

Seamless local prediction filtering

Dave Hale

Center for Wave Phenomena, Colorado School of Mines, Golden CO 80401, USA

ABSTRACT

An efficient method for computing local auto-correlations leads to a new method for local adaptive prediction or prediction error filtering of multi-dimensional images. Using a conjugate-gradient method for least-squares optimization, we compute a different prediction filter for each sample in an image. These adaptive prediction filters preserve locally coherent signals, while attenuating random noise.

Key words: digital signal processing, image processing

1 INTRODUCTION

Prediction filters pass those parts of signals that can be predicted, while rejecting any unpredictable parts, the prediction errors.

In some applications, those unpredictable parts represent the information that we seek. We then perform prediction error filtering by subtracting the predicted signal from an input signal. Examples in seismic signal processing include predictive deconvolution and multiple attenuation (Robinson, 1967; Robinson and Treitel, 1980; Spitz, 1999; Guitton et al., 2001), and plane-wave destruction filters (Fomel, 2002).

In other applications, the prediction errors represent random noise that we wish to attenuate. Examples in seismic signal processing include $t - x$ and $f - x$ prediction filtering (Canales, 1984; Abma and Claerbout, 1995; Soubaras, R., 1995; Gülünay, 1986; Gülünay, 2000).

In all of these applications, we compute filter coefficients from the same signals that we wish to filter. Because characteristics of those signals typically vary with time and space, we must often compute and apply filters that vary accordingly.

Today, we typically account for these variations by dividing our signals into some number of overlapping windows, processing each window independently, and then blending the processed windows together in some way to hide the seams between them. Prediction filters computed and applied in this way are *local*, in that each is tuned to a local window of the signal.

Here, I describe an efficient method for computing and applying a different prediction filter for every

sample in a multi-dimensional signal. No blending of windows is necessary; we have no seams to hide. In this sense, our local prediction filters are seamless.

We begin with a review of the most relevant aspects of prediction filtering. We then describe our implementation of local prediction filtering, and provide examples of its application to a seismic image.

2 PREDICTION FILTERING

In linear prediction filtering of a one-dimensional sampled signal $f[i]$, we compute a predicted signal $\tilde{f}[i]$ defined by

$$\tilde{f}[i] \equiv \sum_{j \neq 0} a[j] f[i - j].$$

In least-squares linear prediction filtering, we compute the filter coefficients $a[j]$ to minimize a sum

$$E \equiv \sum_i e^2[i] \tag{1}$$

of squared prediction errors

$$\begin{aligned} e[i] &\equiv f[i] - \tilde{f}[i] \\ &= f[i] - \sum_{j \neq 0} a[j] f[i - j]. \end{aligned}$$

The set of integer indices j for which we compute coefficients $a[j]$ should not include zero, because that would permit a trivial and useless zero-error solution to this least-squares problem. However, the indices j need not be positive. For example, if $f[i]$ is a time series, then including both positive and negative indices j yields a

predicted sample value $\tilde{f}[i]$ that is a linear combination of both past and future input sample values $f[i-j]$.

For any number of filter coefficients $a[j]$, we can minimize the sum E in equation 1 by setting to zero partial derivatives of E with respect to $a[k]$:

$$\begin{aligned} 0 &= \frac{\partial E}{\partial a[k]} \\ &= \sum_i e[i] f[i-k] \\ &= \sum_i f[i] f[i-k] - \sum_{j \neq 0} a[j] \sum_i f[i-j] f[i-k]. \end{aligned} \quad (2)$$

Defining the auto-correlation

$$r[l] \equiv \sum_i f[i] f[i-l] = \sum_i f[i] f[i+l],$$

equation 2 becomes simply

$$\sum_j r[k-j] a[j] = r[k]. \quad (3)$$

Equation 3 represents a system of *normal equations*, one equation for each filter coefficient $a[j]$.

For example, the normal equations for three prediction filter coefficients $a[-1]$, $a[1]$, and $a[2]$ are

$$\begin{bmatrix} r[0] & r[-2] & r[-3] \\ r[2] & r[0] & r[-1] \\ r[3] & r[1] & r[0] \end{bmatrix} \begin{bmatrix} a[-1] \\ a[1] \\ a[2] \end{bmatrix} = \begin{bmatrix} r[-1] \\ r[1] \\ r[2] \end{bmatrix}. \quad (4)$$

As in this example, the normal equations 3 are always symmetric, due to the symmetry of auto-correlations $r[k] = r[-k]$.

