REDUCING THE PRESSURE ON DATA ACQUISITION AND PROCESSING: I - MULTISHOOTING PROCESSING OF SINGLE-SHOT DATA

LUC T. IKELLE

CASP Project, Department of Geology and Geophysics, Texas A&M University, College Station, TX 77843-3115, U.S.A.

(Received July 10, 2008; accepted October 3, 2008)

ABSTRACT

Ikelle, L.T., 2009. Reducing the pressure on data acquisition and processing: I - Multishooting processing of single-shot data. *Journal of Seismic Exploration*, 18: 93-102.

In some E&P organizations, the amount of data required to image the subsurface has now risen to several terabytes per survey, especially when multiple azimuths are considered. For the processing of converted-wave data, this amount of data rises even further, by twofold or threefold, depending on the components of particle velocity under consideration. The processing time also increases severalfold. In other words, petroleum seismologists are close to reaching a brick wall, if we have not already done so, in regard to the amount of data that we can realistically collect and properly process. In my recent book on coding and decoding, I introduced the concept of multishooting to address the problem. The idea is that seismic waves can be generated from several locations simultaneously (or nearly simultaneously, by introducing small time delays between the shooting points) instead of one single-source location at a time, as is currently the case. Significant savings in time and money in acquiring, processing, and even storing seismic data can be achieved by using this concept. However, the implementation of this concept in actual data acquisition and data processing may take some time, as a number of the solutions associated with multishooting acquisition and with the processing of multishot data require significant modifications of our current practices. In this series of papers, we propose some ideas which may be less effective than the multishooting concept, but can readily be implemented without significant new developments in acquisition or processing.

In this first paper, we consider data collected in the standard form (i.e., from one single-source location at a time). We propose to group these data as if they were acquired with the multishooting technique and process them as multishot data. By doing so, we reduce the size of our data and processing time.

KEY WORDS: multishooting acquisition, multishooting processing, multiple attenuation, Kirchhoff scattering series, coding, decoding.

0963-0651/09/$5.00

A BRIEF REVIEW OF THE MULTISHOOTING ACQUISITION

Before we go further in this discussion, let us introduce some terminologies which help us differentiate the multishooting acquisition process from current acquisition practices. Again, we will call the concept of generating waves simultaneously from several locations *simultaneous multishooting*, or simply *multishooting*. The data resulting from multishooting acquisition will be called multishot data, and those resulting from the current acquisition approach, in which waves are generated from one location at a time, will be called single-shot data. So multishot data are the coded data, and the decoding process aims at reconstructing single-shot data. The single-shot data are made of shot gathers, which we will call here single-shot gathers. The multishot data are also made of shot gathers. But this time, a shot gather is a response of multiple single-shot points. We will call the shot gathers in the multishot data multishot gathers. We will often use the notation **1mIs** to indicate that the multishot gathers are mixtures of I single-shot gathers. For example, 1m4s means that each multishot gather is a mixture of four single-shot gathers.

Let us return to the processing of multishot data. Existing data-processing algorithms can be used to process multishot data as long as they do not requires CMP gathers, receiver gathers, or offset gathers as inputs, because such gathers are not readily available from multishot data. Also, the algorithms which contain numerical operators in these domains, even if their input data are in the shot-gather domain, may not be directly applicable to multishot data. It turns out that multiple-attenuation algorithms are most affected by this restriction. We describe one way of overcoming this difficulty for the particular class of demultiple algorithms, which are based on the predict-then-subtract approach. Our solution does not require any new algorithm development other than grouping single-shot data into multishot gathers.

We have here opted to reformulate one of the algorithms of the predict-then-subtract approach, namely the Kirchhoff inverse scattering algorithm developed by Ikelle et al. (2001), for the demultiple of multishot data. The predict-then-subtract approach consists of predicting multiples from the actual data and then subtracting them from the same data. The advantage of this approach is that it does not require any knowledge of the subsurface and that it is valid for multidimensional data. Note that this demultiple can also be derived from the Lippmann-Schwinger equation, which leads to the Born series, or from Huygens principles [see Ikelle and Amundsen (2005) or Weglein and Dragoset (2005) for references and a description of these alternative approaches]. Our formulation here will focus on towed-streamer acquisition geometry but can easily be extended to all marine-acquisition geometries.

