*Stanford Exploration Project, Report 72, December 18, 1997, pages 1–32*

*Stanford Exploration Project, Report 72, December 18, 1997, pages 1–32*

# 2-D interpolation with Spitz's method

*Ovidiu Balog*[1]

***keywords:*** *interpolation, linear, algorithm, Fourier transform*


## ABSTRACT

I have implemented Spitz's method to interpolate missing data in the presence of aliasing. The program was tested on synthetic and real data. Despite its ability to deal with aliasing, the program has some limitations inherent in the method.


## INTRODUCTION

Most standard interpolation techniques try to recover the true dips by performing a local search on the data (Bardan, 1987). Those techniques are very sensitive to the noise level and perform poorly when crossing events are present. Claerbout proposed a different approach by interpolating with spatial predictors in the $(t, x)$ domain (Claerbout, 1991). In the same time Spitz presented a method to interpolate seismic traces beyond aliasing in the $(\omega, x)$ domain (Spitz, 1991).

The Stanford Exploration Project (SEP) devotes a great deal of effort toward understanding interpolation of aliased seismic data (Nichols, 1990, Ji, 1991). This paper deals with the implementation of Spitz's method (Spitz, 1991) and the evaluation of the results obtained by testing the program on synthetic and real data. In the first part I review the theory, in the second I describe the code and in the last I discuss several results and provide an example where the program fails.

## THE METHOD

In the discussion to follow the predictability of linear events in the $(\omega, x)$ domain and a relation between the filters predicting the original and the interpolated data will be presented. This theoretical work is derived in the papers by Spitz and Canales (Spitz, 1991, Canales, 1984).

The data $u(t, x)$ is assumed to be a collection of $N$ equally spaced traces that is composed of linear events corresponding to $L$ distinct dips, with $L < N$. In the

[1]**email:** not available

case when several lines are parallel, they are treated as part of a single event with a unique dip. It is also assumed that there is no time aliasing and that the signals are invariant along the linear events. Following Canales (Canales, 1984) and with these assumptions, the data can be expressed in the $(t,x)$ domain as

$$u(t,x) = \sum_{k=1}^{L} v_k(t - b_k x) \tag{1}$$

or in the $(\omega, x)$ domain after a Fourier transform over the time axis as

$$U(\omega,x) = \sum_{k=1}^{L} V_k(\omega)\exp(i\omega b_k x), \tag{2}$$

where $v_k(t)$ is a wavelet, $V_k(\omega)$ its Fourier transform and $b_k$ is a measure of the slope, all corresponding to the $k^{th}$ linear event. If equation (2) is sampled in the $x$ direction and we let $b_k$ become the time shift per trace interval corresponding to the $k^{th}$ linear event, then the data can be expressed in the following matrix form :

$$\begin{pmatrix} U_1(\omega) \\ U_2(\omega) \\ \vdots \\ \vdots \\ U_N(\omega) \end{pmatrix} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ z_1 & z_2 & \dots & z_L \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ z_1^{N-1} & z_2^{N-1} & \dots & z_L^{N-1} \end{pmatrix} \begin{pmatrix} V_1(\omega) \\ V_2(\omega) \\ \vdots \\ V_L(\omega) \end{pmatrix} \tag{3}$$

or

$$U(\omega) = \mathbf{Z}(\omega) \times V(\omega), \tag{4}$$

where $U(\omega)$ is a frequency slice on the data $U(\omega,x)$, $z_k = \exp(i\omega b_k)$ and $V_k(\omega)$ is the wavelet from (2). The Vandermonde matrix involved in equation (3) is of full column rank since the dips are distinct. Therefore every row of this matrix may be expressed as a linear combination of the $L$ previous rows :

$$(\, z_1^{k-1} \quad \dots \quad z_L^{k-1} \,) = (\, P_L(\omega) \quad \dots \quad P_1(\omega) \,) \begin{pmatrix} z_1^{k-L-1} & \dots & z_L^{k-L-1} \\ \vdots & & \vdots \\ z_1^{k-2} & \dots & z_L^{k-2} \end{pmatrix} \tag{5}$$

for $k = L+1, \dots, N$. Notice that the coefficients $P_1(\omega)$ to $P_L(\omega)$ are independent of the row number $k$ because of the Vandermonde structure. Multiplying on both sides by $V(\omega)$, equation (5) becomes :

$$U_k(\omega) = (\, P_L(\omega) \quad P_{L-1}(\omega) \quad \dots \quad P_1(\omega) \,) \begin{pmatrix} U_{k-L}(\omega) \\ U_{k-L-1}(\omega) \\ \vdots \\ U_{k-1}(\omega) \end{pmatrix} \tag{6}$$

or

$$U_k(\omega) = \sum_{j=1}^{L} P_j(\omega) U_{k-j}(\omega) \tag{7}$$

for $k = L+1, \ldots, N$. The meaning of the last equation is that in the case of strictly linear events, for each frequency, the data $U(\omega)$ is predictable along the $x$ axis with a one-step-ahead prediction filter.

