Gather flattening

Necati Gulunay, CGGVeritas, Cairo, Egypt
Mag Magesan, CGGVeritas, Houston, USA
Henrik H. Roende, Marathon Oil Company, Houston, USA

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

The need for such an application was shown by Hinkley et al. (2004) and a solution which they called "dynamic gather flattening (DGF)" was presented. The method in this paper is similar in that it is a brute force (i.e., nonphysical) gather-flattening method. We will refer to it as FLATN for brevity. How can we correct gathers which have residual, even abnormal, moveout? First, such a 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 estimate the moveout function by tracking the events (wavelet) across traces at each zero offset (t_0) time sample. Starting at the innermost traces and at time t_0, we track the event times, t, of the wavelet across the gather to get the moveout function. This can be done by cross-correlation of neighboring traces (the "two-trace correlation method") and integrating the static shifts so calculated.

In order to track an event, one needs to allow a correlation window of two traces to climb up and down as the event moves. So, cross-correlation of two nearby traces may not be centered at the starting time t_0 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.

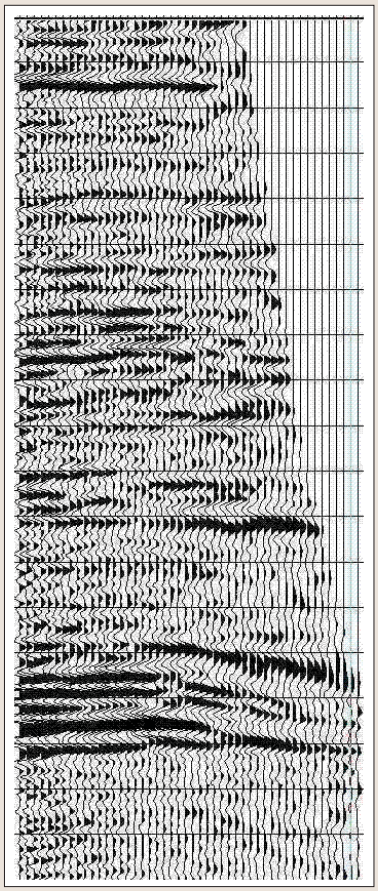

Figure 1. *A gather that could not be corrected during RMO application with analytical curves. The timing line increment is 100 ms.*

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

Figure 2a shows a simple (and noise-free) gather of parabolic events with large residual moveout values that we used while developing FLATN. Here, 120-ms time windows centered around each t_0 time were used during cross-correlation of two consecutive traces; a maximum trace-to-trace static of 12 ms in inner offsets and 36 ms maximum at far offset was used. Figure 2b shows the moveout function obtained with the two-trace, event-tracking algorithm on this gather. Red samples represent undercorrection (darkest = +291 ms) and blue overcorrection (darkest = –291 ms). Figure 2c shows the application of the moveout values to the input gather in Figure 2a. The result is nearly perfect. Moveout values can be smoothed across gathers in one or two spatial directions for lateral consistency as well as within a gather along time and offset axes for consistency

