GEOPHYSICS, VOL. 68, NO. 1 (JANUARY-FEBRUARY 2003); P. 355–369, 20 FIGS.
10.1190/1.1543221

Seismic trace interpolation in the Fourier transform domain

Necati Gülünay*

ABSTRACT

A data adaptive interpolation method is designed and applied in the Fourier transform domain (f-k or f-k_x-k_y) for spatially aliased data. The method makes use of fast Fourier transforms and their cyclic properties, thereby offering a significant cost advantage over other techniques that interpolate aliased data.

The algorithm designs and applies interpolation operators in the f-k (or f-k_x-k_y) domain to fill zero traces inserted in the data in the t-x (or t-x-y) domain at locations where interpolated traces are needed. The interpolation operator is designed by manipulating the lower frequency components of the stretched transforms of the original data. This operator is derived assuming that it is the same operator that fills periodically zeroed traces of the original data but at the lower frequencies, and corresponds to the f-k (or f-k_x-k_y) domain version of the well-known f-x (or f-x-y) domain trace interpolators.

The method is applicable to 2D and 3D data recorded sparsely in a horizontal plane. The most common prestack applications of the algorithm are common-midpoint domain shot interpolation, source-receiver domain shot interpolation, and cable interpolation.

INTRODUCTION

The recorded wavefield is a sampled version of the continuous wavefield, with the spatial sampling provided by the field geometry. Ideally, source and receiver intervals should be chosen so that reconstruction of the continuous wavefield from the recorded samples is possible. The data are then said to be "properly sampled" (Vermeer, 1990). The resulting source and receiver intervals are called "basic sampling intervals."

Often economical reasons dictate that data be recorded with much larger sampling intervals than basic sampling intervals. This can cause harmful effects in prestack and poststack multichannel data processing. Even when the signal (primary energy) is properly sampled, organized noise, such as ground roll and multiples, is often undersampled, especially in areas where the noise has a much lower velocity than the signal. Although normal moveout (NMO) correction is sometimes used to reduce aliasing for some recording geometries (such as offend marine shooting), an inherent aliasing problem remains in the common offset and common midpoint (CMP) domains, even when source and receiver domains are properly sampled (Vermeer, 1990). Shot intervals that are larger than receiver intervals compound the aliasing problem. Trace interpolation during data processing is a well-known solution to this sampling deficiency.

Trace interpolation has received much attention in the geophysical community during the last two decades. Interpolation of correctly sampled data is almost trivial, yet it is not so in the presence of aliasing. Methods of trace interpolation include sinc interpolation (Jakubowicz, 1994, 1997) after alias reduction via NMO, most coherent dip interpolation (Larner et al., 1981), semblance weighted slant-stack interpolation (Lu, 1985), interpolation using event attributes (King et al., 1984), power diversity slant-stack interpolation (Monk et al., 1993), Radon domain trace interpolation for irregularly sampled or missing data (Kostov, 1989; Darche, 1990; Kabir and Verschuur, 1992; Schonewille and Duijndam, 1996), f-x prediction filter interpolation (Spitz, 1989, 1991; Ji, 1993; Manin and Spitz, 1995; Porsani, 1999), f-x projection filter interpolation (Soubaras, 1997), t-x domain prediction error filter (PEF) interpolation (Claerbout and Nichols, 1991; Claerbout, 1992), and others combining different domains, such as f-x domain wavefield decomposition using a picked dip field (Pieprzak and McClean, 1988, 1990). Trace interpolation methods that make use of the frequency-wavenumber (f-k) domain have also emerged (Pan and Fields, 1986; Guo et al., 1996; Gülünay and Chambers, 1996). This domain is appealing because of the availability of fast Fourier transforms (FFT) and the fact that the costly solution of linear equations that appears in the f-x interpolation for the interpolated data points turns into complex number divisions in the f-k domain. Using this approach, Gülünay and Chambers (1997b, c) developed a "generalized f-k trace interpolation method" that can interpolate 2D as well as 3D data by an arbitrary integer factor, L, in both x and y directions. That method can be viewed as a data adaptive t-x (or t-x-y) filtering method where zero traces are inserted at the

Published on Geophysics Online July 19, 2002. Manuscript received by the Editor May 29, 2001; revised manuscript received June 20, 2002.
*Formerly WesternGeco R&D, 10001 Richmond Ave., Houston, Texas 77042-4299; presently 2926 Calendar Lake Drive, Missouri City, Texas 77459. E-mail: n.gulunay@ieee.org.

locations where the interpolated traces are desired. The operator is designed and applied in the frequency-wavenumber domain (f-k or f-k_x-k_y). In this paper, I present this generalized f-k trace interpolation method with some emphasis on its mathematical properties, and show its application to model as well as field data examples. The Fourier transform and its properties used throughout the paper are well documented by Bracewell (1978) and Burrus (1985).

GENERALIZED f-k TRACE INTERPOLATION FOR ALIASED DATA

The basic idea of generalized f-k trace interpolation (GFKI) is that the Fourier transform samples along the wavenumber direction for interpolated data can be obtained in the transform domain by a point-by-point multiplication of a replicated version of the original transform with an interpolation operator. The method is the extension of unaliased f-k interpolation (UFKI), published by Gülünay and Chambers (1996, 1997a). GFKI is "general" in the sense that any arbitrary but integer interpolation factor, L, can be used, and the method is applicable to 3D data. A similar method to GFKI was used by Pan and Fields (1986), who filtered zero-trace inserted data (which produces a replicated spectrum along the wavenumber axis) with user defined dip filters. For complex data, it is desirable to have a data adaptive filter design scheme as in the f-x interpolation method (Spitz, 1991). The operators of GFKI are similar to those used in f-x projection filtering interpolation (Soubaras, 1997). The interpolation operators used in GFKI are obtained from the original data at lower frequencies in a similar way to the f-x domain trace interpolators. Note that the copy of the f-k transform is done along the wavenumber axis. This interpolator is equivalent to a filter to be applied to the zero-trace-inserted data in the time-space domain. Assuming that the input data are made of linear events, the interpolation filter at a given temporal frequency which will generate a space grid that is L times denser than the original space grid is obtained from a temporal frequency that is L times lower than the original temporal frequency.

Description of generalized f-k interpolator

Let us use a simple 2D example to illustrate the method. Figures 1–3 describe the time-space (t-x) and frequency-wavenumber (f-k) representation of the operator design process. They show what happens to the 2D Fourier transform when one pads the data with zeroes, or inserts zeroes periodically, or when one masks zero-padded data with zeroes

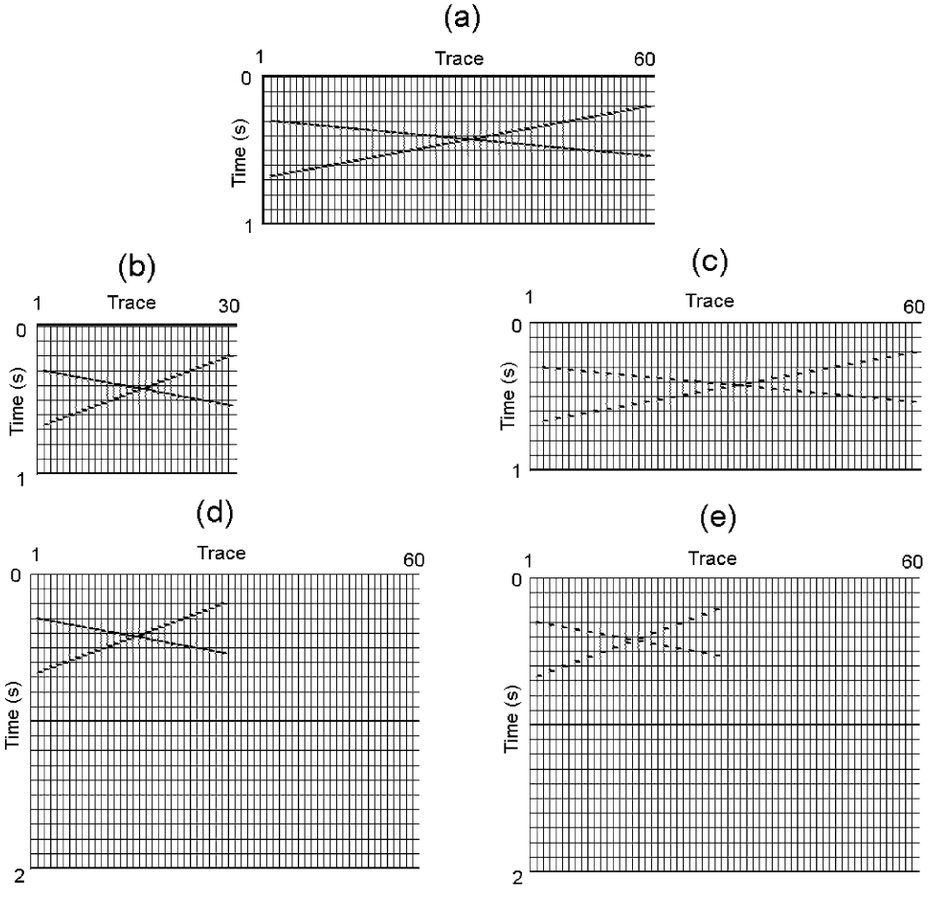

FIG. 1. The t-x domain representation of the 2D GFKI method: (a) desired gather, (b) known gather, (c) zero trace insertion to the known gather, (d) zero padded gather, (e) periodic zeroing of every other trace of the zero padded gather.

periodically. Here, the interpolation factor is assumed to be two ($L=2$) for simplicity.

Figure 1a has 60 traces with two linear events. One of the events dips with 4 ms/trace and the other dips with 8 ms/trace in the opposite direction. The sample interval of the data is 4 ms. The Nyquist frequency is 125 Hz. The traces are convolved with a 4–100 Hz zero phase wavelet with two frequencies (50 and 100 Hz) notched out. Figure 2a shows the amplitude spectrum (the magnitudes of the complex-valued f-k transform samples) obtained from Figure 1a. It shows the two events. One of the events is aliased above $500/8=62.5$ Hz. The wavenumber axis is labeled as $\pm 2k_N$ because I assume that this is the desired result from interpolation and I only have every other trace of this record, as shown in Figure 1b. Let us call the data shown in Figure 1b the "known" data. Event slopes on the known data are an 8 ms/trace and a 16 ms/trace in opposite directions. The amplitude spectrum of Figure 1b is given in Figure 2b. This is the spectrum of the known data, and now both events are aliased; the one with a 16-ms/trace slope aliases above $500/16=31.25$ Hz, the other after $500/8=62.5$ Hz. Figure 1c is obtained by inserting a zero trace in between each pair of traces in Figure 1b. The amplitude spectrum of Figure 1c is Figure 2c. Figure 2c is repetitious along the wavenumber axis. Figure 1c can also be considered to be a version of Figure 1a with every other trace zeroed. There are events originating from $-2k_N$ and $+2k_N$ in Figure 2c. These events are generated because the trace zeroing process laterally shifts the original spectrum by k_N and then wraps and sums it to itself. The peak amplitude in Figure 2c is about half of the peak amplitude in Figure 2a since there are half as many live traces in Figure 1c as in Figure 1a. Our aim is to find an operator that will turn Figure 2c into Figure 2a, since that is the desired spectrum. That is, I need to find an operator that will boost the amplitudes of the events originating from $f=k=0$ by the interpolation-factor (which is equal to two in this example) and zero out the amplitudes of the events originating from $-2k_N$ and $+2k_N$.