Considering equation (5) for $k = L+1$ it appears that $z_1$ to $z_L$ are the roots of the polynomial $Q(z) = z^L - P_1(\omega)z^{L-1} - \cdots - P_L(\omega)$ since in this case equation (5) is equivalent to $(Q(z_1) \ \ldots \ Q(z_L)) = (0 \ \ldots \ 0)$. Therefore the relations between the roots $z_j$ and the the coefficents $P_k(\omega)$ are given by

$$P_1(\omega) = \sum_{j=1}^{L} z_j \quad \mathit{and} \quad P_k(\omega) = (-1)^{k+1} \sum_{1 \le j_1 < \cdots < j_k \le L} z_{j_1} \cdots z_{j_k}, \tag{8}$$

for $k = 2, \ldots, L$. Multiplying those equations by the conjugate of $P_L(\omega)$ gives $P_L^*(\omega)P_k(\omega) = -P_{L-k}^*(\omega)$ which shows that the $z_j$ are also the roots of $z^{k-1}P_L^*(\omega)Q(z) = P_L^*(\omega)z^{k+L-1} + P_{L-1}^*(\omega)z^{k+L-2} + \cdots + P_1^*(\omega)z^k - z^{k-1}$ for $k = 1, \ldots, N-L$. This fact can be expressed in a form equivalent to equation (5)

$$\begin{pmatrix} z_1^{k-1} & \ldots & z_L^{k-1} \end{pmatrix} = \begin{pmatrix} P_1^*(\omega) & \ldots & P_L^*(\omega) \end{pmatrix} \begin{pmatrix} z_1^k & \ldots & z_L^k \\ \vdots & & \vdots \\ z_1^{k+L-1} & \ldots & z_L^{k+L-1} \end{pmatrix} \tag{9}$$

and after multiplication by $V(\omega)$ as

$$U_k(\omega) = \begin{pmatrix} P_1^*(\omega) & P_2^*(\omega) & \ldots & P_L^*(\omega) \end{pmatrix} \begin{pmatrix} U_{k+1}(\omega) \\ U_{k+2}(\omega) \\ \vdots \\ U_{k+L}(\omega) \end{pmatrix} \tag{10}$$

or

$$U_k(\omega) = \sum_{j=1}^{L} P_j^*(\omega) U_{k+j}(\omega) \tag{11}$$

for $k = 1, \ldots, N-L$. Like equation (7), equation (11) states that in the case of strictly linear events, for each frequency, the data $U(\omega)$ is also predictable along the $x$ axis with a one-step-backward prediction filter.

Let $(P'_L(\omega), \ldots, P'_1(\omega))$ be the prediction filter and $b'_k$ with $k = 1, \ldots, L$ be the time shift per trace interval for the interpolated data $U'(\omega, x)$; also let $(P_L(\omega), \ldots, P_1(\omega))$ and $b_k$ with $k = 1, \ldots, L$ be the same parameters for the original data $U(\omega, x)$. The roots of $Q'(z) = z^L - P'_1(\omega)z^{L-1} - \cdots - P'_L(\omega)$ are

$$z_k = \exp(i\omega b'_k) = \exp(i\omega \frac{b_k}{n}) = \exp(i\frac{\omega}{n} b_k) \tag{12}$$

if a $n^{th}$ order interpolation is to be considered (the trace interval before interpolation equals $n$ times the trace interval after interpolation). Since

$$P'_1(\omega) = \sum_{j=1}^{L} z_j \quad \textit{and} \quad P'_k(\omega) = (-1)^{k+1} \sum_{1 \leq j_1 < \cdots < j_k \leq L} z_{j_1} \cdots z_{j_k}, \tag{13}$$

replacing $z_j$ in equation (13) with $\exp(i\frac{\omega}{n}b_k)$ from equation (12) gives

$$P'_k(\omega) = P_k(\frac{\omega}{n}) \tag{14}$$

for $k = 1, \ldots, L$.

As stated at the beginning of this section, equations (7) and (11) show that linear events can be one-step forward-backward predicted and that the coefficients of the prediction filters for the original and the interpolated data are related through equation (14). Those features can be used to implement an interpolation program, which is further discussed in the following section.

## THE PROGRAM

The algorithm suggested by the results of the previous section is to compute from the time Fourier transformed original data $U(\omega, x)$, for each frequency $\omega$, a prediction filter along the $x$ axis which is then used at the frequency $n \times \omega$ to solve for the missing data. I have implemented such an algorithm in the case where $n$ is equal to two.

Since the operations are performed in the $(\omega, x)$ domain, the data $u(t, x)$ as well as a zero padded version $u_{pad}(t, x)$ of the data, with twice as many time samples as $u(t, x)$, are both Fourier transformed over the time axis. The one-step forward prediction filters $(P_1(\omega), \cdots, P_L(\omega))$ are then computed for all the frequencies from zero to half the Nyquist by solving using Levinson's recursion the following linear system

$$\mathbf{R}(\omega) \begin{pmatrix} 1 \\ -P_1(\omega) \\ \vdots \\ -P_L(\omega) \end{pmatrix} = \begin{pmatrix} e \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{15}$$

where $\mathbf{R}(\omega)$ is the autocorrelation matrix of a frequency slice from $U_{pad}(\omega, x)$ and $e$ is the minimum prediction error energy.

