A ProMAX implementation of nonstationary deconvolution

David C. Henley and Gary F. Margrave

ABSTRACT

Deconvolution is an important seismic processing tool for increasing the resolution with which the seismic method can image the earth. There are many techniques for accomplishing the task, each of them using as much information about the data as is known and making certain assumptions about the characteristics of the seismic records or the underlying geology. One of the key assumptions made in most methods is that the data are statistically stationary, that is, the data characteristics are unchanging with time. Since this is often not even approximately true, there is strong motivation to construct a deconvolution method that does not rely on stationarity for its effectiveness. Presented here is a practical implementation of such a method, based on the Gabor transform, in the ProMAX processing package. This chapter describes the step-by-step numerical procedure, provides an explanation of all parameters, and illustrates the use of the algorithm on several examples of real data from field surveys.

INTRODUCTION

The theory of nonstationary deconvolution has been introduced previously (Schoepp and Margrave, 1997), and the theory behind the use of the Gabor Transform is detailed elsewhere in this report (Margrave and Lamoureux, 2001). In that work persuasive arguments are presented for the development of Gabor transform-based deconvolution as an important new tool in the seismic processing toolbox. The argument might be made that only certain geological settings require this approach due to their inherent nonstationarity. Even if the underlying reflectivity sequence is approximately stationary, however, a typical seismic trace ensemble usually is not, due to a variety of effects ranging from significant inelastic absorption to time and offset dependent data contamination by various source-generated noise trains.

Most conventional statistical deconvolution methods, like Wiener or predictive deconvolution, assume time-series data statistics that do not change with time, and such methods are used to derive and apply a single operator to an entire data trace. Nonstationarity implies data characteristics that change with time, however, so a deconvolution technique must adapt with time to most effectively match the time-varying data statistics. While various schemes for deriving and applying stationary deconvolution operators over short overlapping windows of a data trace have been developed, most can accommodate traces on which the statistics are only relatively slowly changing with time, and these methods also usually involve cumbersome splicing of the windowed results.

The Gabor transform, however, is well-suited to analysing and processing time-varying data like seismic traces. The mathematical theory of the Gabor transform can be found in Margrave and Lamoureux (2001). As implemented numerically in

ProMAX, the transform can be described as a 2-D complex function of both frequency and time, which is computed by windowing an input trace with carefully scaled, overlapping Gaussian functions, and taking the Fourier transform of each window. The Fourier components for each window form a row of the Gabor transform matrix, each row being associated with the centre time of its corresponding Gaussian window. Hence any given element, $G(f,t)$, of the Gabor transform is the complex Fourier component at a specific frequency, *f*, of the product of the input trace and the Gaussian window function centred at time, *t*.

With an input trace decomposed as described above, a standard approach for finding a deconvolution operator can be utilised. This approach assumes that the overall shape of the amplitude spectrum of a trace is attributable to the propagating wavelet, while the fine detail is due to the underlying reflectivity series with which the wavelet is convolved. To deconvolve the data, or remove the wavelet, it is thus necessary to estimate the wavelet spectrum from the spectrum of the raw trace. This is usually done by some form of frequency-smoothing of the raw amplitude spectrum. In the case of the Gabor transform, smoothing in both the frequency and time dimensions can be used to estimate the Gabor amplitude spectrum of the time-varying wavelet needed to deconvolve the Gabor transform of the raw data. The smoothed amplitudes of the Gabor transform become the estimated amplitudes of the time-varying wavelet spectrum, but the phase of this wavelet must either be explicitly zeroed, or, for minimum-phase, computed via the Hilbert transform. The most straightforward way to apply the deconvolution is then just to divide the complex Gabor transform of the input trace by the complex spectrum of the estimated propagating wavelet. The seismic trace can then be reconstructed from the deconvolved Gabor transform by summing the rows of the transform (summing the Gaussian windows) into a single complex vector and inverse Fourier transforming the vector.

The above description is a simple sketch of numerical Gabor deconvolution, but there are a number of alternative computational choices at each stage of the deconvolution process, and the ProMAX module described in this chapter was constructed to allow choice of reasonable alternatives wherever feasible, in order to facilitate study of the method. The next section begins with a functional description of the ProMAX module, following the data step-by-step through the algorithm. Following this is a detailed description of the parameters in the Gabor deconvolution menu, including recommendations for reasonable values and warnings about the use of some of the more experimental parameters.

THE PROMAX MODULE

Gabor deconvolution was first implemented and tested (see Margrave and Lamoureux, 2001) using MATLAB code, with its powerful matrix manipulations and diagnostic capabilities. After the concept was shown to be a viable deconvolution method, it was decided to implement an experimental version of Gabor deconvolution in a more production-oriented software package to simplify testing it on larger sets of seismic data. ProMAX was chosen for the implementation both because of its widespread use in industry and the familiarity of one of the authors (Henley) with creating ProMAX modules. It should be emphasised that the version of Gabor deconvolution described here, and included in the CREWES software release, is an experimental version, and likely still contains bugs or even dubious concepts still to be tested and eliminated. There are many parameters and switches which allow extensive testing of various alternative strategies at several stages of the operation, most of which have not been thoroughly examined. In future releases, it is the intention that testing will have revealed which parameters are important and which strategies work best, so that the module can be simplified and the parameter list in the menu significantly reduced. The default parameters in the current release, however, will provide reasonable results on most data sets.

It is important to note that although the module described here was tailored specifically for ProMAX, the main computational structure is a relatively generic Fortran subroutine. The program which calls this subroutine handles seismic trace reading and writing and manages memory for the various variables and arrays, but otherwise is not involved in the Gabor deconvolution, which happens entirely inside the Gabor subroutine. The code could be ported to a different environment by replacing the calls to ProMAX-specific routines (mostly vector and complex arithmetic routines) by calls to equivalent routines from another library. The source code includes sufficient internal documentation to facilitate this process.

Gabor deconvolution, step-by-step

Input stage

At the beginning of the Gabor deconvolution module, the main program examines input parameters, reads an ensemble of traces (typically, but not necessarily, a shot gather), establishes parameters and memory allocation, and presents the traces and parameters to the Gabor subroutine. The input traces are each checked to ensure that they are non-zero and have not been flagged to be killed. All valid traces are then normalised by their average absolute value, and each trace level is stored for possible use in restoring the trace to its original level prior to output. After normalisation, each valid trace is centred in a gate that is a power of two samples in length. To avoid data wrap-around and edge effects, the length of the pad on each end of the live samples in the gate is checked, and the gate length increased by a factor of two if the pad is too short. Also, the pad values are not zeroed, but are filled with a very low level of random noise (-180 dB with respect to maximum signal level) to avoid any difficulties with zeros in computations. The final step in the data input stage is to compute a set of overlapping Gaussian windows, either symmetric or asymmetric about their centres, normalised so that the overlapped window values sum to unity at any point in time. The padded input trace is multiplied by each window and moved to the real part of the row of the complex Gabor transform array corresponding to the centre time index of the window.

Gabor transform stage

The Gabor transform is completed by taking the 1-D Fourier transform of each row in the Gabor array. To get a Gabor amplitude spectrum, magnitudes are either computed from these complex Fourier spectra, or else the Burg (maximum entropy) algorithm is used to independently compute the magnitudes, at the choice of the user. If Gabor deconvolution is being used in the ensemble derive/apply mode, where the propagating wavelet is estimated from a whole ensemble of traces and applied later to each trace in the ensemble, the Gabor spectral amplitudes for the first trace are stored, the second trace transformed, its magnitudes added to the Gabor amplitude array, and so on until the entire ensemble has been transformed and the spectral amplitudes summed. The amplitude array is then normalised by the number of traces in the ensemble. This stage of the Gabor deconvolution module thus results in a Gabor amplitude spectrum to use in estimating the propagating wavelet. If the module is being used in single trace derive/apply mode (the default), the original Gabor transform, including phase, for the current trace is also present; otherwise it must be recomputed prior to deconvolution.

Deconvolution operator construction stage

The next portion of the Gabor deconvolution subroutine constructs the deconvolution operator from the Gabor amplitude spectrum. The first step in the computation is to apply a time and frequency dependent pre-whitening factor based on a Q estimate for the input data. The intent of this step is to reduce the dynamic range of the elements in the Gabor amplitude spectrum before smoothing, so that the strongest frequency components do not dominate the smoothed array. The smoothing operation offers several choices for smoothing the amplitude array. First, the array may be smoothed in linear or logarithmic amplitude-space. Second, the smoothing may be done with either a running mean (boxcar) smoother, or a running median (better for reducing the influence of very large peaks). Finally, smoothing may be applied more than once, which has the effect of applying weighted smoothing (triangular for two passes, for example). After the Gabor amplitude spectrum has been smoothed, the pre-whitening Q factor is backed out to prepare the spectrum for phase computations.

