Stanford Exploration Project, Report 79, November 16, 1997, pages 15–35

Lateral prediction for noise attenuation by t-x and f-x techniques

Ray Abma and Jon Claerbout[1]

keywords: *deconvolution, noise, three-dimensional*

ABSTRACT

Attenuating random noise with a prediction filter in the time-space domain generally produces results similar to those of predictions done in the frequency-space domain. However, in the presence of moderate- to high-amplitude noise, time-space, or t-x prediction passes less random noise than does frequency-space, or f-x prediction. The f-x prediction may also produce false events in the presence of strong parallel events where t-x prediction does not. These advantages of t-x prediction are the result of its ability to control the length of the prediction filter in time. An f-x prediction produces an effective t-x domain filter that is as long in time as the input data. Gulunay's f-x domain prediction, also referred to as FXDECON, tends to bias the predictions toward the traces nearest the output trace, allowing more noise to be passed, but this bias may be overcome by modifying the system of equations used to calculate the filter. The three-dimensional extension to the two-dimensional t-x and f-x prediction techniques allows improved noise attenuation, because more samples are used in the predictions, and the requirement that events be strictly linear is relaxed.

INTRODUCTION

Most seismic reflections are recognized by their lateral continuity, and this continuity is used to distinguish events of interest from the background of noise. In this discussion, we assume the events of interest are linear, that is, events are lines in a two-dimensional image and planes in a three-dimensional image. For nonlinear events, the images are subdivided, or windowed, into smaller sections where the events of interest are approximately linear.

This paper discusses two approaches to predicting linear events: a frequency-space, or f-x, prediction technique, and a time-space, or t-x, prediction technique. The f-x prediction technique was introduced by Canales (1984) and further developed

[1]**email:** not available

by Gulunay (1986). The standard industry name for this method is FXDECON, which stands for frequency-space domain predictive deconvolution. The reference to deconvolution is something of a misnomer, since deconvolution refers to the removal of predictable information, whereas in this paper, the data of interest are the predictable parts of the input. After a transformation of the data from the time-space domain into the frequency-space domain, the process of predicting a linear event can be divided into many smaller problems of predicting periodic events within a frequency. The t-x prediction process introduced in this paper is done with a single prediction filter calculated in the time-space domain using a conjugate-gradient method. The conjugate-gradient method and programs for filter calculations similar to the ones used here are discussed in Claerbout (1992) and Claerbout (1992).

While the two methods generally produce similar results, t-x prediction has several advantages over the older f-x prediction. These advantages allow t-x prediction to pass less random noise than the f-x prediction method. Most of the extra random noise is passed because the f-x prediction technique, while dividing the prediction problem into separate problems for each frequency, produces a filter as long as the data series in time when the collection of filters is transformed into a single filter in the t-x domain. Because the f-x prediction filter is very long in the time direction, it allows some random noise to be predicted. Since the length in time of the t-x prediction filter can be controlled, t-x prediction avoids passing this excess noise.

In three-dimensions, better results are expected for both techniques since more data goes into every prediction and since some of the linearity assumptions can be relaxed in three dimensions (Chase (1992); Gulunay *et al.*, (1993)). We find that extending these prediction techniques from two dimensions into three dimensions produces better results than two passes of the two-dimensional processes in the inline and crossline directions.

This paper examines these two prediction techniques and compares the results in both two and three dimensions. We also show some of the advantages that three-dimensional prediction has over the two-dimensional applications of these techniques.

TWO-DIMENSIONAL LATERAL PREDICTION

Many seismic images, especially those acquired on land, are contaminated with random noise that impedes interpretation and interferes with further processing and analysis. This random noise can be recognized by its dissimilarity from trace to trace. Signal, on the other hand, is recognized by its lateral continuity within the image. In most cases, this continuity results from the sedimentary character of the data being considered.

The methods we consider here predict only linear events within a seismic image. While the human eye recognizes the continuity of nonlinear events, the mathematical tools available to us work best on linear problems. Even though many continuous seismic events are not linear, windowing the image into smaller areas makes most of these events at least approximately linear.

The two lateral prediction methods discussed here are referred to as f-x prediction and t-x prediction. Prediction in the f-x domain, as presented by Canales (1984) and expanded by Gulunay (1986), predicts linear events in the frequency-space domain and has become a standard processing tool. As an improvement over the f-x prediction technique, we propose a t-x prediction technique that predicts linear events in the time-space domain.

Prediction of seismic signals in the t-x domain

The two-dimensional t-x technique predicts linear events with a two-dimensional time-space domain filter. This filter is calculated by minimizing the energy inside a window with a conjugate-gradient routine (Claerbout [1992a]). The filters used in the examples in this article have this form:

$$\begin{array}{ccccc} 0 & a_{-2,1} & a_{-2,2} & a_{-2,3} & a_{-2,4} \\ 0 & a_{-1,1} & a_{-1,2} & a_{-1,3} & a_{-1,4} \\ 1 & a_{0,1} & a_{0,2} & a_{0,3} & a_{0,4} \\ 0 & a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ 0 & a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \end{array} . \tag{1}$$

The vertical axis is the time axis, and the horizontal axis is the space axis. The output position is under the 1 coefficient on the left side of the filter. This filter has no free coefficients in the column corresponding to the output trace, which forces whatever predictions are made to be lateral.

As an example of a calculated filter, a flat event in a noiseless window produces a filter with all zeros except on the row containing the 1, where each of the a_{01}, a_{02}, a_{03}, and a_{04} coefficients in the filter above becomes -0.25. This filter minimizes the energy of the output by exactly predicting the flat event. Anything left after filtering is considered noise and removed from the input. Although making any one of the a_{01}, a_{02}, a_{03}, or a_{04} coefficients -1 produces a filter that exactly predicts an event, we set up the problem so that the coefficients tend to be equal for the best noise attenuation.

The choice of the filter size depends on the size of the window, the maximum dip in the data, the number of dips within the window, and the desired strength of the prediction effect. To keep the application of the calculated filter symmetrical, the filter is applied in both forward and reverse directions in space, with the results averaged.

The process of applying t-x prediction, as well as the f-x prediction discussed next, is applied to windows small enough for events of interest to appear linear. After filtering, these windows are merged to produce the output image.