According to equation (12), for $n = 2$, $(P_1(\omega), \cdots, P_L(\omega))$ predicts one step ahead the interpolated data $U'(2\omega, x)$. This relation may be thought of as a convolutional operation

$$P(\omega) * U'(2\omega, x) = 0 \tag{16}$$

where $P(\omega) = (P_1\omega, \cdots, P_L(\omega))$ and $U'(2\omega, x)$ contains the original data interlaced with missing values. $P(\omega)$ is then separated into an even part and an odd part, respectively $P_e(\omega)$ and $P_o(\omega)$, and $U'(2\omega, x)$ into missing and known data, respectively $M(2\omega, x)$ and $K(2\omega, x)$. $M(2\omega, x)$ is then obtained from

$$M(2\omega, x) = -P_o^{-1}(\omega) * (P_e(\omega) * K(2\omega, x)). \tag{17}$$

The same computations are performed with the one-step backward prediction filters $(P_1^*(\omega), \cdots, P_L^*(\omega))$ giving another set of values for the missing data. The two values found for the missing data in each case are then averaged to give the final result. The interpolated data $U'(\omega, x)$ is then time inverse fourier transformed to $u'(t, x)$.

It is worth noticing that the backward prediction filters are easily computed from the forward prediction filters and that the zero padding in time of $u(t, x)$ is necessary to avoid any type of filter interpolation in the $\omega$ direction between even frequencies ($\omega_k = 2k\pi/N$, with $k$ even).

## EXAMPLES

In this section I discuss the results obtained by applying the program to real and synthetic data. For testing the algorithm I used a series of simple linear events and a real dataset from a seismic crosswell experiment.

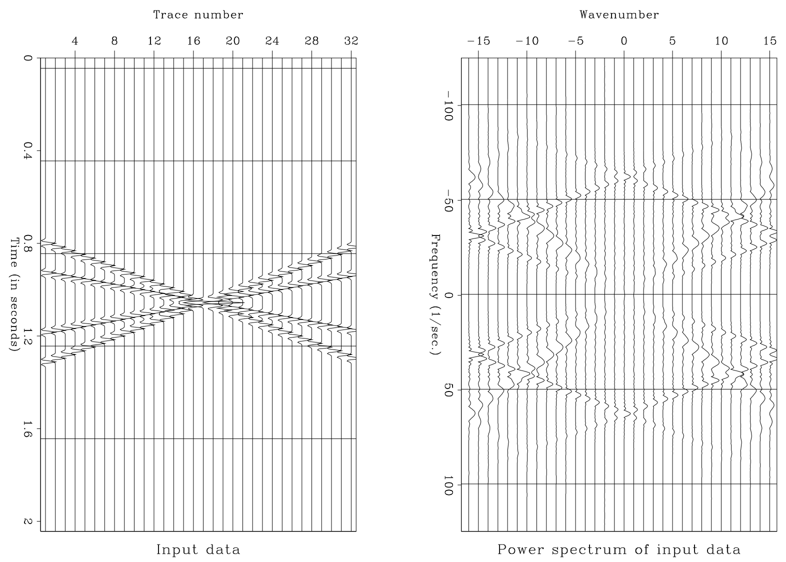



Figure 1: Data before interpolation. Notice the two strongly aliased events. ovid-f1 [NR]

Figures 1 and 2 show an example of interpolation on simple linear events. The input data consists of four events, the steepest ones strongly aliased while the other two suffer only first order aliasing, meaning that after one interpolation they should be unaliased. The program performs very well despite the presence in the frequency wavenumber domain of some of the aliased energy after the interpolation is performed.

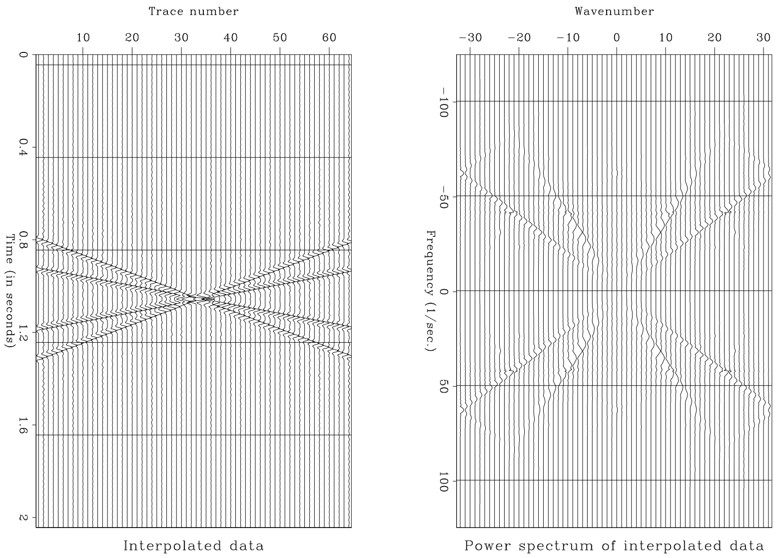



Figure 2: Data after interpolation. Despite the good performance (left) a small amount of aliased energy still remains after interpolation (right). ovid-f2 [NR]

