Seismic data interpolation using a fast generalized Fourier transform

Mostafa Naghizadeh and Kris Innanen

ABSTRACT

We propose a fast and efficient method for interpolation of nonstationary seismic data. The proposed method utilizes fast generalized Fourier transform (FGFT) to identify the space-wavenumber evolution of nonstationary spatial signals at each temporal frequency. Next, a least-squares fitting scheme is used to retrieve the optimal FGFT coefficients representative of the ideal interpolated data. For randomly sampled data on a regular grid we seek a sparse representation of FGFT coefficients in order to retrieve the missing samples. Also, to interpolate the regularly sampled seismic data at a given frequency, we use a mask function derived from the FGFT coefficients of the low frequencies. Synthetic and real data examples are provided to examine the performance of the proposed method.

INTRODUCTION

The problems of seismic data reconstruction and interpolation have attained a special stature in the seismic data processing community in recent years. Reconstruction methods use available seismic traces, measured on irregular and/or coarsely sampled grids in space, to estimate data on a regularly and sufficiently sampled grid. An effective solution can open the door to application of multidimensional wave equation imaging and de-multiple algorithms to a data set, without having had to acquire it with the completeness these methods demand. In a useful method of interpolation/reconstruction we look for speed, stability in the presence of noise and aliasing, and ability to manage complex events. In this paper we propose an interpolation/reconstruction methodology which provides a constructive mixture of the above properties, and is also a theoretically sound framework within which to carry out a range of wave based data analysis tasks.

Reconstruction methods can be divided into two main classes: wave equation based and signal processing based. Most methods in the latter category, including the one we describe in this paper, utilize transform domains such as Fourier (Duijndam et al., 1999; Liu, 2004; Abma and Kabir, 2005; Zwartjes and Gisolf, 2006), Radon (Darche, 1990; Trad et al., 2002), and curvelet (Hennenfent and Herrmann, 2008). A key issue in judging the performance of transform-based reconstruction methods and their management of complex events lies in their response to data nonstationarity.

In seismic data processing, nonstationarity means the frequency/wavenumber content of the signal varies in time/space. For instance, an absorptive medium causes nonstationarity in the time dimension by making the frequency content of a seismic pulse a function of path length. Also, seismic sections which contain hyperbolic and parabolic (or any nonlinear) events produce nonstationary spatial signals in the *f-x* domain at a given frequency. Interpolation/reconstruction methods typically cope with nonstationary signals through spatial windowing. Inside sufficiently small spatial windows nonlinear seismic events appear linear or stationary. Hence, methods which assume stationarity such as those

referenced above may be applied. Naghizadeh and Sacchi (2009) have proposed an alternative method providing a beyond-alias interpolation of nonstationary seismic data. It is essentially a modification of the *f-x* interpolation of Spitz (1991). In their method the windowing of the spatial axis is avoided through use of adaptive prediction filters. Although spatial windowing of data and the adaptive *f-x* interpolation method are capable of handling nonstationarity, they lead to computationally demanding algorithms.

On another note, there exists a range of transformations specifically designed to deal with identification and analysis of nonstationary behavior in signals that might be used as a basis for interpolation. The Gabor transform, for instance, is a class of short-time Fourier transforms that has been used for a range of processing tasks such as nonstationary deconvolution (Margrave et al., 2003). Wavelet (Beylkin et al., 1991), curvelet (Candes et al., 2005; Candes and Donoho, 2004), and S-transform, are further examples. The S-transform (Stockwell et al., 1996) is a type of short-time Fourier transform in which the window size is frequency dependent. Larger windows are used for lower frequencies and smaller windows are used for higher frequencies. Consequently, the spectrum at a given frequency is estimated with a number of samples appropriate to that frequency. Two properties of the aforementioned transforms work against their potential as a base for an interpolation method. They are redundant (the size of the transformed data set is larger than that of the original data set) and computationally demanding. Hence, when applied to already large seismic data volumes, they can cause unwanted data expansion. These transforms also share the computational issues associated with windowing/adaptive methods, and therefore, on their own, do not constitute a viable avenue to respond to them.

However, recent work on the S-transform has led to a fast, non-redundant algorithm (Brown et al., 2010), renewing the possibility of developing an efficient and effective interpolation/reconstruction approach based on S-transform theory. In this paper, we develop such an approach and examine its behavior when applied to synthetic and field data sets. First, because the transform algorithm is new, we will describe a straightforward and intuitive frequency domain computation of the fast generalized Fourier transform (FGFT). We will then combine FGFT with a least-squares fitting principle to formulate our FGFT Interpolation method. Synthetic and field data examples are examined.

BACKGROUND THEORY

Generalized Fourier transform

The S-transform of a time signal $g(t)$ is defined as follows (Stockwell et al., 1996):

$$S(\tau, f) = \int_{-\infty}^{\infty} g(t) \frac{|f|}{\sqrt{2\pi}} e^{-\frac{(\tau-t)^2 f^2}{2}} e^{-i2\pi f t} dt, \tag{1}$$

where τ and f are the time and frequency coordinates, respectively. The term $\frac{|f|}{\sqrt{2\pi}} e^{-\frac{(\tau-t)^2 f^2}{2}}$ is a Gaussian window which depends on time lag and frequency. The width of the Gaussian window decreases with increasing frequency. This results in finer frequency resolution for low frequencies and finer time resolution for high frequencies. If the window function is

set to unity, equation 1 reverts to the ordinary Fourier transform. The S-transform can be generalized by replacing the Gaussian window with other functions, such as Gabor and B-Spline windows. Brown et al. (2010) have used the term "general Fourier-family transform" to refer to this entire group.

Equation 1 is a formula for calculation of the S-transform of an input signal $g(t)$ in the time domain. The same calculation can be carried out in the frequency domain. In fact, the S-transform in the Fourier domain turns out to be simpler to derive and easier to implement than its time domain counterpart. Implementation of the frequency domain also provides a framework for the FGFT method. The equivalence of the time and frequency domain S-transforms, as well as the derivation of the method and algorithm details, are presented by Brown et al. (2010). Since this work is outside of typical geophysical literature, we begin with a brief overview of the frequency-domain S-transform algorithm, exemplifying it with the chirp function illustrated in Figure 1a.

The procedure of S-transform in the frequency domain is as follows

1. Transform $g(t)$ to the Fourier domain to obtain $G(f)$. Figures 1a and 1b show the original chirp function and its Fourier domain representation, respectively.

2. Create a data matrix, $G(f, f')$, using repeated instances of $G(f)$, each time shifting its elements (Figure 1c). This matrix is referred to as the data in the α-domain, where f' represents the frequency shift. To be specific, the column of data at frequency shift zero is the original Fourier domain representation of the data in Figure 1b. The other columns are shifted copies of Figure 1b.

3. Create a window matrix, $W(f, f')$, the same size as the α-domain representation of the data (Figure 1d). The window function can be any symmetric smooth function such as Gaussian, Hanning, B-spline, etc. The size of windows grows linearly from lower to higher frequencies. The windows are chosen to be wider for high frequencies and narrower for low frequencies to gain proper resolution in time-frequency analysis (Brown et al., 2010).

4. Multiply $G(f, f')$ and $W(f, f')$ element by element to window the data in the α-domain, obtaining $G'(f, f')$ (Figure 1e).

5. Apply a 1D inverse Fourier transform along each row (i.e., the f' axis) of $G'(f, f')$ to obtain the S-Transform of the original data (Figure 1f).

The chirp function in Figure 1a has more low frequency content at the start of the time signal and more high frequency content at the end. The S-transform of the chirp function in Figure 1f reflects this evolution with local spectra trending toward high frequency as time increases.

A natural outcome of deploying wider/narrower windows to estimate high/low frequencies in the frequency-domain is a decreased/increased resolution at correspondent frequencies. This is illustrated in Figure 2, in which we apply the S-transform to a time signal

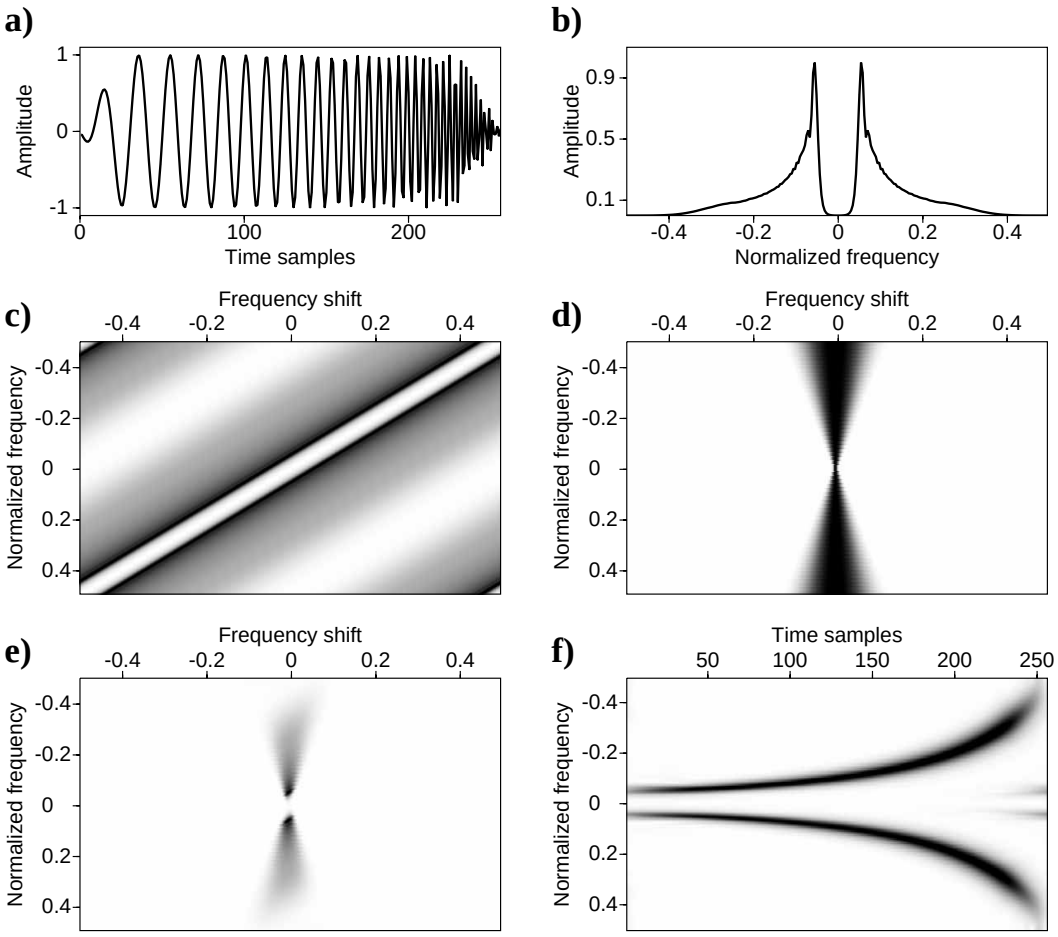

FIG. 1. Frequency domain implementing of S-Transform. a) Original chirp function. b) Fourier transform of (a). c) α-domain matrix built by shifting spectrum in (b). d) Window function with the same size as the α-domain representation of the data in (c). e) Windowed α-domain obtained by element by element multiplication of (c) and (d). f) S-transform of data obtained by applying inverse Fourier transform on each row of (e).

