High fidelity seismic trace interpolation

John Guo[1,2], Ye Zheng[1] and Wenyuan Liao[2]*
[1] *Geo-X Exploration Services Inc.*
[2] *University of Calgary*

Summary

We present the Arbitrarily Sampled Fourier Transform (ASFT) method for 5D interpolation, which incorporates several enhancements over existing methods. The ASFT method is able to use true input trace positions and solves spatial frequency contents at arbitrary locations in wavenumber domain, and is able to achieve better sparsity in the *f-k*^{4} domain. ASFT was tested on a dataset in the Western Canadian Sedimentary Basin (WCSB) and produced excellent interpolation results.

Introduction

Seismic trace interpolation, which spatially transforms irregularly sampled field data to regularly sampled data or to any desired grid in general, is an important step in seismic data processing. A class of algorithms used in the industry, such as Minimum Weighted Norm Interpolation (MWNI) (Liu and Sacchi, 2004; Trad 2009), Projection Onto a Convex Set (POCS) (Abma and Kabir, 2006) and Anti-Leakage Fourier Transform (ALFT) (Xu et al., 2005; Xu et al., 2010) are based on the Fourier theory in the f-k^4 domain by computing the estimated spatial frequency contents of irregularly sampled data.

Fourier Theory Based 5D Interpolation

For Fourier theory based 5D interpolation, the seismic data is composed of 1-dimensional temporal information and 4-dimensional trace spatial information, and it is said to be in the time-spatial *t-x*^{4} domain.

As a first step, it's customary to transform the data from *t-x*^{4} domain to temporal frequency-spatial *f-x*^{4} domain via conventional Fast Fourier Transform (FFT). The data for each temporal frequency f is called a "temporal frequency slice".

Then, for each temporal frequency slice, the interpolation algorithm computes the "spatial frequency contents" of the input data in the wavenumber domain *k*^{4}.

The spatial frequency contents, denoted by S, typically consist of many Fourier coefficients. Each Fourier coefficient, denoted by $c(k)$, is a complex number representing the energy corresponding to its wavenumber k, which is a 4-dimentional vector in the wavenumber domain. All solved Fourier coefficients together form the estimated spatial frequency contents, and interpolated traces at regularized locations are computed by inverse Fourier transform.

Let $c(k)$ be a Fourier coefficient and k be the associated wavenumber, the inverse Fourier transform onto a 4-dimensional location x is, with appropriate normalization,

$$f^{-1}(c(k), x) = c(k)e^{i2\pi<k,x>} \qquad (1)$$

where $<,>$ is the inner product operator.

Let S be the spatial frequency contents of a given temporal frequency slice; the final interpolated result at x is obtained by summing up the inverse transform values from the entire spatial frequency contents, that is,

$$I(S, x) = \sum_{c(k)\in S} f^{-1}(c(k), x) \qquad (2)$$

All the Fourier based 5D interpolation methods mentioned above are based on the same principle; however, there are differences in the actual implementations in estimating the spatial frequency contents, which cannot be computed directly by FFT since the input trace locations are irregular. These differences can have a significant impact on the effectiveness and final quality of the interpolation algorithm.

Estimating Spatial Frequency Contents in the k^4 Domain

Binning is used in POCS and MWNI such that each input trace is treated as if it's at the nearest location in a regular grid in the *x*^{4} domain. From there, the spatial frequency contents are estimated by FFT combined with an optimization scheme. However, by using bin locations it might significantly lose interpolation accuracy.

ALFT uses true input trace locations and computes the spatial frequency contents in an iterative manner: in each iteration, it first calculates the estimated energy associated with each of the Fourier coefficients from a pre-selected set of wavenumbers (typically a regular grid of points, denoted by W) in the *k*^{4} domain by a weighted Discrete Fourier Transform (DFT) method: let T be temporal frequency slice input data in the *x*^{4} domain indexed by l, and each trace t_l be associated with location x_l, energy e_l, and an appropriately defined weight w_l. For a wavenumber $k \in W$, its corresponding Fourier coefficient is estimated as,

$$c(k) \approx \sum_{t_l \in T} w_l e_l e^{-i2\pi < k, x_l >} \tag{3}$$

Then, it optimizes $c(k)$ and selects the Fourier coefficient $\tilde{c}(k)$ with the largest energy,

$$\tilde{c}(k) = \arg\max_{k \in W} |c(k)| \tag{4}$$

Then $\tilde{c}(k)$ is added to the spatial frequency contents, and it performs inverse transform of $\tilde{c}(k)$ to each of the input trace locations x_l, subtracts the result from current iteration input data to get "residual" data, and proceeds into the next iteration with residual data as new input.