A BRIEF REVIEW OF KIRCHHOFF-BASED FREE-SURFACE MULTIPLE ATTENUATION

Let $\phi_0 = \{P_0, V_0\}$ be the two-component vector of towed-streamer data, where P_0 represents the pressure and V_0 represents the vertical component of the particle velocity. The inverse Kirchhoff scattering series for attenuating free-surface reflections from, say, pressure seismic data in the F-X domain can be written as follows:

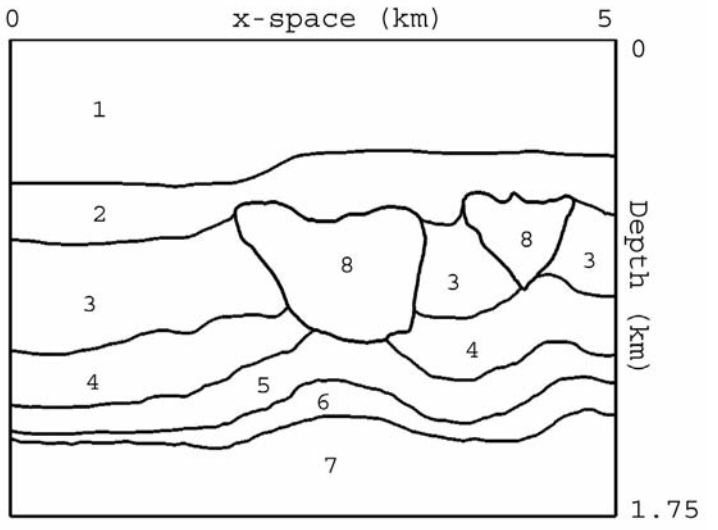
$$P_P(\mathbf{x}_s,\omega,\mathbf{x}_r) = P_0(\mathbf{x}_s,\omega,\mathbf{x}_r) - a(\mathbf{x}_s,\omega)P_1(\mathbf{x}_s,\omega,\mathbf{x}_r) + a^2(\mathbf{x}_s,\omega)P_2(\mathbf{x}_s,\omega,\mathbf{x}_r) - \dots \quad , \tag{1}$$

where P_P represent the data without free-surface-reflection events, $a(\mathbf{x}_s,\omega) = 1/s(\mathbf{x}_s,\omega)$ is the inverse source signature, with $s(\mathbf{x}_s,\omega)$ being the source signature, $\mathbf{x}_s$ the shot point and $\mathbf{x}_r$ the receiver point. Note that the source can vary with the single-shot point. The first term of the scattering series, P_0, is the actual data. The other terms P_1, P_2,etc. are given by

$$P_j(\mathbf{x}_s,\omega,\mathbf{x}_r) = \int_{S_0} dS(\chi)\, P_{j-1}(\mathbf{x}_s,\omega,\chi)V_0^{(nd)}(\chi,\omega,\mathbf{x}_r) \ , \quad j = 1,2,3,\dots \ , \tag{2}$$

where $\chi = (x,y)$, S_0 is the surface in which sources and receivers are located, and $V_0^{(nd)}(\chi,\omega,\mathbf{x}_s)$ is the vertical component of the particle velocity without the direct wave. The second term, P_1, is computed as a multidimensional convolution of the data P_0, sorted in shot gathers, by the vertical component of the particle-velocity data, which we denote V_0, in receiver gathers. The resulting field is in shot gathers and aims at attenuating events which correspond to one bounce at the sea surface; the next term, P_2, which is computed as a multidimensional convolution of P_1 by V_0, aims at attenuating events which correspond to two bounces at the sea surface; and so on. In summary, the terms P_1, P_2, P_3, etc., allow us to predict free-surface multiples, whereas the inverse source $a(\mathbf{x}_s,\omega)$ allows us to properly scale the predicted multiples in such a way that the result of the scattering series in (1) can produce data without free-surface-reflection events. Notice that the application of this series does not require any knowledge of the subsurface.