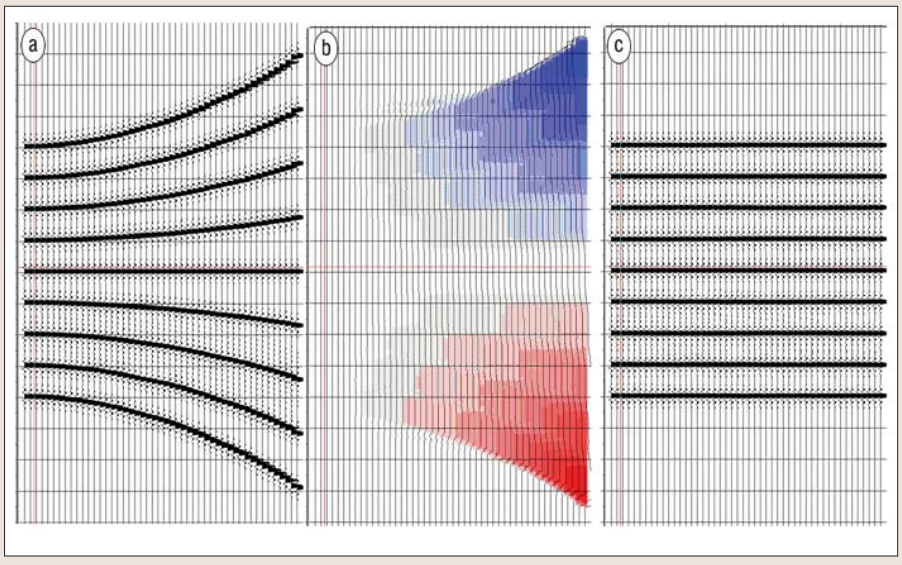

Figure 2. *(a) A simple model with extremely large moveout values. (b) Moveout map obtained by tracking events across the gather. Blue for overcorrection, red for undercorrection. (c) The gather in (a) after flattening with moveout values in (b).*

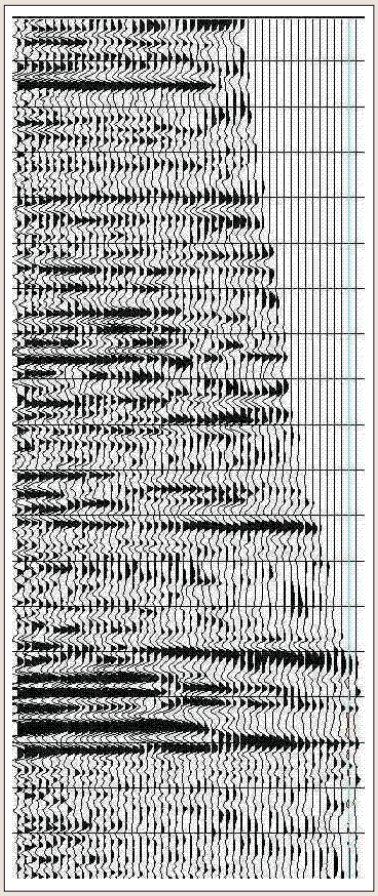

Figure 3. Same gather in Figure 1 after moveout correction that uses event tracking with two-trace correlation algorithm.

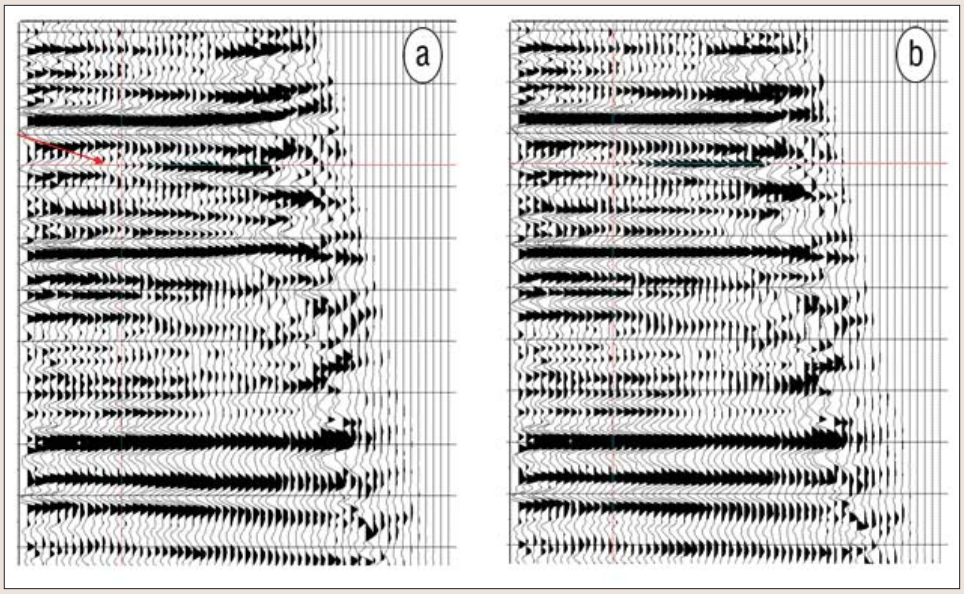

Figure 4. (a) A CMP gather from eastern Canada with Class 2 AVO behavior. (b) Same gather after flattening with event tracking with two-trace correlations.

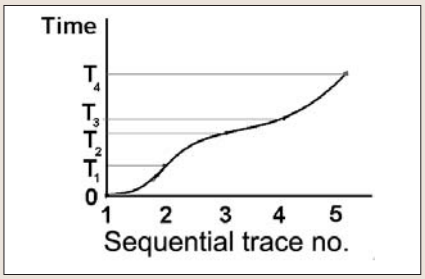

Figure 5. Definition of event times relative to first trace in the five-trace group.

