Separation of blended data by iterative estimation and subtraction of interference noise

Panagiotis Doulgeris, Araz Mahdad, and Gerrit Blacquière, Delft University of Technology*

SUMMARY

Conventional data acquisition practice dictates the existence of sufficient time intervals between the firing of successive sources in the field. However, much attention has been drawn recently to the possibility of shooting in an overlapping fashion. Numerous publications have addressed the issue from different scopes (denoising, compressing, blind signal separation etc.) while others have defined the theoretical background. The term 'blending' was introduced to describe this new trend in acquisition designs, the time-overlapping data acquisition. In turn, the term 'deblending' refers to an algorithm that recovers the data as if they were shot in the conventional way. Such an algorithm is presented in this chapter for application on both impulsive and vibrating sources. This algorithm is based on iterative interference estimation and subtraction and is applied to field data.

INTRODUCTION

Blended acquisition designs - i.e. with temporal and spatial overlap between shot records - appear as the next goal of the oil and gas exploration industry. Currently, sources are fired with large time intervals in order to avoid interference, leading to time-consuming and expensive seismic surveys. Furthermore, the source side of the acquisition geometry is often coarsely sampled, causing spatial aliasing. The concepts of simultaneous sources (Beasly et al., 1998) and incoherent shooting (Berkhout, 2008) have been introduced to address these issues by reducing the temporal interval between succesive shots. The separation process ('deblending') of blended data has already been addressed as a blind signal separation problem by Ikelle (2007), using Independent Component Analysis as the tool to distinguish between the different blended sources. Another fastly evolving field, that of Compressive Sensing or Compressive Sampling, has contributed to the separation of blended data through the work of Lin and Herrmann (2009). It is worth mentioning that both these approaches use source encoding (e.g. sweeps or random phase or/and amplitude encoding). An overview of other simultaneous vibroseis acquisition methods that are currently in use -including cascaded and slip sweeps- was described in Bagaini (2006).

By reforming the deblending problem into a denoising one, treating the interference due to the blending as noise, one can use all kinds of signal processing tools available. It has been reported by various authors - e.g. Moore et al. (2008), Ackerberg et al. (2008) - that by sorting the acquired blended data into a different domain than that of the common source gathers, the interference noise appears as random spikes, thus, the separation turns into a typical random noise removal procedure.

Methods that make use of this property of blended data have already been published. Huo et al. (2009) use a vector median filter after resorting the data into common mid-point gathers. This two-dimensional filter acts locally and effectively reduces the amplitude of the random spikes, leading to separation when the data is resorted into common source gathers. Moore et al. (2008) use filters designed in the Radon domain to perform the removal of the random noise.

Spitz et al. (2008) have introduced the idea of building a noise model based on the earth's velocity model and the wave equation. The modeled responses are then used to adaptively subtract the interference noise from the data. This method requires an accurate velocity model and it is, in general, computationally intensive. Moving a step further, Kim et al. (2009) build a noise model from the data itself and then adaptively subtract it from the acquired data. This algorithm acts in the common offset domain and is applied on OBC data.

In the present work, an iterative estimation - subtraction process has been developed for the effective separation of blended data. A noise model is progressively built from the blended data and subsequently subtracted from it. Processing steps such as resorting the data into another domain and filtering can be integrated into this iterative method for both impulsive and vibrating sources. The method is applied on field data, where the blending process has been simulated numerically.

METHOD

The WRW formulation

Berkhout (1982) showed that seismic data can be arranged in the so-called data matrix $\mathbf{P}$. By transforming the data to the frequency domain the entire seismic experiment can be expressed for one frequency component as a matrix multiplication:

$$\mathbf{P}(z_d, z_s) = \mathbf{D}(z_d)\mathbf{X}(z_d, z_s)\mathbf{S}(z_s) \tag{1}$$

