Raypath interferometry for dummies: a processing guide

David C. Henley

ABSTRACT

The near-surface layer of the earth often causes serious degradation of seismic reflection images due to the irregularity of its thickness and composition. The effects include loss of signal bandwidth as well as phase/timing mismatch of specific reflection events between seismic traces recorded at neighbouring shot or receiver surface stations. In earlier work we have introduced interferometric methods to remove these effects, and have shown that what we term the 'raypath domain' is an effective one in which to work. We have demonstrated the methods on both synthetic and real data, but have not described the details. In this work, we present specific processing flows from the ProMAX processing environment and describe in detail how to apply raypath interferometry to a 2D seismic line.

INTRODUCTION

Raypath interferometry actually embodies two distinct and independent concepts, either of which can be applied on its own: raypath-consistency, and interferometry. The processing flows we have constructed are modular enough that while they are easily combined to apply the complete raypath interferometry method, they may be also be used independently. Our previous work is summarized in Henley (2006, 2007) and Henley and Daley (2007, 2008, and 2009).

Interferometry

Interferometry is a broad spectrum of techniques encompassing many branches of physics; but the application we discuss here is one of the simpler ones. In brief, we propagate a seismic imaging wavefield into the earth, to be reflected from rock layers and perturbed by passage through the irregular near-surface; and we compare it with a 'reference wavefield', ostensibly containing no perturbations, in order to characterize the near-surface irregularities and remove their effects from the propagated wavefield.

There are two parts to this method: we construct or otherwise obtain a reference wavefield; and we cross-correlate this reference wavefield with the imaging wavefield, create inverse filters from the cross-correlations, and apply the inverse filters to the imaging wavefield to remove the near-surface perturbations from the image. This particular implementation of interferometry we also refer to as 'statics deconvolution'. Some of the processing flows we present here are devoted to creating estimates of the 'reference wavefield', which we also refer to as 'pilot traces', and to obtaining inverse filters for deconvolving the raw seismic traces.

Raypath-consistency

For most of the history of seismic processing, near-surface corrections have been approximated by 'static' time shifts applied to entire seismic traces (the assumption of stationarity), and these 'statics' have been derived, for the most part, by assuming that the correction for all traces with a common surface location for source or receiver would be the same. This is the so-called 'surface-consistency' assumption; and it works well enough for a wide variety of situations, particularly when the average velocity of the near-surface earth materials is much less than that of the underlying layers (Figure 1). There are many real situations, however, where these conditions do not apply, and near-surface corrections are neither stationary nor surface-consistent (Figure 2). To accommodate these situations while still allowing surface-consistency when it is present, we introduced the more general concept of 'raypath-consistency', in which the near-surface corrections for all seismic raypaths originating or terminating at a surface location are the same *for a given raypath angle*. This means that instead of a constant time shift, or inverse filter to be applied to every trace associated with a particular surface location, the time shift or inverse filter will also vary with near-surface raypath angle.

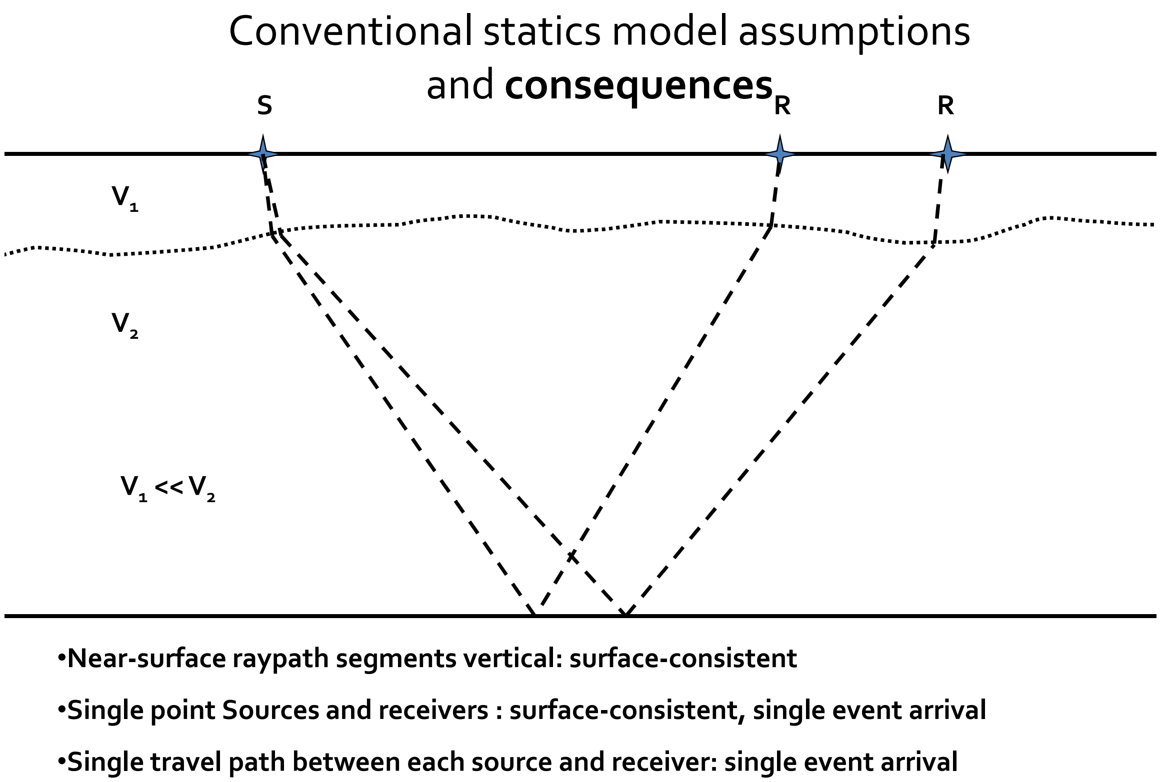


FIG. 1. Near-vertical raypaths in the near surface mean that all raypaths beginning or ending at a particular surface point will share a common near-surface delay, or static (surface-consistency). This means, as well, that all events recorded with one source and one receiver share the same static (stationarity).

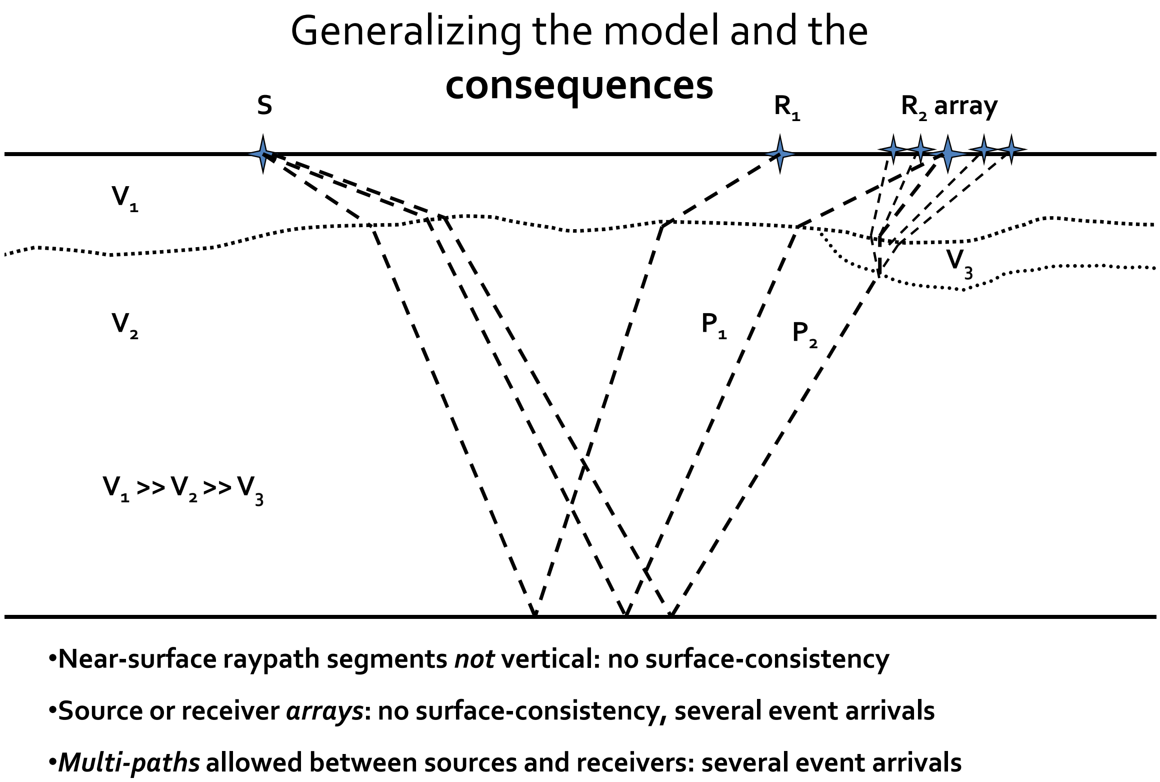


FIG. 2. When near-surface raypath angles are not constrained to be near-vertical by Snell's law, near-surface raypath segment lengths can vary with both reflection depth and offset, so surface-consistency and stationarity are both destroyed. Source and/or receiver arrays and multi-path arrivals mean that a single static is no longer the most appropriate correction for near-surface effects.