In most practical applications of the series in (1) today, we are concerned with data in which direct-wave arrivals have been muted. The ghost reflections are treated as part of the source signature. In this case, the series in (1) allows us to remove free-surface multiples and ghosts of free-surface multiples from the data while preserving primaries and ghosts of primaries. Let us illustrate the application of the series in (1). We consider the 2D model in Fig. 1. We have used a finite-difference algorithm to 320 single-shot gathers with 12.5 m spacing



M	1	2	3	4	5	6	7	8
V	1.5	1.9	2.0	2.1	2.65	3.0	2.8	4.4
ρ	1.0	1.5	1.9	2.2	2.3	2.45	2.65	2.15

Fig. 1. The 2D model used to generate the data used in this paper.

between single-shot points. One of these single-shot gathers is shown in Fig. 2a. We use 320 receivers to record data generated by each single-shot point. The receiver spacing is also 12.5 m. Actually, the 320 single-shot points are locations at the same points near the sea surface as the receivers are. We started the demultiple process by computing the terms P_1 and P_2 of the series in (1). Single-shot gathers associated with P_1 and P_2 are shown in Figs. 2b and 2c, respectively. The Matlab code used for computing P_1 and P_2 is given in Table 1. We then obtain the demultiple results in Fig. 2d by using the first three terms of the series in (1).

A REFORMULATION OF THE KIRCHHOFF DEMULTIPLE FOR MULTISHOT DATA

Let us start by describing some additional quantities and variable coordinates that we need for reformulating the Kirchhoff series in (1) for multishot data. We denote the various positions of multishot arrays in seismic surveys by $\mathbf{x}_{sn}$; $\mathbf{x}_{sn}$ can be the centers of multishot arrays, the positions of the first single-shot in multishot arrays, etc. We denote the fields corresponding to