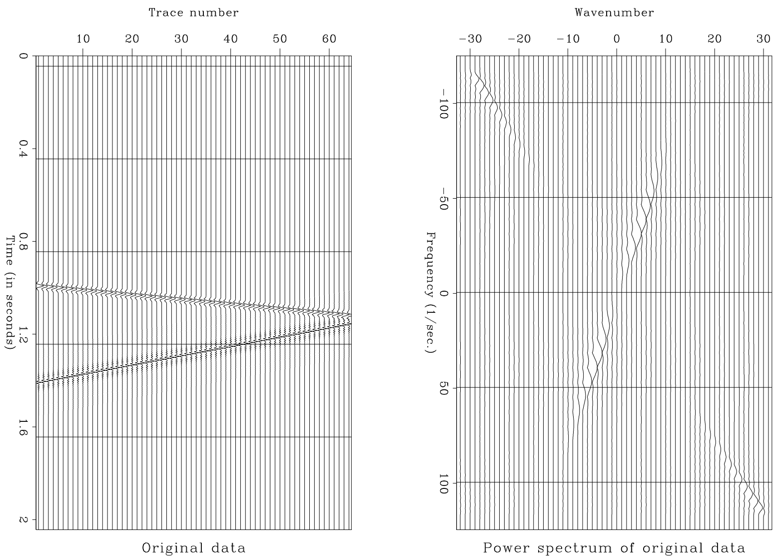



Figure 3: Original data containing a high frequency event (left, lower event). ovid-f3 [NR]

For the second example the original data is composed of two simple linear events. By looking at the power spectrum in Figure 3, one can easily notice that none of them is aliased and that one is a lowpass while the other is a high pass. After subsampling once in the $x$ direction the high pass becomes aliased while the low pass is still unaliased. The subsampled data shown in Figure 4 is used as an input to the program.

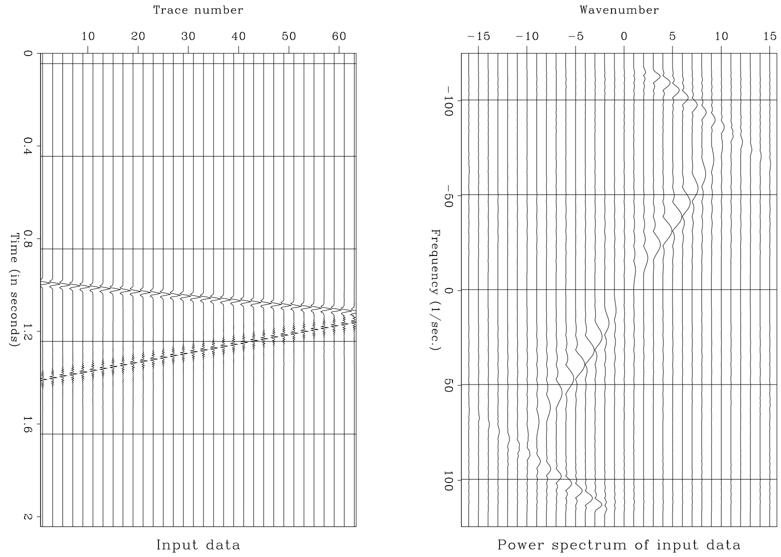



Figure 4: Data before interpolation. A subsampling was performed on the original data so that the high frequency event (left, lower) became aliased (right). ovid-f4 [NR]

By comparing the result of the interpolation shown in Figure 5 to the original data in Figure 3 it appears that the high pass was not correctly interpolated since the aliased energy was not moved to the right place in the spectrum. Furthermore a windowed portion of both the original and the interpolated data in Figure 6 shows that the missing traces are replicas of the known traces with a sign change.

This example shows a limitation of the method. It fails to interpolate correctly when the data contains high frequency events with energy only above half the Nyquist frequency. Therefore the method should be used carefully when such events are present in the data to interpolate Nevertheless this is a very extreme case and it was not encountered with the real data that I processed for this paper.

The third example was obtained by applying the program to a real dataset from a crosswell experiment. For this example the interpolation was performed in overlapping windows containing 32 traces and 512 time samples. Figure 7 shows the original with the strong tubewaves which are aliased. The spacing between the stations in the borehole was five meters for the original data. The data was interpolated twice to reduce the spacing to 1.25 meters so that the strong tube waves became unaliased. Figures 8 and 9 show the data after interpolation. An FK filter was applied to remove the tube wave on both original and interpolated data and then the upgoing and downgoing wavefields were separated. In Figure 10 the upgoing wavefield is shown

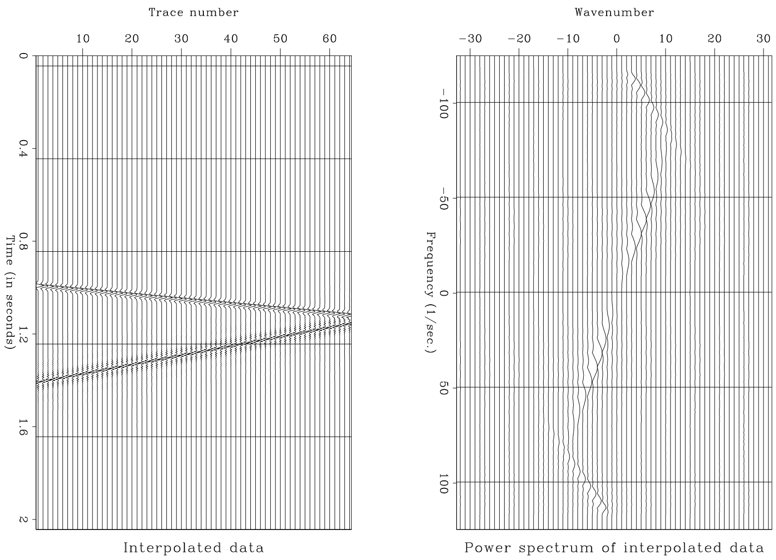



Figure 5: Data after interpolation. The low frequency event (left, upper) was perfectly interpolated. The aliased energy of the high frequency event was not moved at the correct place in the spectrum (right). $\boxed{\text{ovid-f5}}$ [NR]

