurate values for the perturbations δs_i and δg_j. We really seek values that reduce the prediction error. The usual measures of matrix conditioning look at the prediction error surface and give us a measure of how long and narrow the bottom of the valley is. If we think about pouring a little water into this valley, what they measure is how far away from the center of this valley you can be and still get wet feet. The minimum-prediction error criterion is interested in altitude, not latitude, however. As Figs. 2.7 and 2.8 show, the prediction error will change imperceptibly as your choice of filter wanders throughout the the bottom of the long, narrow valley.

To confirm these conclusions I used SYMMLQ to design surface-consistent 64 msec shot and geophone deconvolution filters for the data in Figure 2.1. The result of 5 iterations, displayed in Figure 2.9, is much closer in appearance to the single-trace deconvolution of Figure 2.4 than either of the gather-by-gather deconvolutions in Figs. 2.2 or 2.3. The prediction error is almost ten times smaller than that shown in Fig. 2.2

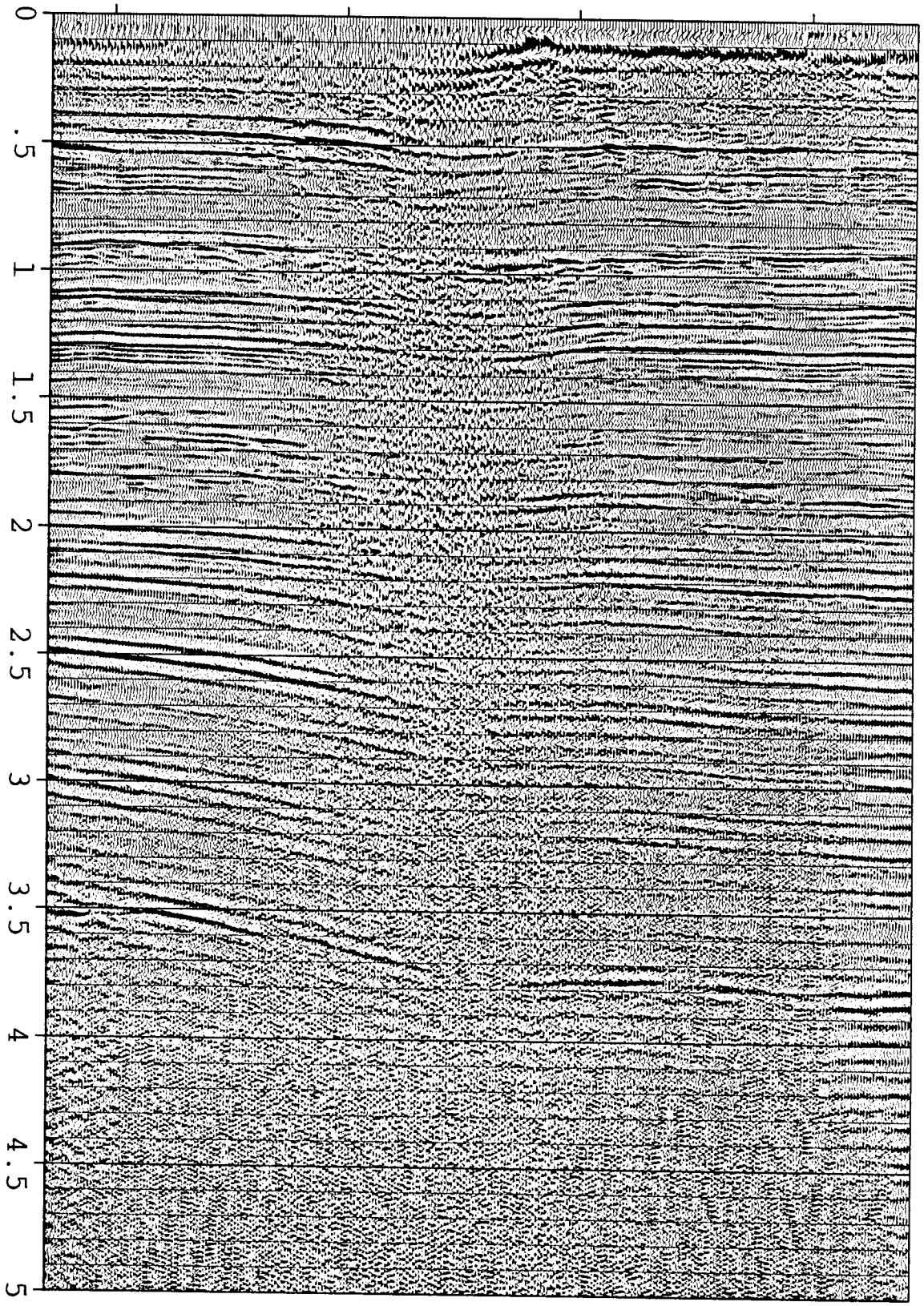


FIG. 2.9. Deconvolution of the data in Fig. 2.1 with shot and geophone consistent filters designed simultaneously. Filter lengths were 64 msec. Much superior to the cascaded deconvolutions of Figs. 2.2 or 2.3, this stack is close to the result of single-trace prediction error filtering in Fig. 2.4. Noise is better controlled and reflector continuity is somewhat stronger here, a result of the surface-consistent averaging.

and the reflection events at early times show better continuity than the single-trace deconvolution. This illustrates the superior performance of simultaneous filter design and shows that the problem is quite manageable.

In the next two sections I will apply surface-consistent prediction-error filtering to two other seismic lines, one land and one marine. The land data is distinguished by very good signal to noise and a decidely irregular recording geometry. For the marine example, I design surface-consistent gapped deconvolution filters for the suppression of seafloor multiple trains.

2.5 SAN JOAQUIN VALLEY - IRREGULAR GEOMETRY

Figure 2.10 is a stack of a line from the San Joaquin Valley collected and preprocessed by Western Geophysical. The source is Vibroseis and the recording geometry is an asymmetric split spread as shown in the stacking diagram of Figure 2.11. The data are corrected for spherical divergence and trace balanced. The recording geometry irregular with changing spreads and shot spacing. This does not complicate the computation; the trace-by-trace loop of Fig 2.5 remains the same. The signal-to-noise is quite high, high enough, indeed, that conventional spiking deconvolution does a bit better job of cleaning up the data. I will explain this shortly; first I study with this line is how the irregular geometry influences the filter design.

In Figure 2.12, I applied trace-by-trace spiking deconvolution before stack with a 64-sample filter. Autocorrelations were estimated from a window containing the reflections from 1 to 5 seconds on the stack of Fig. 2.10. Normal-moveout correction was not applied before deconvolution.

Figure 2.13 displays surface-consistent shot and geophone filters designed by the procedure described in Section 2.4. Each filter was 32 samples long so that the combined length was the same as for the trace-by-trace deconvolution. Again 5 iterations were used in the conjugate-gradient least-squares solver. (Another five iterations, starting with these filters as input, produced changes only in the third significant digit.) The influence of geometry on individual deconvolution filters can be seen by comparing the fold of coverage for shot and receiver stations in Fig 2.14 with the corresponding filters in Fig 2.13. Higher frequency filters go with higher fold. The additional channels help to push down the noise in the spectral gaps and the inverse filter increases the gain to compensate for the added spectral color. After applying these filters, the stack in Figure 2.15 closely matches that of conventional deconvolution (Fig. 2.12).

Individual traces before stacking also are remarkably similar. Figures 2.16 through 2.18 show some CDP gathers selected at roughly equal intervals across the line. In the few places where differences are seen, the single-trace deconvolution shows better reflector continuity. These spots appear as small static misalignments on the surface-consistent result, but closer examination shows the corresponding locations on the unprocessed gathers in Fig. 2.16 are contaminated with coherent, low-frequency ground roll.

FIG. 2.10. Stack of Vibroseis line from the San Joaquin Valley provided by Western Geophysical. Sampling interval is 4 msec and CDP spacing is 55 ft. These data are of high quality.