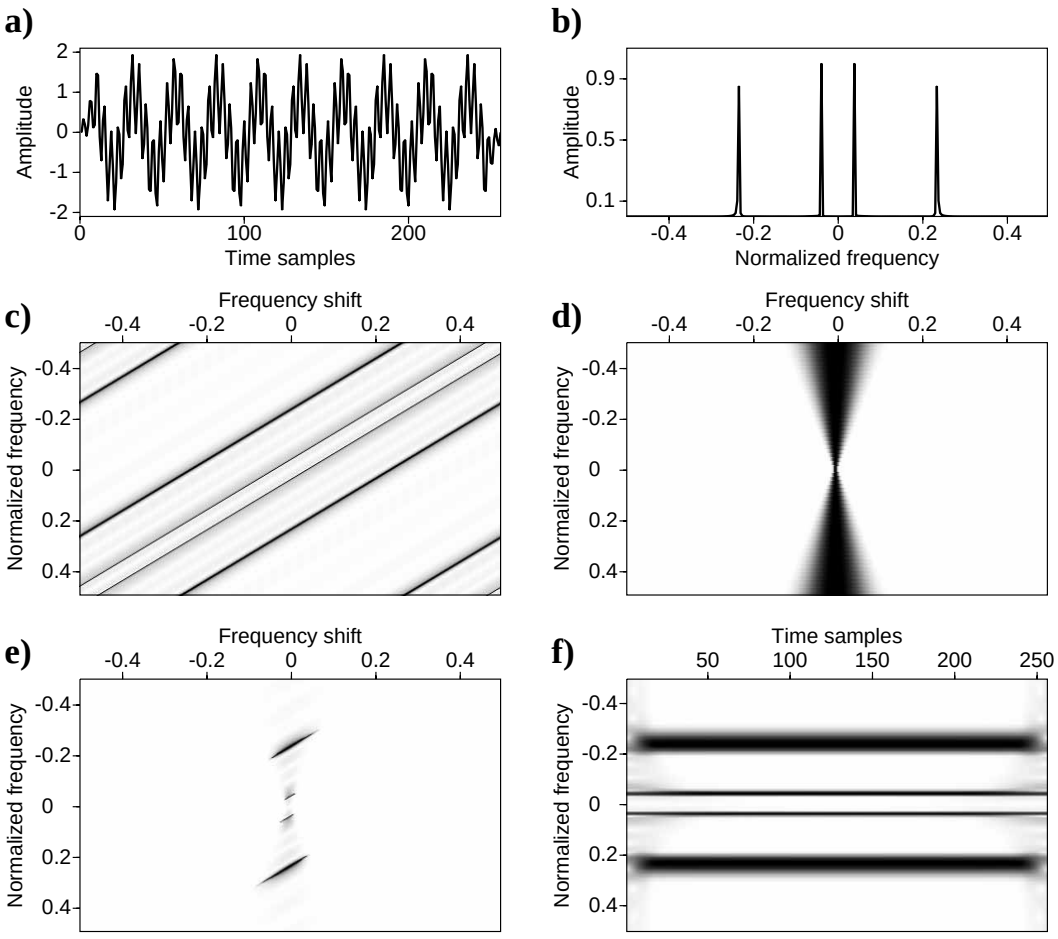

FIG. 2. S-Transform of a signal composed of two sine functions. a) Original function. b) Fourier transform of (a). c) α-domain matrix built by shifting spectrum in (b). d) Window function with the same size as the α-domain representation of the data in (c). e) Windowed α-domain obtained by element by element multiplication of (c) and (d). f) S-transform of data obtained by applying inverse Fourier transform on each row of (e).

composed of two stationary harmonics. Figures 2a and b show a time series composed of the sum of two sinusoids, and its amplitude spectrum, respectively. This signal is transformed to the α-domain (Figure 2c), in which a window function is designed (Figure 2d) and applied to the α-domain signal (Figure 2e). The S-transform of the signal, shown in Figure 2f, clearly represents the two harmonics, constant with respect to time. However, the low frequency harmonic is more highly resolved in the S-transform domain (narrow line) than the high frequency harmonic (wider line). This is an intrinsic property of the S-Transform that can be utilized to build a faster transform algorithm, which we will discuss next.

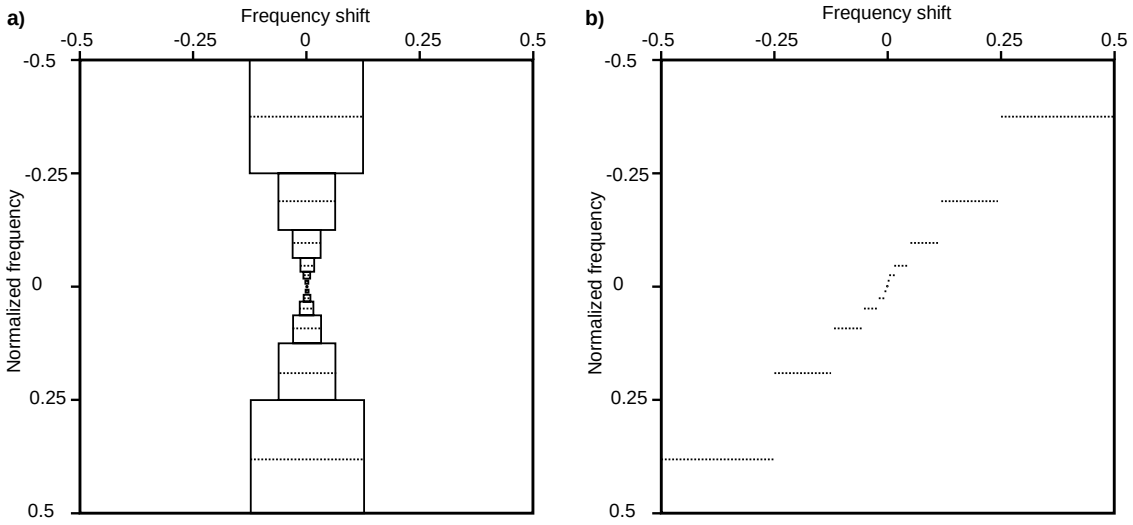

FIG. 3. a) Dyadic segmentation of α-domain. b) Unshifted dyadic segments. If each dashed line segment in (b) is dropped to zero normalized frequency, the data exactly reproduces the original frequency vector of data.

Fast generalized Fourier transform (FGFT)

The frequency-dependence of the resolution of the S-transform suggests that computational efficiency can be increased by adopting sampling criteria varying from high to low frequency. Brown et al. (2010) used a dyadic segmentation approach, through which high frequencies (which are inherently low resolution) are coarsely sampled and low frequencies (which are inherently high resolution) are finely sampled in the α-domain. Figure 3a illustrates the dyadic segmentation of the α-domain. The domain is subdivided into squares (solid lines) centered around zero frequency-shift. The squares are larger for high frequencies and smaller for low frequencies. Within each segment, the center frequency (dashed line) is representative of all the segment's frequencies. The S-transform of the data is then obtained by applying inverse Fourier transform on the data in the dashed lines only. This new segmentation consequently leads to significantly increased efficiency, compared to the full S-transform. The cost is that the inverse Fourier transform on the center frequency (dashed lines) must meaningfully represent the signal's behavior over all the frequencies, in a given square segment of the data. Brown et al. (2010) called this transform as Fast generalized Fourier transform (FGFT). Notice that it is also possible to use other optional forms of segmentation instead of dyadic segmentation.

To finalize the form of the FGFT algorithm, recall that the α-domain was a shifted matrix version of the Fourier transform of the time signal. Unshifting the dashed lines illustrated in Figure 3a recovers their respective locations on the frequency axis (Figure 3b). This implies that the S-transform is equivalent to the application of inverse Fourier transforms, over windows of varying size on the original signal in the Fourier domain. We must then apply short windows at low frequencies and large windows at high frequencies. The dyadic segmentation achieves this optimal segmentation non-redundantly in the Fourier domain. If we imagine dropping each dashed line segment to zero normalized frequency, we see that the vector of utilized data exactly reproduces the original frequency vector. This is the framework for the FGFT algorithm.

Figure 4 illustrates schematically an application of the FGFT algorithm. Figure 4a represents a time signal containing 16 samples. A fast Fourier transform is applied to the time signal to obtain the frequency domain representation illustrated in Figure 4b. In the frequency domain the signal is dyadically segmented (dashed boxes), and within each segment inverse Fourier transforms are applied to the data. The segmentation of Fourier domain into smaller windows follows the rationale that is explained in Figure 3. In this article, we perform the segmentation with the window sizes which are growing by power of 2. However, depending on the application, one can deploy other forms of segmentation. It is also important to mention that each segment needs to be properly tapered on the edges using a tapering window, for instance Gaussian window, to avoid the ringing effects of the Fourier transform. Notice that smaller windows are used for low frequencies and larger windows for high frequencies. The output is illustrated in Figure 4c. Each individual inverse Fourier transform is represented by a particular symbol in Figure 4c (square, diamond, triangle, circle). Assuming a real time signal, we expect a symmetric outcome. So we may then focus our attention on the right hand side of the output, indicated with an underbrace.