TRACE	n-3	n-2	n-1	n	n+1	n+2	n+3	n+4	n+5
	$(dT1)_1$	$(dT2)_1$	$(dT3)_1$	$(dT4)_1$					
		$(dT1)_2$	$(dT2)_2$	$(dT3)_2$	$(dT4)_2$				
			$(dT1)_3$	$(dT2)_3$	$(dT3)_3$	$(dT4)_3$			
				$(dT1)_4$	$(dT2)_4$	$(dT3)_4$	$(dT4)_4$		
					$(dT1)_5$	$(dT2)_5$	$(dT3)_5$	$(dT4)_5$	
						$(dT1)_6$	$(dT2)_6$	$(dT3)_6$	$(dT4)_6$
				4 fold	4 fold	4 fold	...		

Figure 6. Four independent measurements done with moving five-trace groups for the static shift between a pair of traces.

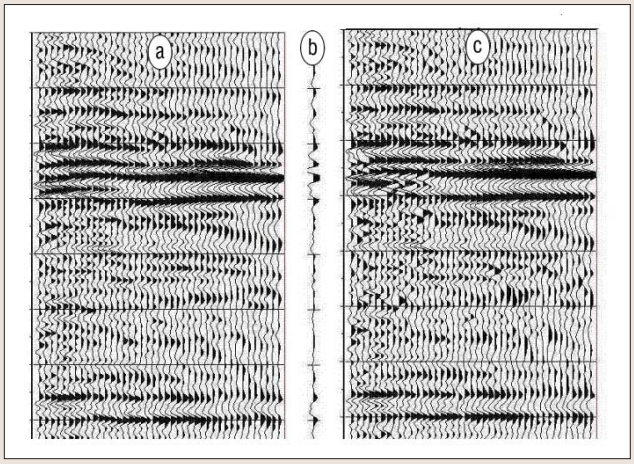

Figure 7. (a) An input gather (b) full-offset stack, (c) FLATN run with the external stack trace.

within the gather. However, before they are processed by external means, we do some internal processing to make them more reliable.

As some noisy data can give erroneous moveout values causing a drift on the integrated event trajectory at a given t_0, it is important that time shifts between two traces are ignored when (a) the value of the statics shift exceeds the maximum trace-to-trace statics expected between a pair of traces and (b) when cross-correlation quality falls below a user-defined threshold. Cross-correlation quality here is defined in the usual way as the ratio of the peak cross-correlation value to the square root of the product of energy on trace one and trace two. This attribute has a value between zero and one. Setting a threshold for the acceptable cross-correlation value allows excluding a bad pick during event tracking. As a given trace pair should have a similar static shift when tracking starts at another nearby t_0 value, one may consider interpolating for poor quality picks along time axis. Similarly, it is possible that event tracking could get off track and end up with large implied residual statics for a trace. Such large moveout values could be flagged as unreliable.

Furthermore, one can impose a consistency check from trace-to-trace on the moveout values because we do not expect sudden jumps when gathers are the results of advanced stages in processing. A deviation check at each time slice over a few traces and editing out the deviant ones suffices. Finally, one can smooth moveout values along the time axis for each trace so that sudden stretch or squeeze does not take place. Figure 3 shows the result of event tracking with the two-trace correlation algorithm on the gather in Figure 1. The method flattens the gather well, although some stretch and squeeze is present However, these artifacts can be reduced if some precautions (to be discussed later) are taken.