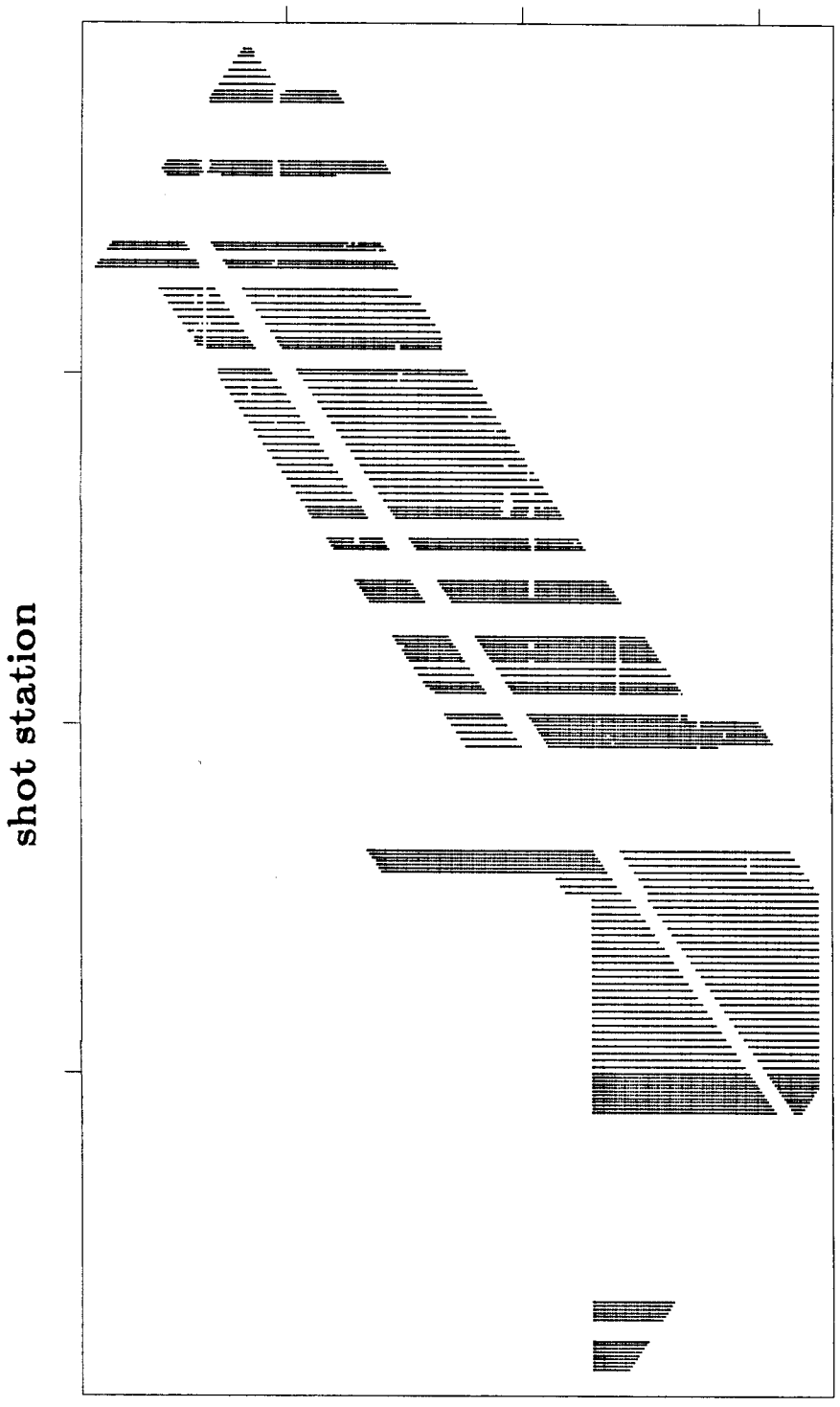


FIG. 2.11. Stacking diagram for the profile in Fig. 2.10. The recording geometry is an asymmetric split-spread. The profile deviates only a small amount from a straight line. CDP fold ranges between 13 and 42.

FIG. 2.12. Result of conventional single-trace spiking deconvolution before stack. Operators of length 256 msec. were estimated from a window containing the reflections from 1 to 5 s on the stack of Figure 2.10. The high signal-to-noise on the input data allowed the deconvolution to sharply define major reflecting horizons.

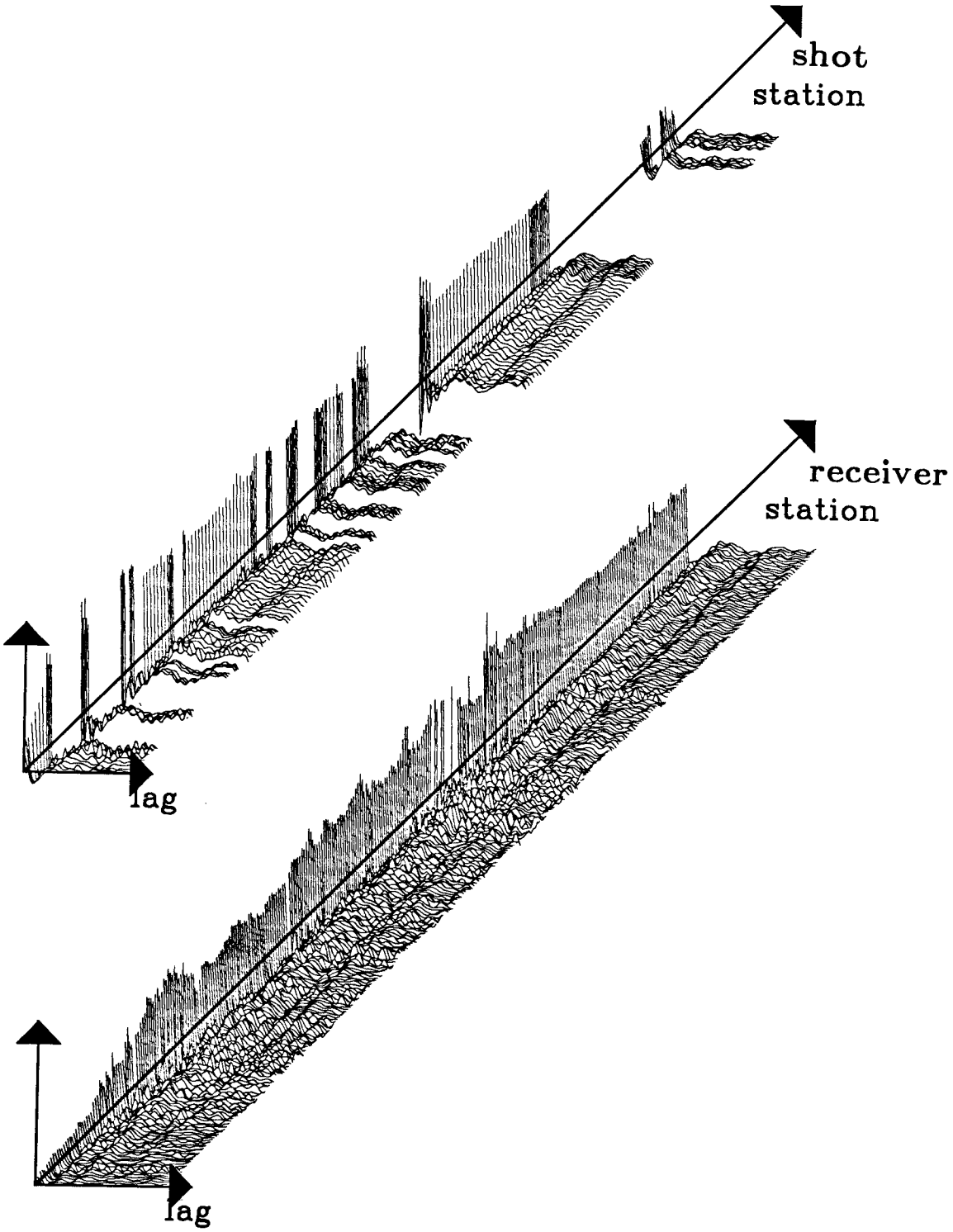


FIG. 2.13. Shot and receiver deconvolution filters designed for the surface-consistent deconvolution in Fig. 2.16. The shot filters are consistent across the line, reflecting the controlled Vibroseis source. The geophone filters also exhibit this waveshape to a lesser degree. I attribute this also to the controlled source being approximately the same on all traces. Frequency content of the inverse filters correlates well with the number of traces per station in Fig 2.14. High fold produced sharp, high frequency inverse filters.

