Chapter 4

Interpolation of near offsets using multiples

As discussed in Chapter 1, 3D marine reflection seismic data are acquired along as many as five axes, most of which are inadequately sampled. Chapter 2 describes how to address this problem by interpolating data with a nonstationary prediction-error filter that is first estimated from fully-sampled training data and then used to interpolate missing data to produce an interpolated output. These training data need not be perfect, and may differ in amplitude and phase but should contain the local multi-dimensional amplitude spectra of the data we wish to recreate. In this chapter, I generate pseudo-primary data by crosscorrelating multiples and primaries in the recorded data (Berkhout and Verschuur, 2003). These pseudo-primary data can be generated at the missing near offsets, but contain many artifacts, so it is undesirable simply to replace the missing data with the pseudo-primaries. Fortunately, many of the problems with the pseudo-primaries do not influence PEF estimation, so a desirable PEF can be obtained from these data, and then used to interpolate the missing near inline offsets to produce a result that is superior to direct substitution of the pseudo-primaries into the missing offsets.

In most data sets, after the over-sampled time axis, the inline receiver axis is the most densely-sampled axis; the receivers are attached to a single cable so the sampling is dense and regular along this cable. Since both the air-gun source and receiver cable are usually towed by the same boat, the maximum inline offset, the largest inline distance between the source and a receiver, is limited by the length of the cable. The near end of the receiver cable is not at the source for obvious reasons; instead the receiver cable is towed a fixed distance behind the source. This distance, the near-offset, is consistent throughout the survey and is typically several times the inline receiver sampling interval. An example of a single shot from a 2D marine survey showing this near-offset gap is in Figure 4.1.

Near-offset traces are particularly valuable. Many methods attempt to recreate zero-offset data from larger offsets; standard multiple-removal techniques, such as surface-related multiple elimination (SRME) (Verschuur et al., 1992), require zero-offset data. Moveout differences between primaries and free-surface multiples are slight at these near-offsets thus compromising the performance of radon-based multiple removal algorithms that discriminate based on differential moveout.

There are many methods that could be used to reconstruct this near-offset gap. A simple way of recreating the missing near offsets is to replace the missing offsets with an NMO-corrected trace from the nearest offset. More sophisticated radon-based methods (Sacchi and Ulrych, 1995; Trad et al., 2002) are commonly used, as are Fourier-based methods (Liu and Sacchi, 2001; Xu et al., 2005). These methods all use existing data recorded adjacent to the missing near offsets to create the missing data, so the performance degrades as the gap increases in size.

Another approach to generating data at the missing near offsets starts by first creating pseudo-primary data. Pseudo-primary data are created by crosscorrelating every trace with every other trace within a shot (Berkhout and Verschuur, 2005). The free-surface multiples correlate with the primaries at lags comparable to times when a primary reflection would arrive if one of the receiver locations was the source. Since the receivers now act as virtual sources, near-offset traces can be generated by crosscorrelating traces from nearby receivers, and zero-offset traces by autocorrelating a single trace. These crosscorrelated traces contain many correlations besides those

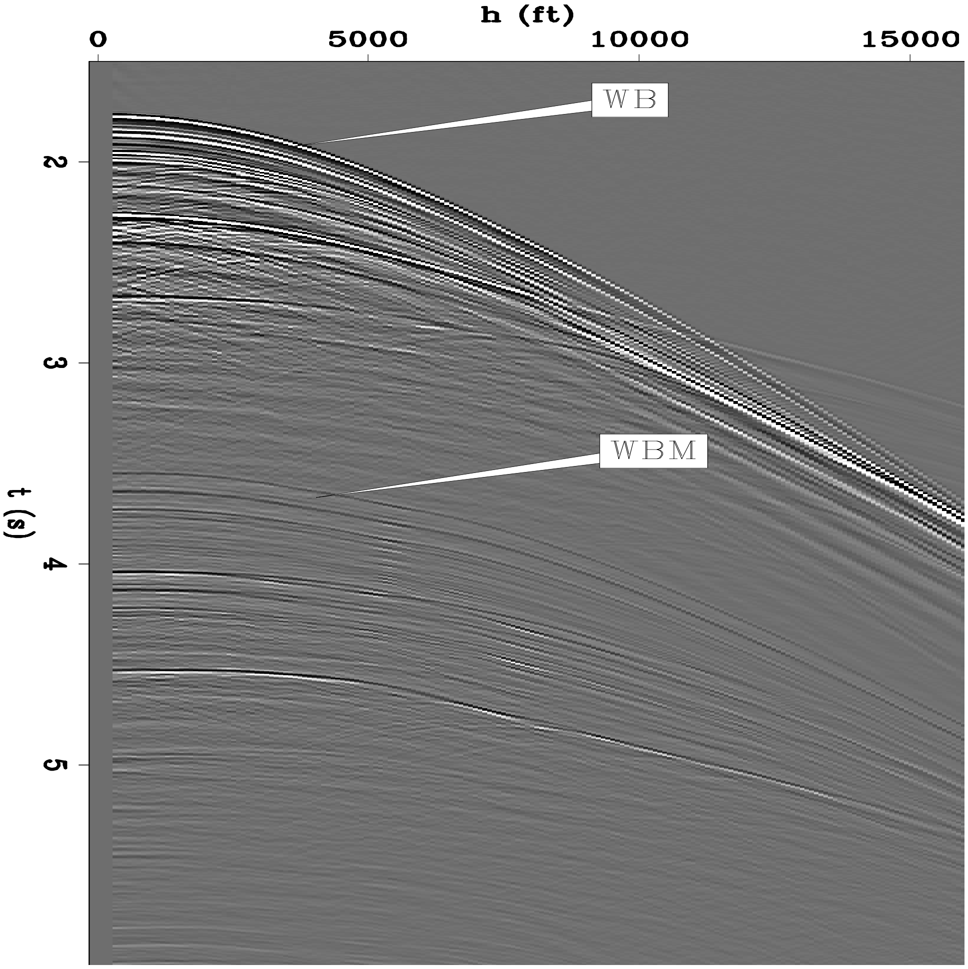

Figure 4.1: A single shot profile from a 2D Gulf of Mexico seismic survey. The nearest offset is at 330 ft. for a near-offset gap of four traces. **ER** Pseudo/. realshotann

between primaries and free-surface multiples or between free-surface multiples and higher-order free-surface multiples, so that the pseudo-primary signal-to-noise ratio for a single crosscorrelation is poor. This can be improved by summing crosscorrelations of the same receiver location pair for many source positions, so the desired pseudo-primary correlations sum while the other correlations interfere destructively. The resulting pseudo-primaries are data that honor the kinematics of the recorded data, but also contain noise, have a different amplitude scale, and have a squared wavelet compared to the recorded data.

Simply substituting these pseudo-primaries for the missing data does not produce an adequate result, as the data contain a squared wavelet, a high level of noise, and spurious events that do not correspond to primary reflections. Instead, here I use the pseudo-primary data as training data for a nonstationary prediction-error filter; whereas the data are inadequate as an interpolation result, they are quite acceptable as training data. The PEF estimation process is relatively insensitive to the phase, amplitude, and squared-wavelet issues that make direct substitution undesirable. This PEF is then used for the interpolation step to fill in the inline near-offset gap.

This approach of using the pseudo-primaries as training data for a PEF can be performed in the time and offset (t-h) domain, the time, offset and source (t-h-s) domain or the frequency, offset and source (f-h-s) domain. In the t-h domain, I interpolate each shot record independently, using a separate non-stationary 2D t-h PEF generated from the corresponding pseudo-primary shot. In the f-h-s domain, the process is performed on overlapping time windows of shot records NMO-corrected using water velocity. A non-stationary 2D PEF is estimated on each source-offset-frequency slice of each time window of the pseudo-primaries, which is then used to interpolate the near-offset gap of the data for the same frequency slice and time window of the original data. The interpolated result is then inverse Fourier transformed back to the time domain, the time windows reassembled, and the NMO correction removed.

GENERATING PSEUDO-PRIMARIES

Multiple reflections are typically viewed as undesired noise to be removed from reflection seismic data. One way to do this is first to predict the multiples and then subtract them from the data. Free-surface multiples can be predicted by autoconvolving data so that the convolution of primaries with themselves creates events at arrival times that are the same as those of multiple reflections with a single bounce point at the free surface. This approach creates free-surface multiple reflections with correct kinematics without the need for any additional subsurface information except the recorded data (Riley and Claerbout, 1976; Reiter et al., 1991; Verschuur et al., 1992).

Autocorrelation can be used to extract synthetic active source data from incoming waves that reflect at the free surface and return within the recording array (Claerbout, 1968; Cole, 1995; Schuster, 2001; Artman, 2007). This has traditionally been though of in a passive context, where random noise is assumed to be arriving from all locations. The reflecting waves from active-source experiments can be treated in the same manner, where the primary reflections correlate with the free-surface multiples (Reiter et al., 1991; Berkhout and Verschuur, 1994, 2003; Shan, 2003). Autocorrelating data, d, (implemented as multiplication of complex-conjugates in the ω domain) for two receiver points, r_1 and r_2, both for a single shot, s, gives the pseudo-primaries, p,

$$p(s, r_1, r_2, \omega) = d(s, r_1, \omega)\bar{d}(s, r_2, \omega). \tag{4.1}$$

One of the receiver coordinates, r_1, becomes the virtual source location, while the other receiver coordinate, r_2, remains the receiver location, or vice-versa, and the $\bar{d}$ denotes the complex-conjugate of d. This is similar to 2D surface-related multiple prediction, where instead of cross-correlation the data are convolved with itself, so that the convolution of primaries with primaries produces surface-related multiples (Verschuur et al., 1992).

Figure 4.2a shows an example of a fully-sampled split-spread shot from the Sigsbee2B data set, while Figure 4.2b shows a slice of the pseudo-primary output of

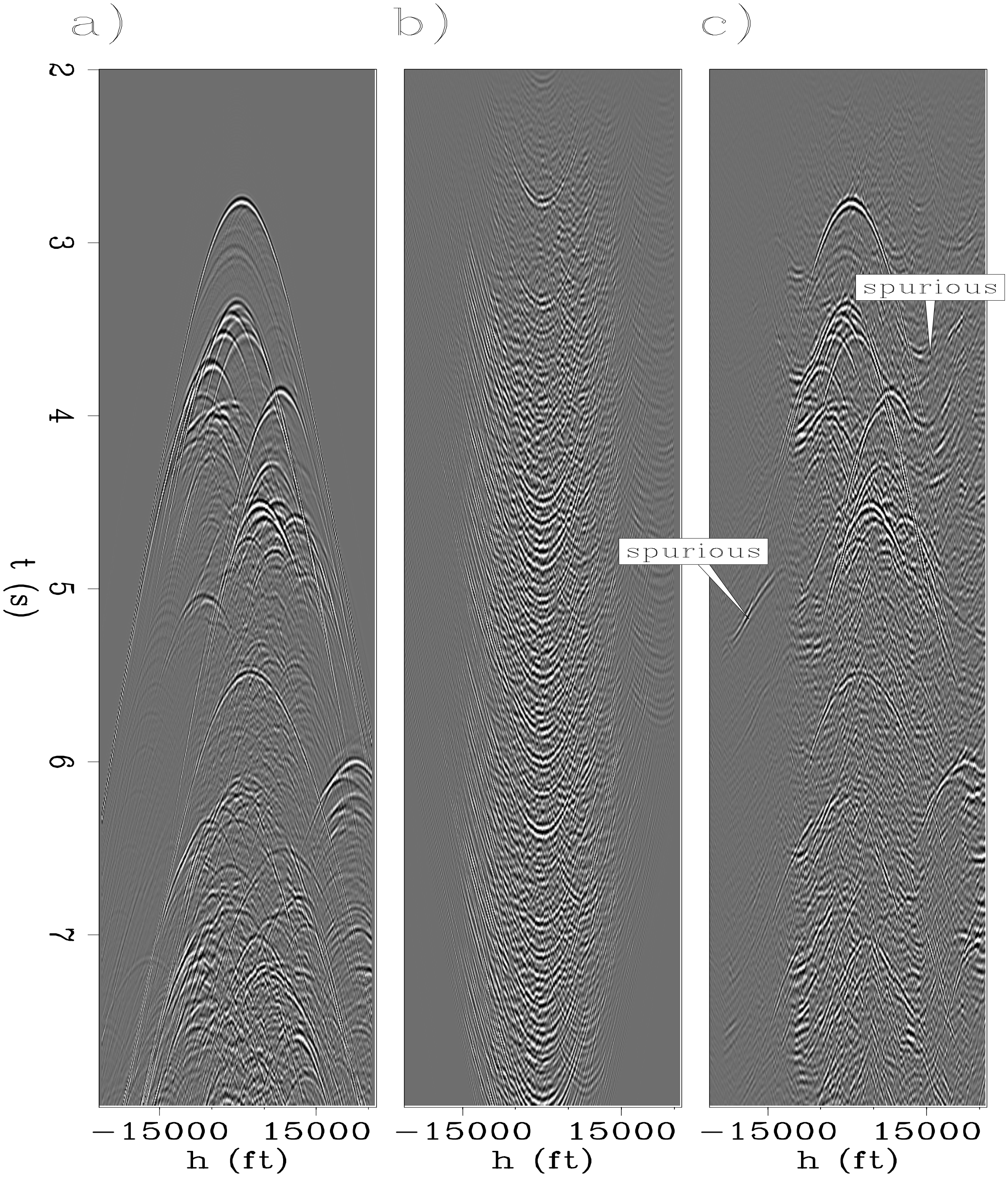

Figure 4.2: Crosscorrelation of a single shot. (a): Original fully-sampled split-spread shot; (b): The same shot recreated from crosscorrelating the traces in (a) using equation 4.1. (c): using equation 4.2 correlations are made for 496 shots and are then summed. The quality of the pseudo-primaries is poor for a single shot, but improves after the summation of many shots. All data are scaled by t. **ER** Pseudo/. sigsbee1shotann

autocorrelating the zero-offset trace with all of the other traces within that single shot. Clearly, the original shot and the pseudo-primaries generated from the autocorrelation are different. The water-bottom reflection at zero offset is present as are the first group of diffractors, but they are not present at the more distant offsets. The many other correlations between events produce undesired noise in the output. Primaries can correlate with other primaries, multiples with multiples, and noise with other noise or signal. Much of these undesired correlations, however, vary as a function of source position, s, so when the correlations for the same receiver pair r_1 and r_2 are performed for many different source locations, s, and are then summed, the unwanted events destructively interfere, while the correct pseudo-primaries constructively interfere.

$$p(r_1, r_2, \omega) = \sum_s d(s, r_1, \omega)\bar{d}(s, r_2, \omega). \tag{4.2}$$

As shown in Figure 4.2c, where 496 sources were used, this summing over multiple source positions thus can greatly improve the pseudo-primary signal. The range of usable offsets is greatly improved, as is the signal-to-noise ratio. The pseudo-primaries now look more similar in character to the original data.