Prediction of seismic signals in the f-x domain

An f-x prediction like FXDECON predicts linear events in the frequency-space domain. A linear event, given by the expression $f(x,t) = \delta(a + bx - t)$, where x is the lateral position and t is time, is Fourier transformed in time into $f(x,\omega) = e^{i\omega(a+bx)}$ or $f(x,\omega) = e^{i\omega a}(\cos(\omega bx) + i\sin(\omega bx))$. For a simple linear event, this function is periodic in x. This periodicity can be seen along any constant frequency line in the f-x domain display in Figure 1.

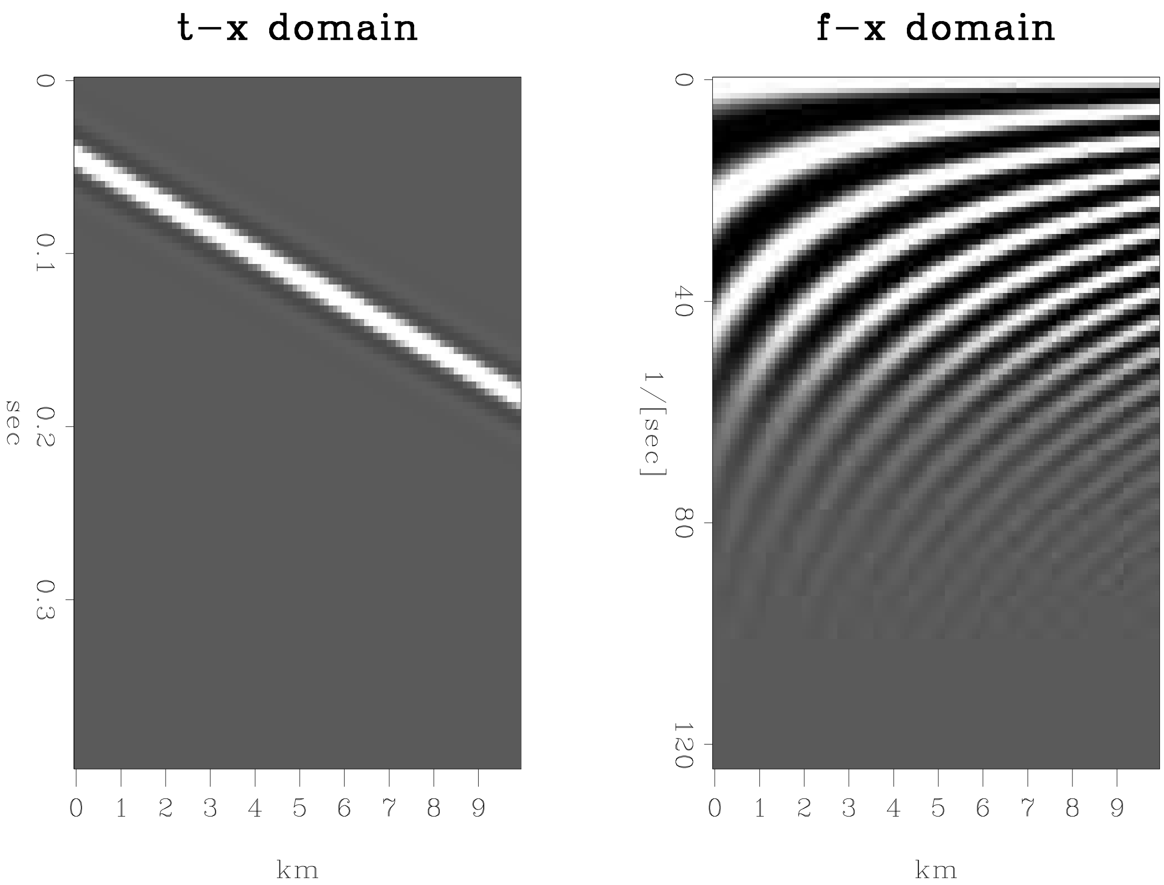

Figure 1: A single dip in the time domain and in the Fourier domain. The real part after Fourier transformation in time is shown. In the Fourier domain, the signal is periodic along any horizontal line. $\boxed{\text{ray1-sdip}}$ [ER]

To predict a linear event in $\underset{\sim}{\mathbf{x}}$, where $\underset{\sim}{\mathbf{x}}$ is a sampled version of $f(x,\omega)$ for a single frequency, Gulunay(1986) proposed calculating a least-squares prediction filter $\underset{\sim}{\mathbf{f}}$ from the system $\underset{\sim}{\mathbf{d}} = \underset{\approx}{\mathbf{X}}\underset{\sim}{\mathbf{f}}$, or, as expanded,

$$\begin{pmatrix} x_2 \\ x_3 \\ x_4 \\ x_5 \\ \cdot \\ \cdot \\ \cdot \\ x_{n+1} \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} x_1 & 0 & 0 & 0 \\ x_2 & x_1 & 0 & 0 \\ x_3 & x_2 & x_1 & 0 \\ x_4 & x_3 & x_2 & x_1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ x_n & x_{n-1} & x_{n-2} & x_{n-3} \\ 0 & x_n & x_{n-1} & x_{n-2} \\ 0 & 0 & x_n & x_{n-1} \\ 0 & 0 & 0 & x_n \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix}. \tag{2}$$

The input to the prediction problem is the data from a single frequency over the width of the window in space. This input is a set of complex numbers $(\; x_1 \quad x_2 \quad x_3 \quad \ldots \quad x_n \;)$. The desired output $\underset{\sim}{\mathbf{d}}$ is $(\; x_2 \quad x_3 \quad \cdots \quad x_n \quad x_{n+1} \;)$, a one-sample step-ahead prediction of the input. While $\underset{\sim}{\mathbf{d}}$ can be built with a longer shift of the input data, a shift of one sample is generally used. Notice that the desired output $\underset{\sim}{\mathbf{d}}$ starts with x_2 and ends with x_{n+1}. Gulunay (1986) set up the problem using this extra element from the input to guarantee that the filter would produce a result with the same amplitude as the input regardless of the length of the filter and of the data. The rows of the matrix $\underset{\approx}{\mathbf{X}}$ are shifted versions of the input that produce a convolution with the desired filter $\underset{\sim}{\mathbf{f}}$. This filter $\underset{\sim}{\mathbf{f}}$ can be calculated using the normal equations $\underset{\sim}{\mathbf{f}} = (\underset{\approx}{\mathbf{X}}^\dagger \underset{\approx}{\mathbf{X}})^{-1} \underset{\approx}{\mathbf{X}}^\dagger \underset{\sim}{\mathbf{d}}$, where $\dagger$ indicates the conjugate transpose, or adjoint. This is a standard solution to least-squares problems, except that the sample values are complex numbers, so the adjoint operation $\dagger$ cannot ignore taking the complex conjugate of the matrix elements on which it operates.