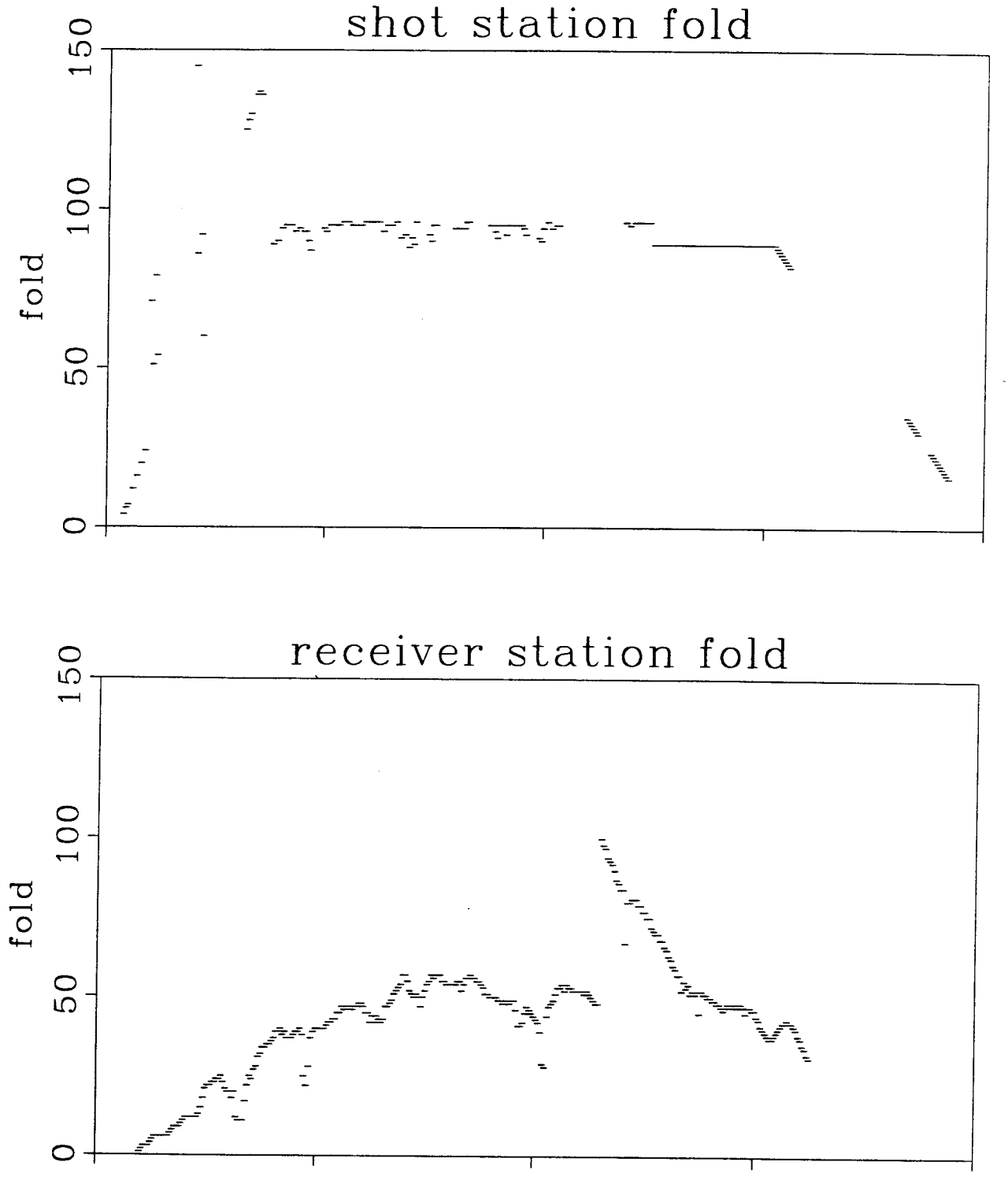


FIG. 2.14. Shot and receiver fold for this Western Geophysical line. The shot fold counts the number of live receivers for each shot. The receiver fold counts the number of traces recorded at each receiver station. These are directly reflected in the quality of the surface-consistent prediction-error filters in Fig 2.13.

FIG. 2.15. Surface-consistent deconvolution of the data of Fig. 2.10. Both this stack and individual gathers are very similar to the results of trace-by-trace deconvolution.

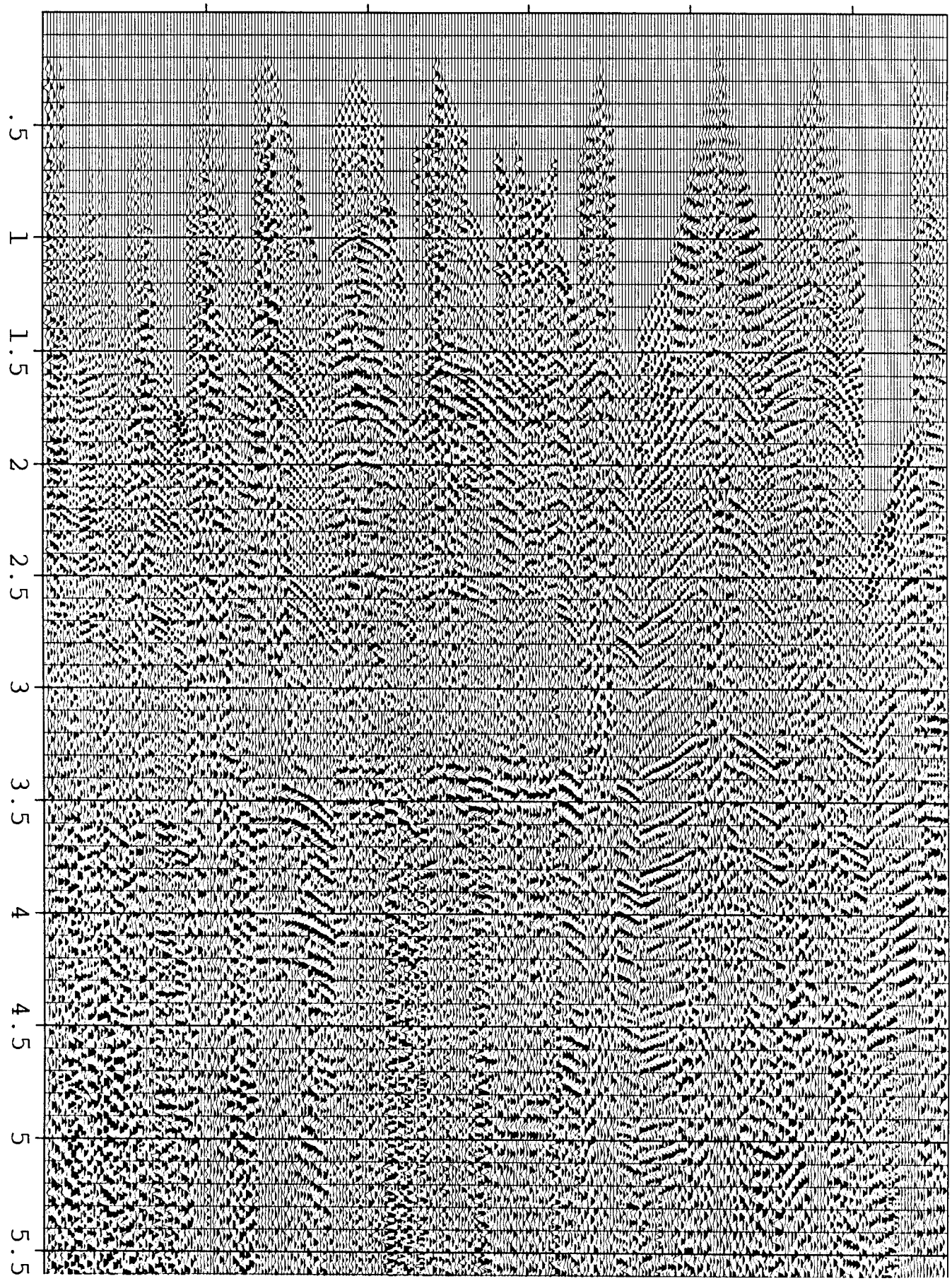


FIG. 2.16. Individual CDP gathers selected at approximately equal intervals across the line of Fig. 2.10. Signal-to-noise is quite high and, except for the very shallow reflectors before 1 s, coherent arrivals are readily picked on the individual traces.