Pseudo-primaries generated from recorded multiples are interesting in part because they have different illumination than do the recorded primaries. Figure 4.3a illustrates a desired near-offset primary ray-path that is not recorded because of the gap between the source and the nearest receiver. In Figure 4.3b, the source is positioned such that the raypath first travels from the source to the water-bottom and back up to the water surface within the recording array. This ray then reflects back into the subsurface and eventually returns into the recording array at another receiver. This four-segment raypath is a free-surface multiple, which can be reconsidered as two distinct events: the first a recorded primary event from the source s to the first receiver r_1 and the second leg as another primary from the virtual source r_1 to another receiver r_2. Crosscorrelating the multiple recorded at r_2 with the primary recorded at r_1 produces a pseudo-primary trace with a virtual source at r_1 and a receiver at r_2. Comparing Figures 4.3a and 4.3b, we see that we can transform multiples that we record into pseudo-primaries with virtual source locations where we did not originally record data. This is most useful at unrecorded near offsets.

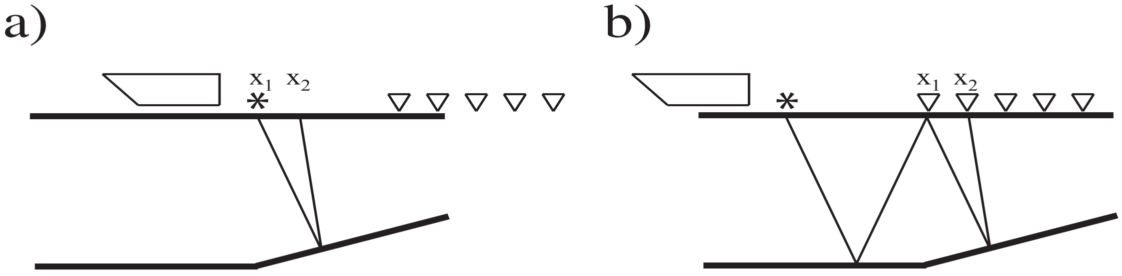

Figure 4.3: Raypaths of a primary and multiple reflection. (a): an unrecorded primary reflection that hits the surface at near offset. (b): a recorded multiple that first reflects at the surface within the recording array, then returns within the recording array. **NR** Pseudo/. nearoffsetcartoon

The generated pseudo-primaries differ from the recorded data in several ways. First, the crosscorrelation of the data squares the amplitude spectrum of the pseudo-primaries, meaning that the wavelet of the pseudo-primaries will become zero-phase, different than that of the input data. Second, the amplitudes of the pseudo-primaries will differ from the amplitudes of the actual primaries, as the pseudo-primaries are a correlation and summation of different events. This amplitude difference will be both on a global scale difference between the pseudo-primaries and the primaries, as well as different relative amplitudes within each data set. Third, spurious events from other correlations may still exist in the data as the number of sources is limited.

Because of these differences between the pseudo-primaries and primaries, a direct substitution of the created near-offset pseudo-primaries is not adequate. The pseudo-primaries, however, can be used as training data for a nonstationary prediction-error filter, as from Chapter 2 we know that the PEF is insensitive to the phase of the training data. This PEF is then used to interpolate the missing near offsets, so that the negative aspects of the pseudoprimaries, such as the incorrect wavelet and extra noise, are ignored, while the positive aspects of the pseudo-primaries, contained in the autocorrelation of these data, are used.

INTERPOLATING WITH NONSTATIONARY FILTERS IN T-X OR IN F-X

Now that an adequate training data set has been found, we can use the interpolation method as described in Chapter 2. Recall that a nonstationary prediction-error filter is first estimated from fully-sampled *training* data and is then used to interpolate *missing* data. The training data required by this method should have the same local autocorrelation as that of the desired output interpolated data. Pseudo-primaries, having roughly the same dip as the missing data, serve as training data for a PEF. Some of the problems with the pseudo-primary data, such as the different phase and amplitude, do not influence the PEF.

Reviewing equation 2.21,

$$\begin{array}{rcl} & & \min_{\mathbf{f}_{\rm ns}} \|\mathbf{r}_{\rm d}\|^2 + \epsilon^2 \|\mathbf{r}_{\rm f}\|^2, \text{where} \\ \mathbf{r}_{\rm d} & = & \mathbf{D}_{\rm ns}\mathbf{K}_{\rm ns}\mathbf{f}_{\rm ns} + \mathbf{d} \qquad . \\ \mathbf{r}_{\rm f} & = & \mathbf{R}\mathbf{f}_{\rm ns} \end{array} \tag{4.3}$$

Here, the unknown nonstationary PEF coefficients, $\mathbf{f}_{\rm ns}$, are estimated from the pseudo-primary data, $\mathbf{d}$, a convolutional matrix, $\mathbf{D}_{\rm ns}$, that is a function of $\mathbf{d}$, and a regularization operator, $\mathbf{R}$, that applies a Laplacian filter across the spatial axes of the PEF coefficients. This system of equations is solved to estimate a multi-dimensional, nonstationary prediction-error filter from a set of fully-sampled pseudo-primaries, such as was generated in the previous section.

Once this filter has been estimated from the pseudo-primary data, the filter is used in a second least-squares problem. In this problem, we estimate an interpolated output, $\mathbf{m}$, composed of missing data, $\mathbf{m}_{\rm unknown}$, and known data, $\mathbf{m}_{\rm unknown}$, that when convolved with the nonstationary PEF produces a minimized output, as in equation 2.29,

$$\mathbf{r}_{\rm m} = \mathbf{F}_{\rm ns}\mathbf{m}_{\rm unknown} + \mathbf{F}_{\rm ns}\mathbf{m}_{\rm known}. \tag{4.4}$$

Here the known nonstationary PEF convolution matrix, $\mathbf{F}_{\rm ns}$, derived from $\mathbf{f}_{\rm ns}$ obtained in the previous step, is multiplied with both the known and unknown data values, $\mathbf{m}_{\text{known}}$ and $\mathbf{m}_{\text{unknown}}$, to create a known quantity and a term with the unknown values to be interpolated. These terms are summed to form the residual $\mathbf{r}_{\text{m}}$. We minimize the L_2 norm of this residual to estimate the interpolated data values.

Interpolation in time and space

The output pseudo-primaries from equation 4.2 are in frequency, source position, and receiver position. We can reorganize this to time, offset, and source position, and then estimate a nonstationary PEF, either shot-by-shot by estimating 2D PEFs in time and offset solving equation 4.3 for each shot, or on the entire data set with a single 3D PEF in time, offset, and source solving a much larger version of equation 4.3. This PEF, or series of PEFs, is then used to fill in the missing near-offset gap using equation 4.4 either once or for each shot.

Interpolation in frequency and space

Interpolation in the frequency domain requires a different approach than in the time domain. Spitz (1991) shows that 2D plane waves can be predicted for a single frequency by using a 1D spatial prediction filter, because a single plane wave at each frequency appears as a complex sinusoid in space. The wavenumber of this sinusoid increases as a function of frequency, so each frequency requires a unique 1D PEF. Data containing a combination of plane waves appear as a summation of complex sinusoids at each frequency, which can still be predicted by a reasonably short 1D PEF. In three dimensions, this filter would be a 2D filter, in four dimensions a 3D filter, and so on, making the frequency-based approach a series of smaller problems whereas the same region in time would be solved as a single larger problem.

Nonstationarity, discussed in Chapter 2, is addressed in a slightly different manner in f-x. Along the spatial axes, a nonstationary PEF can be used to capture slopes that change as a function of position. Since a Fourier transform is performed on the time axis to convert both the training and interpolated data to the frequency domain, we implicitly assume that the slopes of the plane waves do not vary as a function of time.

I address this problem of time nonstationarity by breaking up the problem into overlapping time windows that we assume to be stationary. First, perform a water-velocity normal move-out correction on both the pseudo-primaries and the original recorded data to (roughly) flatten them, and then break up the data into overlapping windows along the time axis. I perform the NMO to reduce the amount of energy crossing the boundaries between patches. From there, each time window of both the pseudo-primaries and the recorded data is Fourier transformed along the time axis, so that the data are sorted into source, offset, frequency, and time window. A unique 2D nonstationary complex-valued PEF is estimated in source and offset on each frequency of each time window of the pseudo-primary data by solving equation 4.3. This nonstationary source-offset (h-s) PEF is then applied to fill the missing data in offset on the corresponding time window and frequency of the recorded data by solving equation 4.4, and this series of problems is repeated for each frequency of each time window.

After the data are interpolated, they are first inverse-Fourier transformed back to time, then the time-windows are reassembled with appropriate weighting in overlapping regions, and finally are inverse NMO-corrected to return the data to their original form of time, source, and offset.

SIGSBEE DATA EXAMPLE

In order to test the effectiveness of this pseudo-primary-based method, I use a synthetic example where we already know the answer. To create a challenging test case, I take a split-spread version of the Sigsbee2B synthetic data set, with a zero-offset section and a shot gather shown in Figures 4.4a and 4.4b, respectively, and remove the nearest 2100 ft of offset on either side, for a total of 4200 ft of missing offset, or a gap of 29 traces at the 150 ft sampling in offset, shown in Figure 4.4c. While this gap is overly large for a single boat, a two-boat "undershooting" of an offshore platform could have gaps this large.

Let us first examine the cross-correlated pseudo-primaries prior to the summation in equation 4.2 for a single output trace from one receiver pair for all shots, both using the fully sampled input data, shown in Figure 4.5b, and using the data missing the near offsets, shown in Figure 4.5c. This would correspond to $p(s, r_1 = 41000, r_2 = 41000, t)$ in equation 4.1. I refer to these images as pseudo-primary-contribution gathers, as they fulfill a role similar to multiple-contribution gathers in the SRME algorithm (Dragoset and Jericevic, 1998). The summations of these gathers, the predicted pseudo-primary traces, are shown in Figure 4.5a, plotted alongside the original trace. We see that the traces in Figure 4.5a look quite similar, especially the two pseudo-primary traces, so the missing sources at near-offsets did not significantly detract from the result.

Moving up from the single output trace in Figure 4.5, we now look at an entire zero-offset section in Figure 4.6. Figure 4.6a shows the original zero-offset section while 4.6b shows pseudo-primaries generated using fully-sampled data, and 4.6c shows pseudo-primaries generated from data with missing near offsets.

Note four important differences between Figures 4.6a and 4.6c. First, the pseudo-primaries have different illumination than do the true primaries, so the relative amplitudes within each image differ. The amplitude of the water-bottom reflection in the pseudo-primary image with limited offsets in Figure 4.6c is more variable than is that in the original image in Figure 4.6a, or even that in the pseudo-primaries generated with all of the offsets in Figure 4.6b. Also, the subtle reflections below the water bottom on the left side of the image at 4-5 s are much less pronounced when the input offsets to the pseudo-primary generation are limited. Second, the pseudo-primaries compared to the original data exhibit cross-talk. The cross-talk is composed of both coherent events such as those above the water bottom and water-bottom multiple in Figure 4.6c, and more random noise. This cross-talk increases only slightly when the offsets are limited in the input to the pseudo-primary generation. Third, the wavelet of the pseudo-primary data differs from that in the original data as a result of the

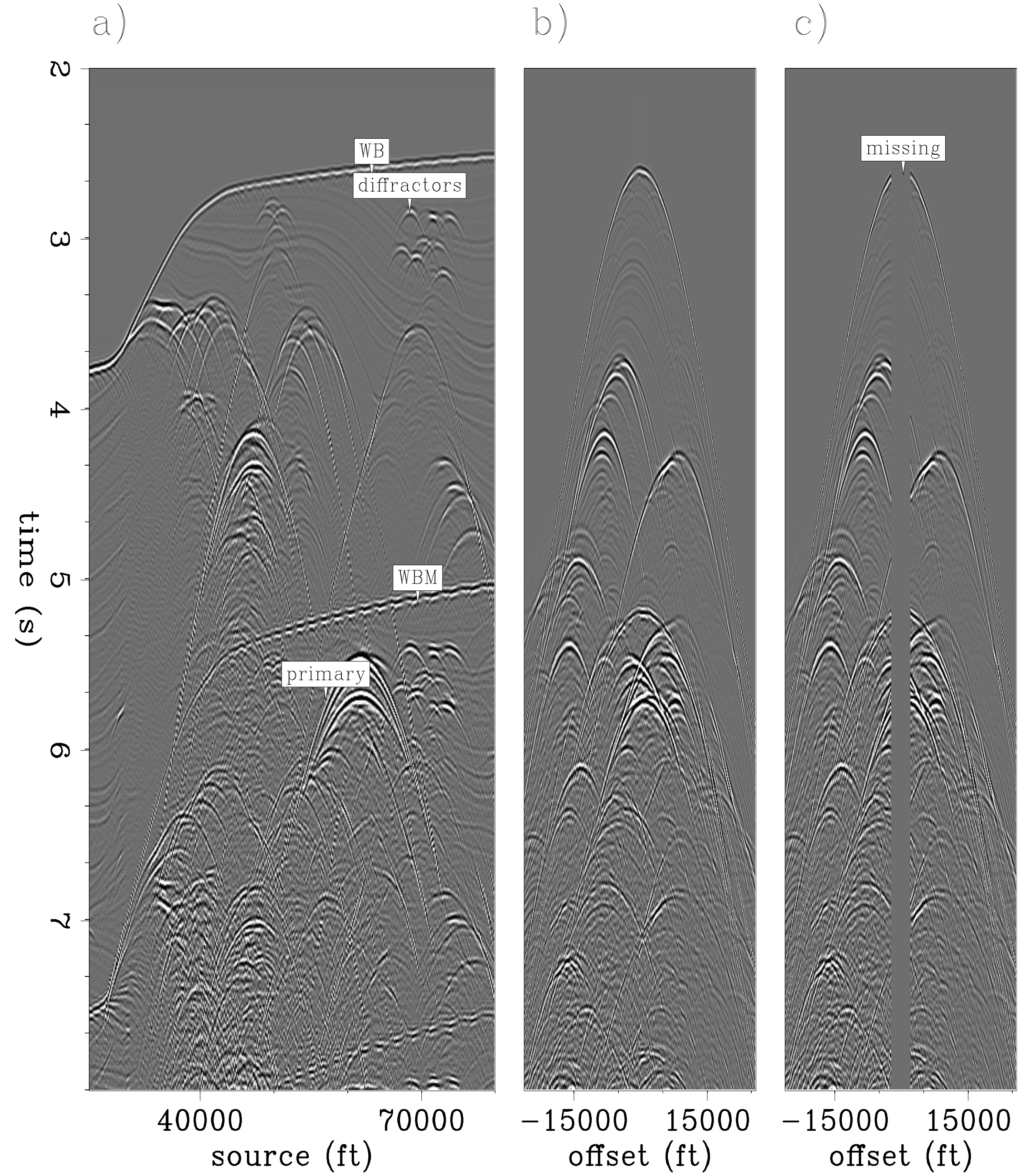

Figure 4.4: Original Sigsbee data. (a): zero-offset section. (b): one shot. (c): resampled shot. The near 4200ft or 29 traces of offset were excluded. All figures have amplitude scaled by $t^{0.8}$. **ER** Pseudo/. splitspreadann