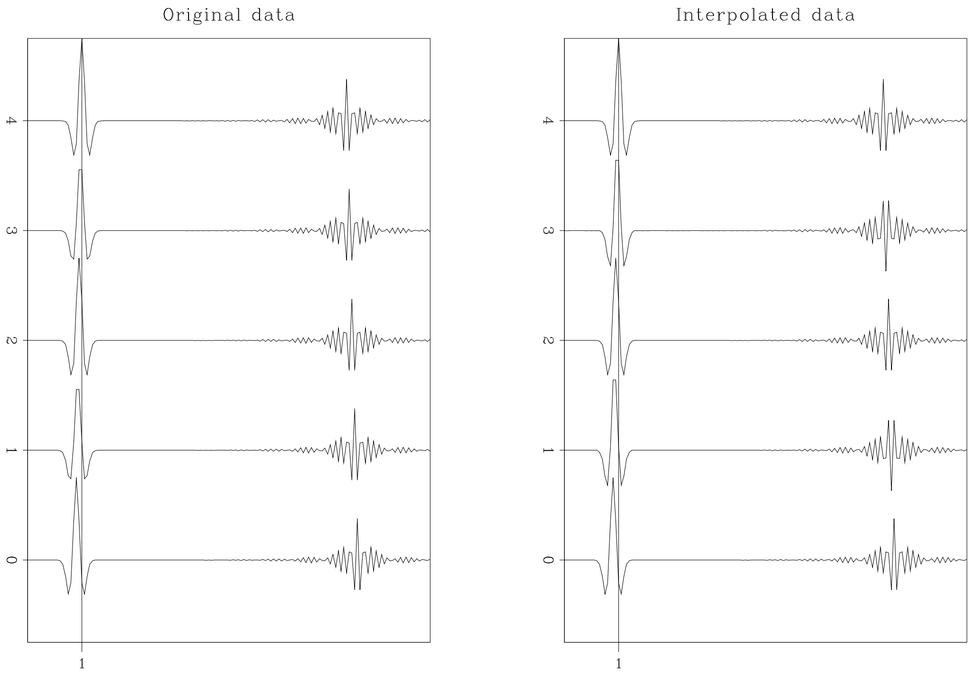



Figure 6: Window on original and interpolated data showing that for the high frequency event the missing traces and the known traces have the same wavelet but with a sign change. $\boxed{\text{ovid-f6}}$ [NR]

before and after interpolation. The reflections in the upper part of the plot are more clear when the processing was performed on the interpolated data.

## CONCLUSIONS

I have implemented Spitz's interpolation method and tested it on real and synthetic data. This method is based upon two theoretical properties of linear events, the first of which is the predictability along the $x$ direction of such events in the $(\omega, x)$ domain. The second property is that the filters predicting the interpolated data at the frequency $\omega$ also predict the original data at the frequency $\omega/n$ if an $n^{th}$ order interpolation is to be considered. The results show that the method performs well when the data has significant energy in the lower part of the spectrum; on the contrary if the energy is concentrated at above half the Nyquist frequency, the method performs poorly.

## REFERENCES

Bardan, V., 1987, Trace interpolation in seismic data processing: Geophys. Prosp., **35**, 343–358.

Canales, L.L., 1984, Random noise reduction: Presented at the 54th Ann. Internat. Mtg., Soc. Expl. Geophys.

Claerbout, J.F., 1991, Earth Soundings Analysis: Processing versus Inversion: Blackwell Scientific Publications.

Claerbout, J.F., 1991, Missing data restoration: SEP–**70**, 305–335.

Ji, J., 1991, Trace interpolation using spectral estimation: SEP–**72**, 57–64.

Nichols, D., 1990, Estimation of missing data by least squares: SEP–**65**, 271–293.

Spitz, S., 1991, Seismic trace interpolation in the F-X domain: Geophysics, **56**, 785–794.

