Coherent noise suppression by learning and analyzing the morphology of the data

Pierre Turquais¹, Endrias G. Asgedom², and Walter Söllner²

ABSTRACT

We have developed a method for suppressing coherent noise from seismic data by using the morphological differences between the noise and the signal. This method consists of three steps: First, we applied a dictionary learning method on the data to extract a redundant dictionary in which the morphological diversity of the data is stored. Such a dictionary is a set of unit vectors called atoms that represent elementary patterns that are redundant in the data. Because the dictionary is learned on data contaminated by coherent noise, it is a mix of atoms representing signal patterns and atoms representing noise patterns. In the second step, we separate the noise atoms from the signal atoms using a statistical classification. Hence, the learned dictionary is divided into two subdictionaries: one describing the morphology of the noise and the other one describing the morphology of the signal. Finally, we separate the seismic signal and the coherent noise via morphological component analysis (MCA); it uses sparsity with respect to the two subdictionaries to identify the signal and the noise contributions in the mixture. Hence, the proposed method does not use prior information about the signal and the noise morphologies, but it entirely adapts to the signal and the noise of the data. It does not require a manual search for adequate transforms that may sparsify the signal and the noise, in contrast to existing MCA-based methods. We develop an application of the proposed method for removing the mechanical noise from a marine seismic data set. For mechanical noise that is coherent in space and time, the results show that our method provides better denoising in comparison with the standard FX-Decon, FX-Cadzow, and the curvelet-based denoising methods.

INTRODUCTION

In marine seismic surveys, the seismic wavefield is generally recorded by sensors located in the streamers that are towed by a vessel. The steering devices that are placed along the streamers, as well as barnacles that grow on the surface of the streamers, can perturb the flow of the water and cause local vibrations of the streamers. These vibrations are recorded by the motion sensors and appear in the seismic data. The recording of these vibrations is often referred to as mechanical noise. This noise significantly hinders seismic processing and imaging if incorrectly removed.

Various analytical transforms, e.g., Fourier, wavelet (Mallat, 2008), curvelet (Candès and Donoho, 2000; Candès and Demanet, 2005), or seislet (Fomel and Liu, 2010), can be used to attenuate random noise in seismic data by sparse approximation. The key is to find the transform that represents the data into a domain in which the signal of interest is sparse; i.e., the signal can be represented with a minor part of the coefficients in the transform domain. When the data are represented with such a transform, the random noise in the data is spread along all the coefficients. The noise is then attenuated by approximating the data to its large-amplitude coefficients only. The sparser the signal in the transformed domain, the higher is the noise attenuation. This strategy has been extensively studied for seismic data denoising (Foster et al., 1994; Hennenfent and Herrmann, 2006; Neelamani et al., 2008; Liu and Fomel, 2010).

Sparse representations can also be obtained via data-driven methods. For example, dictionary learning (DL) methods train redundant dictionaries to sparsify specific data. The dictionary is a set of elements called atoms and is learned such that the atoms represent the morphological structures or waveforms that compose the data. As such, it can be said that a dictionary learned on data contains the morphological diversity of the data. The data can later be approximated with a sparse linear combination of the dictionary atoms, which attenuates the random noise. Many methods for DL have
been proposed, e.g., the method of optimal direction (MOD) (Engan et al., 1999), $k$-means singular value decomposition (K-SVD) (Aharon et al., 2006), data-driven tight frame (DDTF) (Cai et al., 2014), sparse K-SVD (Rubinstein et al., 2010), and SuKro (Dansa et al., 2017). MOD and K-SVD learn unstructured dictionaries; there is no constraint on the structure of the atoms. For more efficient training, DDTF, sparse K-SVD, and SuKro learn structured dictionaries: In DDTF, the dictionary is constrained to be a tight frame; in sparse K-SVD, the atoms are constructed as sparse linear combinations of predefined basis functions; and in SuKro, the learned dictionary is a sum of the Kronecker products of smaller dictionaries. The DL methods have proven to perform well for denoising seismic data (Beckouche and Ma, 2014; Liang et al., 2014; Yu et al., 2015, 2016; Zhu et al., 2015; Turquais et al., 2017). Another data-driven method, the Cadzow filtering method (Trickett, 2002, 2008), also called singular spectrum analysis (SSA) (Sacchi, 2009; Chen and Sacchi, 2015), uses rank reduction for denoising. This method embeds each frequency slice of the data into a Hankel matrix, mutes the low singular values, and averages the antidiagonals elements. The noise attenuation is mainly achieved by muting the low singular values because random noise is spread along all the singular values.

Although mechanical noise is generally unpredictable, it is recorded continuously in time and by several neighboring receivers. Therefore, it can appear coherent in space and time in the data. In that case, i.e., if the noise is not entirely random, the effectiveness of sparse approximation-based denoising methods can be degraded, for either case in which the sparse approximation is in a fixed or a data-driven dictionary domain. In the case of a fixed dictionary, the part of the coherent noise that is described by the dictionary is also represented in the sparse approximation. In the case of a conventional DL method, the elementary patterns of the coherent noise are captured in the dictionary during the training step and the noise is represented by the sparse approximation instead of being attenuated.

