Getting it right: source-receiver offsets in the radial trace transform

David C. Henley*
dhenley@ucalgary.ca

Introduction

The ***radial trace (RT) transform*** is a simple ***re-mapping*** operation introduced many years ago by ***Jon Claerbout*** to facilitate computations in wave-equation migration algorithms. It can also be used in ***other applications***, including the ***attenuation of coherent, source-generated noise***. We introduced our own RT transform in 1999, designed to process only ***2D data***. To simplify the inverse RT transform, and to introduce some diagnostic options, we implemented an ***approximation*** for restoring ***source-receiver offset*** values to the X/T trace ensemble. This did ***not*** require retaining a full set of X/T trace header values during the forward transform. Our shortcut was to place minimum and maximum offset values in unused RT trace headers, using these values to ***linearly interpolate*** the offsets to trace headers during the inverse.

This approximation is quite accurate, ***as long as the data are strictly 2D***; but for any trace ensembles for which the ***source position*** is ***not collinear*** with the ***receiver line***, source-receiver offset values are hyperbolically distributed, and the linear approximation is not appropriate.

Our RT filtering module, ***radfilt***, is not affected, since all RT operations are ***internal*** to the module. The X/T input ensemble, with its full trace headers, is always present to provide correct offset values for inversion.

Our RT forward/inverse transform module, ***radtran***, however, writes out ***RT trace ensembles*** to an ***external*** file, thus losing the input X/T trace headers for RT inversion. Hence, inaccurate, linearly interpolated offset values lead to data value mis-mapping during RT inversion.

We have repaired this problem by adding an option in which the inverse RT transform opens the ***database*** prepared for the current data set and retrieves the appropriate header values prior to the actual RT inversion.

We illustrate this modified ***radtran*** module here, and review the original diagnostic displays still available in the algorithm.

The old algorithm

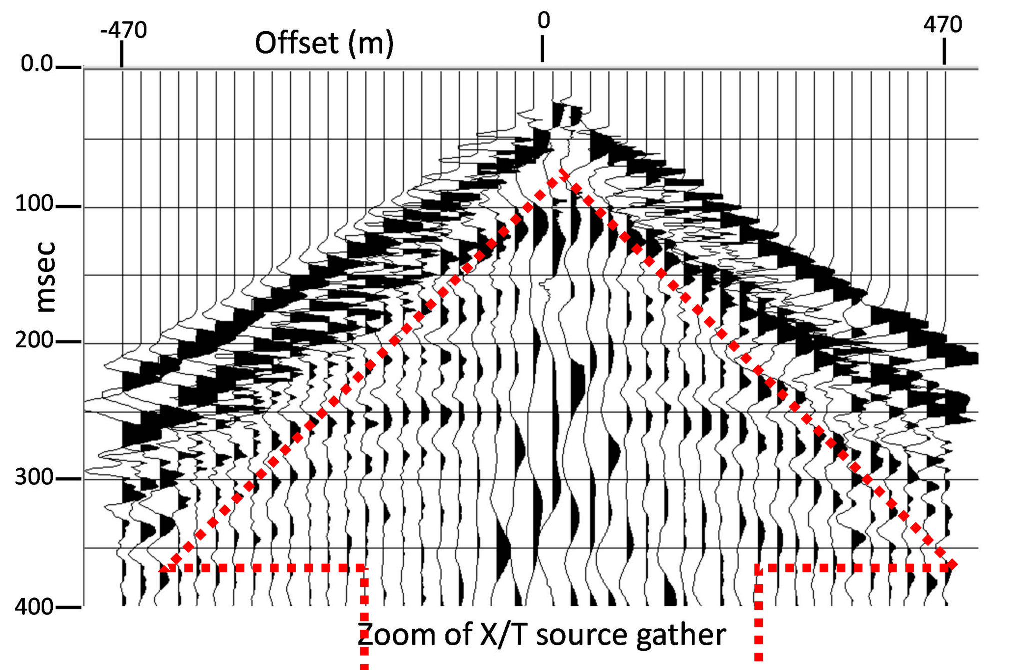

FIG. 1. Close-up of original 2D X/T source ensemble, collinear source.