For signals $f[i]$ with at least one non-zero sample, this system is also positive definite. That is, for any non-zero sequence $v[k]$,

$$\begin{aligned} &\sum_k \sum_j v[k] r[k-j] v[j] \\ &= \sum_k \sum_j v[k] \left(\sum_i f[i-k] f[i-j] \right) v[j] \\ &= \sum_i \left(\sum_k f[i-k] v[k] \right) \left(\sum_j f[i-j] v[j] \right) \\ &= \sum_i \left(\sum_k f[i-k] v[k] \right)^2 \\ &> 0. \end{aligned}$$

In general, though symmetric and positive-definite, the normal equations 3 are not Toeplitz, as illustrated by the example in equation 4.

3 MULTI-DIMENSIONAL PREDICTION FILTERING

Least-squares linear prediction filtering is easily extended to multi-dimensional signals. In two dimensions, for a signal $f[i_1, i_2]$, we compute a predicted signal

$$\tilde{f}[i_1, i_2] \equiv \sum_{\substack{j_1, j_2 \\ \neq 0, 0}} a[j_1, j_2] f[i_1 - j_1, i_2 - j_2]. \quad (5)$$

We determine the filter coefficients $a[j_1, j_2]$ by minimizing

$$E \equiv \sum_i e^2[i_1, i_2],$$

where

$$\begin{aligned} e[i_1, i_2] &\equiv f[i_1, i_2] - \tilde{f}[i_1, i_2] \\ &= f[i_1, i_2] - \sum_{\substack{j_1, j_2 \\ \neq 0, 0}} a[j_1, j_2] f[i_1 - j_1, i_2 - j_2]. \end{aligned} \quad (6)$$

The solution to this least-squares problem is again the solution to a symmetric positive-definite system of normal equations:

$$\sum_{j_1, j_2} r[k_1 - j_1, k_2 - j_2] a[j_1, j_2] = r[k_1, k_2] \quad (7)$$

where

$$r[l_1, l_2] \equiv \sum_{i_1, i_2} f[i_1, i_2] f[i_1 + l_1, i_2 + l_2]. \quad (8)$$

In two dimensions, each *pair* of indices $[j_1, j_2]$ corresponds to one equation and one unknown $a[j_1, j_2]$. Except for the restriction $[j_1, j_2] \neq [0, 0]$, the choice of these indices is arbitrary and defines the support of the prediction filter. Let that support be represented by two arrays of length M , $lag_1[j]$ and $lag_2[j]$, for $j = 1, 2, \dots, M$. Then Algorithm 1 below computes the 2-D prediction filter.

Algorithm 1 Compute a 2-D prediction filter

```

1: for  $k \leftarrow 1, 2, \dots, M$  do
2:    $k_1 \leftarrow lag_1[k]$ 
3:    $k_2 \leftarrow lag_2[k]$ 
4:   for  $j \leftarrow 1, 2, \dots, M$  do
5:      $j_1 \leftarrow lag_1[j]$ 
6:      $j_2 \leftarrow lag_2[j]$ 
7:      $R(k, j) \leftarrow r[k_1 - j_1, k_2 - j_2]$    ▷ equation 8
8:   end for
9:    $r(k) \leftarrow r[k_1, k_2]$                  ▷ equation 8
10: end for
11: Solve  $\mathbf{R}\mathbf{a} = \mathbf{r}$  for  $\mathbf{a}$                  ▷ normal equations 7
12: for  $j \leftarrow 1, 2, \dots, M$  do
13:    $j_1 \leftarrow lag_1[j]$ 
14:    $j_2 \leftarrow lag_2[j]$ 
15:    $a[j_1, j_2] \leftarrow a(j)$ 
16: end for

```

In lines 7 and 9 of Algorithm 1, the variables $R(k, j)$ and $r(k)$ are elements of the $M \times M$ matrix \mathbf{R} and $M \times 1$ column vector \mathbf{r} , respectively, in the normal equations of line 11. Significant time is saved when computing those elements by exploiting the symmetry $r[l_1, l_2] = r[-l_1, -l_2]$ and computing only values that have not been computed previously.

