Stanford Exploration Project, Report 65, January 13, 1998, pages 269–292

Estimation of missing data by least squares

Dave Nichols[1]

keywords: *Fourier transform, missing data, linear, interpolation, estimation, nonlinear, algorithm*

ABSTRACT

I formulate the problem of estimating missing data using least squares. The operators that I use are error operators that have as output that part of the interpolated data that does not fit some parametric model of the data. If all the parameters of the model are known the problem is a linear least squares problem. If the parameters of the model must be estimated at the same time as the missing data the problem is non-linear. I use a model based on local linear events to interpolate aliased data. This procedure depends on good initial estimates of the dips which can be obtained from a smoothed version of the data.

INTRODUCTION

We can estimate missing data from the surrounding data values if we make some assumptions about the characteristics of the interpolated output dataset. One commonly made assumption is that the data is locally smooth i.e. the output spectrum is low-pass. Other possible assumptions are: the data can be well approximated by a polynomial function, the data can be modeled as a small number of linear events, the data has a known output spectrum or the output spectrum can be estimated from the input spectrum.

In this paper I use our knowledge of the output data characteristics to design an error operator. The output of the operator is the residual part of the data that does not conform to our model. For example, if the output data is assumed to be smooth then a high pass operator should produce an output with little energy. Given such an operator I choose the interpolated data values to minimize the energy in the output. I cast this problem as a linear least squares problem that can be solved by standard numerical techniques.

If the parameters characterizing the output data are unknown there are two alternatives. The first is to estimate the parameters from the input data. This then reduces to the linear problem described above. However, there may be problems if the

[1]**email:** not available

missing data points influence the estimation of the parameters or if the parameters are estimated far from the missing data points. The second alternative is to perform a joint estimation of the output data parameters and the missing data values. I do this by estimating the missing data values and the parameters describing the output data in the same least squares problem. This produces a nonlinear problem that is more expensive to solve but also more stable.

When non-linear least-squares problems are solved it is important to have reasonable initial values for the model parameters. I demonstrate a missing data estimation technique that can be used to interpolate aliased data if good initial estimates of the dips present in the data are available. I also show how these initial estimates can be obtained from a smoothed version of the data. The combination of these two methods can be used as an automatic interpolation technique for aliased data.

LINEAR ESTIMATION OF MISSING DATA

Consider a list of samples with some elements missing

$$\phi = \{d_1, d_2, m_1, d_3, d_4, d_5, m_2, \ldots, m_m, d_l\};$$

there are l known data samples, $\{d_1, \ldots, d_l\}$, and m missing data samples, $\{m_1, \ldots, m_m\}$. I wish to estimate values for the missing samples. If I have some model of the expected output I can choose interpolated values that minimize the power of the residual data that does not obey the model. e.g. If we have an expected band limited spectrum we can minimize the power of the output spectrum outside the expected band. In general I write the problem as one of minimizing the residuals resulting from applying a linear operator to the data. The model parameters that can be varied in the minimization process are the missing data values.

I define two new vectors k and u. The known data, k, is a vector of length n with zeros in place of the unknown values. The unknown data, u, is a vector of length n with zeros is place of the known values. Then,

$$\phi \equiv \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \vdots \\ \phi_{n-1} \\ \phi_n \end{pmatrix} = k + u \equiv \begin{pmatrix} d_1 \\ d_2 \\ 0 \\ \vdots \\ 0 \\ d_l \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ m_1 \\ \vdots \\ m_m \\ 0 \end{pmatrix}$$

The objective function I minimize is

$$|R \cdot \phi|^2 = |R \cdot u + R \cdot k|^2$$

This is easily related to the standard form of linear problems, $Am = y$ where $y = -R \cdot k$ and $A = R \cdot S$. Here R is the user supplied operator that is derived from the assumptions made about the output. S is a scatter/gather operator that maps the vector of unknowns, m, into the vector u.

One simple example of a suitable linear operator is a roughening operator. The model that this implies is that the output should be locally smooth. If I have a smoothing operator, SM, I can define a roughening operator, $R = 1 - SM$. The residual vector that I want to minimize is $R \cdot \phi$.

Many numerical methods can be used to solve this problem. I choose to use a simple conjugate-gradient algorithm (Claerbout, 1985). The smoothing operator I use is the triangular smoothing operator "Triconv" described by Claerbout(1989). The adjustable parameters in the interpolation algorithm are the length of the smoothing operator and the number of iterations of the conjugate gradient algorithm.

An important property of this type of interpolation method is that it can be used to estimate extensions off the ends of a dataset. If I had simply designed a filter from the input data and then used the filter to predict off the end of the data it might have produced a diverging solution.

When processing a seismic dataset I treat each time slice independently. This produces one-dimensional vectors of data with unknown samples can be processed using the method described above. Figure 1 shows the processing of a synthetic dataset. The input (a) has equal amplitude spikes across half the gather. Panels (b), (c) and (d) show the results as more iterations of the conjugate gradient algorithm are performed.

Figures 2 and 3 show the processing of a real dataset. A dataset with half the traces missing at far offsets, wz27 (Yilmaz and Cumro,1983), has been windowed to provide a small test dataset. Normal moveout was applied to the data with water velocity. This has made most events reasonably, but not totally, flat. Figure 2 shows the results of processing with a filter of half width 3 samples. The missing traces at far offsets are well interpolated but as the iterations progress, more energy is put in flat extensions at the ends of the data. This is because the algorithm tries to produce a smooth answer on each time slice separately, and the flat extensions are certainly smooth. Figure 3 shows the results of processing with a filter of half width 7 samples. Because the data has some residual dip it is not smooth over a length of 15 samples. The algorithm only interpolates large amplitudes when it is surrounded by data that is smooth over the length of the filter. Because of the residual dip the results for a longer filter are less successful.

A better approach to the problem might be to avoid any assumptions about the desired output spectrum and instead use an error filter that matches the spectral characteristics of the input data.

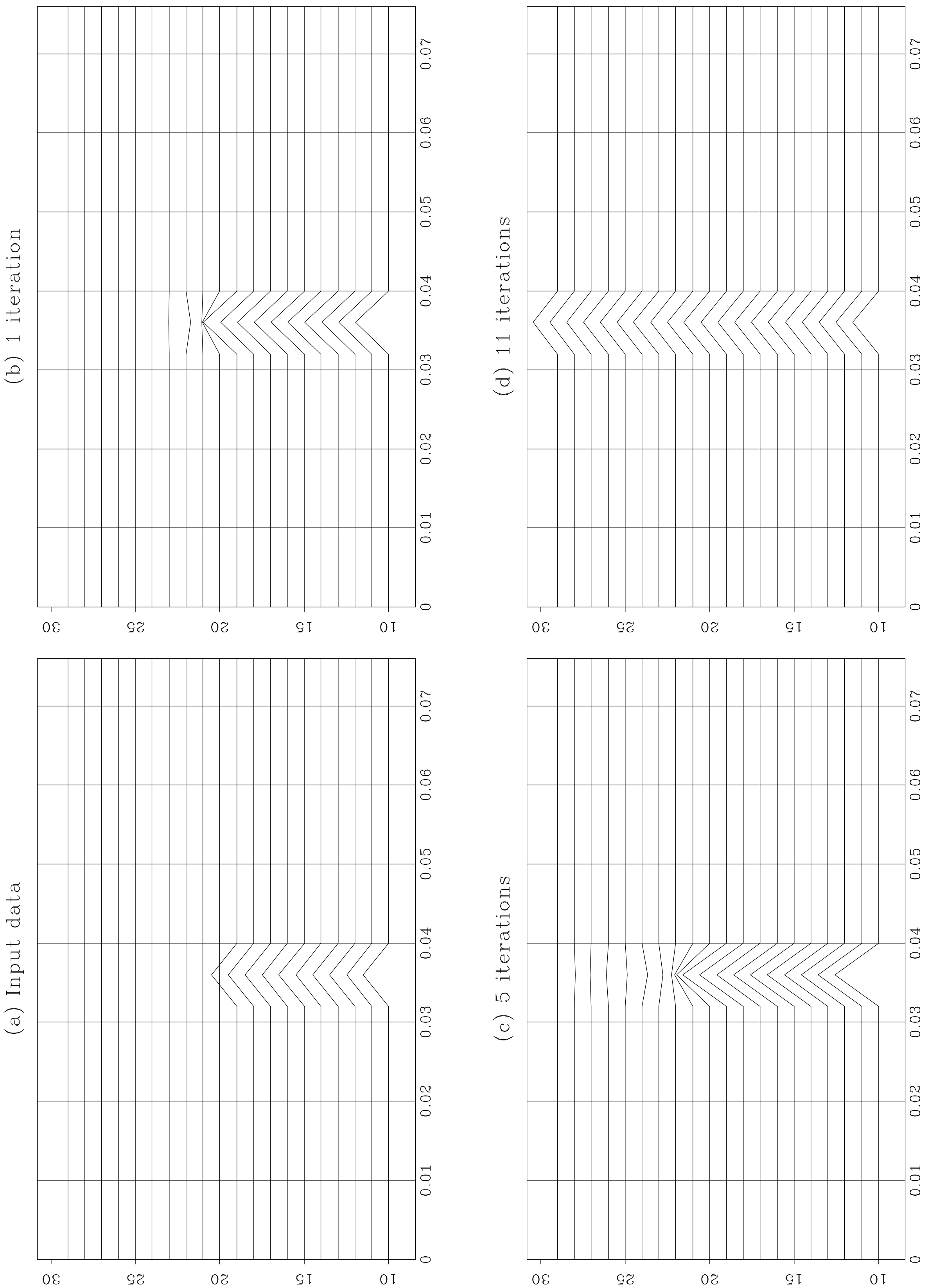

Figure 1: Linear estimation with filter of half width 3. (a) Input data, (b) 1 iteration of conjugate gradient algorithm, (c) 5 iterations, (d) 11 iterations. dave-plotlin1 [NR]