The smoothed Gabor amplitude spectrum becomes the amplitude spectrum of the estimated wavelet. For zero-phase deconvolution, the wavelet phase values are zeroed. Otherwise, the smoothed amplitudes are used to compute corresponding minimum phase values using either the Kolmogoroff-Hilbert transform algorithm (Claerbout, 1976) or convolution by a quadrature filter, at the discretion of the user. The inclusion of both these options enables testing of one against the other; and one will likely be eliminated in later versions of the Gabor deconvolution. After the wavelet phase has been either zeroed or computed, the complex reciprocal of the spectrum becomes the Gabor deconvolution operator.

Deconvolution application stage

The final section of the Gabor deconvolution module applies the operator constructed in the second section. There are three modes for applying the operator: using ‘analysis’ windows, ‘synthesis’ windows, or using the square root of the operator in analysis windows followed by application of the same square-root operator in synthesis windows.

1. In the first case, the derived Gabor deconvolution operator array is multiplied by the complex Gabor transform array of the input trace (the same operator is used for all the traces in an ensemble, if ensemble derive/apply is chosen). The deconvolved Gabor transform array is collapsed by summing the rows (Gaussian windows); and the inverse Fourier transform is applied to extract the deconvolved seismic trace.

2. In the second case, the windows (rows) of the Gabor transform are summed first, then the resulting complex vector is multiplied in succession by each row of the Gabor deconvolution operator to give a deconvolved Gabor array. Each row of this array is then inverse Fourier transformed and the real part multiplied by the Gaussian window corresponding to the row. Summation over the windows collapses the result to a single output trace.

3. The combination analysis/synthesis application method simply uses the square root of the Gabor deconvolution operator and applies it first in analysis window mode, then in synthesis window mode.

Once again, the inclusion of these alternate options for deconvolution application is for test purposes, and later versions of the code will likely allow only one mode, thus simplifying the module.

Trace output stage

At the end of the deconvolution, the padding is removed from the trace, the normalisation factor is backed out if desired, and the trace is inserted back into the original ensemble, ready to be output from the Gabor subroutine and passed back to the main program.

Gabor deconvolution parameters

Following is a guide to the parameters offered by the menu of the Gabor deconvolution module.

- *Choose single trace or ensemble mode* – This parameter directs the algorithm to either derive and apply Gabor deconvolution one trace at a time, or to derive a deconvolution operator for an entire ensemble of traces, then to apply it to each of the individual traces in the ensemble. The default is single trace, which is more consistent with the nonstationary deconvolution concept.

- *Choose filter application point* – This parameter is used to select application of the deconvolution operator to the analysis windows (the original Gabor transform), or to synthesis windows, or both. The default is analysis windows, and this option seems to give more consistent results than the other two at this point.

- *Restore trace scaling on output* – This parameter allows the output traces of the deconvolution to be restored to their input amplitude levels, if desired. The default is for them to not be restored.

- *Protect against data wrap-around* – This parameter, if used, increases the size of the power-of-two arrays by one more power of two, providing more pad values for input data and hence protecting against edge effects and data wrap-around. This is an issue mainly because the usual input traces are raw traces to which no gain function has been applied, allowing large-amplitude initial values of a trace to overprint low-amplitude final values in the forward and inverse Fourier transforms. The default of no protection is strongly recommended because of the greatly increased execution time involved in doubling all array sizes.

- *Half-width of the Gaussian analysis windows in seconds* – The half-width of a Gaussian window is the time interval from the centre point to the 1/e amplitude point on either limb, and this parameter influences the sensitivity of the Gabor transform to the time variation of the input data—the smaller the parameter, the more sensitive the transform. The default is a reasonable value for most data.

- *Ratio of trailing to leading window width* – This parameter allows the Gaussian windows to be asymmetric, with the leading half-width differing from the trailing half-width. This may have some merit for minimum-phase data, but too large asymmetry leads to artefacts in the output. This is a test parameter, and its use is not encouraged. The default value gives symmetric windows.

- *Increment for analysis window application in seconds* – The overlap of the Gaussian windows is set by this parameter. A value no larger than the half-width above is recommended, and values of about one-fourth of the half-width seem to be best. The execution time of the program is inversely proportional to the size of this parameter. The default is a reasonable value for most data.

- *Half-width of the Gaussian synthesis windows in seconds* – Same as above, except for synthesis windows. This parameter appears only if synthesis windows are chosen above

- *Ratio of trailing to leading window width* – Same as above, except for synthesis windows. This parameter appears only if synthesis windows are chosen.

- *Increment for synthesis window application in seconds* – Same as above, except for synthesis windows. This parameter appears only if synthesis windows are chosen.

- *Choose type of frequency transform for operator* – The amplitude spectrum used in the deconvolution operator may be computed by the Fourier transform or by the Burg algorithm, as selected by this parameter. The default choice is the Fourier transform because of its speed relative to the Burg algorithm

- *Number of coefficients for Burg spectrum* – The detail in the Burg spectral estimate is related to the number of coefficients used in the computation. The fewer the coefficients, the smoother the Burg spectrum (requiring less smoothing in the subsequent smoothing section). Choosing too few coefficients leads to poor results, however, so this parameter needs experimentation. The parameter only appears when Burg is chosen above.

- *Choose minimum- or zero-phase deconvolution* – The purpose of this parameter is obvious. The default is minimum-phase.

- *Choose Hilbert transform method* – When minimum phase is chosen above, this parameter allows the selection of either the Claerbout (Kolmogoroff) or quadrature filter methods for computing minimum-phase from the amplitude spectrum. The default Claerbout method seems more stable than quadrature, but more testing is required.

- *Enter estimated Q for pre-conditioning* – The default for this parameter is such that it essentially disables the pre-whitening factor. Values in the range of 10-200, however, will significantly pre-whiten the spectrum prior to smoothing and should be considered if the input data show any sign of decreasing bandwidth with time.

- *Choose application point for smoothing* –Smoothing may be applied to either the linear amplitudes of the spectrum or to the logarithms of the amplitudes. The default is linear amplitudes.

- *Choose number of passes of smoothing* – A single pass of boxcar smoothing is the default for this parameter; choice of more passes has the effect of applying weighted smoothing, as the boxcar is effectively convolved with itself for each additional pass. The only value of this parameter that makes sense for median smoothing is unity, since subsequent passes of median smoothing have no effect unless the smoothing length changes.

- *Choose the type of smoothing to use* – Either the boxcar running mean or the boxcar running median can be chosen here. The default is the running mean.

- *Frequency dimension for smoothing window in Hz* – This is the length of the boxcar filter used in the frequency dimension for smoothing the Gabor amplitude spectrum.

- *Time dimension for smoothing window in seconds* – This is the length of the boxcar filter used in the time dimension for smoothing the Gabor amplitude spectrum.

- *Stability factor for deconvolution* – This parameter is actually applied in two different places in the algorithm. It is added to the wavelet spectrum to eliminate any zeros prior to division; but the previous smoothing of the spectrum virtually guarantees no zeros anyway. It is also added to the Q factor pre-whitening to prevent division by very small numbers in that computation. The parameter has very little influence on the results and may be defaulted.

EXAMPLES

The purpose of this section is to briefly compare Gabor deconvolution to predictive deconvolution, and to show the effects of a few of the parameters. All of the examples shown are field data, and ensembles or groups of traces are shown rather than individual traces.

Blackfoot example

A shot gather from one of the Blackfoot surveys was used extensively for developing and testing the Gabor deconvolution module because of its good reflections and the presence of several noise trains that interfere with the data in a nonstationary way. The gather shown in Figure 1 is a typical one from near the centre of the 2-D line. It has been subjected to bandpass and AGC for display purposes, as have all the deconvolved results. It should be noted, however, that because of its time-varying nature, the Gabor deconvolution itself applies the equivalent of a gain equalisation to each trace.

To provide a standard for comparison, Figure 2 displays the shot gather of Figure 1 after the application of Wiener (predictive) deconvolution, a standard stationary deconvolution technique. In this figure, the shot gather has been divided into ten-trace ensembles, but the deconvolution is derived and applied on a single trace basis. The general whitening due to the Wiener algorithm is obvious on this result. Note as well, however, that there is a considerable shadow zone following the direct arrival. Also, there are visible remnants of both high- and low-frequency noise.

The Gabor deconvolution shown in Figure 3 contrasts with the Wiener deconvolution of Figure 2 in several ways. First, the results appear to be generally higher in frequency over most of the record. Next, there appears to be less residual low-frequency noise, but possibly more random high-frequency noise. Significantly, because of the nonstationarity of the algorithm, deconvolved events can clearly be seen in the shadow of the direct arrival, and more coherent reflection energy is also visible late in the records. The AGC could have been omitted on this gather after the Gabor deconvolution, and the appearance of the display would be affected very little.

To get a feel for the deconvolution effects in the frequency domain, the amplitude spectrum of a single near-offset trace from the Blackfoot raw shot gather is shown in Figure 4. The spectra for the same trace after Wiener and Gabor deconvolution operations are shown in Figures 5 and 6, respectively. These are whole-trace spectra; and the stationarity of the Wiener process is confirmed by the fact that its spectrum greatly resembles that of the raw trace, except for the overall spectral slope, which is clearly whiter (more level) after the deconvolution. The spectral details have changed little from those of the raw trace. In contrast, the Gabor spectrum is overall even whiter than the Wiener spectrum; and its details seem only superficially similar to those of the raw trace spectrum. This is what might be expected for a nonstationary process, however, since the deconvolution operator changes significantly with time. The spectrum in Figure 6, then, shows the *average* spectral effect of the nonstationary process on the whole trace.