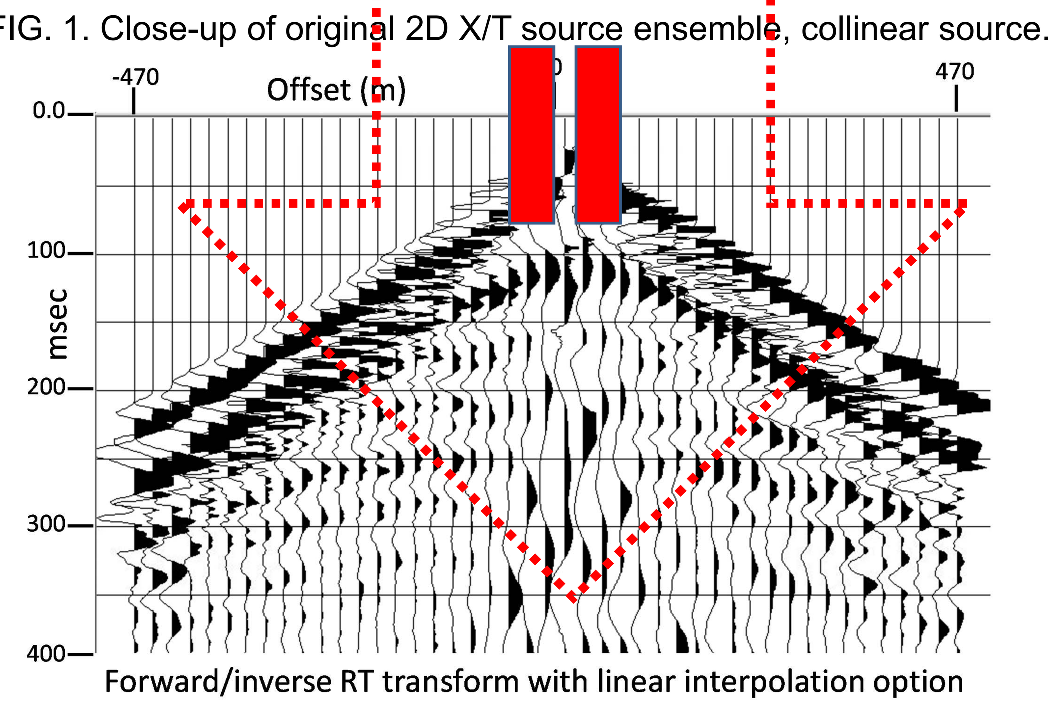

FIG. 2. 2D source ensemble after forward/inverse RT transform with ***linear offset interpolation*** (collinear source position).

