P066

Gather Flattening Based on Event Tracking for Each Time Sample

N. Gulunay* (CGGVeritas), M. Magesan (CGGVeritas) & H. Roende (Marathon Oil Company)

SUMMARY

There are times seismic gathers show abnormal Residual Move Out (RMO) which is difficult to correct by curve based RMO scan methods. This paper illustrates a general gather flattening method that is based on event tracking to correct such gathers.

Introduction

Getting a good image by stacking after normal moveout (NMO) correction and/or after prestack migration requires reasonably flat gathers. Amplitude versus offset (AVO) studies on prestack data requires flat gathers after NMO. Otherwise slope and intercept attributes get contaminated. When ray path of the seismic energy includes inhomogeneities NMO algorithms based on flat layer assumptions, even prestack time migration, may fail to produce flat gathers as shown with the gather in Figure 1. For such gathers one may need a robust, brute force, gather flattening method.

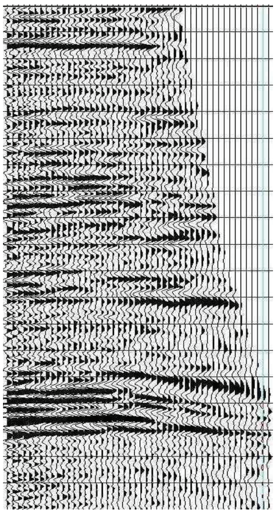

Figure 1 A gather that could not be corrected during RMO application with analytical curves. Timing line increment is 100 ms.

The need for such an application was shown by Hinkley et al. (2003) and a solution which they called "Dynamic Gather Flattening (DGF)" was presented. Flattening by use of an external stack Gulunay et. al. (2007a) is also possible and is a reliable method in terms of structural stability and preserving AVO. Here we will present a more general application of gather flattening which can also preserve AVO. This method is similar to Hinkley et al (2004) in that it is a brute force (i.e. non-physical) gather flattening method and is described in detail in Gulunay et al (2007b).

A Generalized Gather Flattening method

How can we correct gathers which have residual, even, abnormal moveout? Firstly, all such correction should be done as a one to one mapping. That is, every output data sample should come only from one input data sample on the same trace:

$$\mathbf{D}_a(t, x) = \mathbf{D}_b(t + m(t, x), x)$$

where x is offset (or trace number of the gather sorted from smallest to largest offset), t is time, "a" stands for after, and "b" for before, and m (t, x) is the moveout function. To do flattening we first need to estimate the moveout function. We do that by tracking the events (wavelet) across traces at each near offset (t_0) time sample. Secondly, such a correction should not cause sudden stretch or squeeze of traces. This we do by editing and smoothing raw picks along time and space axis. Thirdly, these corrections should not cause structural changes or CDP jitter. This we do by making move out values consistent from CDP to CDP. Finally, such corrections should not alter AVO behavior. This is done by using absolute values of cross correlation results while picking static shifts and also is the by product of the nature of the integration of trace to trace correlations.

Move out Function Estimation by Event tracking with two trace correlations

Starting at the innermost traces and at a time t_0 we need to track the event times, t, of the wavelet across the gather to get the moveout function. This can be done by cross-correlation of consecutive neighboring traces (we call this "2-trace correlation method") and integrating the static shifts so calculated. Note that to be able to track an event one needs to allow correlation window of two traces climb up and down as the event moves. So, cross correlation of two consecutive traces may not be centered at the starting time t_0 any more after one moves away from inner traces. Repeating the same process for other near offset times t_0 completes the determination of the moveout function to be applied to the input gather at time t and offset x.

Note that while picking the static shift that gives the cross correlation maximum one should use maximum of the absolute value so that static shift resulting with negative cross-correlation peaks does not get eliminated from the picks. Also, sudden changes in pick times cause wavelet stretch and squeeze, hence, moveout function resulting from the raw picks needs to be smoothed over time after some spatial consistency check. Afterwards the moveout values can be output for QC displays or for further processing.

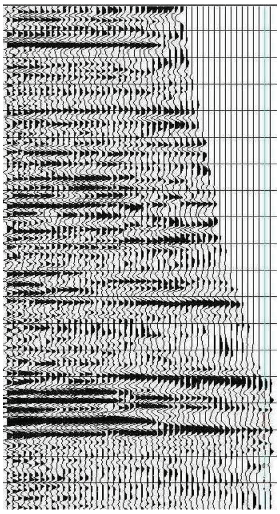

Figure 2 Gather in Figure 1 after gather flattening with event tracking.

The result of event tracking with the two trace correlation algorithm on the gather shown in Figure 1 is given in Figure 2. Although some stretch and squeeze is present the method is able to flatten the gather well. These artifacts can be reduced if some precautions (to be discussed later) are taken.