Figure 7 shows a Gabor deconvolution of the Blackfoot shot gather using the Burg spectral estimation option with 5 coefficients. In comparison with the Fourier-based results in Figure 3, the Burg results appear to be somewhat higher in frequency and somewhat cleaner deep in the record. Because the Burg estimate is a smooth spectrum, particularly for small numbers of coefficients, less smoothing is needed in the deconvolution operator derivation. The Gabor deconvolution shown in Figure 8 also uses the Burg algorithm, but with only 3 coefficients. In addition, the width of the analysis windows (and their time increment) was doubled from the previous example. (Both these changes increase the speed of the algorithm significantly). The frequency of the results in Figure 8 is obviously lower that that in Figure 7, and the events behind the direct arrival are not as well-recovered due to the decreased time-variability of this particular realisation of Gabor deconvolution.

The ensemble derive/apply mode of Gabor deconvolution is demonstrated in Figure 9, using the same parameter settings as in Figure 8. This mode of deconvolution seems to work better on ensembles that have a common noise on their individual traces, as in the two centre ensembles in Figure 9. The reflections on these ensembles are more coherent, with less interfering residual ground roll than in Figure 8. On the other hand, individual traces with unique noise problems, like the single trace with visible 60 Hz pickup, are treated more successfully with the single trace derive/apply mode of Gabor deconvolution. In careful comparison of Figures 8 and 9, the ensembles in Figure 9 show slight, but distinct character differences between them, particularly at longer times, whereas those in Figure 8 seem more continuous in data character between ensembles. Figure 10 shows the ensemble derive/apply results for the Fourier spectral algorithm. Although the results are not dramatically different, the Burg results seem slightly better in terms of event coherence and bandwidth.

The synthesis window method of deconvolution application is illustrated by the results in Figure 11, with the Burg 3-coefficient spectral estimation. In comparison with the results in Figure 8, the results are slightly lower in frequency; and the low-frequency noise not as effectively attenuated. On the other hand, some of the shallower reflection events appear to have more coherence.

Figure 12 illustrates the zero-phase option of Gabor deconvolution with the 3-coefficient Burg spectrum. It seems similar in most respects (except phase) to Figure 8.

Southern Ontario

The next example is intended mainly as a comparison between Wiener deconvolution results and those with Gabor deconvolution. The test data are a few traces from a 2-D line in southern Ontario. These traces are from the final brute stack of this line, and the data were not deconvolved prior to stack. The original traces are shown in Figure 13. It can be seen that the traces near the left end of the profile appear to be lower in frequency as well as deficient in reflection energy with respect to those on the right. The Wiener deconvolution shown in Figure 14 has increased the bandwidth of the leftmost traces, but has enhanced unwanted low-frequency noise at the same time. The Gabor deconvolution in Figure 15, on the other hand, not only has somewhat higher frequencies than the Wiener result, but does not enhance the low-frequency noise. Furthermore, the section looks more spectrally balanced throughout, albeit somewhat noisy.

Okotoks field school

The last example was chosen because of the somewhat surprising performance of the Gabor deconvolution on the data. The raw data are exceptionally noisy, containing several kinds of source-generated noise as well as strong 60 Hz pickup on several channels. Two example shot gathers are shown in Figure 16 after bandpass and AGC. The same gathers after Wiener deconvolution are shown in Figure 17. While this deconvolution appears to work reasonably well on the shallow portion of each trace, the deeper portions show mostly high-frequency noise. The Gabor deconvolution in Figure 18 is a much more spectrally balanced result, and fragments of reflections (one indicated by an arrow) can be seen. As well, the shallow events appear to be more completely deconvolved (single spikes) in this figure.

DISCUSSION

The examples shown above will hopefully attract enough interest to stimulate testing of Gabor deconvolution on a much larger scale. On data that appears to have a non-stationary aspect, it seems possible for Gabor deconvolution to outperform predictive deconvolution. There are, however, numerous modes and combinations of parameters to be tested so that unnecessary ones can be eliminated from the algorithm and so that guidance can be offered for parameter selection. There remain other mathematical and computational issues to be explored, as well. For example, the current windows used in the Gabor transform are Gaussian; but there may be other windows even more advantageous.

CONCLUSIONS

A working, testable ProMAX module for doing nonstationary deconvolution using the Gabor transform has been described; and test results shown. Based on the limited testing thus far, Gabor deconvolution seems to provide better results than predictive deconvolution, which is based on a stationary model of seismic data. Further testing and development is definitely warranted, although one of the key issues to be solved in future implementations of Gabor deconvolution will certainly be the speed of execution, which would currently discourage widespread application.

ACKNOWLEDGEMENTS

The authors thank the sponsors of CREWES for support, and various CREWES staff members for helpful discussions.

REFERENCES

Schoepp, A.R. and Margrave, G.F., 1997, Time variant spectral inversion. CREWES Research Report, vol. 9.

Margrave, G.F. and Lamoureux, M.P., 2001, Gabor deconvolution. CREWES Research Report, vol. 13.

Claerbout, J. F., 1976, *Fundamentals of Geophysical Data Processing*. McGraw-Hill, pp 61-62.

APPENDIX

One serious issue for future work is that of the relatively long machine time required for application of Gabor deconvolution, compared to predictive deconvolution, for example. As a guide to anyone running Gabor deconvolution, the following formula shows a very crude estimate of which parameters and input data characteristics affect execution time the most:

$$T \propto \frac{t \times w \times (b+2) \times (p+2) \times (h+2) \times (n+4)}{s \times i} \quad , \tag{1}$$

where $T =$ module execution time; $t =$ trace length in seconds; $w = 1$ for no wrap-around protection, 2 for wrap-around protection; $b =$ number of Burg coefficients (zero for Fourier spectra); $p = 0$ for zero phase, 1 for minimum phase; $h = 0$ for Claerbout Hilbert computation, 1 for quadrature computation; $n =$ the number of spectral smoothing passes; $s =$ the trace sample increment in seconds; and $i =$ the Gaussian window increment in seconds. The formula (1) shows that trace length, array length (wrap-around protection), sample increment, and window increment are the most significant factors.

FIGURES

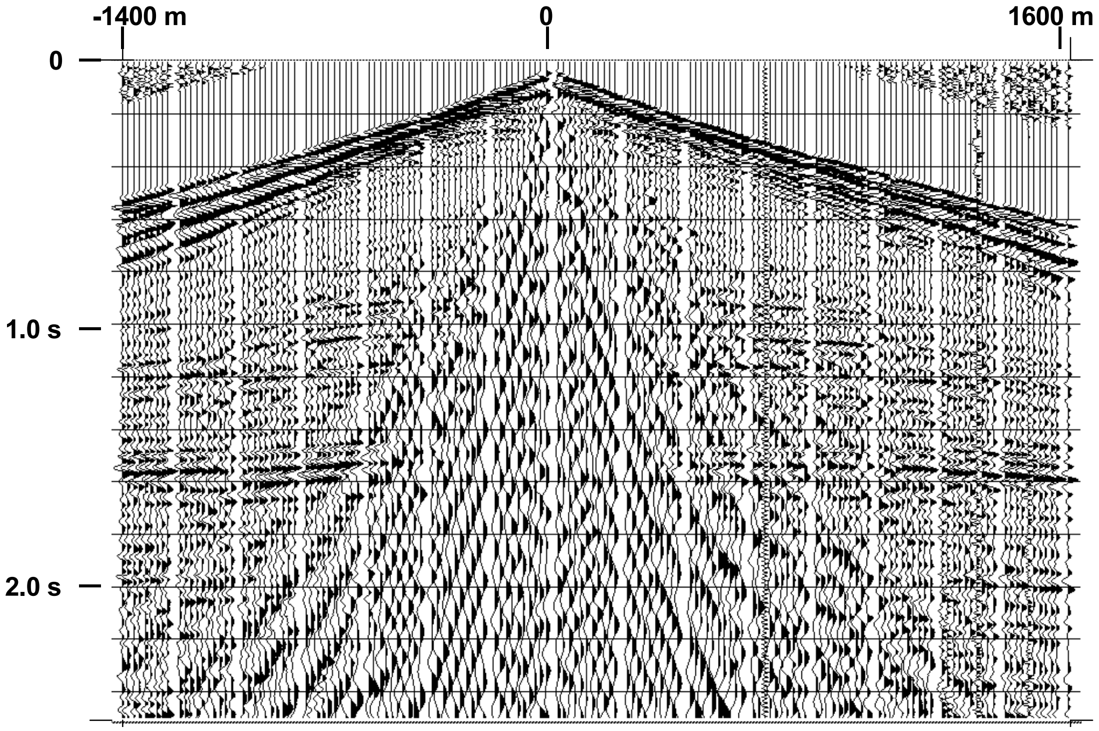

FIG. 1. Raw shot gather from Blackfoot with 8-12-70-90 Hz bandpass and 700 ms AGC applied for display purposes.