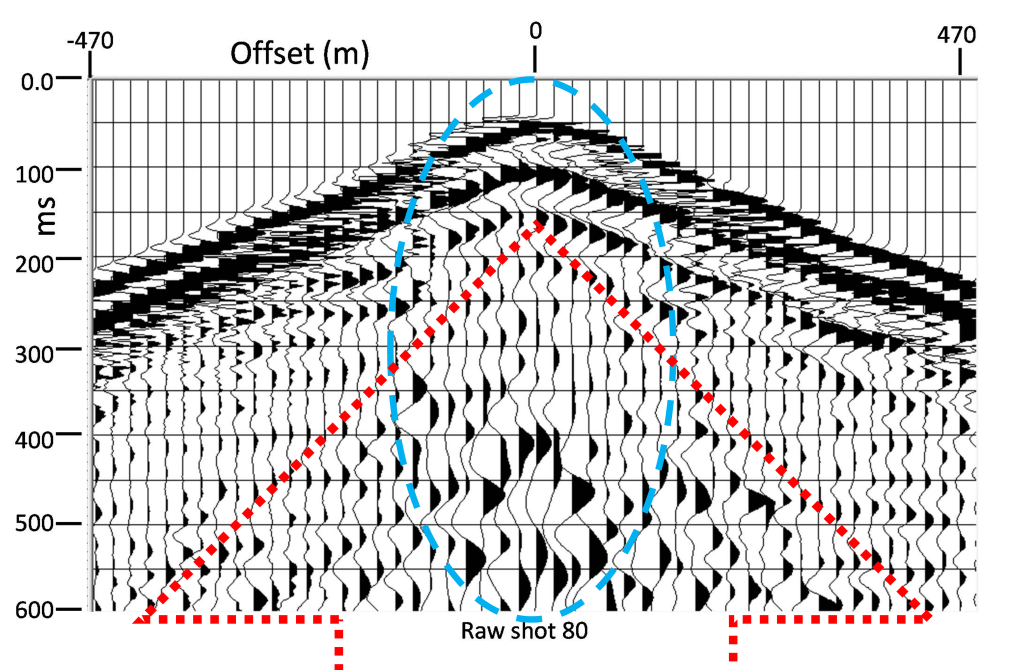

FIG. 3. Source ensemble with source position displaced by three stations laterally to the line—these data are ***not strictly 2D.***

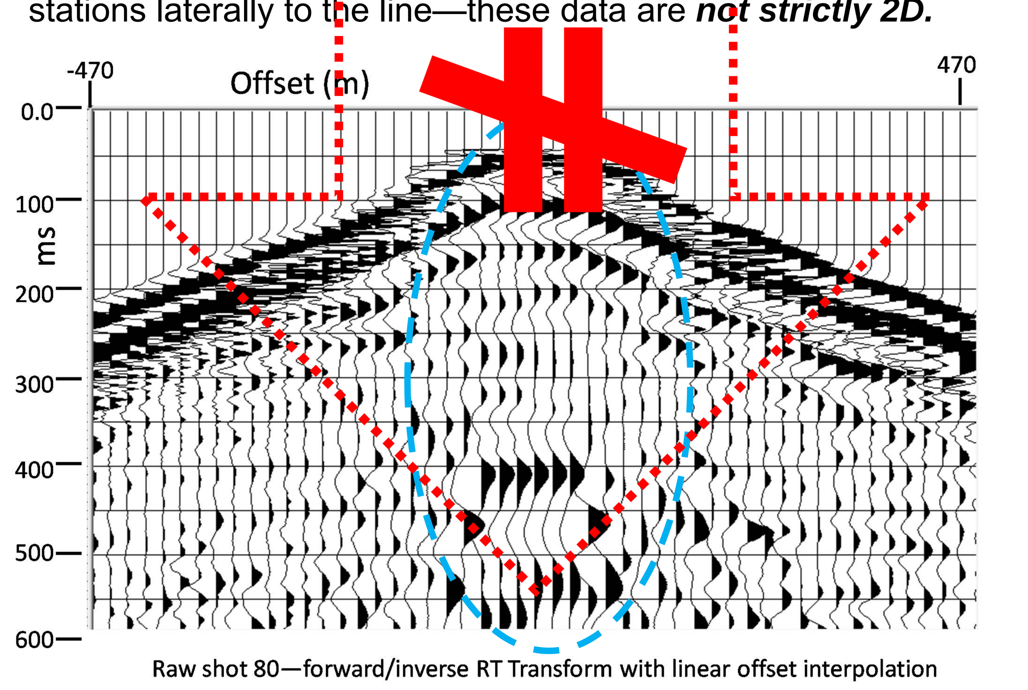

FIG. 4. Source ensemble with displaced source position after forward/inverse RT transform with ***linear offset interpolation***. ***Data distortion*** at nearest offsets is obvious...and ***unacceptable.***