Another application of event tracking with two trace cross correlation algorithm and following moveout correction is shown on a data set from Eastern Canada (Figures 3a and 3b). Note that Class2 AVO that is present in the input (cursor point) is preserved. This result was obtained by using 60 ms time window for cross correlations, and, an offset varying maximum shift allowance (12ms at minimum offset and 20 ms at maximum offset). We rejected normalized correlation values less than 0.7, did 5-trace lateral coherency check and did not allow more than 4 ms deviation from the mean within the five trace space window. We smoothed moveout values by a 40 ms box car filter along the time axis.

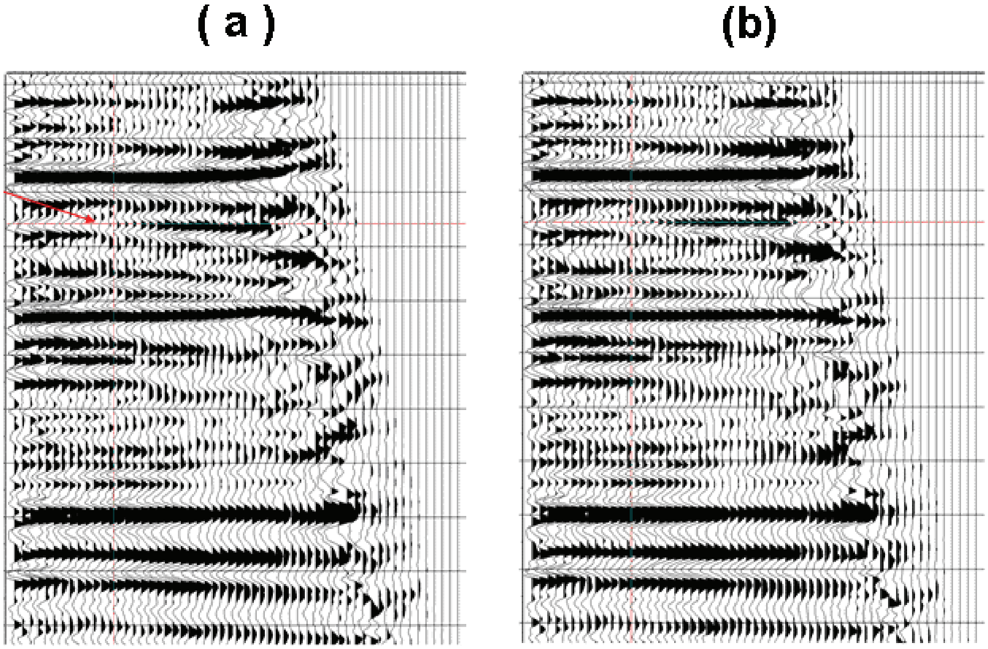

Figure 3 (a) A CMP gather from Eastern Canada with a known Class2 AVO behavior (b) Same gather after flattening with event tracking with two trace correlations.

Consistency of picks across gathers

The moveout picking method described above flattens gathers well. Such a process, although it creates flat gathers and is very pleasing to the eye, may show jitter on far offset stacks as there is nothing in the algorithm discussed so far that will tie moveout values obtained for one gather to the ones on the next. That is, as gathers are individually processed, the moveout values at the same t_0 may be inconsistent from gather to gather. To lessen this undesirable result one may process multiple gathers at a time by saving them and their move out values in memory. Then one can split the individual moveout values into short and long period components, smooth the long period component across gathers, and add the short period component of the individual gathers to the smoothed long period component. The number of gathers that needs to be used in this multi-gather processing scheme could be made user controllable and turned off if external processing of moveout values are more desirable for the user than the internal one to the algorithm.

Event tracking with correlation within 5-trace groups

To bring stability into event tracking one can consider a group of traces instead of two, with their center moving one trace at a time across each gather. The number of traces in the group is somewhat arbitrary. Large number of traces is obviously costly and may not bring any more stability. We chose the group size as five. Referring events times to the first trace of the group four values are needed to define event times: T_1 for trace 2, T_2 for trace 3, T_3 for traces 4, and T_4 for trace 5. There are ten possible pairing (cross-correlations) between five traces. As we have only four unknowns, T_1, T_2, T_3, and, T_4, but ten equations we can solve above equations in the least squares sense for these four unknowns (Gulunay et al, 2007b)

Internal and External Stacks

There are times one may wish to align traces of a CMP gather to an external stack trace at that gather as was done in Gulunay et al (2007a). Using stack of the gather to correlate its traces against provides control on the time alignment of the near and far stacks. In this approach a short window of data centered at each t_0 is correlated with same window of the external stack to find the time shifts. Then the process is repeated for all t_0 time samples. If moveout values are not too large this might be a good way of guaranteeing far and near stacks match after gather flattening. If, however, the moveout is large than the external stack fits neither near nor far traces. In this case a short offset (inner offsets) might be preferable and this can indeed be done internally (internal stack) by stacking a certain percent of the inner offset traces. In such cases it might be a good idea to either use inner offset stacks as reference or to correct the gathers first for the long period component before forming the stack trace for further correlation. Hence, here is the next section.