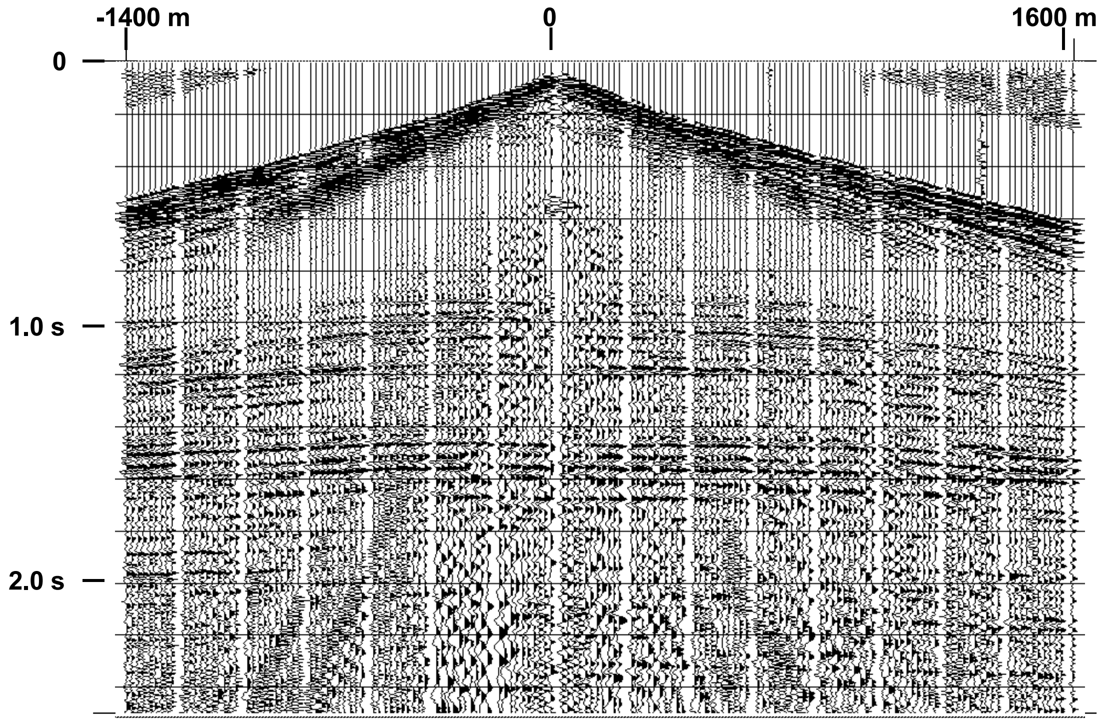

FIG. 2. Blackfoot shot gather with Wiener deconvolution, followed by bandpass and AGC for display.

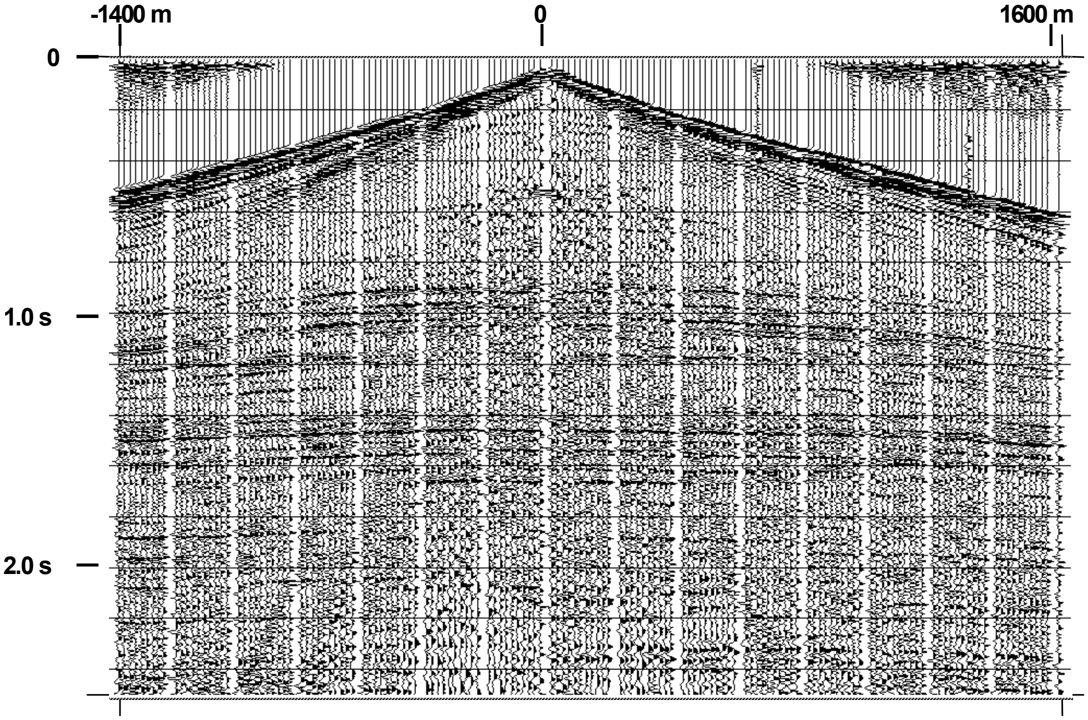

FIG. 3. Blackfoot shot with Gabor deconvolution followed by bandpass and AGC for display.

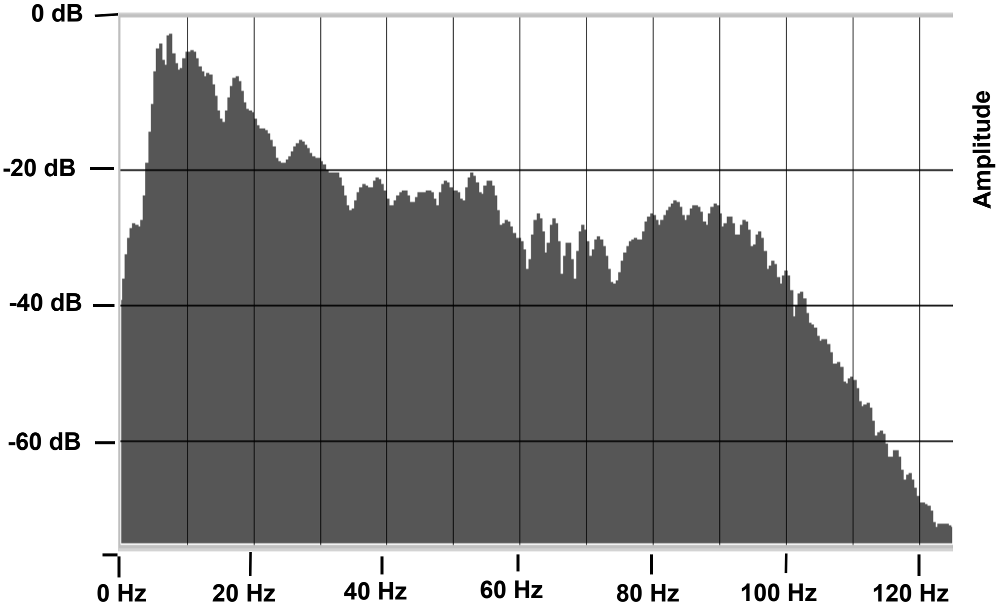

FIG. 4. Amplitude spectrum of near offset raw trace from Blackfoot shot gather

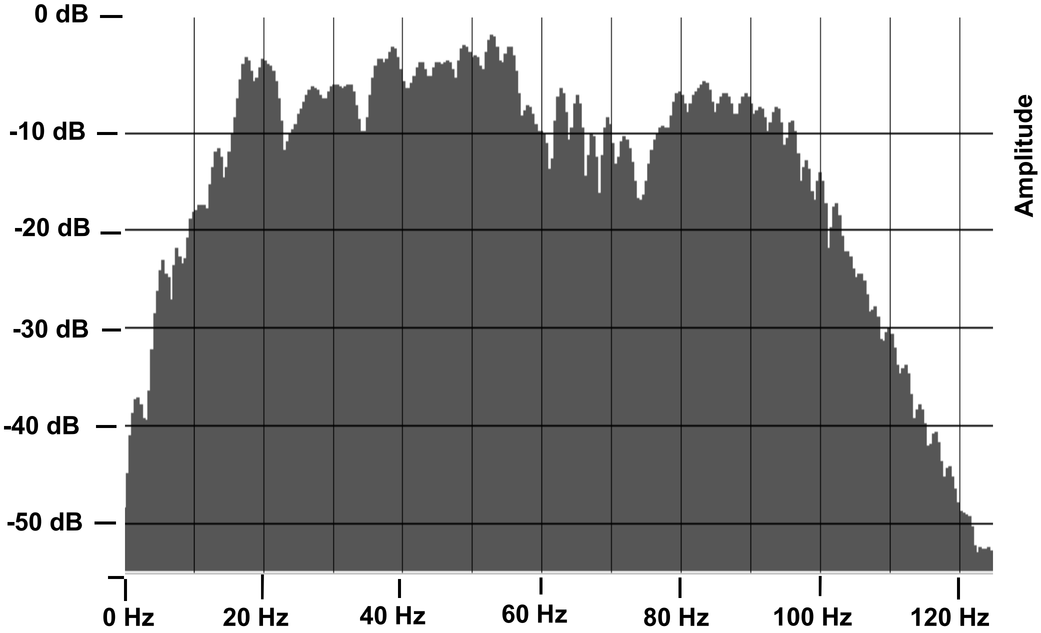

FIG. 5. Amplitude spectrum of near offset Blackfoot trace after Wiener deconvolution.