Next, to properly represent the time-frequency behavior of the data, the underbraced FGFT coefficients must be arrayed in a 2D plot. Figure 4d illustrates this arrangement of FGFT coefficients. Each individual element of a given inverse Fourier transform is distinguished from the others in that group via size. Hence each FGFT coefficient is uniquely represented by a symbol type and size. Figure 4d illustrates how these outputs are distributed. We note that there is better time resolution in the high frequencies and better frequency resolution in the low frequencies. The inverse or adjoint FGFT is performed by reversing the order of operations from Figures 4a-c, by replacing inverse Fourier transforms with Fourier transforms and vice versa. The algorithms for both forward and adjoint FGFT are described by Brown et al. (2010).

Figure 5a shows the same chirp function illustrated in Figure 1a. Figure 5b shows the FGFT of Figure 5a. Figure 5c illustrates the 2D array of FGFT coefficients after proper up-scaling, i.e., the time-frequency decomposition of the chirp. Because the original chirp input is real, we only include the positive frequencies for this example. The FGFT evidently captures the nonstationary nature of the chirp function. The low frequencies predominate at the beginning of the signal and high frequencies predominate at the end. Figure 5d illustrates the adjoint FGFT acting on the FGFT coefficients of the chirp function. The adjoint FGFT recovers the original data within a small error level produced by the windowing step.

The undesirable windowing error and the desirable time-frequency sensitivity of the FGFT are more pronounced in a signal with sharp discontinuities. For example, Figure 6a illustrates a box function, which contains two regions of relatively dramatic nonstationary behavior. Figures 6b and 6c illustrate the FGFT coefficients and their 2D time-frequency interpretation, respectively. The FGFT coefficients have successfully identified the location of the box function in the time axis. Figure 6d shows the result of applying adjoint FGFT on the coefficients of FGFT.

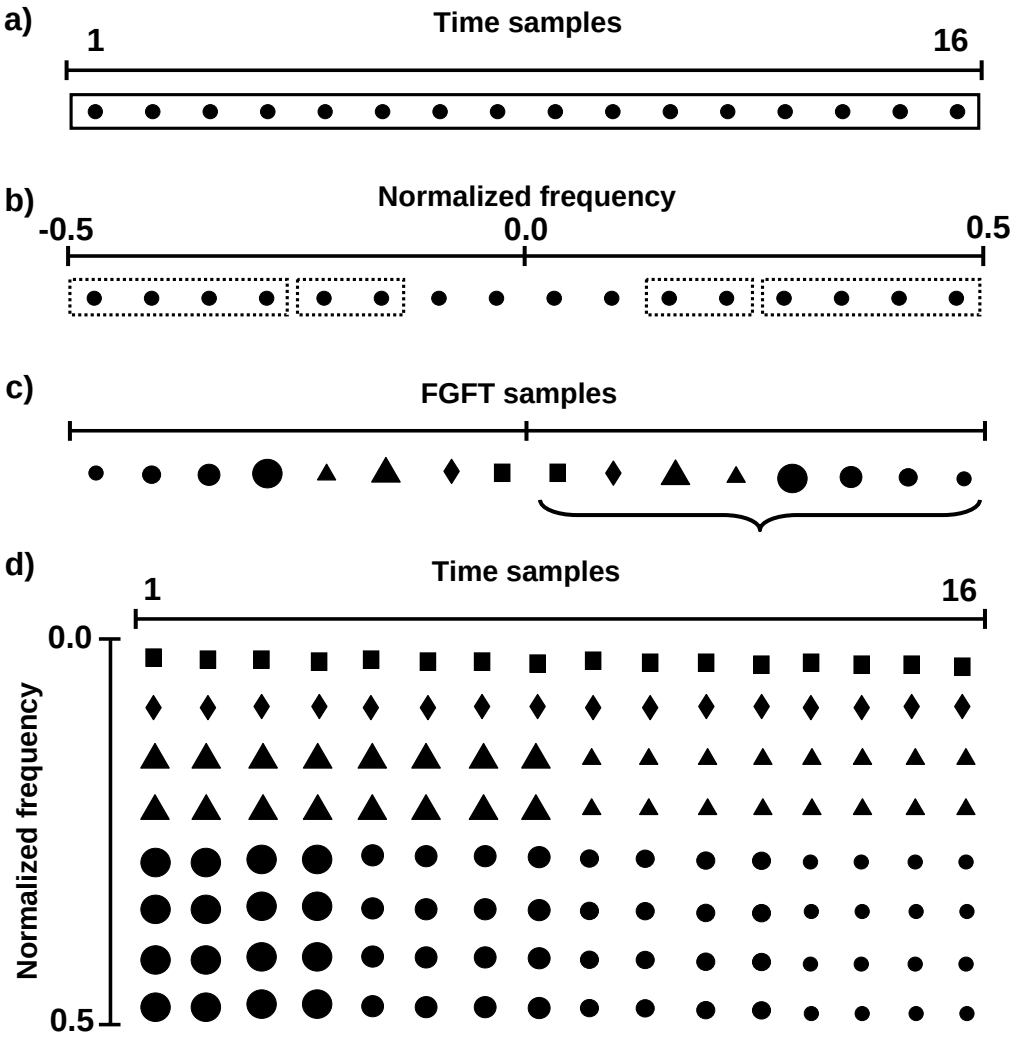

FIG. 4. Graphical representation of implementing FGFT. a) Original signal with 16 time samples. b) Fourier transform of original data in (a). c) FGFT representation of data after applying inverse Fourier transform on each dashed box of data in (b). d) The time-frequency interpretation of FGFT coefficients in (c) only for positive frequencies.

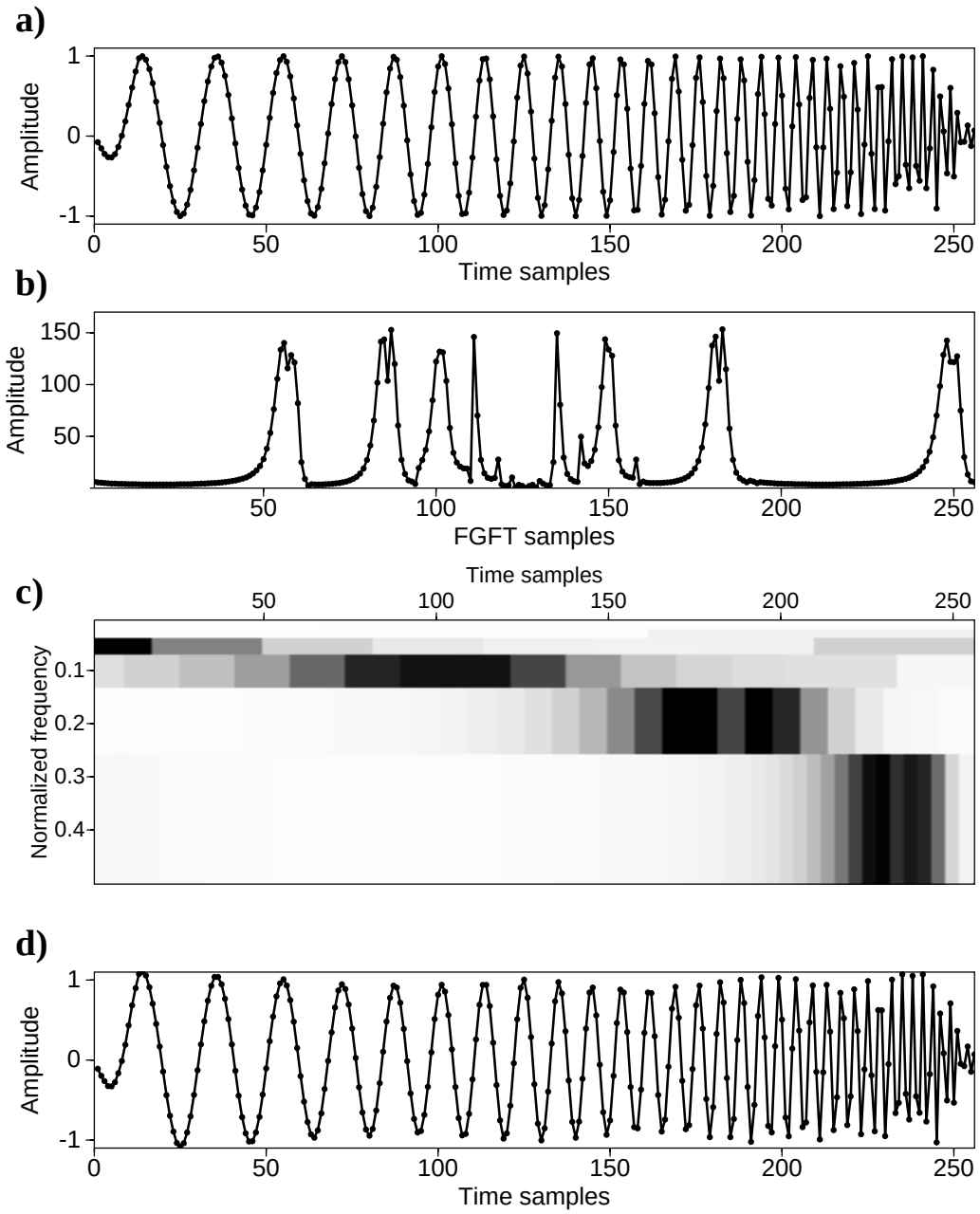

FIG. 5. a) Original chirp signal. b) The FGFT coefficients of (a). c) 2D plot of FGFT coefficients clearly showing the time-frequency distribution of chirp signal. d) The recovered chirp signal by applying inverse FGFT on (b).