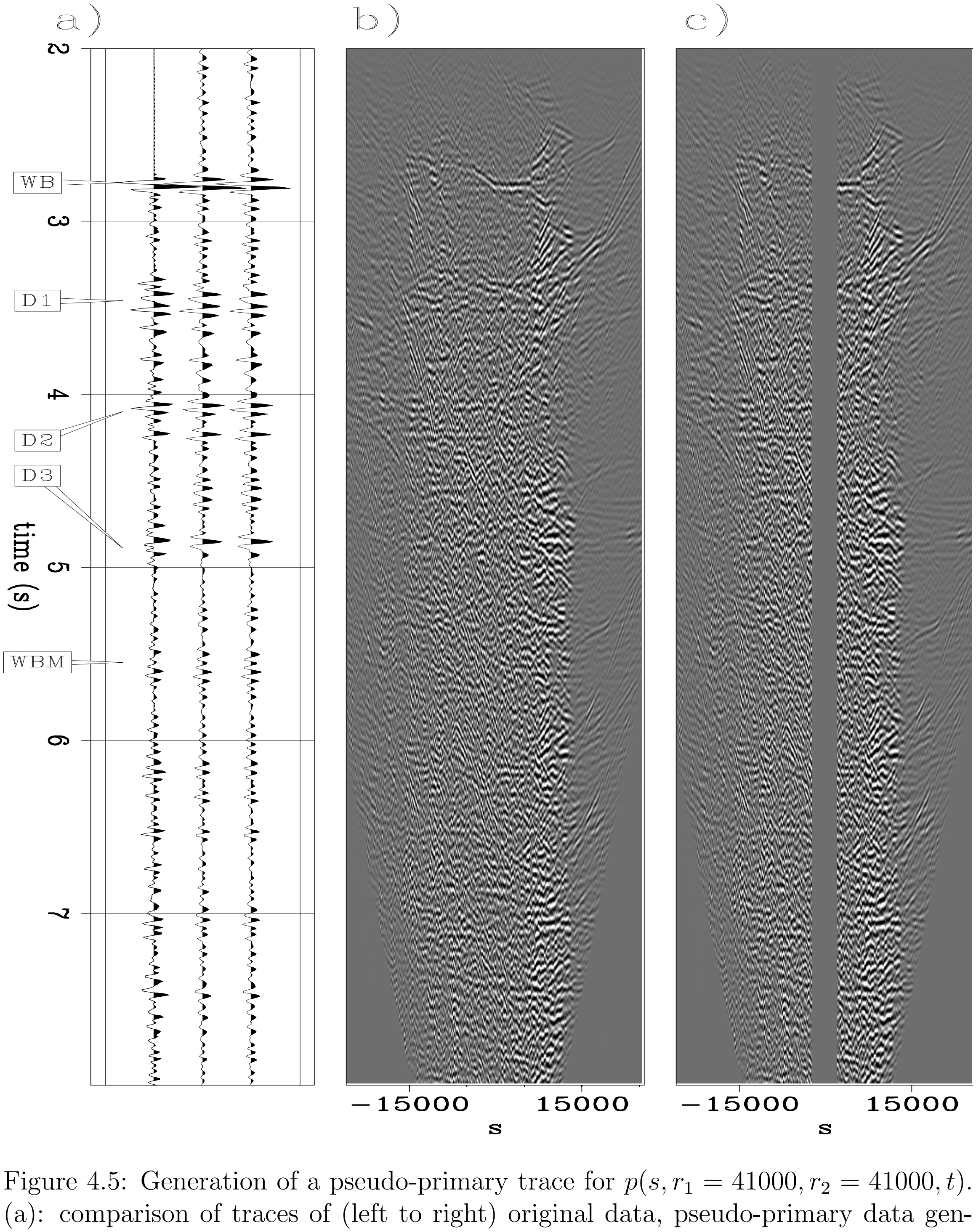

Figure 4.5: Generation of a pseudo-primary trace for $p(s, r_1 = 41000, r_2 = 41000, t)$. (a): comparison of traces of (left to right) original data, pseudo-primary data generated from fully-sampled input data, and pseudo-primary data generated from data with missing near offsets. (b): A pseudo-primary contribution gather, where the horizontal axis is shot location s. (c): The same pseudo-primary contribution gather as (b), but with the missing near offsets. The images are scaled by $t^{0.8}$ for display purposes. **CR** Pseudo/. pseudocontribann

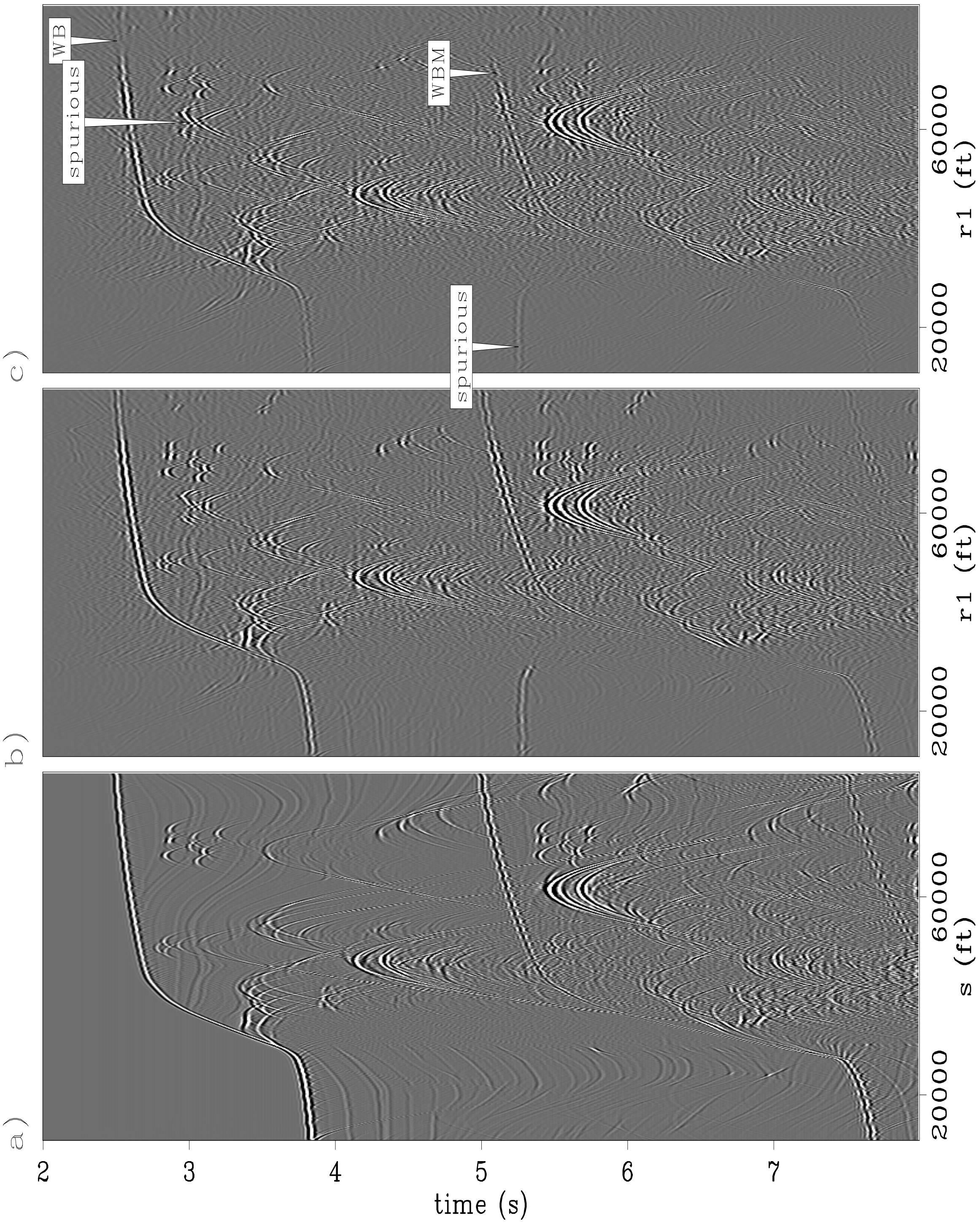

Figure 4.6: A comparison of zero-offset sections: (a) original data, (b) pseudo-primaries generated using all offsets, (c) pseudo-primaries generated using all offsets other than the missing near offsets. The quality of the pseudo-primaries degrades slightly with missing near offsets. The images are scaled by $t^{0.8}$ for display purposes. **CR** Pseudo/. pseudocontribsliceann

cross-correlation of the primary wavelet and the multiple wavelet. Finally, the amplitude scale of the pseudo-primary images (4.6b-c) is roughly a factor of 20 higher than in the original data.

Figure 4.7 shows the nearest 10000 ft of offset of the pseudo-primaries generated from all of the split-spread shots from Figure 4.4c that are missing the near offsets. The common-offset section on the left panel is at zero offset. Looking at the shot gather on the right panel, we see that the water-bottom reflection is the strongest event at early times, but a considerable amount of cross-talk is present before the water-bottom reflection. The time slice shown on the top panel of Figure 4.7 shows that in addition to the desired reflections and diffractions extracted from the multiple reflections, there are also strong spurious events that are not easily identifiable as cross-talk; in particular, note the event at roughly 60000 ft and near offsets.

The most straightforward way to use these pseudo-primaries to interpolate the missing near offsets would be to replace the missing traces with the pseudo-primaries, shown in Figure 4.8. The pseudo-primaries were scaled by a factor of 0.05 to match the mean amplitude of the sampled data. The cross-talk and squaring of the wavelet are both obvious in this Figure 4.8, making the region of interpolated data is easily distinguishable, as do the spurious events, and the extra slope in the more complex areas. As we see next, we can improve greatly on this result by using the pseudo-primaries as training data for a prediction-error filter.

Interpolation of Sigsbee in time and offset

A single output shot record of the pseudo-primaries ($p(r_1, r_2, t)$) generated in Figure 4.7 is used as the training data ($\mathbf{d}$) in equation 4.3 to generate a nonstationary PEF, $\mathbf{f}_{\rm ns}$. The regularization operator in equation 4.3, $\mathbf{R}_{\rm ns}$, is a two-dimensional Laplacian that operates over time and offset for each local filter lag. Once estimated, the PEF is then used to interpolate the missing near offsets of the sampled data, $\mathbf{m}_{\rm unknown}$, using equation 4.4. This process is repeated for all of the shots of both the pseudo-primary data and the original sampled data.

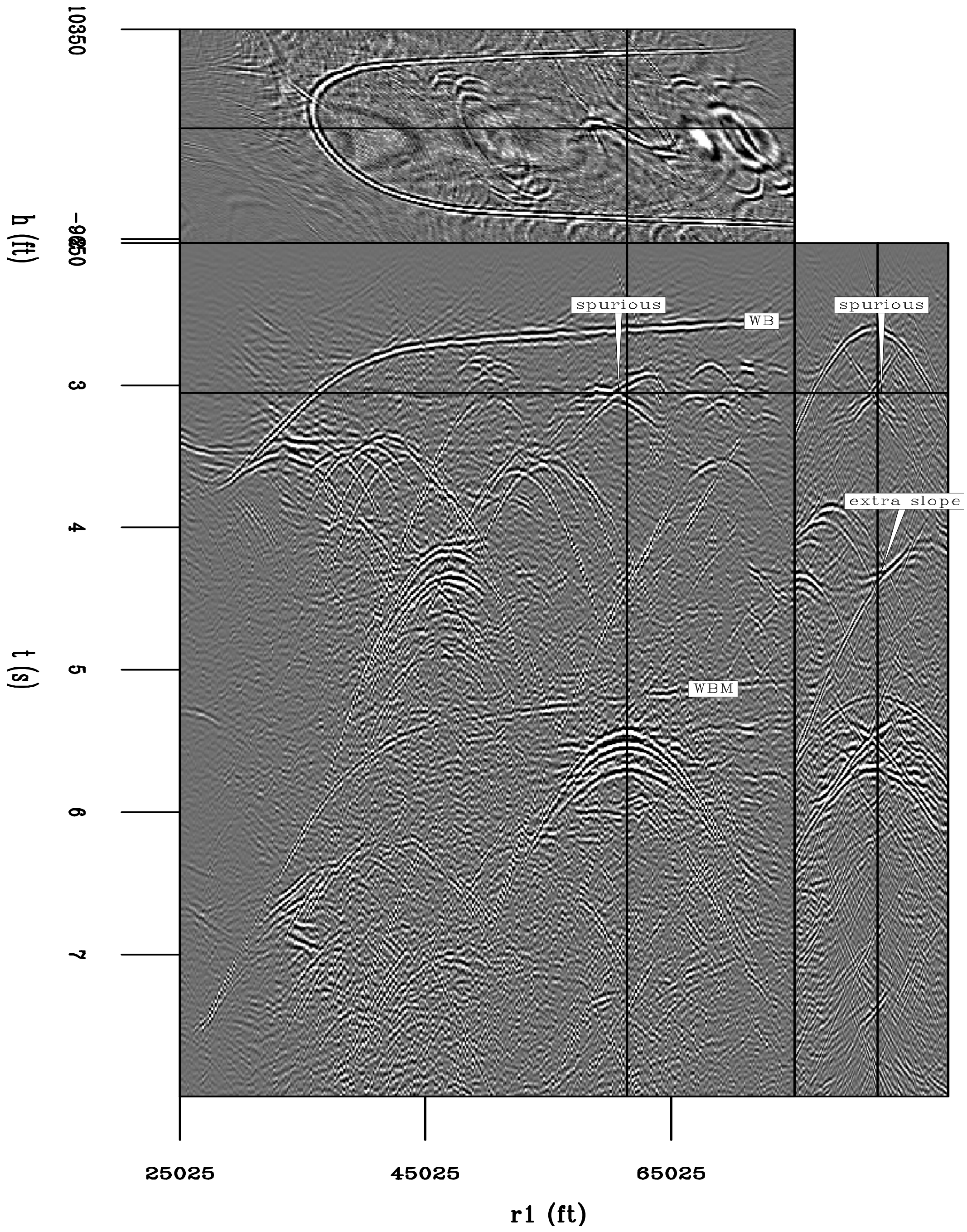

Figure 4.7: The near 10000 ft of offsets of pseudo-primaries generated from input data missing the 2000 ft of near offsets. The front panel is a constant-offset section, the right panel is a single shot, and the top panel a time slice. The image is scaled by $t^{0.8}$ for display purposes. **CR** Pseudo/. pseudoann