The f-x prediction is applied to small windows to ensure that events are locally linear, just as in the t-x prediction case, and the data within each window are then Fourier transformed. For the spatial series created at each frequency by the Fourier transform, a prediction filter is calculated as described in the preceding paragraph. Each calculated filter is first applied forward and then reversed in space, with the results averaged to maintain a symmetrical application, as in the t-x prediction case. The inverse Fourier transform is then applied to the filter result in each window, and the windows are merged to form the output image.

The calculation of the filter for each frequency is independent of the calculations of the filters for other frequencies. While the filters calculated at each frequency are a least-squares solution, this multitude of least-squares solutions does not necessarily produce a collective filtering action that is a global least-squares result. In the next section, we discuss the effect of this partitioning and the resulting differences between the actions of t-x prediction and f-x prediction.

The relationship of f-x prediction to t-x prediction

When applied to data with relatively mild random noise problems, most of the cases where t-x prediction was compared to f-x prediction showed practically identical results. In tests on synthetics with simple flat or dipping events in the absence of noise, both techniques passed events without distortion. These same simple cases with an added background of relatively low-amplitude random noise also produced comparable results. The results using real data sets with both techniques were also similar. Differences appeared only when the energy of the noise became a significant fraction of the energy of the signal.

These similarities and differences are explained by comparing the form of the t-x prediction filter and the form of the effective t-x domain filter created by f-x prediction. The individual filters calculated by the f-x prediction have the form

$$1 \quad a_1 \quad a_2 \quad a_3 \quad a_4 \ , \tag{3}$$

where a_1, a_2, a_3, and a_4 are complex numbers. Collecting these filters and presenting them in the frequency-space domain produces the following composite filter:

$$\begin{array}{ccccc}
\vdots & \vdots & \vdots & \vdots & \vdots \\
1 & a_{-4,1} & a_{-4,2} & a_{-4,3} & a_{-4,4} \\
1 & a_{-3,1} & a_{-3,2} & a_{-3,3} & a_{-3,4} \\
1 & a_{-2,1} & a_{-2,2} & a_{-2,3} & a_{-2,4} \\
1 & a_{-1,1} & a_{-1,2} & a_{-1,3} & a_{-1,4} \\
1 & a_{0,1} & a_{0,2} & a_{0,3} & a_{0,4} \\
1 & a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\
1 & a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\
1 & a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\
1 & a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \\
\vdots & \vdots & \vdots & \vdots & \vdots
\end{array} . \tag{4}$$

Transforming this filter from the frequency-space domain into the time-space domain gives the effective t-x domain filter

$$\begin{array}{ccccc}
\vdots & \vdots & \vdots & \vdots & \vdots \\
0 & b_{-4,1} & b_{-4,2} & b_{-4,3} & b_{-4,4} \\
0 & b_{-3,1} & b_{-3,2} & b_{-3,3} & b_{-3,4} \\
0 & b_{-2,1} & b_{-2,2} & b_{-2,3} & b_{-2,4} \\
0 & b_{-1,1} & b_{-1,2} & b_{-1,3} & b_{-1,4} \\
1 & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,4} \\
0 & b_{1,1} & b_{1,2} & b_{1,3} & b_{1,4} \\
0 & b_{2,1} & b_{2,2} & b_{2,3} & b_{2,4} \\
0 & b_{3,1} & b_{3,2} & b_{3,3} & b_{3,4} \\
0 & b_{4,1} & b_{4,2} & b_{4,3} & b_{4,4} \\
\vdots & \vdots & \vdots & \vdots & \vdots
\end{array} , \tag{5}$$

where the number of rows is the number of time samples in the window being considered. The Fourier transform has converted the first column of 1s into a single 1 at the output position, creating a time-space filter with the same form as the t-x prediction filter shown in (1), but with the number of rows greatly increased.

For data with random noise that has low amplitude compared to the signal, the coefficients of the filter producing most of the prediction are expected to lie close to the center of the filter in time, since events far from the output point in time are unlikely to affect data at the output position. For data with high amplitude random noise, random correlations of events widely separated in time create undesirable lineups with significant amplitudes in the output of f-x prediction. Since the f-x prediction produces a very long effective t-x filter in time, its prediction is contaminated with random correlations, where the t-x prediction, with its smaller filter, is not.

The time-length problem of the f-x prediction filter can be compared to the problem of calculating a one-dimensional prediction filter from a short time series. If a long one-dimensional prediction filter is calculated and applied to a short time series, practically everything in the time series will be predicted. Having a two-dimensional filter with a significant lateral width compensates for some of this overprediction, but a short t-x prediction filter is a more complete solution to the problem.

Another effect of the long time-length filter in f-x prediction is the generation of false events. If two strong parallel events are embedded in a background of random noise, f-x prediction generates weak, but significant events parallel to the strong events. While both t-x and f-x techniques tend to line up noise with strong events, f-x prediction actually generates false events with its long filter in time. These spurious events occur because parallel events in a random noise background produce an f-x filter where the prediction of one strong event is influenced by the presence of other parallel events. The t-x prediction is much less likely to be affected by the influence of widely separated events because of its short filter in time.

Comparison of two-dimensional f-x and t-x predictions

We compared the effect of varying the time-extents for prediction filters in the presence of high-amplitude noise. In a simple noise-only case, when we used the same number of coefficients in the lateral direction for both processes, f-x prediction passed about twice the noise energy as a short time-length t-x prediction. As the time-length of the t-x prediction filter increased, the t-x prediction passed more random noise. When filters with a time-length comparable to the data time-length were used, we found almost no difference between the filters or between the results of t-x prediction and f-x prediction. Thus, because of its ability to limit the filter length in time, t-x prediction has a definite advantage over f-x prediction in removing random noise.

This difference in passing random noise is shown in Figure 2, where the results of the t-x prediction and f-x prediction are applied to a two-dimensional stacked section. We used a time length of five coefficients for the t-x prediction result and a spatial length of five coefficients for both the t-x and f-x prediction results shown in Figure 2. The window sizes in both cases were 30 traces by 300 samples, or 0.6 seconds in time. This data set contains a moderate amount of noise and was one of the few nonsynthetic data sets showing recognizable differences between t-x prediction and f-x prediction. While the results are similar, the f-x prediction passes somewhat more noise than t-x prediction.