For this purpose, I apply the following process: Zero pad the known data (Figure 1b) in both time and space with enough samples to raise the data dimensions by a factor of two. This generates Figure 1d. The Fourier transform of Figure 1d produces twice as many data samples in each direction (note: since I use only positive temporal frequencies, it is not exactly two in the case of frequencies). The amplitude spectrum of Figure 1d

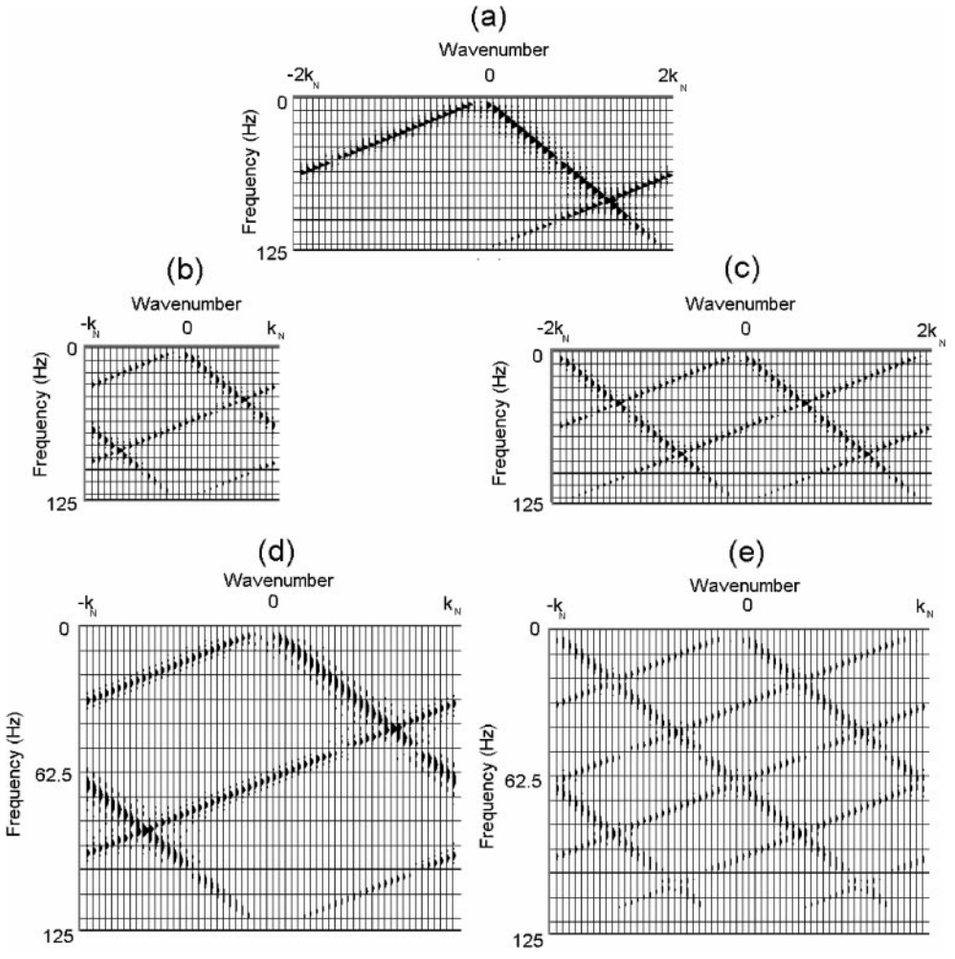

FIG. 2. The f-k domain representation of the 2D GFKI method. Amplitude spectra shown are for (a) the desired gather, (b) the known gather, (c) zero trace insertion to the known gather, (d) zero padded gather, and (e) periodic zeroing of every other trace of the zero padded gather.

is shown in Figure 2d. I observe that the data content is the same as that of Figure 2b and the Nyquist values for frequency and wavenumber have not changed. Figure 2d is an interpolated version of the spectrum in Figure 2b, and the two spectra are identical at original f-k samples.

Now, consider the spectrum given in Figure 2e. Figure 2e is a wrapped version of Figure 2d since Figure 1e is the same as Figure 1d except that every other trace is zeroed. The peaks in the transform have about half the amplitude because half of the traces are missing from Figure 1e as compared to Figure 1d. This transform (Figure 2e) can be calculated from Figure 2d by shifting it laterally by k_N, summing this result to Figure 2d, and then by dividing the sum by two. I added 1% of the peak amplitude as the bias to Figure 2e since it is going to be used for a division. The ratio of the spectrum in Figure 2d to the spectrum in Figure 2e is shown in Figure 3a for the first half of the frequency spectrum. For events originating from the origin, the results are equal to two (which is the interpolation factor). The maximum value from this division should be two (the interpolation-factor) unless there are amplitudes in Figure 2e that are smaller than the ones in Figure 2d (e.g., event crossings). Therefore, the values shown in Figure 2e are clipped at two. The ratio shown in Figure 3a is zero for events originating from $f=0$ and $k=\pm k_N$, and two for events originating from $f=k=0$. Indeed, the product of Figure 2c with Figure 3a is shown in Figure 3b, which is very close to the desired spectrum shown in Figure 2a. Note that there are many nonzero points in Figure 3a that are generally from low-energy spots in Figure 2d.

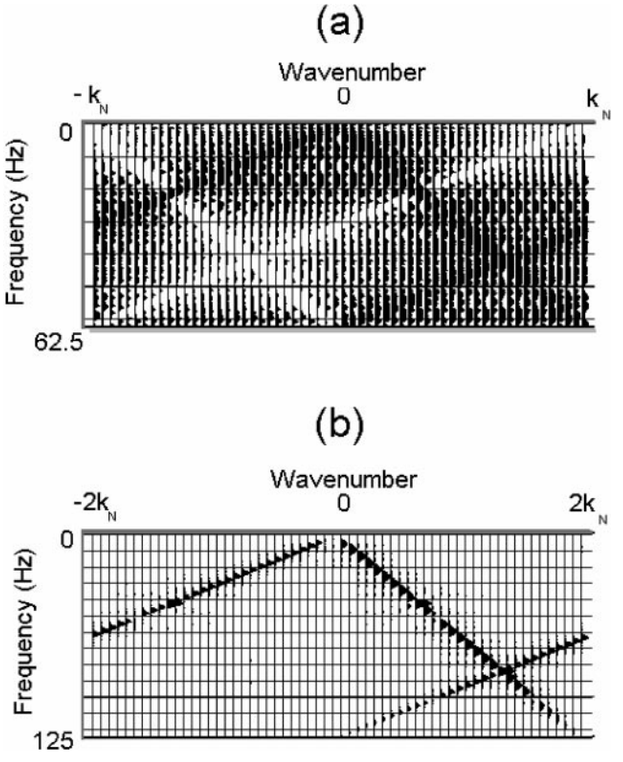

FIG. 3. (a) Interpolation operator as the ratio of Fourier transforms in Figures 2d and 2e. (b) The result of the multiplication of the copied spectrum (Figure 2c) with the interpolation operator in (a).

In fact, many wavenumbers at the 50-Hz (notch) frequency slice produce large ratios (on the order of thousands) that are later clipped to two. Since data sample values are nearly zero at these points (Figure 2c), the result of the product is also near zero (Figure 3b). This example shows that it is possible to filter the f-k spectrum of the zero-trace inserted data (repeated spectrum) and generate a spectrum close to the desired spectrum. For simplicity during this discussion, I have ignored the phase of the f-k transforms. Therefore, what I present here is not a proof but an illustration of the interpolation process used in GFKI.

For an integer interpolation factor, L, the steps for a 3D GFKI can be summarized as follows.

1) Prepare the data for interpolation.
   a) Fourier transform the data (e.g., Figure 4a) from the time-space to the frequency-wavenumber domain. If the time dimension is N_t and the space dimensions are N_x and N_y, the dimensions of the resulting data, $K(f,k_x,k_y)$, are N_f, N_x, and N_y, where $N_f=1+N_t/2$, and arguments, f,k_x,k_y, show indices rather than physical quantities.
   b) Generate a zero-trace inserted volume ($L-1$ zero-traces between each pair of traces along each space dimension; see Figure 4b for $L=2$) and transform it to the frequency-wavenumber domain. (This can be achieved much faster by laterally copying the Fourier transform samples along wavenumber direction(s) $L-1$ times. This produces L^2-1 replications for two space dimensions.) Thus,

$$C(f,k_x+iN_x,k_y+jN_y)=K(f,k_x,k_y)$$
$$i=0,1,\ldots,L-1,\quad j=0,1,\ldots,L-1. \tag{1}$$

   This process increases Nyquist wavenumbers by a factor L since trace spacing is reduced by L in both directions.