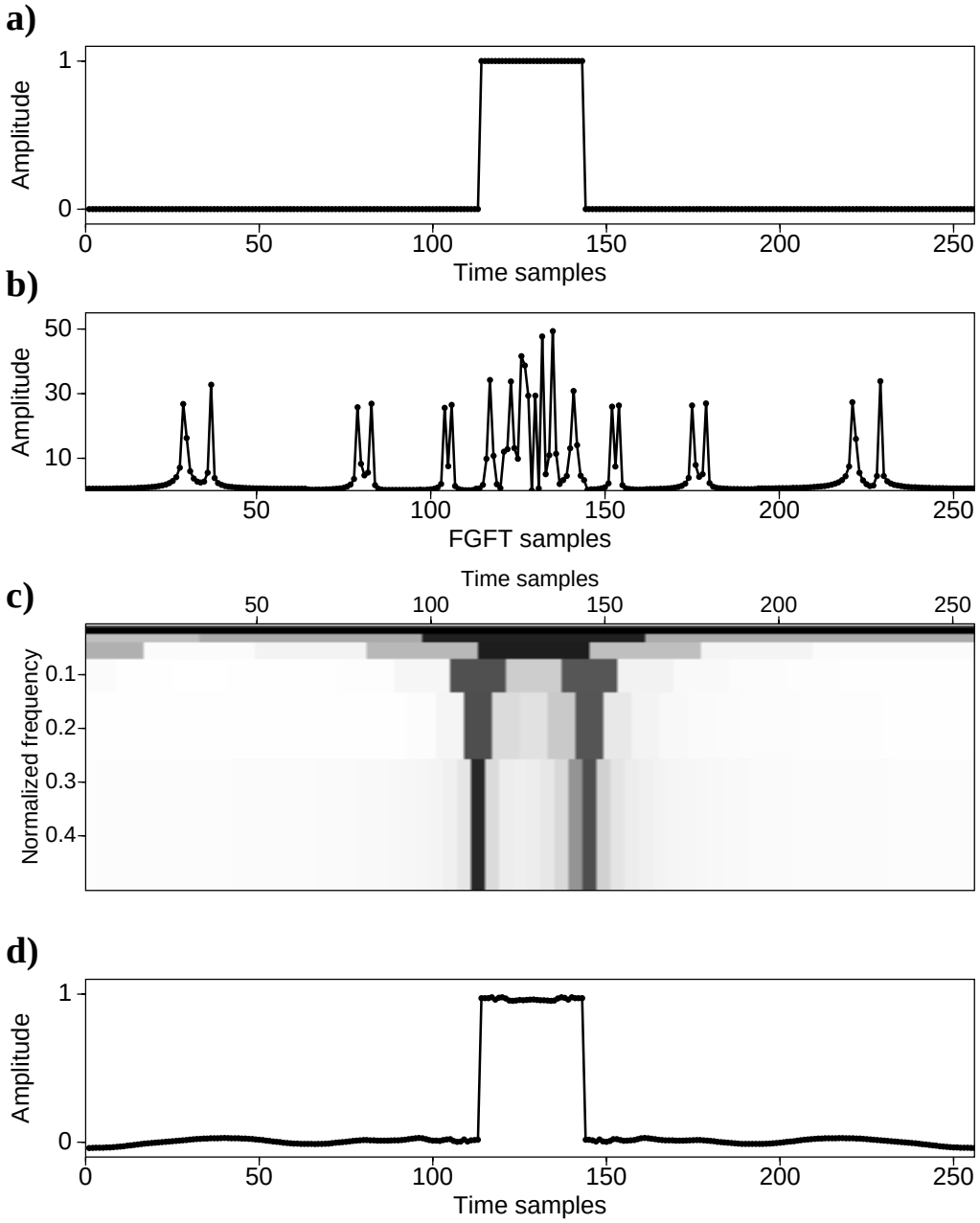

FIG. 6. a) Original box signal. b) The FGFT coefficients of (a). c) 2D plot of FGFT coefficients in (b). d) The recovered signal by applying inverse FGFT on (b).

FGFT INTERPOLATION OF SEISMIC DATA

We now turn our attention to the problem of reconstructing seismic data. We represent the desired regularly sampled nonstationary signal of length N as $\mathbf{d} = (d_1, d_2, \ldots, d_N)^T$. Next we assume that we only have M out of N samples available, and that the remaining $N-M$ samples are missing. We represent the available samples as $\mathbf{d}_{\text{obs}}$. The desired signal $\mathbf{d}$ and observed samples $\mathbf{d}_{\text{obs}}$ are related by a sampling matrix $\mathbf{T}$ (Liu, 2004), through:

$$\mathbf{d}_{\text{obs}} = \mathbf{T}\mathbf{d}\,. \tag{2}$$

We investigate the structure of sampling matrix $\mathbf{T}$ using a simple example. Assume that the desired signal $\mathbf{d}$ has 5 samples but only 3, say $\{d_2, d_4, d_5\}$, are available. Equation 2 can be written as

$$\begin{pmatrix} d_2 \\ d_4 \\ d_5 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \end{pmatrix}. \tag{3}$$

The transpose of the sampling matrix, $\mathbf{T}^T$, correctly places the available samples in the output and puts zeros in the samples corresponding to missing data.

The interpolation problem is under-determined and hence to solve it we require some prior information. To provide this, let us consider the FGFT coefficients $\mathbf{g}$ of the desired, fully sampled signal. These must be related to $\mathbf{d}$ by

$$\mathbf{g} = \mathbf{G}\,\mathbf{d}\,, \tag{4}$$

where $\mathbf{G}$ represents the forward FGFT operator. The adjoint FGFT operator $\mathbf{G}^T$ can furthermore be used to express the desired interpolated data $\mathbf{m}$ in terms of $\mathbf{g}$ as follows

$$\mathbf{d} \approx \mathbf{G}^T\,\mathbf{W}\,\mathbf{g}\,, \tag{5}$$

where we have introduced a diagonal weight function $\mathbf{W}$ that preserves a subset of FGFT coefficients. Inserting equation 5 into 2 yields

$$\mathbf{d}_{\text{obs}} \approx \mathbf{T}\mathbf{G}^T\,\mathbf{W}\,\mathbf{g}\,. \tag{6}$$

Let us assume for the moment that the operator $\mathbf{W}$ is known. The system of equations in equation 6 is under-determined (Menke, 1989) and therefore, it admits an infinite number of solutions. A stable and unique solution can be found by minimizing the following cost function (Tikhonov and Goncharsky, 1987)

$$J \;=\; ||\mathbf{d}_{\text{obs}} - \mathbf{T}\,\mathbf{G}^T\mathbf{W}\mathbf{g}||_2^2 + \mu^2||\mathbf{g}||_2^2\,, \tag{7}$$

where μ is the trade-off parameter. We minimize the cost function J using the method of conjugate gradients (Hestenes and Stiefel, 1952). The conjugate gradients method does not require the explicit knowledge of $\mathbf{G}$ in matrix form. It requires the action of operators $\mathbf{G}^T$ and $\mathbf{G}$ on a vector in the coefficient and data spaces, respectively (Claerbout, 1992). The goal of the proposed algorithm is to find the coefficients $\hat{\mathbf{g}}$ that minimize J, and use them to reconstruct the data via the adjoint FGFT operator $\hat{\mathbf{d}} = \mathbf{G}^T\hat{\mathbf{g}}$.

Derivation of the weight function W

The weight function plays an important role in the FGFT interpolation method. An optimal design for the weight function will depend on the form of the sampling function $\mathbf{T}$.

Case 1: Randomly missing samples on a regular grid

For a randomly sampled signal one can impose a sparsity constraint to the FGFT coefficients. This corresponds to a weight function which diminishes small amplitude coefficients and preserves large ones (Sacchi et al., 1998). The large FGFT coefficients correspond with coherent features in a data set. By suppressing the small FGFT coefficients, the coherent data features will be reintroduced across regions where data was initially missing.

We adapt an Iteratively Re-weighted Least Squares (Scales and Gersztenkorn, 1988) approach to achieve this purpose. The algorithm is summarized as follow

$$\begin{aligned}
&\text{Initialization } \mathbf{W}^0 = \mathbf{I} \\
&\text{For } k = 1, 2, 3 \ldots \\
&\quad \hat{\mathbf{g}}^k = \underset{\mathbf{g}}{\operatorname{argmin}}\{||\mathbf{d}_{\text{obs}} - \mathbf{T}\,\mathbf{G}^T\mathbf{W}^{k-1}\mathbf{g}||_2^2 + \mu^2||\mathbf{g}||_2^2\} \\
&\quad \mathbf{W}^k = \operatorname{diag}(\operatorname{abs}(\hat{\mathbf{g}}^k)) \\
&\text{End}
\end{aligned} \tag{8}$$

where $\operatorname{diag}(\operatorname{abs}(.))$ builds a diagonal matrix from the absolute values of a vector. The weight function was initiated with an identity matrix and was updated by the absolute value of FGFT coefficients after each least-squares fitting. Our tests show that using a small number of internal iteration for conjugate gradients (3 or 4) and more iterations of external re-weighting, produces optimal results. Using more internal iterations for conjugate gradients could lead to spurious high frequency features.

Case 2: Regularly missing samples

For a signal with regularly missing samples (zeros in the location of missing samples), a different strategy is needed. Regular sampling leaves large amplitude artifacts in any Fourier domain representation of data (Naghizadeh and Sacchi, 2010). In the S-transform domain, the high amplitude artifacts appear in high frequencies at all times. This means that a sparse constraint in the S-transform domain will tend to poorly interpolate the high frequencies (such artifacts may also appear, though generally to a lesser extent, in random sampling cases). In order to avoid this pitfall and correctly interpolate regularly missing

samples using the FGFT framework, we must be able to predict the location of desirable vs. undesirable coefficients. The weight function can then be a simple mask function which has the value of 1 for the coefficients in the desirable locations and zero elsewhere. For 1D signals, making such a prediction would be difficult, likely impossible. However, for multidimensional seismic data one can use the half frequency of the original data to obtain proper weight function for a desired frequency in the interpolated data (Spitz, 1991). We will pursue this further in the numerical examples to follow.

EXAMPLES

1D chirp function

Randomly missing samples

We begin by applying FGFT interpolation on a 1D chirp function. Figure 7a shows the original chirp function, illustrated in Figure 5a, after randomly eliminating 50% of its samples. Figure 7b depicts the FGFT domain representation of the signal in Figure 7a. Comparison of the FGFT of the original chirp function (Figure 5b) with the FGFT of the randomly decimated chirp function (Figure 7b) illustrates the introduction of high amplitude artifacts, especially in the high frequencies. Figure 7c shows the FGFT of reconstructed data after applying equation 8. The reconstructed chirp function in Figure 7d is obtained by applying inverse FGFT to the recovered coefficients in Figure 7c. Notice that the reconstruction algorithm has been able to recover the nonstationary signature of the original chirp function.