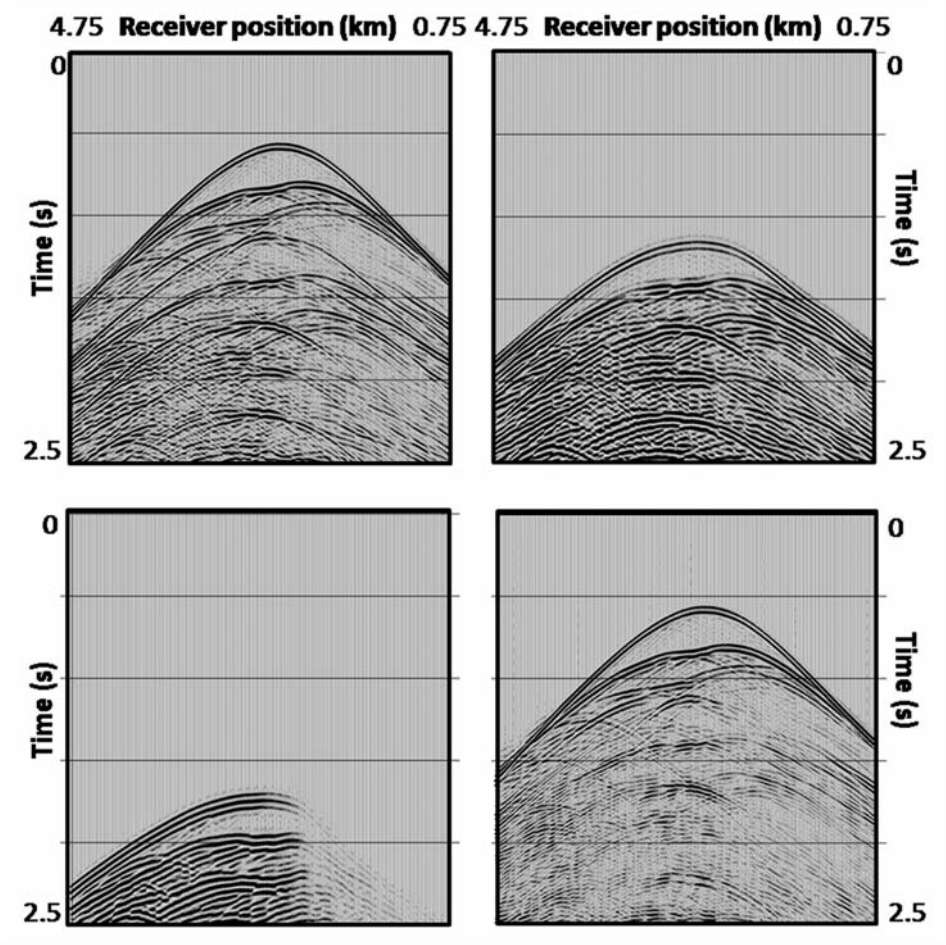


Fig. 2. (a) A single-shot gather of a 2D dataset before the demultiple process. (b) The field of predicted free-surface multiples (P_1); the second term of the Kirchhoff series. (c) The field of predicted free-surface multiples (P_2); the second term of the Kirchhoff series. (d) The single-shot gather in Fig. 1a after the demultiple process. We have used the first three terms of the series in (1).

multishot data with a tilde and those corresponding to single-shot data without the tilde. For example, $\tilde{P}_j(\mathbf{x}_{sn},\omega,\mathbf{x}_r)$ will represent the multishot data, and $P_j(\mathbf{x}_s,\omega,\mathbf{x}_s)$ will continue to represent the single-shot data, as introduced earlier.

We also find it useful to identify each single-shot point of our multishooting survey. So we introduce an additional variable, $\mathbf{x}_{nm}$, where the index n indicates the multishooting array under consideration and the index m indicates the single-shot point of the n-th multishooting array, as illustrated in Fig. 3. If we have a survey of N multishot gathers, and if each multishot array

Table 1. A Matlab code for computing the terms of the Kirchhoff series.

```
function [m1,m2]=predmult(frmax, dt, pres, vz, N)

% frmax: maximal frequency
% dt: time interval
% pres: pressure data
% vz: The vertical component of the particle velocity
% N: time length plus padding
% m1: predicted multiples, starting with first-order multiples
% m2: predicted multiples, starting with second-order multiples

[nt,nr,ns] = size(pres); ifrmax=1+floor(frmax*N*dt);
if(nr~=ns); error('no of receivers= no of shots should'); end;
cpres=fft(pres,N,1); cvz=fft(vz,N,1);

for iom=1:N
   iom
   cp=squeeze(cpres(iom,:,:)); cv=squeeze(cvz(iom,:,:));
   cm1(iom,:,:)=cp*cv;
   cm2(iom,:,:)=cp*cv*cv;
end
m1=ifft(cm1,[],1); m2=ifft(cm2,[],1);
```

has I single-shot points, then m will vary from 1 to I, and n will vary from 1 to N. The number of single-shot points in the entire survey will be N $\times$ I. The position of the multishot arrays can be described by either $\mathbf{x}_{sn}$ or $\mathbf{x}_{n1}$. The variable $\mathbf{x}_r$ will still denote the receiver positions, as in the previous subsection. We assume that receiver-point locations are chosen to coincide (or interpolated to coincide) with shot-point locations through the entire survey. Therefore, we can describe the receiver points by the variable $\mathbf{x}_{mn}$. Using these notations, the term of the Kirchhoff scattering series which allows us to predict free-surface reflections from single-shot seismic data can be rewritten as follows:

$$P_j(\mathbf{x}_s,\omega,\mathbf{x}_r) = \sum_{n=1}^{N}\sum_{m=1}^{I} P_{j-1}(\mathbf{x}_s,\omega,\mathbf{x}_{nm})V_0(\mathbf{x}_{nm},\omega,\mathbf{x}_r) \quad . \tag{3}$$