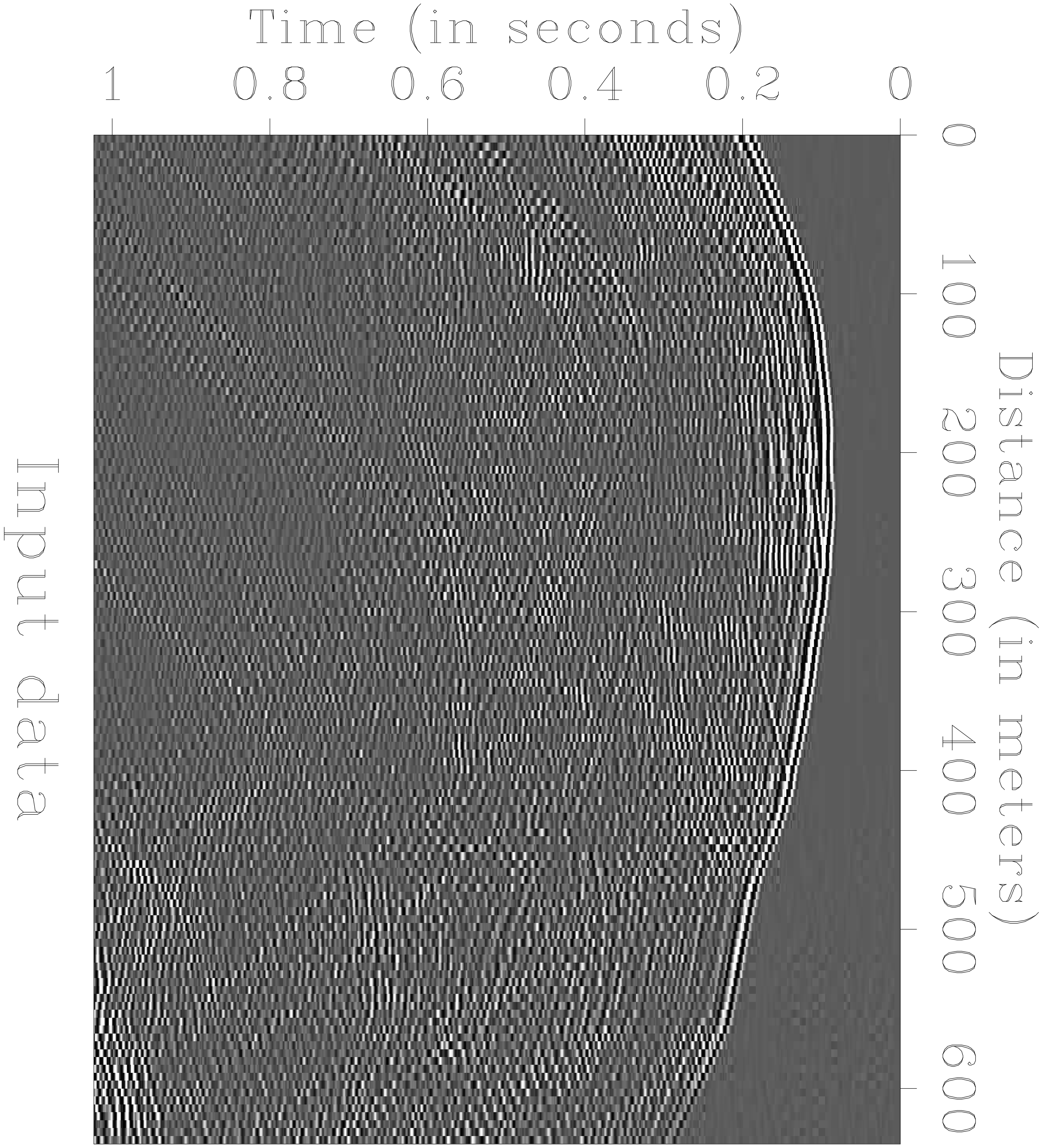



Figure 7: Crosswell data. Notice the aliased tubewaves. ovid-f7 [NR]