Regularly missing samples

Next, we examine the performance of FGFT interpolation on a regularly decimated version of the same chirp function. Figure 8a illustrates the regularly decimated input. Figure 8b illustrates the FGFT domain representation of the data in Figure 8a. The regular decimation of the chirp function has created a series of high amplitude artifacts in all of the time samples in the high frequency range. Figure 8c illustrates the data in the FGFT domain after applying an un-modified version of equation 8. The sparsity constraint imposed by Equation 8 has evidently retained undesirable time sample locations at high frequencies. Figure 8d shows the reconstructed chirp function after applying an inverse FGFT to the recovered coefficients in Figure 8c.

Since this is a 1D signal, we would normally be unable to interpolate the data without knowing in advance what the original signal was. Let us nevertheless demonstrate the improvement that a well-designed mask can have on the results in Figure 8. We create the mask function, pictured in Figure 9a, using the FGFT domain representation of the original, fully sampled, chirp function in Figure 5b. We give the mask value of one to the 25% highest FGFT amplitudes and set the remaining 75% to zero. This mask function is used in equation 7 to retrieve the FGFT coefficients in Figure 9b. The reconstructed chirp function is shown in Figure 9c. Evidently, in the presence of a priori information pointing us preferentially to certain regions of the FGFT domain, we may expect dramatic improvement in interpolation of a regularly decimated signal.

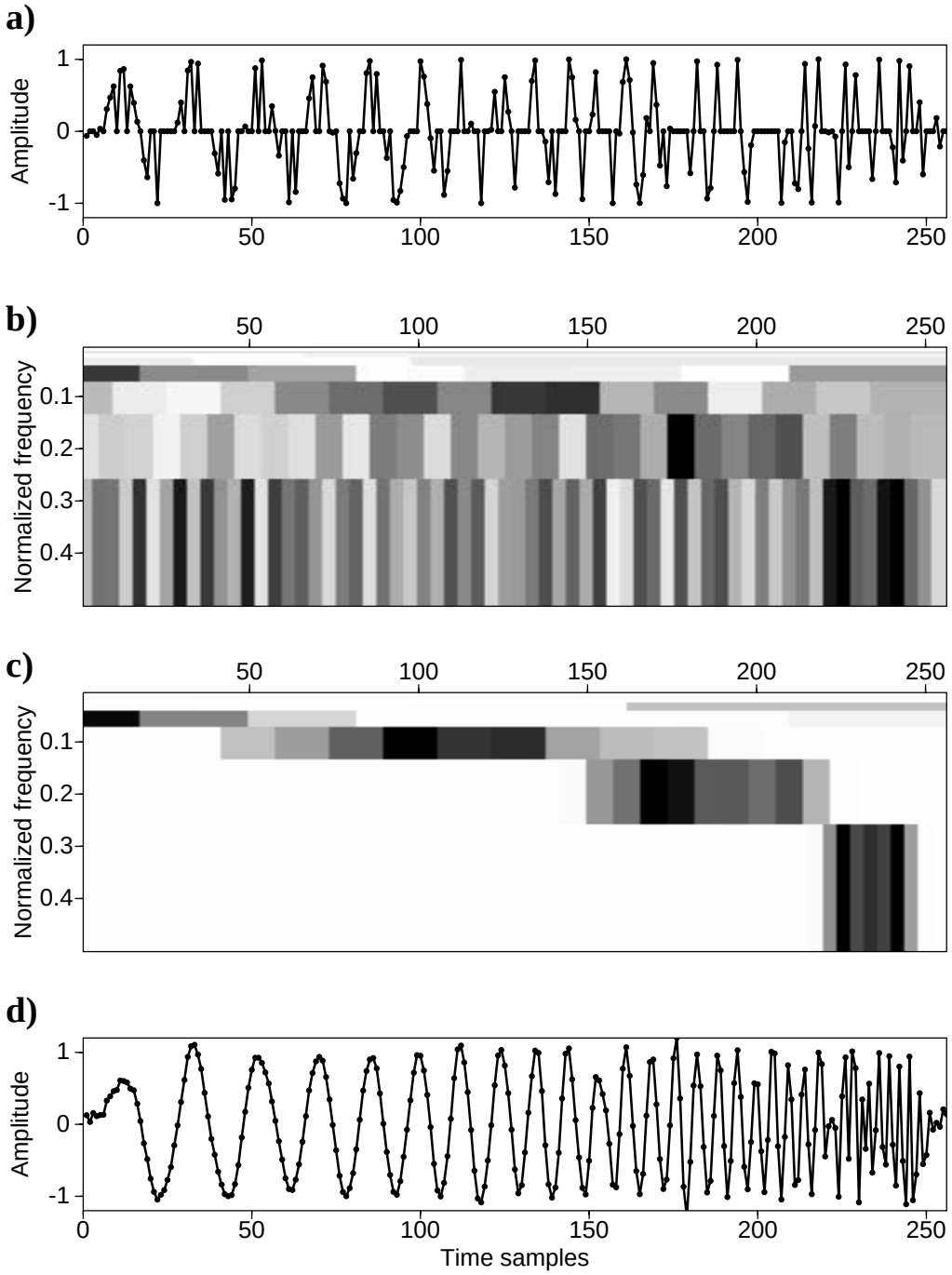

FIG. 7. a) randomly sampled chirp function by randomly replacing 50% of original chirp function in Figure 5a with zeros. b) 2D plot of FGFT coefficients of (a). c) Retrieved FGFT coefficient using algorithm 8. d) The reconstructed chirp signal using the FGFT interpolation.

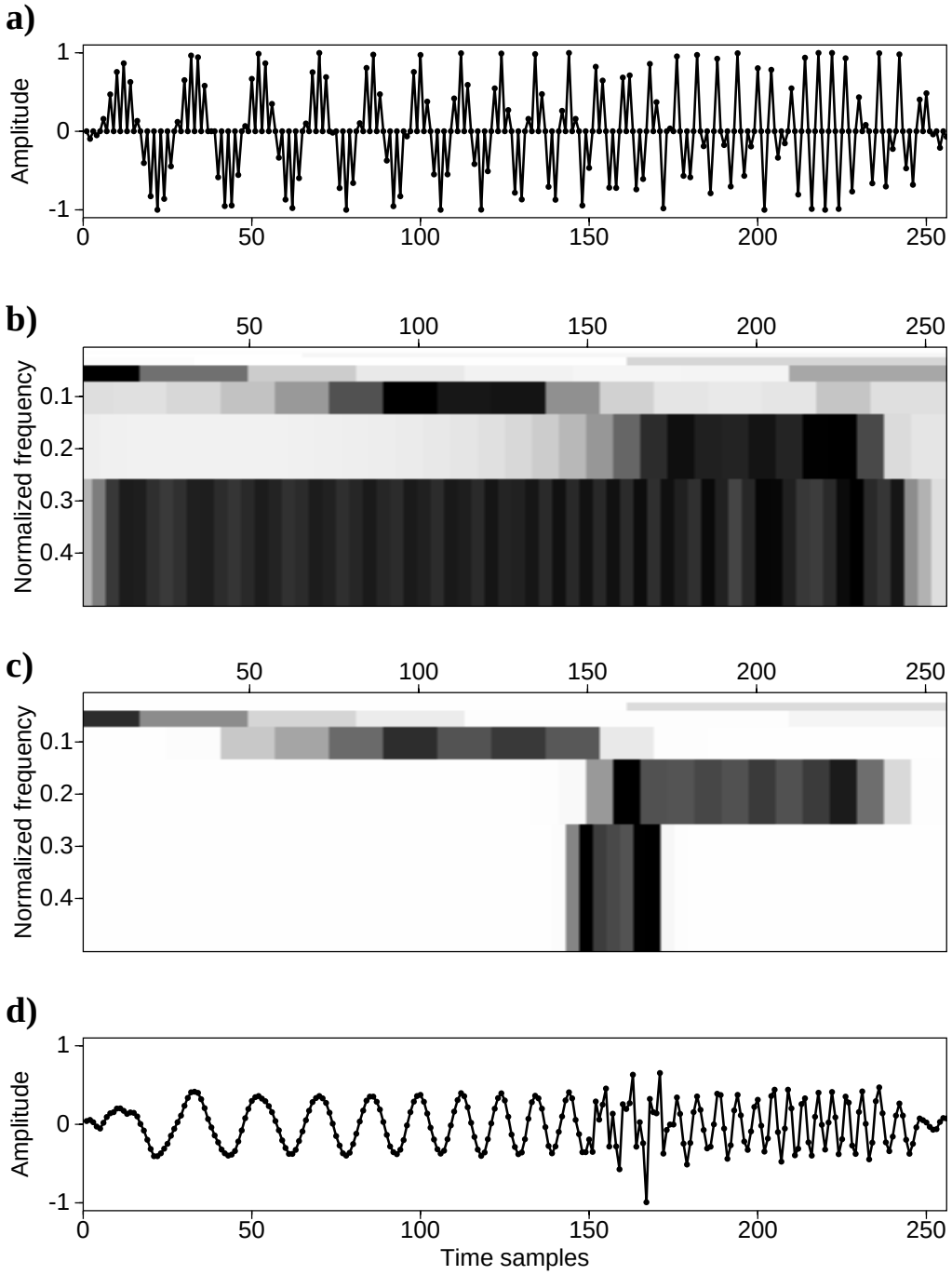

FIG. 8. a) Regularly decimated chirp function by replacing odd number samples of original chirp function in Figure 5a with zeros. b) 2D plot of FGFT coefficients of (a). c) Retrieved FGFT coefficient using algorithm 8. Reconstructed chirp function by applying inverse FGFT to (c).