2) Construct the interpolation operator.
   a) Zero pad the time-space data in all dimensions by a factor equal to $L-1$ (final t-x-y data lengths become L times the original data lengths, LN_t, LN_x, and LN_y). See Figure 4c.
   b) Transform the zero-padded time-space data into the frequency–wavenumber domain. This process forms the "stretched transform", $S(f,k_x,k_y)$. The number of positive temporal frequencies in this transform is $1+L(N_f-1)$; the Nyquist frequency of this function is the same as of the original data, $K(f,k_x,k_y)$. Nyquist wavenumbers are also the same as those of the original data. The wavenumber increment is L times finer than that of the original data. Use only the first N_f of the temporal frequencies produced, ignore the rest. This becomes the numerator of the interpolation operator.
   c) Impose the same zero-trace pattern that exists in the zero-trace inserted volume (e.g., Figure 4b) to the zero-padded time-space data (e.g., Figure 4c) by zeroing the appropriate traces in the zero-padded volume (see Figure 4d). Transform those data into

the frequency–wavenumber domain. Call the result $Z(f, k_x, k_y)$. Alternatively, this transform can be derived from the stretched transform by shift and sum operations along the wavenumber axis:

$$Z(f, k_x, k_y) = \frac{1}{L^2} \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} S(f, k_x + iN_x, k_y + jN_y). \quad (2a)$$

For one wavenumber, it is calculated from

$$Z(f, k_x) = \frac{1}{L} \sum_{i=0}^{L-1} S(f, k_x + iN_x). \quad (2b)$$

d) Add a small bias to $Z(f, k_x, k_y)$ where its magnitude falls below the bias.
e) Form the operator by dividing the result of step 2b by the result of step 2c:

$$H(f, k_x, k_y) = \frac{S(f, k_x, k_y)}{Z(f, k_x, k_y)}. \quad (3)$$

f) Clip the operator, H, at the expected peak value, L^2 (at L if there is only one wavenumber axis).

3) Filter and inverse transform.
a) Multiply the operator derived in step 2f with the copied spectrum in step 1b:

$$G(f, k_x, k_y) = H(f, k_x, k_y)C(f, k_x, k_y). \quad (4)$$

b) Inverse Fourier transform, G, from the f-k_x-k_y domain to the t-x-y domain.

The denominator in equation (3), $Z(f, k_x, k_y)$, is repetitious in the wavenumber plane. It has L repetitions along both the k_x and k_y directions. Therefore, it needs to be calculated only over the original wavenumber range ($k_x = 0, 1, \ldots, N_x - 1$ and $k_y = 0, 1, \ldots, N_y - 1$). The rest can be obtained by laterally copying it $L^2 - 1$ times in the wavenumber plane (Gülünay and Chambers, 1997b, c). Note that $C(f, k_x, k_y)$ has one original and $L^2 - 1$ copies as well. Division of $C(f, k_x, k_y)$ with $Z(f, k_x, k_y)$ [when equation (3) is inserted into equation (4)] can be considered to be the main factor in filling the zero traces inserted into original data.

The interpolation filter described above is like an on-off switch. It is equal to zero where events are aliased (linear events that do not originate from $f = k_x = k_y = 0$), and it is equal to the square of the interpolation factor, L, for strong linear events that originate from $f = k_x = k_y = 0$. At places where the numerator is very small or when side lobe ringing occurs, one may get other values (as in Figure 3a). Such values need to be dealt with separately (see the next section).

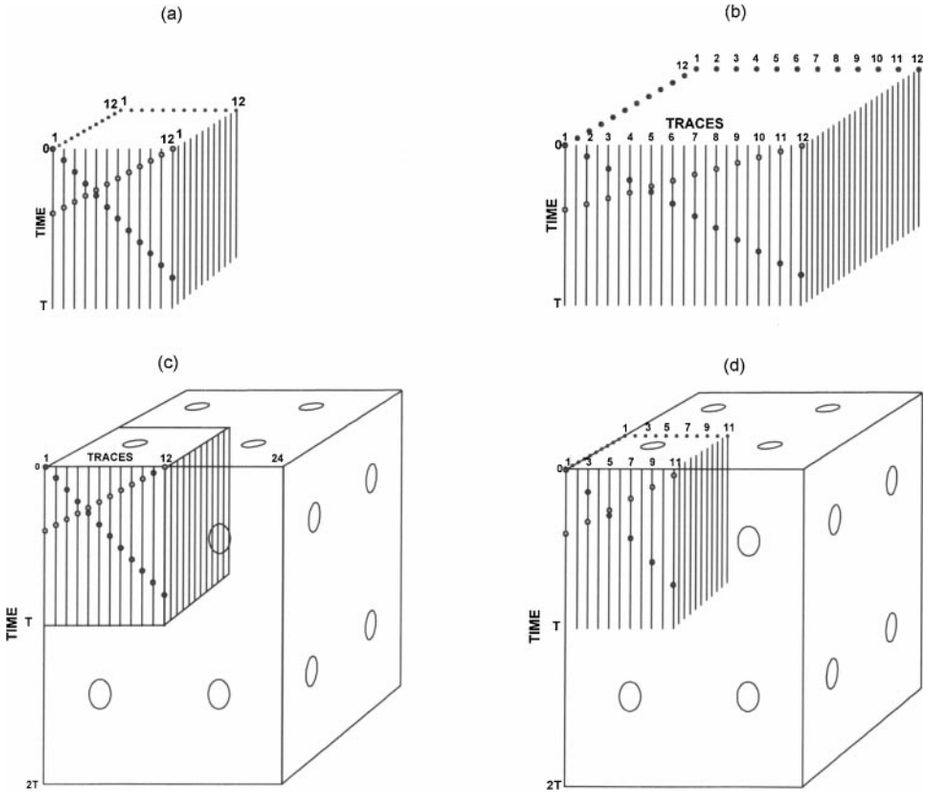

FIG. 4. The t-x-y domain representation of the 3D GFKI method: (a) known data volume, (b) zero trace insertion between existing traces, (c) zero padding of the original volume both in time and space directions, (d) zeroing of every other trace of the zero padded volume. Note that zero trace insertion in (b) and periodic trace zeroing in (d) are done in both space directions.

This operator interpolates the data at the original temporal frequencies. Note that both the frequency and wavenumber content of the data that are used in the operator design are lower by a factor of L than the content of the output (wavenumber content of the operator is the same as of the original data; it is just "stretched").

For a single dipping event, it is possible to prove mathematically that this operator interpolates the data exactly. For multidip data, I don't have such a proof, but hope that the model and field data examples I provide here will justify the method. Let it suffice to say that the 2D version of the GFKI operator derived from equation (3) is similar to the f-x interpolator (as shown in Appendix A) and that it is defined as the interpolator that interpolates data that are coarser in the space direction by a factor L, but at a frequency that is lower by a factor L. The GFKI method simply assumes that the same spatial interpolator applies to the data at original trace positions and at original frequencies.

PRACTICAL CONSIDERATIONS

Linearity and spatial-temporal windowing

The GFKI process is based on the assumption that events to be interpolated are linear in the time-space domain and that there is a small number of events. This assumption may be violated in real data, especially in prestack data where there is significant curvature in the offset direction. Therefore, any curvature in the data must be reduced as much as possible before interpolation. Since interpolation is often done to help multichannel processes like Radon transform to work better, the interpolator should successfully interpolate multiples as well as primaries. NMO application with a velocity trend between primaries and multiples works well in reducing offset-related curvature. Dividing the data into small time-space gates is also necessary since events can have curvature (e.g., Figure 8). Depending on the trace spacing, dividing the data into overlapping time-space windows of 8–16 traces with 64–128 time samples often seems satisfactory. Since manipulation of data in the Fourier transform domain (such as operator taper, discussed below) can produce wraparound artifacts, the data windows may need zero padding (e.g., four-trace and 100-ms zero padding).

FFT sizes

Data sizes N_t and N_x (and N_y) must be valid FFT sizes to be able to make use of FFT for Fourier transforms. In addition, all other FFT sizes used in the process [such as LN_t, LN_x (and LN_y), and N_x/L, (and N_y/L)] must be valid.

Hybrid interpolator

The GFKI interpolator and f-x interpolator can be used for all frequencies whether the data alias or not as long as the data has low-frequency content. When the data do not alias, one might as well use a sinc interpolator since it is inexpensive and adequate for such data. This leads to a hybrid GFKI interpolator. Li (1995) similarly used a sinc interpolator at frequencies that are not aliased during the f-x interpolation. This requires a maximum dip specification by the user. In the GFKI method, the sinc interpolator is implemented by setting the GFKI operator to zero at wavenumbers that are beyond the original wavenumber range supported by the data, and to a constant (L for 2D GFKI and L^2 for 3D GFKI) otherwise.

In most prestack data, dips are less than 10 samples per trace after application of NMO. This suggests that frequencies lower than one-tenth of the temporal Nyquist frequency, both f-x and GFKI operators can be replaced with their sinc counterpart, leading to "hybrid" trace interpolators. Furthermore, in the CMP domain, data in a space window that contains near offsets do not have as much dip as the one that contains far offsets. For such near-offset space windows, almost all of the temporal frequencies are below the aliasing frequency and hence can be processed with the sinc interpolator given above. Therefore, one may use an offset-dependent maximum dip specification and vary the aliasing frequency used in the hybrid interpolation as a function of the location of the space window.

Lagrangian quadratic interpolation of the f-k domain samples

The data size after zero padding is L^2 times as big as the original data size for 2D data. For 3D data, the data size is L^3 times as big. Fourier transforms that do not take advantage of the large number of zeroes in such data volumes can get expensive. As stated previously, the zero padding in the input domain is the same as interpolating in the Fourier transform domain. Since interpolated Fourier transform samples are only going to be used during operator design, some errors in their calculation can be tolerated. Indeed, these errors are somewhat inconsequential since ratios are used rather than actual transform samples. Therefore, one may attempt to interpolate Fourier transform samples with short operators rather than calculate the Fourier transform of the zero-padded sequences with a large number of zeroes. Lagrangian interpolation filters described by Laakso et al. (1996) can be used for this purpose.

Operator taper in the t-x domain

The spatial and temporal extent of the resulting interpolation operator may need to be reduced. When the data are noisy, the operator is known to extend in time, as t-x domain equivalents of f-x prediction filters do on noisy data. This problem can be effectively solved by taking the f-k domain operator into the t-x (or t-x-y) domain and then temporally and spatially tapering it before taking it back to the f-k domain to apply to the data. Such a taper on the interpolator is equivalent to smoothing the operator in the f-k domain. Whereas a temporal taper suppresses inaccuracies in the operator due to data noise, a spatial taper forces the f-k interpolator to be small in the spatial extent like f-x domain prediction filter.

Preserving the original traces and noise reduction on the interpolated traces

In GFKI, the interpolation operator can generate new traces at the original locations. These traces are later replaced with the original traces. Keeping original traces intact, however, creates a practical problem: interpolated traces are less noisy than the original traces. For this reason, phase-preserving modifications on the amplitude spectrum of the interpolated traces may be needed. Alternatively, the amount of noise on the original traces can be estimated and added to the interpolated traces.

The need for addition of white noise

GFKI operators are derived by dividing two spectral functions. Trace interpolation can be considered as an f-k domain deconvolution process. As in all deconvolution processes, the f-k domain trace interpolation algorithm needs the addition of white noise. This addition is done to the denominator to eliminate division by very small numbers. A threshold determined from all of the amplitude spectrum (f-k or f-k_x-k_y) samples can be used, and samples in the denominator that are below this threshold can be replaced by this threshold during the division.