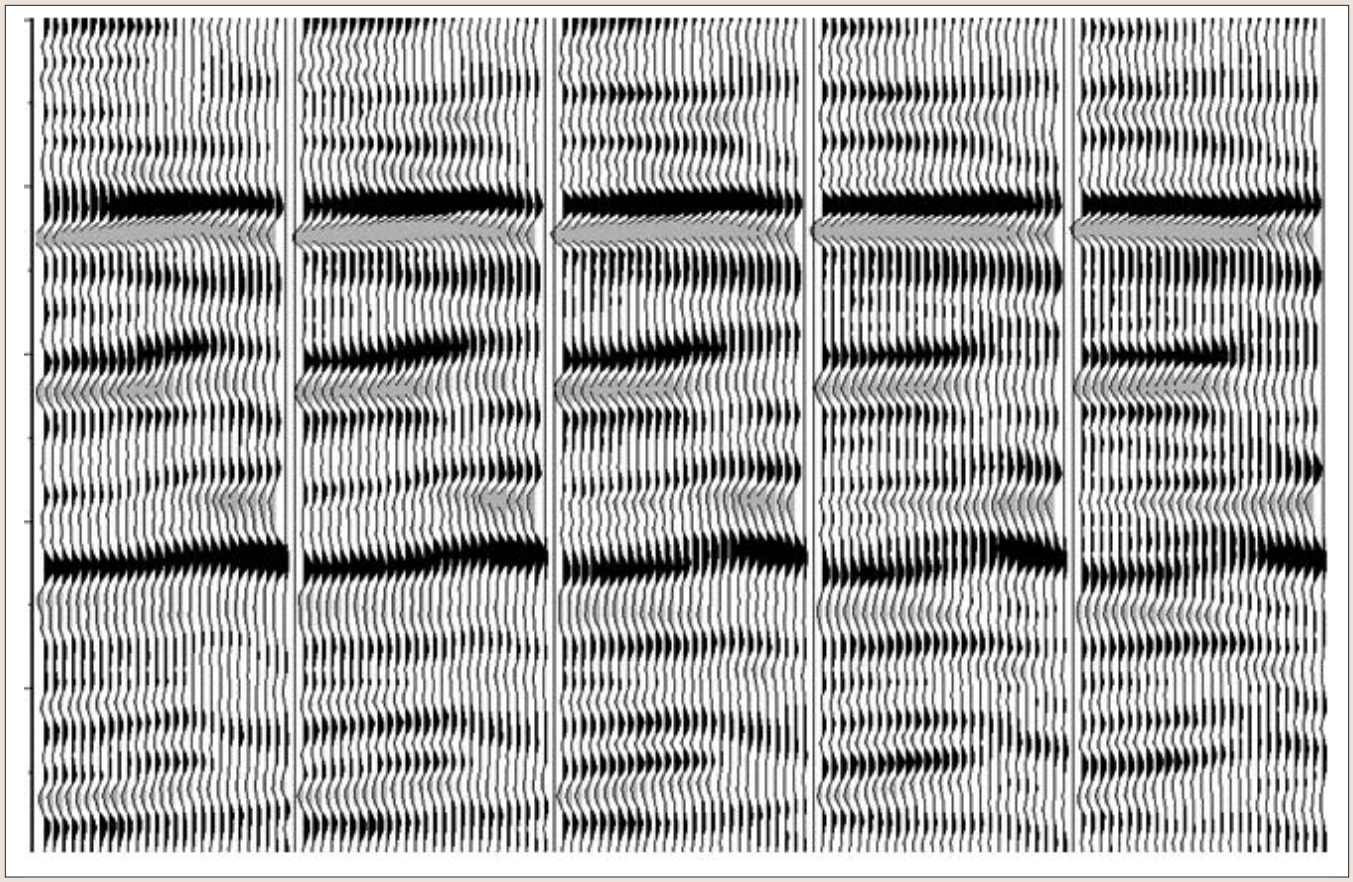

Figure 8. *Input gathers from North Sea.*

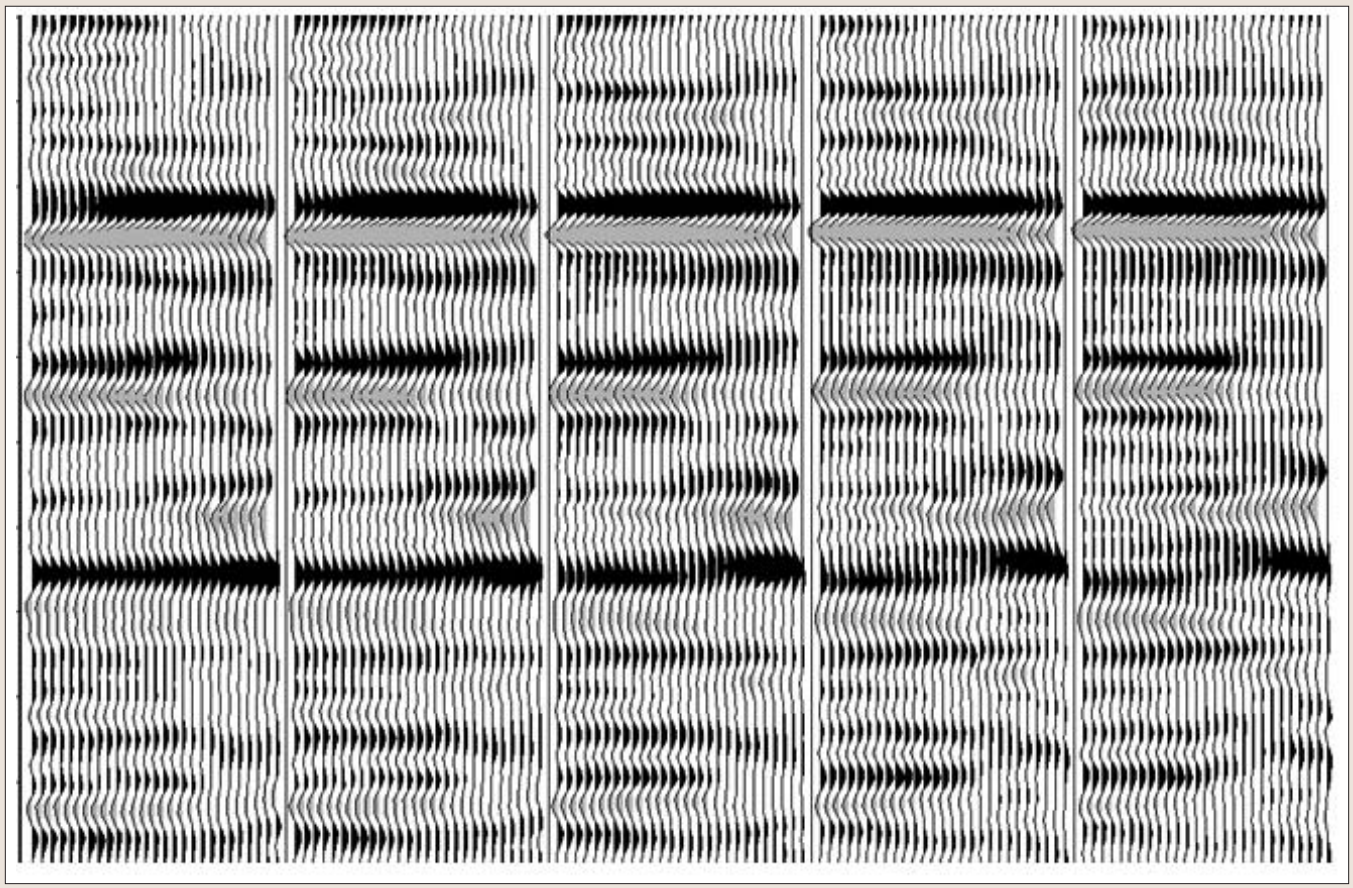

Figure 9. *FLATN run with the long period, internal-stack option on gathers in Figure 8.*