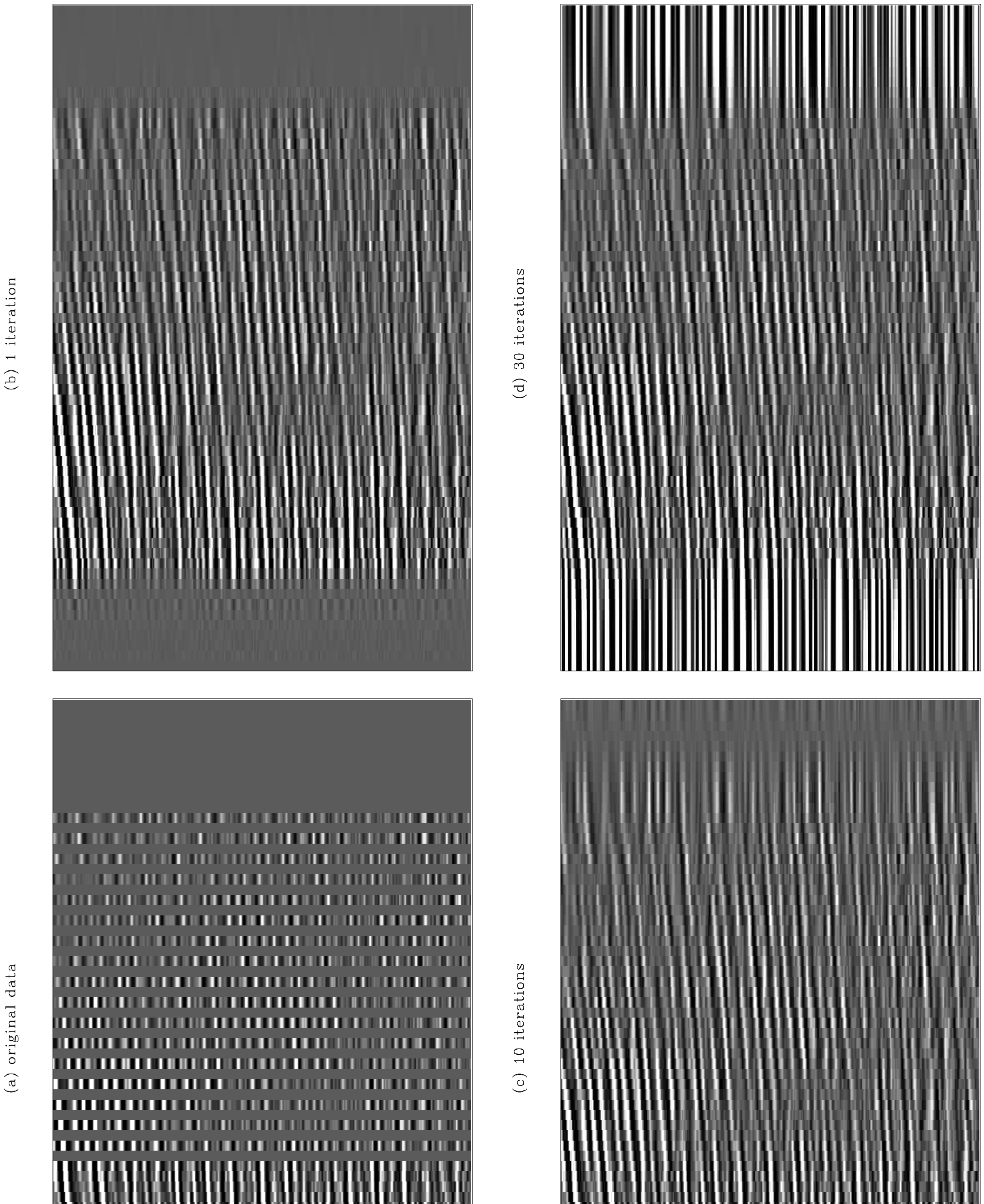

Figure 2: Linear estimation with filter of half width 3. (a) Input data, (b) 1 iteration of conjugate gradient algorithm, (c) 10 iterations, (d) 30 iterations. dave-plotlin2 [NR]

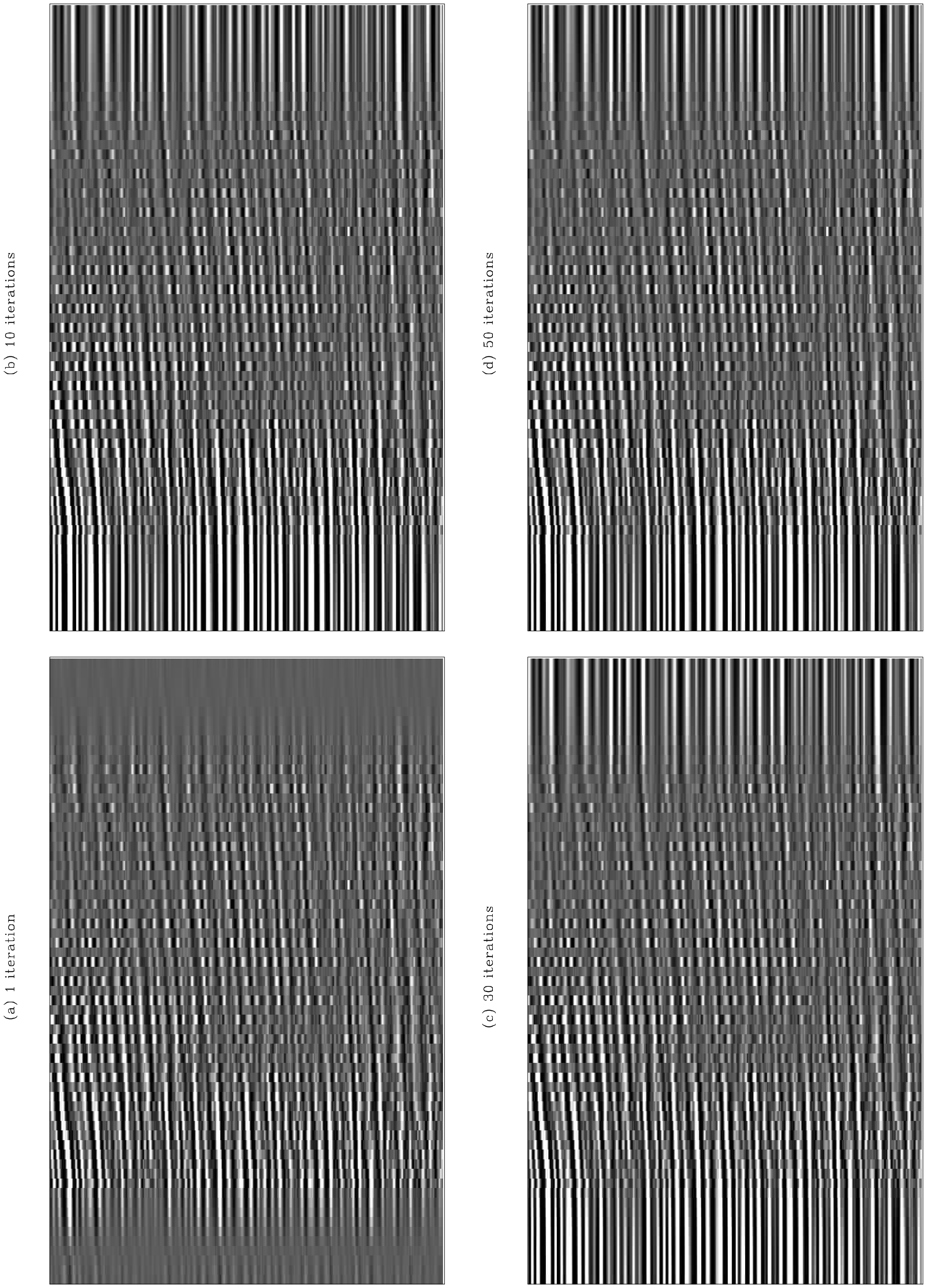

Figure 3: Linear estimation with filter of half width 7. (a) 1 iteration of conjugate gradient algorithm, (b) 10 iterations, (c) 30 iterations, (d) 50 iterations. dave-plotlin3 [NR]

JOINT ESTIMATION OF OPERATOR AND MISSING DATA

Interpolation Error Filters

Claerbout (1989) described the interpolation error filter. This is a filter with a one at zero lag and coefficients either side of zero lag designed such that the power of the output data is minimized.

I.e. we design a filter

$$f = f_{-n}, f_{-(n-1)} \ldots, f_{-1}, 1, f_1, f_2, \ldots, f_n$$

to minimize $|f * d|^2$.

Estimation of missing data

To apply the estimation procedure using interpolation error filters I must minimize

$$|f * \phi|^2 = |f * u + f * k|^2.$$

Since both the filter, f, and the estimated values, u, are model parameters I have a non-linear problem. This problem can be linearized as follows,

$$\begin{aligned} &|(f_i + \delta f) * (u_i + \delta u) + (f_i + \delta f) * k|^2 \\ &\approx |f_i * (u_i + k) + \delta f * (u_i + k) + f_i * \delta u|^2 \\ &= |(u_i + k) * \delta f + f_i * \delta u - (-f_i * (u_i + k))|^2 \end{aligned}$$

At each step of the non-linear algorithm I solve the linear problem for the model parameters δu and δf and then update the estimates of the filter coefficients and missing samples by

$$u_{i+1} = u_i + \delta u \text{ and } f_{i+1} = f_i + \delta f.$$

Figure 4 illustrates a case where the linear estimation does not work, there are dips present on the input data. The local smoothness assumption that I made in deriving the linear interpolation process is violated. When I interpolate using the linear operator the dips are not preserved, as seen in the second panel. The nonlinear operator shown in the third panel preserves the spatial spectrum of the input time slices and therefore preserves the dip in the interpolated region. The fourth panel shows the results for the case where the filter is designed on a 10 row window of the data. This should ensure stability in cases where there is noise present in the input.