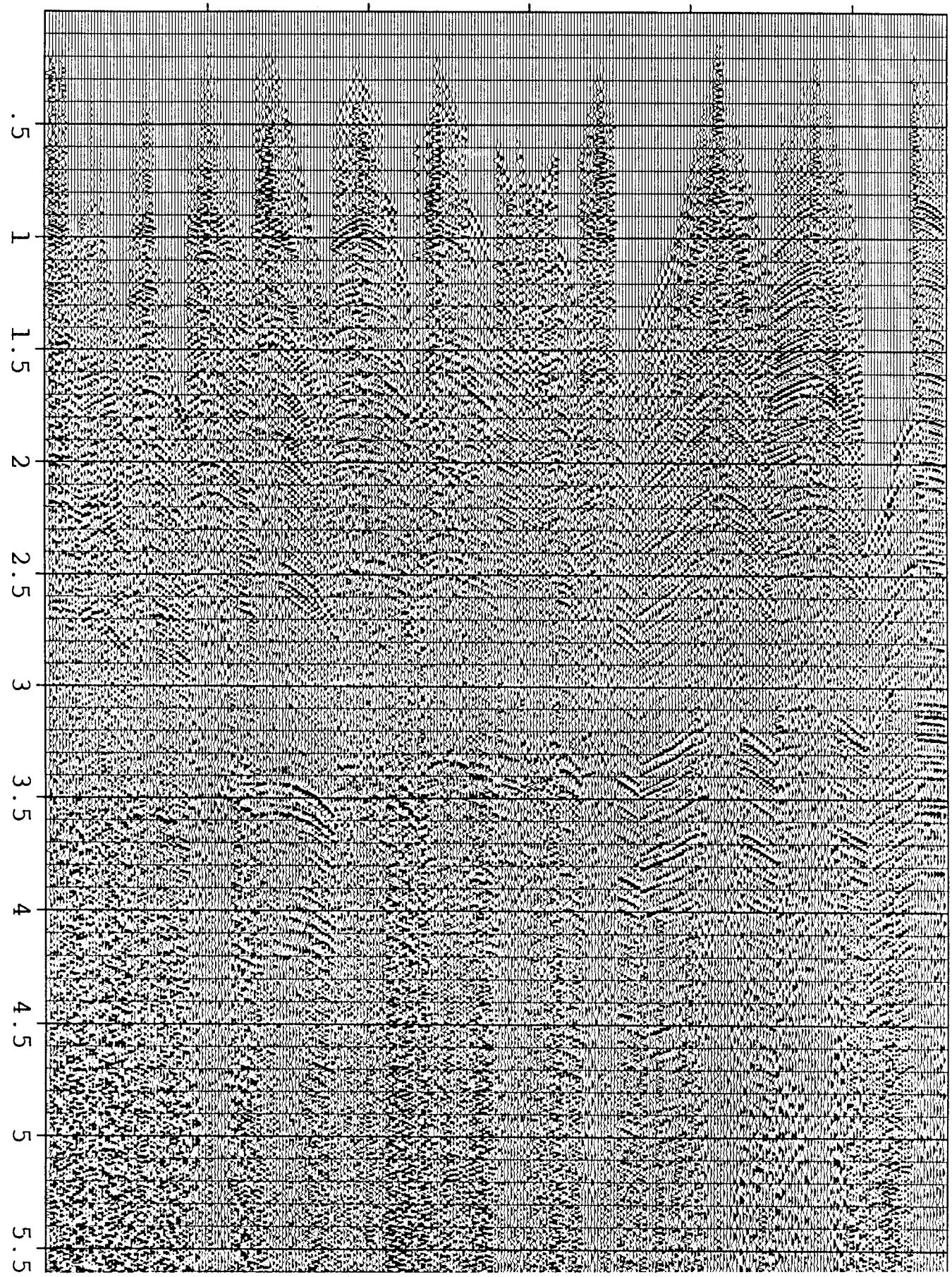


FIG. 2.17. Individual CDP gathers in Fig. 2.16 after trace-by-trace deconvolution. As expected wavelets are compressed, coherent low-frequency ground roll is suppressed and incoherent high-frequency noise is boosted.

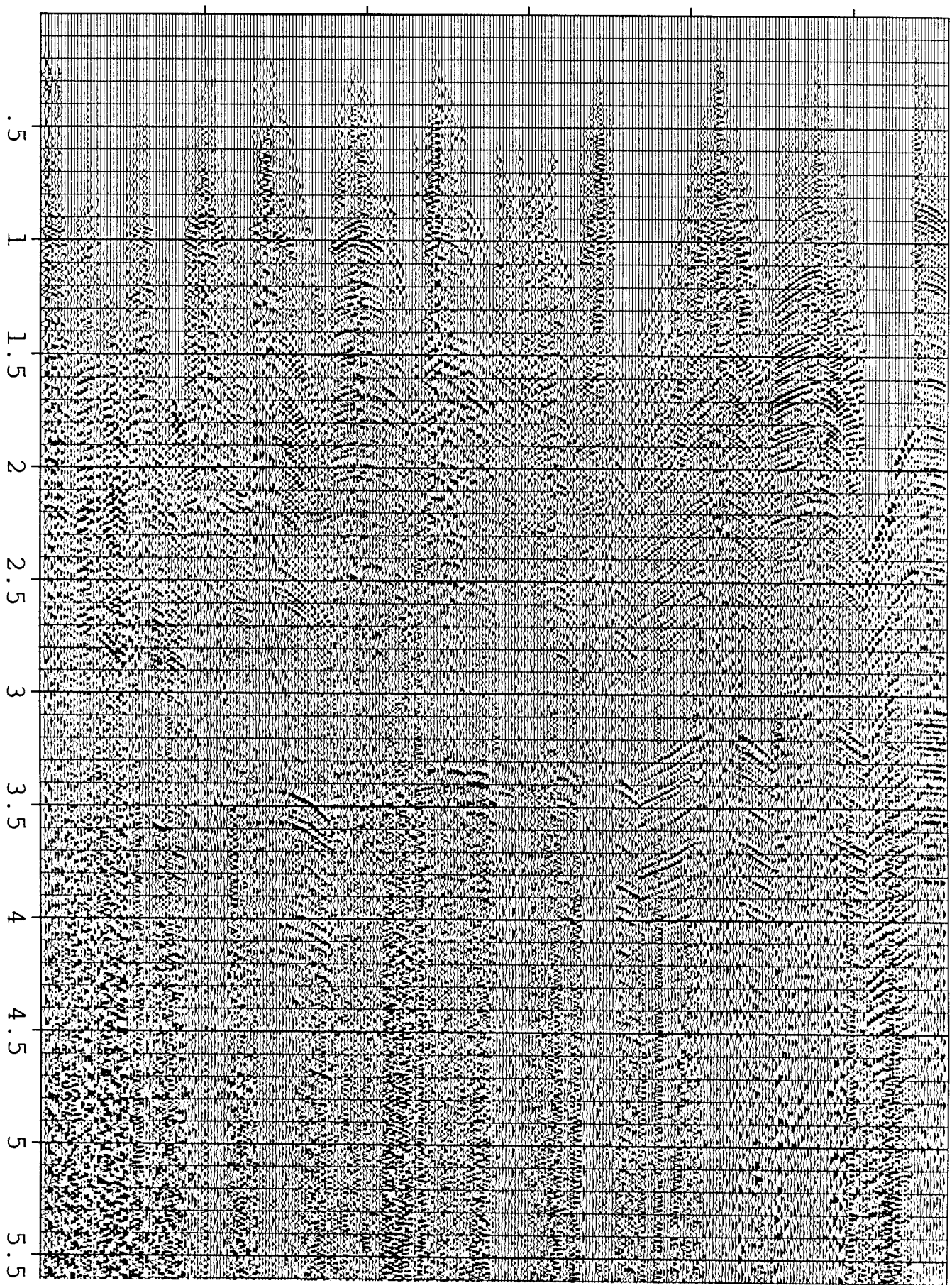


FIG. 2.18. Individual CDP gathers in Fig. 2.16 after surface-consistent deconvolution. Results are very similar to the trace-by-trace result of Fig. 2.17. Where they differ is mostly on inner offset traces that contain ground roll on the unprocessed gathers in Fig. 2.16. For these traces single-trace decon has suppressed more of the ground roll than surface-consistent decon.

Ground roll does not fit the surface-consistent model I've used. It appears on varying portions of some traces and not at all on other traces within the common-shot and common-receiver gathers. As a result it is not well predicted and suppressed by surface-consistent deconvolution and further acts to degrade the the deconvolution of that portion of the data that does fit the model. On the near-offset traces ground roll begins at almost the start of the trace and continues to the end. On these traces it is highly predictable by a single filter and so is strongly suppressed by single-trace deconvolution. Surface-consistent deconvolution sees only a average over all the offsets and so finds it significantly less predictable by a single filter.