In seismic processing, coherent noise is often removed by exploiting its coherence property. For instance, the linearity of seismic interference noise helps to separate it from the signal in the $\tau - p$ domain (Yu, 2011). For swell noise, one can use its time-frequency characteristics, which are often different from those of the seismic signal. Vaezi and Kazemi (2016) propose to separate this noise based on non-negative matrix factorization of the power spectrum of its time-frequency representation. The coherence properties of the noise can also be exploited via morphological component analysis (MCA). MCA has been developed to decompose images into different morphological components (Starck et al., 2004, 2005). To separate a noise component from a signal component, the data are represented with a sparse combination of the elements from two dictionaries; each of the dictionaries describing the morphology of one of the components. If the two dictionaries have a low mutual coherence, and the noise and signal have highly sparse representations in their respective dictionaries, MCA separates the signal and the noise correctly (Starck et al., 2004; Bruckstein et al., 2009). For instance, ground-roll noise can be removed from land data by solving the MCA problem using a wavelet transform to represent the signal and a discrete cosine transform to represent the noise (Wang et al., 2010). However, predefining dictionaries to represent the signal and the noise components in a sparse manner is risky because the signal or the noise might not have a sparse representation in its attributed dictionary. In this case, the signal and noise separation is incomplete, and the quality of the denoising is poor.

In this paper, we propose a method that combines DL and MCA to separate coherent noise from seismic data. This method has been briefly introduced by Turquais et al. (2016). A dictionary is learned on the noise-contaminated data using the K-SVD method (Aharon et al., 2006; Rubinstein et al., 2008). The learned dictionary contains both elements representing seismic signal patterns and elements representing noise patterns; they are segregated using a statistical classification (Anderson and Bahadur, 1962). The learned dictionary is hence divided into two subdictionaries; one describing the morphology of the noise and the other describing the morphology of the signal. Both are included into an MCA problem in which the data are represented with a sparsity constraint. This sparsity constraint enables signal and noise separation because the signal cannot be sparsely represented in the noise dictionary, and is therefore represented in the signal dictionary, and vice versa for the noise.

The rest of the paper is organized as follows: The first section presents the theory and methodology, the second section illustrates the proposed method using a simple synthetic example, and the third section shows a successful application for removing the high-frequency mechanical noise from marine seismic data.

**METHODOLOGY**

The proposed method is composed of three steps, namely, DL, atom classification, and MCA. The three steps will be presented separately, and then a workflow will describe how they are assembled to remove coherent noise.

**Dictionary learning**

The first step of the proposed method is using a DL algorithm to extract the morphological diversity of the data as a redundant dictionary. Practically, a dictionary $D$ is a matrix containing unit vectors $a_1, \ldots, a_K$ in its columns (i.e., $D = [a_1 \ldots a_K]$). These unit vectors are referred to as the atoms of the dictionary. Dictionaries are used to compute a sparse representation of a data. Computing a sparse representation of the recording $z \in \mathbb{R}^N$ in a dictionary $D \in \mathbb{R}^{N \times K}$ requires finding a sparse coefficient vector $x \in \mathbb{R}^K$ such that $Dx$ equals or closely approximates $z$. The dictionary is the key element of the sparse representation problem. Its atoms need to describe the morphology of the data to be able to compute a representation that is sparse and accurate. A dictionary representing the morphology of data can be obtained by applying a DL algorithm on the data set or a representative subpart of the data set. The subpart of the data set used to learn the dictionary is referred to as the training set.

For seismic data application, DL is often applied in 2D on a gather. In this case, the training set $z_1, \ldots, z_M$ is a set of 2D patches that have been extracted from the gather and vectorized. One possibility to learn the dictionary on the training set is to find the dictionary $D \in \mathbb{R}^{N \times K}$, with $K \ll M$, and the set of sparse coefficient vectors $x_1, \ldots, x_M$, which minimize the representation error given a sparsity constraint $T$ placed on the sparse coefficient vectors. This minimization problem is mathematically expressed as

$$
\min_{\{x_i\}_{i=1}^M, D} \sum_{i=1}^M \| z_i - Dx_i \|_2^2
$$

subject to $\| x_i \|_0 \leq T, i = 1, \ldots, M.$ (1)
In the proposed method, the DL problem is solved using the K-SVD method (Aharon et al., 2006). The resulting dictionary describes 2D morphological structures in the \( t-x \) domain that are redundant in the training set and complementary for representing the recordings of the data set.

**Atom classification**

In a dictionary learned on an image, the atoms \( a_1, \ldots, a_k \) of length \( N \) describe patterns when rearranged as 2D patches of size \( \sqrt{N} \times \sqrt{N} \). In the case in which the image is a gather that is contaminated by coherent noise, the learned dictionary describes the morphology of the noise and the signal. If the signal and the noise are independently distributed in the window, and if their morphologies have low correlation, the two morphologies are described by different atoms of the dictionary. Hence, the atoms can be classified to create two subdictionaries: one signal dictionary \( D_s \), containing the atoms describing the signal morphology and one noise dictionary \( D_n \) containing the atoms describing the noise morphology. Below, we describe three different methods that can carry out this classification.

**Attributes**

To classify the atoms as signal or noise, it is necessary to use attributes. Here, an attribute is a value that is computed on the 2D pattern described by an atom. The quality of the classification depends on the attributes used to discriminate noise from signal. Therefore, it is necessary to select attributes that have different values for noise and signal atoms. In this work, we use textural attributes that are based on the gray-level co-occurrence matrix (GLCM) (Haralick et al., 1973). The GLCM is a discrete description of the probability of co-occurrence of two gray levels for two pixels with a given relative position in the pattern. For seismic applications, the gray level is the dynamic range that has been rescaled and the pixels are the recorded samples. For the given relative position \( (\Delta t, \Delta x) \), the element at the \( i \)th line and \( j \)th column of the GLCM is the probability of changing from the amplitude \( i \) to \( j \) when moving by \( \Delta t \) samples in time and \( \Delta x \) samples in space. The seismic data are generally stored with 32 bits per sample and can have \( 2^{32} \) possible amplitude values. Because computing a GLCM for such a high number of amplitude values would be expensive, the data are rescaled prior to computing the GLCM. The data are often converted to 4 or 5 bit data where each sample is rescaled to an integer value between 1 and 16 or 1 and 32 (Gao, 2003). The GLCM of a 2D array \( A \) can be computed as described in Algorithm 1.