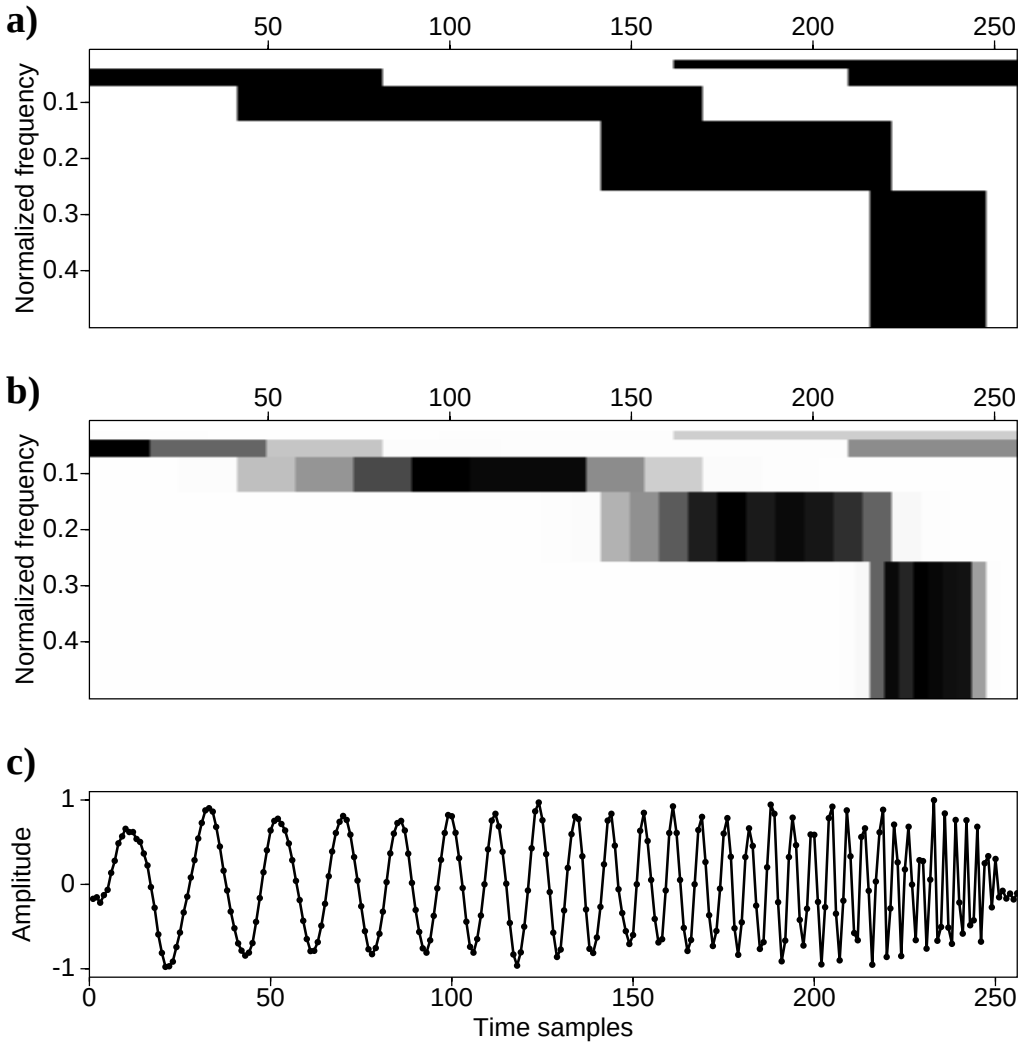

FIG. 9. a) The mask function derived from the FGFT coefficients of original chirp function (Figure 5c) by setting the 25% highest values equal to 1 and the rest equal to 0. b) Retrieved FGFT coefficients of regularly decimated chirp function (Figure 8b) using the mask function in (a). c) Reconstructed chirp function by applying inverse FGFT to (b).

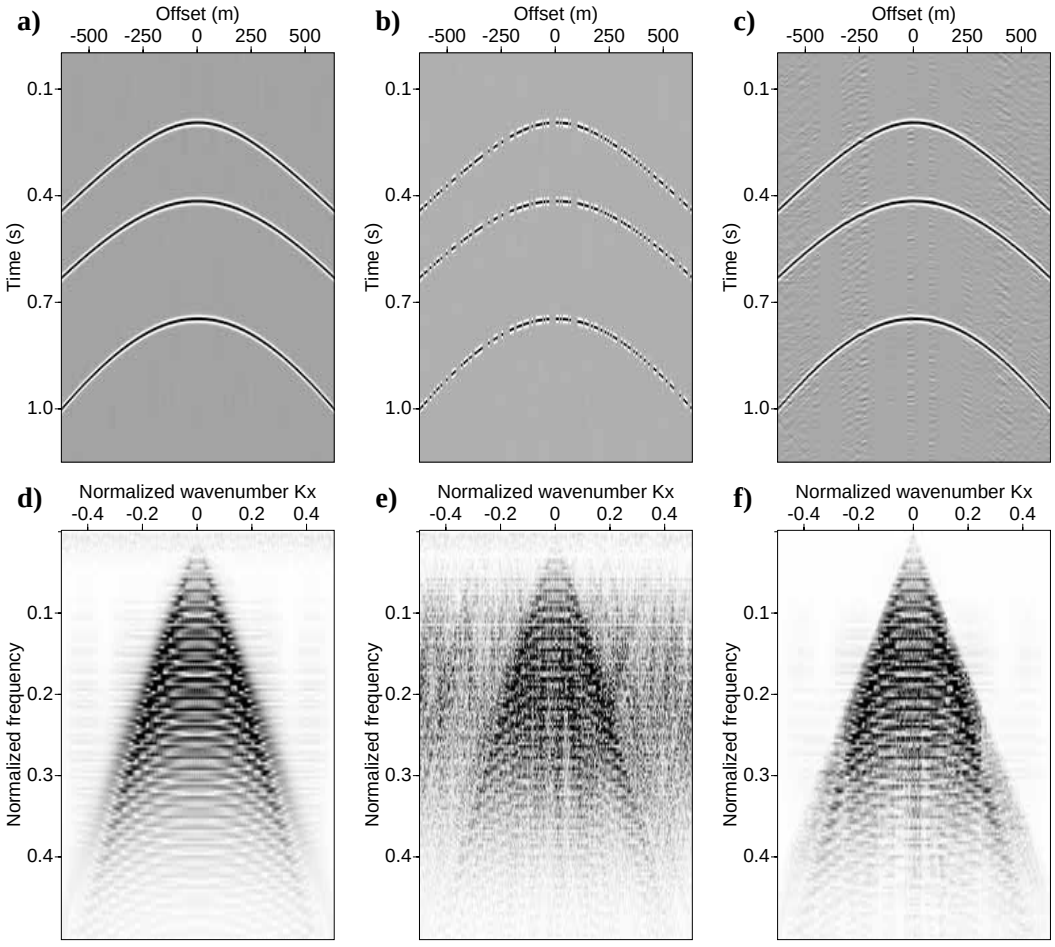

FIG. 10. a) Original synthetic seismic section with 3 hyperbolic events. b) The seismic section with 50% randomly missing traces. c) Reconstructed data using the FGFT interpolation method. d)-f) are the *f-k* representations of (a)-(c).

Synthetic seismic data

Randomly missing samples

For randomly sampled seismic sections each frequency can be reconstructed independently. This is because random sampling creates low amplitude artifacts in the Fourier domain which can be eliminated by sparsity constraints. The amplification of high values and diminishment of low values in the frequency domain occurs within equation 8. Figure 10a shows an original synthetic seismic section with three hyperbolic events composed of 202 traces. Figure 10b shows the section of missing data after randomly eliminating 50% of the traces. Figure 10c shows the FGFT reconstructed data after applying Equation 8 for each single frequency. Figures 10d, 10e, and 10f show the *f-k* spectra of Figures 10a, 10b, and 10c, respectively.

Regularly missing samples

We next demonstrate how the half frequency information in multidimensional seismic data, 2D in this case, can provide weights necessary to interpolate data with regularly missing samples. Merging the basic arguments of Spitz (1991) and Naghizadeh and Sacchi (2009) with the FGFT methodology, a procedure for interpolating regularly sampled non-stationary seismic data emerges:

1. Transform the original data from *t-x* to *f-x* domain.
2. Compute the FGFT of the data at a given frequency, $d(f)$, along all spatial axes, to obtain $g(f)$.
3. Create the weight function $\mathbf{W}(f)$ to have a value one for coefficients larger than a threshold value and zero elsewhere.
4. For the frequency $f' = 2f$, interleave zero values between each pair of available spatial samples to obtain $d_{\text{dec}}(f')$.
5. Upscale the weight function $\mathbf{W}(f)$ to fit the size of $d_{\text{dec}}(f')$ and create the new weight function $\mathbf{W}'(f')$. The upscaling operator is a simple nearest neighbor interpolation scheme or in other words it creates a copy of each sample beside it. Notice that the size of $\mathbf{W}'(f')$ will be twice the size of $\mathbf{W}(f)$.
6. Use equation 7 to reconstruct the missing samples of $d_{\text{dec}}(f')$.
7. Repeat steps 2-6 for all frequencies.
8. Transform the reconstructed *f-x* data to *t-x* domain.

To exemplify the procedure, in Figure 11a we illustrate a synthetic seismic section composed of 3 hyperbolic events with 81 traces. Next, we decimate the original data to obtain the decimated seismic section in Figure 11b with 41 traces. The FGFT interpolation of the decimated data is shown in Figure 11c. Figures 11d, 11e, and 11f represent the *f-k* panels of data in Figures 11a, 11b, and 11c, respectively. This underscores an important property of the FGFT interpolation method, which is the ability to cope with severely aliased energy in interpolating the decimated data. The *f-k* spectra of the interpolated data contains some artifacts because of the compromise made in FGFT method to gain speed rather than resolution.

Figure 12a illustrates another synthetic seismic section which contains events with conflicting dips with the total number of traces equal to 81. The decimated data and the reconstructed data using FGFT interpolation method are shown in Figures 12b and 12c, respectively. Figures 12d, 12e, and 12f represent the *f-k* spectra of data in Figures 12a, 12b, and 12c, respectively. The FGFT interpolation method was able to remove the wrap-around energy in the decimated data. However, it is clear that the speed advantage of FGFT interpolation has come with the cost of some leftover artifacts in the *f-k* spectrum of the interpolated data.