The new algorithm

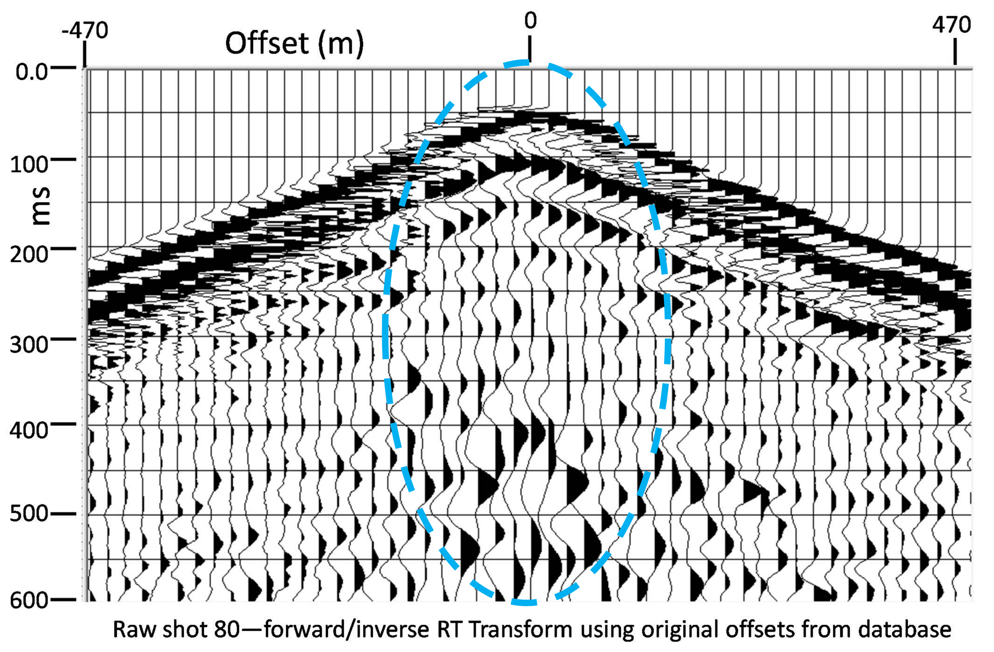

FIG. 5. Source ensemble with displaced source position after forward/inverse RT transform using original offset values retrieved from the database—***no data distortion*** (compare with Figure 3).

Other RT transform operations

Our original RT transform includes options for interpolating the source-receiver offset values during the inverse transform. The options can be used to ***interpolate or decimate*** ensembles, and to create ***diagnostic displays*** as shown below.

In particular, the ***X^2/T^2 ensemble*** display is useful for analyzing hyperbolic moveout, since all events which are ***hyperbolic in the X/T*** are ***linear*** in the ***X^2/T^2*** domain.