where matrix $\mathbf{X}(z_d, z_s)$ contains the earth's impulse response. The matrix $\mathbf{S}(z_s)$ contains the source properties (wavelet, layout, etc.) at the source level z_s and each column describes one source array. Likewise, matrix $\mathbf{D}(z_d)$ contains the detection properties (impulse response of the receivers, layout, etc.) at the detection level z_s and each row describes one detector array. In this arrangement, the $\mathbf{P}_{ij}$ element is a complex valued frequency component of the trace that contains the response of the source array j as recorded by the detector array i. Hence, a column of the matrix $\mathbf{P}$ describes a shot record, whereas a row describes a common detector gather. Source blending can be incorporated in equation 1 as a matrix multiplication with a blending matrix. In the simple case of one large continuous recording, the whole dataset becomes one blended shot record and the $\mathbf{P}$ matrix folds into one vector:

$$\vec{P'} = \mathbf{P}\vec{\Gamma} = \mathbf{DXS}\vec{\Gamma} \tag{2}$$

where the depth terms z_d, z_s have been ommited for notational conveniency. The $\vec{\Gamma}$ vector describes how blending was performed in the field and its elements γ_i are phase and/or amplitude terms. For example, in the simple case of a marine survey

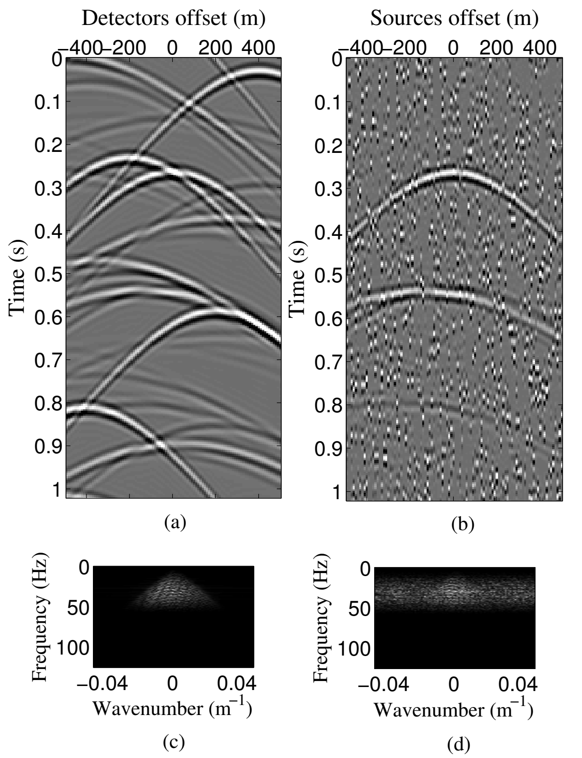

Figure 1: (a) A synthetic pseudo-deblended shot record of 5 unblended shot records, (b) A pseudo-deblended common detector gather for the same case and (c-d) the corresponding 2D spectra.

with random firing times, $\gamma_i = e^{-j\omega\tau_i}$ expressing the time delay τ_i given to each source. Similarly, in the case of vibrating sources transmitting a linear sweep, γ_i becomes a quadratic phase term, $\gamma_i = e^{-j\beta_i\omega^2}$, that describes the source code.

Iterative Deblending

The data acquired during a blended survey contain all the reflectivity information - as long as the sources fire in an incoherent manner, e.g. with random time delays. In order to restore the *unblended* data from the measured data, an inversion process has to be carried out on equation 2. Since this is an underdetermined problem, the pseudo-inverse of the blending vector is computed instead. It can be shown that in the case of phase encoding this is given by the complex conjugate transpose of the vector $\vec{\Gamma}$. Applying this process to the blended data yields:

$$\langle \mathbf{P} \rangle = \vec{P}\vec{\Gamma}^H \qquad (3)$$

where the $\langle \mathbf{P} \rangle$ matrix is the so-called *pseudo-deblended* data matrix and the superscript H denotes the complex conjugate transpose. From the physics point of view, this process corrects for the time delays introduced in the field and decodes the encoded sources, if applicable. However, it does not separate the responses from the different sources that appear as interference noise in the pseudo-deblended result. Such a process will now be discussed.