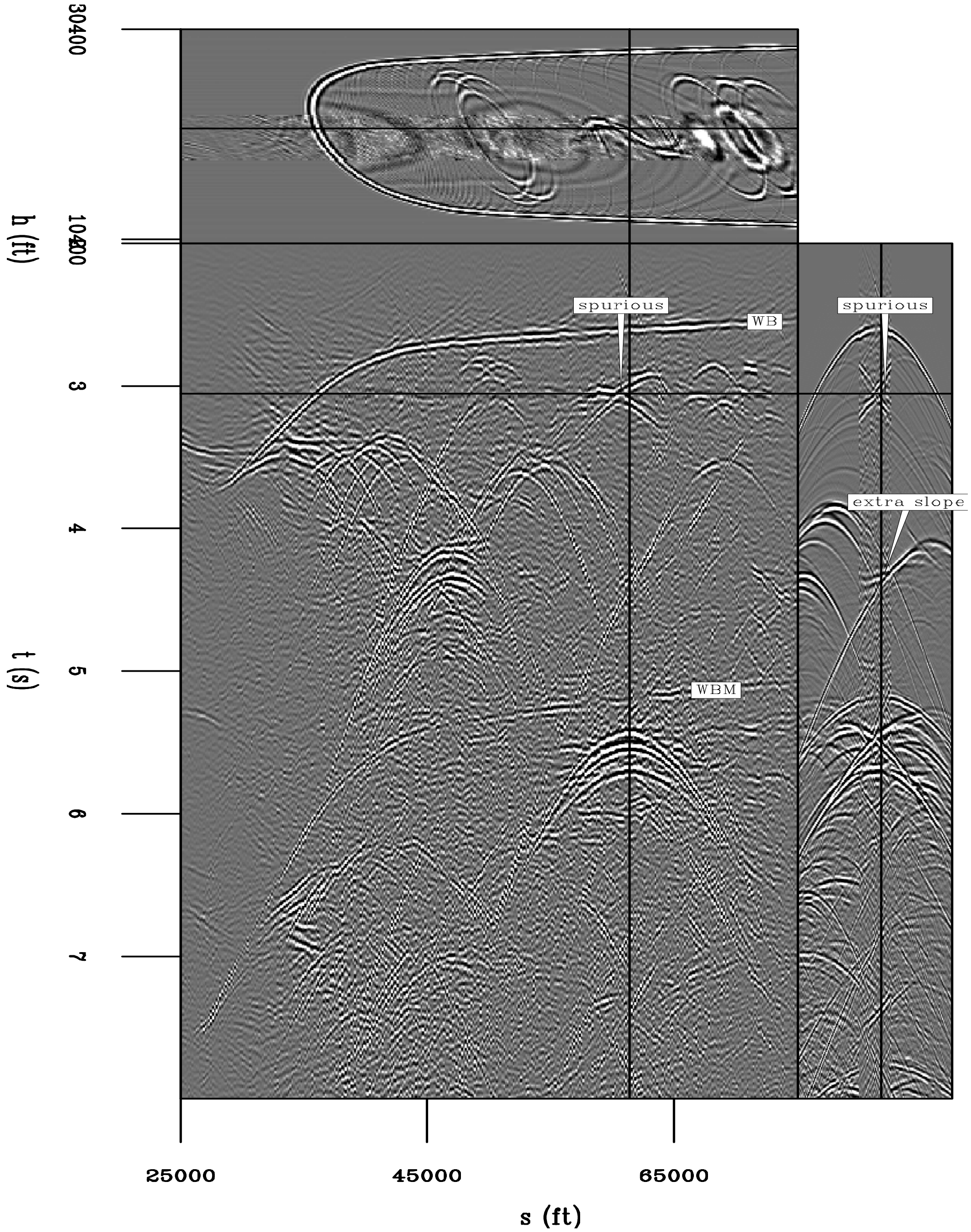

Figure 4.8: Original data missing the near offsets, with pseudo-primaries spliced into the locations where traces were missing. Note three points of interest, a spurious event caused by the correlation of primaries with other primaries, seen in both the constant-offset section and the shot, and an area with crossing events and noise. The image is scaled by $t^{0.8}$ for display purposes. **CR** Pseudo/. pseudocutann

Figure 4.9 was generated by interpolating the near-offset gap using a *t*-*h* nonstationary PEF estimated on each shot record of the pseudo-primary data. Each 2D PEF has ten points in time and six points in offset, and varies every four points on each axis for a total of 76000 free coefficients for each shot. I applied 400 iterations of the conjugate-direction solver to solve both the filter-estimation and interpolation problems.

There are several interesting things to see in the *t*-*h* interpolated result in Figure 4.9. First, no cross-talk is present in the interpolation, such as a spurious event. This is one of the benefits of using the *t*-*h* PEF-based approach, since low amplitude at the edges of the known data results in no large residuals in equation 4.4, regardless of the crosstalk present in the training data for the PEF. Second, the events present in the original data in a single shot shown in the right panel have been interpolated, in most cases with good results, although the quality of the interpolation degrades deeper in the section. This can largely be traced to the quality of the input pseudo-primaries. It also appears that problems with the relative amplitudes in the pseudo-primaries are amplified in this result. Third, the common-offset section shows significant differences from shot to shot. This is because a 2D PEF was used and each shot was interpolated independently. This is different when a 3D PEF in time, offset *and shot* is used, where correlations between shots are be used and the inconsistencies between shots penalized, discussed next. Finally, the wavelet issues in the pseudo-primaries appear to be mostly removed in the *t*-*h* interpolated result.

While the *t*-*h* approach gives a reasonable result when viewing a single shot, the extremely choppy constant-offset section shows one limitation of the *t*-*h* approach. I next use a 3D nonstationary *t*-*h*-*s* PEF that is $10 \times 5 \times 5$ elements and varies every 10 elements on the time axis, 3 elements on the offset axis, and 4 elements on the source axis. I solve for this PEF using 100 iterations of a conjugate-direction solver. Once estimated, I use this PEF, with 121 million coefficients, to interpolate the missing data, using 200 iterations of a conjugate-direction solver on equation 4.4, with a starting guess for the unknown data created from the nearest recorded offset from the same midpoint NMO-corrected to the missing offset.

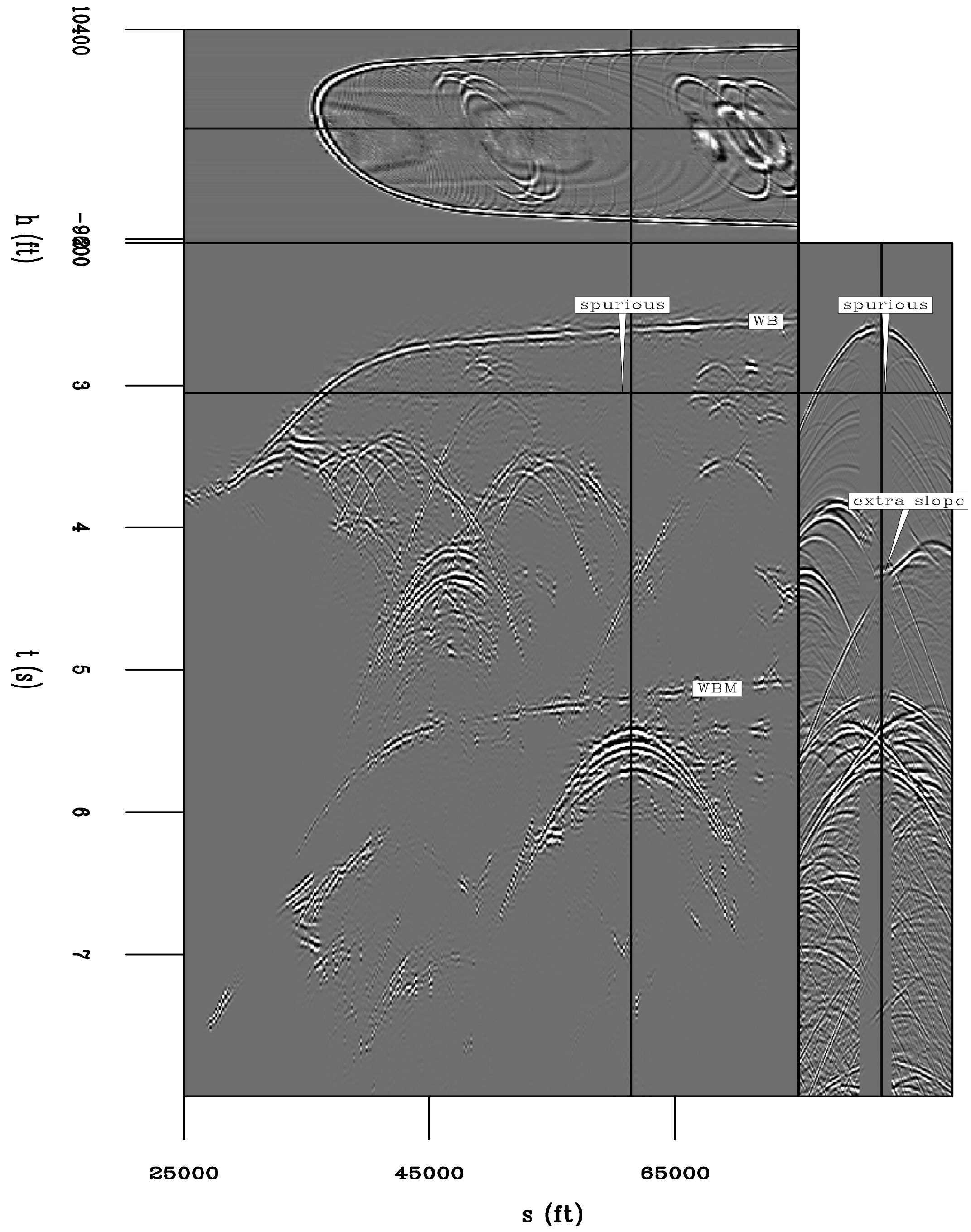

Figure 4.9: Interpolation with pseudo-primaries and a t-h PEF. The front panel is an interpolated constant-offset section, the right panel is a single shot record, and the top panel a time slice. The spurious event is not present, but the region with conflicting slopes is only partially interpolated, with an incorrect slope present. The image is scaled by $t^{0.8}$ for display purposes. **CR** Pseudo/. txinterpann

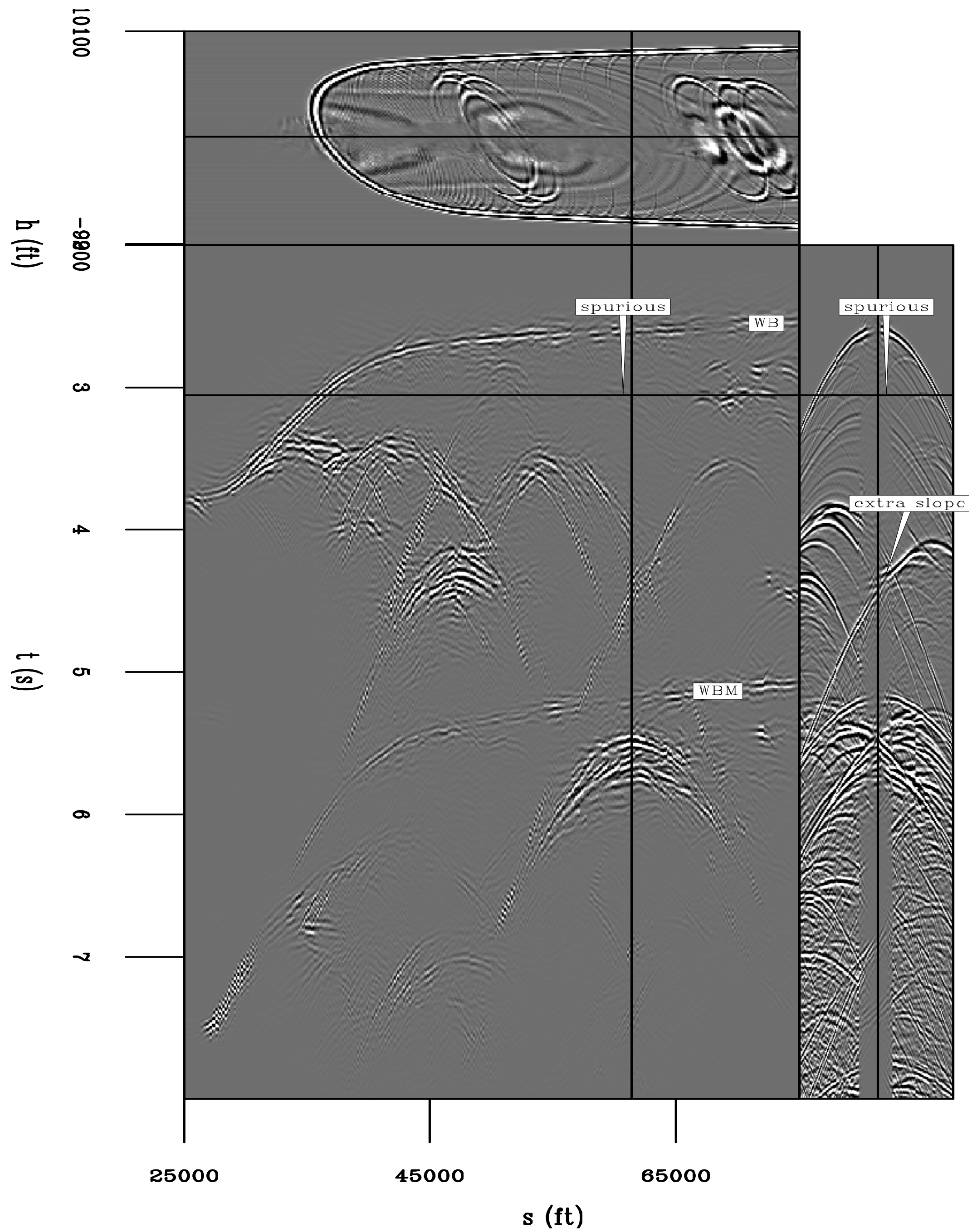

Figure 4.10: Interpolation with pseudo-primaries and a 3D *t*-*h*-*s* PEF. The front panel is an interpolated constant-offset section, the right panel is a single shot record, and the top panel a time slice. The spurious event is not present and the region with crossing slopes is better interpolated than in the 2D example. The water-bottom is more variable in amplitude, but the constant-offset section is more consistent from shot to shot. The image is scaled by $t^{0.8}$ for display purposes. **CR** Pseudo/. txyinterpann

This 3D result, shown in Figure 4.10, is in part an improvement over the 2D approach. The roughness of the 2D approach along the source axis is gone, with a smoother result that contains slightly more noise. The spurious event is still gone, and the the diffractions in the constant-offset section are more continuous, although the water-bottom amplitude is more variable. Next I show that using a 2D PEF on frequency slices provides most of the benefits of using a 3D *t*-*h*-*s* PEF, but with a much lower computational cost and memory requirement.