At first consideration, it seems that we've complicated the problem by introducing more variables. There is, however, a simple and convenient transformation of the raw seismic data traces that brings us quite naturally into the domain of near-surface raypath consistency—the radial trace domain. Figure 3 shows a raypath schematic for one trace from a seismic shot gather, in which we see that the near-surface raypath angle is different at both source and receiver location for every different reflector. Because of the way in which the data are remapped in the radial trace transform, however, a trace in the R-T domain (Figure 4) has a constant raypath angle at the source and the same raypath angle, in parallel, at each of the receivers contributing to the radial trace. Whereas an ordinary shot or receiver gather consists of a group of traces having raypath schematics similar to that in Figure 3, but with different shot-receiver distances; the R-T transform of a shot or receiver gather is a group of traces having raypath schematics similar to that in Figure 4, but with different near-surface raypath angles.

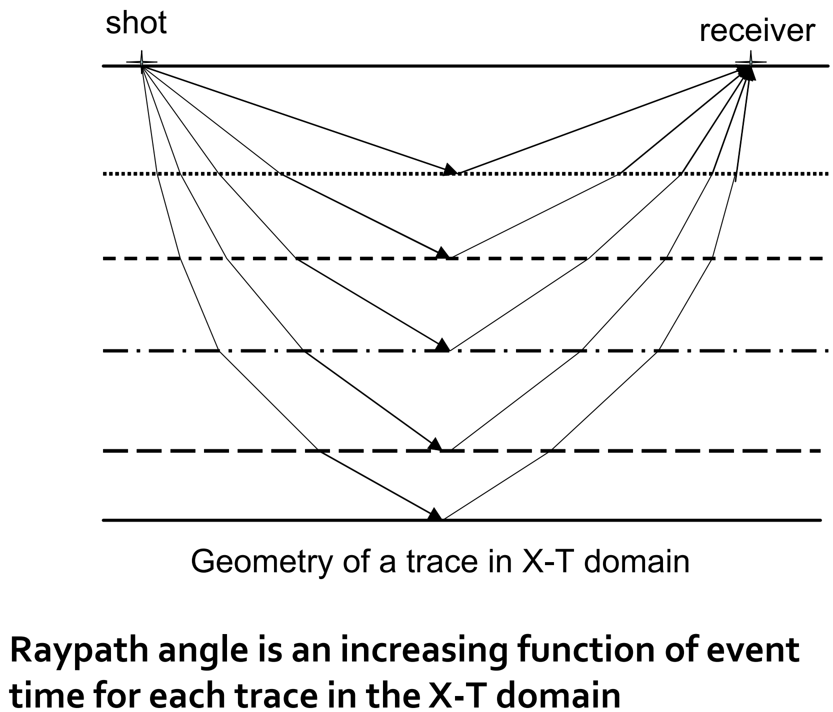


FIG. 3. Raypath schematic for a single trace in the ordinary X-T domain.

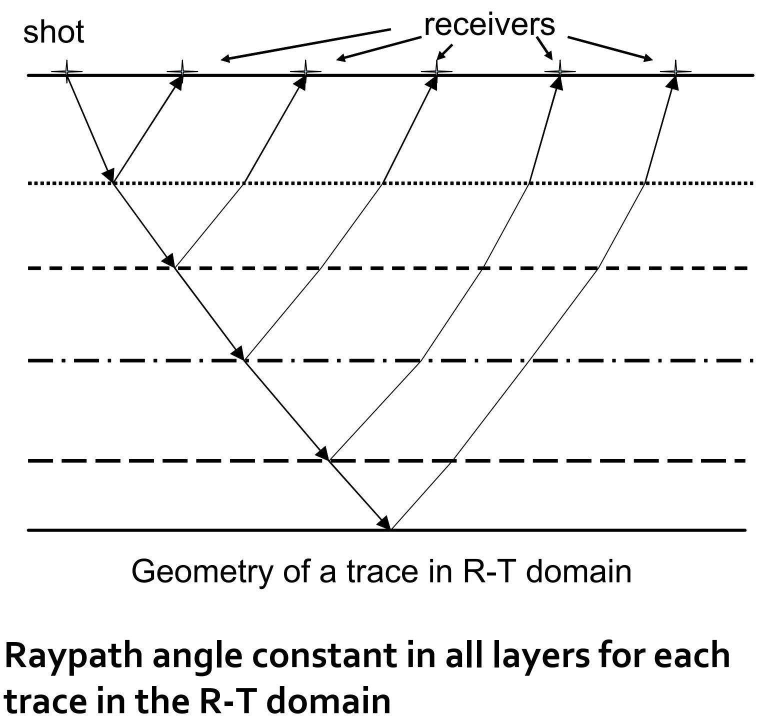


FIG. 4. Raypath schematic for a single trace in the radial trace (R-T) domain.

This leads very naturally to the concept of a common-angle gather, in which all the radial traces with the same near-surface raypath angle for an entire seismic line are sorted by surface location, as in Figure 5. This plot is analogous to a common-offset gather for conventional X-T data. Interestingly, raypath-consistent (or angle-consistent) statics show up on this gather as vertically aligned reflection event disturbances, while those which are surface consistent also show up as diagonally aligned disturbances, parallel to the apparent raypath angle for this particular gather. Figure 6 shows another common-angle gather, for a different raypath angle. Common-angle gathers associated with large apparent velocities (shallower raypath angles) contain only the shallow reflections, while those at small apparent velocities (steeper raypath angles) include deeper reflections, as well. Hence, if we use a full set of common-angle gathers for residual statics correction, the higher velocity gathers will yield solutions for the shallow reflections, and the lower velocity ones will solve for shallow and deep reflections together (or, with proper correlation windowing, just the deep reflections), leading naturally to non-stationary statics. The key to non-stationary statics is to derive and apply the near-surface (statics) solutions for each common-angle gather independently. Note that if the statics for a particular line are strictly surface-consistent, all the solutions for the different common-angle gathers will be very similar and redundant.

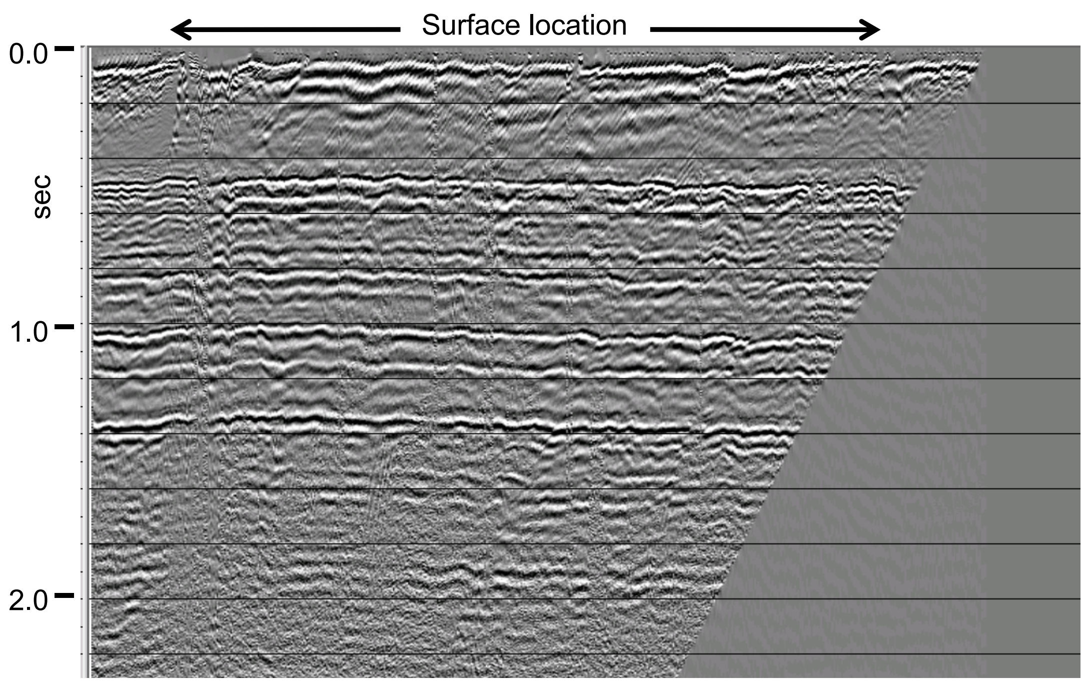


FIG. 5. Common angle gather for the apparent velocity of -890 m/s (apparent velocity is the 'angle' parameter). Raypath-consistent static disturbances line up vertically, surface-consistent ones line up diagonally, parallel to the raypath angle at the right edge of the live data zone.

Whether or not common-angle gathers are used to actually derive statics corrections, they can be useful diagnostics on their own; and they often demonstrate higher S/N than the original shot/receiver gathers from which they are derived.