Here is an example of the normal equations $\mathbf{R}\mathbf{a} = \mathbf{r}$ solved in Algorithm 1 for $M = 3$:

$$\begin{bmatrix} r[0, 0] & r[1, -1] & r[0, -1] \\ r[-1, 1] & r[0, 0] & r[-1, 0] \\ r[0, 1] & r[1, 0] & r[0, 0] \end{bmatrix} \begin{bmatrix} a[1, 0] \\ a[0, 1] \\ a[1, 1] \end{bmatrix} = \begin{bmatrix} r[1, 0] \\ r[0, 1] \\ r[1, 1] \end{bmatrix}. \quad (9)$$

Although symmetric and positive-definite, this system of equations is again not Toeplitz. In some cases, the normal equations for multi-dimensional prediction filters can be converted into block Toeplitz systems (e.g., Claerbout, 1976). But these systems are generally not Toeplitz, which means that we cannot always use the efficient Levinson recursion (e.g., Golub and van Loan, 1989) to solve them.

The extension of Algorithm 1 to higher-dimensional images is straightforward. In any dimension, the number of equations to be solved equals the number of prediction filter coefficients.

4 LOCAL PREDICTION FILTERING

To account for variations of multi-dimensional signals in space and time, we compute seamless local prediction filters from local auto-correlations. Figure 2 displays local auto-correlations for the 2-D seismic image shown in Figure 1. The method described by Hale (2006) computes a local auto-correlation for every sample in that image. Figure 2 shows $225 = 15 \times 15$ lags of each auto-correlation, but for only $1/225$ of the auto-correlations computed.

Let $f[i_1, i_2]$ denote the 2-D image in Figure 1. Then each of the 2-D local auto-correlations displayed in Figure 2 corresponds to a windowed image:

$$f[i_1, i_2; k_1, k_2] \equiv f[i_1, i_2] \times w(i_1 - k_1, i_2 - k_2),$$

where

$$w(x_1, x_2) \equiv e^{-(x_1^2 + x_2^2)/2\sigma^2}$$

defines the Gaussian window.

Gaussian windowing yields auto-correlations in Figure 2 that contain local information about the image of Figure 1. And for each of these local auto-correlations we can use Algorithm 1 to compute a local prediction filter.

Solution of the normal equations in line 11 of Algorithm 1 for every sample in an image may be costly. The computational complexity of solving arbitrary symmetric positive-definite linear systems for M unknowns is $O(M^3)$. For Toeplitz systems, that complexity is only $O(M^2)$, but our systems are generally not Toeplitz.

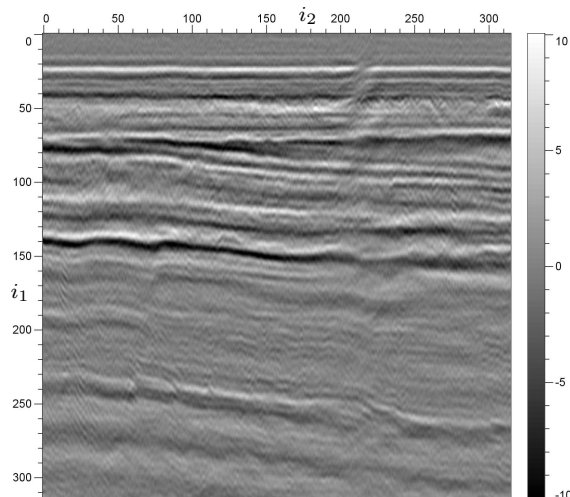


Figure 1. A 2-D seismic image, with 315×315 samples.