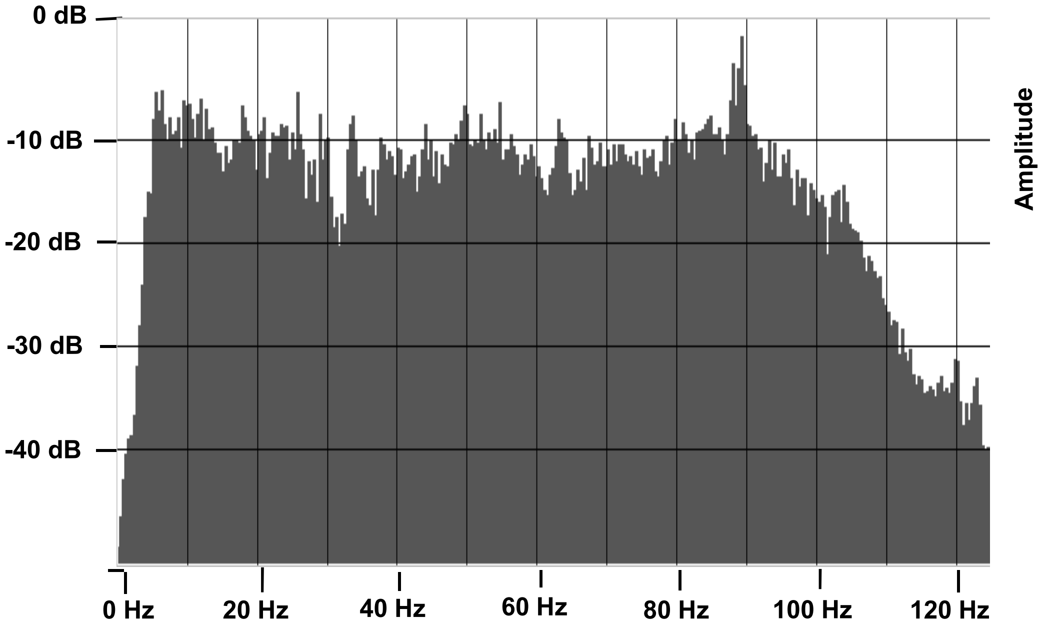

FIG. 6. Amplitude spectrum of near offset trace after Gabor deconvolution.

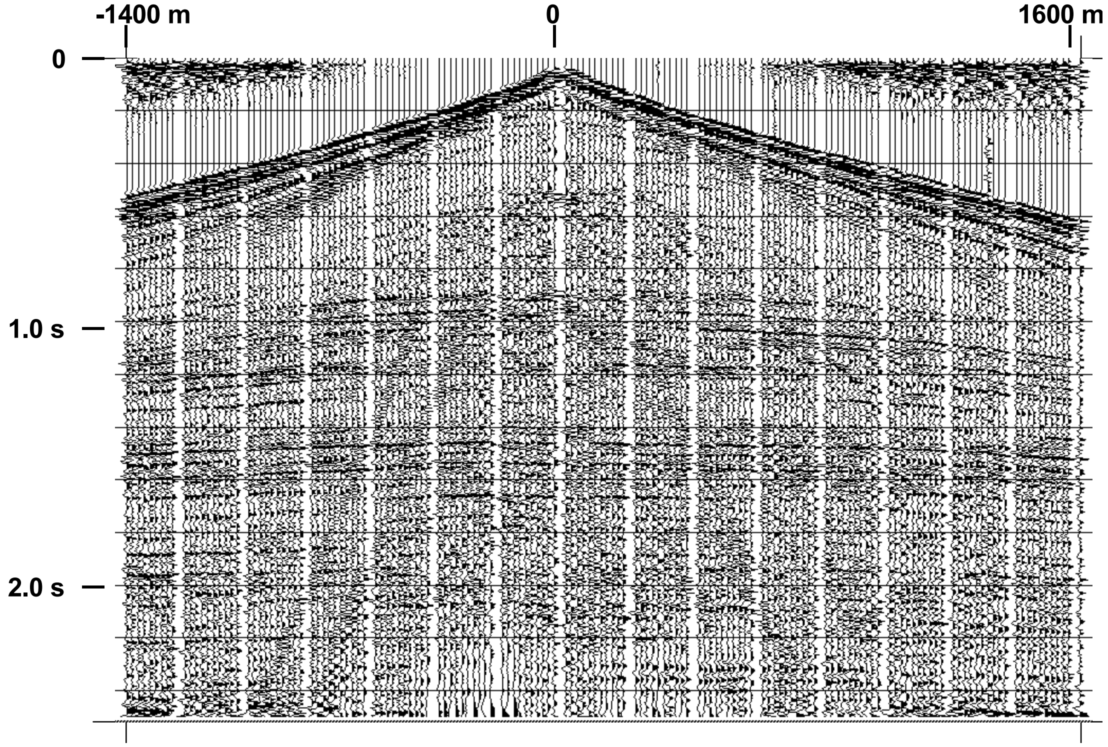

FIG. 7. Gabor deconvolution of Blackfoot shot gather using the Burg algorithm with 5 coefficients for the operator derivation.

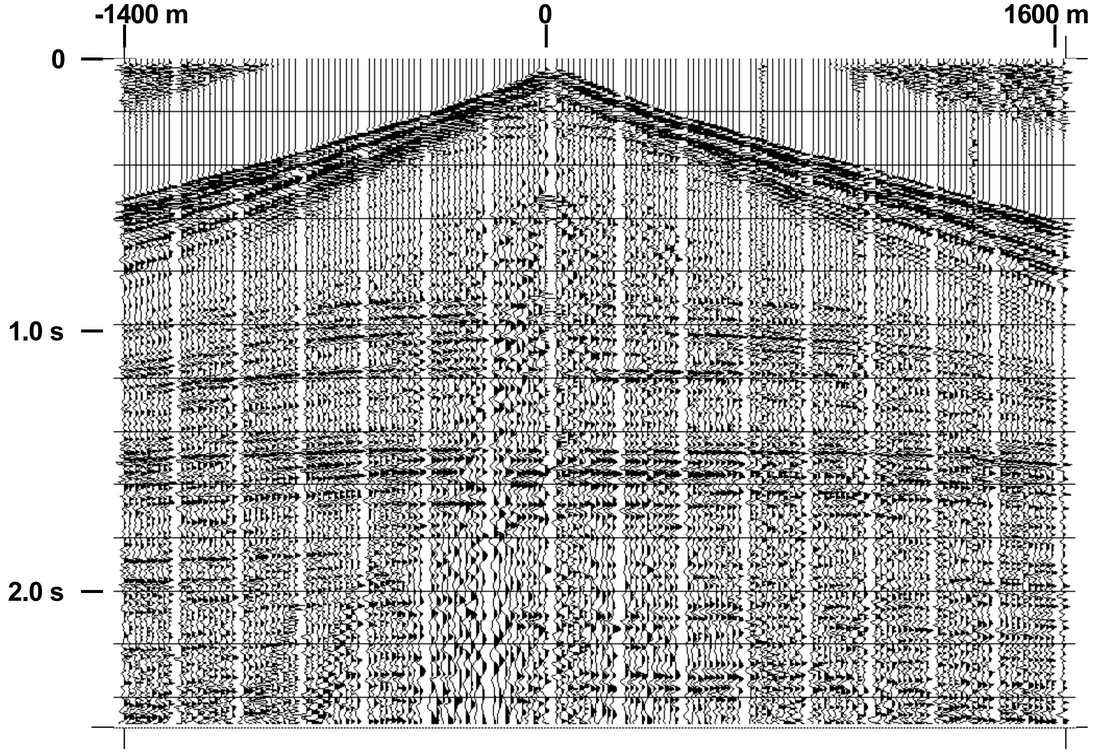

FIG. 8. Gabor deconvolution of Blackfoot shot gather using the Burg algorithm with 3 coefficients for operator derivation, as well as longer analysis windows, less spectral smoothing.