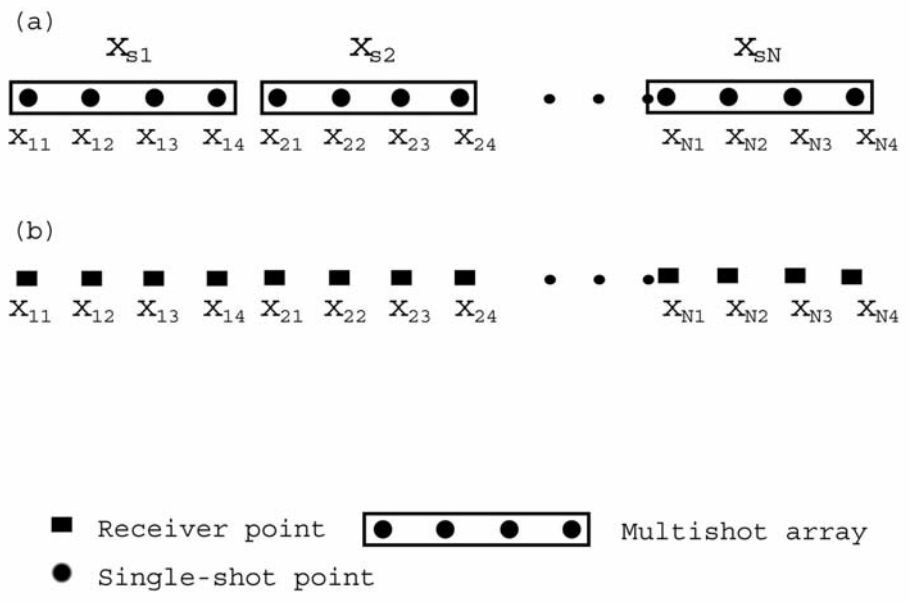


Fig. 3. A description of notations of single-shot points and receiver points in a 1m4s multishooting configuration. (a) Single-shot-point distribution and their nomenclature. (b) receiver-point distribution and their nomenclature.

Passing from (1) and (3) to the Kirchhoff scattering series for attenuating free-surface reflections from multishot data, we simply have to sum the single-shot gathers associated with each multishot location. Thus we arrive at

$$\tilde{P}_P(\mathbf{x}_{sn},\omega,\mathbf{x}_r) = \tilde{P}_0(\mathbf{x}_{sn},\omega,\mathbf{x}_r) - a(\mathbf{x}_s,\omega)\tilde{P}_1(\mathbf{x}_{sn},\omega,\mathbf{x}_r) + a^2(\mathbf{x}_s,\omega)\tilde{P}_2(\mathbf{x}_{sn},\omega,\mathbf{x}_r) - \dots \ , \tag{4}$$

with

$$\tilde{P}_j(\mathbf{x}_{sn},\omega,\mathbf{x}_r) = \sum_{n=1}^{N}\sum_{m=1}^{I} \tilde{P}_{j-1}(\mathbf{x}_{sn},\omega,\mathbf{x}_{nm})V_0(\mathbf{x}_{nm},\omega,\mathbf{x}_r) \ , \tag{5}$$

$$\tilde{P}_0(\mathbf{x}_{sn},\omega,\mathbf{x}_r) = \sum_{m=1}^{I} P_0(\mathbf{x}_{nm},\omega,\mathbf{x}_r) \ . \tag{6}$$

The function $\gamma(n,m)$ describes the encoding coefficients of the various single-shot points. Notice that all the fields in (4) and (5) correspond to a multishooting experiment except one, the vertical component of the particle velocity, $V_0^{(nd)}(\chi,\omega,\mathbf{x}_r)$.

Let us remark that the obvious results in (4) and (5) have important practical implications for the demultiple of single-shot acquired data. Suppose that we have recorded N $\times$ I single-shot gathers. We can regroup the single-shot gathers into multishot gathers $\tilde{P}_0$ while leaving only $V_0^{(nd)}$ in a single-shot-gather domain. This regrouping allows us to reduce the computer storage space of the data before and after the demultiple data by a factor of I. We will also reduce the storage space of the fields of predicted multiples by the same factor. Moreover, the CPU time of predicted multiples will also go down, as we are now predicting N gathers instead of N $\times$ I gathers for each value of j in (5). Another interesting aspect of the results in (4) and (5) is that all the remaining seismic processing steps after the demultiple process can be performed in the multishot gather domain. In other words, the gains realized by demultipling data in the multishot-gather domain can be preserved through the remaining part of the seismic processing chain.