Interpolation of Sigsbee in frequency, offset, and shot

Figure 4.11 shows the result of near-offset interpolation using 2D PEFs in the *f*-*h*-*s* domain. Here the training and sampled data were first NMO-corrected at water velocity, then were broken into overlapping windows along the time axis. Both sets of these time windows were then Fourier transformed to frequency. Each frequency slice of each window is treated independently, where we estimate a nonstationary PEF on a frequency slice of the pseudo-primaries, solving equation 4.3 with the frequency slice of the pseudo-primaries serving as training data ($\mathbf{d}$) to generate a nonstationary complex-valued PEF $\mathbf{f}_{\rm ns}$. This PEF is then used in equation 4.4 as $\mathbf{F}_{\rm ns}$, where $\mathbf{m}_{\rm known}$ is the corresponding frequency slice of the recorded data, and $\mathbf{m}_{\rm unknown}$ are the missing near offsets of this frequency slice.

In Figure 4.11 the time axis (1500 samples long) of both the input sampled data and the pseudo-primaries were broken into 40 overlapping windows of 64 points each. Both sets of windows were then Fourier transformed to produce 2560 2D frequency slices. Each frequency slice is treated as a separate problem, with a 2D nonstationary PEF estimated on the pseudo-primary frequency slice; the 2D PEF was four samples long on both the offset and shot axes, and the filter varied every four points on each axis, for a total of 67456 filter coefficients for each 496-shot by 136-offset frequency slice, estimated with 100 iterations of a conjugate-direction solver. The data were then interpolated using 200 iterations of a conjugate-direction solver on equation 4.4, again with an initial model of the missing data an NMO-corrected copy of the trace from the same midpoint and the nearest offset.

The frequency-domain interpolation in Figure 4.11 differs in several respects from the *t*-*h* domain interpolation in Figure 4.9. First, the *f*-*h*-*s* result still has some crosstalk present, while the *t*-*h* and *t*-*h*-*s* results have almost no crosstalk from spurious events. This is most visible prior to the water-bottom reflection. I believe this is in part due to the stationarity assumption within each time window. However, the large spurious event at a (1) and (2) is not present in the result in either case. Second, the *f*-*h*-*s* result shows less shot-to-shot variation than does the *t*-*h* result, and less amplitude variability than the *t*-*h*-*s* result. This is because in the *t*-*h* result each shot is interpolated separately as an independent problem, while the *f*-*h*-*s* interpolation estimates a PEF that spans both the offset and source axes, minimizing this jitter. Third, there is some ringing present in the *t*-*h* result that is not present Finally, the *f*-*h*-*s* result appears to contain more detail in the result than does the *t*-*h* result. In particular, events below the water-bottom in the shot that were not interpolated in the *t*-*h* case, in the right panel of Figure 4.9, were interpolated in the *f*-*h*-*s* case on the right panel of Figure 4.11.

These results can also be viewed after further processing. Figure 4.12a shows the zero-offset of a shot-profile migration of the fully-sampled data with the correct velocity. Figure 4.12b shows a multiple prediction based upon autoconvolution of the fully-sampled data, followed by the same migration as in 4.12a. The multiples kinematically match those in 4.12a, but the amplitudes are boosted in order to observe differences. The deformation of the multiples in the center of the image is due to the salt body present in the velocity model used in the migration. The same migrated multiple model is shown in 4.12c, but the multiples were generated with data missing the near 4000 ft of offsets, while 4.12d shows the migrated multiple model with these near offsets replaced by the *f*-*h*-*s* interpolation result in Figure 4.11. The missing near offsets in 4.12c significantly degrades the multiple model, as the water bottom multiple is uneven and the multiples near the edges of the salt body are different. The multiples generated from the *f*-*h*-*s* interpolation are much more similar to the fully-sampled multiples than to the multiples generated without the near offsets.

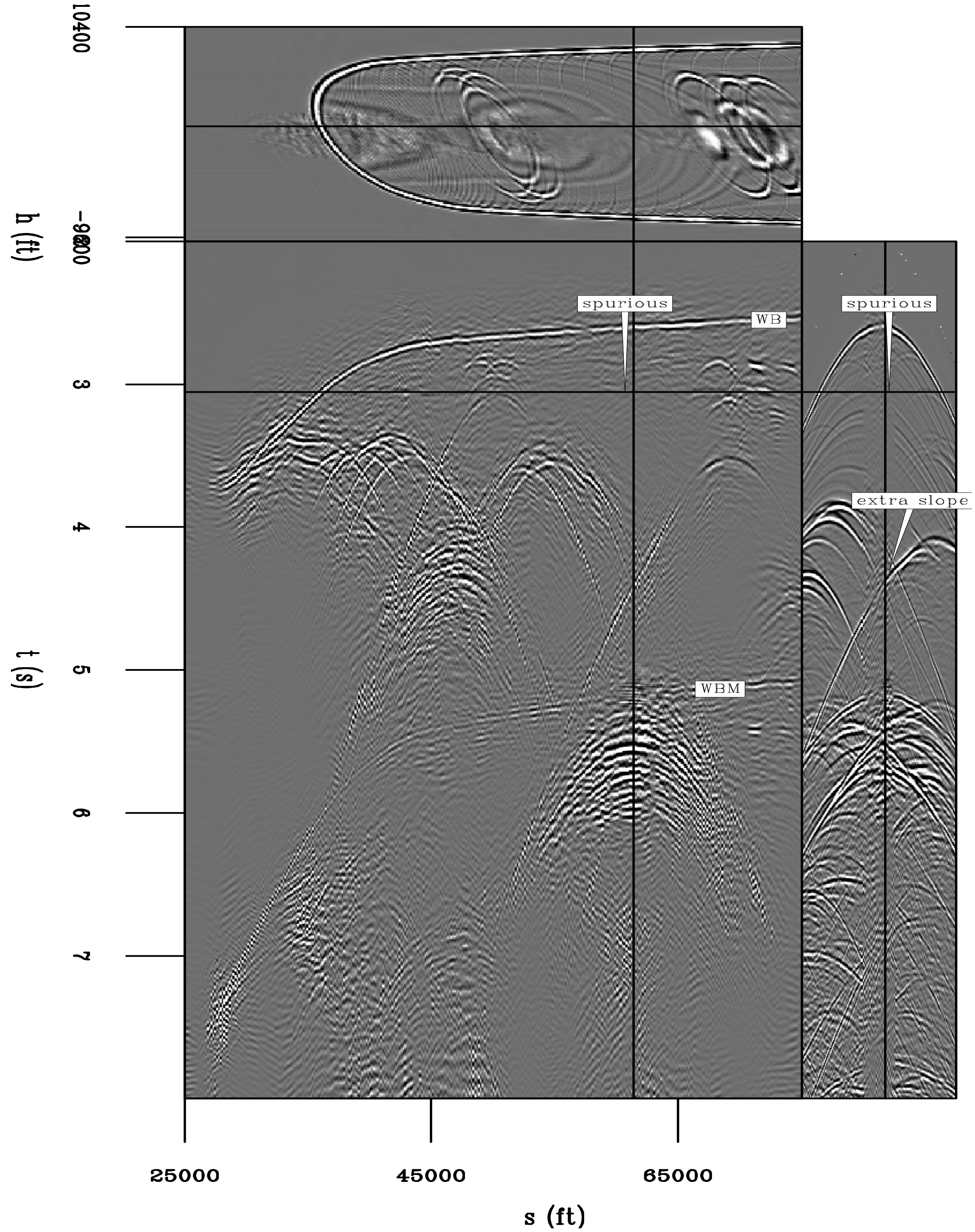

Figure 4.11: Interpolation with pseudo-primaries with 2D *f*-*h*-*s* PEFs. The result is more consistent from shot-to-shot, but still contains some cross-talk (around the water bottom) from the pseudo-primary data. The spurious event has been removed and the crossing slopes are believably interpolated. The image is scaled by $t^{0.8}$ for display purposes. **CR** Pseudo/. fxinterpann

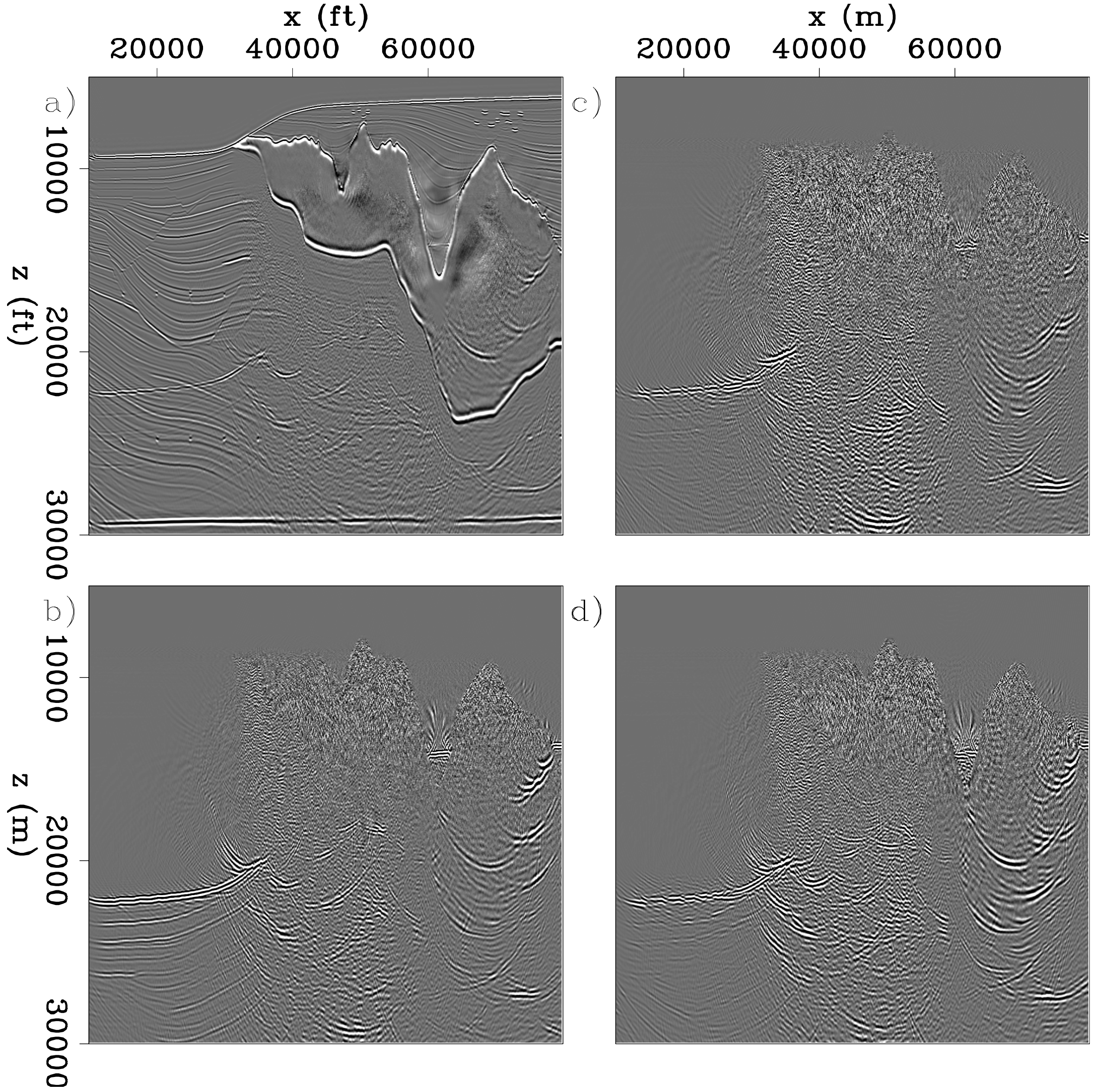

Figure 4.12: Migrated multiple prediction results for the *f*-*h*-*s* interpolation. (a): migrated fully-sampled data with multiples; (b): migrated multiple model generated with fully-sampled data; (c): migrated multiple model generated with no near offsets; (d) migrated multiple model generated with *f*-*h*-*s* interpolated near offsets. The interpolation restores much of the detail in the multiple model, especially under the edges of the salt body. **CR** Pseudo/. migmult

In addition to examining a migrated multiple model, we can also examine angle-domain common-image-gathers from both a migration of both the original data without the near offsets and a migration of the *f*-*h*-*s* interpolated data, as shown in Figure 4.13. Figure 4.13a is an angle gather for the water-bottom and a diffraction below the water-bottom, where the missing near offsets cause a dimming of the low angles, while Figure 4.13b is the angle gather from the interpolated data, where the low angles have comparable amplitude to the steeper angles.

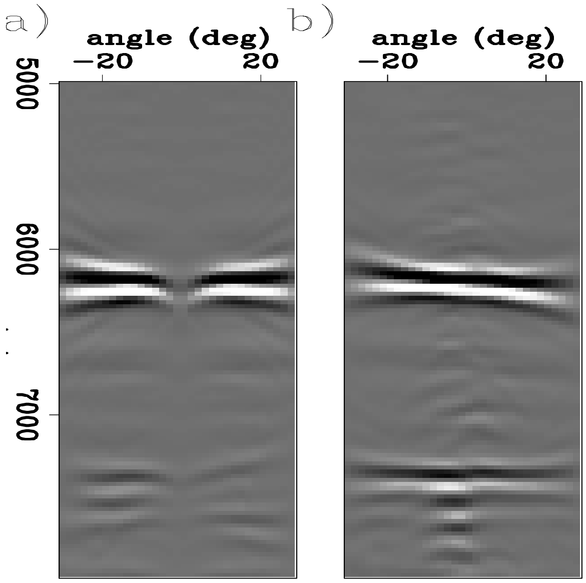

Figure 4.13: Angle-domain-common-image-gathers from migrations with and without near-offset *f*-*h*-*s* interpolation: (a) the original data migrated with missing near offsets; (b) the interpolated data after migration. The interpolation boosts the signal at flat angles for the water bottom and diffraction shown here. **CR** Pseudo/. migangle

This synthetic data example produced promising results for interpolation of a large gap using information only from the recorded data. The *f*-*h*-*s* approach slightly improved results and was still multiple times faster than the *t*-*h* or *t*-*h*-*s* interpolation methods. The synthetic data were noise-free, and only contained the desired primaries and multiples, with an idealized 2D geometry. Next, we examine how this method works on field data.