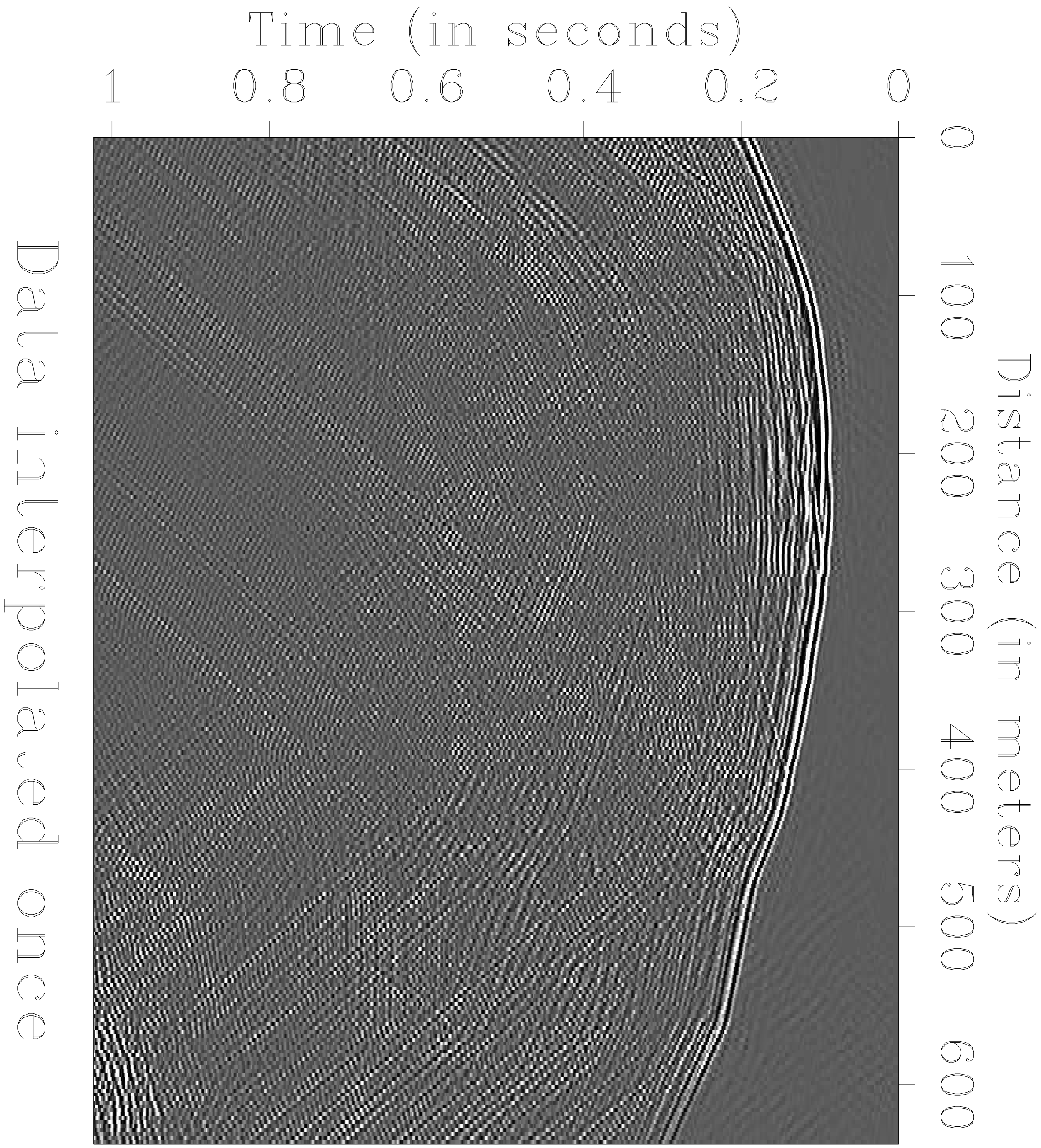



Figure 8: Crosswell data interpolated once. $\boxed{\text{ovid-f8}}$ [NR]

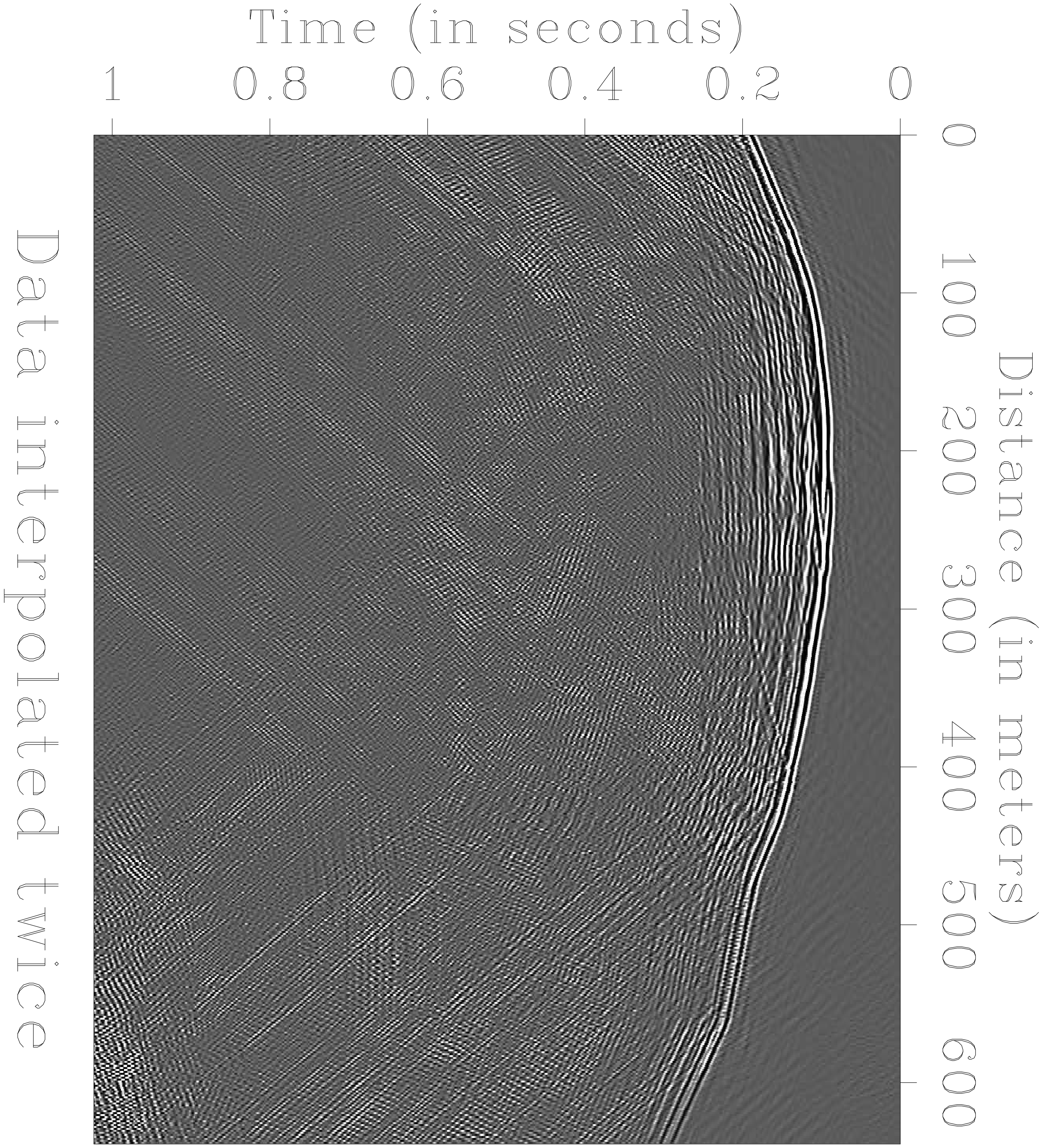
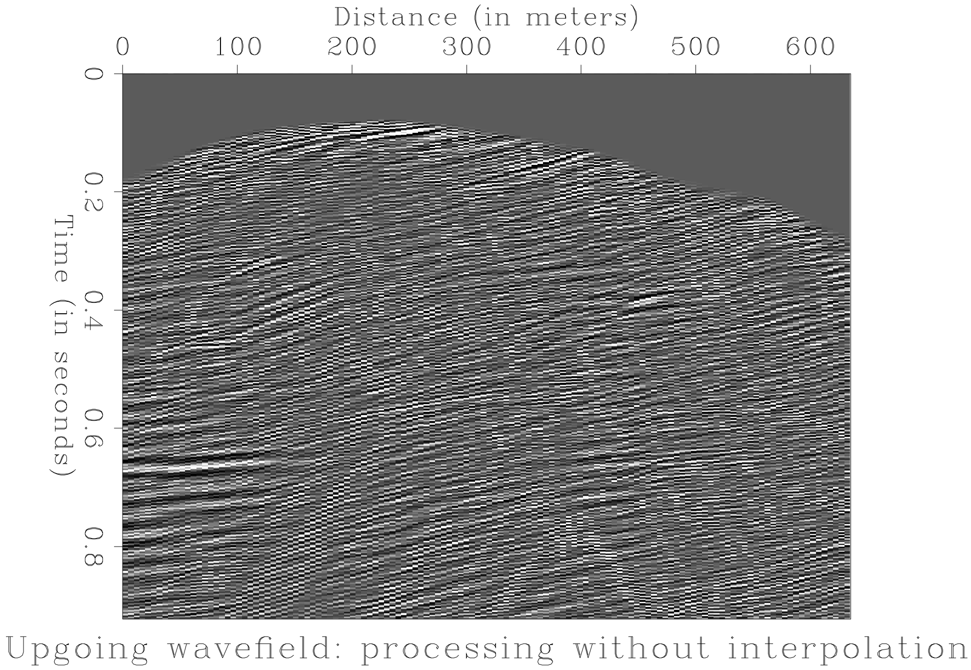