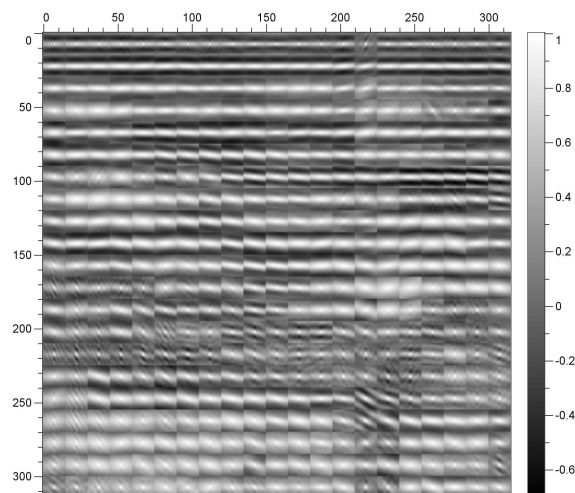


Figure 2. A subset of the local auto-correlations of the image in Figure 1. Shown here are $225 = 15 \times 15$ lags for only $1/225$ of the local auto-correlations computed. The Gaussian half-width is $\sigma = 8$ samples. Each auto-correlation is normalized by its value at lag $[0, 0]$.

Therefore, instead of solving the normal equations directly, we solve them using an iterative conjugate-gradient method.

The method of conjugate gradients is ideal for local prediction filtering. First, the normal equations are symmetric and positive-definite as required by the method. Second, as with any iterative method, a good initial guess can significantly reduce the number of iterations required for convergence to a solution.

In seamless local prediction filtering, we compute a prediction error filter for every sample in our signal. Therefore, a good initial guess for each sample is simply

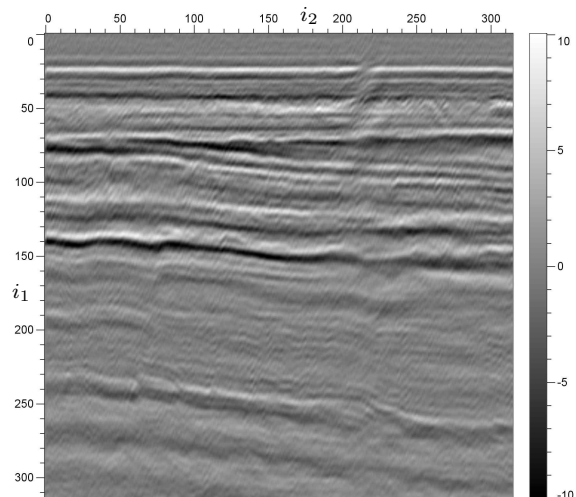


Figure 3. Predicted image for a single prediction filter with three coefficients. Compare with Figures 1 and 5.

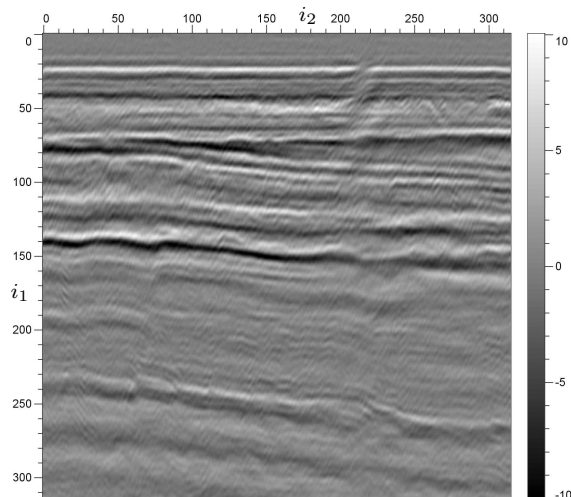


Figure 5. Predicted image for three-coefficient local prediction filters. Compare with Figures 1 and 3.

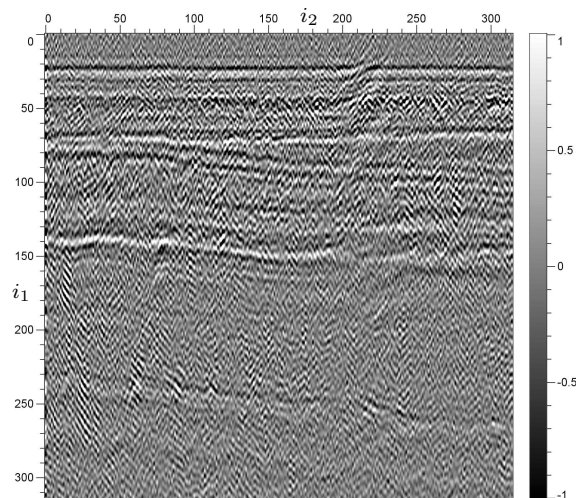


Figure 4. Prediction errors for a single prediction filter with three coefficients. Compare with Figure 6.