The need for clipping the operator

When both the numerator and the denominator used in the operator design have small values, the division may produce large values even after the thresholding described above. Values larger than the expected peak value (L^2 or L) need to be clipped to that value. Also, the operator values that are less than half the expected peak value can be safely set to zero since the operator is generally equal to the expected peak value when the amplitudes in the spectrum are large. One exception is where events cross in the f-k domain. At these spots, the result of the division may have amplitudes larger than the expected peak value, and those values need to be clipped as well.

MODEL DATA EXAMPLES

A 2D synthetic gather made of four dipping events is shown in Figure 5a. The f-k spectrum of this gather is shown in Figure 5b. This figure shows that all of the dipping events are aliased. The result of GFKI trace interpolation of this gather

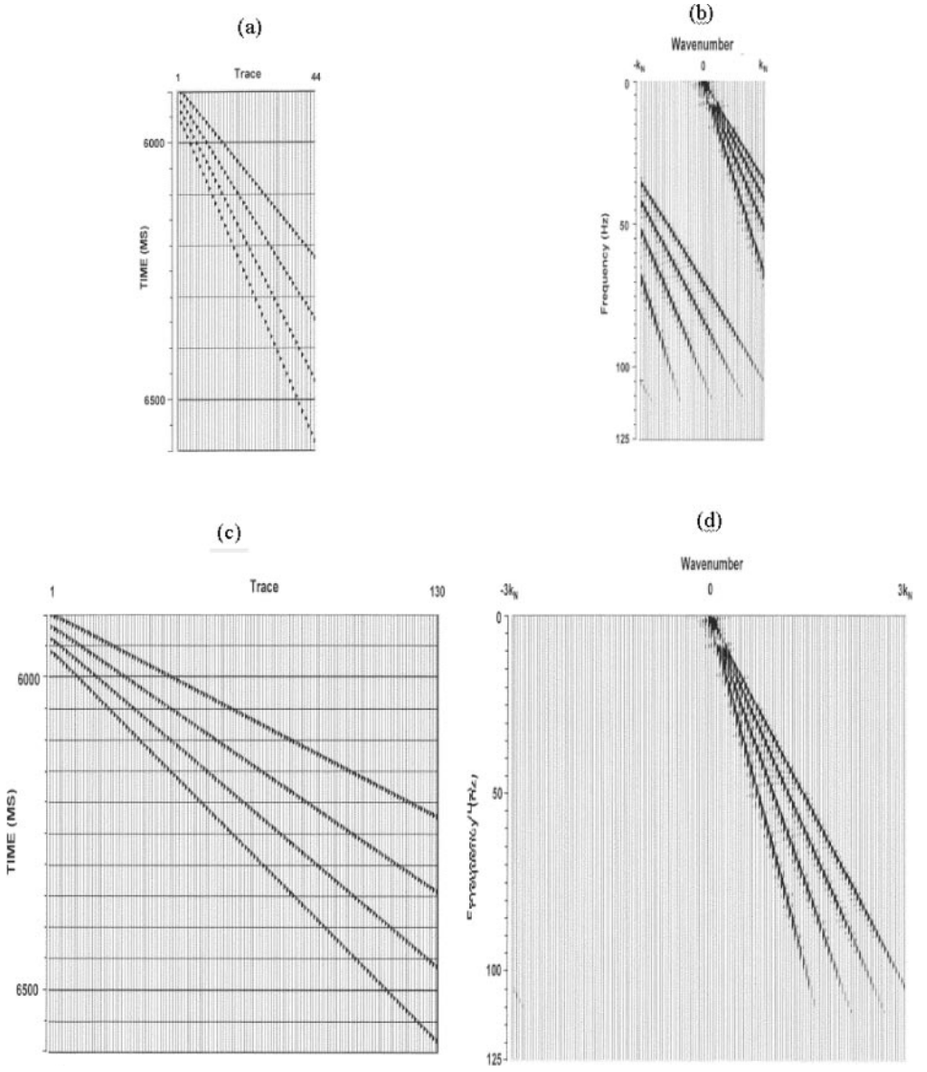

FIG. 5. (a) A synthetic gather containing four dipping events; all of them are aliased. (b) The f-k spectrum of the gather. (c) GFKI interpolation of the gather. The interpolation factor is three. Traces 1, 4, 7, . . . , 127, 130 are the original traces. Traces 2, 3, 5, 6, ... , 128, 129 are the interpolated traces. (d) The f-k spectrum of the GFKI interpolated gather. Note that the spectrum is unwrapped and the original Nyquist wavenumber is extended three times compared with (b).

with an interpolation factor $L=3$ is given in Figure 5c. The f-k spectrum of the interpolated gather is shown in Figure 5d. Clearly, the GFKI method unwraps the spectrum as expected.

An example of the application of GFKI to 3D data uses the model given in Figure 6. This model contains two aliased dipping planes (made of single spikes) given by equations $k=2i+2j$ and $k=2i+4j$ where i and j are trace indices along the receiver and shot directions; k is the time sample index. The terms "source" and "receiver" are used here instead of crossline and inline since source and receiver directions are two orthogonal directions used when GFKI interpolation is applied to prestack 2D data. There are 32 input traces in both the source and receiver directions. Plots of the 54th, 55th, and 56th time slices are shown in Figures 7a, 7b, and 7c, respectively (triangular wavelet shapes are due to the interpolation in the plotting routine). On the 54th time slice, a steeply dipping event occurs on a line joining (receiver, source) points (25, 1) and (1, 13). The other dipping event occurs on a line joining (26, 1) to (1, 26). The 55th time slice (Figure 7b) is empty because there is no energy in odd-numbered time samples since the slopes are even multiples of the time index. In other words, both $2i+2j$ and $2i+4j$ are even and cannot be equal to an odd number like 55. Figures 7d, 7e, and 7f show the corresponding time slices after 3D GFKI using an interpolation factor of 2, which results in 64 traces in each direction. The wavelet shapes of the interpolated data are fairly consistent, and time slice 55 (Figure 7e) contains nonzero amplitudes on interpolated traces.

FIELD DATA EXAMPLES

Shot interpolation in the CMP domain with 2D GFKI

For 2D data, or for a CMP line of a 3D marine survey, interpolation is often needed because of the coarse shot interval (Jakubowicz, 1994). Two-dimensional versions of the methods presented above can be applied on CMP gathers to interpolate new shots, which reduces aliasing before applying multichannel processes like f-k filter or Radon transform. Applications in other domains, such as common receiver or common offset, are also possible. Although the CMP domain has the disadvantage of having large trace intervals (and hence more curvature for a given number of traces), it has the practical advantage of being the domain where a multichannel process like Radon transform for multiple elimination will be applied. It also has the advantage of avoiding smear over nearby CMPs since traces of each CMP are interpolated independently. Another advantage is the offset-varying maximum dip that allows most of the temporal frequencies of the inner offsets to be processed with a sinc interpolator.

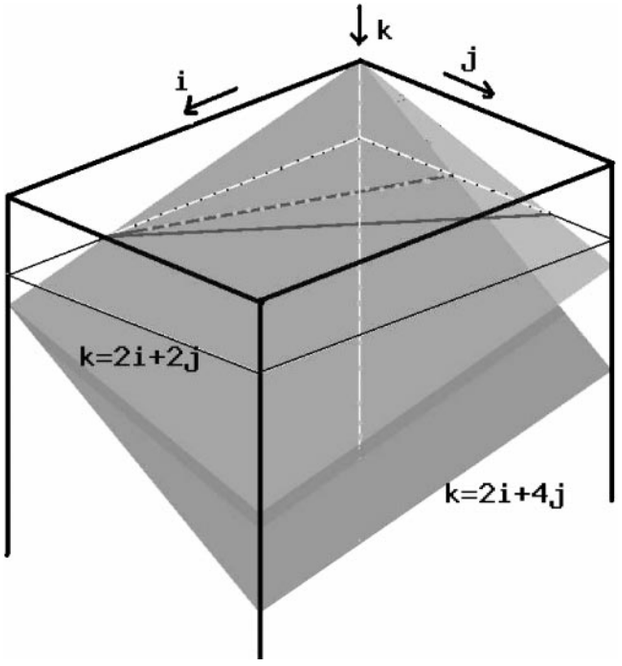

FIG. 6. Two dipping planes with an inline dip of 2 samples per trace and crossline dips of 2 and 4 samples per trace.

A small time window (5800–6800 ms) of an 88-fold CMP gather from the Gulf of Mexico is shown in Figure 8. An NMO function derived from a velocity function chosen between primary and multiple trends was applied to this gather. The maximum offset is about 9000 m. Primaries are the overcorrected events, and the water bottom multiple is the strong undercorrected event at about 6030 ms. Figure 9 shows a decimated gather formed from Figure 8. A sinc-type trace interpolation is bound to fail at the far offsets due to the severe dips and the presence of high frequencies. Indeed, the moveout difference between the primary and the multiple events is more than 2 s at the far offset. Figure 10 is a 2:1 GFKI interpolation on the decimated gather and has 87 traces (trace 88 is not output since it is an extrapolated trace). Figure 11 is the difference between the original full-fold gather and the gather formed by interpolating the decimated gather (plotted at the same scale as Figures 8–10). The interpolation is fairly successful. It is least accurate at the locations where there is large curvature.

Note that the 2D GFKI algorithm to interpolate new shots can be applied in the common receiver domain as well. Indeed, in this domain there is less aliasing, and interpolation can succeed better. Whereas the new shots interpolated in a CMP are derived from other shots and receivers, the new shots interpolated in the common receiver domain are derived from other shots shooting into the same receiver. The only disadvantage here is that one has to sort the data back to the common receiver domain. The CMP domain is already available when one is ready to apply Radon transform or even stack.

Shot interpolation in the source-receiver domain with 3D GFKI

The shot interpolation in the CMP domain discussed above becomes less reliable as noise in the data increases. One may improve interpolation results by operating in more than one space direction simultaneously. One such domain is the source-receiver domain. Assuming that the source interval is an integer multiple of the receiver interval, overlapping time windows of a small data window such as 10 shots $\times$ 20 receivers can be processed with the 3-D version of the algorithm discussed above to produce new shots. This process is illustrated for a surface geometry shown in Figure 12. Of course 3D GFKI creates interpolated receivers as well, but these receivers are not needed and can be dropped after spatially limiting the receiver wavenumber axis to the original band. Space windows overlap along the source and receiver directions. Note that a temporary

NMO application derived from a velocity function between primary and multiple velocity trends (as discussed previously) is necessary to make the seismic events approximately planar in this domain.