Figure 9: Crosswell data interpolated twice. ovid-f9 [NR]



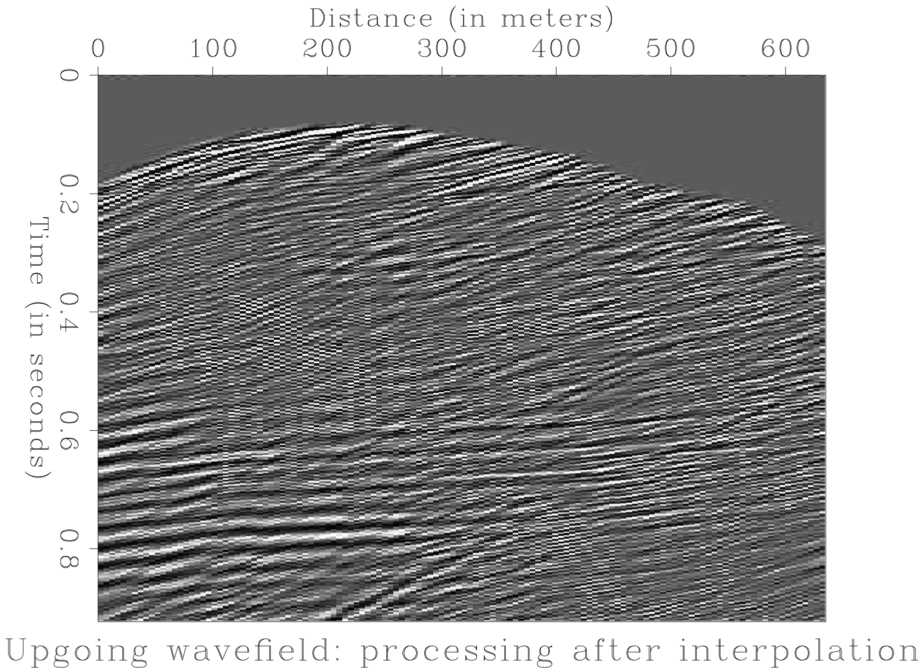




Figure 10: Upgoing wavefield. The early arrivals (reflections) appear more coherent when the FK filter is applied on the interpolated data. ovid-f10 [NR]