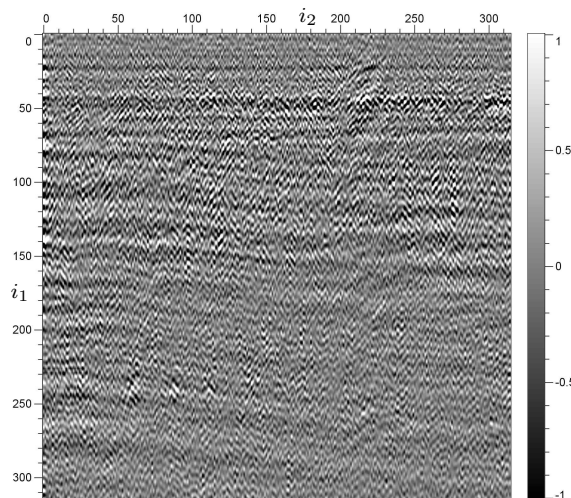


Figure 6. Prediction errors for three-coefficient local prediction filters. Compare with Figure 4.

the prediction filter computed previously for a nearby sample.

That initial guess is best when the system of normal equations is well-conditioned and changes only slightly from one sample to the next. Although the normal equations are positive-definite, systems that are only weakly so may yield prediction filters that vary significantly from one sample to the next. In such cases, convergence to a conjugate-gradients solution may be slow.

5 EXAMPLES

For comparison with local prediction filtering below, we first show the result of computing and applying a

single *global* prediction filter, using only a single auto-correlation of the entire image in Figure 1. Figures 3 and 4 display the predicted image $\tilde{f}[i_1, i_2]$ (equation 5) and prediction errors $e[i_1, i_2]$ (equation 6), respectively.

In this example, the prediction filter is that of equation 9, with only three coefficients:

$$\begin{bmatrix} a[1, 1] & a[1, 0] \\ a[0, 1] & X \end{bmatrix} = \begin{bmatrix} -0.40 & 0.55 \\ 0.84 & X \end{bmatrix}.$$

This stencil shows the weights used in the linear combination of three image samples to obtain a predicted sample in the lower-right corner marked by X .

Differences between the original image in Figure 1 and the predicted image in Figure 3 are difficult to see, because the prediction errors are relatively small. The

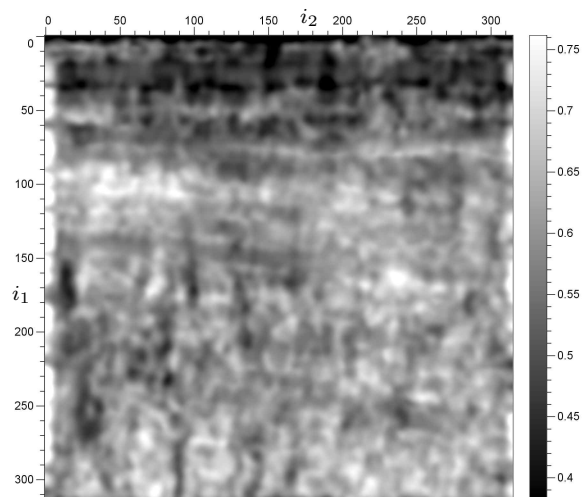


Figure 7. The coefficient $a[1, 0]$ of the local prediction filters used to obtain the predicted image of Figure 5.