Figure 5 shows the results of applying the nonlinear process to real data. The results generally improve as more nonlinear iterations are performed. Some time slices have extensions that appear to have a different character to the neighboring time slices. This is caused by anomalous amplitudes in the input data. If I stabilize the filter design by estimating it on 20 rows at a time I obtain the more uniform results shown in figure 6

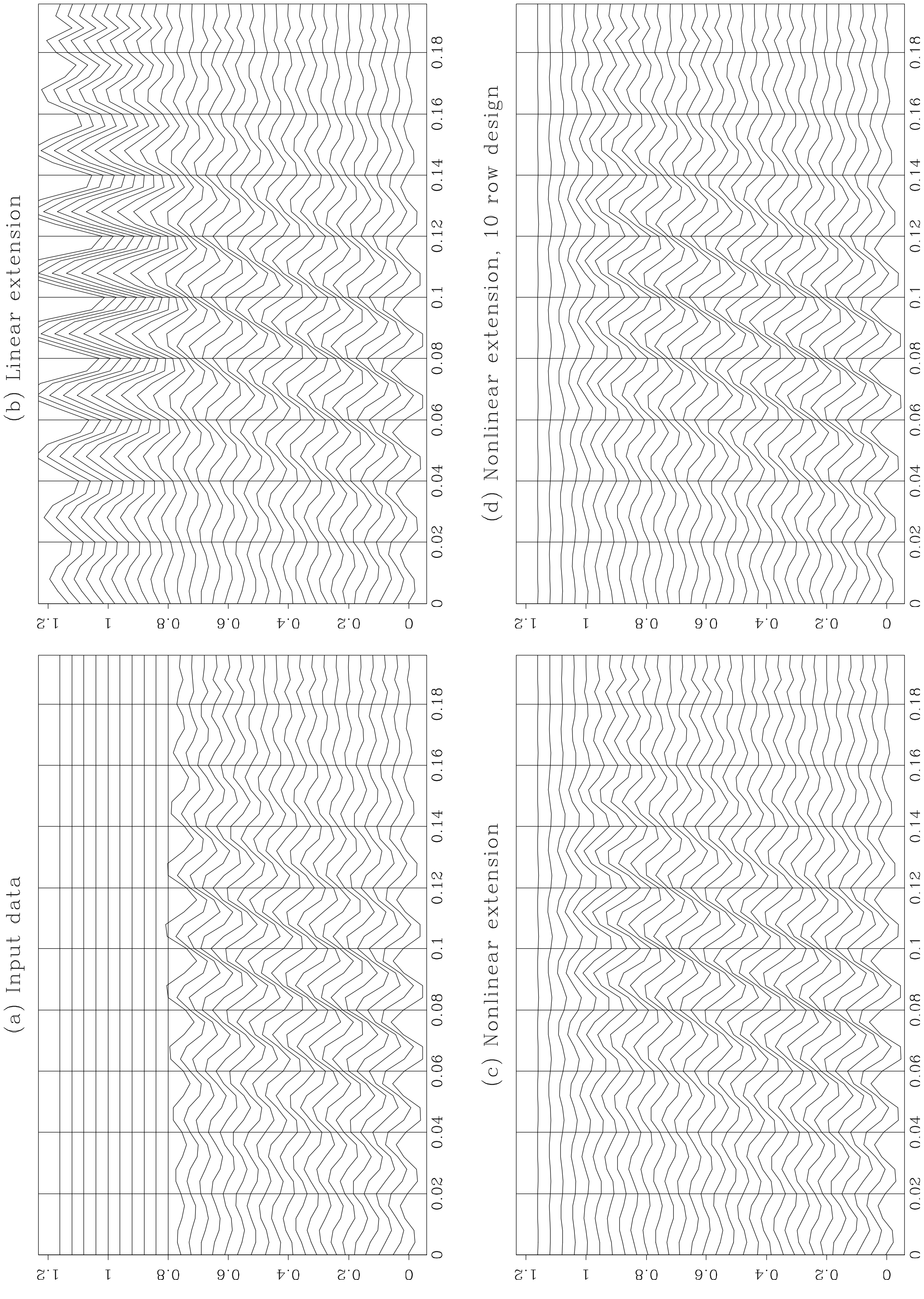

Figure 4: Linear vs. Nonlinear estimation (a) Input data, (b) Linear interpolation, (c) Nonlinear interpolation one row at a time, (d) Nonlinear interpolation, filters designed on a 10 row window. dave-plotnl1 [NR]

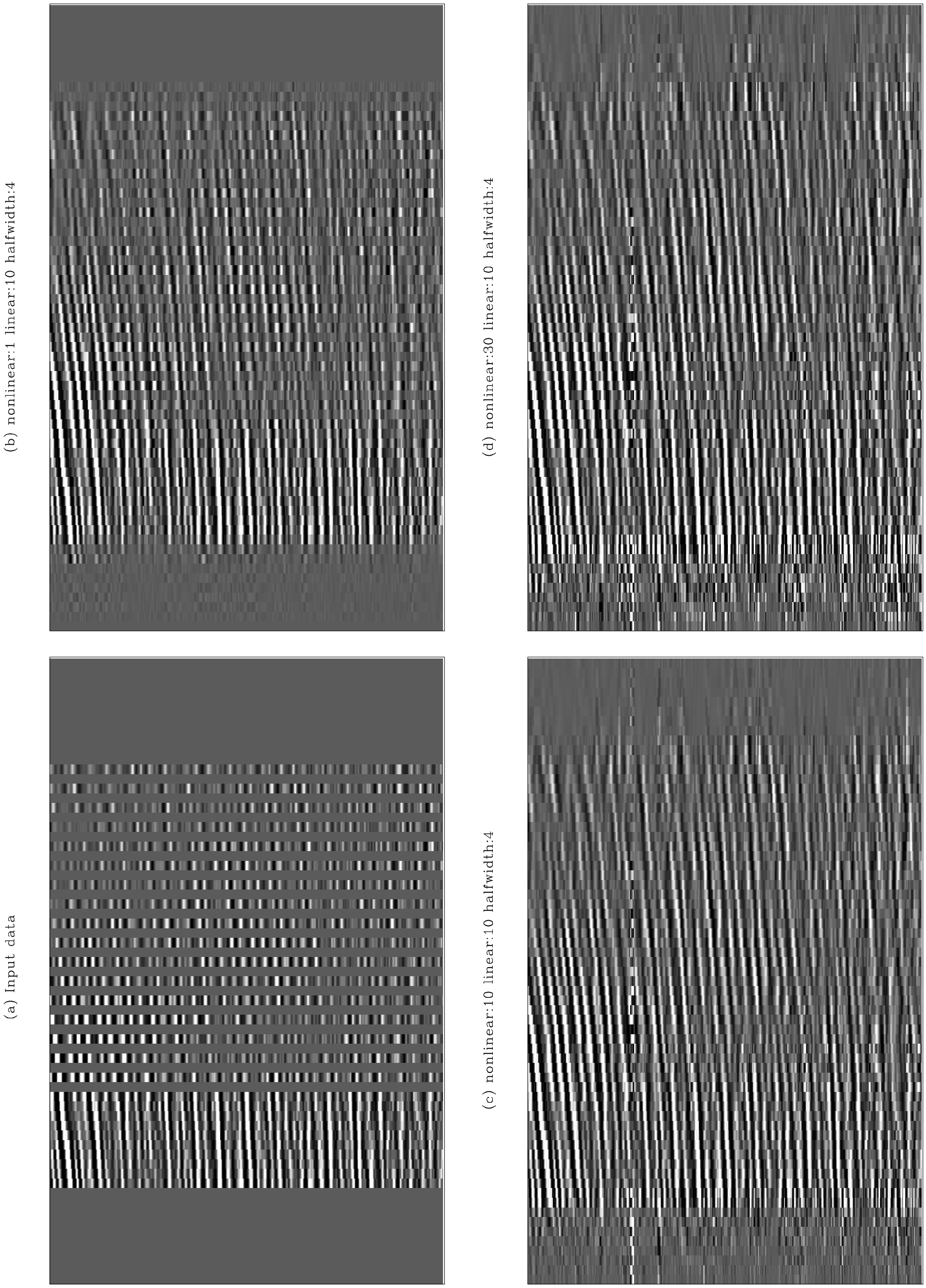

Figure 5: Nonlinear estimation with filter of half width 4. (a) Input data, (b) 1 nonlinear step, (c) 10 nonlinear steps, (d) 30 nonlinear steps. dave-plotnl2 [NR]

