Time domain 3D prestack seismic trace interpolation with input optimization

Xishuo Wang, Geo-X Systems Ltd., Calgary, Alberta, Canada.

2004 CSEG National Convention

Introduction

Most existing prestack interpolation algorithms are frequency domain schemes, which take a contiguous block of input data and resample it to generate a new block of interpolated data. My method operates in the time domain and is intended to best utilize input data at every single output (i.e., interpolated) location. We have submitted a patent application for this algorithm under the commercial name **Trinity**.

Time domain Interpolation by polynomials in 4D space

To interpolate a trace at location $\mathbf{r}_{out}$ = (X_S,Y_S,X_r,Y_r), a group of M traces at $\mathbf{r}_m=(x_{sm},y_{sm},x_{rm},y_{rm})$, m=1,2…M, are chosen as input. At a fixed time sample, the amplitudes of the input traces are fitted to a polynomial of fixed order, (n,N), in 4D space:

The polynomial coefficients a_{ijkl} are solved by least squares fitting. Then substituting the interpolated position $\mathbf{r}_{out}$ into the polynomial,

$$P(a_{ijkl},\mathbf{r}_m)=\sum_{i,j,k,l=0}^{\substack{i,j,k,l\le n\\ i+j+k+l\le N}} a_{ijkl}x_{sm}^i y_{sm}^j x_{rm}^k y_{rm}^l$$

$P(a_{ijkl},\mathbf{r}_{out})$ gives the interpolated amplitude for this time sample. The procedure is repeat for all the time samples to complete the interpolation of this single trace.

Optimization of input trace selection

To interpolate a single trace at $\mathbf{r}_{out}$ = (X_S,Y_S,X_r,Y_r), the input traces are chosen to be Those M traces among *all* the input traces at $\mathbf{r}_m$ = $(x_{sm},y_{sm},x_{rm},y_{rm})$ such that the distance between $\mathbf{r}_{out}$ and $\mathbf{r}_m$ in 4D space,

$$d_{4i}^2=(x_{sm}-X_s)^2+(y_{sm}-Y_s)^2+(x_{rm}-X_r)^2+(y_{rm}-Y_r)^2\ ,$$

or its *reciprocal* (if smaller),

$$d_{4r}^2=(x_{rm}-X_s)^2+(y_{rm}-Y_s)^2+(x_{sm}-X_r)^2+(y_{sm}-Y_r)^2$$

is the smallest. Notice that the above definition of 4D distance and its reciprocal is symmetric in both input and interpolated positions. In other words, switching source/receiver positions of input *or* interpolated position in the definition of d_{4i} results in the same definition of d_{4r}.

I consider the above input selection optimal for three reasons, (1).Only traces closest to the interpolated position contribute to the interpolation, (2).A specific subset of M input traces are independently chosen for every single output trace, and (3).The reciprocity principle expands the coverage of the input data set in the sense that a “new” shot/receiver appears to “exist” at true receiver/shot positions. One of the major advantages of this optimized input trace selection lies in the fact that if the 4D distance between input & output positions are small, aliased and structured data may be less problematic since a single nearby input trace will naturally constrain the interpolation.

Three modes of interpolation

1. Interpolating to a new uniformly sampled shot and receiver grid. This is intended to improve the performance of prestack migration.

2. Interpolating to a uniformly sampled CMP grid, where each CMP gather consists of uniformly sampled offsets and azimuths. Inspecting such CMP gathers, and selectively stacking them by offset/azimuth ranges, may provide some insight to structural response, anisotropy and AVO effects etc. This also offers an alternative to *bin-balancing.*

3. Interpolating to input locations. This will serve as a new method of noise reduction. Any input trace will contribute to the “interpolation” of itself, since the 4D distance to itself is zero. Consequently this NR method incorporates an intrinsic mixing with “raw” data.

Data examples

Fig.1 shows part of one even numbered shot gather of a highly irregular and sparse 3D data set. Fig.2 shows the same shot interpolated by using only odd numbered shots as the input. The difference of true and interpolated is shown in Fig.3. More data examples will be shown in my presentation.