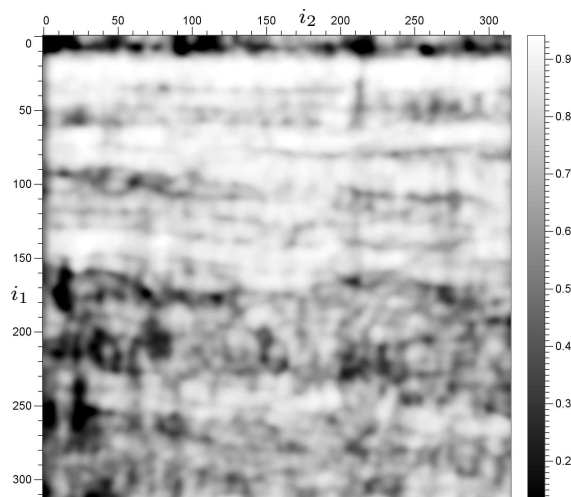


Figure 8. The coefficient $a[0, 1]$ of the local prediction filters used to obtain the predicted image of Figure 5.

prediction errors in Figure 4 have roughly 1/10 of the amplitude of the images in Figures 1 and 3.

Although relatively small, these prediction errors contain a significant amount of correlated signal that is lost in prediction filtering. Correlated signal appears in the prediction errors for two reasons. First, the prediction filter used here has too few coefficients to predict all of the correlated signal in the image. With only three coefficients, some correlated signal is unpredictable. Second, a single global prediction filter cannot account for even the mild variations in bandwidth and orientation of features apparent in this image. To account for such variations, we may use local prediction filters.

Figures 5 and 6 show the predicted image and prediction errors, respectively, after local prediction filtering of the image of Figure 1. The Gaussian half-width in this example is $\sigma = 4$ samples, which enables the local prediction filters used here to vary rapidly.

The prediction errors in Figure 6 for local prediction filtering appear to be less correlated than those in Figure 4. As expected, local prediction filters can better adapt to variations in characteristics of image features.

Each local prediction filter is again that of equation 9, with only three coefficients. We compute these local prediction filters by solving that equation with auto-correlations computed for every image sample. Figures 7–9 illustrate the variations in their coefficients.

The coefficient $a[1, 0]$ in Figure 7 tends to increase from top to bottom, corresponding to a general increase in dominant wavelength of features in Figure 1.

As expected, the coefficient $a[0, 1]$ in Figure 8 is largest for features oriented horizontally in Figure 1. This coefficient is smallest in the lower-left portion of the image, where steeply-dipping coherent signal appears in the image of Figure 1.

The coefficient $a[1, 1]$ in Figure 9 is generally nega-

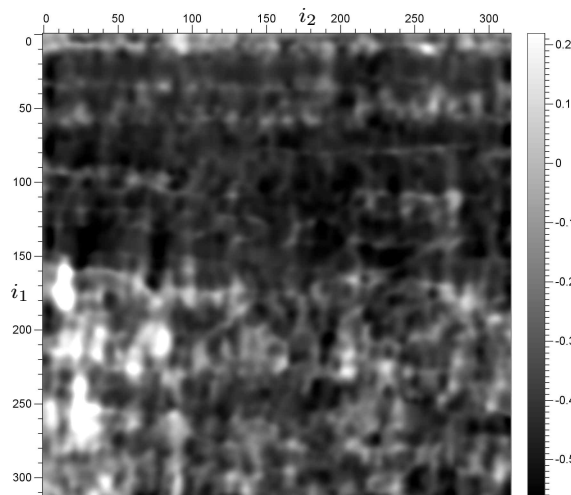


Figure 9. The coefficient $a[1, 1]$ of the local prediction filters used to obtain the predicted image of Figure 5.

tive, except near the steeply-dipping coherent signals in the lower-left part of the image in Figure 1. For these events dipping down to the right, the image sample in the upper-left corner of the prediction filter stencil best predicts the sample in the lower-right corner.

All of the prediction coefficients shown in Figures 7–9 vary significantly and seamlessly throughout the image. Seamless adaptation enables local prediction filters to predict locally correlated signal.