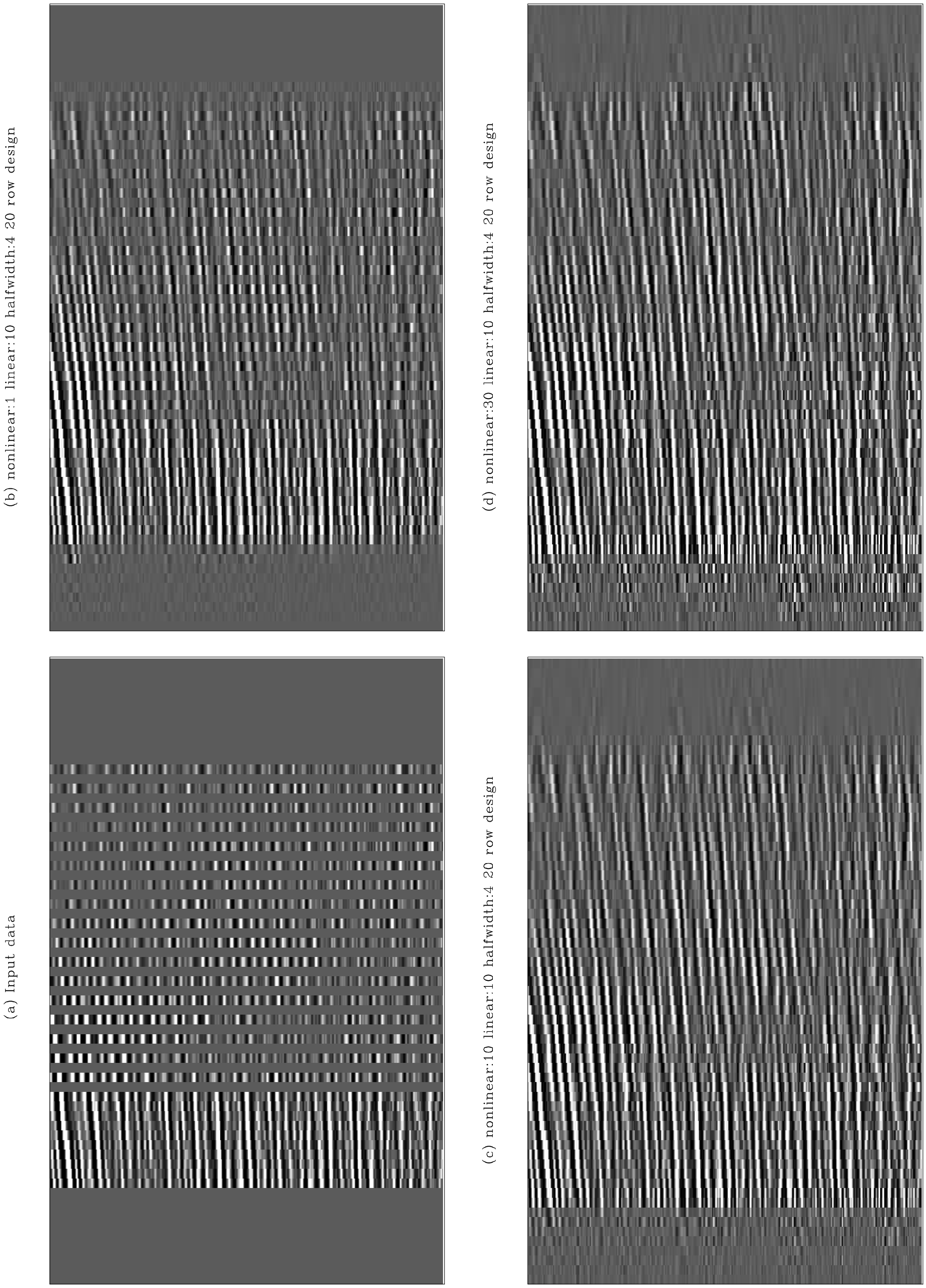

Figure 6: Nonlinear estimation with filter of half width 4, filters designed on 20 row windows. (a) Input data, (b) 1 iteration of conjugate gradient, (c) 10 iterations, (d) 30 iterations. dave-plotnl3 [NR]

TWO DIMENSIONAL INTERPOLATION ERROR FILTERS

One dimensional filters operating on time slices in the $t - x$ domain are implicitly estimating the spatial spectrum of the data. If there are multiple dips present in the data they cannot be separated on the basis of their spatial spectrum alone. There are two possible solutions to this problem, transform the time axis to the frequency domain and work with constant frequency slices. This is the method used in $f - x$ filter design (see for example Canales,1984). The other approach is to design two dimensional filters. I have chosen to use two dimensional interpolation error filters.

The 2-D filters will estimate the spectrum of the data in both time and space. The 1-D filters that I have described up to now are a special case of the 2-D filter. The notation used earlier may be preserved except now the vectors k and u of known and unknown data become 2-D arrays, the filter f becomes a 2-D operator and $*$ denotes 2-D convolution. The model parameters are now the unknown data points within some 2-D window and the coefficients of a 2-D interpolation error filter.

Figure 7 illustrates a case where we might expect the 1-D algorithm to fail, the input dataset has two conflicting dips, one flat and the other dipping at 45°. The linear 1-D process extends only the horizontal energy. The nonlinear 1-D process does not successfully extend both dips.

If a 2-D algorithm is applied with a filter half length of 4 samples in both x and t the output, shown in figure 8, is not very different to the output of the nonlinear 1-D algorithm. One possibility is that more iterations might be needed to make the problem converge. However the cost of the 2-D algorithm is already much more than the 1-D algorithm so I decided to try another approach.

THE “WAVEKILL” FILTER

Jon Claerbout suggested that I should use a filter that had the property that it would destroy plane waves of a given dip. Given such an operator I could perform a joint estimation of the local dip and the missing data values. He suggested a form for such a filter, which he calls the “plane wave destruction” filter or the “wavekill” operator.

I can reject a dip, p, with the operator,

$$\mathbf{A} \equiv (\partial_x - p\partial_t).$$

An approximation to this operator can be implemented on a 2×2 differencing star. The best estimate of the dip of an array of data can be found by minimizing

$$|A \cdot \phi|^2$$

If $\mathbf{x}$ is the array $\partial\phi/\partial x$ and $\mathbf{t}$ is the array $\partial\phi/\partial t$ then the minimum value of $|\mathbf{x} - p\mathbf{t}|^2$ is at $p = \mathbf{x} \cdot \mathbf{t}/\mathbf{t} \cdot \mathbf{t}$.

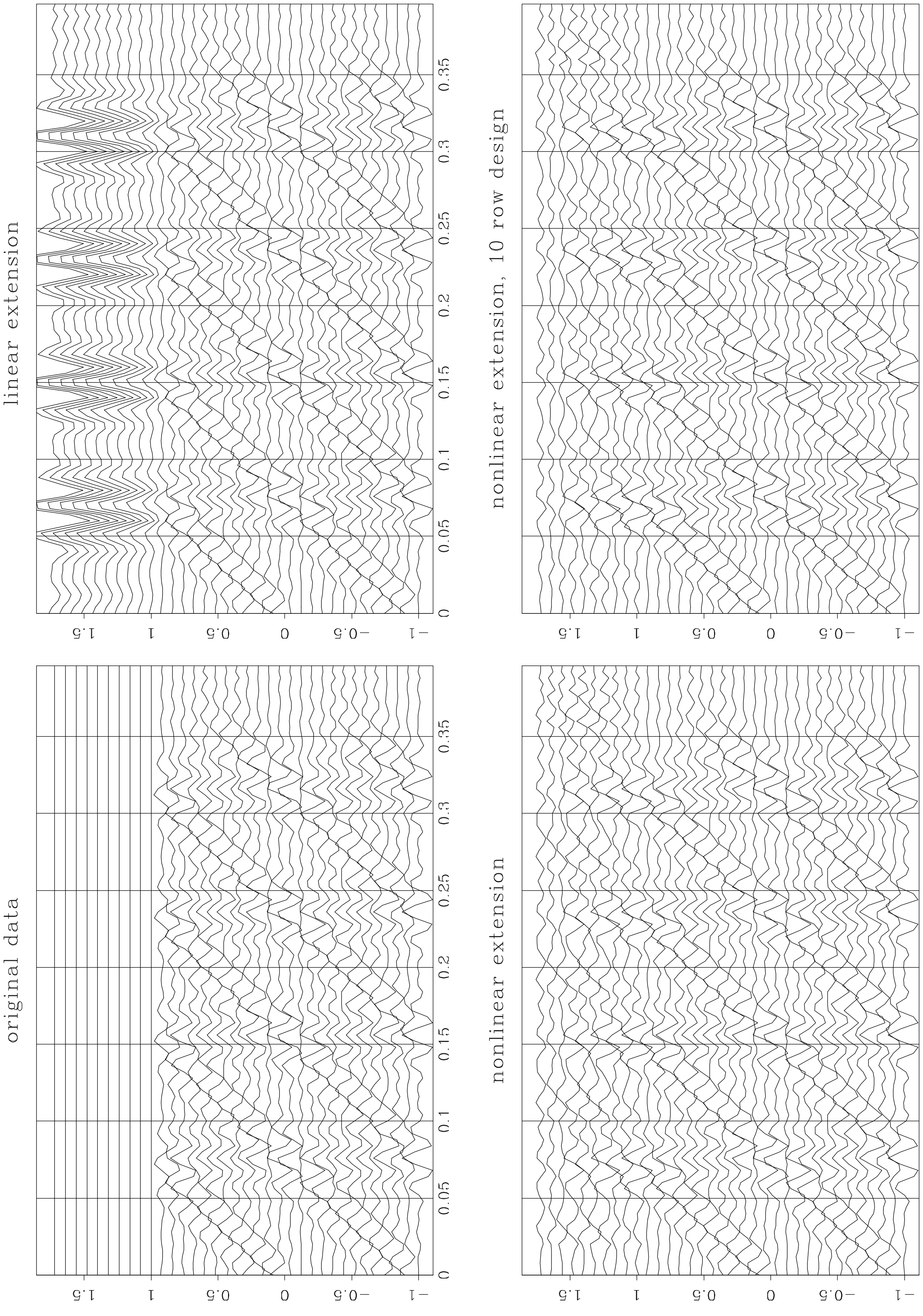

Figure 7: 1-D processing of multi-dip synthetic, (a) Input data, (b) linear estimation, (c) nonlinear 1-D estimation (d) nonlinear 1-D estimation, filters designed on 10 row windows. dave-plot2d1 [NR]