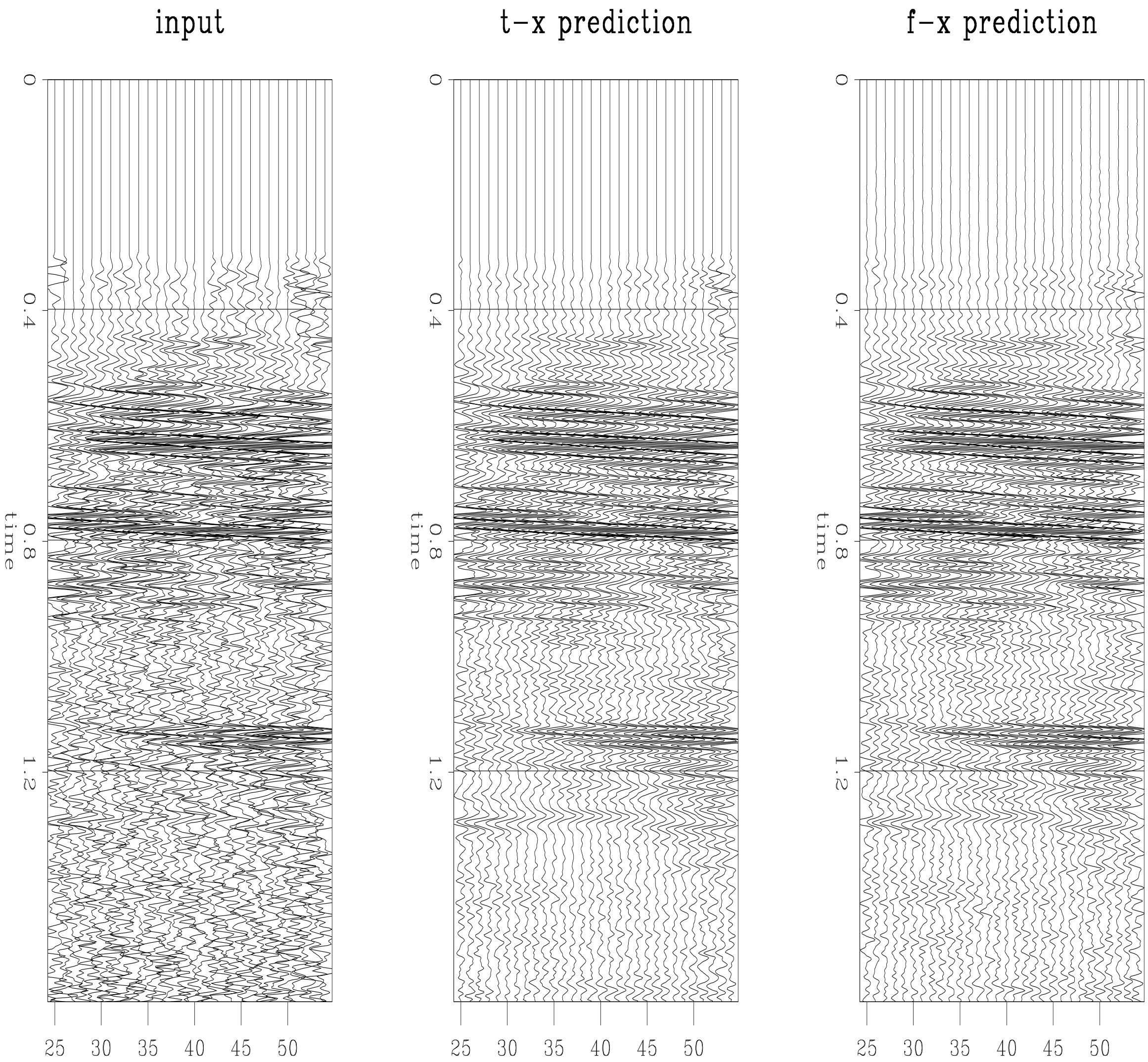

Figure 2: Two-dimensional t-x prediction and f-x prediction on a stacked line. The top section is the input; the middle section the t-x prediction result; the bottom section the f-x prediction result. The results of the two techniques are similar, but somewhat less noise is passed with t-x prediction than with f-x prediction. ray1-DDw [ER]

The biasing of f-x prediction toward the output point

Though less of a problem than the long time-length filter, the tendency for the lateral prediction coefficients to be concentrated near the output trace position with Gulunay's f-x prediction causes more noise to be passed. The source of this problem can be seen in the system describing the f-x prediction filtering, $\underset{\sim}{\mathbf{d}} = \underset{\approx}{\mathbf{X}}\underset{\sim}{\mathbf{f}}$, or

$$\begin{pmatrix} x_2 \\ x_3 \\ x_4 \\ x_5 \\ \cdot \\ \cdot \\ \cdot \\ x_{n+1} \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} x_1 & 0 & 0 & 0 \\ x_2 & x_1 & 0 & 0 \\ x_3 & x_2 & x_1 & 0 \\ x_4 & x_3 & x_2 & x_1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ x_n & x_{n-1} & x_{n-2} & x_{n-3} \\ 0 & x_n & x_{n-1} & x_{n-2} \\ 0 & 0 & x_n & x_{n-1} \\ 0 & 0 & 0 & x_n \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix}, \qquad (6)$$

where the vector $\underset{\sim}{\mathbf{d}}$ is the desired output, $\underset{\approx}{\mathbf{X}}$ is the input data shifted with respect to the filter, and $\underset{\sim}{\mathbf{f}}$ is the filter to be calculated. The zeros in the last few coefficients of $\underset{\sim}{\mathbf{d}}$ tend to reduce the f_2, f_3, and f_4 coefficients of the filter. The zeros at the top of the $\underset{\approx}{\mathbf{X}}$ matrix have a similar effect. The result of this tendency to weight the f-x prediction filter coefficients toward the output point of the filter may be appealing in terms of producing an output trace made up of the nearest traces, but the noise in the nearest traces is also passed with less attenuation. This increased weighting of the nearest traces produces a filter that is less effective in rejecting noise.