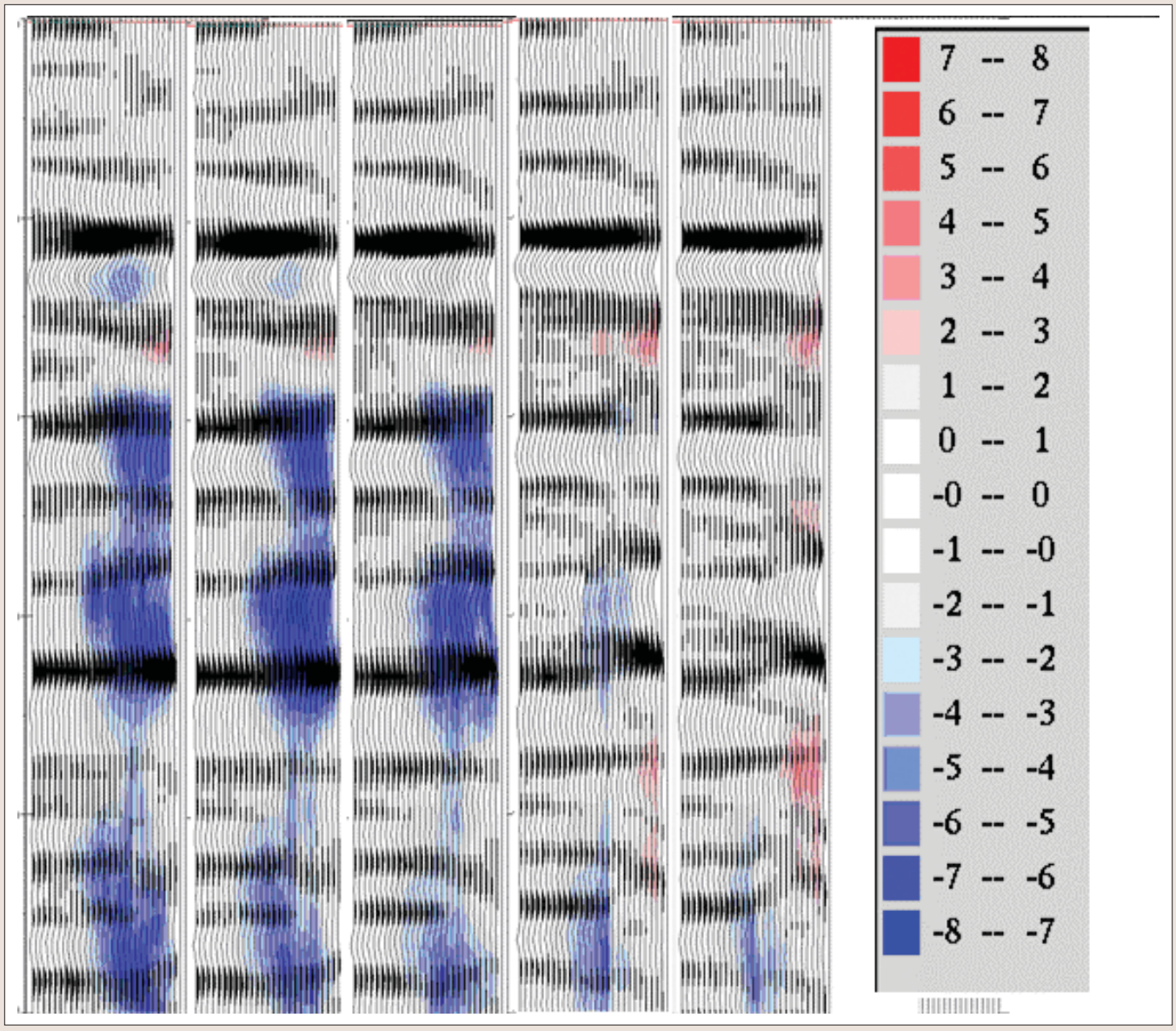

Figure 10. *Moveout-corrected gathers with the method presented in Figure 9 but with moveout values (as ms) indicated in color.*

Another application of event tracking with the two-trace cross-correlation algorithm and moveout correction is illustrated on a data set from Eastern Canada (Figures 4). Note that class 2 AVO behavior in the input is preserved in the processing (Figure 4b). This result was obtained by using a 60-ms time window for cross-correlations and an offset-varying maximum shift allowance (12 ms at minimum offset and 20 ms at maximum offset). We rejected correlation below 0.7, did a five-trace lateral coherency check, did not allow more than 4-ms deviation from the mean, and smoothed moveout values by a 40-ms box car filter along the time axis.

The moveout picking method described above flattens gathers well but, although it creates flat gathers and is very pleasing to the eye, it 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 in memory, 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 multigather processing scheme could be made user-controllable and turned off if external processing of moveout values is more desirable.

To bring stability into event tracking, one can examine a group of traces, with the center moving one trace at a time across each gather. The number of traces in the group is somewhat arbitrary. A large number of traces is obviously costly and may not bring 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 (Figure 5): 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. If $T_{i,j}$ represents time shift between trace i and trace j (i=1,2,...,5 and j=1,2,...,5) measured from their cross-correlation, we observe that these ten cross-correlation

values relate to the event times as follows:

$T_{1,2}=T_1$, $T_{1,3}=T_2$, $T_{1,4}=T_3$, $T_{1,5}=T_4$, $T_{2,3}=T_2-T_1$, $T_{2,4}=T_3-T_1$, $T_{2,5}=T_4-T_1$, $T_{3,4}=T_3-T_2$, $T_{3,5}=T_4-T_2$, and, $T_{4,5}=T_4-T_3$