Textural attributes (e.g., known as energy, homogeneity, inertia) are the weighted sum of the GLCM elements. They have proven to be successful for seismic data classification (Winther et al., 1995; Winther, 1997; West et al., 2002; Gao, 2003). In this study, the attributes used are the inertia for several relative positions. The inertia for the relative position \( (\Delta t, \Delta x) \) is mathematically given by

\[
\text{Inertia} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i-j)^2 P[i,j],
\]

where \( P[i,j] \) is the element at the \( i \)th line and \( j \)th column of the GLCM computed for the relative position \( (\Delta t, \Delta x) \). The larger the probability that two samples separated by \( \Delta t \) samples in time and \( \Delta x \) samples in space have close amplitude values, the lower is the inertia. The inertia is therefore sensitive to the frequency content and the orientation of a pattern. For instance, if a pattern describes a high-frequency signal, it has sharp amplitude variations in time, so the probability that its samples would conserve the same amplitude while moving in time is small, and its inertia is high for non-null \( \Delta t \) and null \( \Delta x \). Similarly, if a pattern describes a flat linear event, it contains samples that conserve the same amplitude while moving only in space, and it has low inertia for null \( \Delta t \) and non-null \( \Delta x \).

**Supervised classification**

A supervised classification analyzes the available examples in a training set to derive a law or condition that can classify new examples. For instance, if a training set contains some atoms labeled as “signal” and some atoms labeled as “noise”, a supervised process can be used to classify the atoms of the learned dictionary into the signal and the noise classes. The supervised processes include multivariate Gaussian classifiers (Anderson and Bahadur, 1962). To classify atoms that are either noise or signal, a multivariate Gaussian classifier assumes that the attributes of the signal and noise atoms follow two different multivariate Gaussian distributions. A multivariate Gaussian distribution is entirely defined by a mean vector \( \mu \), which gives the centroid of the distribution, and a covariance matrix \( \Sigma \), which gives the shape and orientation of the distribution. Such a distribution is denoted with \( \mathcal{N}(\mu, \Sigma) \). The distribution of the signal is defined with the mean vector and the covariance matrix of the attributes computed on the atoms that are labeled as signal. Likewise, the distribution of the noise is defined using the atoms that are labeled as noise. The probability that a vector \( f \) containing \( n \) attributes belongs to a distribution \( \mathcal{N}(\mu, \Sigma) \) is given by

\[
p(f \in \mathcal{N}(\mu, \Sigma)) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^\frac{1}{2}} \exp\left(-\frac{1}{2}(f-\mu)^T\Sigma^{-1}(f-\mu)\right).
\]

In this classification, the formula in equation 3 is used to compute the probabilities that each atom from the dictionary belongs to the signal and noise distributions. Then, the atom is classified as signal or noise according to the highest probability.

**One-class classification**

If the training set contains only atoms labeled as signal or only atoms labeled as noise, the fully supervised model described above cannot be used to classify the atoms of the learned dictionary. In this case, a one-class classification can be performed by computing the probability that each atom belongs to the signal or noise distributions. The probability that a vector \( f \) containing \( n \) attributes belongs to the signal distribution is given by

\[
p(f \in \mathcal{N}(\mu, \Sigma)) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^\frac{1}{2}} \exp\left(-\frac{1}{2}(f-\mu)^T\Sigma^{-1}(f-\mu)\right).
\]

In this classification, the formula in equation 3 is used to compute the probabilities that each atom from the dictionary belongs to the signal distribution. The atom is classified as signal or noise according to the highest probability.

**Algorithm 1. Computation of the GLCM of A for the relative position \((\Delta t, \Delta x)\).**

1: Input: matrix \( A \) of size \( M \times N \); relative position \((\Delta t, \Delta x)\); number \( G \) of integer values.
2: Rescale the samples in \( A \) to a few integer values \( 1, 2, \ldots, G \).
3: Initialize the GLCM, \( P \), of size \( G \times G \) with zeros.
4: for \( i = 1 \) to \( M \), and \( j = 1 \) to \( N \), increment by 1 the sample at the \( A[i,j] \)th line and \( A[i+\Delta t, j+\Delta x] \)th column of \( P \), where \( A[i,j] \) stands for the value at the \( i \)th line and the \( j \)th column of \( A \).
5: Divide \( P \) by the sum of all its elements.
6: Output: \( P \).
this case, a one-class classification (Moya and Hush, 1996; Tax, 2001) is suitable. A one-class classification aims to identify patterns of a specific class among other patterns by learning from a training set containing only the patterns of that class. There are two possible scenarios: Either the training set contains only signal atoms, or it contains only noise atoms. From now on, we will consider that it contains only noise atoms. This does not aim to restrict the application of the method; it is to simplify the explanation and understanding. In that case, a one-class classifier uses the atoms labeled as noise to identify the noise atoms of the learned dictionary and classifies the rest of the atoms as signal. It is assumed that the distribution of the attributes computed on the noise atoms is a multivariate Gaussian distribution. This distribution is defined by the mean vector and the covariance matrix of the attributes computed on the atoms labeled as noise. An atom from the learned dictionary is identified as noise if it has a small Mahalanobis distance to the noise distribution. The Mahalanobis distance for an atom of feature vector \( f \) to a distribution of mean vector \( \mu \) and covariance matrix \( \Sigma \) is given by

\[
d_{M}(f) = \sqrt{(f - \mu)^T \Sigma^{-1} (f - \mu)}.
\]