Short or Long (Spatial) period shifts

As we sometimes do not wish to do full moveout corrections one may consider a spatially smoothed version of the moveout values at a given time. Once long period is so corrected then it should be possible to do a stack and use the result to correlate traces against. This can be done internally or externally. We call this long period followed by internal stack option. In this option event tracking is done with the 5-trace correlation scheme and the least squares static solution mentioned above. A set of gathers on which this method is applied is shown in Figure 4a. Same gathers after final shift corrections applied are shown in Figure 4b where the following processes were applied to derive the required shifts: a)-an initial event tracking was done, b)-long period corrections from the result were applied, c)-a pilot stack trace formed and d)-each trace of the gather was correlated to that pilot to obtain the final shifts.

In this run a correlation window size of 40 ms in the shallow and 80 ms in the deep was used. Maximum trace to trace static shift was kept at 12 ms. during event tracking. Maximum final static was kept at 8ms. Correlations giving correlation quality less then 0.8 were ignored. Long period components were obtained by 25 trace box car smoothing in offset direction and 10 CMPs in CMP direction. Finally, 15 percent of all the traces from inner offset side were used in forming an internal stack which was used again for correlation with individual traces as described above. At both stages a 24 ms temporal smoothing was used on the moveout values.

Keeping the short period static corrections only is another possibility which can also have applications where, for example, one wishes to eliminate the jitter caused by neighboring traces of given CMP that belongs to different sail lines. Such short period statics derivations can be done where movement of the correlation windows up or down along to axis is prohibited and short spatial period of the resulting moveout times are used.

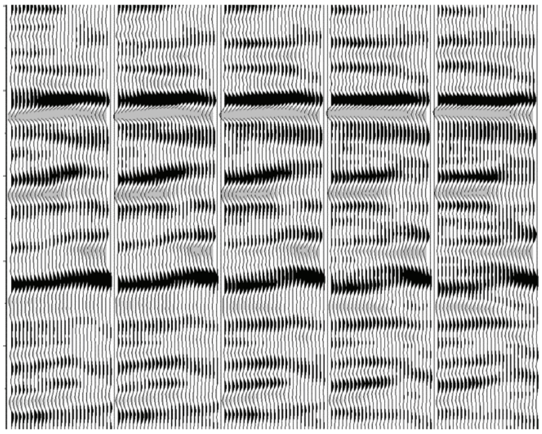

Figure 4 (a) Input gathers from North Sea.

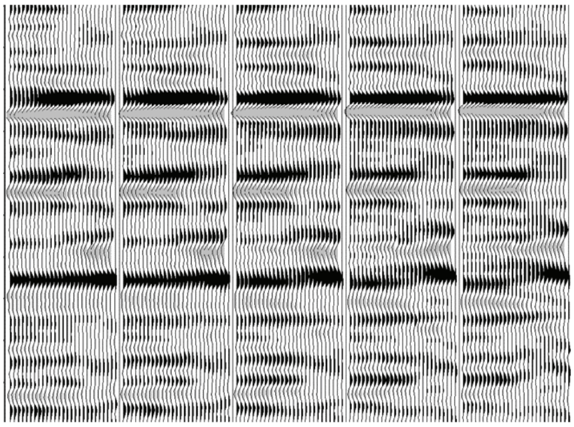

Figure 4(b) Output gathers. Flattening was done by correlating each trace with a reference stack that was done by applying the long period corrections first.

Conclusions

AVO analysis requires flat gathers. There are times flatness may not be achievable with analytical curves. We have described a brute force (non physical) gather flattening method that is based on tracking events at each t0 time. This method we described has been successfully used on many types of data including prestack time and prestack depth migration gathers and provides consistently flat gathers especially after external processing of the moveout curves in the inline and crossline directions. Here we gave examples of some of such field applications. It is expected that gather flattening can also be applied along with de-stretching done in angle gather domain (Lecerf et al., 2007) for improved quality results.

Acknowledgments

We thank Marathon Oil Company for show rights on the North Sea example, EnCana Corporation for show rights on the Eastern Canada example, and CGGVeritas management for allowing us to publish this paper.

References

Gulunay, N., H. Hoeber, F. Gamar, M. Dyce, C. MacKenzie, and, D. Whitcombe (2007a) Robust Residual Gather Flattening, 77th SEG Annual International Meeting.

Gulunay, N., M. Magesan, and, H. Roende (2007b) Gather Flattening, The Leading Edge, December 2007 issue.

Hinkley, D., G.W. Bear, and C. Dawson (2004) Prestack gather flattening for AVO, 74th SEG Annual International Meeting.

Lecerf, D., P. Herrmann, G. Lambare, JP Tourre, and S. Legleut (2007) Common-angle processing using reflection angle computed by kinematic pre-stack time demigration, 77th SEG Annual International Meeting.