The results of such a 2:1 shot interpolation in the source-receiver domain from a dual-source "flip-flop" marine 3D recording from offshore Angola are shown in Figures 13–18. These results are from an area where the water-bottom arrival is near 3000 ms. The shot interval between two "flips" (or "flops") is 50 m, and the group interval is 25 m. Figure 13 shows a part of an original shot. The event near 6120 ms at trace 1 is the first multiple of the strong water bottom, and the event around 7000 ms at trace 1 is the first strong interbed multiple. Figure 14 shows an interpolated shot, 25 m away from the shot shown in Figure 13. Figure 15 is a CMP gather (NMO corrected with primary velocities) formed from the original shots. Figure 16 shows the same CMP gather after source-receiver domain (3D GFKI) interpolation. Note that every other trace in Figure 16 is an interpolated trace, and these traces are consistent with the original data. In general, when data are noisy, I find the 3D version of the method to provide more consistent interpolated traces than the 2D version. This may be attributed to the stability obtained from the additional space direction. Figures 17 and 18 show stacks of the data (zoomed around multiples) before and after shot interpolation, respectively. Chatter in the stack in Figure 17 is due to the multiples shown in Figure 15 leaking into the stack due to the coarse trace spacing. Although

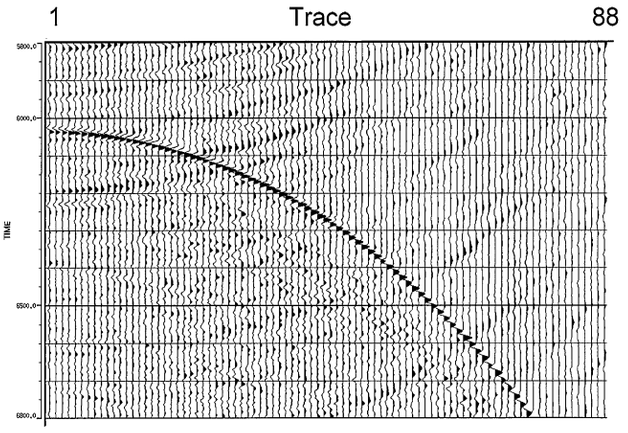

FIG. 8. An 88-fold CMP gather from the Gulf of Mexico.

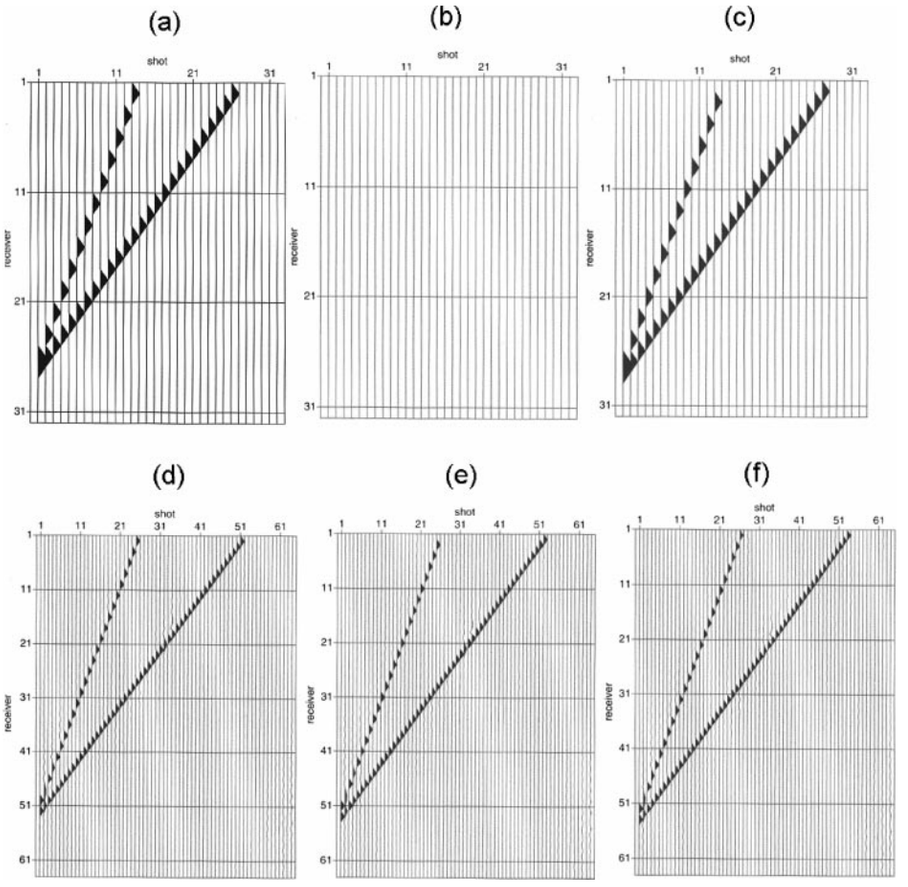

FIG. 7. (a) The 54th time slice of data generated by the model in Figure 6. (b) The 55th time slice. There are no data in this time slice. (c) The 56th time slice. (d) The 54th time slice after 3D GFKI interpolation. The interpolation factor is two. (e) The 55th time slice after 3D GFKI interpolation. (f) The 56th time slice after 3D GFKI interpolation.

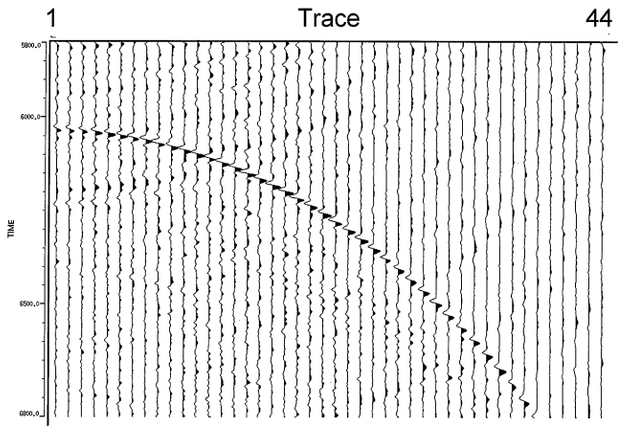

FIG. 9. CMP gather of Figure 8 after decimation by a factor of two, resulting in a 44-fold gather.

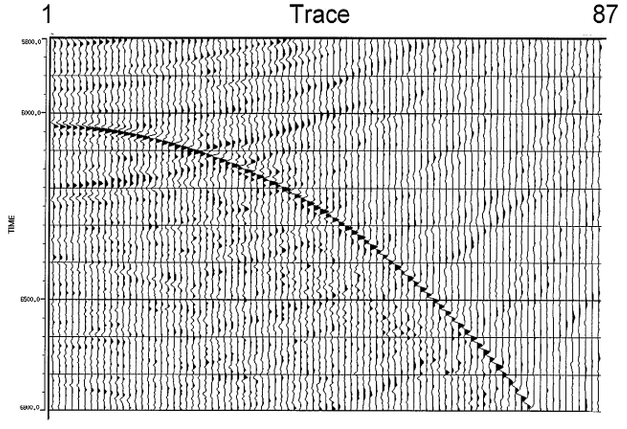

FIG. 10. A 2D GFKI interpolation of the decimated gather in Figure 9. The interpolation factor is two. Traces 1, 3, 5, . . . , 85, and 87 are the original traces, and traces 2, 4, . . . , 84, 86 are the interpolated traces.

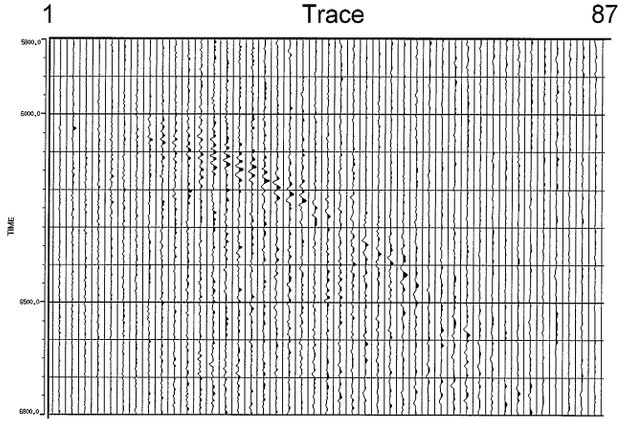

FIG. 11. Difference between the original gather in Figure 8 and the interpolated gather in Figure 10.

the chatter which is due to the flanks of the multiple curves goes away after stacking the interpolated data, the apexes of the multiple curves stack coherently, and therefore most of the data shown in Figure 18 are multiples requiring suppression by Radon transform filtering or f-k filtering.

Cable interpolation with 3D GFKI

Recent 3D marine recordings employ more than one cable for each shot. Although the typical receiver interval is 25 m, the cable separation could be 100 or even 200 m, which leads to crossline artifacts in the processed data. When cable separation is an integer multiple of the receiver interval, data can be interpolated in the crossline direction using the 3D GFKI algorithm. Such a crossline interpolation suppresses artifacts. About 20 receivers at a time from all of the existing cables (in the form of overlapping space gates) are used to interpolate new cables.

One problem in cable interpolation using 3D GFKI is the narrowness of the space window across the cables. One may only have a small number of cables (e.g., four) to interpolate, which may produce a very blurry f-k_x-k_y spectrum, and the

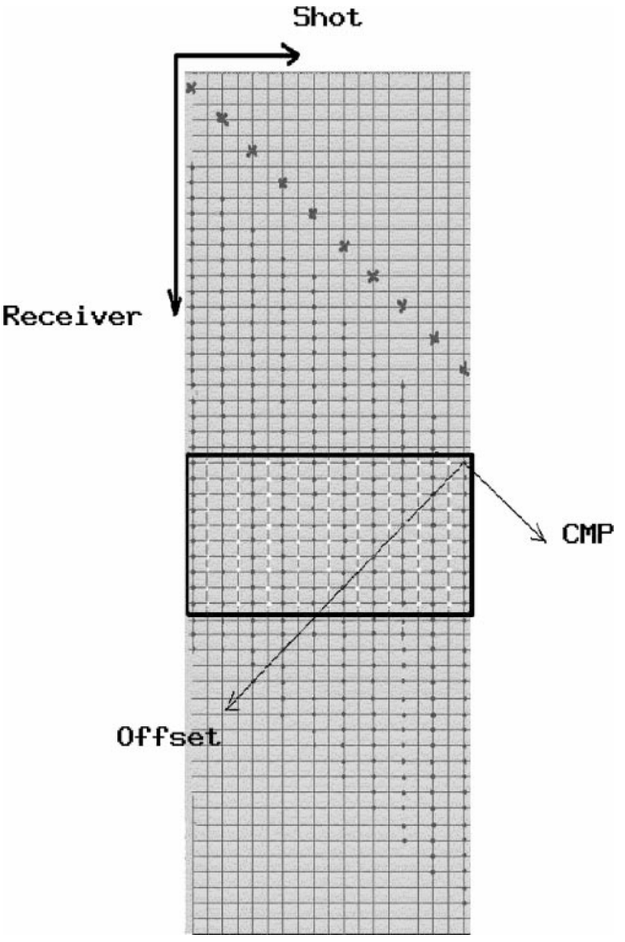

FIG. 12. Surface diagram used in source-receiver domain shot interpolation. A small space window is used to generate interpolated shots. Space windows overlap along both source and receiver directions.