Let us look at a numerical illustration of these results. We have constructed a 1m4s dataset from the single-shot data in Fig. 2a. One gather of the multishot data is shown in Fig. 4a. Notice that three of the four single shots in each multishot gather are simulated with 200 ms, 400 ms, and 600 ms with respect to the first single-shot point. We have added these delays just to confirm that the results in (4) and (5) are independent of the way the single-shot points are encoded. Fig. 5a shows the results of predicted multiples using (5). The Matlab code in Table 1 was used for these computations. Also, we have used the actual field $V_0^{(nd)}(\chi,\omega,\mathbf{x}_r)$ in the single-shot domain in these computations.

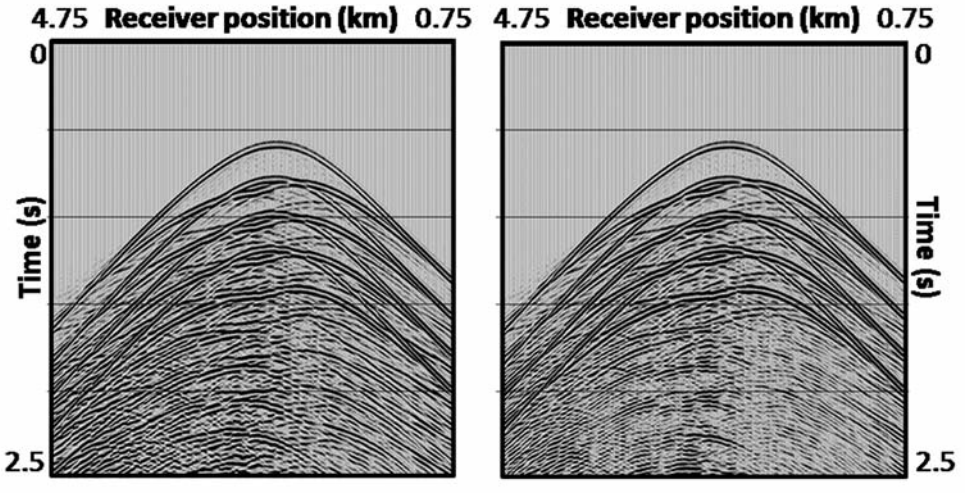


Fig. 4. (a) A multishot gather of a 1m4s-multishot dataset before the demultiple process. (b) The multishot gather in Fig. 4a after the demultiple process. We have used the first three terms of the series in (4).

One way of verifying the accuracy of predicted multiples in Fig. 5a is to compare them to the encoded version of the predicted multiples in Figs. 1b and 1c of single-shot data. In other words, for each field of predicted multiples, we sum four shot gathers with the same encoding time delays as those of the data in Fig. 4a and compared them to the predicted multiples in Fig. 5a of the multishot data. The difference between predicted multiples from multishot data and mixtures of predicted multiples from single-shot data in Fig. 5c are almost null; that is,

$$\tilde{P}_j(\mathbf{x}_{sn},\omega,\mathbf{x}_r) - \sum_{m=1}^{I} \gamma(n,m) P_j(\mathbf{x}_{nm},\omega,\mathbf{x}_r) \approx 0 \quad , \tag{7}$$

thus confirming that we can indeed predict multiples of multishot data using (5), as long as $V_0(\mathbf{x}_{nm},\omega,\mathbf{x}_r)$ is known. The demultiple results of multishot data are shown in Fig. 4b. We have used the first three terms of the series in (4). We can see that this demultiple process is quite effective.