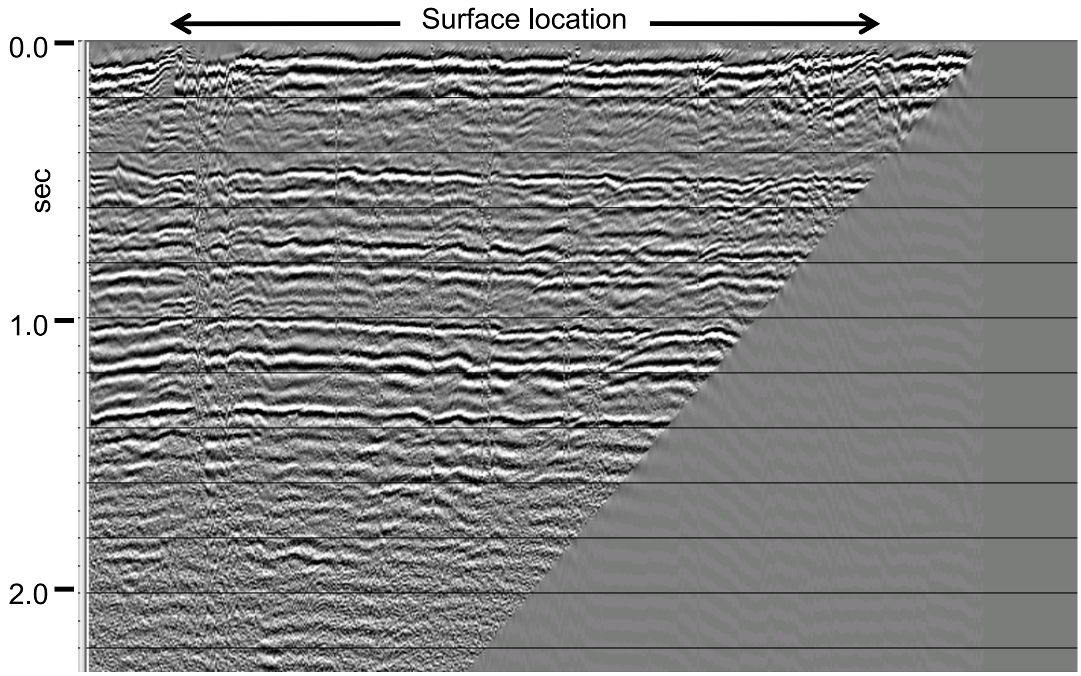


FIG. 6. Common angle gather for the apparent velocity of -1500 m/s. The higher apparent velocity means a shallower near-surface raypath angle.

While we've indicated that statics deconvolution and raypath-consistency are two independent parts of the raypath interferometry method, and can be used separately, the processing flows presented below demonstrate the entire raypath interferometry technique from start to finish.

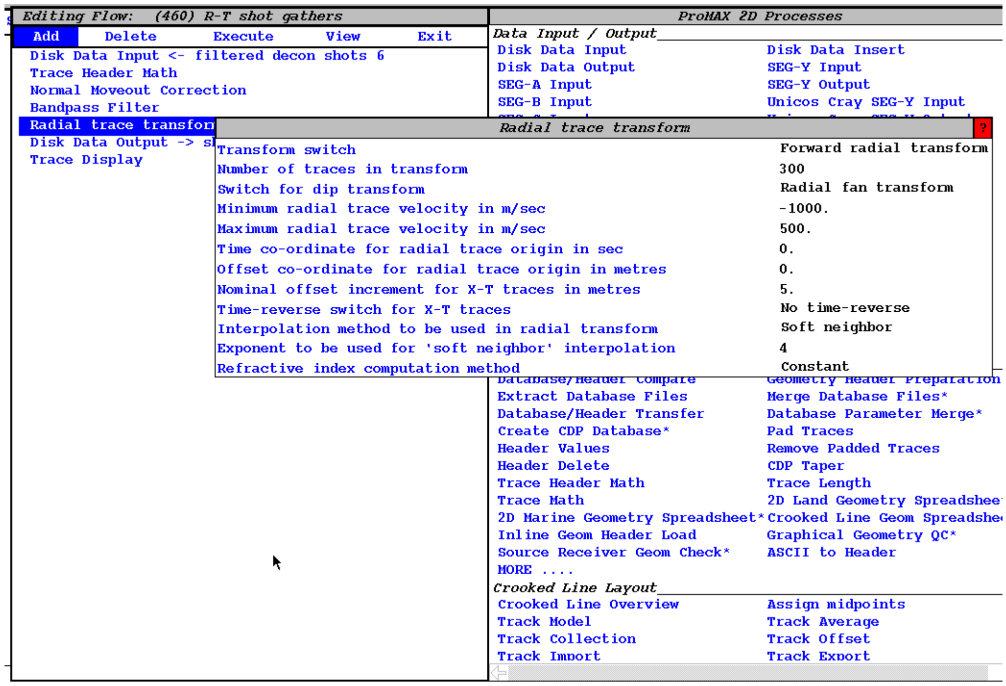
RAYPATH INTERFEROMETRY IN STEPS

Creating common-angle gathers

In preparation for raypath interferometry, the raw trace gathers (usually source gathers) of a seismic line should undergo the following rudimentary processing steps:

- Apply elevation statics.
- Attenuate any strong coherent noise (direct arrivals, ground roll, etc.).
- Deconvolve traces (to improve signal bandwidth).

While these steps aren't essential, they do tend to improve the results. The first step in creating common-angle gathers is to transform all the raw data gathers to the radial trace (R-T) domain (usually source gathers, since reflections are often better sampled in the source domain). The ProMAX processing flow shown in Figure 7 creates R-T source gathers from the input source gathers (in this example, filtered and deconvolved). The 'Normal Moveout Correction' shown in this flow need only use an approximate velocity function, since the objective is just to approximately flatten reflections. The moveout is restored to the data after the interferometry process.



Flow for reading shot gathers and transforming to the radial trace domain. Parameters in the RT transform are data-dependent. Minimum and maximum velocities should define a fan which captures most of the gather…number of traces should be at least 300-500 to avoid aliasing shot gather. Normal moveout correction need only use an approximate function

FIG. 7. A processing flow for creating R-T transforms from X-T shot gathers.

Figure 8 shows an example of a source gather and its radial trace transform. Note that the appearance of the reflections is largely unchanged by the transform. Note, as well, that the R-T transform usually contains many more traces than its original source gather, in order to avoid aliasing, and to increase the redundancy of the resulting angle gathers. Although the transform in the illustrated flow only specifies 300 output traces, we would normally use at least twice as many. Instead of source-receiver offset, the horizontal dimension of the R-T gather is apparent velocity (the angle of each particular radial trace).

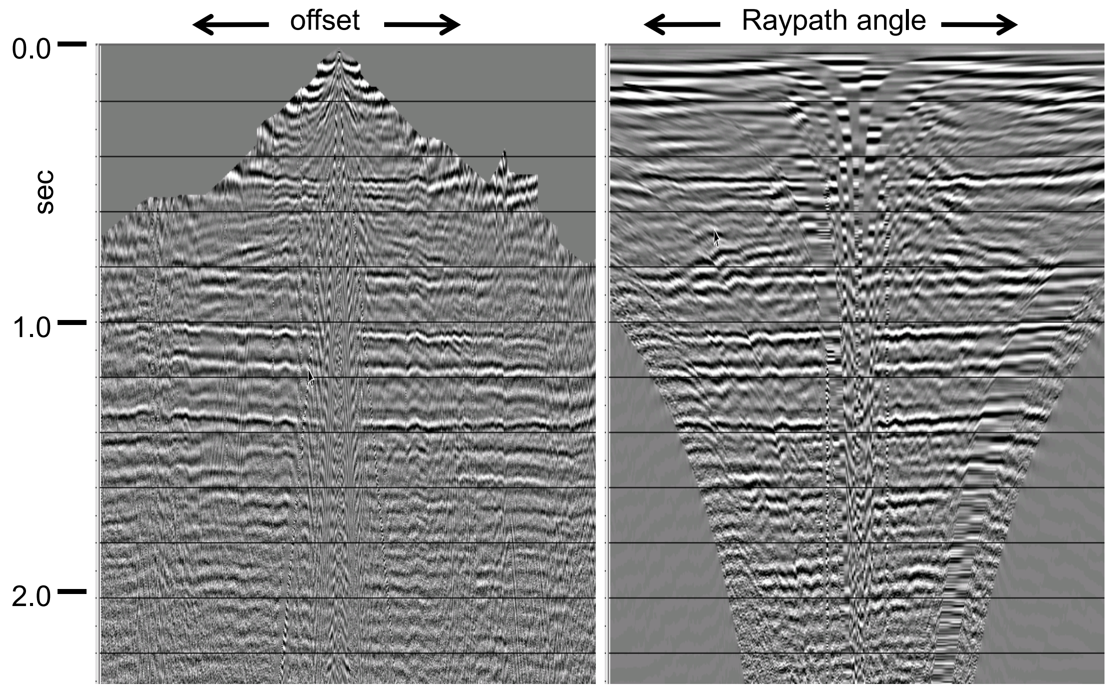
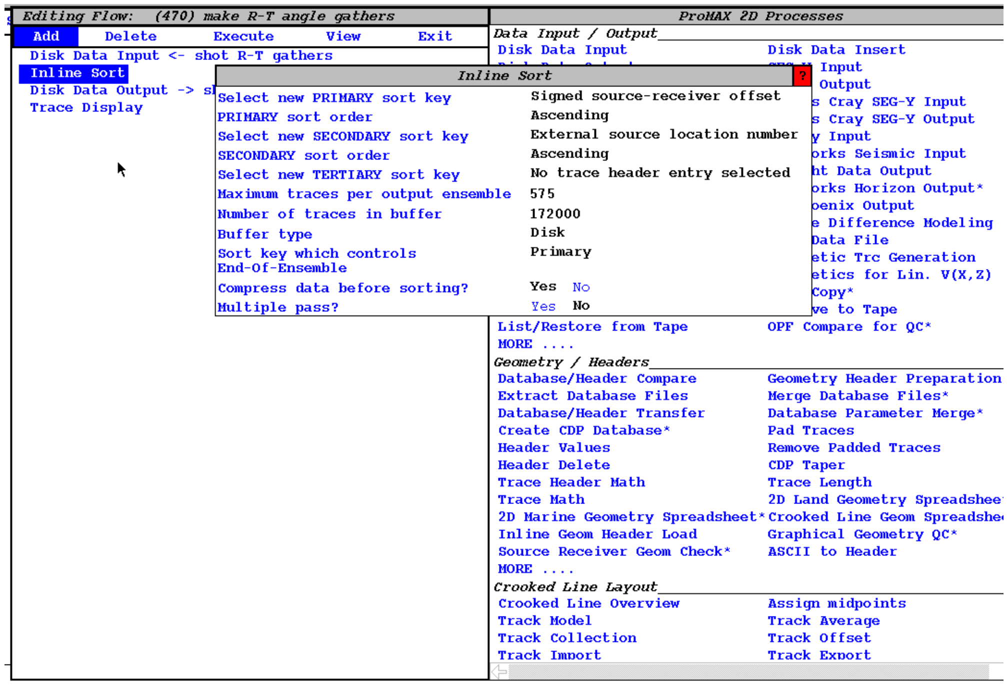