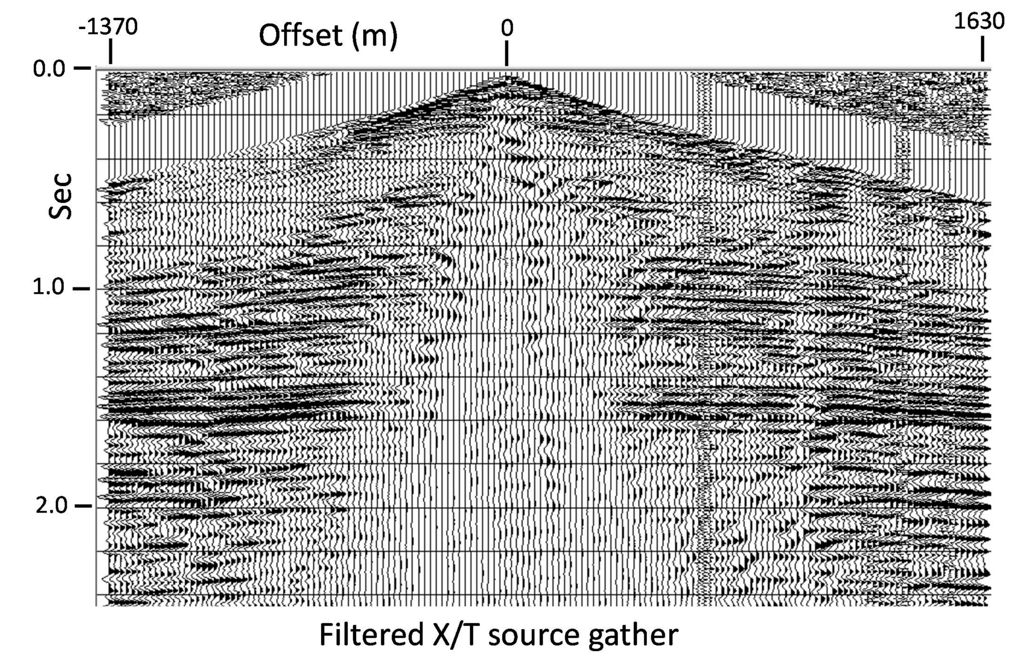

FIG. 6. 2D source ensemble, RT filtered to help prevent aliasing

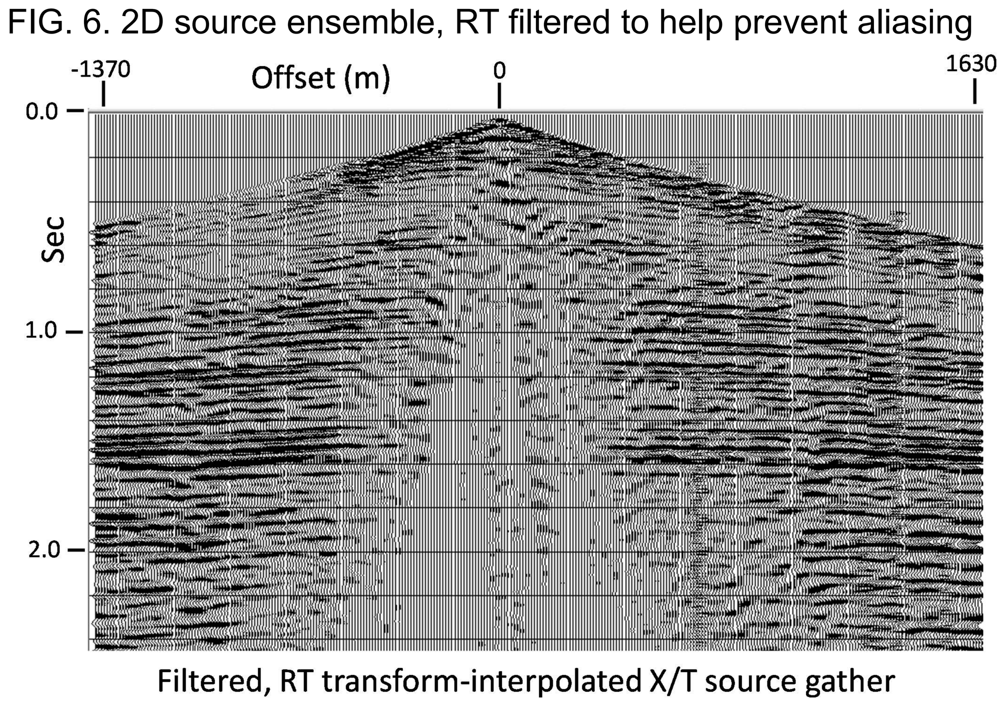

FIG. 7. 2D source ensemble in Figure 6 ***interpolated*** by factor of 2, using forward/inverse RT transform.