Since the operation of blending is exactly known -in terms of source coding and/or time delays- the interference introduced to the pseudo-deblended data could be computed exactly if the unblended data $\mathbf{P}$ were known. However, the initial unblended data are not available and obviously, if they were there would be no need of such a deblending method. Suppose though, that *part* of $\mathbf{P}$ could be extracted from the pseudo-deblended data $\langle \mathbf{P} \rangle$. Then, an iterative estimation - subtraction process could be initiated where more of the interference noise could be removed at each iteration. Such a method can be formulated as:

$$\langle \mathbf{P}^{i+1} \rangle = \vec{P}\vec{\Gamma}^H - \overline{\langle \mathbf{P}^i \rangle}(\vec{\Gamma}\vec{\Gamma}^H - \mathbf{I}) \qquad (4)$$

where $\langle \mathbf{P}^{i+1} \rangle$ is the deblended estimate on iteration $i+1$ and $\overline{\langle \mathbf{P}^i \rangle}$ is the deblended estimate on iteration i processed in such a way that only *unblended* data are contained. The second term on the right hand side of equation 4 transforms the estimated unblended data $\overline{\langle \mathbf{P}^i \rangle}$ into interference noise. This is achieved by blending and pseudo-deblending it by applying the term $\vec{\Gamma}\vec{\Gamma}^H$ to it, while making sure that the initial signal is removed by subtracting $\overline{\langle \mathbf{P}^i \rangle}\mathbf{I}$.

This noise estimate can then be adaptively subtracted from the pseudo-deblended data, providing a better estimate of the unblended data. Using this as the new input to the filter, the threshold can be lowered and an even better estimate can be obtained. Repeating this process leads to the gradual removal of interference noise from the pseudo-deblended data, until no further improvement is achieved.

The estimate of the unblended data $\overline{\langle \mathbf{P}^i \rangle}$ can be obtained by a processing step. Any kind of process capable of distinguishing between the signal and the interference noise can be integrated in this step. In our current implementation, a filtering and a thresholding process have been used. Three critical decisions regarding these two processes are:

1. Domain Selection

If the pseudo-deblended data are resorted in a domain perpendicular to the common source domain -such as the common offset, common mid-point or common detector domain- the signal is arranged in coherent events but the blending interference appears as ‘randomly’ distributed spikes, see figure 1b. Hence, a method that can distinguish between coherent and incoherent events, to some degree, shall be able to -partly- suppress the blending interference, see Doulgeris et al. (2010).

Moreover, the method can be implemented in the common source domain if the signal-to-noise ratio is sufficient for the iterative algorithm to start, i.e. at least part of the signal can be extracted from the pseudo-deblended data only by filtering and thresholding. Source encoding, as can be used in land data acquisition, meets this requirement for small blending factors, see Mahdad and Blacquière (2010).

2. Filter implementation

A simple example of a filter that can suppress the interference noise -in a domain perpendicular to the common source domain- is an f-k filter. Such a filter should keep only the part of the pseudo-deblended data that resides in the signal frequency-wavenumber band. The spiky interference noise that appears in these domains is a function that has a white spectrum in the spatial wavenumber direction, extending out of the signal cone, see figure 1d. Thus, by passing only the spatial signal bandwidth, these spikes are somewhat suppressed and we may assume that the highest amplitudes of the pseudo-deblended data now most likely belong to the desired signal **P** only. Obviously, any spike removal tool can be used instead or in combination with the f-k filter.

The spike suppression may also take place in the so-called focal domain, see Berkhout and Verschuur (2006). A-priori information about the subsurface can then be included in the transform operators. In this domain, most of the signal energy maps on a central point -depending on the quality of the operators used- whereas the interference maps elsewhere, thus making the distinction between the two easier.

3. Thresholding

In order to compute the $\overline{\langle \mathbf{P}^i \rangle}$ from the output of the filter, a threshold that selects only the highest amplitudes is applied. This process can take place either in the physical domain (x-t) or in the filtering domain, being integrated in this way in the filtering process.

A stopping criterion has been implemented to monitor and terminate this iterative method when the energy of the output is minimum. The procedure is depicted in Figure 2 for processing in the common detector domain.