However, this selection only guarantees that the Fourier coefficient $\tilde{c}(k)$ has the largest energy among $k \in W$, while in general there likely exists some $\bar{k}$ satisfying $\bar{k} \in k^4$ but $\bar{k} \notin W$ such that $|\bar{c}(\bar{k})| > |\tilde{c}(k)|$. Therefore, W must be sufficiently large (fine) so that $\tilde{c}(k)$ is close enough to $\bar{c}(\bar{k})$.

The need to have a large pre-selected set of wavenumbers is a major limitation for ALFT. Since the weighted DFT energy estimation, done to every wavenumber in W, is typically the most computationally intensive part of ALFT, and the computational cost is linearly proportional to the number of wavenumbers, therefore, a finer grid with more pre-selected wavenumbers would cause significant increase of computational time.

The Arbitrarily Sampled Fourier Transform

To address the problems, we propose a new method, Arbitrarily Sampled Fourier Transform (ASFT), for 5D interpolation. ASFT is an iterative method. In each iteration, a Fourier coefficient is computed in two stages, by an optimization scheme.

Let T be input data in the x^4 domain indexed by l, and each trace t_l be associated with location x_l and energy e_l. The cost function for Fourier coefficient $c(k)$ is defined as the sum of squared energy of residual data,

$$\mathcal{J}(c(k)) := \sum_{t_l \in T} \|e_l - f^{-1}(c(k), x)\|^2 \tag{5}$$

The cost function is the same for both stages but the optimization will be different.

In the first stage, we start with a pre-selected set of wavenumbers W. For each $k \in W$, ASFT computes $c(k)$ that minimizes the cost function,

$$c(k) = \arg\min_{k \text{ fixed}} \mathcal{J}(c(k)) \tag{6}$$

The role of $c(k)$ optimization in the first stage is similar to the weighted DFT part in ALFT. The optimization can be computed by analytical methods, and the computational cost for each $c(k)$ optimization is the same as one weighted DFT. However, the result $c(k)$ obtained from the first stage of ASFT is more accurate and easier to interpret, since the optimization is a direct cost function minimization, whereas the weighted DFT is an estimation. At the end of the first stage, the largest Fourier coefficient $\tilde{c}(k)$ is selected and is feed into the second stage as the starting point.

The second stage uses the same cost function, but it now allows k to be at an arbitrary point in the wavenumber domain. This can be done by using a gradient-based numerical optimization method such as the Conjugate Gradient (CG) or the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm. At the end of the second stage, the Fourier coefficient with the largest energy at a wavenumber location is computed, that is, we have

$$\bar{c}(\bar{k}) = \arg\min_{\bar{k} \in k^4} \mathcal{J}(c(k)) \tag{7}$$

which is the desired outcome.

Then $\bar{c}(\bar{k})$ is added to the to the spatial frequency contents, and ASFT performs inverse transform of $\bar{c}(\bar{k})$ to each of the input trace locations x_l, subtracts the result from current iteration input data to get "residual" data, and proceeds into the next iteration with residual data as new input.

ASFT has three advantages compared with ALFT.

First, it allows the use of a coarser grid W of pre-selected wavenumbers. Since the optimization for each $\tilde{c}(k)$ with fixed k in the first stage of ASFT is as quickly as the weighted DFT in ALFT, and in the second stage of ASFT, while requiring gradient-based optimization, only optimizes one Fourier coefficient, therefore, ASFT takes much less time than ALFT in one iteration due to fewer wavenumbers in W.

Second, the precise optimization of the Fourier coefficients causes the residual energy to be reduced at a faster rate. If a threshold of residual energy or Fourier coefficient energy is used, then ASFT is able to arrive at that threshold in less number of iterations. This further strengthens its speed advantage.

Third, ASFT results in better interpolation quality, due to the fact that the spatial frequency contents are sparser and

more accurate. The better accuracy and sparsity reduce leakage, which is originally the objective of ALFT.

1D Synthetic Data Test

We present a 1D synthetic data to demonstrate the advantages of the Arbitrarily Sampled Fourier Transform.