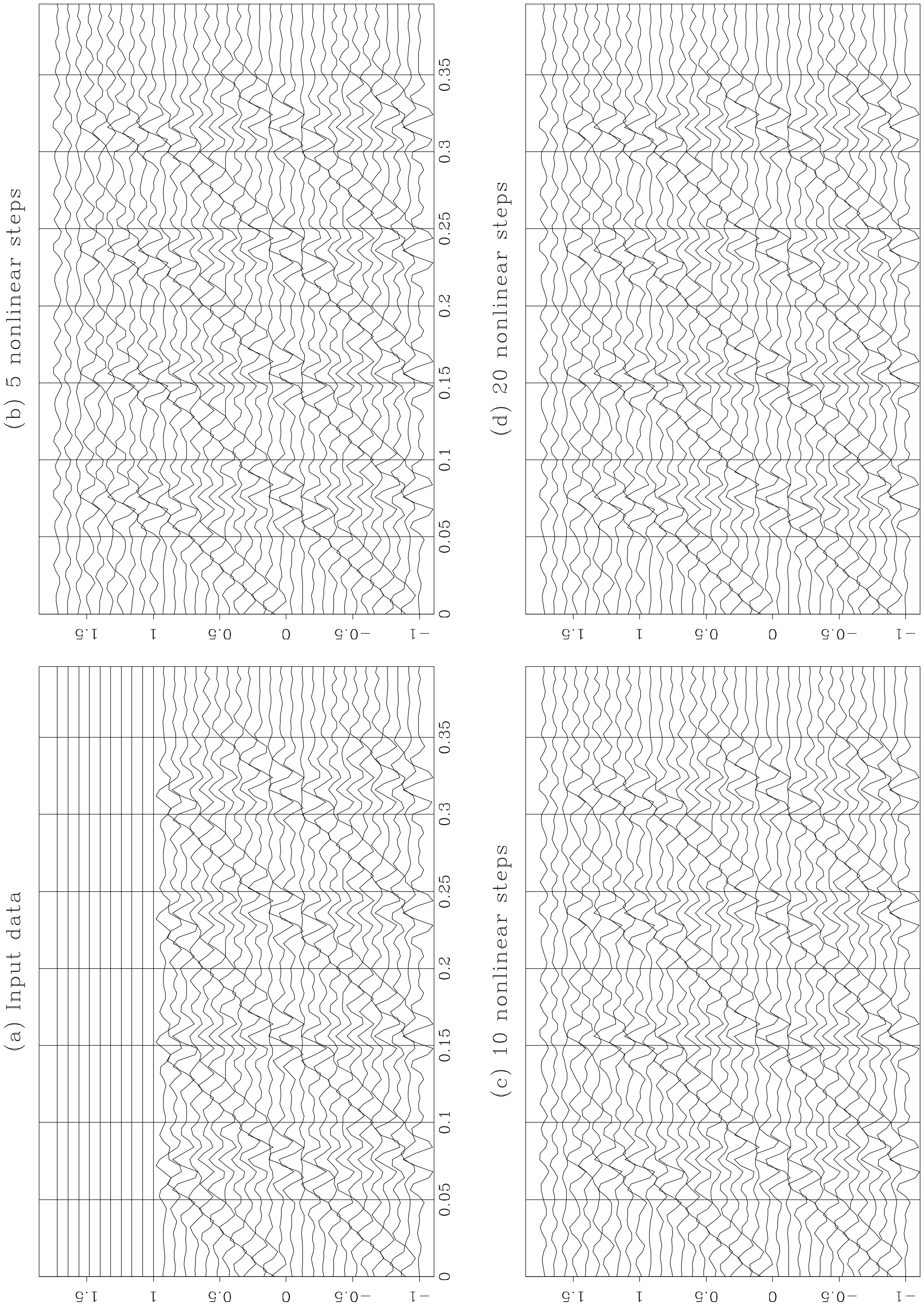

Figure 8: 2-D processing of multi-dip synthetic. (a) Input data, (b) 5 nonlinear steps, (c) 10 nonlinear steps, (d) 20 nonlinear steps. dave-plot2d2 [NR]

To use this operator for interpolation when the dip is known I minimize the residual,

$$|A \cdot u + A \cdot k|^2.$$

This is a linear least squares problem of the type described earlier. If the dip is not known I have to solve a non-linear problem for both the dip and the missing data values.

Multiple dips

In most seismic data there are crossing dips present. In order to interpolate this type of data accurately I need an operator that will reject more than one dip. An obvious form for this operator is a cascade of single dip rejection operators. If I wish to reject three dips, p_1, p_2 and p_3, I need an operator of the form,

$$\mathbf{A}_3\mathbf{A}_2\mathbf{A}_1 \equiv \left(\frac{\partial}{\partial x} - p_3\frac{\partial}{\partial t}\right)\left(\frac{\partial}{\partial x} - p_2\frac{\partial}{\partial t}\right)\left(\frac{\partial}{\partial x} - p_1\frac{\partial}{\partial t}\right).$$

The estimate of the three dominant dips present in the data is found by minimizing the residual,

$$|\mathbf{v}|^2 = |\mathbf{A}_3\mathbf{A}_2\mathbf{A}_1\phi|^2.$$

Because there are terms involving $p_1 p_2$ etc. this is a nonlinear problem. When linearized for small perturbations in the dip parameters we obtain the following expressions,

$$\Delta v = \sum_{i=1}^{3} \frac{\partial v}{\partial p_i} \Delta p$$

In operator form this is

$$\Delta\mathbf{v} \;=\; [(\mathbf{A}_3\mathbf{A}_2\delta_t\phi)(\mathbf{A}_3\delta_t\mathbf{A}_1\phi)(\delta_t\mathbf{A}_2\mathbf{A}_1\phi)] \begin{bmatrix} \Delta p_1 \\ \Delta p_2 \\ \Delta p_3 \end{bmatrix}$$

This expression and the conjugate relationship,

$$\begin{bmatrix} \Delta p_1 \\ \Delta p_2 \\ \Delta p_3 \end{bmatrix} = \sum_{t,x} \Delta\mathbf{v}_p \begin{bmatrix} (\mathbf{A}_3\mathbf{A}_2\delta_t\phi) \\ (\mathbf{A}_3\delta_t\mathbf{A}_1\phi) \\ (\delta_t\mathbf{A}_2\mathbf{A}_1\phi) \end{bmatrix}$$

can be used to solve the minimization problem by the method of conjugate gradients.

If both the local dip and the missing data values are unknown a non-linear estimation of both the missing data and the local dips can be performed. Figure 9 illustrates the results of applying this process to a simple synthetic with a single dip. In this synthetic every other trace is unknown so the problem is the standard data resampling problem. Panel a) shows the input data and panel b) shows the correct answer, panel c) shows the interpolated output and panel d) shows the difference between the correct answer and the interpolated data. The interpolation has not worked well, the interpolated points are not the same amplitude as the input points, why is this ?

Some insight can be obtained by looking at the data in the $\omega - k_x$ domain. Figure 10 shows four panels transformed into the $\omega - k_x$ domain. Panel a) is the correct answer, panel b) shows the spectrum a dataset that contains every other trace of the correct answer i.e. the known data for the problem. This plot shows that the data is aliased. Panel c) is the spectrum of the known data with zero traces inserted, this is the spectrum of panel b) replicated. This is the initial data that the algorithm uses. The final panel, d), is the spectrum of the interpolated data. There are clearly some artifacts present that are remnants of the aliased, replicated spectrum shown in panel c). When the algorithm chooses the best values of p it is doing so based on aliased data. Because the problem is nonlinear we are not guaranteed to obtain the global solution. Instead the algorithm will find the minimum nearest to the initial solution. In this example the initial solution is that the missing data is zero and the p-values are -1,0,1. The comparison of the desired output spectrum and the spectrum of our initial data proves that this is a poor initial guess.

However, this problem is not insurmountable, there is a way to "bootstrap" the solution to find a better initial guess. Aliasing is usually only a problem at large ω and k_x, if I estimate initial values for p using only the unaliased part of the spectrum they should be more reliable than those estimated from the aliased data. I can obtain an unaliased version of the spectrum by smoothing the data in space and time. The smoothing in space and time removes the energy in the spectrum at large ω and k_x. The fundamental assumption made in using this process is that the dips in the data at low ω and k_x are the same as the dips present in the correct interpolated output.

I now use a three stage process in interpolating the data:

1. Smooth the data and estimate initial values for the dips.

2. Perform a linear estimation of the missing data with fixed dips.

3. Perform a non-linear, joint estimation of the missing data and dips.

Figure 11 shows the results of applying this three stage process to the single dip synthetic. The interpolated points are much closer to the correct amplitude. Figure 12 shows four spectra. The first is the correct output spectrum, the second is the spectrum of the input data. The third panel shows the spectrum of the smoothed data that is used for the initial estimate of the dips. The last panel shows the spectrum of the interpolated data. The use of better initial values for the dips and the missing data has produced a clearly superior result.