FIG. 8. Ordinary NMO corrected shot gather (left), compared to its radial trace transform (right). The same reflections can be easily identified on both gathers. Each trace in the original shot gather shows the energy recorded at one receiver for that shot. Each trace in the R-T transform, however, represents energy recorded from the shot into several receivers sequentially (see Figure 4).

Figure 9 is the flow which sorts the RT gathers into common-angle panels, which are analogous to common-offset gathers in X-T space. Although we show an 'Inline Sort' operation in this flow, the sorting can actually be done more quickly within the 'Disk Data Input' operation. Notice that the primary sort field is designated as 'Signed source-receiver offset'. The reason for this is that this trace header is used to carry the 'apparent velocity' in the radial trace domain. Figures 5 and 6 are examples of common-angle gathers, each corresponding to a different raypath angle, or 'apparent velocity'.



Flow for sorting radial trace gathers into 'constant-angle gathers'. The 'offset' header of each radial trace contains the apparent velocity used to gather the samples for that trace from the original X-T shot gather; so sorting by signed offset, then external source location creates a 'constant-angle gather' for each apparent velocity value.

FIG. 9. Processing flow for creating common-angle gathers from R-T transforms of shot gathers. Another term for common-angle gather is constant-angle gather.

Creating pilot traces

The task of creating pilot traces, or the 'reference wavefield' for interferometry can be done in many ways, and not every method works for every data set. What we illustrate here is an approach that worked well for a particular set of seismic data from the MacKenzie Delta, where statics were demonstrably not surface-consistent. In general, pilot traces are created by averaging together various groups of raw input traces in order to capture the common character of the events while attenuating random noise and averaging out the misalignment between traces. One way to do this, illustrated here, is to pick one or more horizons on the brute stack, then to use the horizon picks to flatten the events on individual trace gathers so that they can be enhanced by trace mixing to form pilot traces for use with the raw traces of the input gathers. Figure 10 shows our MacKenzie Delta example with two picked horizons visible. As can be seen, the horizons are picked simply with an eye to aligning the respective events for later smoothing.

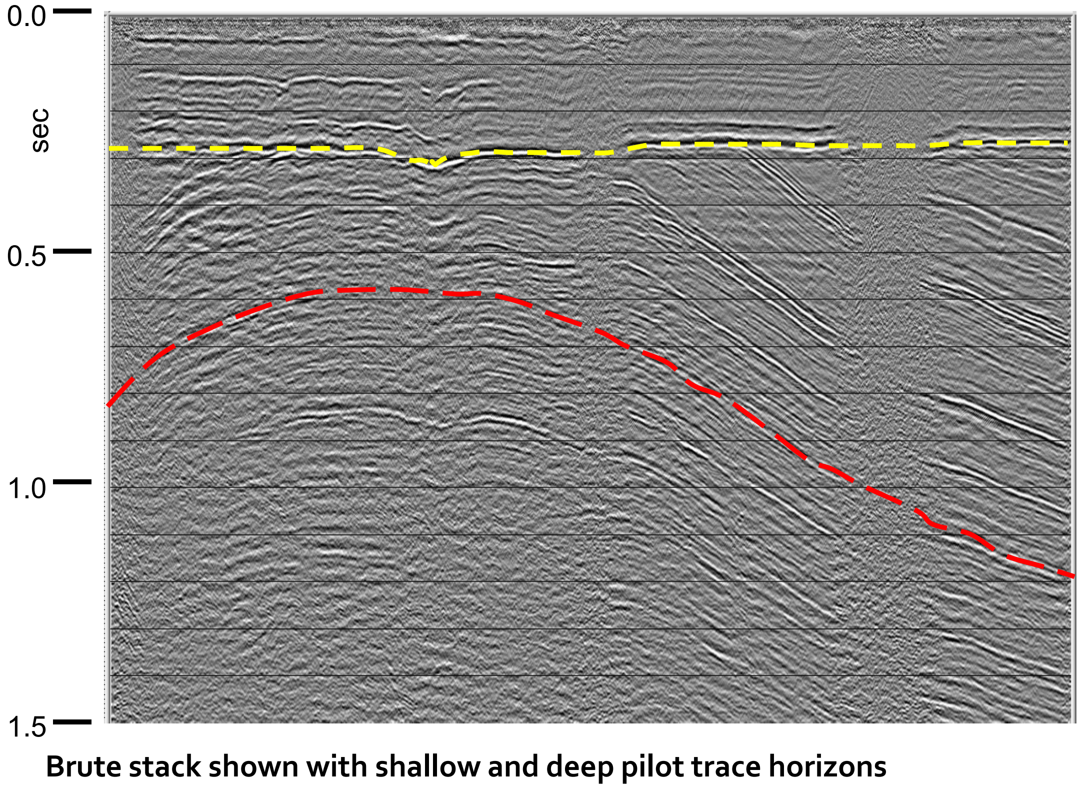
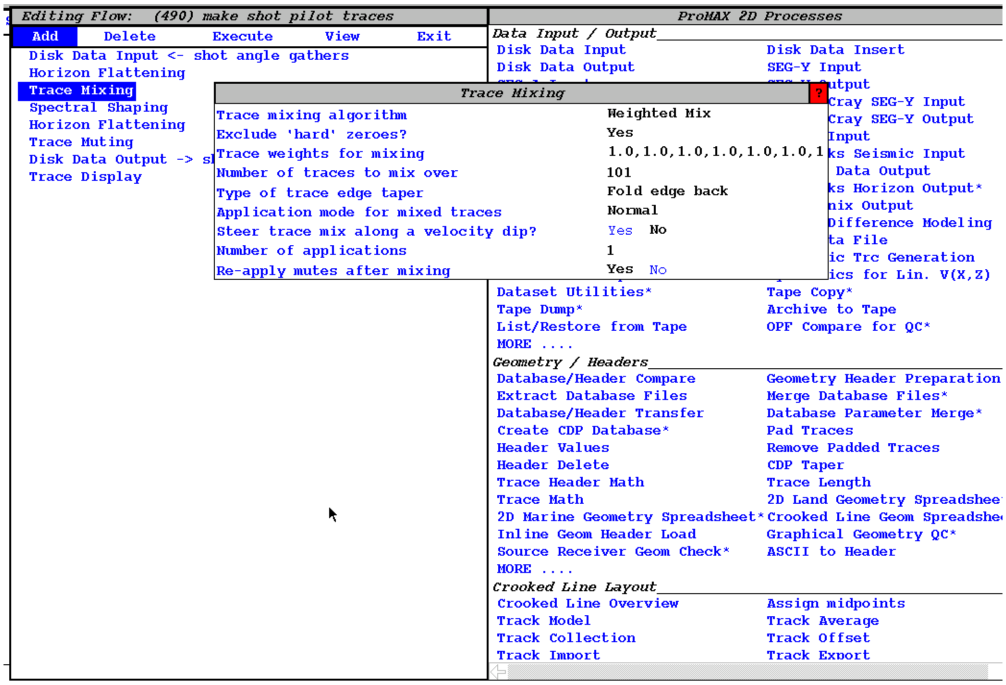


FIG. 10. MacKenzie Delta brute stack, with two picked horizons used to guide the smoothing used to create pilot traces from common-offset gathers.

Figure 11 displays a processing flow for applying the horizon picks to individual gathers (in this case, common-angle gathers). The first 'Horizon Flattening' operation applies a set of horizon picks as time shifts to roughly align the traces in each gather; the 'Trace Mixing' creates the pilot traces, and the second 'Horizon Flattening' removes the flattening from the pilot traces. 'Spectral Shaping' is used to optionally broaden the band of the pilot trace events. The 'Trace Muting' operation is used to mute portions of the input gather that do not actually conform to the picked horizon used for flattening (pilot traces are muted below the yellow horizon when aligned using the yellow horizon, and muted above the yellow horizon when aligned using the red horizon).