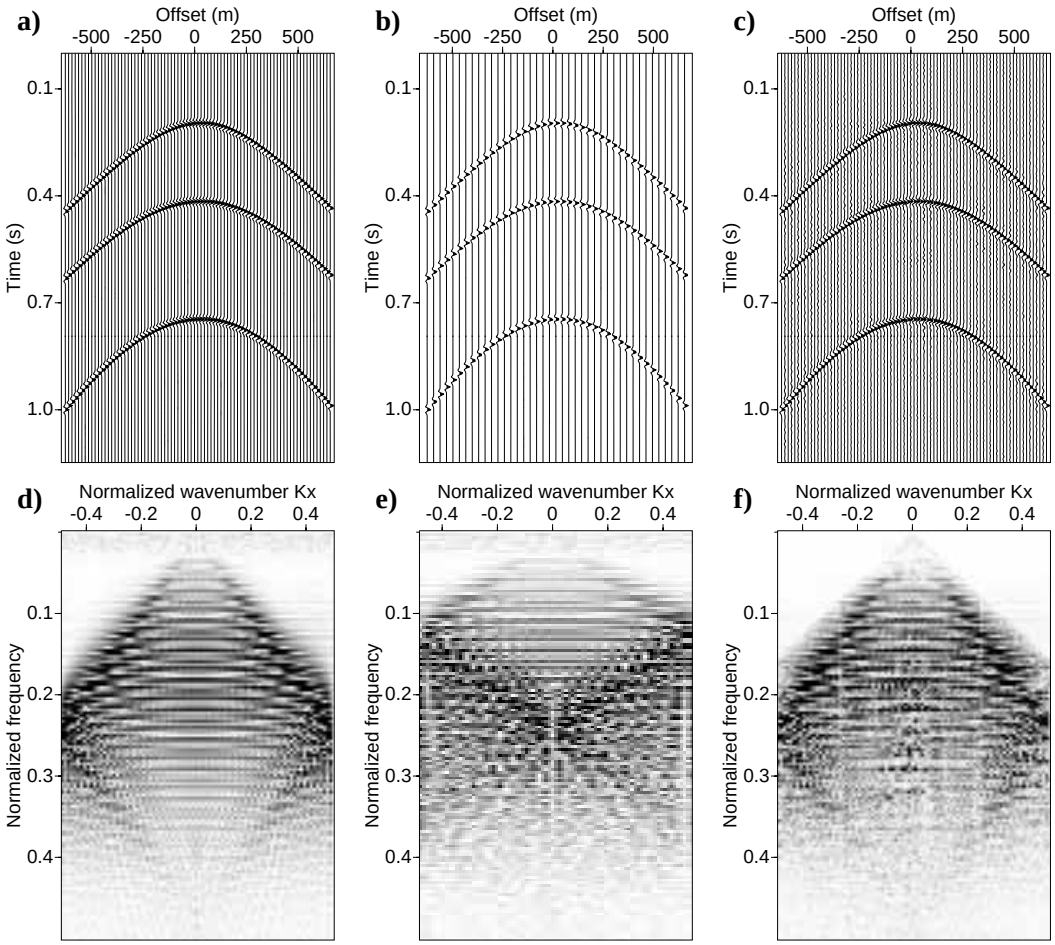

FIG. 11. a) Original synthetic seismic section with 3 hyperbolic events. b) Seismic section after decimating every other traces. c) Reconstructed data using the FGFT interpolation method. d)-f) are the *f-k* representations of (a)-(c).

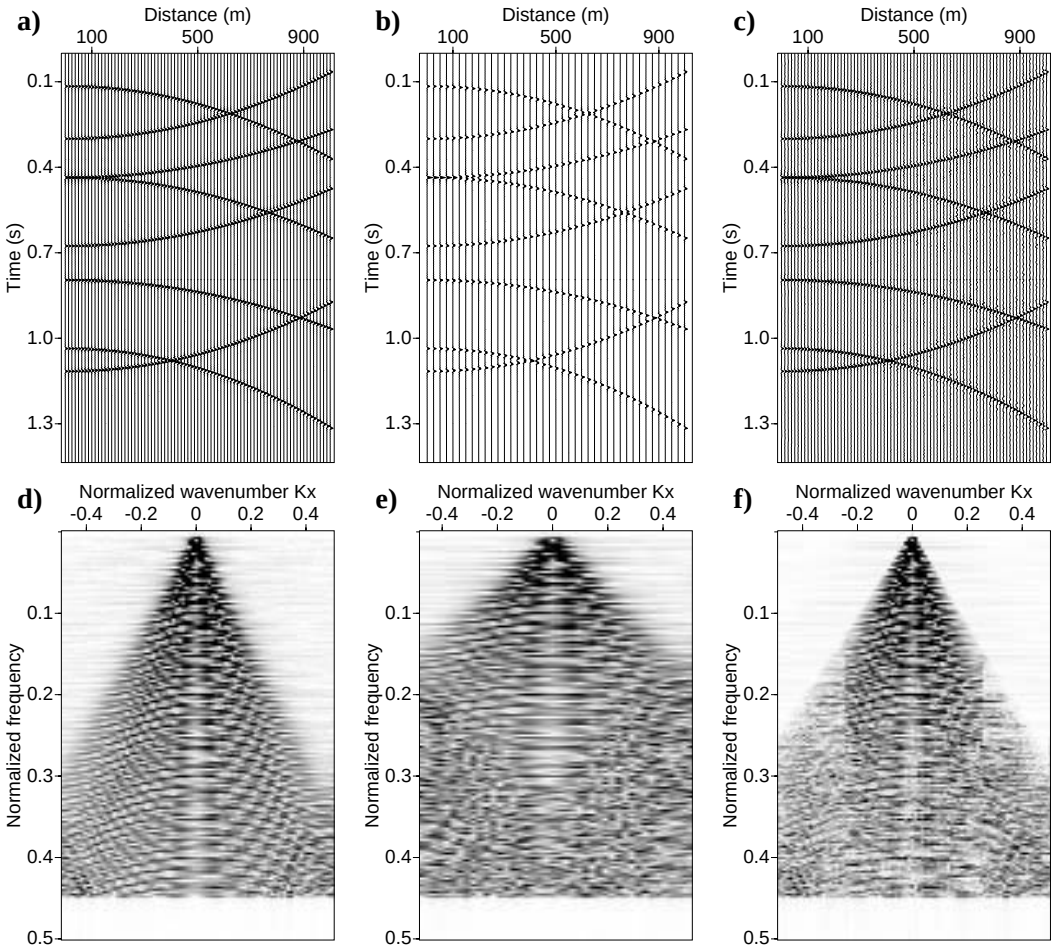

FIG. 12. a) Original synthetic seismic section with conflicting dip events b) Seismic section after decimating every other traces. c) Reconstructed data using the NFST interpolation method. d)-f) are the *f-k* representations of (a)-(c).

Field data example

In order to investigate the performance of FGFT interpolation method on real data, we chose a near offset section from a Gulf of Mexico data set. The original data which contains 145 equally spaced traces is shown in Figure 13a. Figure 13b shows the interpolated data using the FGFT interpolation method. The interpolated section contains 290 traces. To provide a comparison, we also interpolated the original data using the *f-x* adaptive interpolation method of Naghizadeh and Sacchi (2009). The result is shown in Figure 13c. The run-time was 9 seconds for FGFT interpolation and 180 seconds for the *f-x* adaptive interpolation. Therefore, speed was significantly increased with the FGFT interpolation method. The interpolated traces using the FGFT interpolation method contain more noisy features than when using the *f-x* adaptive interpolation method. However, the performance of both methods was comparable. These operations were implemented in Matlab and run on a single quad-core notebook computer. Figures 14a-c shows the *f-k* spectra of the data in Figures 13a-c.

DISCUSSION

The FGFT method is an effort to recover missing samples of a nonstationary signal without the necessity of windowing in spatial directions. For a randomly sampled stationary signal on a regular grid, seeking a sparse representation in the Fourier domain leads to successful reconstruction of missing samples. This is because a sparse Fourier representation is a property of stationary signals. This is not true of nonstationary signals. A mixture of regular sampling and a nonstationary signal further complicates the task of reconstruction by introducing artifacts in the Fourier domain. The commonly used approach to tackle this issue is to divide the signal into small windows, within which data have stationary properties. One needs to choose an optimal length for window sizes as well as overlap between windows for smooth reconstruction of missing samples. The FGFT interpolation method combines all these tasks into one operator to obtain a sparse representation of FGFT coefficients. The FGFT interpolation handles random distribution of samples on a regular grid, windowing and nonstationarity problems simultaneously. The sparsity constraint on the FGFT domain is not effective in eliminating the aliased energy of regularly sampled data, especially for high frequencies.

The computational cost of FGFT interpolation is very low compared to the alternative methods such as f-x adaptive interpolation (Naghizadeh and Sacchi, 2009). The FGFT method is memory-efficient because of its non-redundant nature. The speed advantage of FGFT comes with loss of resolution both in time/space and frequency/wavenumber axes. However, for the spatial interpolation of seismic data the resolution of FGFT is expected to be sufficient in most cases. In addition to the speed and memory advantages, the FGFT interpolation method is applicable for both regular and random sampling. The FGFT is designed to be a non-redundant transform i.e. the number of samples is equal in the original and transformed signal. This property can be compromised by using overlapping segmentations in the Fourier domain to gain a transformed signal with higher resolution. Such a trade-off might be necessary in the situations where the resolution is more important than the speed of the operation. The performance of FGFT interpolation method can be improved by increasing the number of available samples and the total size of data.