The Mahalanobis distance is used because it is unitless, scale invariant, and takes into account the correlation of the attributes. If \( f \) follows the multivariate Gaussian distribution of mean \( \mu \) and covariance matrix \( \Sigma \), there is a 68.3% probability that \( d_{M}(f) < 1 \), a 95.5% probability that \( d_{M}(f) < 2 \), and a 99.7% probability that \( d_{M}(f) < 3 \). Keeping this last probability in mind, one can classify an atom as signal if the Mahalanobis distance is smaller than 3 and one can classify an atom as signal as signal if the Mahalanobis distance is higher than 3. The signal atoms would not be misclassified if the density probability function of their attributes does not significantly overlap the one of the noise attributes.

**Unsupervised classification**

If there is no training set, one has to use an unsupervised classification. An unsupervised classification groups together the patterns of a data set that have the closest attributes. In the case in which the number of classes in the set is a priori known and in which multivariate Gaussian distributions are assumed for the classes, the unsupervised classification can be carried out with the \( k \)-means clustering algorithm (MacQueen, 1967). For a given number \( C \) of classes, it finds the clusters \( G_1, \ldots, G_C \) of atoms that minimize the sum of the squared Mahalanobis distances between each attribute vector and its closest cluster centroid. This minimization problem is written such as

\[
\min_{G_1, \ldots, G_C} \sum_{k=1}^{C} \sum_{i \in G_k} (f_i - \mu_k)^T \Sigma_k^{-1} (f_i - \mu_k),
\]

where \( f_i \) is the attribute vector of the atom \( i \) and \( \mu_k \) and \( \Sigma_k \) are the mean vector and the covariance matrix of the attribute vectors in the cluster \( k \). For the signal and noise separation problem, \( C \) is set to 2, and each of the resulting clusters, \( G_1 \) and \( G_2 \), should contain either the noise atoms or the signal atoms.

**Morphological component analysis**

MCA (Starck et al., 2004, 2005) is a method that uses sparse representation as the driving force to separate the different morphological components in a mixture. Consider a recording \( z \) that contains a signal component that can be sparsely represented in a dictionary \( D_s \), and a noise component that can be sparsely represented in a dictionary \( D_n \). The MCA problem consists in finding the sparse representation of the recording in both dictionaries. This can be done by finding the sparse vectors \( x_s \) and \( x_n \) that are the solution of the following minimization problem:

\[
\min_{x_s, x_n} \| z - D_s x_s - D_n x_n \|_2
\]

subject to \( \| x_s \|_0 + \| x_n \|_0 \leq T \),

where \( T \) is the sparsity of the representation. Such a problem can be solved using orthogonal matching pursuit (OMP) (Rezaiifar and Krishnaprasad, 1993). The resulting \( D_s x_s \) and \( D_n x_n \) are sparse approximations of the signal and noise components, respectively. The separation of the components is exact if the sparsity of the recording in the two dictionaries is below a threshold dictated by the mutual coherence of the dictionaries (Starck et al., 2004; Bruckstein et al., 2009). This requires the signal and noise components to both have a very sparse representation in their corresponding dictionary and the correlation between the signal and noise morphologies to be low. The sparse approximations \( D_s x_s \) and \( D_n x_n \) are random noise free because random noise cannot be represented sparsely. For denoising, the component \( D_s x_s \) is of interest because it contains neither coherent noise nor random noise. If the signal is not strictly sparse in its attributed dictionary, \( D_s x_s \) might not represent the entire signal. In this latter case, the signal can be retrieved by subtracting the reconstructed noise component from the recording, i.e., \( z - D_s x_s \). This solution contains random noise but preserves better the signal.

In the proposed method, the atoms in \( D_s \) and \( D_n \) are vectors of length \( N \) describing small 2D patterns of size \( \sqrt{N} \times \sqrt{N} \). To separate the signal from the noise in a 2D gather, MCA needs to be applied to all the juxtaposed patches of size \( \sqrt{N} \times \sqrt{N} \) in the gather. The resulting signal and noise patches need to be respectively assembled to generate the signal and noise components. To obtain a more accurate separation result, MCA can also be applied to overlapping patches. In the latter case, the multiple versions of the same sample are averaged when the signal or noise patches are assembled.

**The proposed workflow**

When the proposed method is used to separate the signal and noise components of a 2D data \( Y \), the workflow can be summarized as follows:

1) Dictionary learning: The K-SVD algorithm is used to learn a dictionary \( D \) from the data \( Y \).
2) Atom classification: The signal and the noise atoms from the dictionary \( D \) are segregated to obtain a dictionary \( D_s \) containing the signal atoms and a dictionary \( D_n \) containing the noise atoms. The segregation of the dictionary atoms is carried out with either a supervised, one-class, or unsupervised classification, depending on the availability of labeled atoms.
Learning and analysis of the data morphology

In this section, we illustrate the proposed morphological decomposition position method with a synthetic example. We synthesized a noisy data by adding a recording of mechanical noise to a window of a synthetic shot gather. The mechanical noise was recorded during a marine survey for the same configuration as the signal was synthesized. The resulting noisy data are of size $100 \times 100$ samples with a sampling of $2$ ms in time and $12.5$ m in space and with a signal-to-noise ratio (S/N) of $2.23$ dB. The S/N in dB of a data $d$ is given by the formula

$$\text{S/N}(d, d_{\text{ref}}) = 10 \log_{10} \frac{||d_{\text{ref}}||_2^2}{||d_{\text{ref}} - d||_2^2},$$

where $d_{\text{ref}}$ is the noise free data reference. Here, the data do not contain the frequencies below $10$ Hz because they were removed due to very poor S/N in this range. The windows of signal, noise, and noisy data are presented in Figure 1. Note that these data are atypically small-sized. We selected a small-scale example because it provides results that are easier to display, explain, and understand, but in practice the method would be more efficient on larger-sized data.