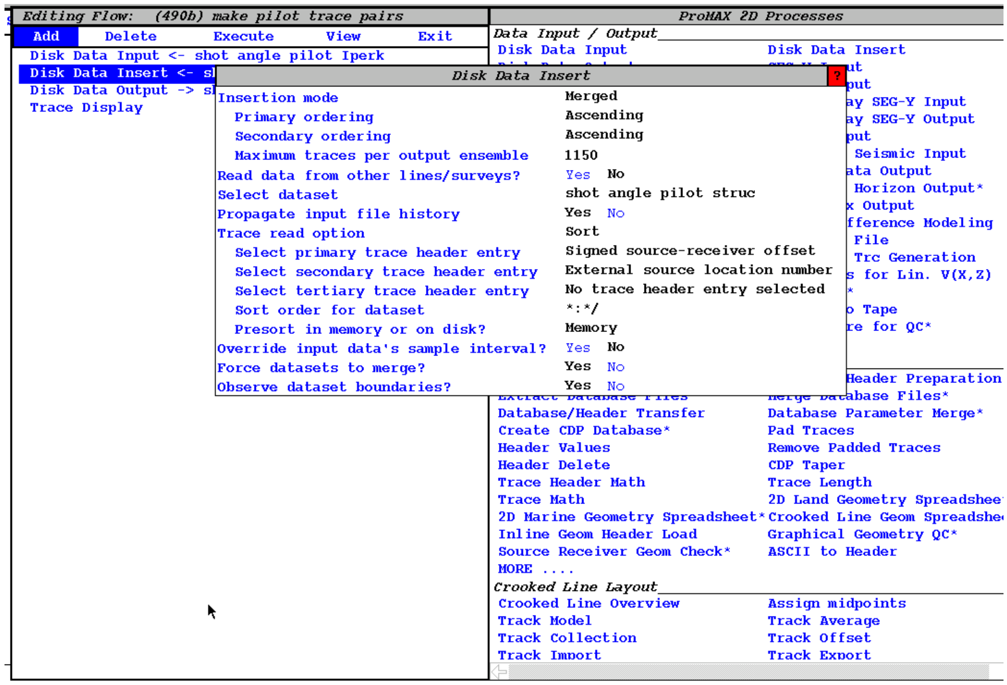


Flow for creating pilot traces for one picked horizon from constant-angle gathers. First horizon flattening applies horizon times, second removes them after trace mixing. Spectral shaping whitens pilot traces, trace muting zeros portions of pilot traces which do NOT conform to the picked horizon. This flow is applied once for each horizon picked on the brute stack to create sets of horizon pilot traces.

FIG. 11. A processing flow for creating 'pilot traces' using a picked reflection horizon. This flow processes all the common-angle gathers for a line, creating pilot trace common-angle gathers, one pilot trace gather for each input common-angle gather.

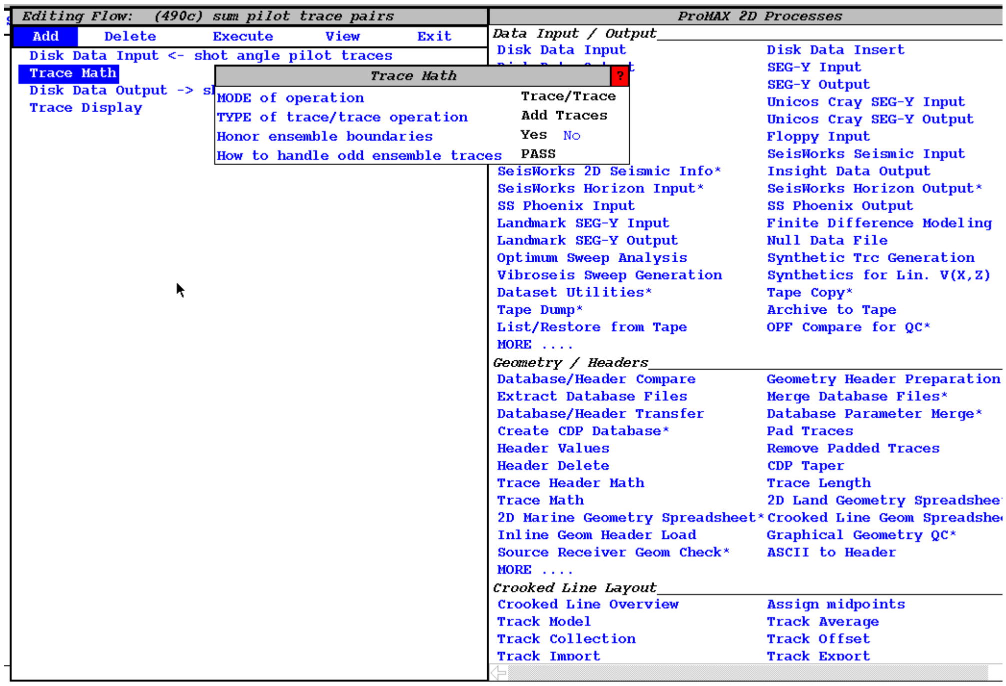
Preparing the correlation panels

When using more than one horizon to guide the creation of pilot traces by trace mixing, each horizon will result in a complete set of pilot traces, with portions of the traces muted. To combine the results from two or more separate horizons into a complete set of composite pilot traces, the separate files resulting from the processing flow in Figure 11 must be merged and summed. Figure 12 is the processing flow which merges and matches the pilot traces from two different horizons (each contained in a separate disk file created by the processing flow in Figure 11), and Figure 13 is the flow which sums the pilot traces from two different horizons to form composite pilot traces, as shown in Figure 14. If more than two horizons are used, the flows in Figures 12 and 13 must be used more than once, to incrementally merge and add the pilot trace segments from additional horizons.



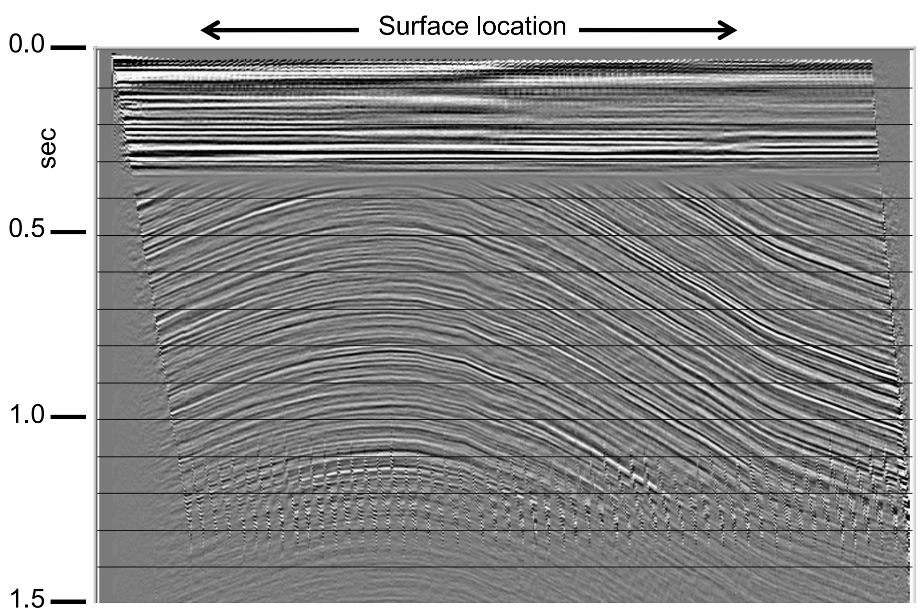
Flow to merge two sets of horizon pilot traces. The disk data insert adds a new set of horizon pilot traces, so that the output is a set of constant-angle gathers with two horizon pilot traces at every shot position. These traces will be summed in the subsequent flow to make composite pilot traces.

FIG. 12. A processing flow to merge two sets of pilot traces created by using the flow in Figure 11 to create two separate pilot trace files, one for each horizon. Since the trace headers for traces from the two files will be identical, they will be merged into pairs of traces with common headers, which can subsequently be summed by the processing flow in figure 13.



Flow to sum the horizon pilot trace pairs created by the previous flow. If more than two sets of horizon pilot traces are created, the previous flow and this one must be repeated for each new set of horizon pilot traces.