The increased weighting of the nearest trace can be remedied by setting up the problem differently. Removing the top and bottom rows of Equation (6) produces

$$\begin{pmatrix} x_5 \\ x_6 \\ x_7 \\ \cdot \\ \cdot \\ \cdot \\ x_{n-1} \\ x_n \\ x_{n+1} \end{pmatrix} = \begin{pmatrix} x_4 & x_3 & x_2 & x_1 \\ x_5 & x_4 & x_3 & x_2 \\ x_6 & x_5 & x_4 & x_3 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ x_{n-2} & x_{n-3} & x_{n-4} & x_{n-5} \\ x_{n-1} & x_{n-2} & x_{n-3} & x_{n-4} \\ x_n & x_{n-1} & x_{n-2} & x_{n-3} \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix}, \qquad (7)$$

so the filter is calculated only where nonzero data are available. Unfortunately, the normal equations $\underset{\sim}{\mathbf{f}} = (\underset{\approx}{\mathbf{X}}^\dagger \underset{\approx}{\mathbf{X}})^{-1} \underset{\approx}{\mathbf{X}}^\dagger \underset{\sim}{\mathbf{d}}$ cannot be used, since $(\underset{\approx}{\mathbf{X}}^\dagger \underset{\approx}{\mathbf{X}})$ does not necessarily have an inverse. For a flat event, all the coefficients in the $\underset{\approx}{\mathbf{X}}$ matrix have equal values, and all the elements of $\underset{\approx}{\mathbf{X}}^\dagger \underset{\approx}{\mathbf{X}}$ also have equal values, making $\underset{\approx}{\mathbf{X}}^\dagger \underset{\approx}{\mathbf{X}}$ singular.

Expanding Equation (7) to add a damping condition to equally weight the filter coefficients gives

$$
\begin{pmatrix} x_5 \\ x_6 \\ x_7 \\ \cdot \\ \cdot \\ \cdot \\ x_{n-1} \\ x_n \\ x_{n+1}a \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} x_4 & x_3 & x_2 & x_1 \\ x_5 & x_4 & x_3 & x_2 \\ x_6 & x_5 & x_4 & x_3 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ x_{n-2} & x_{n-3} & x_{n-4} & x_{n-5} \\ x_{n-1} & x_{n-2} & x_{n-3} & x_{n-4} \\ x_n & x_{n-1} & x_{n-2} & x_{n-3} \\ \epsilon & 0 & 0 & 0 \\ 0 & \epsilon & 0 & 0 \\ 0 & 0 & \epsilon & 0 \\ 0 & 0 & 0 & \epsilon \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix}, \tag{8}
$$

where ϵ is a small number. Solving this set of equations avoids the tendency for the largest coefficients to cluster near the output position that we have seen in Gulunay's method.

Although these modifications should improve the noise attenuation properties of the complex prediction filter, the biasing effect on the noise attenuation is very small compared to the improvement from using the shorter time-length filter of a typical t-x prediction. A feature of Gulunay's method is, that for noiseless data, prediction is unneeded and concentrating the strongest predictions coefficients near the output has no effect on the results. As the strength of the noise increases, the distribution of the amplitudes of the filter coefficients becomes more even, since adding noise to the input provides the same effect as adding a factor to the diagonal of $\underset{\approx}{\mathbf{X}}^{\dagger}\underset{\approx}{\mathbf{X}}$.

It might be thought that this noise-dependent coefficient weighting gives f-x prediction an advantage over t-x prediction. However, if an irregularity such as a fault exists in the input data, both t-x and f-x predictions will generate identical filters in the noiseless case, since the difference between the input and the predicted data will be smallest for a filter with strong filter coefficients adjacent to the output trace and producing the least smearing of the irregularity. For perfectly regular data with noise added, the f-x prediction filter coefficient distribution will approach that of the t-x prediction filter as the strength of the noise is increased. Thus f-x prediction always has an effectiveness that is less than or equal to the t-x predictions effectiveness.

Computer time requirements

The number of filter coefficients also controls the computer time required by these techniques. Since the applications being considered here are post-stack and generally fast, at least for two-dimensional cases, the computer time required is generally insignificant. It is interesting to note that while the effective t-x prediction time-domain filter has more coefficients than the individual filters used in f-x prediction, because f-x prediction produces a different filter for each frequency, the total number of filter coefficients is much larger than the number for t-x prediction. Nevertheless, the computer time needed to apply these two processes in the two-dimensional case is comparable, since each filter calculated by f-x prediction is smaller than the single filter calculated by t-x prediction. Since the time to calculate a filter is proportional to n^3, where n is the number of filter coefficients, the calculation times used by the two approaches become nearly equal. For the three-dimensional case, the application of t-x prediction tends to be more costly than f-x prediction, but not tremendously so, since the filter sizes used by t-x prediction can be smaller than those for a comparable f-x prediction. However, as the size of the t-x prediction filter increases, the processing time increases rapidly. On the other hand, should cost be an issue, the number of iterations in the conjugate-gradient routine used by t-x prediction can be significantly reduced from its theoretical limit with good results.

THREE-DIMENSIONAL LATERAL PREDICTION

Three-dimensional lateral prediction has two important advantages over two-dimensional prediction. The first is that, in a three-dimensional volume, any output sample is close to more samples in the input than in the two-dimensional case. Having more nearby samples allows better random noise attenuation. Second, the assumption that events are linear within a two-dimensional window or three-dimensional subvolume can be relaxed. As pointed out by Chase (1992) and Gulunay *et.al.* (1993), events that are nonlinear in one direction but linear in another are predicted exactly with a three-dimensional filter.

Prediction over a three-dimensional volume can be done with two passes of a two-dimensional process or with a true three-dimensional technique. Since two passes of either two-dimensional technique does not allow processors to take full advantage of the opportunities three-dimensional data presents, we have extended the two-dimensional techniques to true three-dimensional techniques using three-dimensional filters.

The three-dimensional extension of t-x prediction

Extending the t-x prediction to three dimensions is a simple matter of defining the three-dimensional convolution and its adjoint, that is, its conjugate transpose. Instead of dividing the input image into two-dimensional windows, the input volume is broken into three-dimensional subvolumes. Each of these subvolumes is used in a separate least-squares problem for calculating a three-dimensional filter, which is a straightforward extension of the calculation of a two-dimensional t-x filter. Figure 3 shows the configuration of the three-dimensional filter we used. As in the two-dimensional case, the three-dimensional prediction is applied in both forward and reverse spatial directions to maintain the symmetry of the application.