First, 128 arbitrary sample positions in $(0, 128)$ are generated by a random function. Let the sorted result (in ascending order) be $x_l, l = 1, \dots, 128$. The synthetic data at those irregular positions are generated by the function

$$f(x_l) = \sin\left(\frac{x_l}{5}\right). \quad (8)$$

Then we compare the reconstructed results at regular grid points (1, 2, …, 128) by 1-iteration ALFT and 1-iteration ASFT. They are shown in the Figure 1 below along with input data.

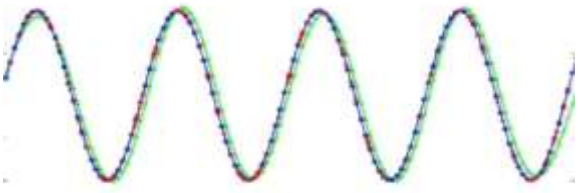


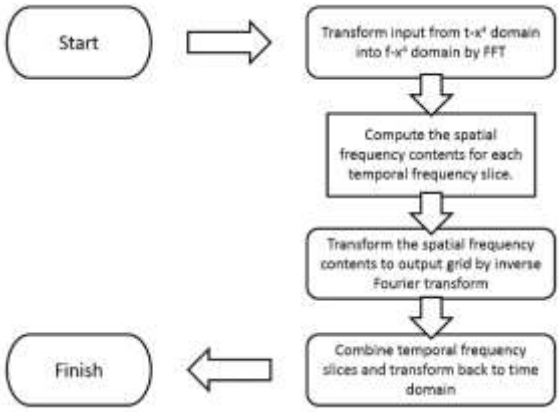
Figure 1. 1-D synthetic data test; red, irregular input, green, reconstruction by ALFT, blue, reconstruction by ASFT.

It can be seen that ASFT is able to reconstruct the data with high accuracy, while there's a small but noticeable error in the ALFT reconstruction. This is because that ALFT is only able to generate Fourier coefficients at W. In this case, the true wavenumber of the data signal is $10\pi = 31.4\dots$ However, in a typical setting for ALFT the wavenumber only takes integer values, $k = 31$ is selected, resulting in interpolation inaccuracy.

The problem can be (partially) mitigated by either increasing the number of iterations or specifying a finer grid in ALFT, but either of the solutions comes with the cost of increasing computational time significantly, especially considering that in real data W is 4-dimensional, so to make the grid 2 times finer requires 16 times the computing time. Moreover, even then the reconstruction by ALFT would only be an approximation; in contrast, ASFT is able to find the exact wavenumber.

Real Data Example

A typical work flow of the ASFT method is shown in the flowchart. The input data are CMP gathers with deconvolution, statics, scaling and NMO corrected with final velocity. Noise attenuation should be applied to the input data as well for optimal interpolation output. The gathers are sorted into Common Offset-Azimuth (COA) domain or Common Offset Vector (COV) domain as processers' preference. At first, 1-D FFT is applied to each trace to transform the data from time domain to temporal frequency domain.



Then, data at each temporal frequency is transformed by ASFT into the x^4 domain using the original spatial coordinates. Interpolation is applied to each temporal frequency slice in 4-dimensional space/wavenumber domain. After solving all uneven spaced wavenumbers in the k^4 domain, they are inverse transformed back to temporal frequency domain by ASFT and further transformed back to time domain by FFT.

We have applied ASFT interpolation to a dataset in the Western Canadian Sedimentary Basin (WCSB). It is a MegaBin dataset with 2:1 aspect ratio; in other words, every second inline is empty if the bin is a square. The task for interpolation with this dataset is: a. regularizing data in azimuth and offset direction, and b. filling the empty inlines.

Figure 2 (a) and (b) are CMP gathers before and after interpolation. Interpolation output has more traces than the input gather as requested. The characters of the input gather are naturally preserved after interpolation.

Figure 3 (a) and (b) are inline stacks before and after interpolation. They are almost identical, because the inline showing here is a live inline in the input data. For this inline, interpolation has regularized azimuth and offset. The slight difference between the before and after sections is caused by different offset/azimuth distribution, since the distribution is not uniform before interpolation.

Figure 3 (c) and (d) are xline stacks before and after interpolation. As mentioned above, every second trace in an xline is empty. For easy visual comparison, poststack interpolation was applied to the section of Figure 3 (c) so that the dead traces are filled. It is clear that the section of Figure 3 (d) with ASFT interpolation provides more details and higher resolution compared to that with poststack interpolation Figure 3 (c). Meanwhile, the comparison also demonstrates that ASFT is able to handle MegaBin geometry (upsampling) properly.