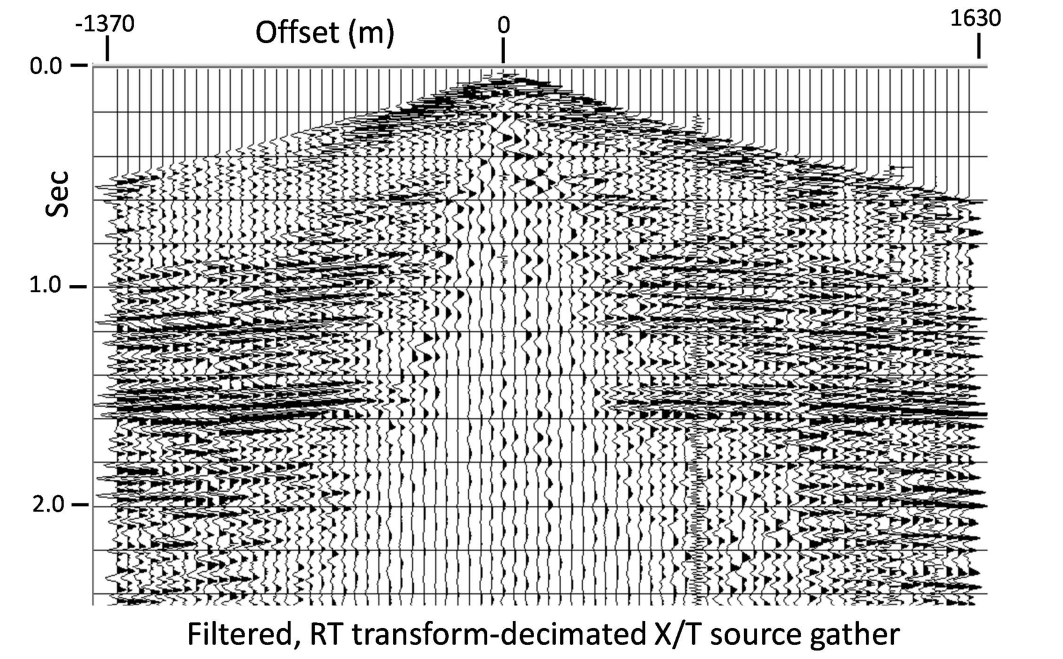

FIG. 8. 2D source ensemble in Figure 6 ***decimated*** by factor of 2, using forward/inverse RT transform.

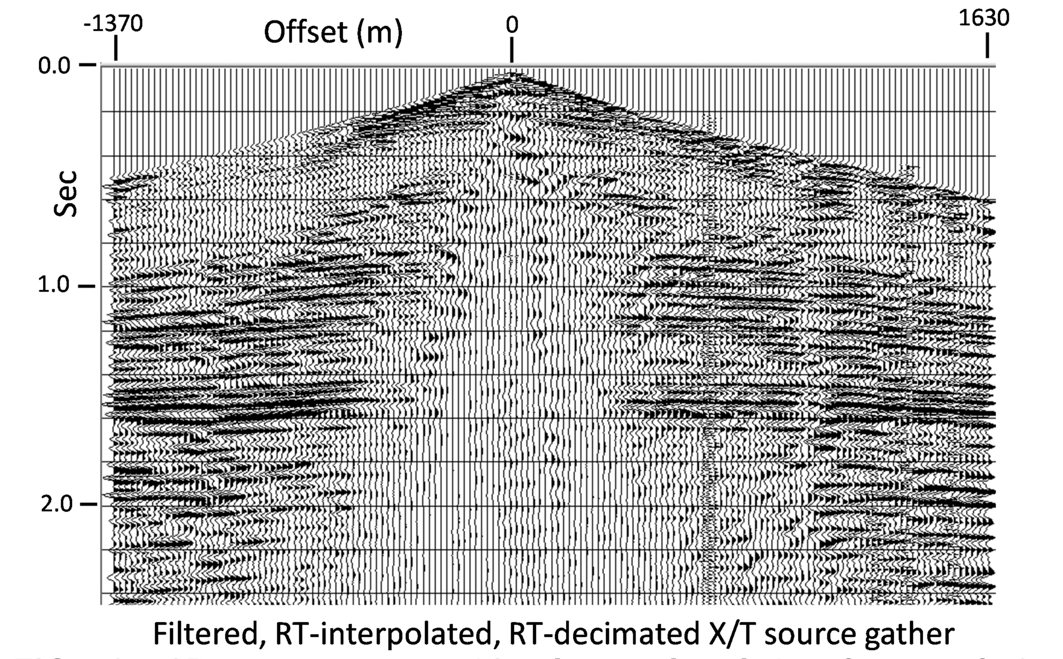

FIG. 9. 2D source ensemble ***interpolated*** by factor of 2 in forward/inverse RT transform, then ***decimated*** by factor of 2 in forward/inverse RT transform. Compare with Figure 6.