This adaptation is responsible for the reduction in correlated signal observed in the local prediction errors of Figure 6, when compared with the global prediction errors of Figure 4. However, for the three-coefficient local prediction filters used here, some correlated signal

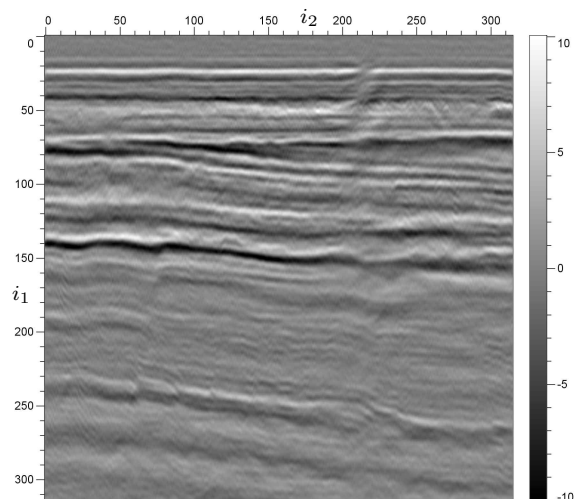


Figure 10. Predicted image for a single prediction filter with 13 coefficients. Compare with Figures 1 and 12.

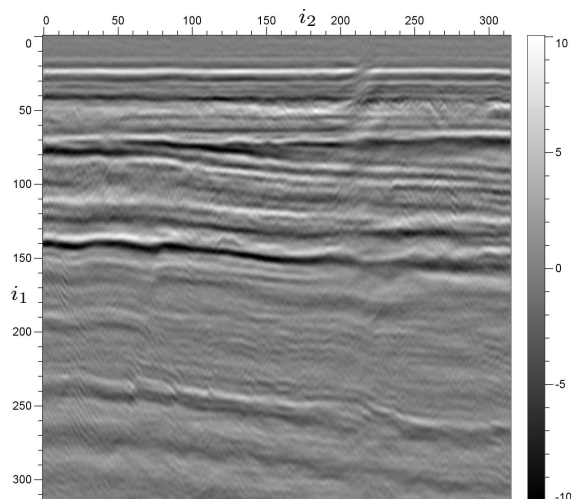


Figure 12. Predicted image for local 13-coefficient prediction filters. Compare with Figures 1 and 10.

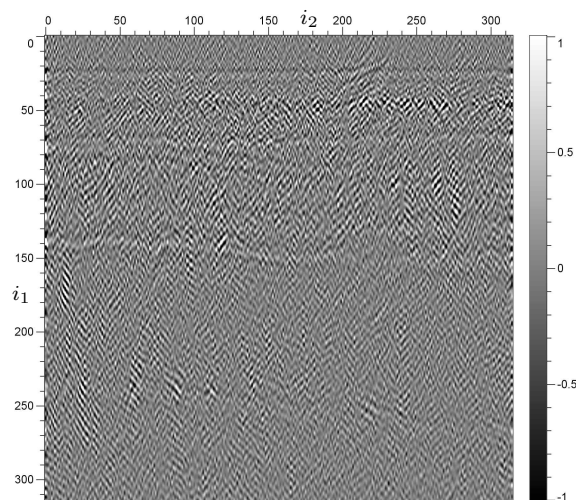


Figure 11. Prediction errors for a single prediction filter with 13 coefficients. Compare with Figures 4 and 13.

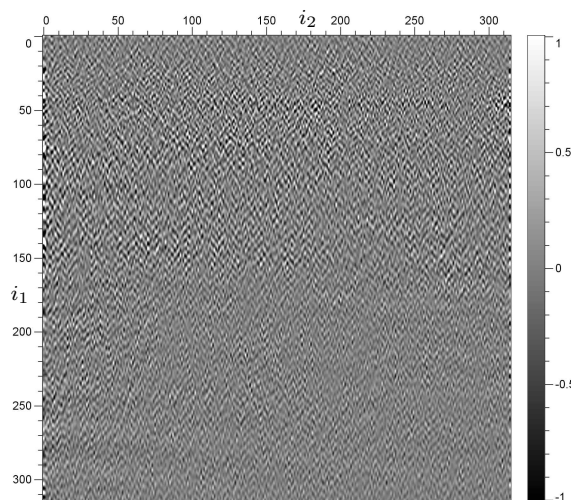


Figure 13. Prediction errors for local 13-coefficient prediction filters. Compare with Figures 6 and 11.

remains unpredictable. To improve prediction, we must use more coefficients.