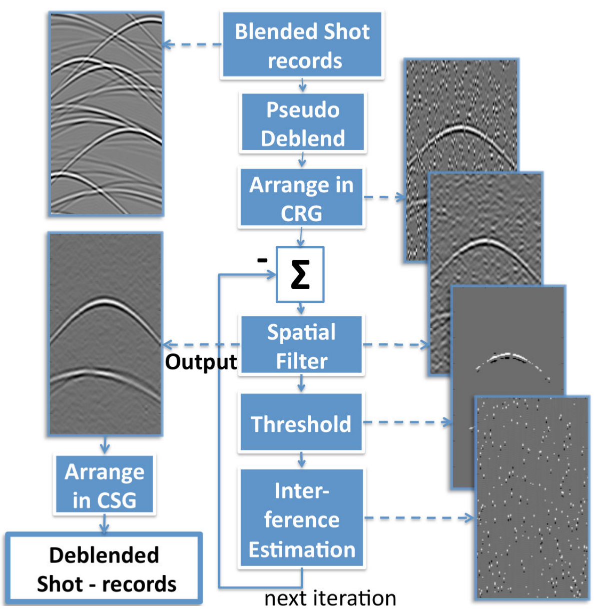

Figure 2: The flowchart of the deblending algorithm (CRG: Common Receiver Gather, CSG: Common Source Gather).

EXAMPLE

A 2-D blended marine survey has been simulated based on unblended field data acquired at the Haltenbanken field in Norway. The dataset was acquired with spatial and temporal sampling intervals of 25 m and 4 ms respectively. The numerically simulated blended acquisition design consists of a 2-D line on which three sources fire with small random time delays. All detectors of the simulated survey record all three sources, resulting in blended shot records that contain negative offsets as well. The additional traces required for the simulation of this survey were computed from the existing data using reciprocity. Figure 3a shows an unblended shot record containing both existing and computed offsets. A shot record of the simulated blended survey, after pseudo-deblending, can be seen in figure 3b. Notice the three different shot records that have been blended into one. The signal-to-noise ratio of this shot record -as far as interference noise is concerned- is around -6 dB (i.e., more noise than signal).

The deblending procedure is carried out in the common offset domain. The near offsets are processed first until no further improvement can be achieved; the middle and far-offsets follow. The filter used for this example is a simple f-k filter and the thresholding process takes place in the x-t domain. Figure 3c shows the deblended estimate obtained after the first iteration. The largest amplitudes of the near offsets have already been used to estimate and then subtract the interference noise. Figures 3d-f show the results of the algorithm after 26, 36 and 44 iterations where the algorithm has finished processing the near, the middle and the far offsets respectively. The interference noise was gradually being removed until the algorithm reached a point where no further improvement could be achieved. The final deblended output of the algorithm is shown in figure 3f. Although some residual energy is left, the result is close to the desired output, with the signal-to-noise ratio being approximately 12 dB. Hence, the enhancement, in terms of signal-to-noise ratio, achieved by this simple implementation of the algorithm is around 18 dB in this example.

SUMMARY AND CONCLUSIONS

An iterative method for the separation of blended data has been presented in this paper. In the method's core exists an interference noise estimation process. The noise estimate is computed based on parts of the signal energy -that can readily be extracted from the blended data- and the blending parameters. This estimate is then subtracted from the blended data. The result, which now contains less blending noise, is the input to the next iteration. This iterative scheme can handle both vibrating and impulsive sources and can integrate any existing denoising technique for the extraction of unblended data from the pseudo-deblended data. However, resorting the data in a domain perpendicular to the common source domain, lets the method separate blended impulsive sources with the aid of a simple f-k filter. Application of the method to a marine dataset, where the blending process has been simulated numerically, produced promising results.

ACKNOWLEDGMENTS

We would like to thank Statoil for the permission to use and publish the marine dataset from the oilfield of Haltenbanken in Norway, and we would like to acknowledge the sponsors of the Delphi Consortium.