In this example, the departure of the data from the surface-consistent model was not large. Low-cut frequency filtering prior to deconvolution could handle the problem here. It illustrates though that the surface-consistent model is only an approximation. The departure from the model is a slowly decreasing function of offset, i.e. the ground roll arrives later and later and finally disappears. Such slowly-varying deviations from the model have been considered by Claerbout (1982). He suggests that instead of constraining the adjustable parameters to exactly fit the model, the sum-of-squares for the unconstrained problem should be modified by adding quadratic terms that penalize against all but small, smooth departures from the model. For surface-consistent deconvolution, this means specifying two filters for each trace and minimizing the sum of the squares of trace-by-trace prediction errors plus the sum of squares of some spatially lowcut version of the filters. I expect there will be considerable practical difficulty making this work. In addition to the units or scaling problem arising from adding data and filter terms, the unconstrained problem is massively underdetermined and will lead to numerical problems in the iterations. I do not recommend developing this extension to surface-consistent deconvolution until we have further experience with smaller applications.

2.6 THE FAILURE OF MARINE MULTIPLE SUPPRESSION

One of the early applications of surface-consistent deconvolution was in the suppression of marine multiple reflections (Morley and Claerbout, 1983.) Similar to conventional gapped deconvolution, this method assumes nearly vertical emergent raypaths and water deep enough for the source bubble pulse to die away before the onset of the seafloor reflection (see also Nedlin, 1985.)

Like Morley and Claerbout, I selected a marine profile from the Barents Sea which meets these criteria. The near-offset section of Figure 2.19 and the common-shot gather of Figure 2.20 show the features of this line. The duration of the source waveform is about 100 ms as measured from the direct arrival, and the nearly flat seafloor reflection arrives at about 400 ms, with a velocity of 1.46 km/s as determined from fathometer readings. The 2.58 km/s refraction seen at the further offsets shows that velocity increases rapidly below the seafloor. (This refraction is not directly from

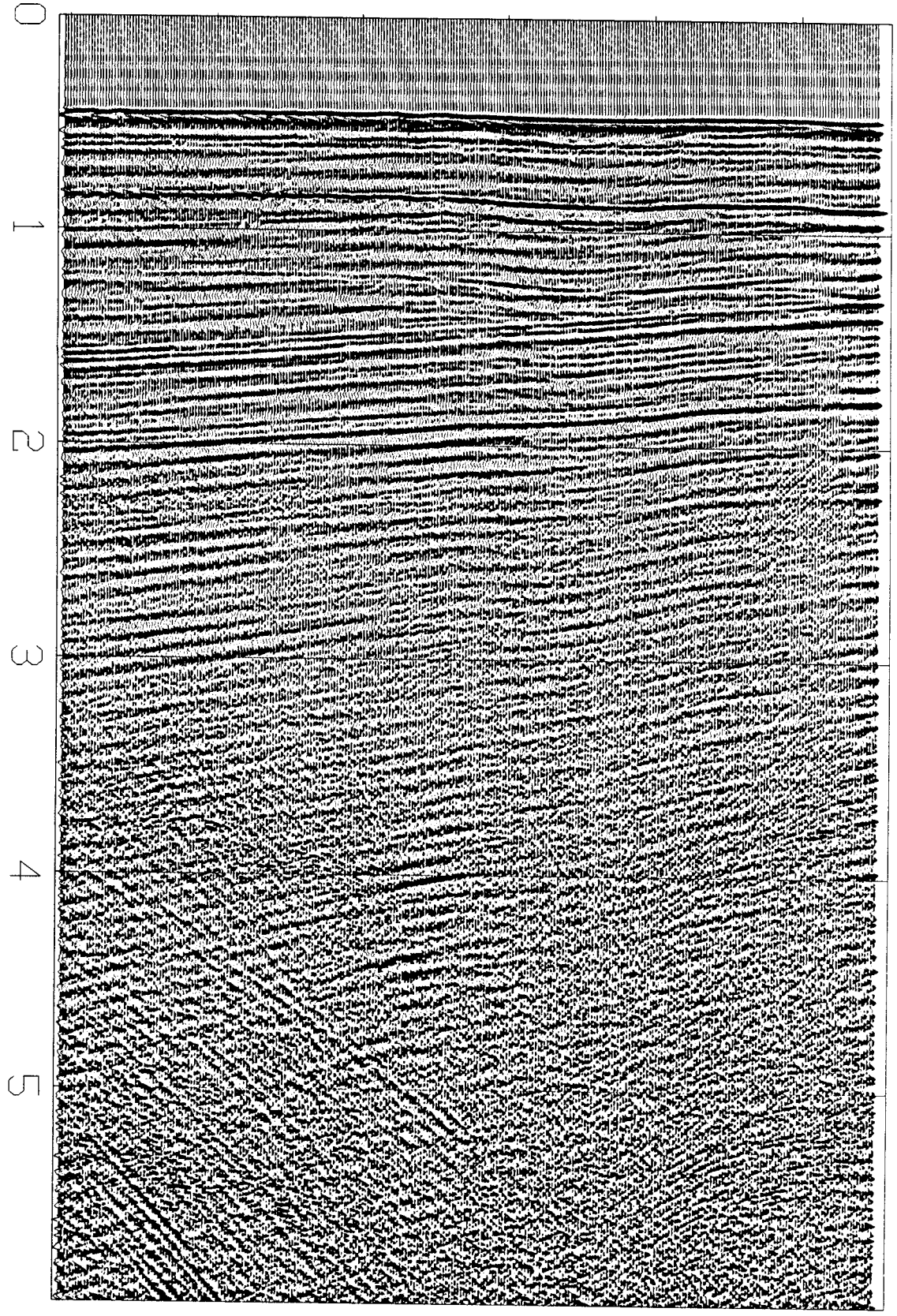


FIG. 2.19. Near-offset section from the Barents Sea. The offset is 294 m and the trace spacing here is 25 m. Prominent features are the seafloor reflection at .4 s; its first multiple at .8 s; a strong reflector dipping at about three degrees arriving at about 1.5 s; and its first pegleg arriving just before 2 s. Display gain here is proportional to t^2.