FIELD DATA EXAMPLE

Many have used the Mississippi Canyon multiples dataset from WesternGeco as a benchmark for testing multiple removal methods (Verschuur and Prein, 1999; Guitton and Cambois, 1999; Hadidi et al., 1999; Dragoset, 1999; Lamont et al., 1999; Lokshtanov, 1999; Matson et al., 1999; Berkhout, 1999; Weglein, 1999). We now use the strong free-surface multiples present in these data to generate pseudo-primaries. Figure 5.22 shows a source-offset cube of the recorded data; I created split-spread data from the off-end marine acquisition using source-receiver reciprocity. The near-offset gap in this case is six traces, much smaller than that in the Sigsbee2B example, but these field data contains much more than just primary and multiple reflections.

Pseudo-primary generation from field data

The pseudo-primary-contribution gather, shown in Figure 4.15, does contain the water-bottom, top-of-salt, bottom-of-salt reflections, and water-bottom multiples, caused by the correlation between the water-bottom reflection and the first-order multiple of the water-bottom, top-of-salt, bottom-of-salt, and second-order water-bottom multiple, respectively. This cube is a series of cross-correlations of single traces, with no amplitude scaling or deconvolution performed before cross-correlation. The signal-to-noise ratio in the data is lower than that in the synthetic example. The side panel shows the same receiver-pair correlation for multiple shots that are later summed to produce a single output trace.

Figure 4.16 shows the volume of pseudo-primaries generated by summing the cross-correlations from all of the sources in the recorded data. The pseudo-primary data are somewhat poorer than the recorded field data in Figure 5.22, with the long wavelet present in the original data strongly featured in the pseudo-primaries. Additionally, when compared to the pseudo-primaries in the synthetic case, the quality of the pseudo-primaries for this example is significantly worse. Reviewing three of the points I made about the pseudo-primaries in Sigsbee, we see that these problems are even more pronounced in this field data example. For example, while there were variations

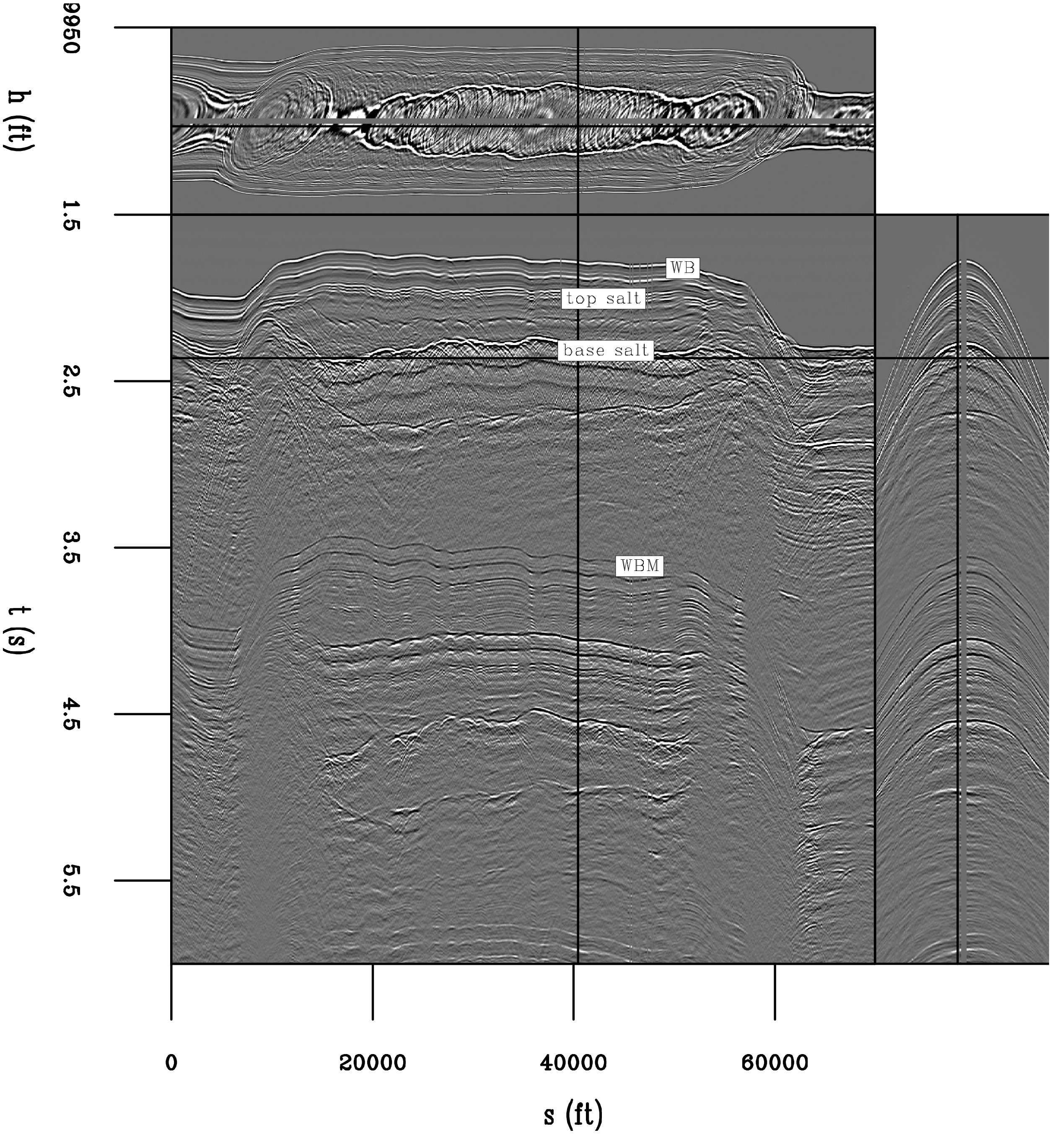

Figure 4.14: A Gulf of Mexico dataset. The front panel is a constant-offset section, and the side panel is a single shout, with the negative offsets predicted by reciprocity. **ER** Pseudo/. fieldinann

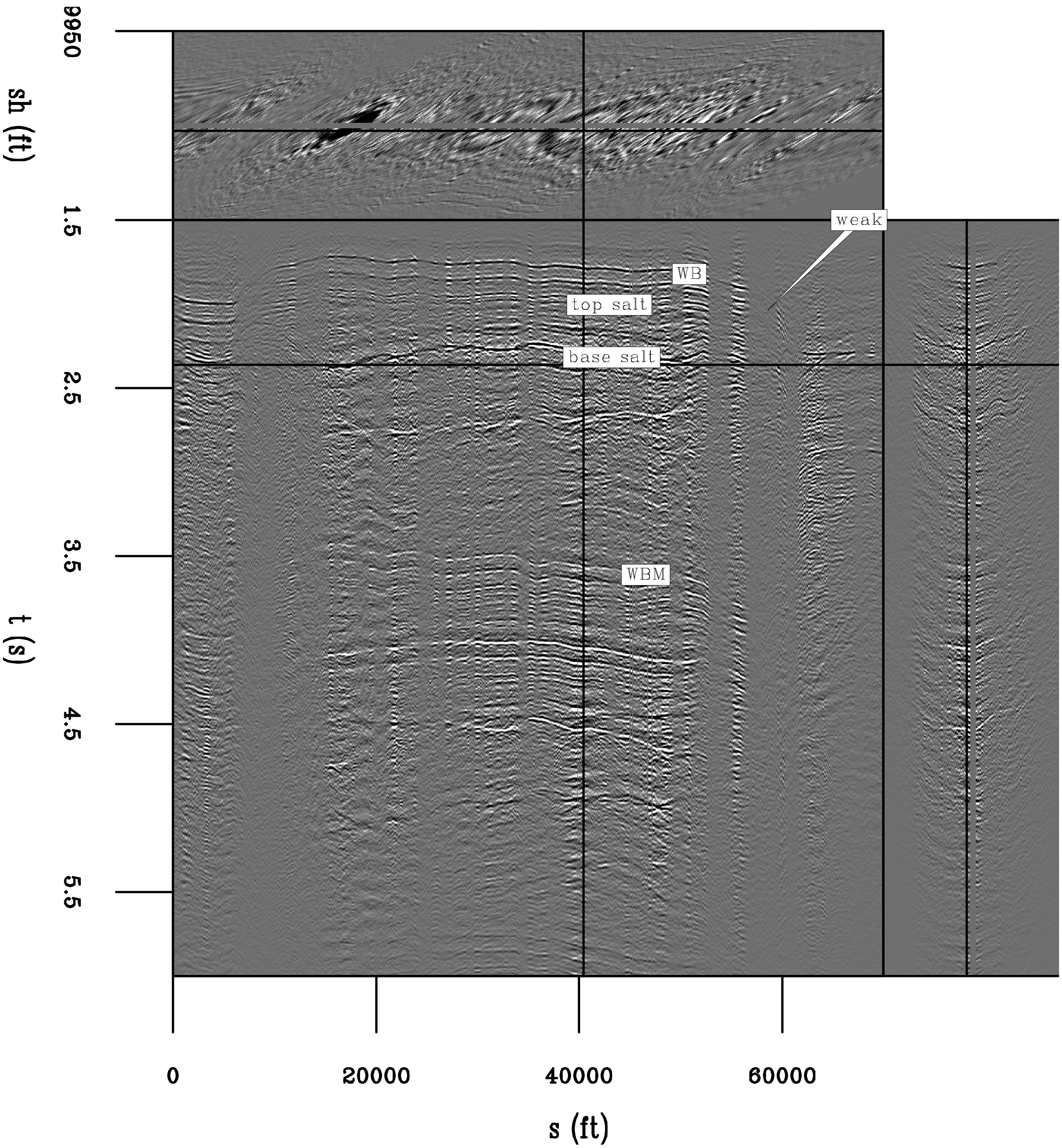

Figure 4.15: Pseudo-primary contribution gather. The front face is a constant-offset section, while the side face contains the shots that will be summed to produce the pseudo-primaries. A $t^{0.8}$ gain has been applied. **CR** Pseudo/. fieldcontribann

in the relative amplitude of the Sigsbee pseudo-primaries, the variations in the pseudo-primaries from the field data are much more pronounced; where the water-bottom reflection is dipping, the pseudo-primary reflection is nearly absent, and the pseudo-primaries of the salt body are much more pronounced than anywhere else. Second, the amplitude spectrum in the Sigsbee example was squared as a result of the cross-correlation involved in pseudo-primary generation. The much longer wavelet in the field data makes this more obvious, with the side-lobes of the water-bottom reflection appearing before the water-bottom reflection in the recorded data. This ringing is probably associated with a water bubble from the source. Third, the cross-talk in this field data is also present in this pseudo-primary result. These slight smile-shaped events have a slope the direction opposite that of the desired pseudo-primaries, and are more obvious at farther offsets, although the base of these smiles are at roughly zero-offset in Figure 4.16.

Interpolation of field data in f-h-s

As in the Sigsbee example, we first take the pseudo-primaries in Figure 4.16, perform a water-velocity normal moveout, and then break the result into 150 overlapping time windows of 32 samples each. These time windows are then Fourier transformed into 150×16 source-offset frequency slices. We then estimate a 3×4 nonstationary PEF that varies every sample along the source and offset axes for each of these slices, and use it to interpolate the corresponding slice of the recorded data with the near-offset gap. For each frequency of each time patch I use 80 iterations of a conjugate-direction solver both to estimate the PEF and to interpolate the missing data. The data are then transformed back to time, the windows are reassembled, and the NMO correction is reversed. This result is shown in Figure 4.17.

The interpolation result in Figure 4.17 is greatly improved over that in the pseudo-primary data. The most obvious differences between the pseudo-primaries and the interpolated result are that the ringing wavelet has been removed and the polarity of the images differ. The dipping portions of the water-bottom that were not present

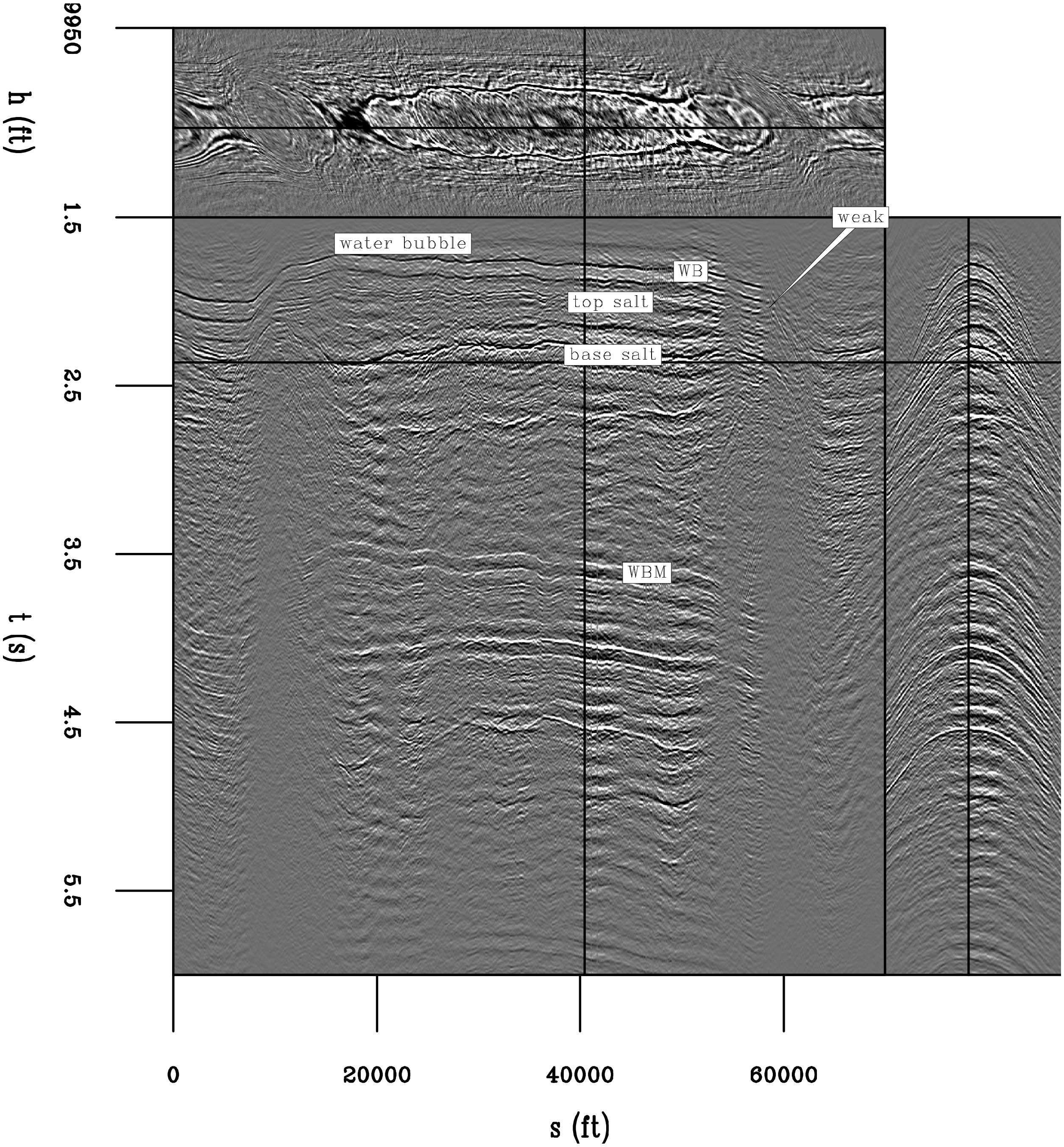

Figure 4.16: Pseudo-primaries of data. The front face is a constant-offset section, and the side face is a shot gather. A $t^{0.8}$ gain has been applied. The gross structure of the original data is present, but the squared wavelet strongly dominates the data. **CR** Pseudo/. fieldpseudoann