A more complicated synthetic is shown in figures 13 and 14. This is a similar synthetic to the one shown by Spitz(1989). The technique he used is related to the one

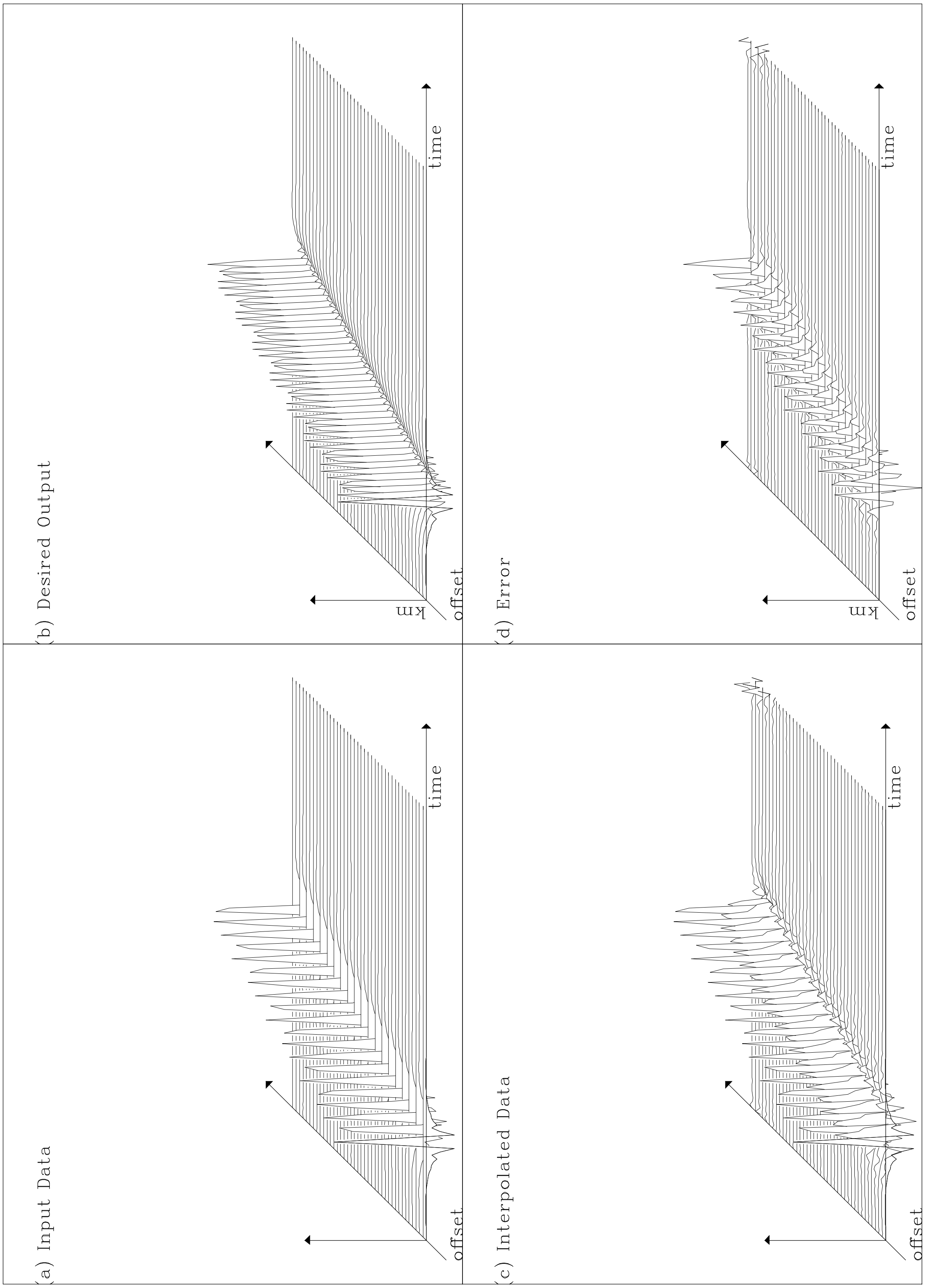

Figure 9: Wavekill processing of single-dip synthetic. (a) Input data, (b) desired output, (c) interpolated data, (d) difference of (b) and (c). dave-altmod1 [NR]

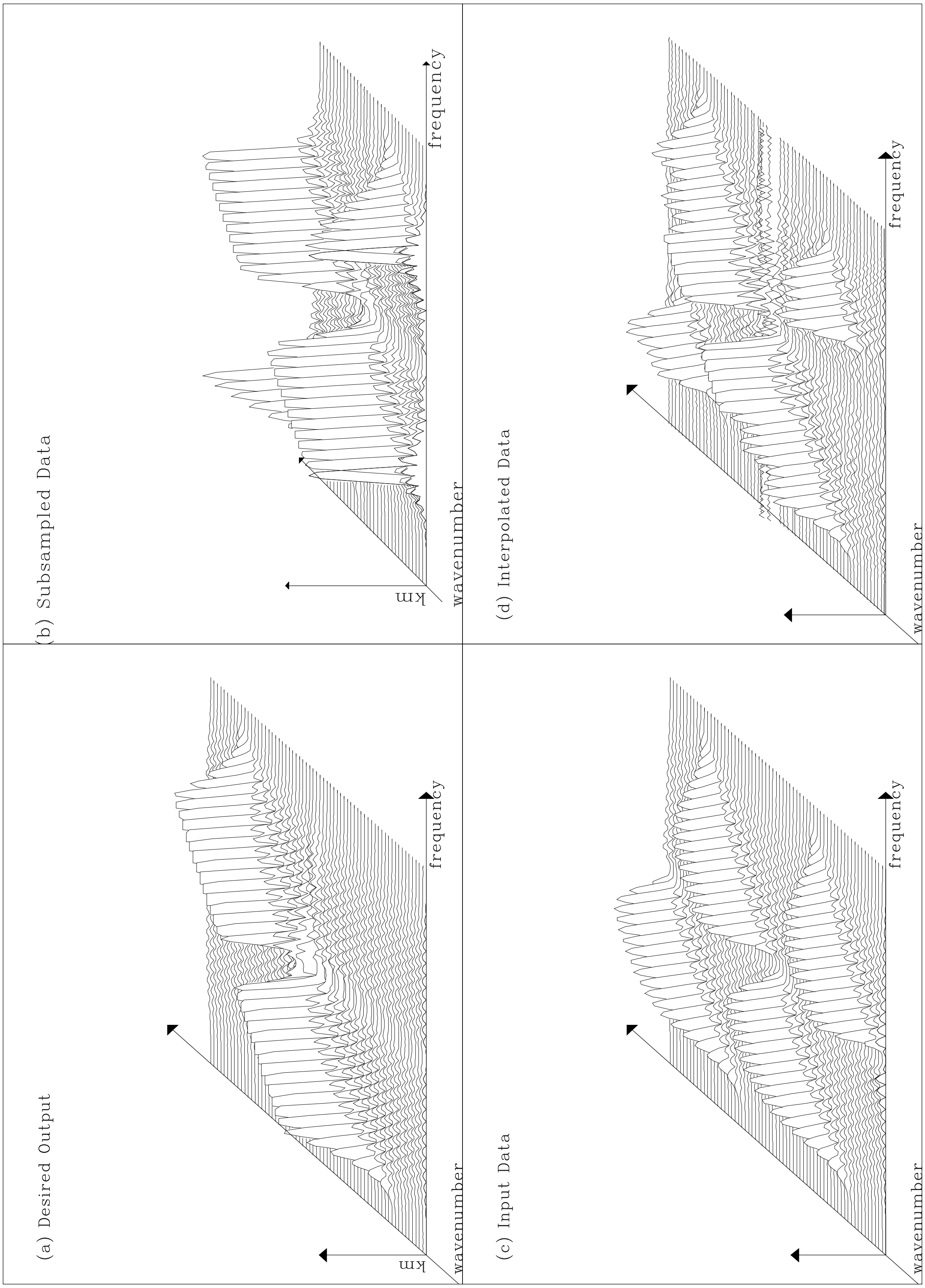

Figure 10: $\omega - k_x$ plots of wavekill processing of single-dip synthetic. (a) Desired output, (b) spectrum of subsampled data , (c) spectrum of input data, (d) spectrum of interpolated data. dave-altmodfk1 [NR]