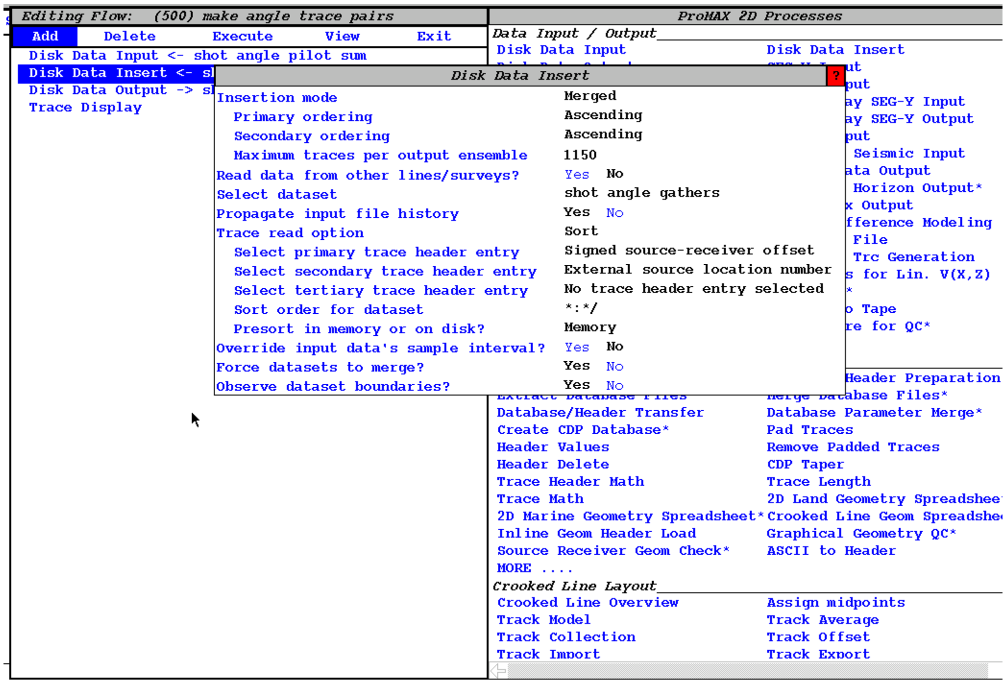
FIG. 13. This processing flow sums the adjacent traces of the pilot trace pairs created by the flow in Figure 12 in order to create composite pilot traces like those in Figure 14.



Composite pilot traces for constant angle gather at -231 m/s

FIG. 14. A pilot trace common-angle gather corresponding to the common-angle gather for apparent velocity -231 m/s. Every input common angle gather will have a corresponding pilot trace gather like this one

The processing flows in Figures 11, 12, and 13 each process all the common-angle gathers for an entire line, so each raw common-angle gather will have its own unique corresponding pilot trace panel. In order to prepare for the cross-correlation of raw common-angle gather traces with their corresponding pilot traces, the processing flow in Figure 15 must be run, to merge the separate input files and create pairs of raw and pilot traces, matched by surface location.

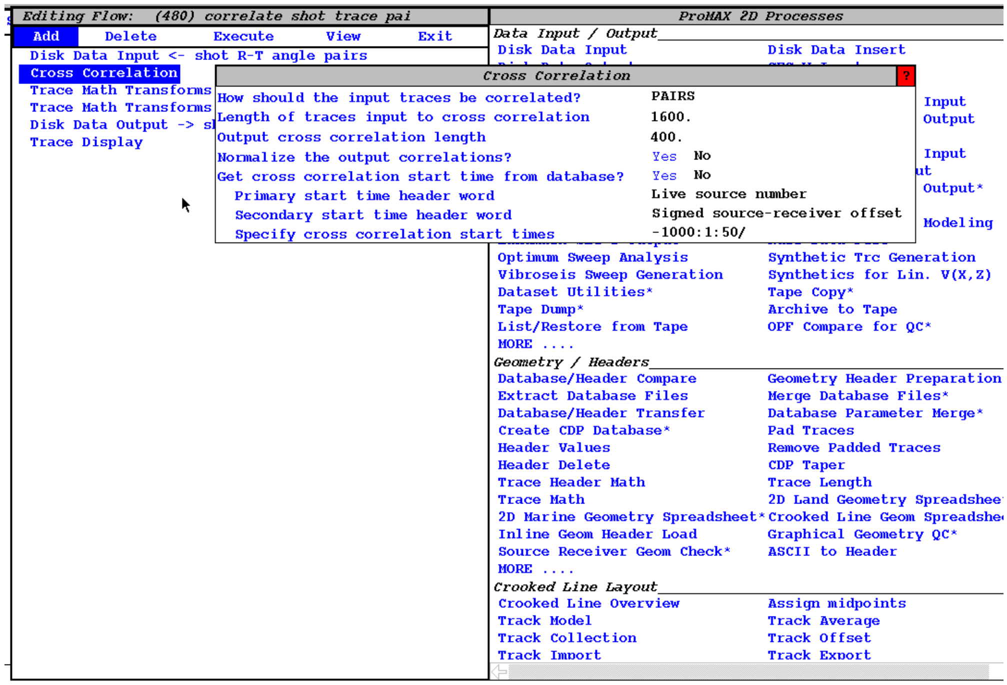


This flow takes the composite pilot traces from the previous flow and merges them with their corresponding raw traces from the constant-angle gathers to create trace pairs for correlation.

FIG. 15. This processing flow merges the corresponding traces from input common-angle gathers and pilot trace common-angle gathers to prepare for cross-correlation between each input trace and its unique pilot trace.

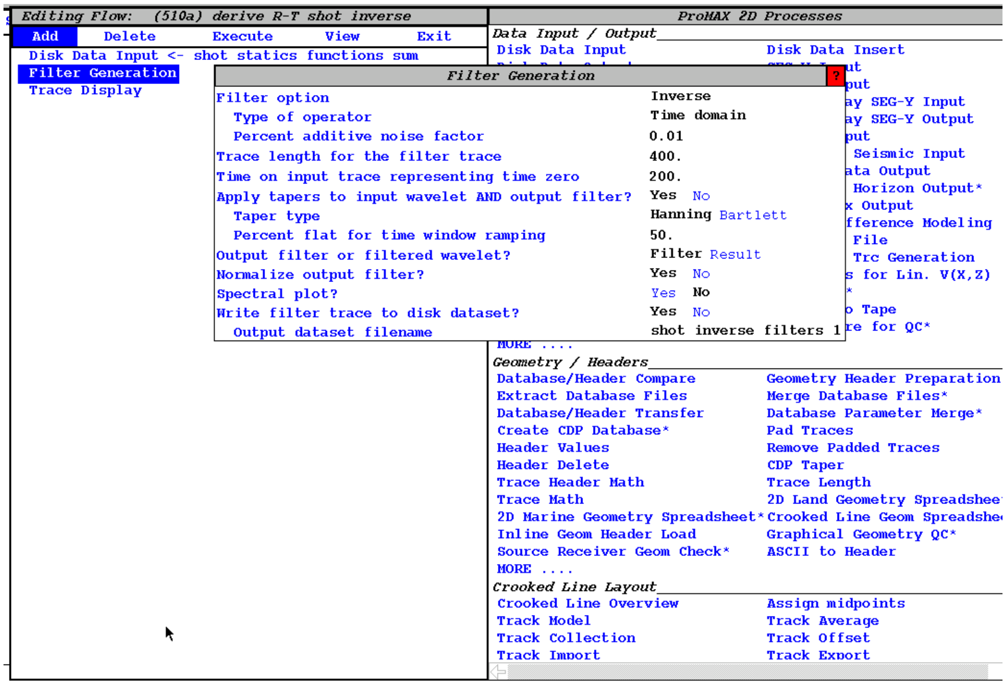
Cross-correlating and deriving inverse filters

Figure 16 shows the processing flow which performs the cross-correlations between raw and pilot traces and "conditions" the cross-correlation functions. The two 'Trace Math Transform' operations accomplish this conditioning by 1) raising the samples of each cross-correlation to an integer power (3, 4, or 5 work well), then 2) applying a Hanning window to the modified cross-correlation function. The conditioning has the effect of whitening the cross-correlation function without adding any new peaks, favouring the largest peak and emphasizing peaks nearest the zero cross-correlation lag. Note that the cross-correlation functions normally use most of the length of the input traces, excluding possibly the earliest parts. As well, the cross-correlation length should exceed twice the absolute value of the largest static expected in the data. The flow shown in Figure 17 simply derives an inverse filter for each conditioned cross-correlation function, to be used to deconvolve the static/phase shift captured by the cross-correlation. The length of the inverse filter is normally chosen to be the same as the length of the input cross-correlation function, in order to be able to correct the largest statics captured by the correlation functions.



This flow creates the 'statics distribution functions' used to deconvolve the constant-angle traces. The correlations use basically the entire input trace and its matching composite pilot trace, and the output correlation length is long enough to include any conceivable 'static'. The first trace math transform raises each sample to an odd power (often 5) to whiten the function without adding new peaks, while the second applies a Hanning window.

FIG. 16. This processing flow produces the 'conditioned' correlation functions used in the next step to derive inverse filters to undo the statics of each common-angle gather. The length of traces selected for the cross-correlation should include most of the length of the input traces for the common-angle gathers at the steepest angles, and the length of the output correlation should be larger than twice the largest possible static in the data. Start times for the correlations should avoid direct arrivals or early muting.



Although the output functions of the previous flow can be applied as 'match filters' to their corresponding raw constant-angle traces, this flow can be used to derive inverse filters, instead. The inverse filter option seems to give a broader band result; probably because the 'whitening' applied to the correlation function by raising samples to a power is rather modest.

FIG. 17. This simple flow derives full bandwidth inverse filters for the conditioned cross-correlation functions created by the flow in Figure 16.

Figures 18 and 19 show examples of the conditioned cross-correlation functions obtained. Most of the functions shown in these two examples are quite clean, with only small side-lobes; but some of the functions in Figure 18, particularly in the vicinity of the large statics deviations, exhibit more than one peak. This can indicate the presence of multi-path phenomena; but inverse filters derived from such correlation functions are perfectly capable of deconvolving the static and reducing the multi-path to a single arrival simultaneously.