The K-SVD algorithm was used to learn a dictionary on the noisy data in Figure 1. The parameters were the following: For the training, $8000$ patches of size $10 \times 10$ were extracted from the noisy data; the algorithm ran with $15$ iterations; the number of atoms in the learned dictionary was set to $200$; the sparsity threshold, $T$ in equation 1, was set to $8$. The choice of these parameters needs to be chosen considering that within a patch of chosen size, the shape of the noise needs to be different from the shape of the signal, to enable later the separation of the noise atoms from the signal atoms. The number of atoms in the dictionary should be higher than the number of samples in an atom to enforce redundancy in the dictionary. Redundant dictionaries lead to sparser representations, and from our experience, to more accurate signal and noise separation. Also, for an accurate DL, the number of patches in the training set should be several times higher than the number of dictionary atoms. Despite these constraints, we were left with many possibilities for setting the parameters. The rest of the decision was empirically-based as we aimed for a solution that compromises between tractability of the algorithm and accuracy of the representation. However, how to set optimally these parameters is open to discussion. The $200$ atoms of the output dictionary have been rearranged as $10 \times 10$ patches and represented in Figure 2. In this figure, it can be observed that some of the atoms represent signal patterns whereas others represent noise patterns.

The inertia of the GLCM for the relative distances $(\Delta t = 1, \Delta x = 1)$ and $(\Delta t = 1, \Delta x = 0)$ were selected as attributes for the classification of the atoms. A look at the efficiency of these two attributes for discriminating signal and noise atoms is given in Figure 3. The two highest plots show a noise and a signal atom selected from the learned dictionary. Below each atom is successively presented its GLCMs computed for the relative positions $(\Delta t = 1, \Delta x = 1)$ and $(\Delta t = 1, \Delta x = 0)$. The GLCMs have been computed with a number $G$ of integer values set to $16$. For the signal atom and the relative position $(\Delta t = 1, \Delta x = 1)$, the high-value elements of the GLCM are around the diagonal. This indicates that the values of two samples that are separated by one sample in time and one sample in space have a high probability to be close. This is explained by the atom describing an event that dips roughly in the direction of the selected relative position. In contrast, the high-value elements of the GLCM for $(\Delta t = 1, \Delta x = 0)$ are spread more. This is because the amplitude values of the signal pattern are sharply varying in time. For the noise atom, the high-value elements of the GLCM are more concentrated around the diagonal for the relative position $(\Delta t = 1, \Delta x = 0)$ compared with the relative position $(\Delta t = 1, \Delta x = 1)$. This is explained by the noise pattern being smoother in time than in space. For the four GLCMs presented in Figure 3, the inertia is written below the plots. The inertia is small when the high-value elements of the GLCM are close to the diagonal because the inertia is a weighted sum of the GLCM elements where a higher weight is given to the elements further away from

Figure 1. (a) A window of synthetic signal and (b) a window of recorded noise were summed to generate (c) a window of noisy data. The S/N is indicated above the noisy data.
the diagonal (cf. equation 2). Thus, a small inertia for $(\Delta t = 1, \Delta x = 1)$ may be a distinguishing characteristic of a signal atom whereas a small inertia for $(\Delta t = 1, \Delta x = 0)$ may be a distinguishing characteristic of a noise atom.

Later, we will display the location of the atoms in the attribute space, i.e., the space with the attributes as axes. As the dimensionality of this space is equal to the number of attributes, selecting three or four attributes would give a 3D or 4D space, which would be hard to visualize. This is why we limited the number of attributes to two in this example.

To divide the dictionary into a signal dictionary and a noise dictionary, the three types of classification presented in the section "Atom classification" were tested. For the three possibilities, the processes applied are successively presented below:

- For the supervised classification, a noise and a signal model were used. Both models were of size $100 \times 100$ samples and taken from the same gathers where the signal and noise data presented in Figure 1 have been extracted. These models are presented in the two highest plots of Figure 4. For each model, K-SVD was used to learn a dictionary. The parameters used to learn the dictionary on the models are the same as the ones used to learn the dictionary on the noisy data except for the number $K$ of atoms and the sparsity threshold $T$ that are respectively set to 100 and 4. The parameters $K$ and $T$ are twice smaller when learning on the models compared with when learning on the noisy data because there are less data to represent in a signal or a noise model compared to in a mixture of both. The signal and noise output dictionaries are presented in the middle plots of Figure 4. For each atom of the dictionaries, the inertia for $(\Delta t = 1, \Delta x = 1)$ and $(\Delta t = 1, \Delta x = 0)$ were computed. The location of the atoms in the attribute space for both dictionaries is presented in the two lowest plots of Figure 4. We then computed the mean vector $\mu_s$ and the covariance matrix $\Sigma_s$ of the attributes of the 100 signal atoms, and the mean vector $\mu_n$ and the covariance matrix $\Sigma_n$ of the attributes of the 100 noise atoms. Both multivariate Gaussian density functions defined by $(\mu_s, \Sigma_s)$ and $(\mu_n, \Sigma_n)$ are shown in the two lowest plots of Figure 4 with lines of equal probability. For each atom of the learned dictionary, we computed its probability to belong to the signal class and its probability to belong to the noise class. To compute the probability that an atom belongs to the signal class, the formula in equation 3 was used with $\mu_s$ and $\Sigma_s$. Similarly, $\mu_n$ and $\Sigma_n$ were used to compute the probability that an atom belongs to the noise class. Finally, an atom was classified according to the highest probability.