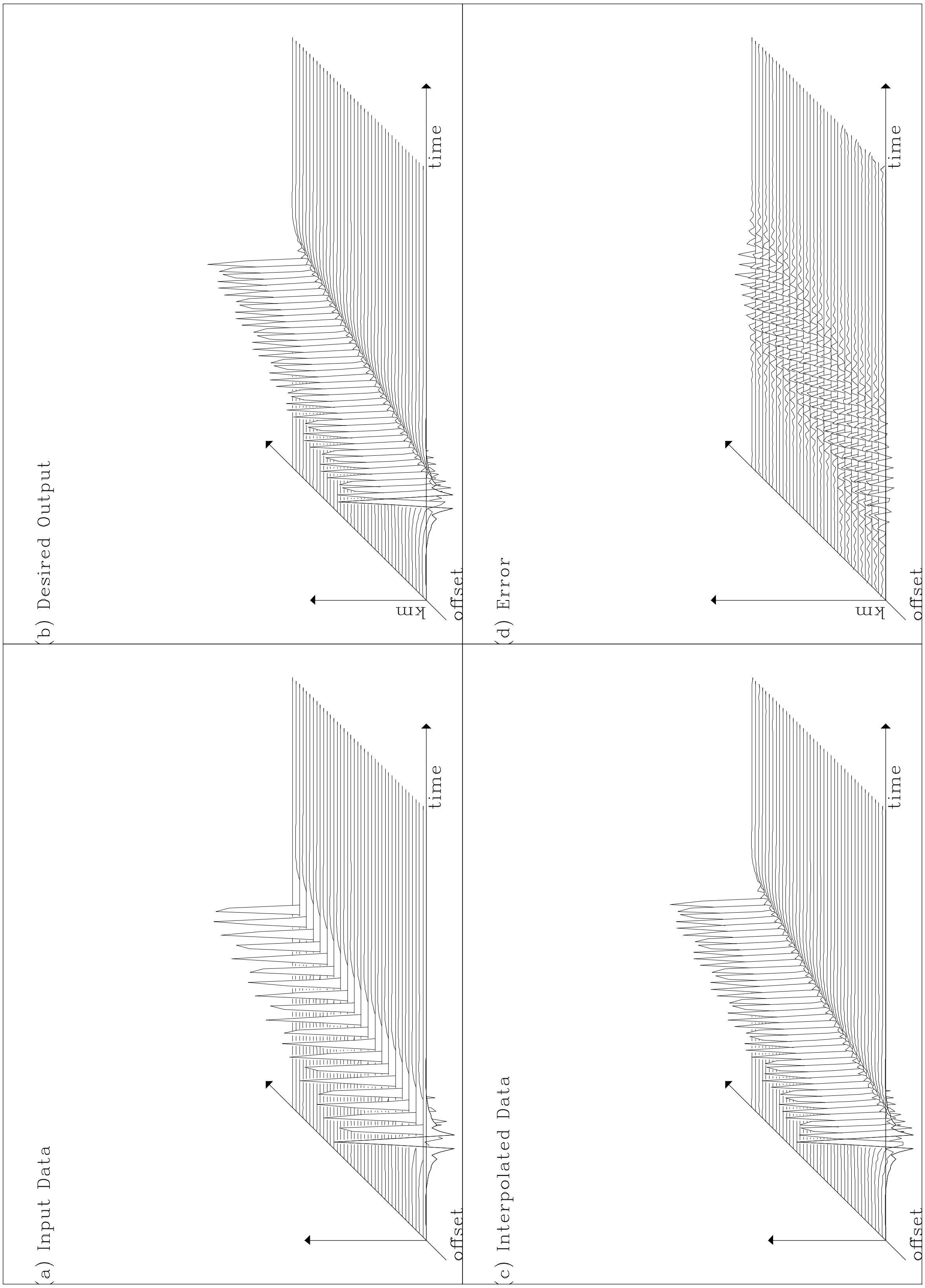

Figure 11: Bootstrapped wavekill processing of single-dip synthetic. (a) Input data, (b) desired output, (c) interpolated data, (d) difference of (b) and (c). dave-altmod2 [NR]

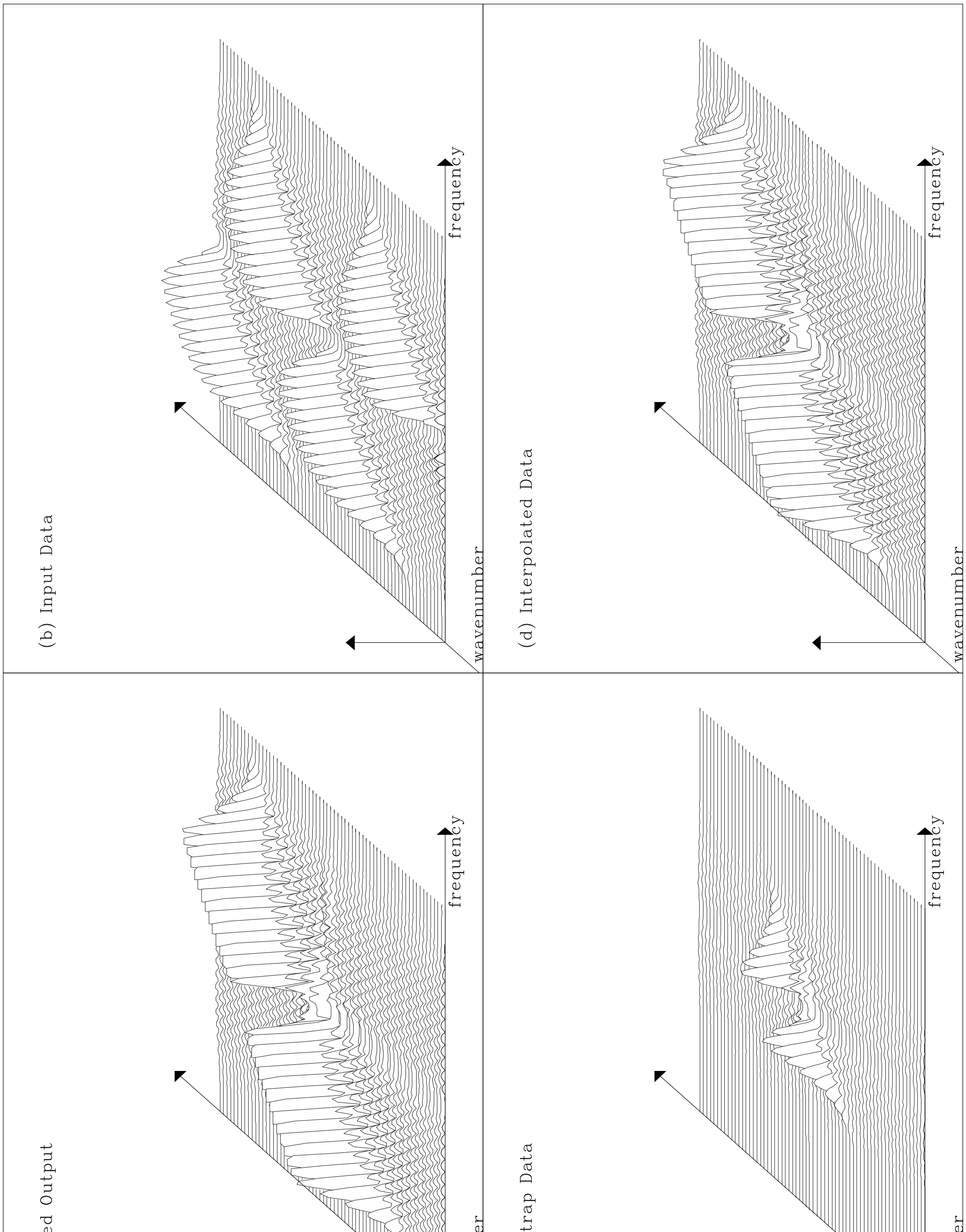

Figure 12: $\omega - k_x$ plots of wavekill processing of single-dip synthetic. (a) Desired output, (b) Spectrum of input data, (c) Spectrum of smoothed data, (d) Spectrum of interpolated data. dave-altmodfk2 [NR]

I have presented. He builds an operator by finding the polynomial representation of a spatial prediction filter in the frequency domain. Once he has found the polynomial he solves for the missing data values. One advantage of the method I have presented is that it is a time domain method. This means that local operators may be constructed by estimating dips within small windows of the data. It is more difficult to construct local operators when working in the frequency domain.

The interpolation of this synthetic is not totally successful as there is still some residual "jitter" on the steepest dip. The spectrum of the interpolated data appears to be unaliased but not as extended is $\omega - k_x$ space as the correct answer. It is possible that more iterations are needed to find the correct solution. Another possible solution might be to restart the algorithm and bootstrap it from a smoothed version of the interpolated data.

Figure 15 illustrates the application of this four stage process to a real dataset, wz08 (Yilmaz and Cumro, 1983). The dataset has conflicting dips present and a large gap near zero offset. Although the interpolation is unsatisfactory in the gap at zero offset, the rest of the data is well interpolated. The steeply dipping linear events have been correctly interpolated despite the fact that they are aliased at some frequencies.

CONCLUSIONS

The missing data estimation methods that I have presented are conceptually simple and easily extendible to arbitrary dimensionality. I have shown the importance of careful choice of initial values when a non-linear method is used. By making the assumption that the dips at aliased frequencies are the same as the dips at unaliased frequencies I have shown that we can make good choices of initial values that allow interpolation of aliased data.

REFERENCES

Claerbout, J., 1985, Conjugate gradients for beginners: SEP–**44**, 161-166.

Claerbout, J., 1989, Time series analysis: SEP–**61**, 391-414.

Canales, L.L., 1984, Random noise reduction : SEG Expanded Abstracts, 525-527.

Spitz, S., 1989, Trace interpolation beyond aliasing : SEG Expanded Abstracts, 1121-1122.

Yilmaz, O., and Cumro, D., 1983, Worldwide Assortment of Field Seismic Records, released by Western Geophysical Company of America, Houston.