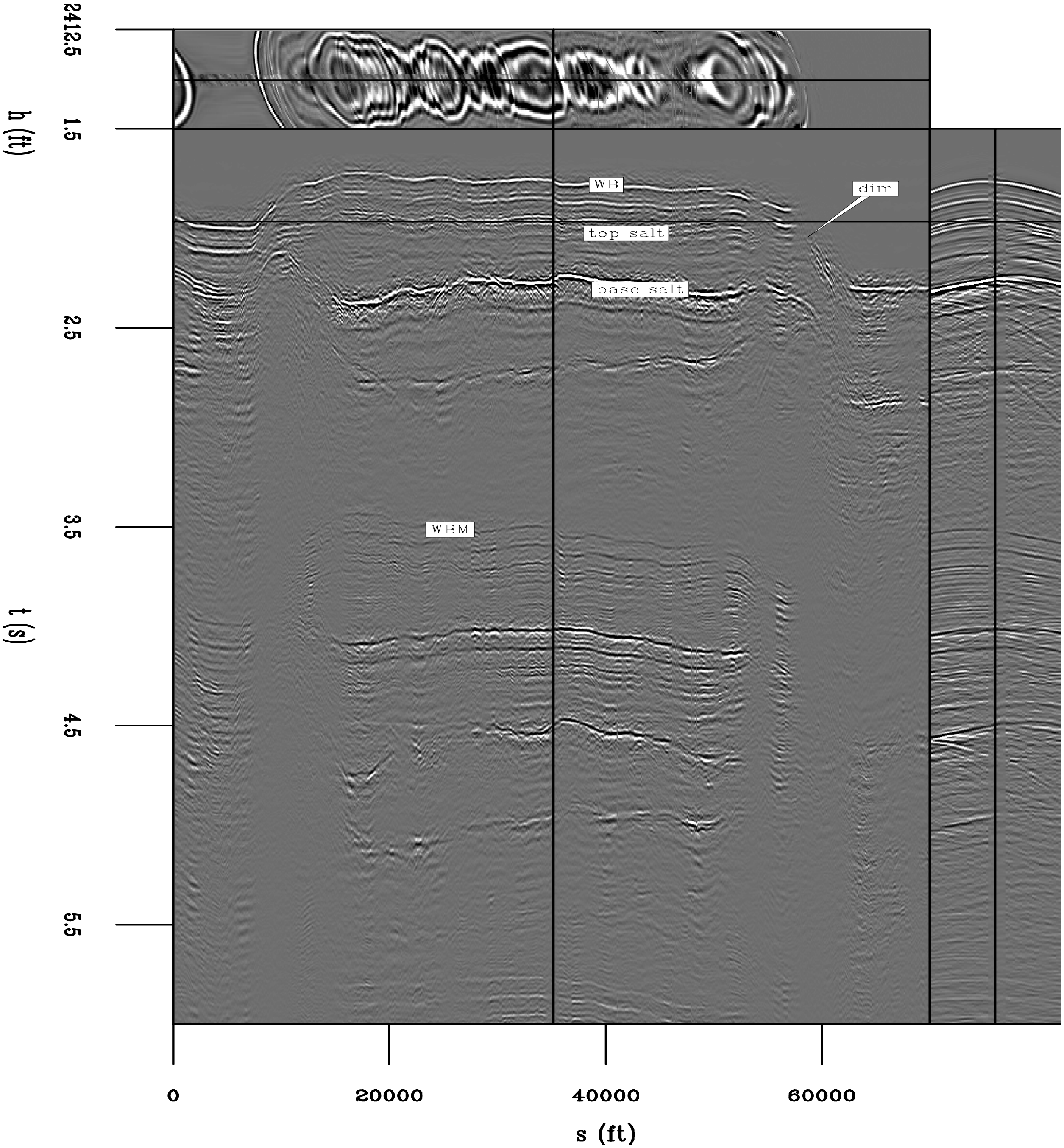

Figure 4.17: Interpolation of field data with a 2D f-h-s PEF using pseudo-primaries. The front face is a constant-offset section, and the side face is a shot. The near offsets appear much more reasonable than the pseudo-primaries in Figure 4.16. A $t^{0.8}$ gain has been applied, and the top and side panels are zoomed in relative to earlier figures in order to emphasize the interpolated values. **CR** Pseudo/. fieldfxann

in the pseudo-primaries were not interpolated. While this gap is much smaller than that in the Sigsbee example, the result is not as impressive. The water bottom, top of salt, bottom of salt, and the associated multiples are reasonably well interpolated, but the subtle stratigraphic reflectors outside of the salt body and the steeply dipping portions of the water bottom are not present in the interpolated result. For example, the diffractions below the bottom of salt reflection or at the edges of salt are not present in the interpolated result. The subtler stratigraphic reflectors outside of the salt body are well interpolated, even at later times.

PSEUDOPRIMARIES IN 3D

The pseudoprimaries generated so far have been either for a two-dimensional synthetic model or for two-dimensional field data example with strong multiple reflections. I now attempt this same 2D pseudoprimary generation on 3D prestack field data containing a region with significant crossline dip. The original recorded data in Figure 4.18a shows a single constant-offset section from 250m inline offset and the second receiver cable. This is the same data used in Chapter 1 in Figure 1.5 where a 2D SRME algorithm fails to produce accurate multiples in the submarine canyon. I generate pseudo-primaries using the same 2D subset of these data, where I take the second receiver cable and solve equation 4.2 to produce the pseudoprimaries for the same source locations and offset in Figure 4.18b.

The produced pseudoprimaries have a virtual source sampling that is equal to the receiver spacing, which is three times the original source spacing. The pseudoprimaries have a lower signal-to-noise ratio than the previous two examples, because the multiples are not as strong in these data as in the previous examples. The areas with a largely two-dimensional water bottom ($s_x : 17000 - 25000$m) have a pseudoprimary reflection that is kinematically in the correct place, and the top of the salt body reflection at $s_x = 19000$ is also visible. The other side of the canyon ($s_x : 5000 - 7000$) is also at roughly the correct arrival time. The water-bottom canyon, with a significant crossline slope, is not visible.

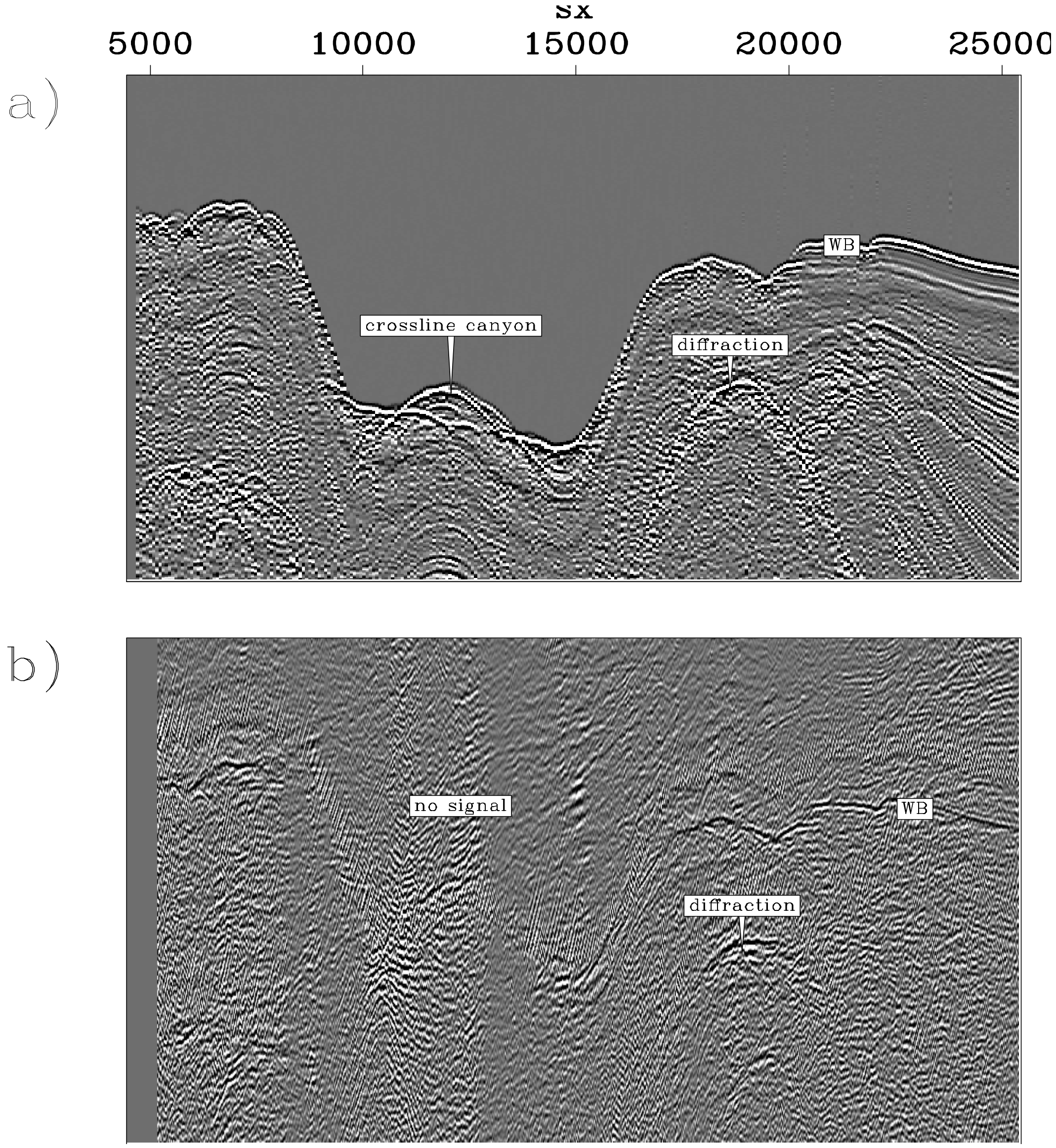

Figure 4.18: 2D pseudoprimary generation for 3D field marine data. a) original data; b) pseudoprimaries. The canyon present in the water-bottom has a large crossline component and the pseudoprimaries are incorrect. **CR** Pseudo/. 3dfieldpseudoann

To address this problem, producing a full 3D volume of pseudoprimaries from the recorded field data is not an appealing prospect. The source density is very low in the crossline direction, and the variable cable feathering along each sail line as well as from sail line to sail line would reduce the number of sources for each output virtual source location. Instead of dealing with these added complexities, I use a synthetic example where the geology is fully known and the acquisition is along ideal overlapping straight lines.

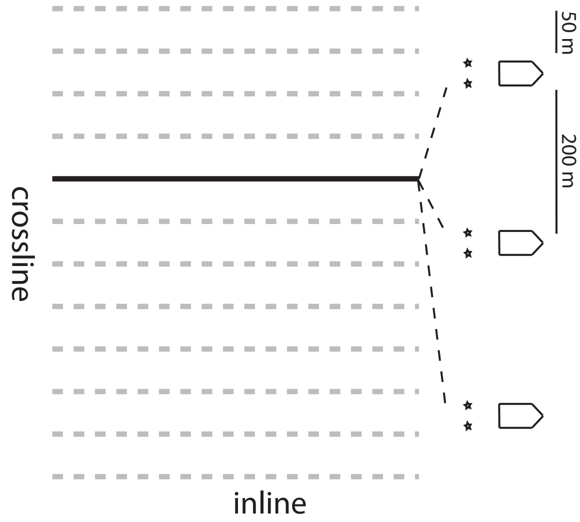

Figure 4.19: Crossline source distributions for a pseudoprimares generated from a single receiver cable at zero offset. **NR** Pseudo/. 3dpseudoann

Figure 4.19 is an illustration of the source distribution for a single receiver line from the 3D synthetic dataset used in both Chapters 1 and 5, courtesy of ExxonMobil. As shown in Figure 5.7a, the data are composed of a horizontal water-bottom reflection under which there is a prism filled with point diffractors. At slightly below the arrival of the diffracted multiples, three reflectors are present. The acquisition geometry, shown in Figure 4.19, is ideal, so that at each subsequent sail line the cables shift by a distance exactly equal to that between four receiver cables. As such, any one receiver cable location occurs for three sail lines given the recording aperture of 550m, which when multiplied by the two crossline source positions per sail line, gives six crossline source contributions for any one crossline receiver location under these ideal circumstances.