- For the one-class classification, only the noise model was used. Similar to the supervised classification, $\mu_n$ and $\Sigma_n$ were computed. For each atom of the learned dictionary, the Mahalanobis distance was computed using the formula in equation 4. The atoms with a Mahalanobis distance smaller than 3 were classified as noise and the rest as signal.

Figure 2. The dictionary that is learned on the noisy data. Each atom of the dictionary is displayed as a small 2D patch.

Figure 3. Inertia of the GLCM for classification of the signal and noise patterns. A signal atom (a) and a noise atom (d) were selected from the learned dictionary. Below each atom are presented its GLCMs for the relative positions $(\Delta t = 1, \Delta x = 1)$ and $(\Delta t = 0, \Delta x = 1)$. The inertia values of the GLCMs are written below them. The signal atom has low inertia for the relative position $(\Delta t = 1, \Delta x = 1)$, and the noise atom has low inertia for the relative position $(\Delta t = 1, \Delta x = 0)$. 

Figure 4. The dictionary that is learned on the noisy data. Each atom of the dictionary is displayed as a small 2D patch.
For the unsupervised classification, no model was used. The selected attributes were computed on all the atoms of the learned dictionary. Then, the $k$-means clustering algorithm was used to solve the problem in equation 5 with the number of clusters set to 2. The cluster of signal atoms was manually identified, and its atoms were classified as signal. The atoms from the other cluster were classified as noise.

For the three types of classification, the decision map and the classification results in the attribute space are presented in Figure 5. The decision map is the class that would be given to an unlabeled atom in function of its location in the attribute space. The subspace where an unlabeled atom would be classified as signal is colored in blue, whereas the subspace where an unlabeled atom would be classified as noise is colored in red. The location of the atoms from the dictionary presented in Figure 2 is superposed to the decision map. The atoms classified as signal are indicated with blue dots, whereas the atoms classified as noise are indicated with red dots. The three decision maps are very similar, and the classifications of the learned atoms are the same except for five atoms.

The classified signal and noise dictionaries for the three types of classification are presented in Figure 6. To display the signal dictionary, the noise patches are masked with gray patches. Similarly, the signal patches are masked to display the noise dictionary. Five atoms have been classified differently by the three types of classification; their positions in the dictionaries have been indicated with black framed boxes.

Finally, MCA was applied to separate the signal and noise components. The OMP algorithm was used to solve the problem presented in equation 6 for patches of size $10 \times 10$ overlapping on nine samples in both dimensions. The sparsity threshold $T$ was set to 8, as when the dictionary was learned. For the three pairs of signal and noise dictionaries resulting from the classifications, the separated components are presented in Figure 7a. The S/N of the signal components is given under the higher plots. The differences between the true and recovered components are shown in Figure 7b. The three results provide high noise attenuation as well as signal preservation.

For this example, the three classifications seem to have equivalent effectiveness. The unsupervised classification has managed to place the signal atoms in one cluster and the noise atoms in the other cluster because there was a large quantity of both of them in the dictionary. If there would have been much fewer noise atoms than signal atoms, the unsupervised classification could have divided the signal atoms into two clusters to minimize the cost function in equation 5. The signal would have been separated in two parts instead of being separated from the noise. Therefore, a supervised or one-class classifier would be preferred in cases in which the method is blindly applied.

**FIELD DATA APPLICATION**

**Steering device-related noise removal**

We selected a raw shot gather acquired during a marine survey. The data are the vertical particle velocity sampled at 2 ms in the...
Figure 5. Decision map for the supervised, one-class, and unsupervised classifications. The blue and red areas are the subdomains in which an atom is classified as signal and noise, respectively. The location of the dictionary atoms is indicated with blue dots for the atoms classified as signal and with the red dots for the atoms classified as noise.

Figure 6. Classified signal and noise dictionaries for the supervised, one-class, and unsupervised classifications. The black boxes point out the positions of the atoms for which the results of the classifications differ.
temporal dimension and 12.5 m in the offset dimension, and they are contaminated by steering device-related noise. The frequency range 0–10 Hz was muted due to the very poor S/N. Denoising this frequency range is not in the scope of this work. The muted signal is generally retrieved using pressure measurements (Day et al., 2013). This shot gather is shown in Figure 8.