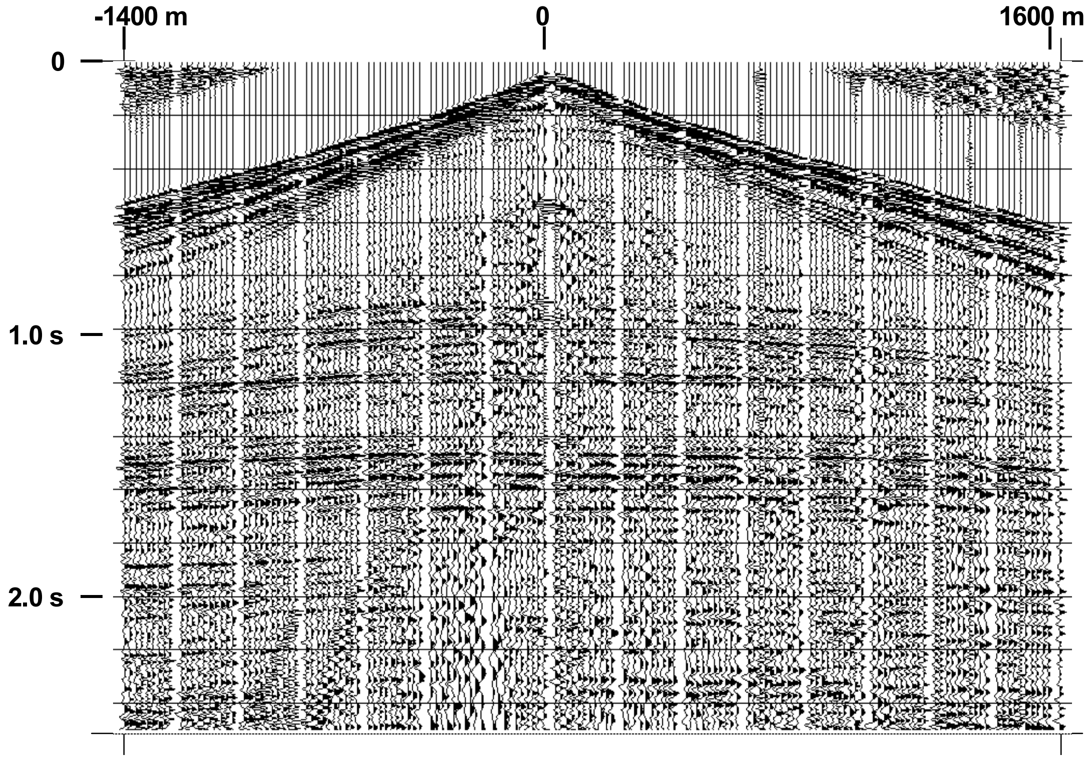

FIG. 9. Gabor deconvolution of Blackfoot shot gather using the Burg spectral algorithm with 3 coefficients, and deriving and applying one operator for every ten-trace ensemble.

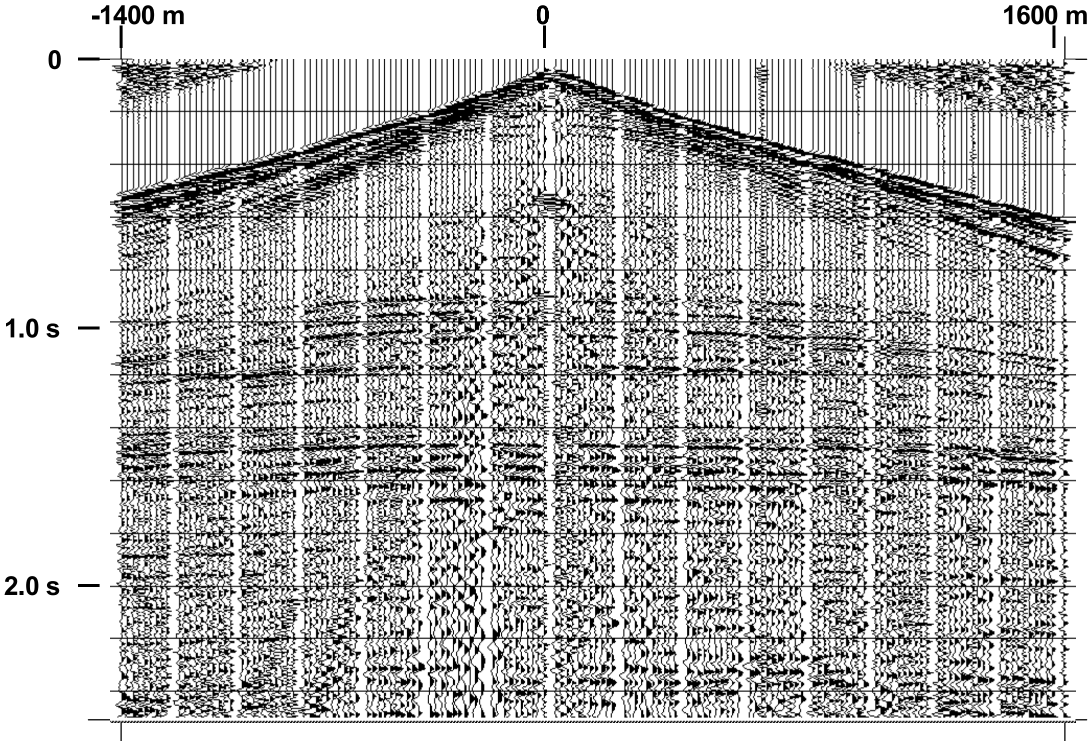

FIG. 10. Gabor deconvolution of Blackfoot shot gather using the Fourier algorithm, increased spectral smoothing, and derivation and application of one operator per ensemble.

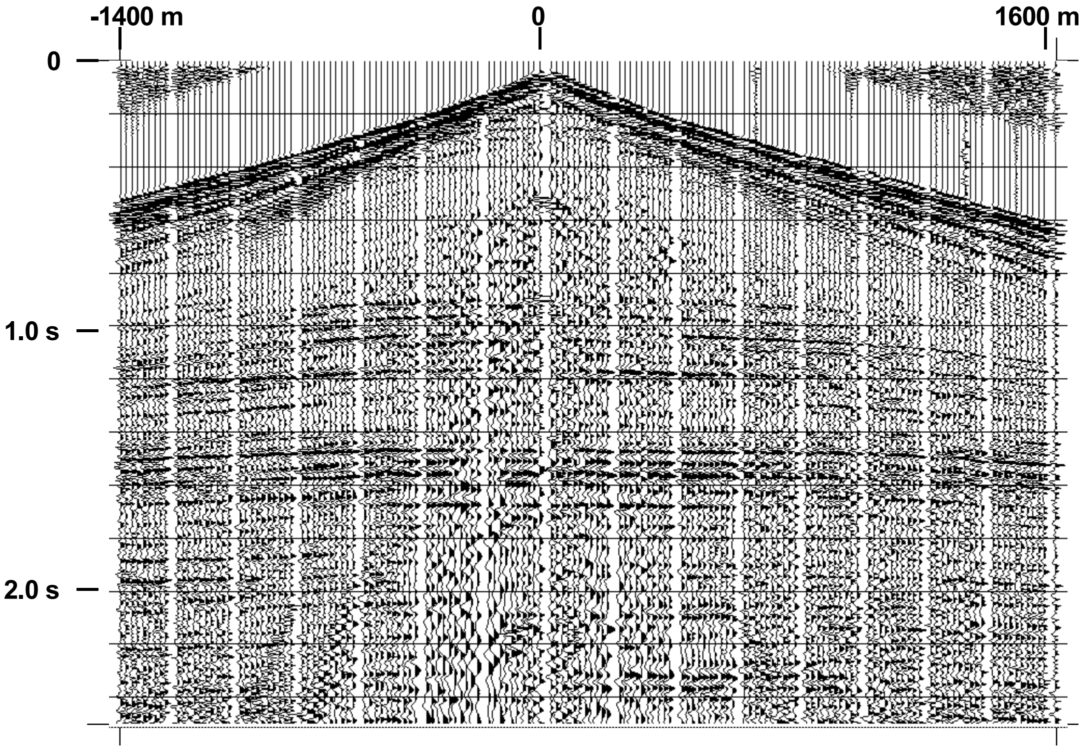

FIG. 11. Gabor deconvolution of Blackfoot shot gather using the Burg 3-coefficient algorithm in the synthesis mode of deconvolution.