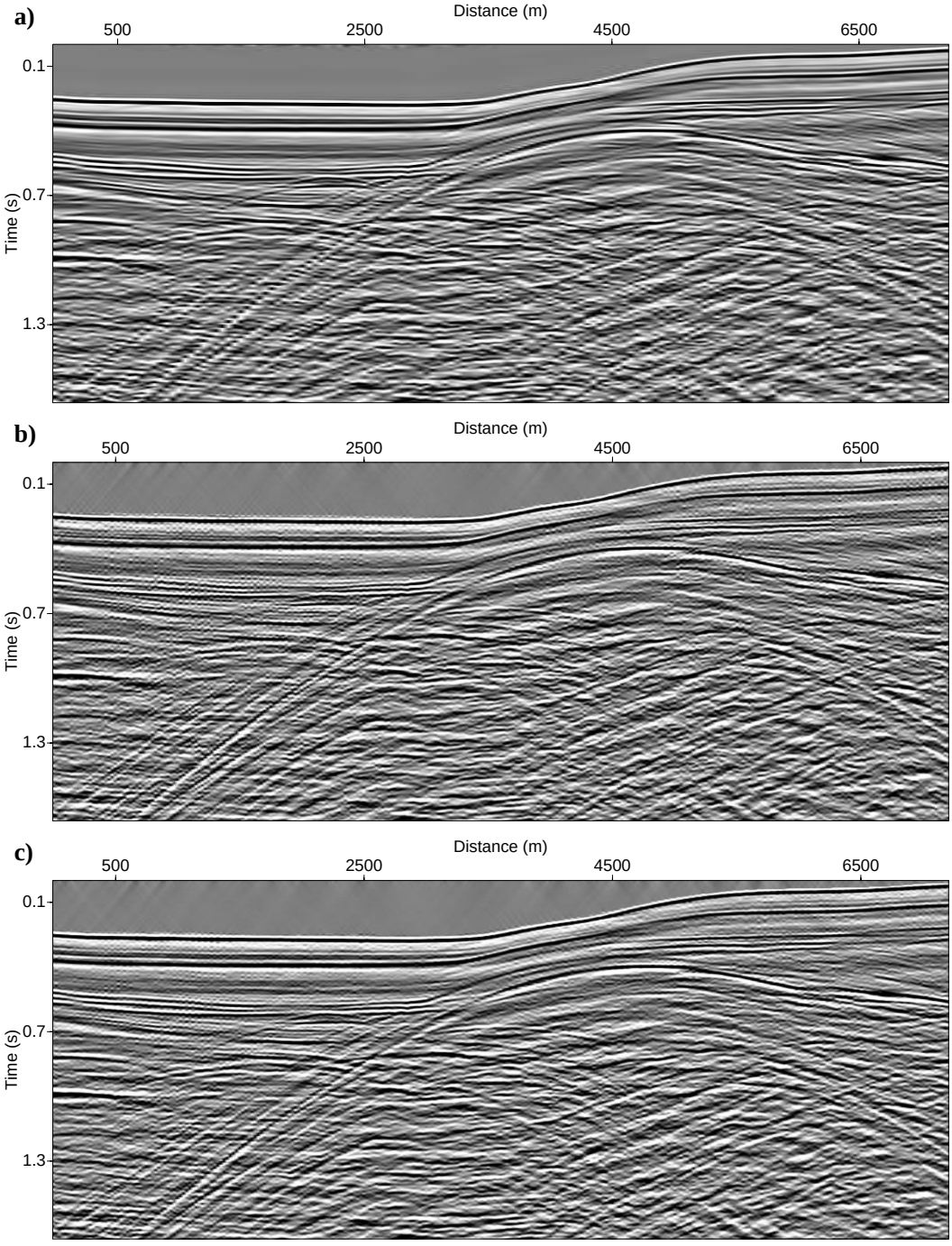

FIG. 13. a) A near offset section form the Gulf of Mexico data set. b) Interpolated data using the FGFT interpolation method. c) Interpolated data using *f-x* adaptive interpolation method (Naghizadeh and Sacchi, 2009).

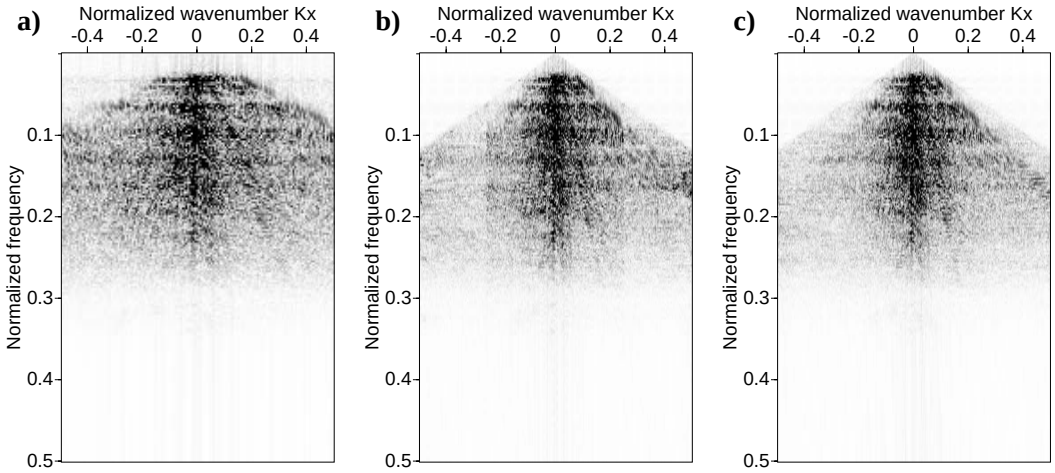

FIG. 14. a)-c) are the *f-k* representations of Figures 13a-13c.

We have a further interest in the FGFT domain because of its close connection to wave physics. The FGFT locally decomposes the data directly in terms of frequency or wavenumber. Processing, imaging, and inversion methods benefit from locally available frequency and plane-wave angle information. We are examining the FGFT algorithm as a means of providing this information, while concurrently acting as a framework for managing incomplete data.

CONCLUSIONS

The FGFT is a fast and efficient way of analyzing nonstationary signals and identifying their time-frequency evolution. We used FGFT inside a least-squares fitting algorithm to interpolate nonstationary seismic data. The method has the ability to cope with rapid and local changes of dip information in the seismic data. For randomly sampled data on a regular grid, we seek a sparse representation in the FGFT domain to recover the missing samples. For regularly sampled data, the FGFT interpolation method utilizes the low frequency portion of data for beyond-alias reconstruction of the high frequencies. The proposed method is very fast and less demanding on computational memory compared to the alternative methods. The extension of FGFT interpolation to multidimensional applications is a straightforward task because in principle, it involves simple multidimensional Fourier transforms.

ACKNOWLEDGEMENTS

We thank the financial support of the sponsors of the Consortium for Research in Elastic Wave Exploration Seismology (CREWES) at the University of Calgary. We would also like to thank Dr. Mauricio Sacchi for his inspiring discussions and useful comments.

REFERENCES

Abma, R., and Kabir, N., 2005, Comparison of interpolation algorithms: The Leading Edge, **24**, No. 10, 984–989.

Beylkin, G., Coifman, R., and Rokhlin, V., 1991, Fast wavelet transforms and numerical algorithms: Comm. on Pure and Appl. Math., **44**, 141–183.

Brown, R. A., Lauzon, M. L., and Frayne, R., 2010, A general description of linear time-frequency transforms and formulation of a fast, invertible transform that samples the continuous s-transform spectrum nonredundantly: IEEE Trans. Signal Processing, **58**, No. 1, 281–290.

Candes, E. J., Demanet, L., Donoho, D. L., and Ying, L., 2005, Fast discrete curvelet transforms: Multiscale Modeling and Simulation, **5**, 861–899.

Candes, E. J., and Donoho, D. L., 2004, New tight frames of curvelets and optimal representations of objects with piecewise-c2 singularities: Comm. on Pure and Appl. Math., **57**, 219–266.

Claerbout, J., 1992, Earth Soundings Analysis: Processing Versus Inversion: Blackwell Science.

Darche, G., 1990, Spatial interpolation using a fast parabolic transform: 60th Annual International Meeting, SEG, Expanded Abstarcts, 1647–1650.

Duijndam, A. J. W., Schonewille, M. A., and Hindriks, C. O. H., 1999, Reconstruction of band-limited signals, irregularly sampled along one spatial direction: Geophysics, **64**, No. 2, 524–538.

Hennenfent, G., and Herrmann, F. J., 2008, Simply denoise: Wavefield reconstruction via jittered undersampling: Geophysics, **73**, No. 3, V19–V28.

Hestenes, M. R., and Stiefel, E., 1952, Methods of conjugate gradients for solving linear systems: Journal of Research of the National Bureau of Standards, **49**, No. 6, 409–436.

Liu, B., 2004, Multi-dimensional reconstruction of seismic data: Ph.D. thesis, University of Alberta.

Margrave, G. F., Henley, D., Lamoureux, M. P., Iliescu, V., and Grossman, J., 2003, Gabor deconvolution revisited: SEG Expanded Abstracts, **22**, 714–718.

Menke, W., 1989, Geophysical Data Analysis: Discrete Inverse Theory: Academic Press.

Naghizadeh, M., and Sacchi, M. D., 2009, *f-x* adaptive seismic-trace interpolation: Geophysics, **74**, No. 1, V9–V16.

Naghizadeh, M., and Sacchi, M. D., 2010, On sampling functions and fourier reconstruction methods: Geophysics, **Accepted for publication**.

Sacchi, M. D., Ulrych, T. J., and Walker, C. J., 1998, Interpolation and extrapolation using a high-resolution discrete fourier transform: IEEE Transaction on Signal Processing, **46**, No. 1, 31–38.

Scales, J. A., and Gersztenkorn, A., 1988, Robust methods in inverse theory: Inverse Problems, **4**, 1071–1091.

Spitz, S., 1991, Seismic trace interpolation in the *F-X* domain: Geophysics, **56**, No. 6, 785–794.

Stockwell, R. G., Mansinha, L., and Lowe, R. P., 1996, Localization of the complex spectrum: The s transform: IEEE Trans. Signal Processing, **44**, No. 4, 998–1001.

Tikhonov, A. N., and Goncharsky, A. V., 1987, Ill-posed problems in the natural sciences: MIR Publisher.

Trad, D., Ulrych, T. J., and Sacchi, M. D., 2002, Accurate interpolation with high-resolution time-variant radon transforms: Geophysics, **67**, No. 2, 644–656.

Zwartjes, P., and Gisolf, A., 2006, Fourier reconstruction of marine-streamer data in four spatial coordinates: Geophysics, **71**, No. 6, V171–V186.