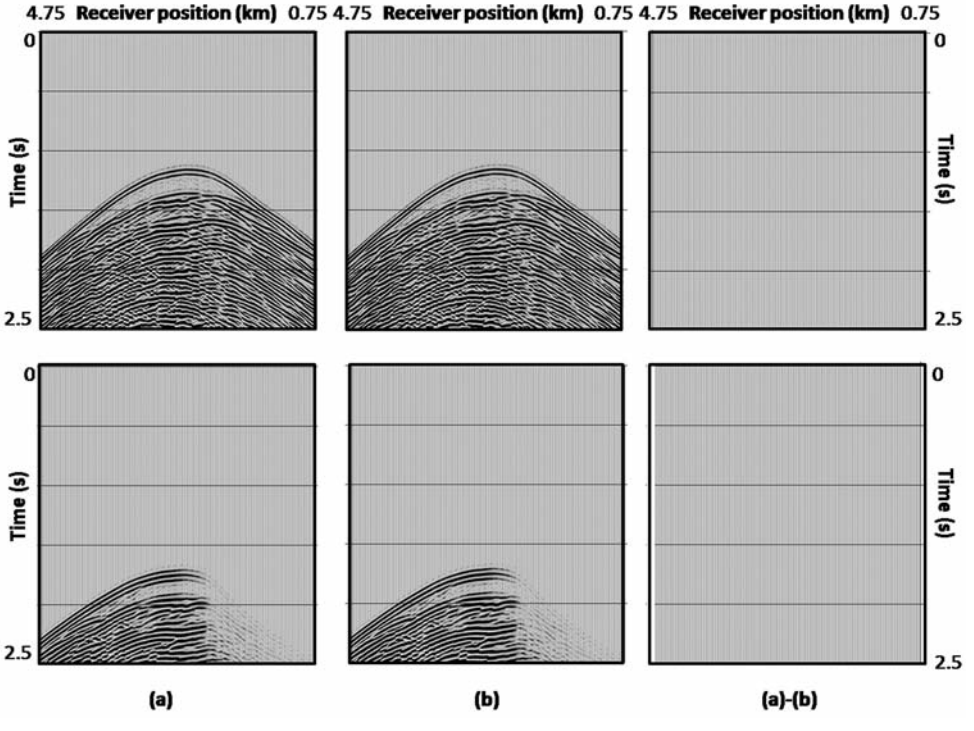


Fig. 5. A comparison of predicted multiples of multishot data using (5) and the encoded version of the predicted multiples of single-shot data. (a) Predicted multiples of multishot data (i.e., $\tilde{P}_1$ and ($\tilde{P}_2$). (b) Encoded version of the predicted multiples of single-shot data; i.e., we sum four shot gathers of each of the fields P_1 and P_2 with the same encoding time delays as those of the data in Fig. 4a. (c) The difference (a)-(b) at the same scale as (a).

Let us emphasize that the demultiple solution described in (4) and (5) and illustrated in Figs. 4 and 5 is valid, irrespective of the encoding system of mixtures. For example, we can encode the mixtures with time delays associated with single-shot points varying from one multishot array to another, even randomly varying, or with no time delays between single-shot points at all. Table 2 summarizes the key steps of this demultiple. Let us also emphasize that after the demultiple process, we can throw away the only single-shot data, $V_0(\mathbf{x}_{nm},\omega,\mathbf{x}_r)$, involved in these computations and carry on the imaging with demultiple multishot data only.

Table 2. A summary of the key steps in demultipling single-shot data in a multishooting form.

Step 1: Input N × I single-shot gathers.
Step 2: Create a second dataset by grouping the N × I single-shot gathers into N multishot gathers, as described in (6).
Step 3: Use a demultiple algorithm like the one in (4)-(6) to attenuate free-surface multiples of multishot data. Throw away any single-shot data and carry on the remaining with multishot data only.
Step 4: Use any migration algorithms for which the input data are in the shot-gather domain to estimate the velocity model and the final migration.

CONCLUSIONS

We have described one way of decreasing the pressure of data processing by reducing the amount of data that are output from the demultiple process and by reducing the computation time of the demultiple process. We also suggest that the imaging process occurring after the demultiple be carried out with multishot data rather than reverting to single-shot data.

ACKNOWLEDGMENTS

We would like to thank the sponsors of the CASP project for their comments and suggestions during the review process.

REFERENCES

Ikelle, L.T., Amundsen, L., Gangi, A. and Wyatt, S., 2001. Kirchhoff scattering series: Insight into the multiple attenuation method. Geophysics, 68: 16-28.

Ikelle, L.T. and Amundsen, L., 2005. An Introduction to Petroleum Seismology. Investigations in Geophysics. SEG, Tulsa, OK.

Weglein, A.B. and Dragoset, W.H., 2005. Multiple Attenuation. Reprint series: Investigations in Geophysics. SEG, Tulsa, OK.