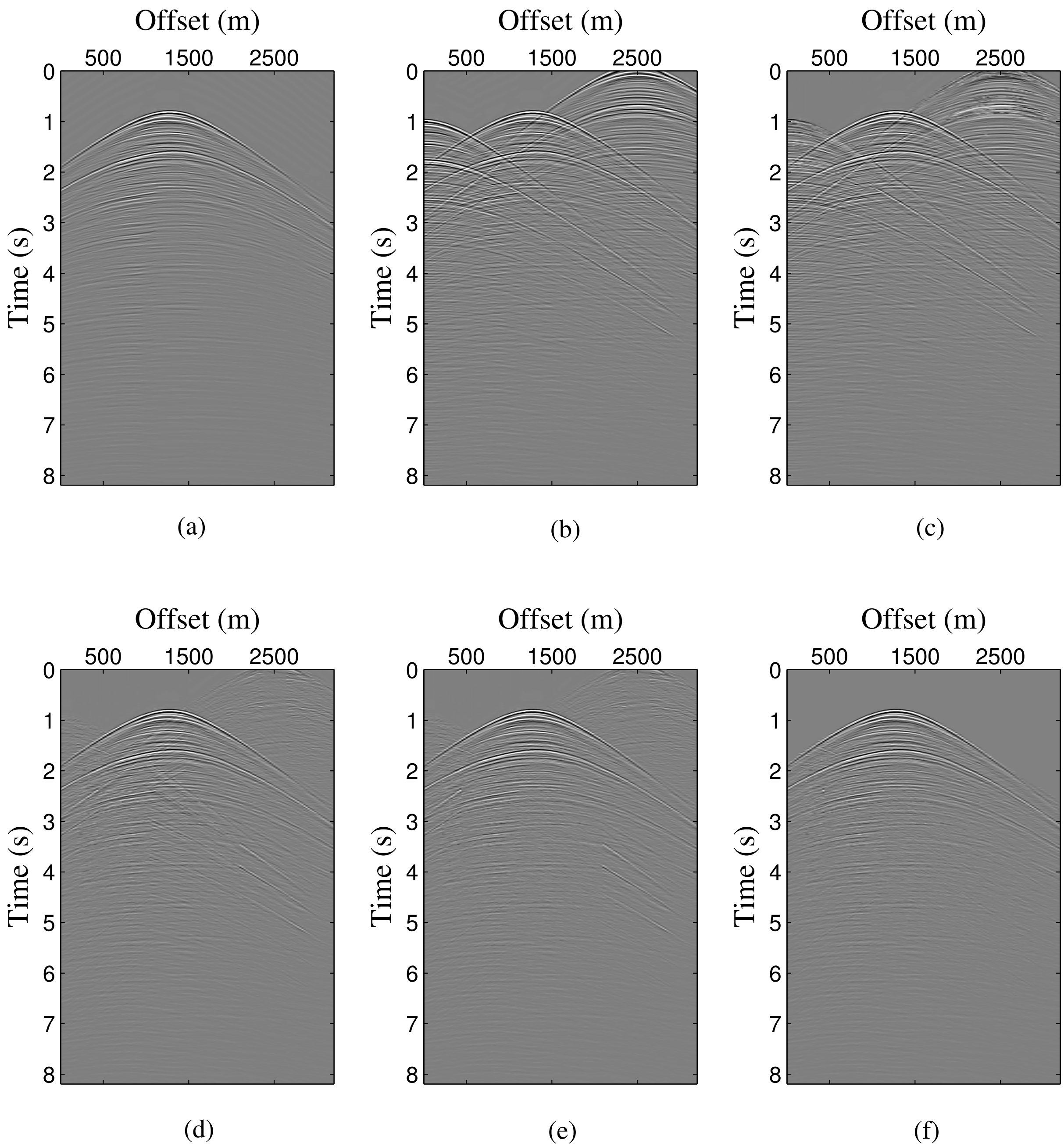

Figure 3: (a) Unblended shot record, (b) initial estimate (pseudo-deblended result), (c) estimate after one iteration, (d) estimate after 26 iterations, (e) estimate after 36 iterations and (f) final deblended result after 44 iterations.