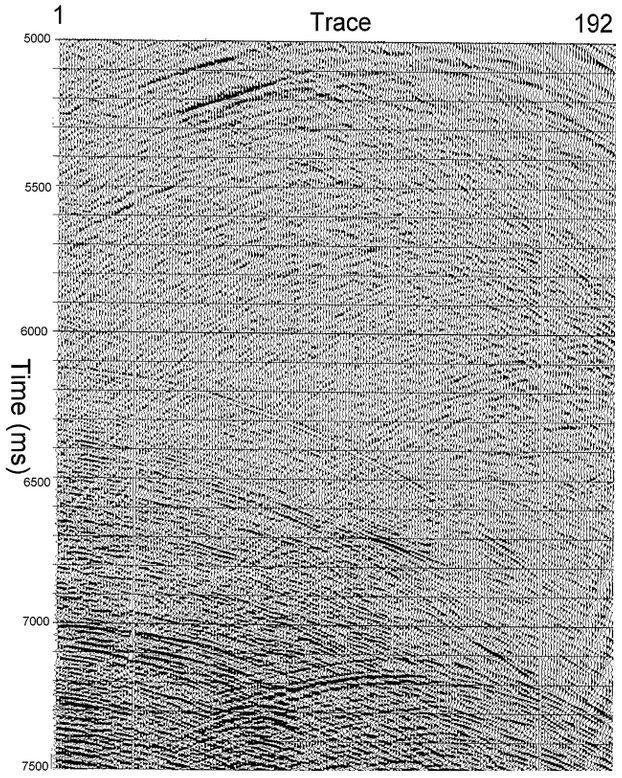

FIG. 13. An original 192-trace shot record from a 3D marine survey, offshore Angola.

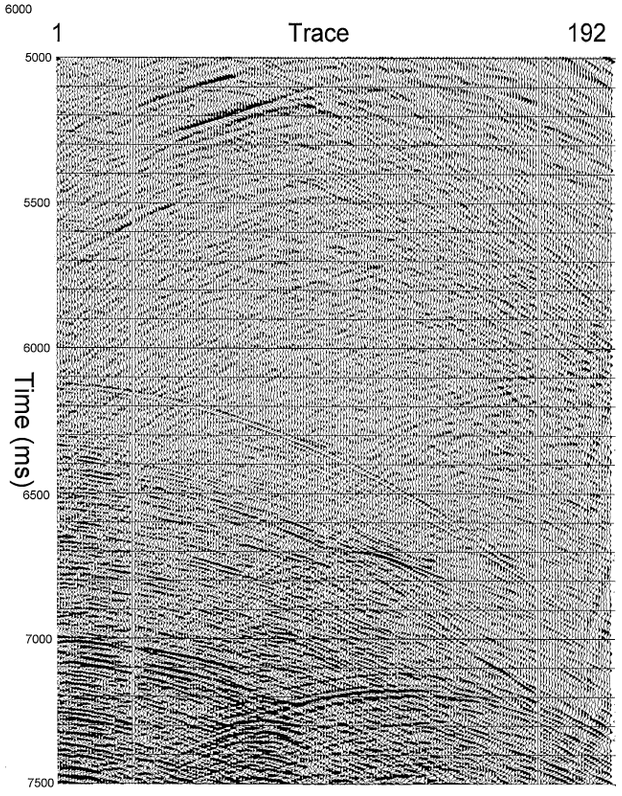

FIG. 14. A shot interpolated with the 3D GFKI algorithm. This shot is 25 m away from the original shot.

wide cable separation may add to the problem. For these situations, it is still possible to obtain reasonable interpolation results as shown in Figure 19. Here 6-cable data from a 3D survey in Angola were interpolated to produce 11-cable records. Even numbered cables are the interpolated cables. The original cable separation is 100 m. There are 96 receivers per cable, and

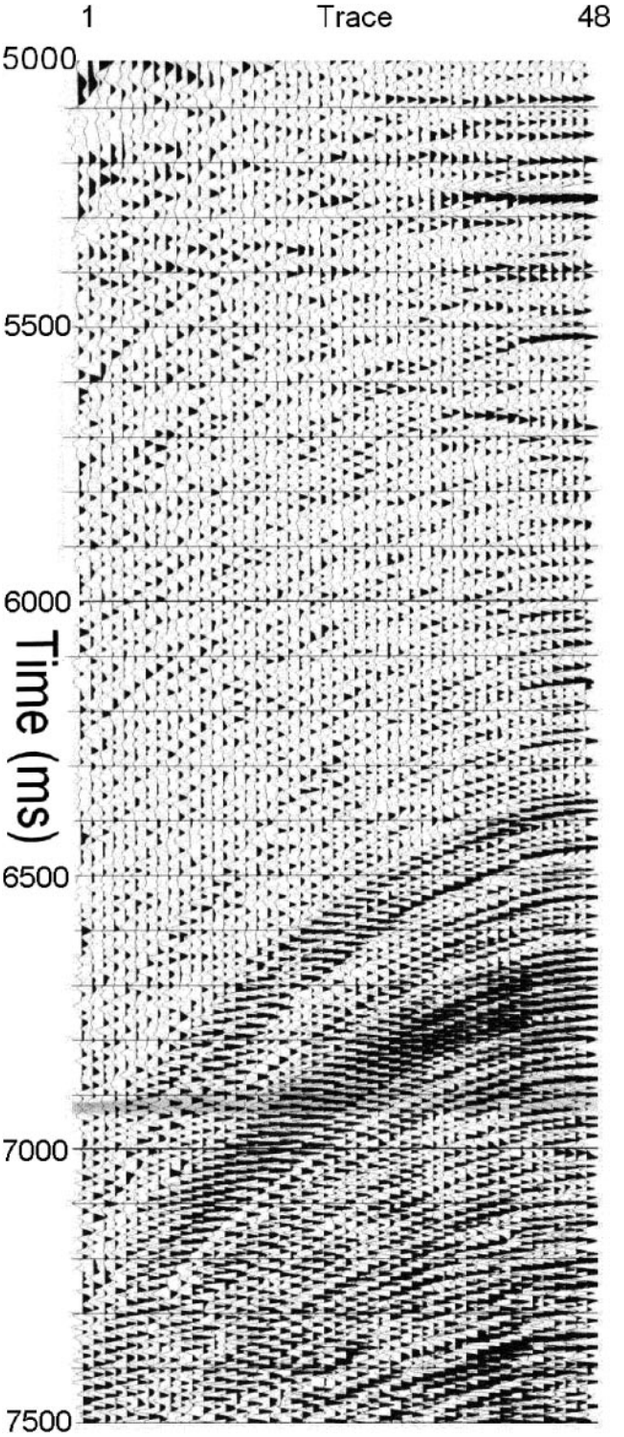

FIG. 15. A 48-fold CMP gather formed from the original shots.

the receiver interval is 25 m. The zoom area that is marked with a box is shown in Figure 20. The middle cable in this figure is the interpolated cable, and the other two are the original cables.

CONCLUSIONS

Data-adaptive trace interpolation of aliased data is possible by making use of properties of Fourier transforms. The interpolation algorithm is very efficient since fast Fourier transforms are used. The method, which is referred to in this paper as the generalized f-k interpolation (GFKI) method, works on frequency slices. Data are first Fourier transformed from the t-x (or t-x-y) domain to the f-k (or f-k_x-k_y) domain. Then the f-k data are laterally copied along the wavenumber axis (along k or along k_x and k_y). Lateral copying corresponds to zero-trace insertion at periodic locations where interpolated traces are desired in the t-x (or t-x-y domain). At a given frequency slice, an interpolation operator is designed from the data at a frequency that is equal to the original frequency divided by the interpolation factor. While designing the operator, it is assumed that the dip content of the lower frequencies is the same as the dip content of the original frequencies and that events are linear in the input records. To satisfy this assumption, curvature reduction on input gathers is done by application of NMO corrections, which are later removed from interpolated data. Small time-space windows are used to make the events appear linear in these windows. The operator produces the Fourier transform samples of the interpolated data when multiplied with the samples of the current frequency. Inverse Fourier transform of these samples form the interpolated data in the t-x (or t-x-y) domain.