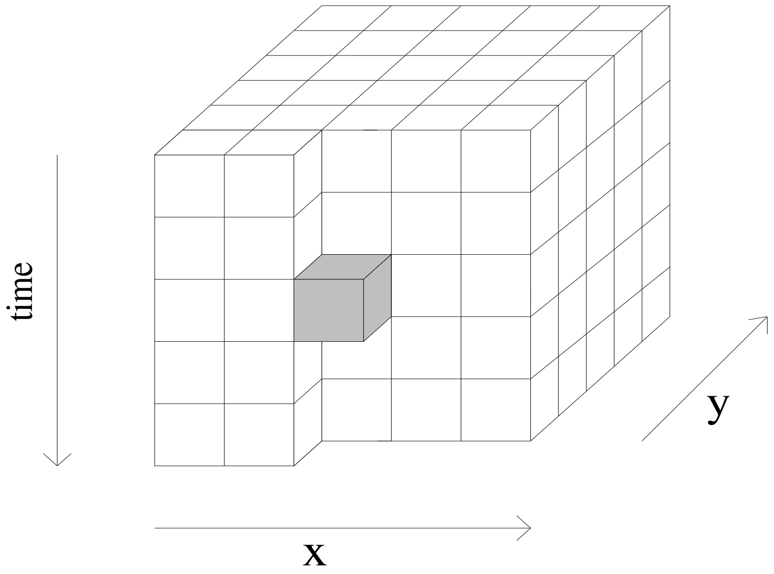

Figure 3: A three-dimensional lateral prediction filter. The darkened cube is the output position. ray1-DDDfilter [NR]

The three-dimensional extension of f-x prediction

The extension of f-x prediction to three-dimensions is more difficult than that of t-x prediction. For each frequency, instead of a prediction along a vector, the prediction of a set of complex numbers within a plane is required. For the examples of three-dimensional f-x prediction shown here, we used a complex-valued two-dimensional filter calculated with a conjugate-gradient routine for each frequency. While other techniques for computing this filter exist, they should produce similar results. The advantage of our approach is that the huge matrix $\underset{\approx}{\mathbf{X}}$ used to describe the three-dimensional convolution of the filter with the data does not need to be stored, and the inverse of $\underset{\approx}{\mathbf{X}}^{\dagger}\underset{\approx}{\mathbf{X}}$ does not need to be computed, which simplifies the problem significantly(Claerbout, 1992).

The shape of the two-dimensional filter used to predict numbers in a two-dimensional frequency slice has the form

$$\begin{array}{ccccc} c_{-2,0} & c_{-2,1} & c_{-2,2} & c_{-2,3} & c_{-2,4} \\ c_{-1,0} & c_{-1,1} & c_{-1,2} & c_{-1,3} & c_{-1,4} \\ 1 & c_{0,1} & c_{0,2} & c_{0,3} & c_{0,4} \\ 0 & c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} \\ 0 & c_{2,1} & c_{2,2} & c_{2,3} & c_{2,4} \end{array} , \tag{9}$$

where all the coefficients are complex-valued. The justification for this shape is more fully discussed on page 198 of(Claerbout, 1992). This form may be compared to a horizontal slice through the center of the filter shown in Figure 3, and if the collection of the filters for all frequencies is Fourier transformed in time, a filter similar to the one shown in Figure 3, but extended in time, is formed.

Examples of three-dimensional lateral prediction

An example of three-dimensional prediction's ability to predict nonlinear events is shown in Figure 4, where the input consists of several dipping layers cut by a fault in the crossline direction, so that events are nonlinear in the inline direction. The one-pass three-dimensional prediction did not smear the fault, because the calculated three-dimensional filter created a prediction in the crossline direction that preserved the discontinuity in the inline direction. With the two-pass prediction, the inline pass smeared the reflections across the fault. For the noiseless case of Figure 4, the f-x prediction is not shown because it gave the same results as the t-x prediction.

The results of applying f-x prediction and t-x prediction to a three-dimensional land survey provided by ARCO are shown in Figures 5 to 9 to demonstrate the differences between the two processes. This data set is interesting because it has a significant noise level with fairly flat, predictable events.

For both two-pass applications, the filter size was five elements in the spatial direction. For both three-dimensional one-pass applications, we employed a filter with five elements in both spatial directions. The t-x prediction used a five-element filter length in the time direction for both one- and two-pass applications. The window sizes were 60 traces in the inline direction, 60 traces in the crossline direction, and 200 samples, or 0.4 seconds in time.

Both the two-pass and the one-pass t-x prediction results in Figures 8 and 9 show less noise than the corresponding f-x results; otherwise the results are similar. While the one-pass t-x prediction and f-x prediction results are much the same, the t-x prediction output shows somewhat less noise.

The advantage of using three-dimensional lateral prediction is especially clear in the results shown in Figures 7 and 9. Both the one-pass results show significantly less smearing of the structure. On the top faces of the cubes in Figures 7 and 9, the one-pass results appear clean and reasonable, whereas the two-pass results show smearing along the inline and crossline directions. An example of the smearing of the detail can be seen in point A of the figures, where a small doughnut-shaped feature is badly smeared in both the two-pass results. The front face of the cubes in Figures 6 to 9 are significantly different for the one- and two-pass results; with the one-pass results showing much more detail. The features at point B in the figures once again demonstrate the loss of detail. Although the differences between the three-dimensional t-x and f-x results in Figures 7 and 9 are less than those between the two-dimensional t-x and f-x results in Figure 2, the results of the three-dimensional t-x prediction appear slightly cleaner than those of the three-dimensional f-x prediction.

CONCLUSIONS

Although t-x prediction gives results similar to those of f-x prediction in low noise cases, t-x prediction provides better random noise reduction than the older f-x prediction technique in the presence of moderate- to high-amplitude noise. The t-x prediction also avoids the generation of false events in the presence of strong parallel events when using f-x prediction. These advantages of t-x prediction are the result of its ability to control the length of the prediction filters in time. Because t-x prediction has a shorter effective filter length in time than f-x prediction, t-x prediction passes significantly less random noise than f-x prediction. While Gulunay's f-x prediction biases the prediction toward the traces nearest to the output point, allowing more noise to be passed, this bias appears unimportant when compared to the problem with the length of the effective filter in time.