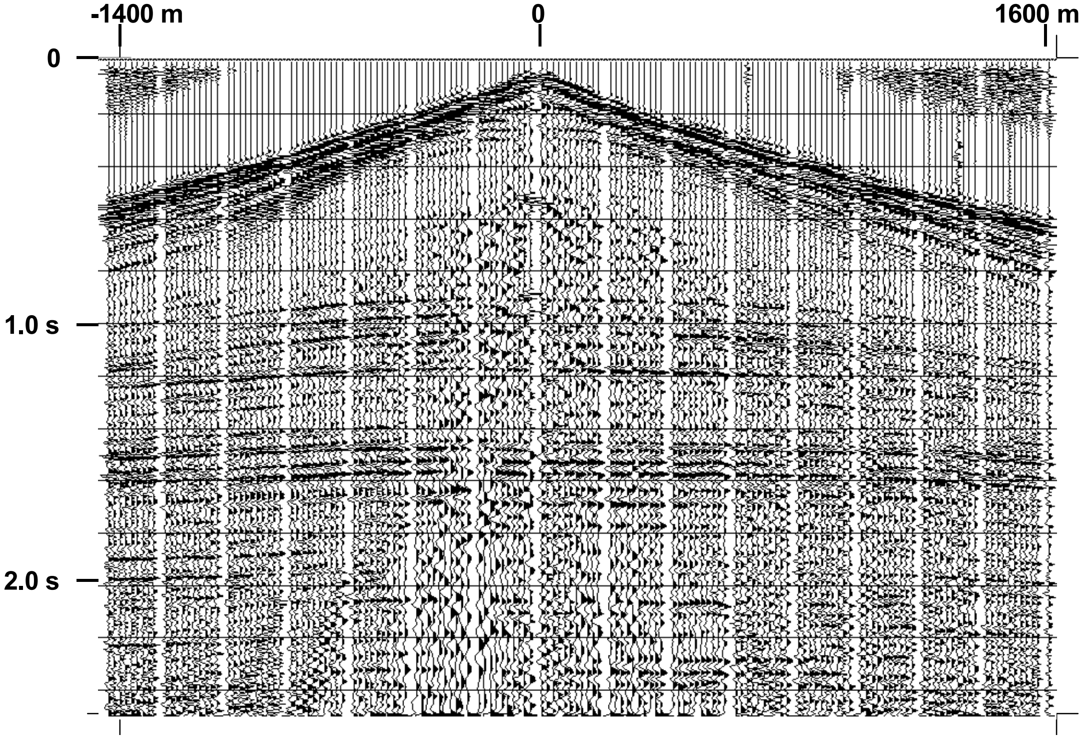

FIG. 12. Gabor deconvolution of Blackfoot shot gather using the Burg 3-coefficient algorithm in the analysis mode of deconvolution, and applying a zero-phase operator.

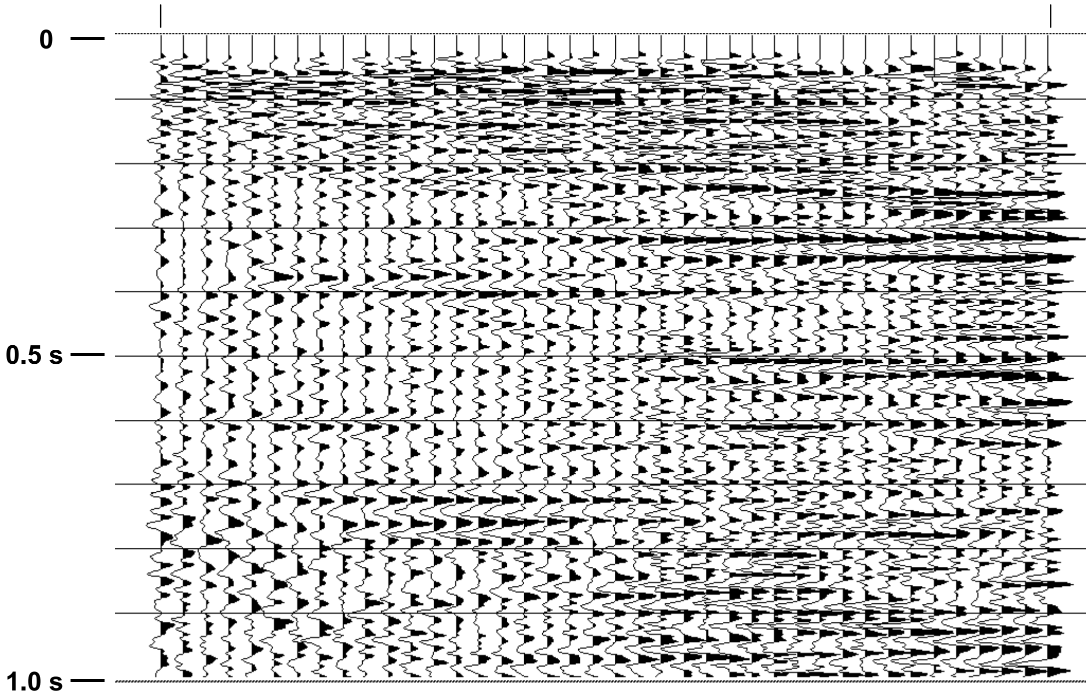

FIG. 13. Portion of a final stack of data from Southern Ontario with no deconvolution, 8-12-90-125 Hz bandpass applied for display.

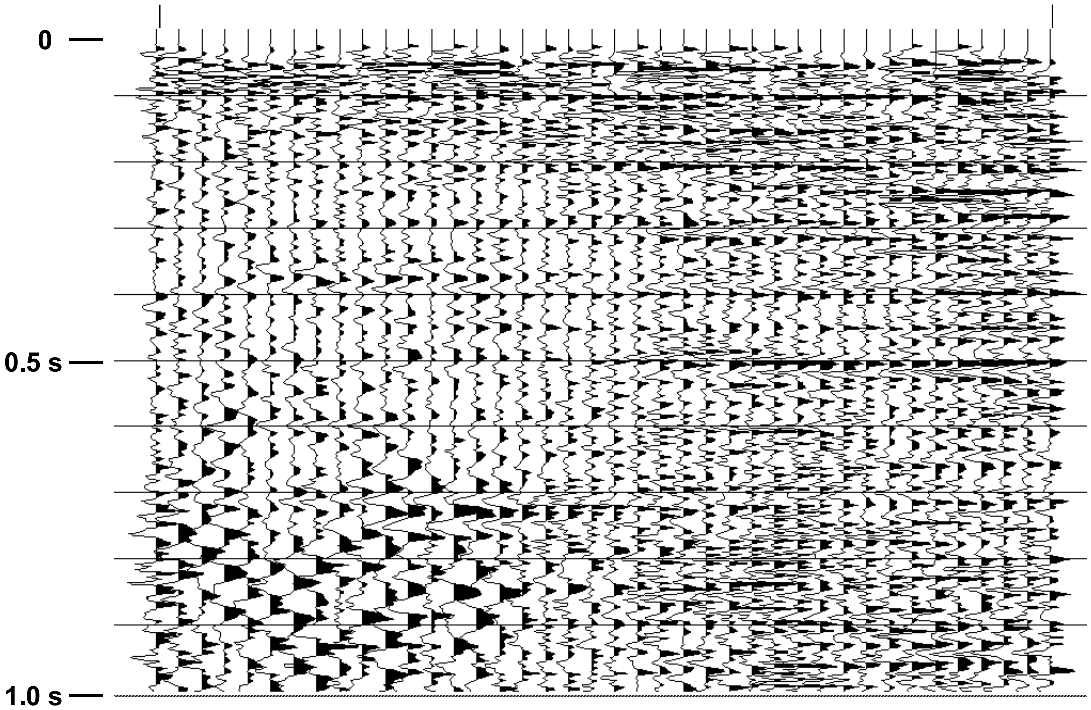

FIG. 14. Southern Ontario stack after Wiener deconvolution, bandpass filter.

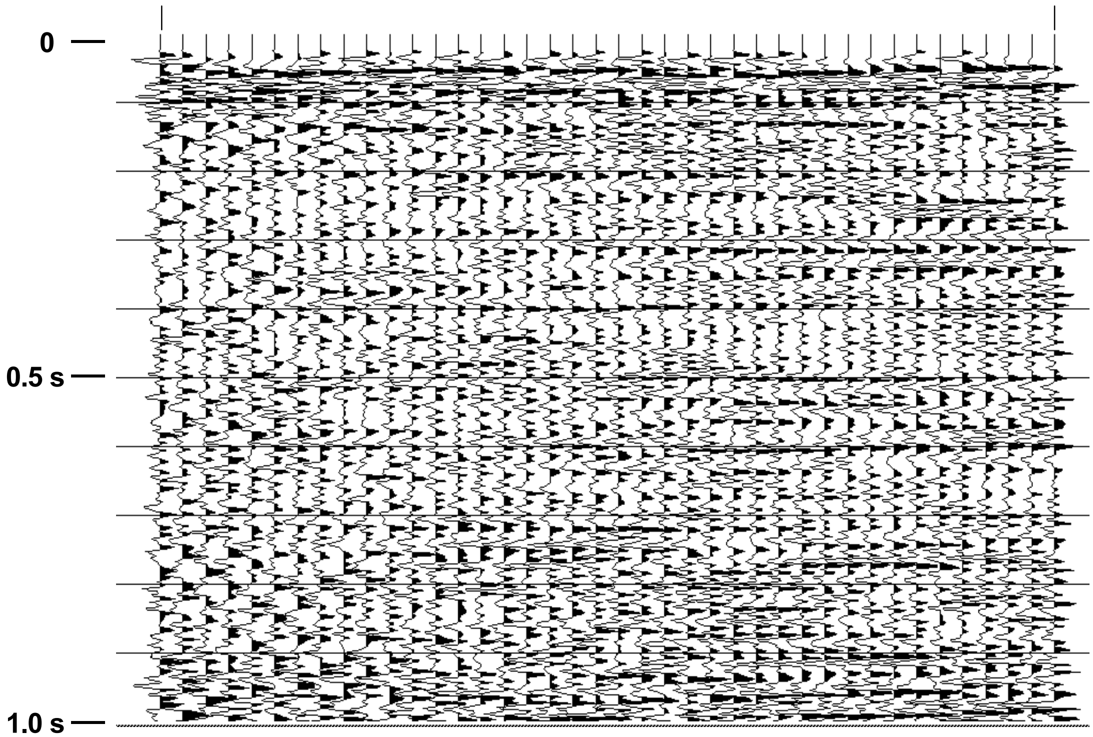

FIG. 15. Southern Ontario section with Gabor deconvolution applied (5-coefficient Burg spectrum), followed by bandpass.