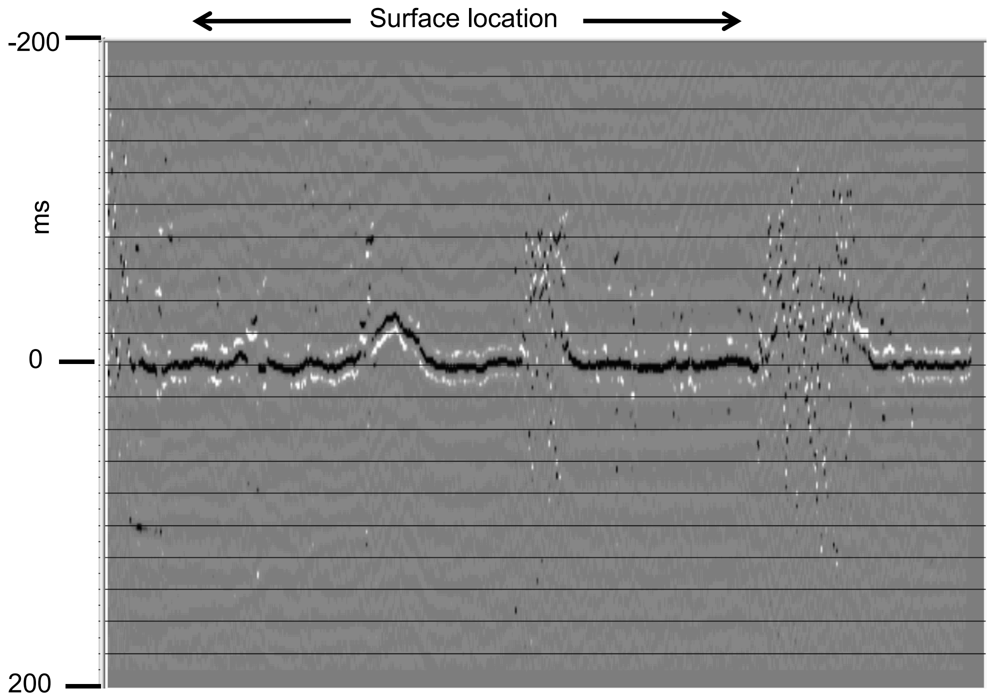


FIG. 18. A set of 'conditioned cross-correlation functions', or "statics functions" obtained for one common-angle gather (apparent velocity = -429 m/s) for the MacKenzie Delta data. These functions consist mostly of a central peak, with minor side ripples. Some functions in the vicinity of the large statics anomalies show more than one peak, indicative of multi-path phenomena.

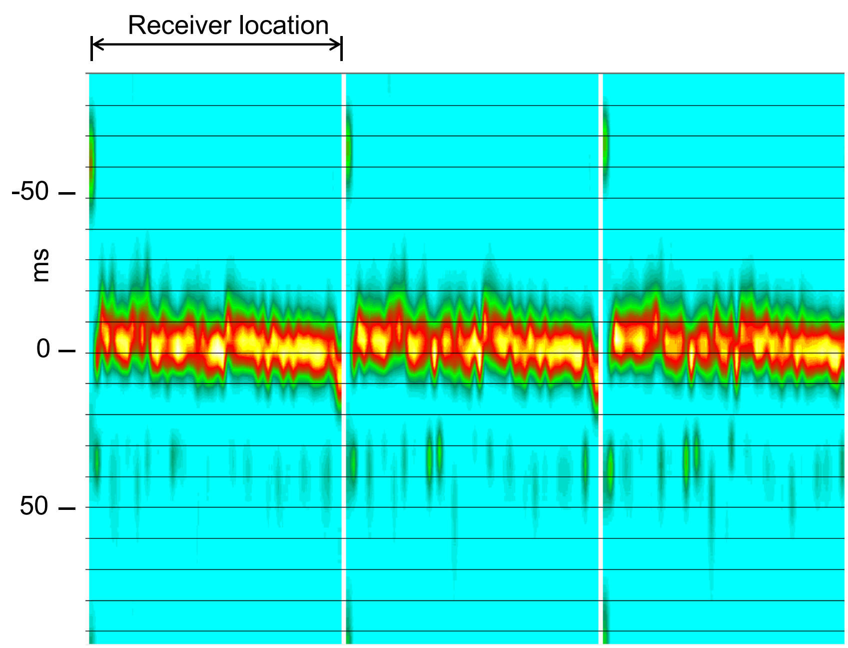
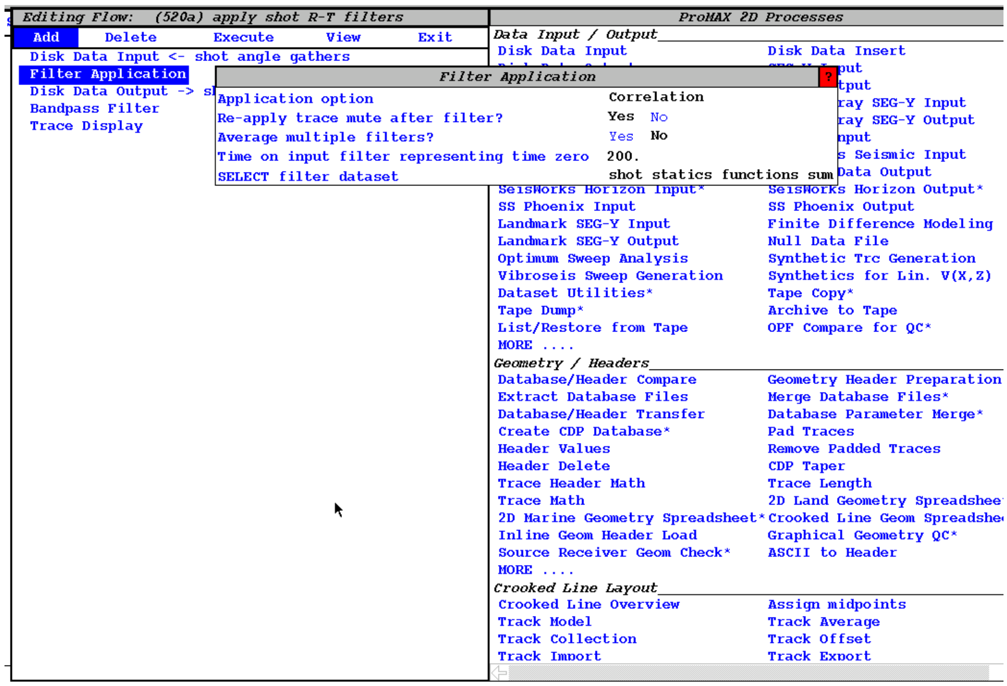


FIG. 19. Three sets of 'conditioned cross-correlation functions' corresponding to three different common-angle gathers. Note the similarity of the functions from panel to panel; apparently statics functions vary only slowly with angle.

Applying the inverse filters

The flow needed to apply statics deconvolution is shown in Figure 20. The flow shown actually applies the conditioned cross-correlation functions by ‘correlation’, but by re-setting the first parameter to ‘convolution’, the inverse filters can be applied instead, which usually results in results with broader bandwidth. Figure 21 shows the comparison between a raw common-offset gather and the ‘corrected’ gather after applying inverse filters.



This flow applies the match filters or inverse filters to the constant-angle gathers, trace-by-trace. If the shot statics functions are used, then filter application is by correlation; if inverse filters, then convolution. It is particularly useful to use the trace display to look at each constant-angle gather to judge the effectiveness of the filter application.

FIG. 20. A processing flow to apply the inverse filters derived from conditioned cross-correlation functions to the traces of the common-angle gathers. This particular example applies the conditioned correlation functions, themselves, by cross-correlation. To use the inverse filters, the ‘Application option’ parameter in the ‘Filter Application’ module would be set to ‘Convolution’.