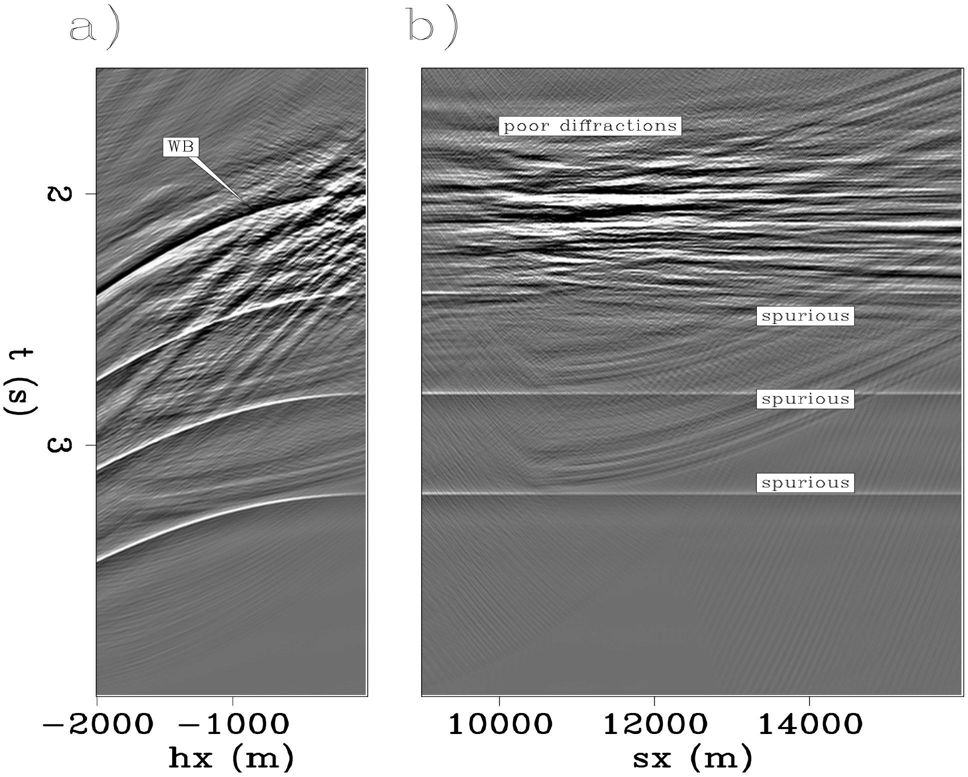

Figure 4.20: pseudoprimaries generated for a single receiver cable from six different crossline source positions. **CR** Pseudo/. 3dsynpseudoann

Figure 4.20 is the result of creating pseudoprimaries for zero crossline offset for a single receiver cable, where each active receiver location is cross-correlated with every other receiver location along the same cable for all sources in the three sail lines. These 640 sources for each of the three sail lines are added together to produce the pseudoprimary image in Figure 4.20, generated from roughly 4TB of individual crosscorrelations. While the water-bottom reflection initially appears to be in the correct place, the diffracted multiples are a blurred mess at near offsets and are absent at the far offsets where the limbs of the recorded diffractions are in the recorded data. There are also apparent reflectors below the water bottom that are created from the correlation of the water-bottom primary reflection with the much deeper series of three primary reflections at below the water-bottom multiple arrival time, illustrating the potential pitfalls of unwanted correlations.

In order to determine the variability of the predicted pseudoprimares with crossline source position, I have produced a crossline gather similar to the pseudoprimary contribution gathers I produced earlier in this chapter. While the full 3D uncollapsed cube of crosscorrelations would have 7 dimensions: time, inline and crossline source, inline and crossline first receiver, and inline and crossline second receiver, I only show the predicted multiples for a single first receiver (virtual source), second receiver, and inline source position, showing how the predicted multiples vary as the crossline location of the source varies. Figure 4.21 contains two of these gathers, where 4.21a is a zoom in on an area of the data with little diffracted multiples at the water bottom. The arrival time of the water bottom changes as a function of distance from the receiver cable to the source, with the stationary phase point at zero. Looking at a larger section of the time axis for a different location located in the cloud of diffractions below the water-bottom in Figure 4.21b, there is some similarity between the traces in adjacent flip-flop shots, but very little similarity between the three different sail lines, which in part explains the incoherence of the diffracted pseudoprimaries in Figure 4.20.

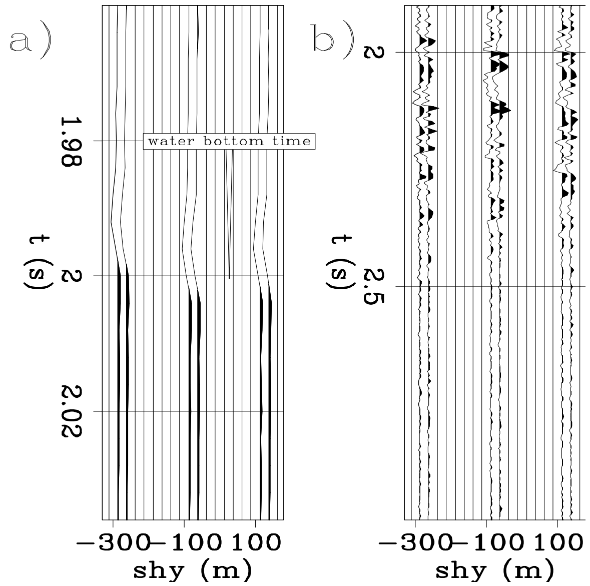

Figure 4.21: A crossline pseudoprimary contribution gather. The location of the water-bottom reflection changes as a function of the crossline distance from the inline location of the pseudoprimaries to the six sources. **CR** Pseudo/. 3dsynpseudogatherann

Pseudoprimaries and source sampling: Sigsbee analogue

I now use the Sigsbee2B dataset to further explore the limitations of source distribution, and attempt to separate the issues of source density and source aperture, using a subsampling of a 2D dataset as a proxy for 3D crossline sampling. Figure **??** is a comparison of different source densities on the zero-offset section of the pseudoprimaries, with the source sampling interval increased to twice (a), 10 times (b), and 50 times (c) that of the receiver sampling. The signal quality of the pseudoprimaries is still reasonable with 1/10 of the sources, but reducing it to 1/50 degrades the signal beyone recognition. This source density can also be viewed in shot gathers, as in Figure 4.23. Here the shot gathers look reasonable even at 1/20 the original source sampling, and the far offsets degrade only somewhat more than the near offsets.

Now instead of comparing the density of sources I compare their spatial extent in Figure 4.24. Figure 4.24a is a shot gather generated from all of the sources present in the full recording aperture of 53000 ft, while in subsequent figures I limit it to 15000 ft (4.24b), 10000 ft (4.24c), 5000 ft (4.24d), and 2000 ft (4.24e). The spatial extent of useful pseudoprimaires reduces with the source aperture, when the source aperture is larger than the range of offsets useful pseudoprimaries.

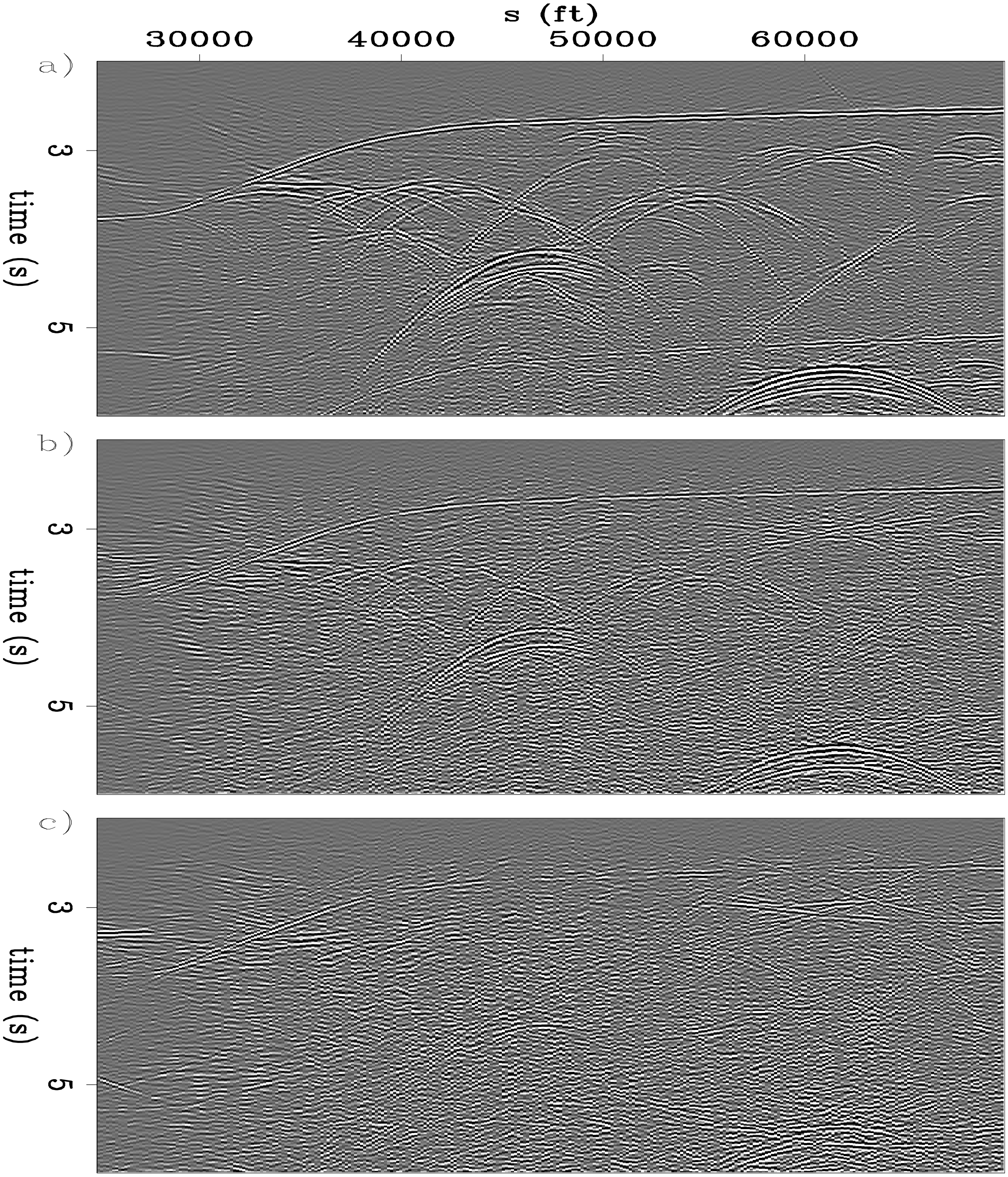

Figure 4.22: A comparison of source intervals on pseudoprimary quality, viewed as a zero-offset section: (a) 300 ft, (b) 750 ft, (c) 1500 ft, (d) 3000 ft, (e) 7500 ft. The signal quality degrades as the density of sources is reduced, but is still coherent with one source for every 20 offsets. **CR** Pseudo/. dsdeps

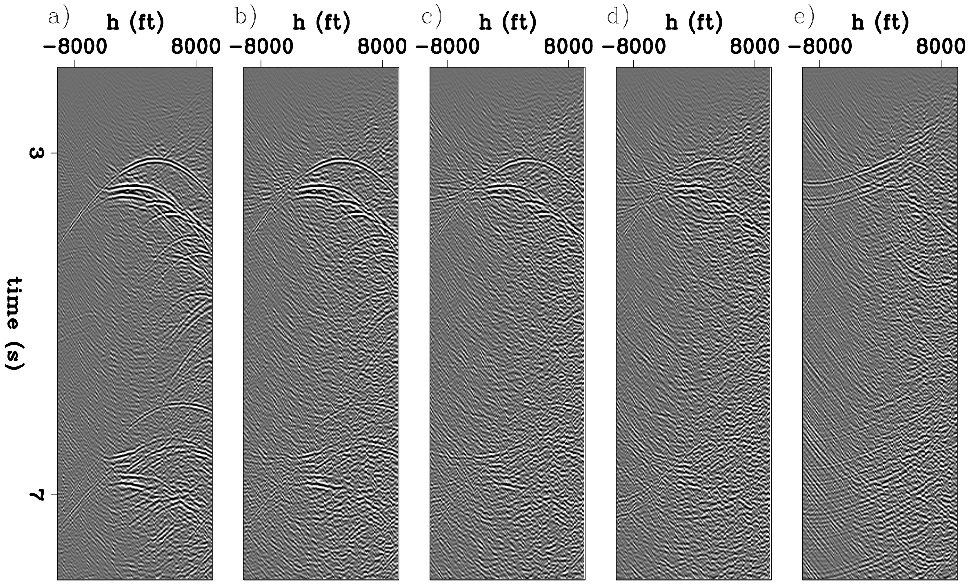

Figure 4.23: The same comparison as in Figure 4.22, but viewed as a virtual source gather at s=35000 ft and source sampling of: (a) 300 ft, (b) 750 ft, (c) 1500 ft, (d) 3000 ft, (e) 7500 ft. **CR** Pseudo/. dsdeph

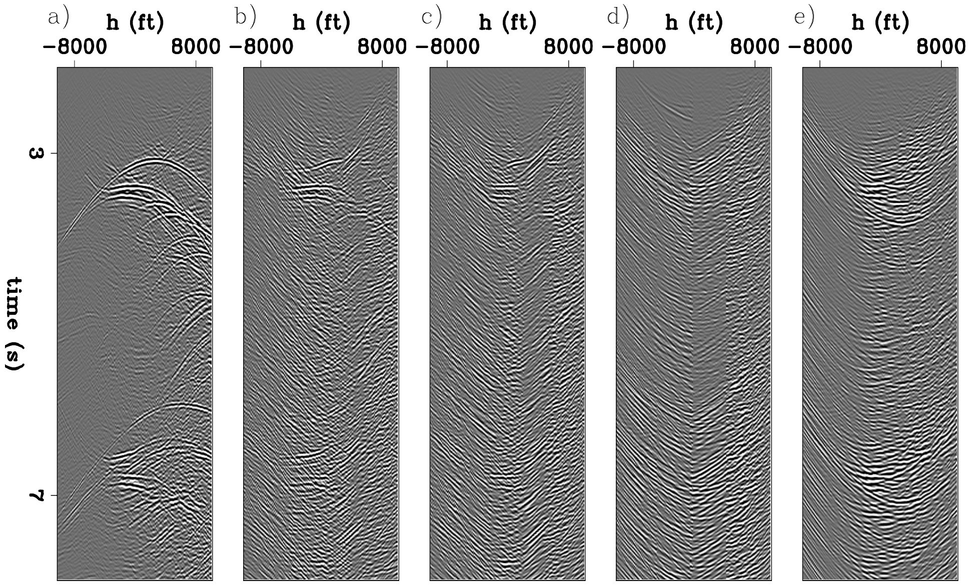

Figure 4.24: A comparison of aperture for pseudoprimaries at s=35000 ft with sources along (a) 15000, (b) 10000, (c) 5000, (d) 2000 ft surrounding the virtual source point. The data rapidly degrades as the aperture is limited. Pseudo/. nsdeph

CONCLUSIONS AND FUTURE WORK

The results in both the synthetic and field data examples show that this method works at capturing large inline features in the data. The field data example shows in particular that the PEF is insensitive to issues with the squared spectrum, amplitude scale and phase, which avoids the need for preprocessing involving amplitude scaling and deconvolution prior to cross-correlation.

The utility of this method in 3D is limited by the poor spatial distribution of sources, which has also been the case with passive interferometric methods. However, future acquisition methods such as wide-aziumth seismic may increase source density and more importantly aperture, which could in turn produce more useful pseudoprimaries in the crossline direction. Alternatively, this approach could be combined with moveout operators to produce data to add to the correlation.