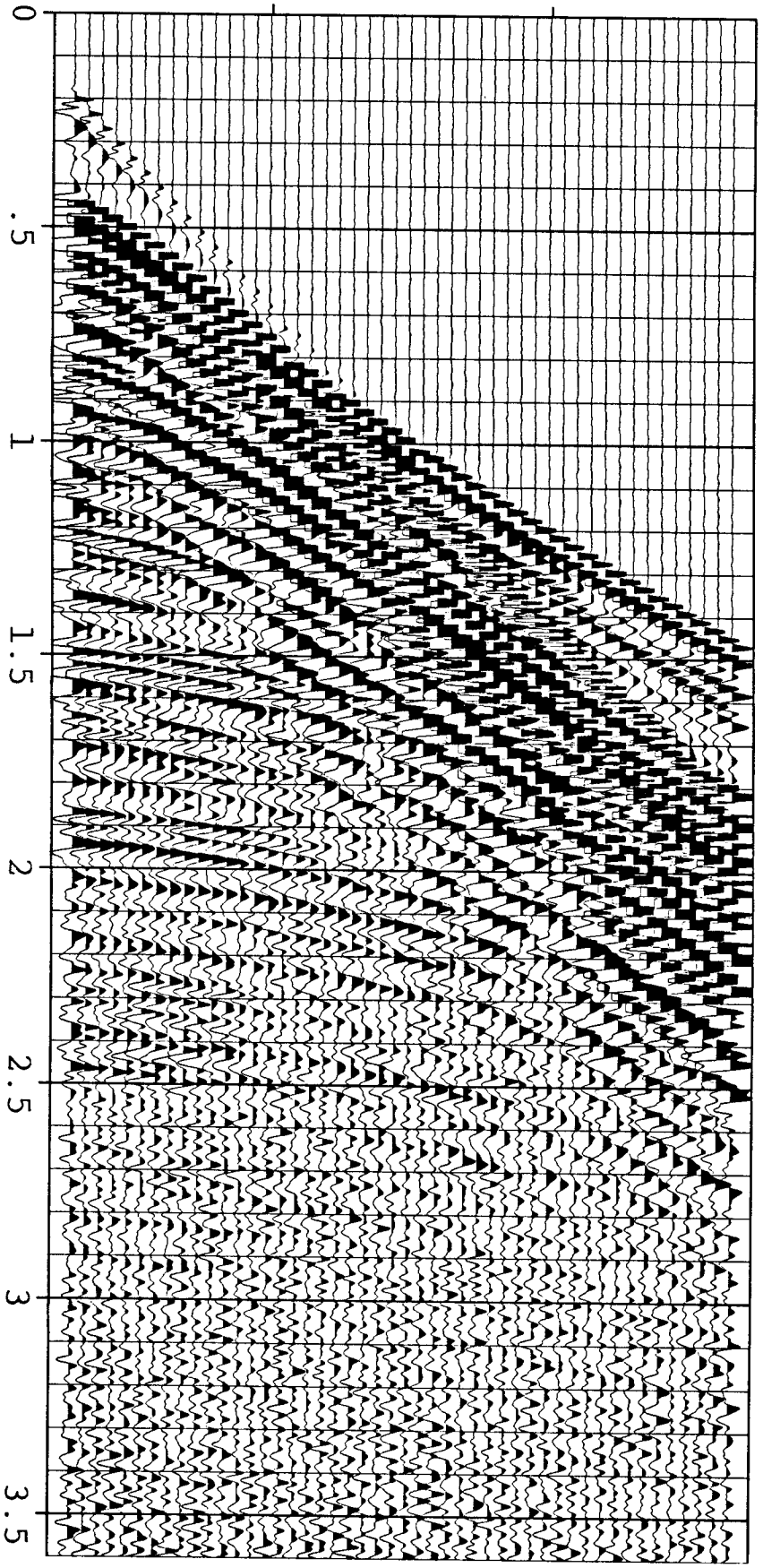


FIG. 2.20. Common-shot gather from the middle of the Barents Sea line. In addition to the reflectors appearing on the near-offset section of Fig. 2.19, a multiply-reflected refraction is visible at the wide offsets. The refraction has a velocity of 2.58 km/s; further study shows it arises from a hidden layer below the seafloor.

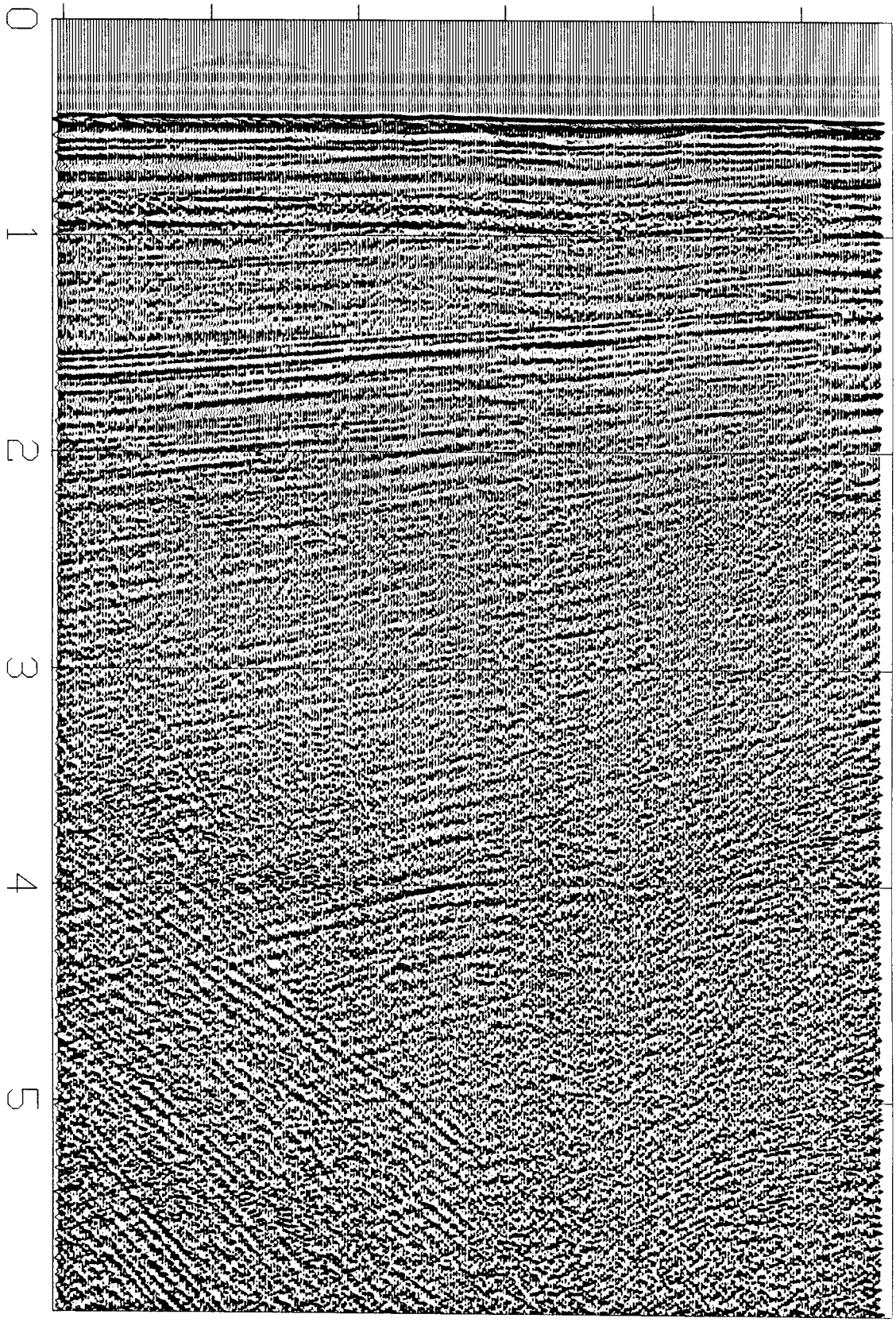


FIG. 2.21. Near-offset section after trace-by-trace gapped deconvolution. Normal-moveout correction at primary velocity was applied before deconvolution and removed afterwards before display. The filters were 128 ms long and began just prior to the seafloor reflection, a gap of 380 ms. t^2 weighting was used to reduce the influence of the strong, shallow seafloor reflection on filter estimation and increase pegleg attenuation. The multiple attenuation is good on this single offset panel, but, as Fig. 3.6 shows, is only fair on the stack.