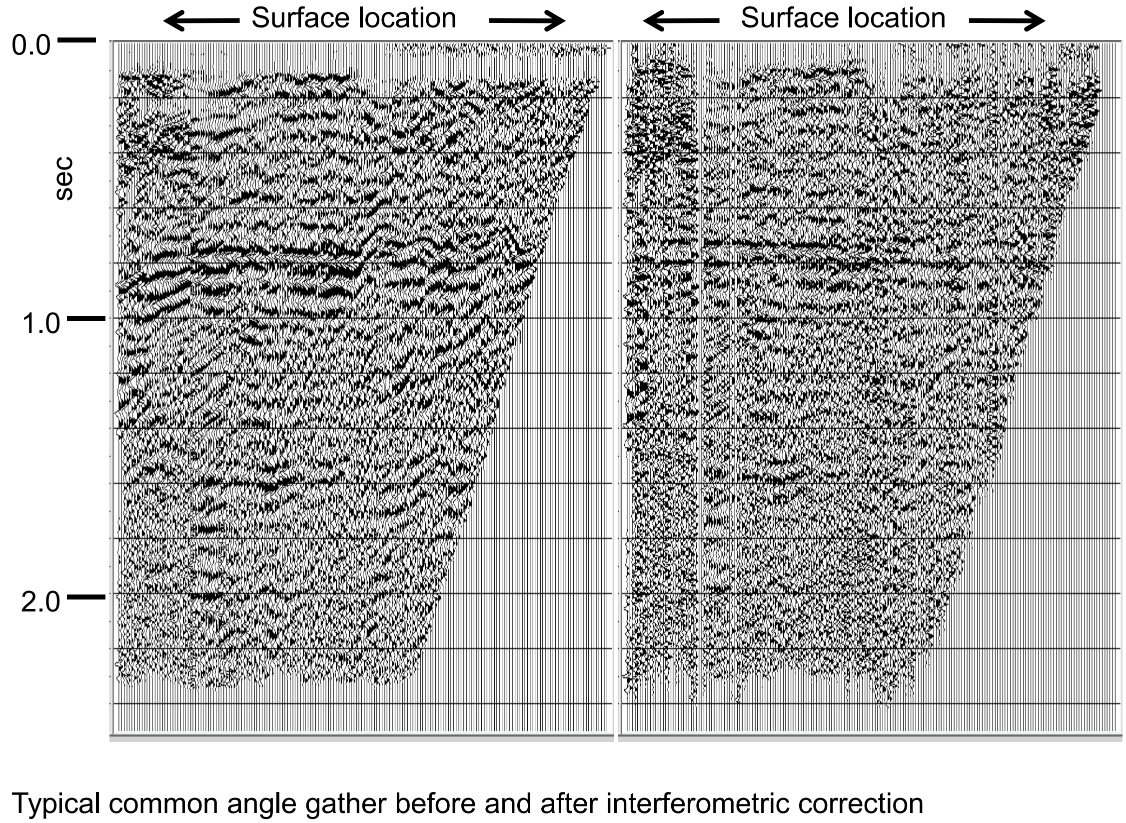
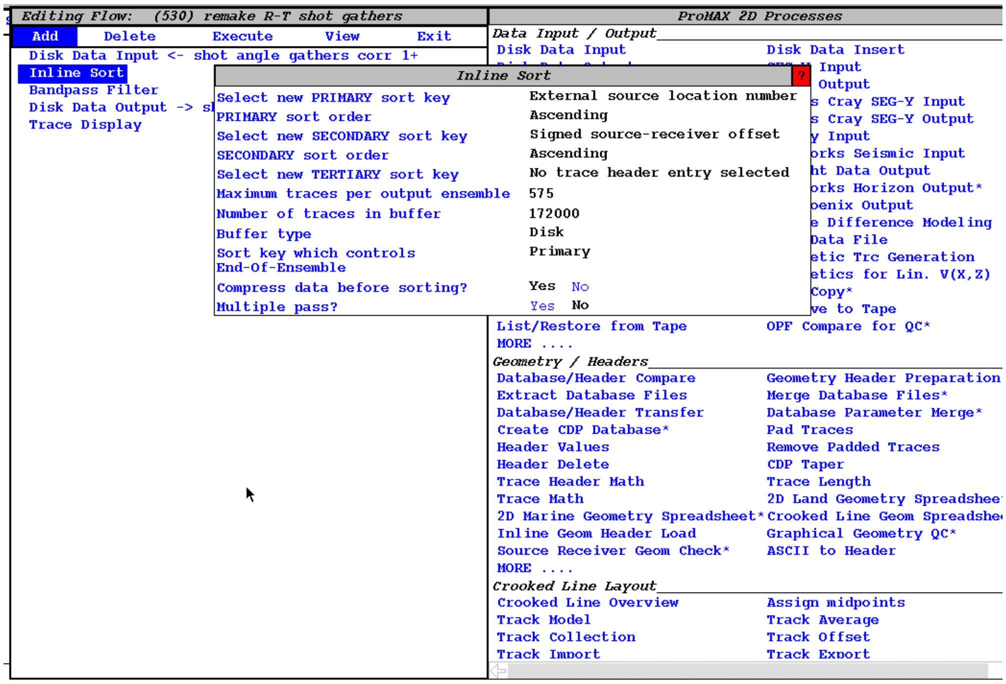


FIG. 21. A common-angle gather before (left) and after (right) being corrected by deconvolving the inverse filters derived from conditioned cross-correlations of the raw traces and pilot traces.The S/N of the deconvolved gather is sometimes less than that of the raw gather, but statics are improved.

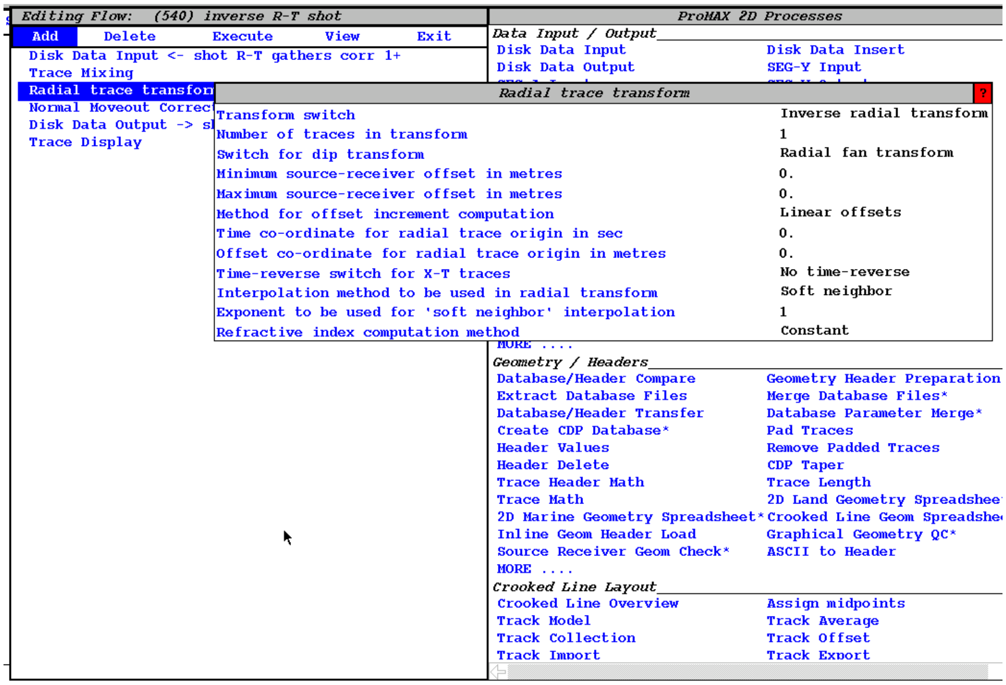
Inverting common-angle gathers to source gathers

Figures 22 and 23 show the flows needed to first re-sort the common-angle traces back to R-T source gathers, then to invert the R-T gathers back to X-T source gathers. As in the flow for creating the common-angle gathers, the actual sorting can be accomplished during the 'Disk Data Input', and will probably run faster. Note that 'Signed source-receiver offset' continues to carry the angle, or 'apparent velocity' until the data are formally inverted from the R-T domain in the next flow (Figure 23). In the 'Radial Trace Transform' operation, which is used to apply the inverse R-T transform, the parameters must be as shown in order to ensure a proper inversion to X-T, with the proper number of traces, correct offset headers, etc. The 'Normal Moveout Correction' operation restores moveout to the source gathers so that they may be treated like raw gathers and processed to CMP stack.



This flow sorts the corrected constant-angle gathers back into shot radial trace transforms.

FIG. 22. A processing flow for sorting corrected common-angle gathers to R-T source gathers for inversion back to the X-T domain. The sort can also be done in the 'Disk Data Input' operation, where it is usually faster.



This flow applies the inverse radial trace transform to obtain the corrected shot gathers. The parameters in the radial trace transform operation must be set as shown in order to properly invert the transform. The trace mixing operation is optional, but may be used sparingly (no more than 3 to 5) to improve redundancy of the corrections. Normal moveout correction removes the approximate function applied in the first flow.

FIG. 23. The inverse Radial Trace Transform flow, which restores the static-corrected data to the X-T domain. The parameters should be as shown, to properly invert the transform.

DISCUSSION

As can be seen, none of the processing flows needed to do raypath interferometry are complicated. We have chosen to break the process up into short, readily monitored steps, and to include a Trace display operation at the end of each flow, in order to visually monitor the operation. Observing the data, gather by gather, as it proceeds from raw X-T source gathers with visible statics down through the various stages of the raypath interferometric process can give the processor a more intuitive feel for the data and can help detect problems before processing the complete data set. As currently conceived, raypath interferometry will remain an interactive process, rather than being folded into a large ‘black box’ operation.

The processing flows used to produce pilot traces, merge them with raw traces, cross-correlate the traces, and apply the inverse filters can be run on ordinary source or receiver gathers without ever going to the R-T domain. Likewise, the flows used to create common-angle gathers can also be used to create the gathers for diagnostic purposes, entirely independent of raypath interferometry.

ACKNOWLEDGEMENTS

Thanks to CREWES sponsors and staff for support and discussion. Thanks to Shell Canada for use of their MacKenzie Delta data; and thanks to Pat Daley for discussion and consultation on interferometry.

REFERENCES

Henley, D.C., 2006, Application of raypath-dependent statics to Arctic seismic data, CREWES 2006 research report,**18.**

Henley, D.C., 2007, Raypath statics revisited: new images, CREWES 2007 research report,**19.**

Henley, D.C., and Daley, P.F., 2007, Connecting statics deconvolution and seismic interferometry, CREWES 2007 research report,**19.**

Henley, D.C., and Daley, P.F., 2008, Applying interferometry to converted wave statics, CREWES 2008 research report**, 20.**

Henley, D.C., and Daley, P.F., 2009, Hybrid interferometry: surface corrections for converted waves, CREWES 2009 research report, **21.**