We applied the proposed method to the presented shot gather. We learned a dictionary of 4000 atoms using the K-SVD algorithm. The parameters were the following: seven iterations of the algorithm were run, 40,000 patches of size 12 × 12 samples were used for the training, and the sparsity threshold \( T \) was set to 8. For the classification of the learned dictionary, three attributes were selected; they were the inertia for the relative positions \( (\Delta t = 1, \Delta x = 0) \), \( (\Delta t = 0, \Delta x = 1) \), and \( (\Delta t = 1, \Delta x = 1) \). For such a field data application, a noise-only model is available in the upper right part of the shot gather; the area where the first arrival of the seismic waves is not yet recorded. Due to the possibility to have a noise model, we used a one-class classifier to segregate the atoms of the learned dictionary. The noise model used was the data between 1 and 2 s in time and 4 and 6.5 km in space. The one-class classification was applied in the same way as it was applied in the synthetic example presented earlier. Because three attributes were selected, the classification was carried out given the location of the atoms in a 3D attribute space. Because it is hard to visualize the location of the atoms in this space from a 3D plot, we present the results via projections of the space onto planes; in Figure 9, each atom is located given its inertia for the relative distances (panel a) \( (\Delta t = 1, \Delta x = 1) \) and \( (\Delta t = 1, \Delta x = 0) \), (panel b) \( (\Delta t = 0, \Delta x = 1) \) and \( (\Delta t = 1, \Delta x = 0) \), (panel c) \( (\Delta t = 1, \Delta x = 1) \) and \( (\Delta t = 0, \Delta x = 1) \). If an atom was classified as signal, the location is indicated with a blue dot, and if it was classified as noise, the location is indicated with a red dot. In Figure 9d–9f, we give a view on the shape of the atoms depending on their location in the attribute space. The attribute spaces presented in Figure 9a–9c were divided into small squares, and for each square, an atom that was located in the square was displayed. When several atoms were located in the square, the closest to the center of the square was displayed. If no atom was in the square, no atom was displayed. In addition, the atoms that were classified as noise were framed with red boxes. The two dictionaries resulting from the classification were used to separate the signal and the noise with MCA. The separation was carried out for patches of size 12 × 12 overlapping on 10 samples in both dimensions and with the sparsity threshold \( T \) set to 8. To well preserve the signal, the noise component was subtracted from the data to obtain the denoised data. This is equivalent to adding the residual of the sparse approximation to the signal component.

The proposed method was compared with academic implementations of FX-Decon (Gulunay, 1986), FX-Cadzow (Trickett, 2002), and curvelet (Hennenfent and Herrmann, 2006) denoising. For the FX-Decon method, the algorithm used was from SeisLab and corresponds to the implementation proposed by Ulrych and Sacchi.
where both forward and backward error prediction filters are used. The filtering was applied to windows of size $50 \times 50$ samples overlapping on 25 samples in both dimensions and with filters of size six samples. These parameters have been selected because they perform well on this example and they have been shown to give the best denoising results on other examples (Chen et al., 2016). FX-Cadzow was applied with a rank parameter of 1 and with windows of size $130 \times 8$ samples overlapping on 65 samples in time and six samples in space (Oropeza, 2010). In the curvelet method, the curvelet coefficients of the data were calculated using wrapping curvelets, 16 angles, and a spgl1 solver for the $l_1$ norm constraint. Then, the smallest curvelet coefficients that correspond to 4% of the data energy were muted for denoising.

The denoised data obtained with the proposed method, FX-Decon, FX-Cadzow, and the curvelet method are presented in Figure 10. There, we observe that the proposed method provides a cleaner result compared with the other methods. The removed noise gathers, i.e., the difference between the input and the denoised gathers, are presented in Figure 11. There, we observe that the proposed method preserves better the signal compared with the other methods.

We shall examine the denoising results obtained with the proposed method in the $f-k$ domain. In the $f-k$ spectrum of the noisy data presented in Figure 12a, we see that the noise is overlapping the signal between 10 and 50 Hz. In the spectra of the denoised data and removed noise presented in Figure 12b and 12c, we observe that the noise has been effectively removed when the seismic reflections remain untouched. However, we see that a minor part of the direct wave has also been removed.

### Barnacle-related noise removal

We selected a second raw shot gather acquired during another marine survey. The data are the vertical particle velocity sampled at 2 ms in the temporal dimension and 12.5 m in the offset dimension. We removed the frequencies below 10 Hz for the reason explained in the previous example. This time, the shot gather is contaminated by barnacle-related noise. This shot gather is presented in Figure 13.

We applied the proposed method, FX-Decon, and FX-Cadzow with the same parameters as the ones used in the previous example. In the curvelet method, the removed energy was increased to 10% because there is more noise in this shot gather compared with the previous example.

The denoised data obtained with the proposed method, FX-Decon, FX-Cadzow, and the curvelet methods are presented in Figure 14. The proposed method removes more noise than FX-Cadzow and the curvelet method but slightly less noise than FX-Decon. For the four methods, the removed noise sections are presented in Figure 15. In these gathers, we observe that the proposed method better preserves the signal compared with the other methods.

---

**Figure 8.** Shot gather contaminated by steering device-related noise. The close-up on the data framed by the yellow dashed box is shown in the top-right corner of the plot.

**Figure 9.** Classification of the dictionary learned from the data contaminated by steering device-related noise: (a, b, c) location of the atoms in projections of the 3D attribute space. The location of the atoms classified as signal are indicated with blue dots, and the location of the atoms classified as noise are indicated with red dots. (d, e, f) The shape of atoms is presented given their location in the projections of the attribute space. The atoms that were classified as noise are framed with red boxes.
In the previous example, we observed that the proposed method was better than FX-Decon for removing steering device-related noise. For removing the barnacle-related noise of this example, it is not better. This could be explained by the different characteristics of the noises; the barnacle-related noise was less sparse and less coherent in space than the steering device-related noise. The fact that the barnacle-related noise was less sparse reduced the efficiency of the proposed method because the sparse representation is less accurate when the sparsity decreases (Bruckstein et al., 2009). The fact that the barnacle-related noise was less coherent from one trace to another benefited FX-Decon, which is hindered when the noise is linear.

The f-k spectra of the data and the results obtained with the proposed method are presented in Figure 16. This time, we observe that the noise is overlapping the signal in the frequency range of 10–70 Hz. In the spectra of the denoised data and removed noise, we observe that the noise has been effectively removed when the seismic signal is preserved. These results attest that the proposed method removes noise that cannot be removed with simple frequency and/or wavenumber-based muting. They also show that the noise is highly aliased and suggest that it does not affect the proposed method. Aliasing often affects denoising methods that remove noise based on predefined dip information because aliasing leads to dip miscalculations. The proposed method, on the other hand, is not denoising based on predefined dip information.