Three-dimensional prediction allows improved noise attenuation because more samples are used to make predictions. Three-dimensional prediction also relaxes the requirement that events be linear. Comparisons of one- and two-pass predictions on a land data set show that the one-pass results show significantly less noise than the two-pass results. In both one- and two-pass predictions, the f-x prediction passes more undesirable random noise than the t-x prediction.

ACKNOWLEDGMENT

We thank the Atlantic Richfield Company for generously providing the three-dimensional data set.

REFERENCES

Canales, L. L., 1984, Random noise reduction: 54th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Session: S10.1.

Chase, M. K., 1992, Random noise reduction by 3-D spatial prediction filtering: 62nd Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Session: SP3.4.

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

Claerbout, J. F., 1992b, Information from smiles: mono-plane-annihilator weighted regression: Stanford Exploration Project–**73**, 409–420.

Gulunay, N., Sudhakar, V., Gerrard, C., and Monk, D., 1993, Prediction filtering for 3-D post-stack data: Preprint of an article submitted for the 63rd Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts.

Gulunay, N., 1986, FXDECON and complex Wiener prediction filter: 56th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Session: POS2.10.

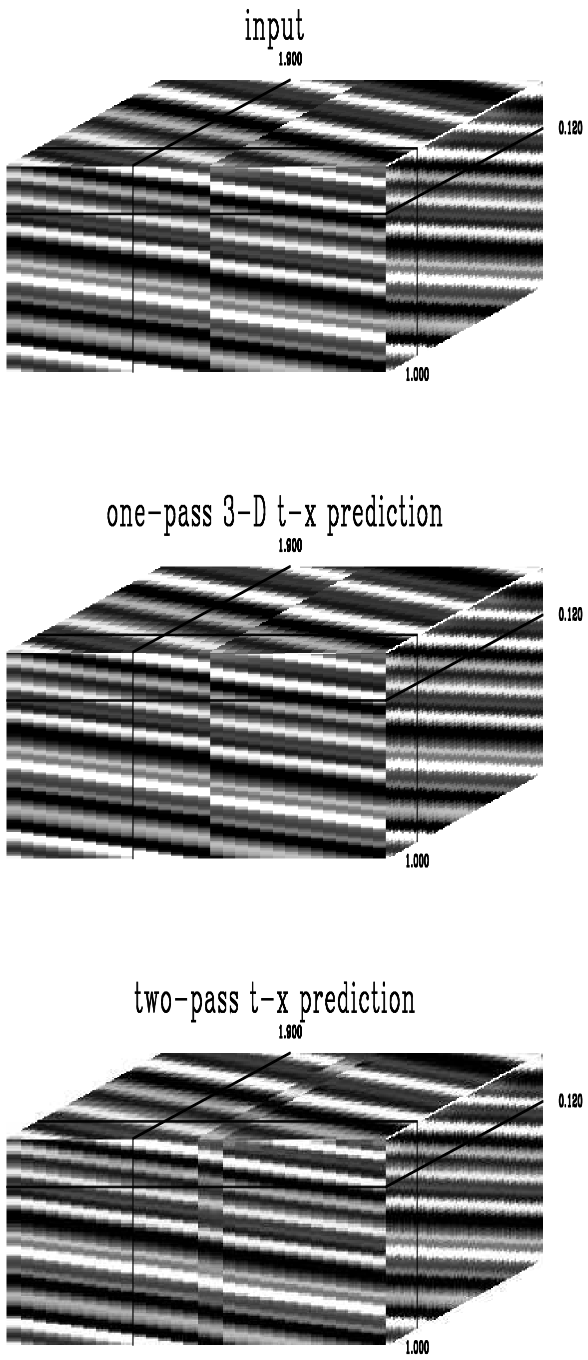

Figure 4: A comparison of smearing on with one- and two-pass t-x prediction. The top cube is the input, the middle cube is the result of a one-pass three-dimensional t-x prediction, and the bottom cube is the result of two passes of one-dimensional t-x prediction in the inline and crossline directions. The one-pass result shows no smearing of the fault; the two-pass result shows a smeared fault image. ray1-yplanesld [ER]

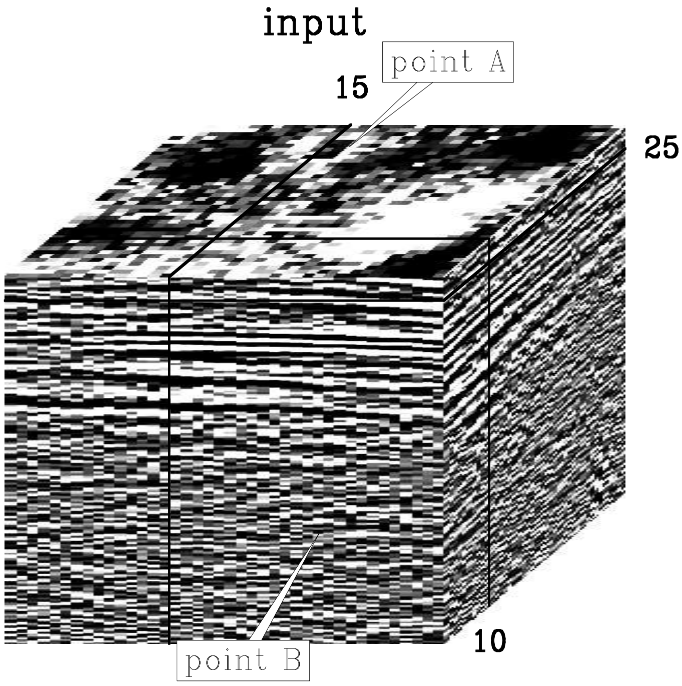

Figure 5: The input to the t-x and f-x predictions. A significant amount of noise is seen here. ray1-Arcoorig [CR]