Figures 10 and 11 show the predicted image and prediction errors for a single prediction filter with 13 coefficients. The stencil for this prediction filter is

-0.092	0.056	-0.19	0.54	-0.25	0.024	-0.064
0.086	-0.006	0.41	X	0.41	-0.012	0.085

Figures 12 and 13 show the predicted image and prediction errors for local prediction error filtering with the same number of coefficients and the same stencil. In this example, we computed local prediction filters using a Gaussian window with half-width $\sigma = 8$ samples. An average of 4.1 conjugate-gradient iterations per image sample were required to solve the normal equations

$\mathbf{R}\mathbf{a} = \mathbf{r}$ in Algorithm 1 for the coefficients \mathbf{a} of each local prediction filter.

6 CONCLUSION

The examples of prediction filtering shown in this paper are not dramatic, primarily because samples in these images are so highly predictable from neighboring samples. Amplitudes of prediction errors are only about 1/10 the amplitude of the original or predicted image samples.

Nevertheless, prediction errors obtained with local prediction filters contain less correlated signal than those obtained with a single global prediction filter. The examples demonstrate that local prediction filtering can

adapt to variations in signal characteristics, and is less likely to attenuate locally coherent signal.

The rate of adaptation is controlled by the width of Gaussian windows used to compute local auto-correlations. That width and prediction filter stencil (support) are parameters that we must specify.

We need not specify a number of windows or an amount of window overlap. Such parameters are typical of window-filter-blend methods for adaptive filtering. In seamless local prediction filtering, each predicted sample has a corresponding window centered on that sample, and the amount of window overlap is almost as large as the window itself.

The method for seamless local prediction error filtering described here is certainly more costly than computing and applying a single global prediction filter. For each sample in an image, we must compute required lags of a local auto-correlation and then solve a system of linear equations. Because those equations are symmetric and positive-definite, we can use efficient conjugate-gradient iterations to solve them.

These iterations are reminiscent of other forms of seamless adaptive filtering (e.g., Widrow and Stearns, 1985), in which a prediction filter for one sample is updated to obtain a prediction filter for a future sample. In our method, conjugate gradient updates are performed using local auto-correlations that have been pre-computed for every image sample.

REFERENCES

- Abma, R., and Claerbout, J., 1995, Lateral prediction for noise attenuation by t-x and f-x techniques: *Geophysics*, v. 60, no. 6, p. 1887–1896.
- Canales, L.L., 1984, Random noise reduction: 54th Annual International Meeting, SEG, Expanded Abstracts, p. 525–527.
- Claerbout, J.F., 1976, *Fundamentals of geophysical data processing*: McGraw-Hill.
- Fomel, S., 2002, Applications of plane-wave destruction filters: *Geophysics*, v. 67, no. 6, p. 1946–1960.
- Golub, G.H., and Van Loan, C.F., 1989, *Matrix computations* (2nd Edition): John Hopkins University Press.
- Guitton, A., Brown, M., Rickett, J., and Clapp, R., 2001, Multiple attenuation using a t-x pattern-based subtraction method: 71st Annual International Meeting, SEG, Expanded Abstracts, p. 1305–1308.
- Gülünay, N., 1986, F-x deconvolution and complex Wiener prediction filter: 56th Annual International Meeting, SEG, Expanded Abstracts, p. 279–281.
- Gülünay, N., 2000, Noncausal spatial prediction filtering for random noise reduction on 3-D poststack data: *Geophysics*, v. 65, no. 5, p. 1641–1653.
- Hale, D., 2006, An efficient method for computing local cross-correlations of multi-dimensional signals: this report.
- Robinson, E., 1967, Predictive decomposition of time series with application to seismic exploration: *Geophysics*, v. 32, no. 3, p. 418–484.
- Robinson, E., and Treitel, S., 1980, *Geophysical signal analysis*: Prentice-Hall.
- Soubaras, R., 1995, Prestack random and impulsive noise attenuation by f-x projection filtering: 65th Annual International Meeting, SEG, Expanded Abstracts, p. 711–714.
- Spitz, S., 1999, Pattern recognition, spatial predictability, and subtraction of multiple events: *The Leading Edge*, v. 18, no. 1, p. 55–58.
- Widrow, B., and Stearns, S.D., 1985, *Adaptive signal processing*: Prentice-Hall.