Using the GFKI algorithm, a CMP domain or a source-receiver domain shot interpolation is done to reduce the shot interval if it is an integer multiple of the receiver interval. The GFKI algorithm is also used for cable interpolation. The algorithm can also be applied to poststack data to reduce

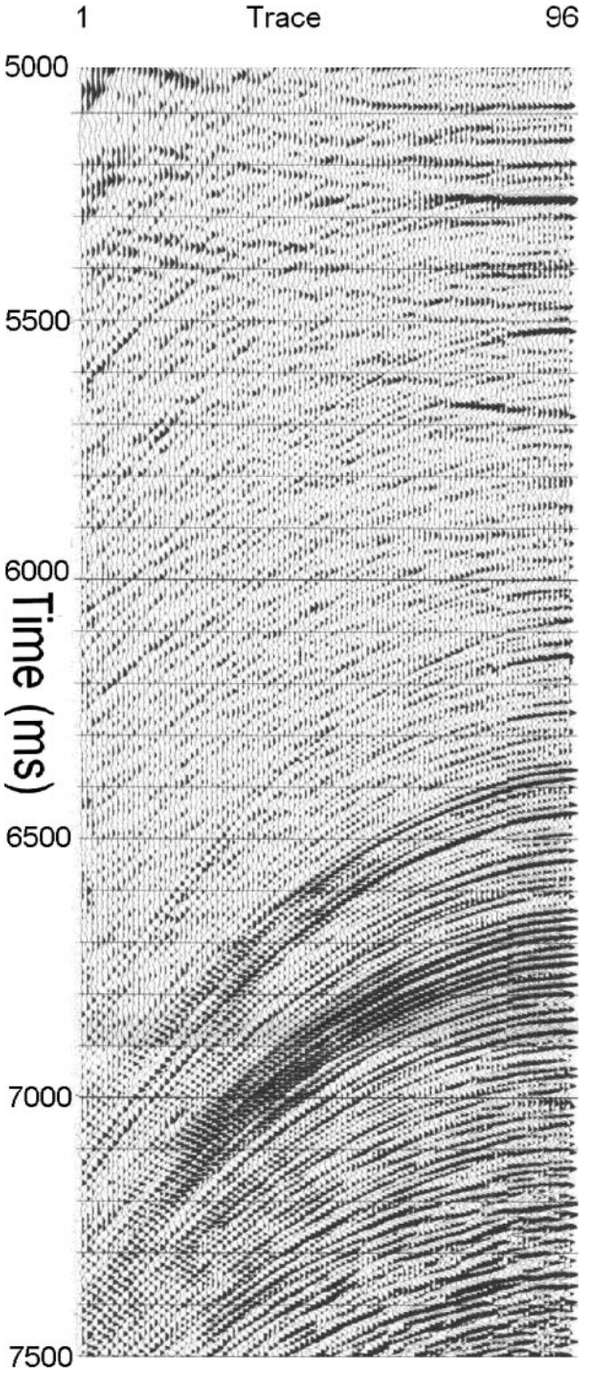

FIG. 16. CMP after source-receiver domain shot interpolation with 3D GFKI.

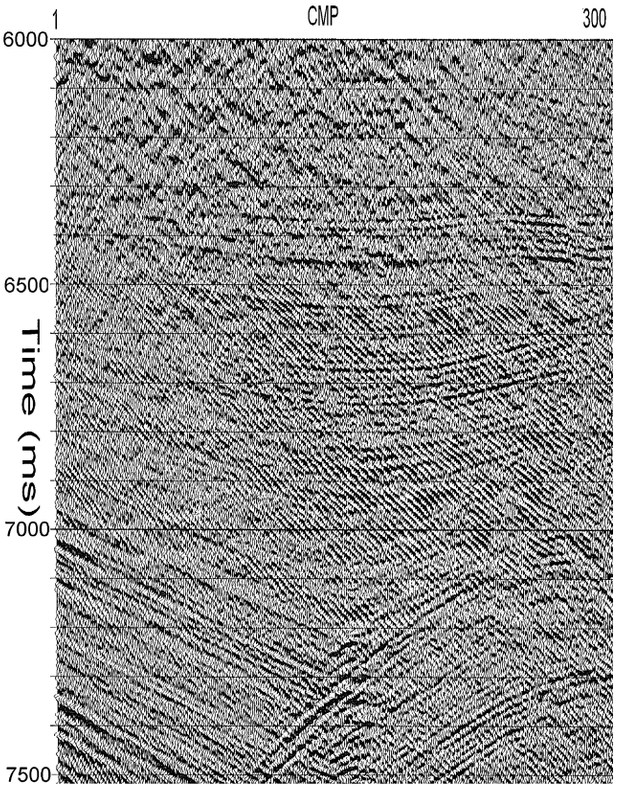

FIG. 17. Stack of the 48-fold CMP gathers (like the one in Figure 15) containing original traces. The chatter in the deep portions of the data is due to the aliasing of the multiple energy.

crossline (and, if desired, inline) bin spacing. Interpolation of field data during data processing helps to suppress artifacts that aliased data create when processed with multichannel algorithms.

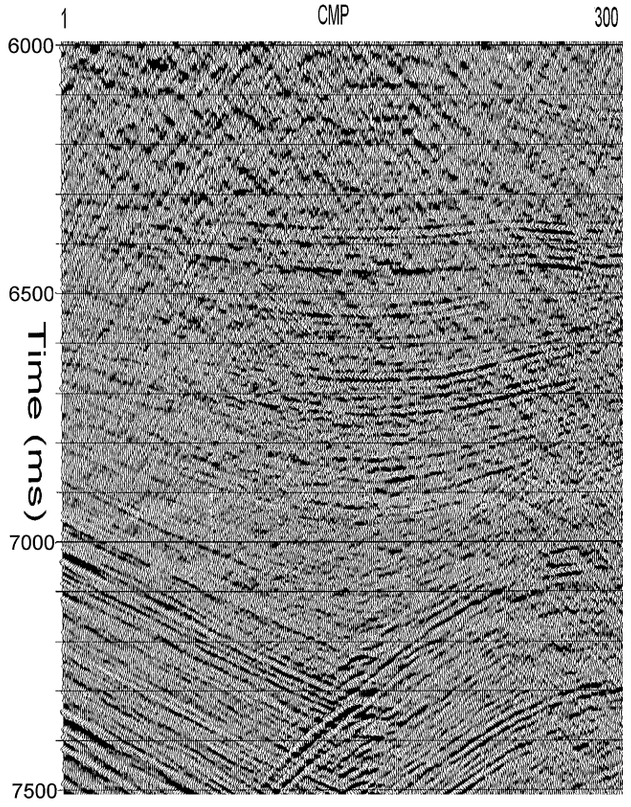

FIG. 18. Stack of CMP gathers (like the one in Figure 16) containing interpolated data.

The GFKI algorithm is similar to the *f-x* prediction filter interpolation. Both use the low-frequency portion of the data to derive the interpolation operators. Major differences are that there is a filter length specification involved in *f-x* interpolation while there is no such requirement in the GFKI algorithm. In practice, a taper is used in the GFKI operator in the *t-x* domain, which corresponds to having short *f-x* domain operators along the *x* direction. Both algorithms produce similar results on field data, with GFKI proving to be the most cost effective of the two.

ACKNOWLEDGMENTS

I thank Craig Beasley for pointing out the need for *f-k* domain trace interpolators and Ron Chambers for many inspiring discussions during the development of the algorithm. I am

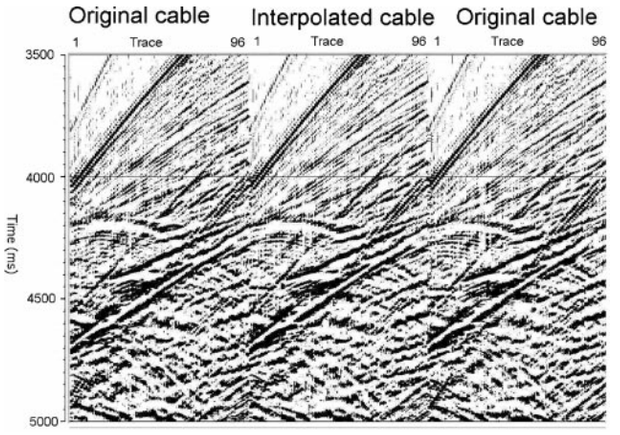

FIG. 20. An interpolated cable between two original cables (boxed area in Figure 19).

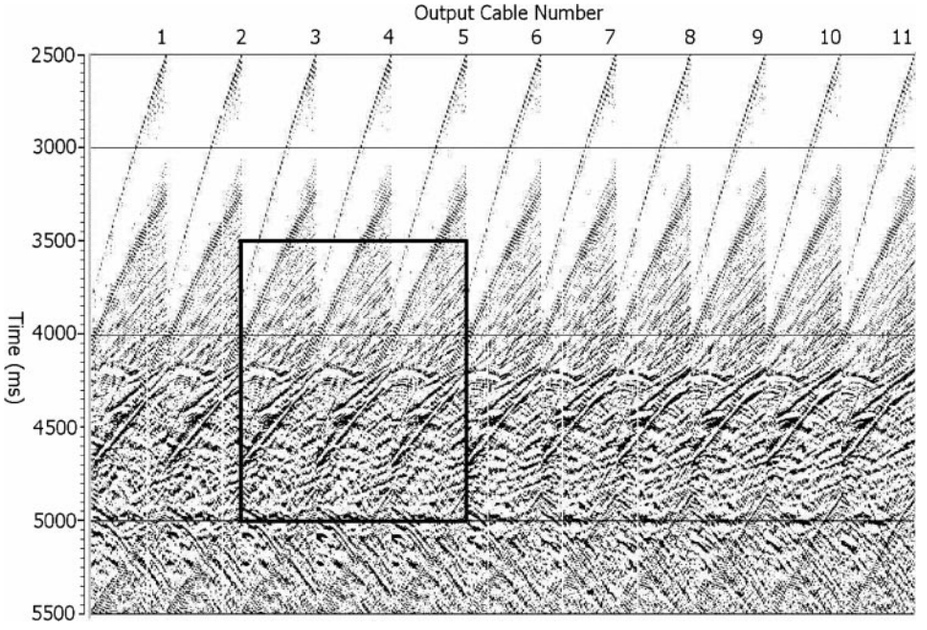

FIG. 19. Eleven cables resulting from cable interpolation of six original cables. Odd numbered cables are the original cables. Boxed area is shown in Figure 20.

grateful to Roger Strahl, Del Byrd, Stan Goldberg, and especially to Melanie Ianniello for their programming support. Chris Koeninger, Andrew Furber, Stewart Archer, Rainer Moerig, Everett Mobley, Robert Vauthrin, Fran Goloway, and Jinny Minter kindly provided feedback on the production implementation of this algorithm. Reviews by my colleagues, Bill Dragoset, Laurent Meister, Sven Treitel, and by the reviewers of *Geophysics*, S. Spitz, H. Jakubowicz, P.W. Cary, and one anonymous reviewer made the paper more understandable. I thank them all. I also thank WesternGeco for allowing me to publish this paper.

REFERENCES

Bracewell, R. N., 1978, The Fourier transform and its applications: McGraw-Hill Book Co.

Burrus, S., 1985, DFT/FFT and convolution algorithms: John Wiley & Sons.

Claerbout, J., 1992, Earth soundings analysis: Blackwell Scientific Publ.

Claerbout, J., and Nichols, D., 1991, Interpolation beyond aliasing by (t-x) domain PEFs: 53rd Ann. Internat. Mtg., Eur. Assn. Geosci. Eng., Expanded Abstracts, paper A001.

Darche, G., 1990, Spatial interpolation using fast parabolic transform: 60th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1647–1650.

Gülünay, N., and Chambers, R. E., 1996, Unaliased f-k domain trace interpolation (UFKI): 66th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1461–1464.