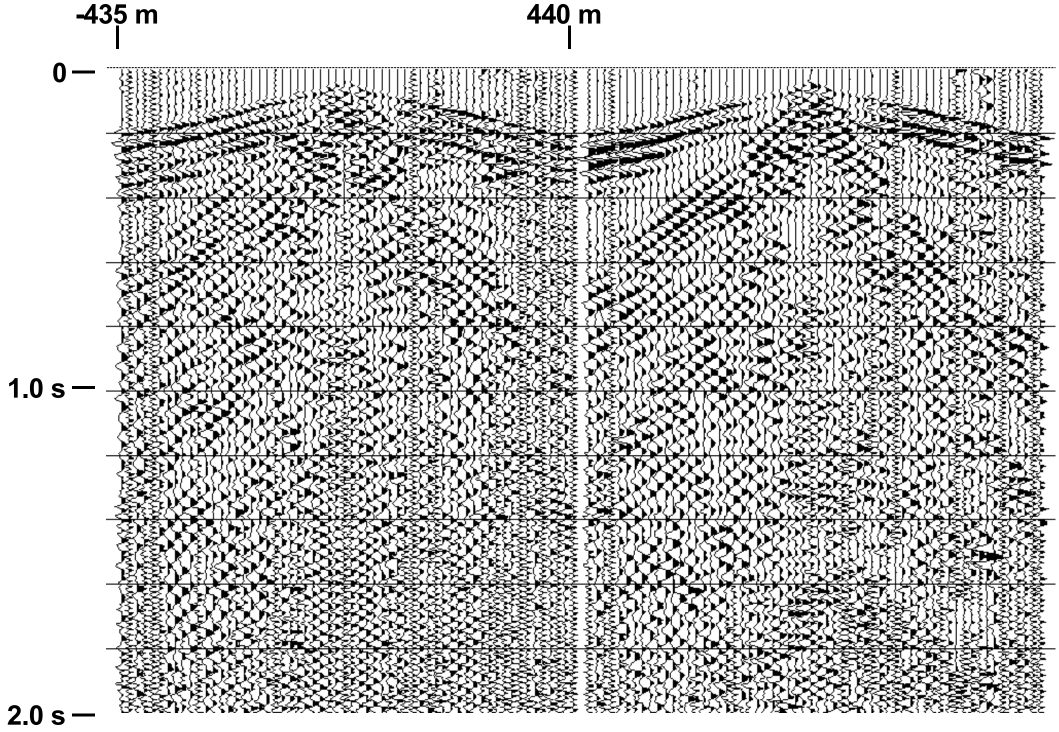

FIG. 16. Two shot gathers from 2000 Okotoks field school line with 12-15-65-80 Hz bandpass and 500 ms AGC applied for display.

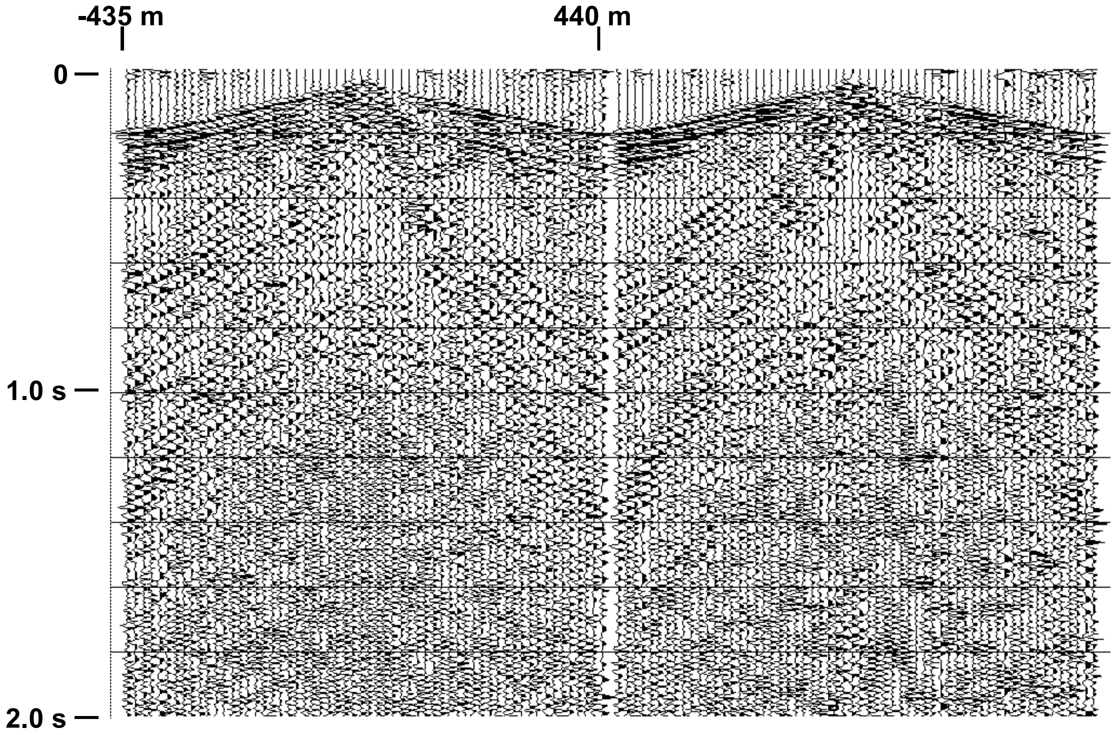

FIG. 17. Okotoks field school records with Wiener deconvolution applied, followed by bandpass and AGC.

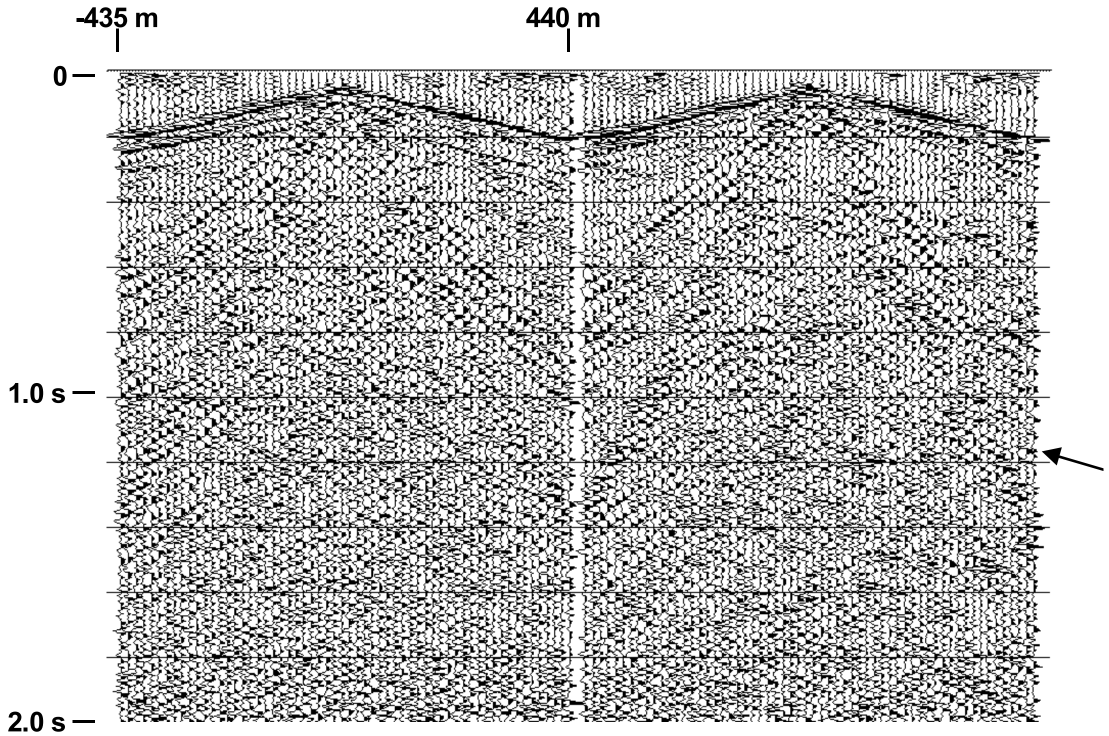

FIG. 18. Okotoks field school records with Gabor deconvolution (5-coefficient Burg) applied, followed by bandpass and AGC. Arrow indicates reflection fragment.