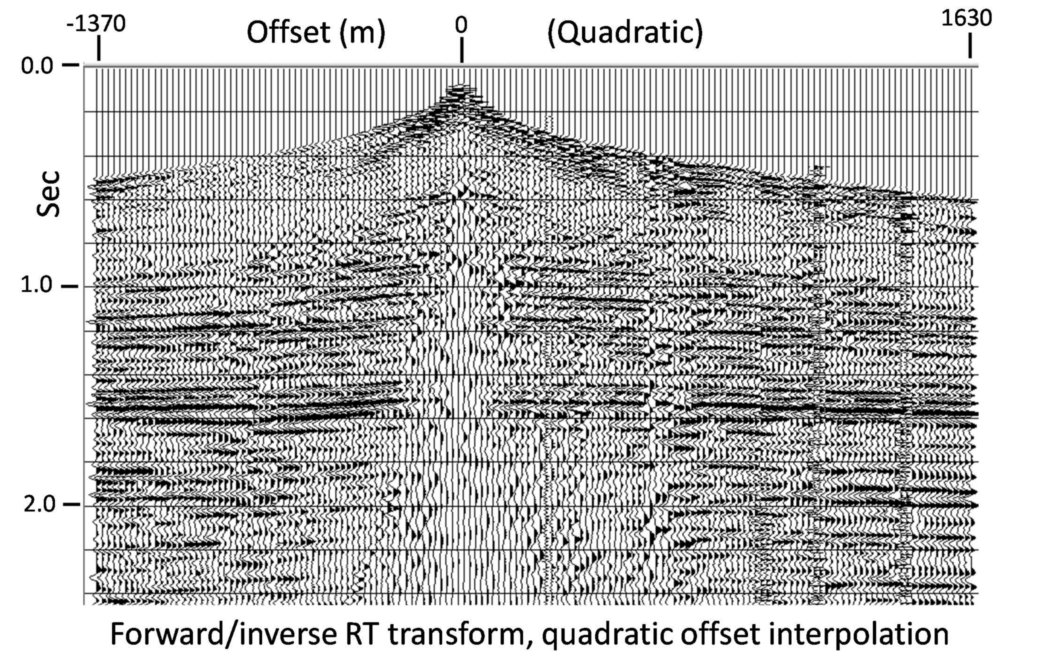

FIG. 10. source ensemble in Figure 6 interpolated to quadratic offsets.

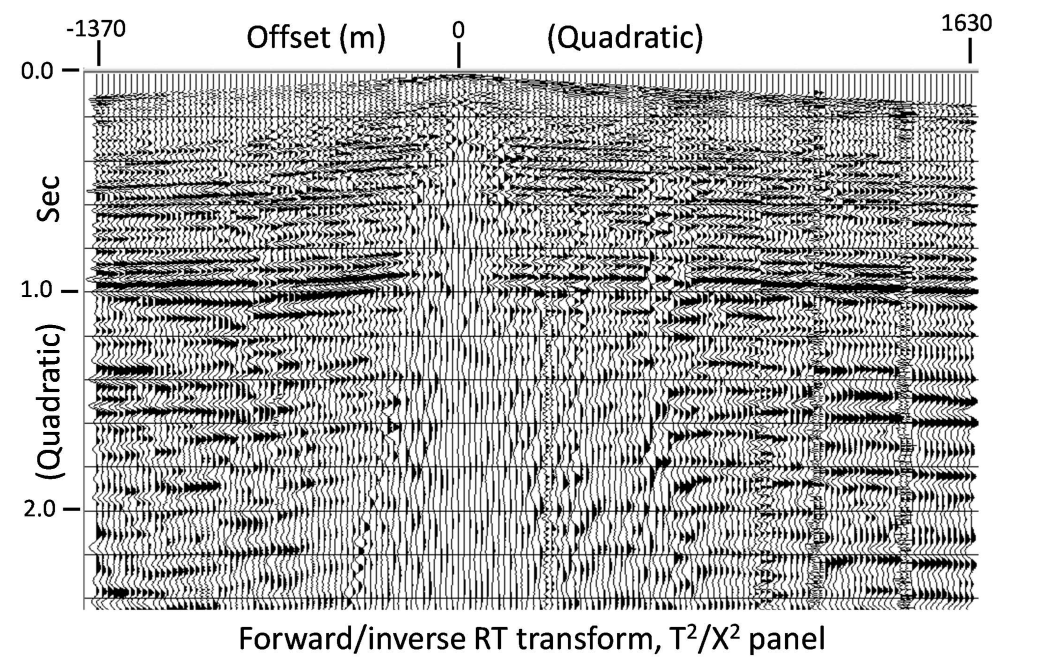

FIG. 11. source ensemble in Figure 6 interpolated to quadratic offsets and quadratic time. ***Hyperbolic events are all linear***, regardless of moveout velocity in this domain.