——— 1997a, Unaliased f-k domain trace interpolation in the f-k domain: US Patent 5 617 372.

——— 1997b, Generalized f-k domain trace interpolation: 67th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1100–1103.

——— 1997c, Unaliased spatial trace interpolation in the f-k domain: US Patent 5 677 892.

Guo, J., Zhou, X., and Yang, H. J., 1996, Efficient f-k domain seismic trace interpolation for spatially aliased data: 66th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1457–1460.

Jakubowicz, H., 1994, Wavefield reconstruction: 64th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1557–1560.

——— 1997, Seismic data acquisition: US Patent 5 648 938.

Ji, J., 1993, Interpolation using prediction-error filter simulation (PEFs): 63rd Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1170–1173.

Kabir, M. M. N., and Verschuur, D. J., 1992, Applications of the generalized Radon transform: 54th Ann. Internat. Mtg., Eur. Assn. Geosci. Eng., Extended Abstracts, paper B052.

King, G. A., Leong, T. K, and Flinchbaugh, B. E, 1984, The role of interpolation in seismic resolution: 54th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 766–767.

Kostov, C., 1989, Finite aperture slant-stack transforms: Stanford Exploration Project **61**, 261–279.

Laakso, T. I, Valimaki, V., Karjalainen, M., and Laine, U. K., 1996, Splitting the unit delay: IEEE Signal Processing Mag., January, 30–60.

Larner, K., Gibson, B., and Rothman, D., 1981, Trace interpolation and the design of seismic surveys: Geophysics, **46**, 407.

Li, G., 1995, Joint trace interpolation in f-k and f-x domains: Chinese Oil Geophys. Prosp., **30**, 693–701.

Lu, L., 1985, Application of local slant-stack to trace interpolation: 55th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 560–562.

Manin, M., and Spitz, S., 1995, Wavefield de-aliasing for acquisition configurations leading to coarse sampling: 65th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, p. 930–932.

Monk, D., McBeath, R. G., and Wason, C. B.,1993, Interpolation of aliased seismic traces: US Patent 5 235 556.

Pan, K., and Fields, J. L., 1986, Seismic trace interpolation using f-k filtering: US Patent 4 594 693.

Pieprzak, A. W., and McClean, J. W., 1988, Trace interpolation of severely aliased events: 58th Ann. Internat. Mtg., Soc. Exp. Geophys., Expanded Abstracts, 658–660.

——— 1990, Interpolation of severely aliased events: US Patent 4 922 465.

Porsani, J. M., 1999, Seismic trace interpolation using half-step prediction filters: Geophysics, **64**, 1461–1467.

Schonewille, M. A., and Duijndam, A. J. W., 1996, Decomposition of signal and noise for sparsely sampled data using a mixed Fourier/Radon transform: 66th Ann. Internat. Mtg., Soc. Exp. Geophys., Expanded Abstract, 1438–1441.

Soubaras, R., 1997, Spatial interpolation of aliased seismic data: 67th Ann. Internat. Mtg., Soc. Exp. Geophys., Expanded Abstracts, 1167–1190.

Spitz, S., 1989, Trace interpolation beyond aliasing and without picking: 59th Ann. Internat. Mtg., Soc. Exp. Geophys., Expanded Abstracts, 1121–1122.

——— 1991, Seismic trace interpolation in the f-x domain: Geophysics, **56**, 785–794.

Vermeer, G. J. O., 1990, Seismic wavefield sampling: Soc. Expl. Geophys.

APPENDIX A

SIMILARITY OF GFKI, UFKI AND f-x PREDICTION INTERPOLATORS

The f-x prediction filter interpolator of Spitz (1989, 1991) can be expressed in the f-k domain. To do this, I assume that the operator and its components are all zero padded up to the Fourier transform length, N. If $\mathbf{e^o}$ and $\mathbf{e^e}$ represent odd and even indices of the prediction error filter, $\mathbf{e}$, that annihilates known, $\mathbf{k}$, and unknown, $\mathbf{u}$, data together at a given frequency index, f, then

$$\begin{aligned} \mathbf{e^e} * \mathbf{u} &= -\mathbf{e^o} * \mathbf{k} \\ \mathbf{e^o} * \mathbf{u} &= -\mathbf{e^e_{+1}} * \mathbf{k}, \end{aligned} \tag{A-1}$$

where $+1$ represents the one sample advanced version. In f-x prediction filter interpolation, this system of equations are solved, in the least squares sense, for the unknown data samples, $\mathbf{u}$, given $\mathbf{e^e}$, $\mathbf{e^o}$, and the known components, $\mathbf{k}$, of data (Spitz, 1991). This is because splitting the convolution equation

$$\begin{aligned} &\left(e^e_0, e^o_0, e^e_1, e^o_1, e^e_2, e^o_2, \ldots, e^e_{N-1}, e^o_{N-1}\right) \\ &\quad * (k_0, u_0, k_1, u_1, k_2, u_2, \ldots, k_{N-1}, u_{N-1}) = 0 \end{aligned} \tag{A-2}$$

into odd and even parts leads to

$$\begin{aligned} &\left(e^e_0, e^e_1, e^e_2, \ldots, e^e_{N-1}\right) * (u_0, u_1, u_2, \ldots, u_{N-1}) \\ &\quad = -\left(e^o_0, e^o_1, e^o_2, \ldots, e^o_{N-1}\right) * (k_0, k_1, k_2, \ldots, k_{N-1}) \\ &\left(e^o_0, e^o_1, e^o_2, \ldots, e^o_{N-1}\right) * (u_0, u_1, u_2, \ldots, u_{N-1}) \\ &\quad = -\left(e^e_0, e^e_1, e^e_2, \ldots, e^e_{N-1}\right)_{+1} * (k_0, k_1, k_2, \ldots, k_{N-1}). \end{aligned} \tag{A-3}$$

Now consider only the first line in equation (A-3) and write this equation in the Fourier transform domain. Keep in mind that the prediction error filters are derived at half the original frequency while the data are at original frequency. Then, the f-k transform of the unknown data, $U(f, k)$, can be expressed in terms of the f-k transform of the known data, $K(f, k)$, and the f-k transform of the even and odd operator components:

$$U(f, k) = \left[-E^o(f', k')/E^e(f', k')\right] K(f, k). \tag{A-4}$$

(A different notation for the frequency index, f', is used for the prediction error filter to alert the reader to the fact that this frequency index represents lower frequencies, although in reality $f' = f$ since these symbols represent indices rather than actual frequencies.) The same comment applies to k', that is, $k' = k$, but it corresponds to lower wavenumbers because every other sample of the operator means coarser sample increment along the space direction.

Also, consider the following:

1) The FFTs of the odd and even components of the data relate to the FFT of the full data:

$$A^e(f, k') = (1/2)(A(f, k) + A(f, k + N/2))$$

$$A^o(f, k') = (1/2)e^{j(2\pi/N)k}(A(f, k) - A(f, k + N/2)). \quad \text{(A-5)}$$

2) The prediction error filter is obtained from the original data at lower frequencies (by a factor of 2) after zero padding in time, (and at the original wavenumbers). It is a whitening filter for the data:

$$E(f', k) = 1/D(f', k), \quad \text{(A-6)}$$

where $D(f', k)$ represents the Fourier transform of the data after padding the traces with zeroes to double the data length along the time axis. That is, $D(f', k)$ is a stretched, low-frequency version of $K(f, k)$.

3) A simple identity,

$$\frac{\frac{1}{a} - \frac{1}{b}}{\frac{1}{a} + \frac{1}{b}} = \frac{b - a}{b + a}.$$

One can then convert equation (A-4) into

$$U(f, k) = e^{j(2\pi/N)k}[(D(f', k) - D(f', k + N/2))$$

$$/(D(f', k) + D(f', k + N/2))]\,K(f, k). \quad \text{(A-7)}$$

Applying equation (A-5) to $D(f', k)$ yields

$$U(f, k) = [D^o(f', k')/D^e(f', k')]\,K(f, k). \quad \text{(A-8)}$$

That is, the Fourier transform of unknown data can be obtained by multiplying the Fourier transform of the known data by a function that is the ratio of the Fourier transforms of the odd and even indexed traces of known data calculated at half temporal frequency. Due to doubling of the trace distance by selection of odd or even traces, physical wavenumbers used in the operator are half of the wavenumbers in data (but they are "stretched").

Equation (A-8) is the UFKI interpolation (Gülünay and Chambers, 1996, 1997a). Since UFKI and GFKI are related (Gülünay and Chambers, 1997b) the *f-x* interpolator becomes similar to the GFKI operator as well. Indeed, the traces used in derivation of the *f-x* domain "half-step prediction filters" by Porsani (1999) for an interpolation factor equal to 2 are the same as the odd and even numbered traces used in the UFKI method. Porsani's filters correspond to the UFKI filters but are designed and applied in the *f-x* domain.

Note, however, that there are differences between *f-x* and *f-k* interpolators. First, the *f-x* interpolator is a least-squares interpolator. Interpolated data are forced to fit the known data in the least-squares sense. This is not so in the *f-k* interpolation. Second, there is a user-defined parameter, the length of the prediction filter, which corresponds to the number of linear events assumed to exist in the input data in the *f-x* interpolation method. The length of this filter is short, and the filter calculation can be relatively fast, especially for 2D data where a Toeplitz structure can be used for minimum phase-type prediction-error filters. First, the filter is solved (at low frequencies), then the coefficients of this filter are used to form a set of linear equations at the original frequencies to solve for the unknown data samples. This part is generally the most time consuming part of the *f-x* interpolation (especially in 3D) since the number of unknown data points is generally much larger than the filter length. GFKI (and UFKI) turn the linear equation solving problem into a complex number division in the Fourier transform domain.

The number of events present in field data is generally not known. Applying an NMO and splitting data into small time-space windows reduces data complexity and therefore the size of the required prediction filters. Even then, there is a chance of specifying filters that are too short and therefore the need to specify a filter length may be considered a negative point for the *f-x* interpolation method. This, however, has not been a problem in practice. Furthermore, specifying a finite length for the filter is analogous to smoothing the *f-k* spectrum. Smoothing the *f-k* operator explicitly and spatially, and temporally tapering the *t-x* domain version of the operators in both GFKI and UFKI methods are then similar to using a short prediction filter in *f-x* interpolation.

Indeed, all three interpolators produce similar results with field data. With synthetics using small space gates, *f-x* interpolation results are cleaner than *f-k* interpolation results because small windows blur the Fourier transform. Blurring might cause the *f-k* interpolator to be nonzero at certain points in the *f-k* domain where it is supposed to be zero.