As we have only four unknowns (T_1, T_2, T_3, and T_4) but ten equations, we can solve the equations in the least squares sense for these four unknowns, obtaining

$$T_1 = (2T_{1,2} + T_{1,3} + T_{1,4} + T_{1,5} - T_{2,3} - T_{2,4} - T_{2,5})/5$$
$$T_2 = (T_{1,2} + 2T_{1,3} + T_{1,4} + T_{1,5} + T_{2,3} - T_{3,4} - T_{3,5})/5$$
$$T_3 = (T_{1,2} + T_{1,3} + 2T_{1,4} + T_{1,5} + T_{2,4} + T_{3,4} - T_{4,5})/5$$
$$T_4 = (T_{1,2} + T_{1,3} + T_{1,4} + 2T_{1,5} + T_{2,5} + T_{3,5} + T_{4,5})/5$$

Relative shifts between traces are then calculated from this solution, i.e., $dT_1=T_1$, $dT_2=T_2-T_1$, $dT_3=T_3-T_2$, and $dT_4=T_4-T_3$ which are more reliable than the corresponding raw measurements, $T_{1,2}$, $T_{2,3}$, $T_{3,4}$, and $T_{4,5}$. Furthermore their reliability can be increased by averaging four independent relative shifts calculated as these groups of five traces pass a given trace pair four times (Figure 6).

There are times one may wish to align traces of a CMP gather to an external stack trace at that gather as was done by Gulunay (2007). Using a stack of the gather, and correlate traces against it, 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. The process is repeated for all t_0 time samples. If moveout values are not too large, this could guarantee far and near stacks match after gather flattening. If, however, the moveout is large, the external stack fits neither near nor far traces. In this case, a short offset stack (inner offsets) might be preferable and this can indeed be done internally (internal stack) by a certain percent of the inner offset traces. Figure 7 illustrates a run with the external stack. The stack trace in Figure 7b is a good representation of far offset in the input data (Figure 7a) but not so for its inner-offset traces. It is clear in this case that forming an external stack with full offsets is not a good idea unless moveout values are corrected to some degree beforehand. Hence, some undesirable shifts may get introduced (Figure 7c). It is obvious that external stack with full-offset range is not suitable for gathers like the one in Figure 2a.

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. 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. This forms the long-period option. After the long period is corrected, it should be possible to do a stack and correlate traces against it. This can be done internally or externally. We call this the long-period, internal-stack option. In this option, event tracking is done with the five-trace correlation scheme and the least squares static solution mentioned above. Figure 8 shows a set of gathers on which this method was applied. Figure 9 shows the same gathers after moveout correction with that algorithm. In this run, a correlation window of 40 ms was used for the shallow data and one of 80 ms for the deep data.

Maximum trace-to-trace static shift was kept at 12 ms during event tracking. Maximum final static was kept at 8 ms. Correlations of 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% of the traces from the 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, 24 ms temporal smoothing was used on the moveout values. Figure 10 shows the gathers in Figure 9 with the moveout values overlain.

Keeping the short-period static corrections only can have applications where, for example, one wishes to eliminate the jitter caused by neighboring traces of a given CMP that belong 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.

Conclusion. AVO analysis requires flat gathers but at times flatness may not be achievable with analytical curves. We have described a brute force (nonphysical) gather flattening method that is based on tracking events at each t_0 time. This method, which we call FLATN, has been used on many types of data, including prestack time- and prestack depth-migration gathers, and provides consistently flat gathers especially with external processing of the moveout curves in the inline and crossline directions.

Suggested reading. "Robust residual gather flattening" by Gulunay et al. (*SEG 2007 SEG Expanded Abstracts*). "Prestack gather flattening for AVO" by Hinkley et al. (*SEG 2004 Expanded Asstracts*). TLE

Acknowledgments: We thank Olivier Clauss, CGGVeritas (Kuala Lampur) F. Gamar, H. Hoeber, CGGVeritas (Aberdeen), and V. Durussel, CGGVeritas (Houston), for testing and feedback; T. Mojesky, CGGVeritas (Calgary), for encouragements and for obtaining permission to show the eastern Canada example; Marathon Oil Company for permission to publish the North Sea example; and finally CGGVeritas management for permission to publish this paper.

Corresponding author: necati.gulunay@cggveritas.com