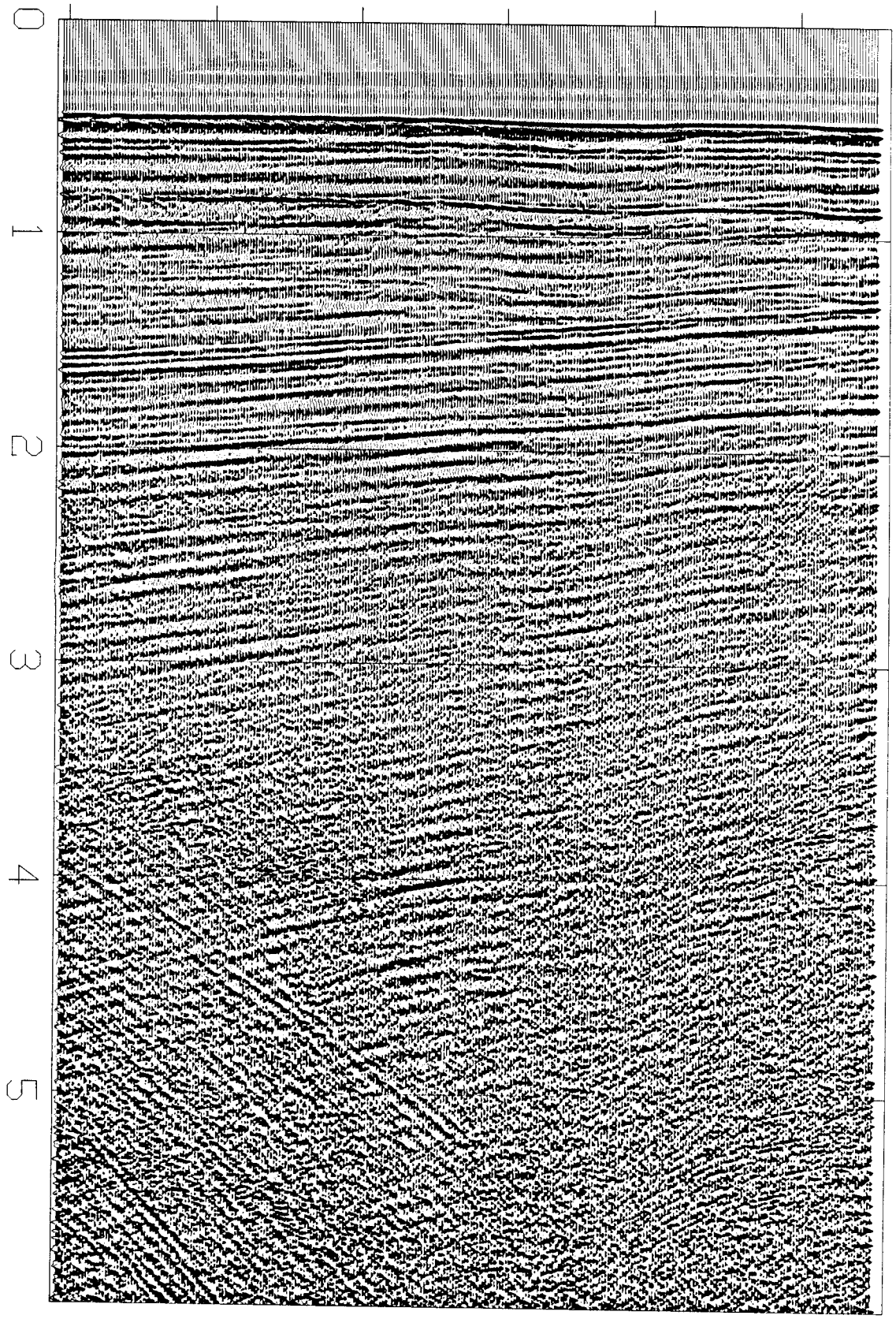


FIG. 2.22. Near-offset section after surface-consistent deconvolution. Again, normal-moveout was applied before deconvolution and removed afterwards. Shot and receiver filter lengths were each 128 ms; both were specified with a gap of 380 ms. Unlike single-trace gapped deconvolution, the process was unable to attenuate even the pure seabottom multiple. These data do not fit the surface-consistent model.

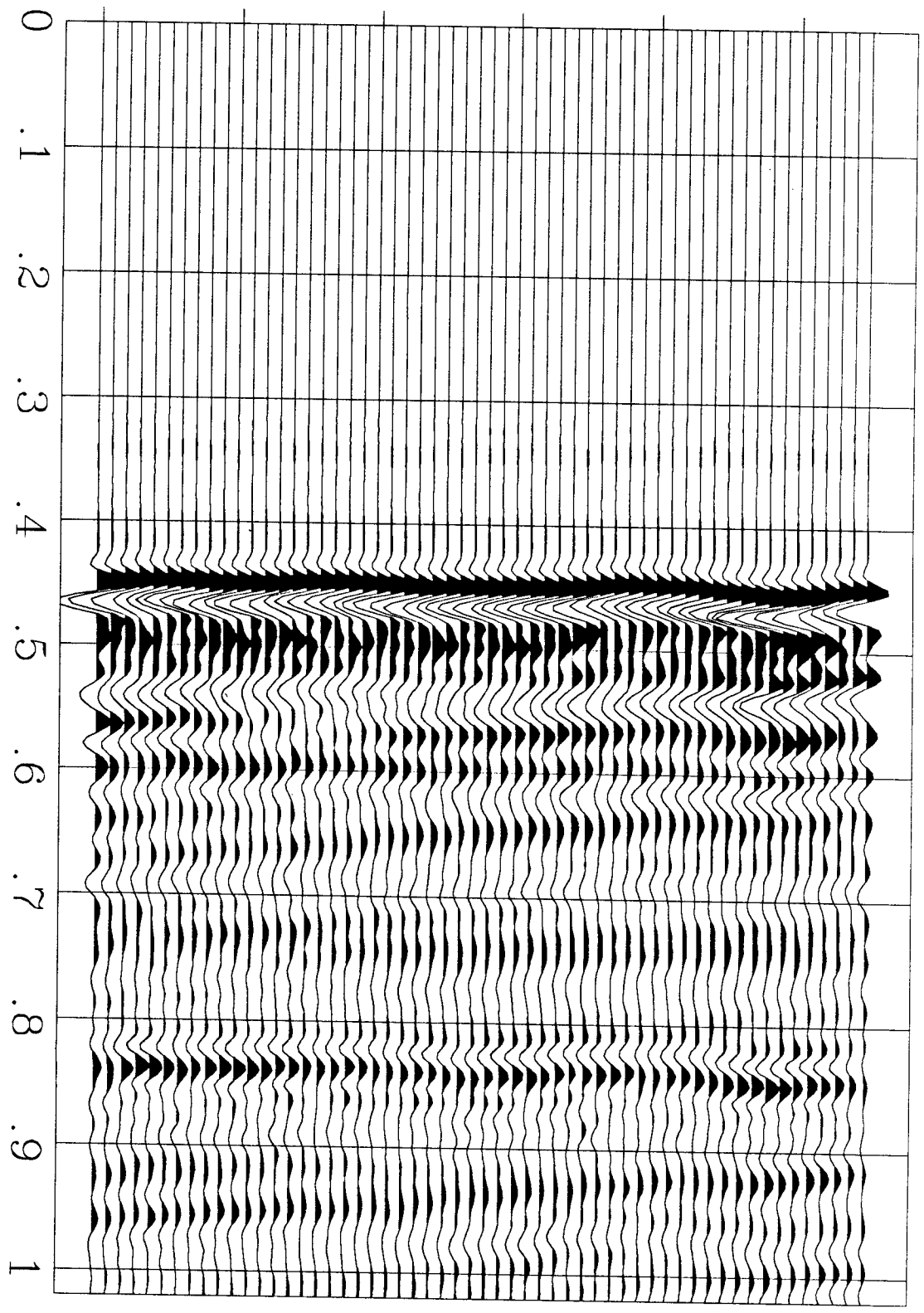


FIG. 2.23. Enlargement of the near-offset seafloor reflection in Fig. 2.19 showing the seafloor texture on this "flat" water bottom clearly. This texture diffracts and diffuses incident waves away from the vertical, giving rise to a composite waveform on the waterpath multiples poorly predicted by vertical time delays.

the seafloor; the strong peak that arrives at about 1.2 km offset implies a critical angle of 60 ° and a seafloor velocity about 1.7 km/s. Detailed refraction modelling bears this out.)

After normal-moveout correction, I applied the $t-x$ surface-consistent deconvolution developed in this chapter to the profile. The shot and receiver filters were specified to have a gap of 380 ms, i.e. just prior to the onset of the seafloor reflection, and a length of 32 samples (128 ms) beyond that. Figure 2.21 is the near-offset section after single trace-deconvolution; Figure 2.22 is the near-offset section after surface-consistent deconvolution.

The surface-consistent deconvolution was unable to suppress the seafloor or pegleg multiples. The single-trace deconvolution did a reasonable job on the near-offset panel, but, as Fig. 3.6 will show, the improvement is only fair on the stack. The problem is two-fold. First, the pegleg amplitudes do not fit the same model as the pure seabottom multiple train. This is a well recognized problem and will be discussed in greater detail in the next chapter. We can expect to suppress peglegs or the pure seabottom multiples but not generally both. Usually we downweight the pure seabottom multiple as I've done here because it is normally attenuated by the stack. Second, the deep water requirement conflicts with the vertical raypath assumption. The seafloor texture that we want the prediction-error operators to sense is diffused by spreading and healing of waves as they transit between the seafloor and the surface. This texture is readily seen in the enlargement of Figure 2.23. As a result, each new multiple bounce arrives with a different composite wavelet that is not properly predicted by vertical time delays either before or after normal-moveout correction. In the next chapter I will use multichannel wave extrapolation to successfully unravel these effects.

2.7 PHASE

Minimum phase

For the three field datasets in this chapter I designed one-sided, causal prediction-error filters. I did not constrain the filters to be minimum phase. However, by converting the inverse filters to reflection coefficients, I found they all were. If so desired, minimum phase can be forced by adapting the Burg maximum-entropy deconvolution to estimate surface-consistent reflection coefficients (Burg, 1975.) By estimating one reflection coefficient for each shot, then one coefficient for each receiver, then a second for each shot, etc., the minimum-phase property is automatic when reflection coefficients are constrained to be less that unit magnitude. This interleaving is not the same as simultaneous design, but is a good deal closer than designing all the reflection coefficients for each shot and then all the reflection coefficients for each receiver. A drawback of this method is that it requires a separate pass over the data for each new coefficient. This will generally be several times more than the half dozen passes I've used in the least-squares prediction-error filtering.