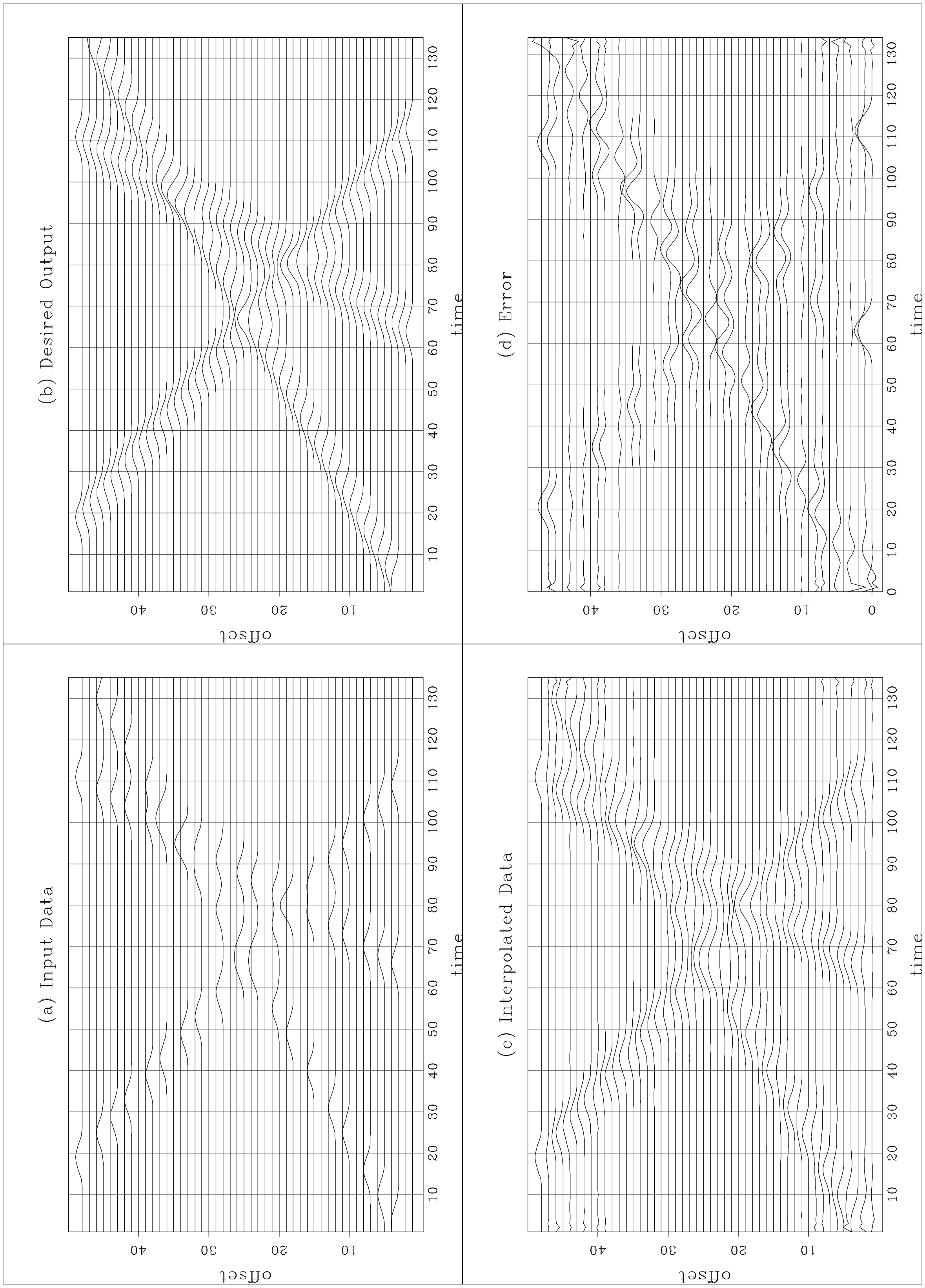

Figure 13: Bootstrapped wavekill processing of spitz synthetic. (a) Input data, (b) desired output, (c) interpolated data, (d) difference of (b) and (c). dave-spitz [NR]

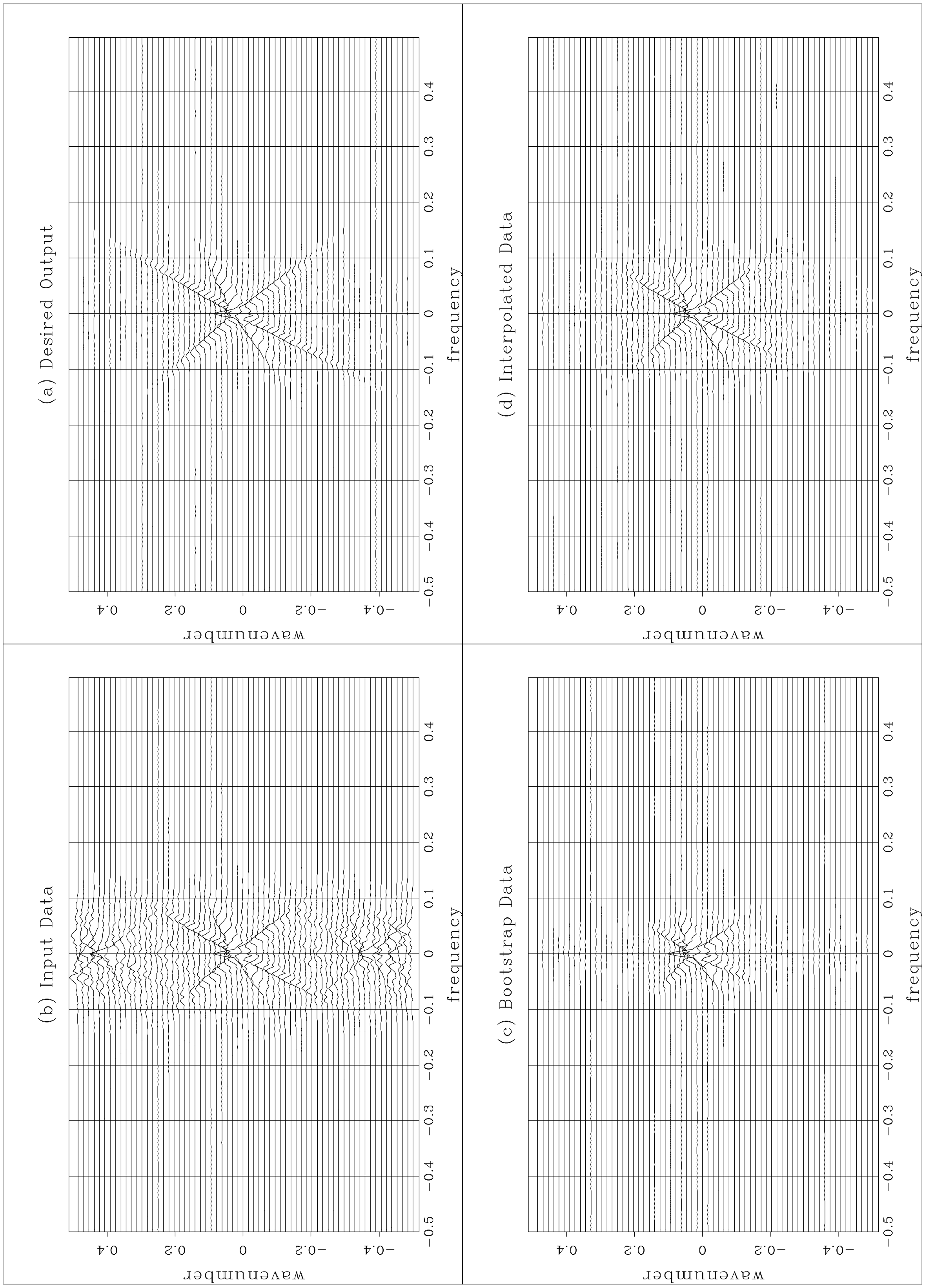

Figure 14: $\omega - k_x$ plots of wavekill processing of spitz synthetic. (a) Desired output, (b) Spectrum of input data, (c) Spectrum of smoothed data, (d) Spectrum of interpolated data. dave-spitzfk [NR]

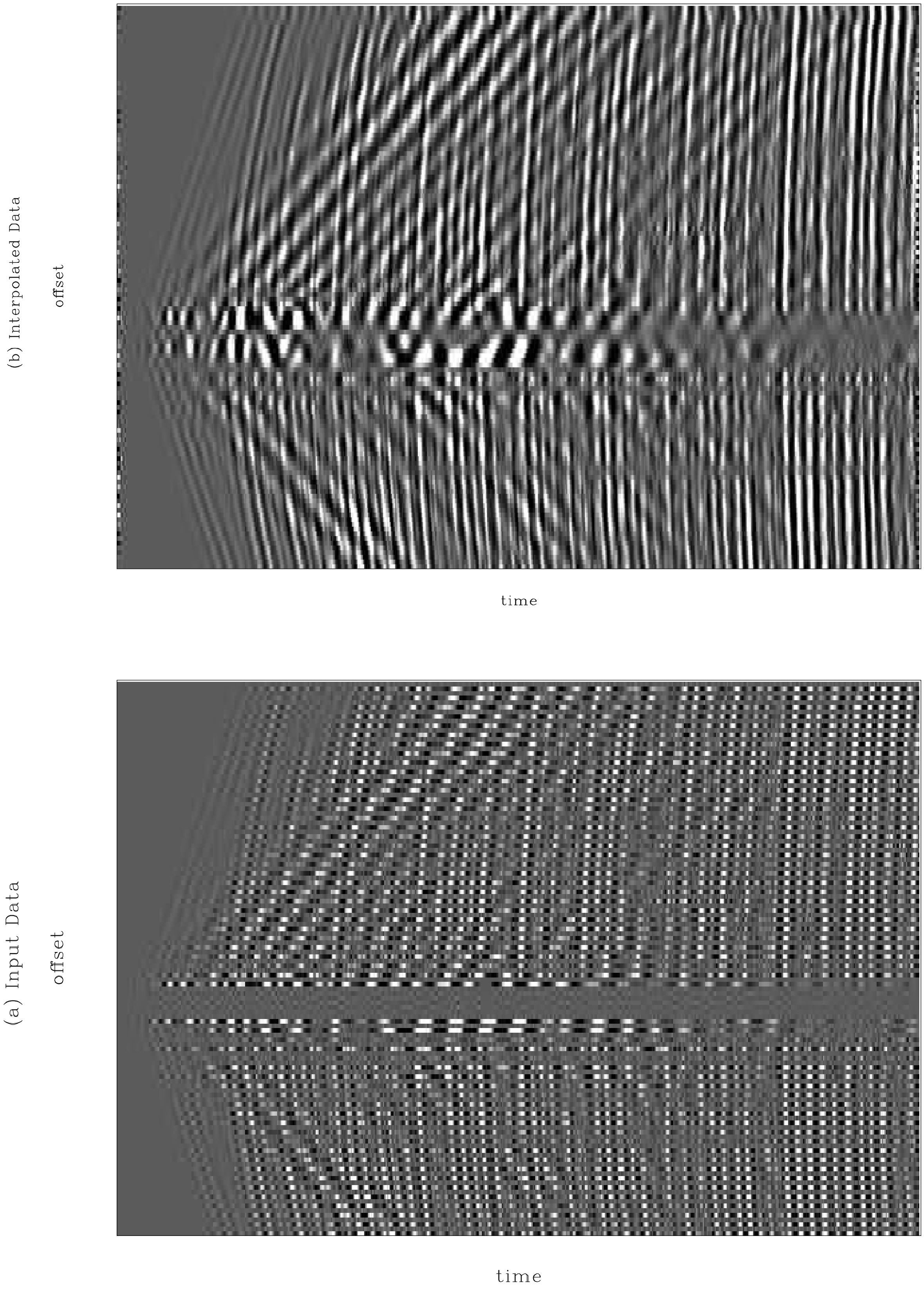

Figure 15: Bootstrapped wavekill processing of real data (a) Input data, (b) Interpolated output. dave-realpext [NR]