Figure 3 (e) and (f) are time slices before and after interpolation. The comparison shows that the geological features are well preserved after interpolation.

Figure 4 (a) and (b) are time slices after migration without and with interpolation. The comparison shows that ASFT interpolation has reduced migration artifacts and shown sharper and clearer geological structures.

Conclusions

We have presented a seismic trace interpolation algorithm, the Arbitrarily Sampled Fourier Transform (ASFT) method. The ASFT method is able to use true input trace positions and solves spatial frequency contents at arbitrary locations in wavenumber domain, and is able to achieve better sparsity in the f-k^4 domain.

The examples presented here have shown that ASFT effectively interpolates seismic traces, preserves geological structures and enhances prestack migration images.

Acknowledgements

We would like to thank Balazs Nemeth and Christian Escalante at BHP Billiton for providing real data, and also thank John Chiu, Don Gee and Laurie Ross at Geo-X Exploration Services for processing the data. The reported work is jointly sponsored by Geo-X and NSERC Engage / Engage-Plus programs. Their genereous financial support is greatly appreciated.

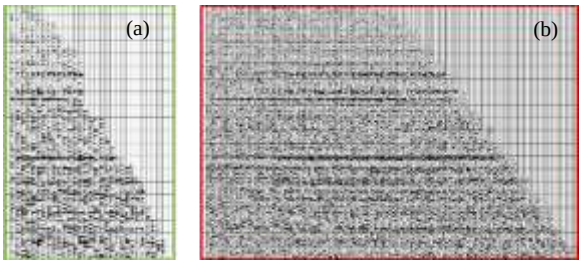


Figure 2. Comparison of gather traces before (a) and after (b) ASFT interpolation.

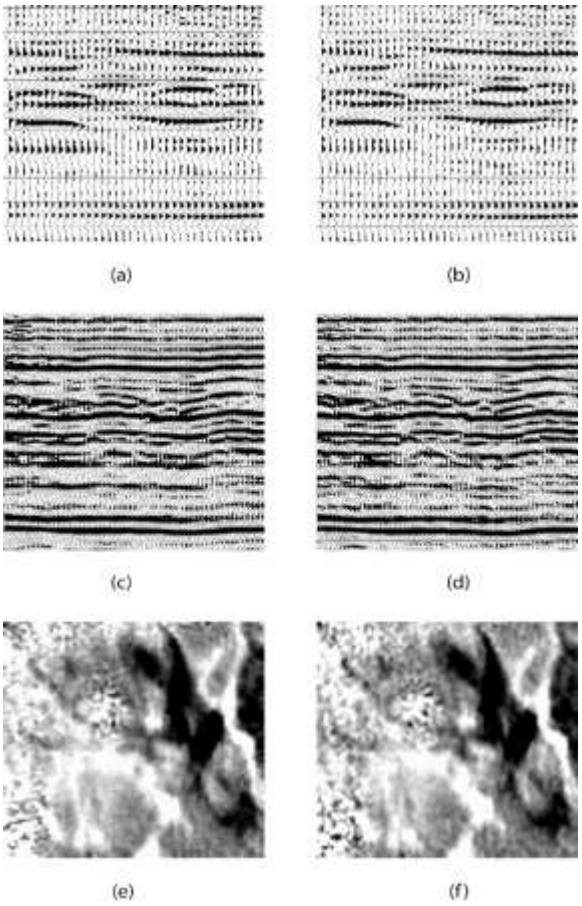


Figure 3. Comparison of post-stack results. (a) and (b): inline before and after ASFT interpolation; (c): xline with poststack interpolation for easy comparison with (d), and (d): xline after ASFT interpolation; (e) and (f): time slice before and after ASFT interpolation.

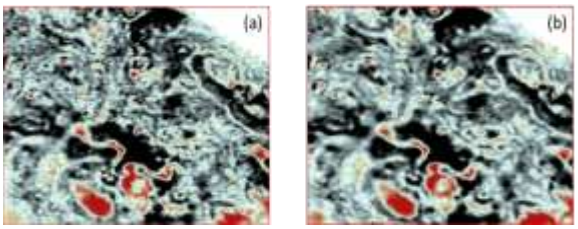


Figure 4. Comparison of time slices after interpolation without (a) and with (b) ASFT interpolation. ASFT has reduced migration artifacts and shown shaper and clearer structures.