Zero phase

The two land field examples used in the chapter were acquired with Vibroseis sources. One expects these data to exhibit an approximately symmetric, two-sided source waveform. Yet all my processing was done with one-sided deconvolution filters.

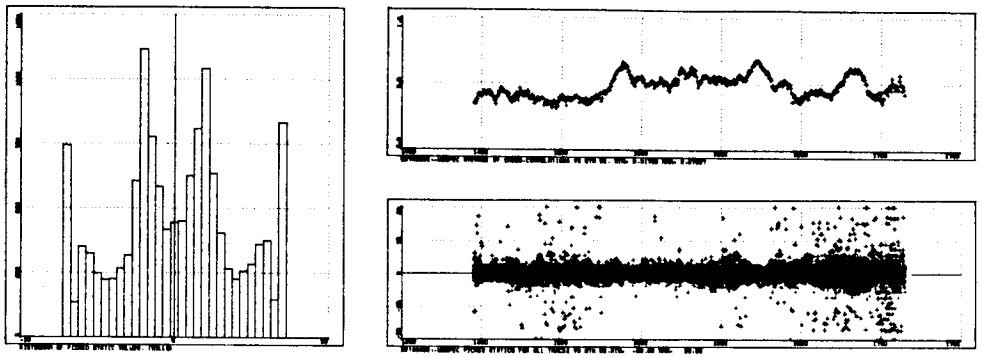
Symmetric, two-sided (zero-phase) source filters can be generated with minor modifications to the least-squares procedure. Specify one side of the filters and reflect them symmetrically to make two-sided filters. Cross-correlations must also be summed symmetrically. The receiver filters may, and ought to, remain causal.

While symmetric filters are easily incorporated into surface-consistent deconvolution, one has to question whether this is really how Vibroseis data is best handled. The source is very repeatable; with a constant or slowly varying source waveform, surface-consistent estimation of common-receiver filters degenerates to a gather-by-gather deconvolution. Also, the far-field source waveform is not really zero-phase both because of unknown phase shifts introduced by baseplate coupling and because of (minimum-phase) scattering, multiple reflection, and attenuation in the vicinity of the source. Finally, it is common to apply a phase-shifting filter before deconvolution that is given by minimum-phase rearrangement of the Vibroseis wavelet (Gibson and Larner, 1984.) With this source reshaping, a one-sided source filter is appropriate.

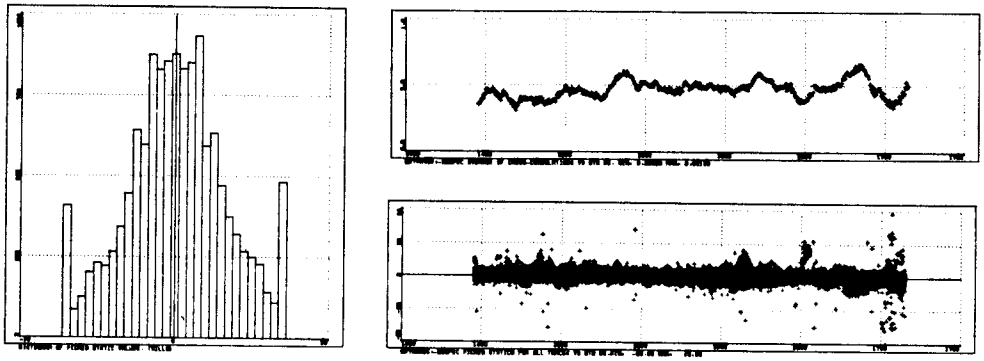
Linear phase

One of the potential benefits of surface-consistent deconvolution suggested by Ronen and Claerbout (1985), is improved residual statics estimation, i.e. determining unknown linear or more general phase shifts. By equalizing spectra surface-consistently, trace-to-trace cross-correlation may be improved without arbitrary realignment of reflectors on the input traces. Surprisingly, they were unable to design surface-consistent linear phase shifts that improved data quality. They argue this didn't work because the stack power, or cross-correlation, that they maximized is not identical to maximizing reflector continuity. The work in this chapter suggests that the problem may also be that they determined their corrections using gather-by-gather estimation rather than simultaneous estimation. An example by Rothman (1985, Fig. 8) supports this explanation.

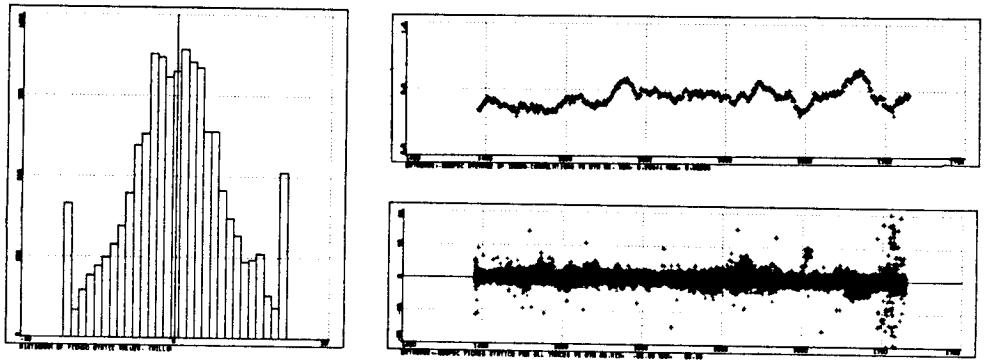
Ronen and Claerbout's result does not mean that surface-consistent deconvolution degrades conventional residual statics estimation. Fig 2.24 shows quality control plots of static picks for the San Joaquin Valley data. These clearly demonstrate that deconvolution, both trace-by-trace and surface-consistent, significantly improved the reliability and distribution of cross-correlation picks.



no deconvolution



trace-by-trace deconvolution



surface-consistent deconvolution

FIG. 2.24. Quality control plots of static time shifts picked for the San Joaquin Valley examples in section 2.6. The attributes displayed are a histogram of picked static values, the distribution of the individual picks, and the average normalized cross-correlation at these picks. We see that deconvolution has reduced the scatter of picked time shifts and increased the average cross-correlation for most of the line.

Unknown phase

Prediction-error filtering is not the only deconvolution method based upon minimizing or maximizing a global measure of goodness. Chief among these are the variable norm methods that try to maximize some measure of spikiness in the output (Gray, 1979.) Because phase is an unknown, these typically require multiple channels in order to reduce the variance of the estimates (Rocca and Kostov, 1987). Thus, it is reasonable to formulate and solve these problem in a surface-consistent setting, but I have not tried to do so for this thesis.

I did try one experiment in unknown phase while studying surface-consistent prediction-error filtering. In this experiment I let the shot filters be two-sided but not constrained to be symmetric. The least-squares design was unstable, with condition number estimates exceeding 10^6, and the quality of the data deteriorated when such filters were applied. This occurred despite diagonal damping that was built into the least-squares design. The shot filters that resulted were nearly symmetric, matched filters to the Klauder wavelet for the correlated sweep. This suggests that additional constraints will need to be incorporated in order to design deconvolution filters with unknown phase.

2.8 SUMMARY

In this chapter I've established that much of the potential power of surface-consistent deconvolution can be lost when implemented by one or multiple gather-by-gather deconvolution passes through the unstacked data. Using some simple tools for nonlinear least-squares, I have developed an effective and efficient surface-consistent deconvolution in the natural $x-t$ coordinates that generalizes conventional trace-by-trace prediction-error filtering. This procedure converged in three to five passes over the data for the field examples I studied.

Two of the examples clearly fit the surface-consistent model, another clearly didn't. It is hard to justify not using surface-consistent deconvolution in the former case, but it happened that trace-by-trace deconvolution was a bit better in one of those example because signal-to-noise was very good and I had not tried to filter out ground roll before deconvolution.

For the marine field example that didn't fit the model, trace-by-trace deconvolution was much better at suppressing water-path multiples. For pegleg multiples, stacking dilutes this gain. In the next chapter I use a different spatial constraint – seafloor-consistency – to improve pegleg suppression.