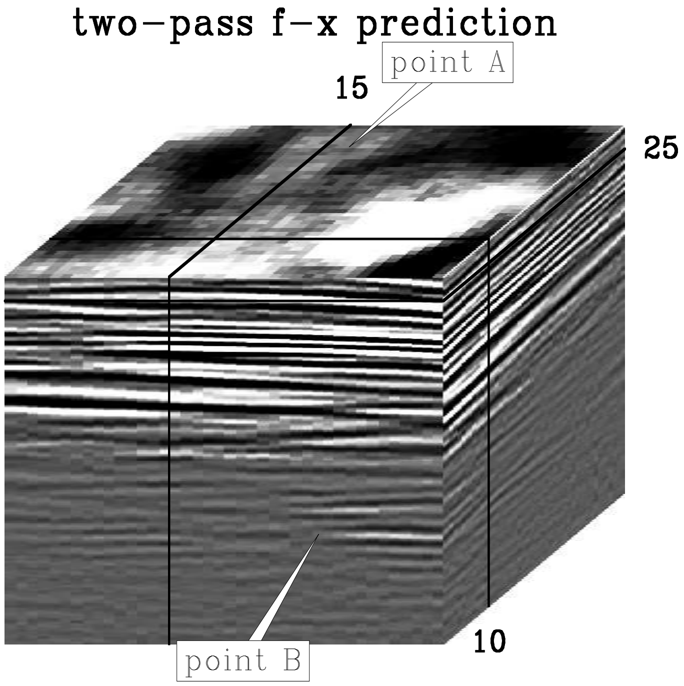

Figure 6: The result of two passes of two-dimensional f-x prediction processing. While this result is an improvement over the input, some of the details is seen to be lost when compared to the three-dimensional f-x prediction. ray1-Arcofx2d [CR]

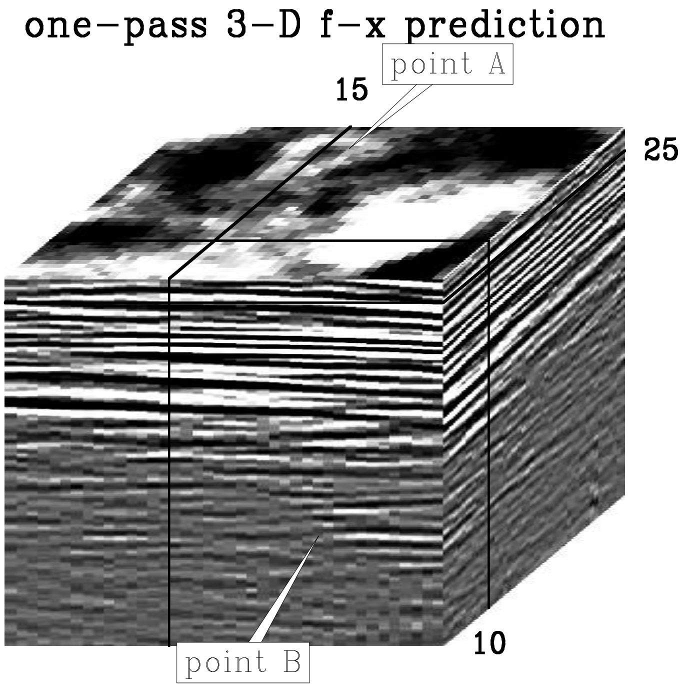

Figure 7: The result of three-dimensional f-x prediction processing. This result preserves the details lost in the two-dimensional f-x prediction. ray1-Arcofx3d [CR]

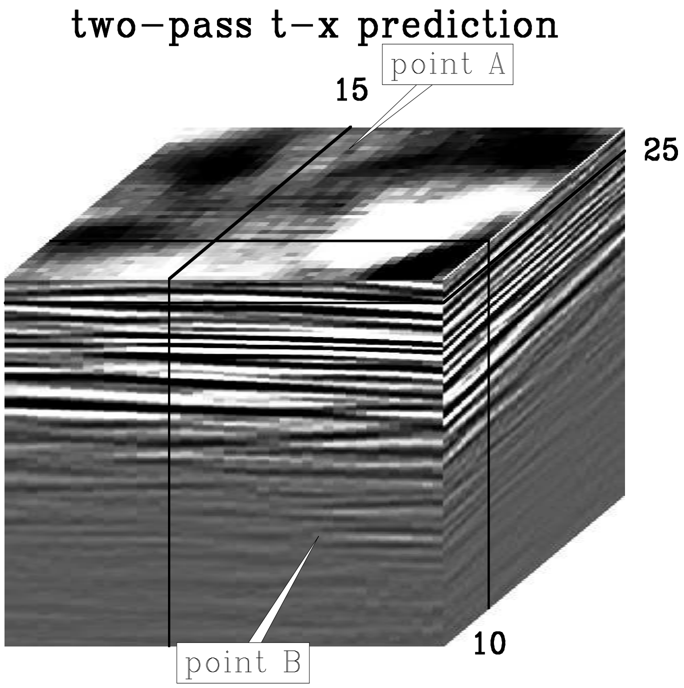

Figure 8: The result of two passes of two-dimensional t-x prediction processing. While this result is an improvement over the input, some of the details is seen to be lost when compared to the three-dimensional t-x prediction. More noise has been attenuated than with the two-dimensional f-x prediction. ray1-Arcotx2d [CR]

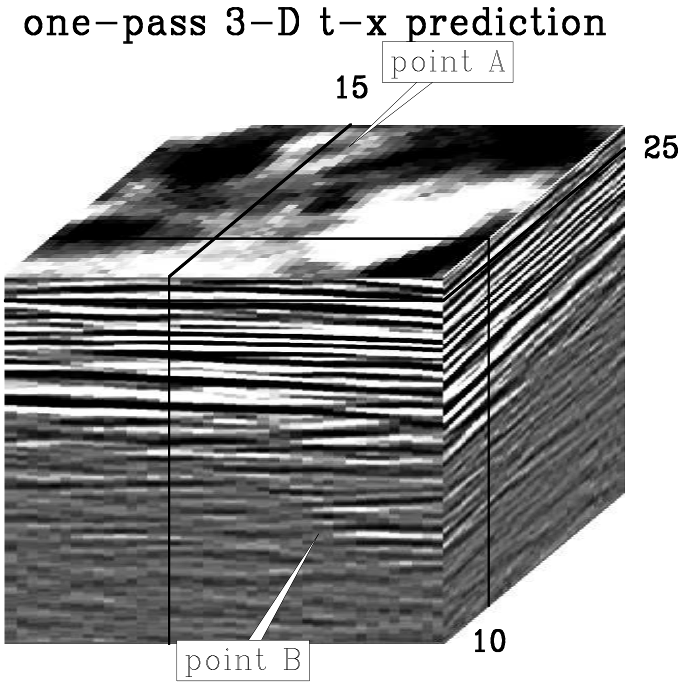

Figure 9: The result of three-dimensional t-x prediction processing. This result preserves the details lost in the two-dimensional t-x prediction. ray1-Arcotx3d [CR]