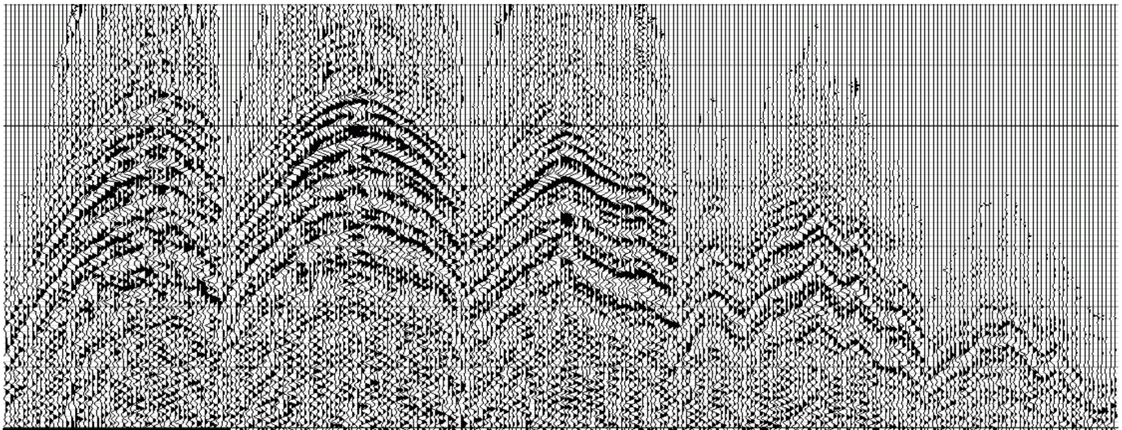

Fig.1 input shot

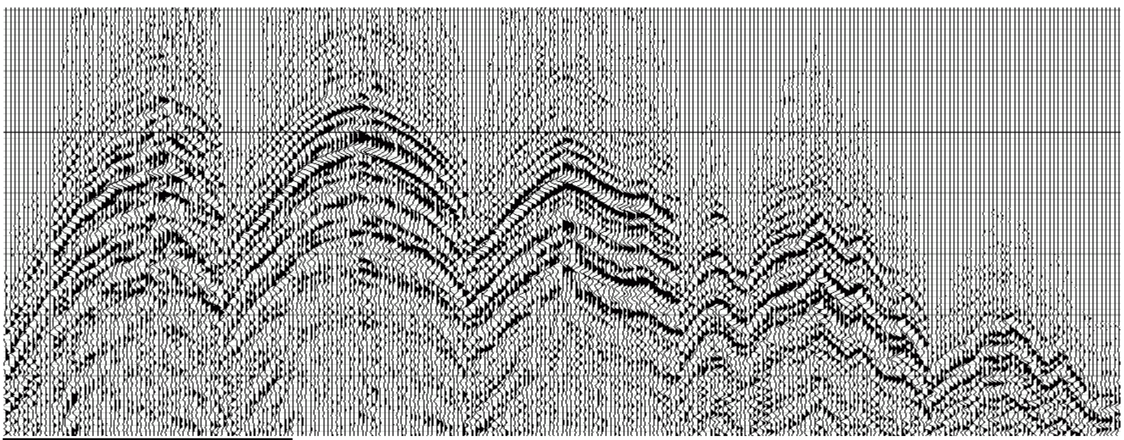

Fig.2 interpolated shot

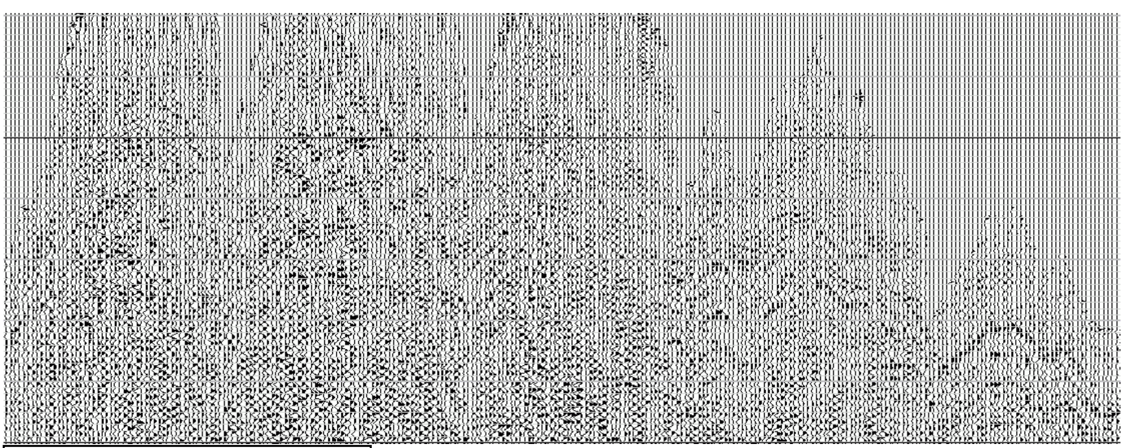

Fig.3 input minus interpolated