**DISCUSSION**

In the proposed method, the strength of the denoising is mainly controlled by the sparsity threshold $T$. If the threshold is low, only the main noise features are reconstructed and separated; the amount of removed noise is hence small. On the contrary, if the threshold is high, the data are almost entirely reconstructed and separated; almost all the noise is removed. The accuracy of the signal and noise separation can be increased by simultaneously increasing the patch size, the number of patches in the training set, and the number of atoms in the dictionary. As the patch size increases, the signal and noise contents of the patches are less correlated, which improves the classification and MCA steps of the denoising process. As will be shown later in the analysis of the method complexity, increasing these parameters also increases the run time.

We shall justify the choice of the methods that were compared with the proposed method in the field data examples. The noise in these examples is not impulsive in time, so it cannot be removed with a median filter. Neither is it a white Gaussian noise, so it cannot be attenuated with conventional DL methods. It is spread on a large
frequency range and has low coherency from trace to trace, so it could hardly be isolated and removed in the f-x, f-k, wavelet, or curvelet domain. However, since it is not highly coherent from trace to trace, it does not appear linear in the data, and that is why it can be filtered using FX-Decon, which reconstructs only the events that are linearly predictable. Likewise, FX-Cadzow attenuates non-linear events and can be used to filter such noise. As is shown in Figures 12 and 16, the noise appears quite white within a considerable part of the f-k spectrum. This indicates that it cannot be represented with few curvelets because a curvelet is localized in the f-k domain. Contrary to that noise, the seismic wavefield is usually sparse in the curvelet domain (Hennenfent and Herrmann, 2006). Hence, the compared curvelet method is based on the following assumption: Because the signal is concentrated on few curvelet coefficients and the noise is spread on many, the signal coefficients have a high amplitude while the noise coefficients have a low amplitude. Therefore, picking only the largest coefficients may preserve the signal and attenuate the noise. The results obtained with this method suggest that the noise was too correlated with the curvelets for the method to work effectively. We note that there might be better ways to use the curvelet domain for removing such noise.

In the field data examples shown, we observe signal leakage for the proposed and compared methods. This signal leakage could be decreased by integrating additional steps to the denoising process. For instance, one could add a noise detection step (Bekara and van der Baan, 2010). This would consist of identifying the part of the data that is contaminated by noise. The identification could be in the t-x or f-x domain and would result in the location of the noisy samples in the t-x or f-x domain, correspondingly. In parallel, the denoising method would be applied to the data to get a full noise model, but only the part of the noise model that was flagged as noisy during the detection would be removed from the data. It would guarantee to preserve the signal at locations where the data are not or little noisy. Another additional step in the processing could consist in detecting eventual signal in the obtained noise model and adding it back to the denoised results. Such additional processing steps are often applied in industrial processing.

For the presented examples, the proposed method was more expensive than FX-Decon but cheaper than FX-Cadzow or the curvelet method. For instance, denoising the shot gather in Figure 10 took 8.48 min on one CPU for the proposed method when it took 1.01, 21.48, and 36.62 min for FX-Decon, FX-Cadzow, and the curvelet methods, respectively. These run times were found for academic implementations of the different methods; be aware that they may not reflect run times found for industrial implementations. In the proposed method, the major part of the time is used to learn the dictionary. For this part of the method, we use the implementation of the K-SVD algorithm proposed by Rubinstein et al. (2008). The authors of this implementation evaluate the number of operations per iteration of the algorithm to approximately $M \cdot (2NK + T^2K +$...
$7TK + T^3 + 4TN) + 5NK^2$, where $M$ is the number of patches in the training set, $N$ is the number of samples in a patch, $K$ is the number of atoms in the dictionary, and $T$ is the sparsity of each coefficient vector.

In the proposed method, we selected only three textural attributes for the classification of the atoms. They have proven to be very discriminative, and no additional attributes were needed to identify a mechanical noise pattern from a signal pattern. There is no guaranty that they would discriminate any other types of noise. This is why increasing the number of attributes for the classification would lead to a more robust method that could target many different types of noises. With more attributes, the method could be used for other applications such as removing algorithm artifacts or noise resulting from simultaneous source shooting. For an application in which there is no noise or signal model, the method could still be used with an unsupervised classification. In contrast, if a noise and a signal model are available, the method could be applied with a supervised classification. Yet, this method would have limitations on the types of noise that it could separate from seismic data. The separation would be complete only if the morphology of the noise is different from the morphology of the signal and if the occurrence of the noise in the data is independent from the occurrence of the signal. For instance, it could not remove multiples because the atoms could not be classified as signal or noise since they would represent both, i.e., a primary and a multiple. Neither could it remove a processing artifact that would smear from the signal. This artifact would be systematically associated to the signal, and, therefore, atoms that contain both signal and noise would be learned in the dictionary. These atoms could not be later classified as signal or noise because they would contain both.

The proposed method can be improved in several other ways. To increase the accuracy of the sparse representation, basis pursuit (Chen et al., 1998) could be used instead of OMP for solving the MCA problem. To speed up the denoising process, K-SVD could be replaced with a more efficient DL method, e.g., DDLF, or sparse K-SVD. However, we note that replacing K-SVD with sparse K-SVD could affect the effectiveness of the proposed denoising process. Sparse K-SVD learns the atoms as sparse linear combinations of fixed basis functions in contrast to K-SVD that learns unstructured atoms. Using sparse K-SVD in the proposed process would result in signal and noise dictionaries that are sparse linear combinations of the same fixed basis functions. This could increase the mutual coherence between the signal and noise dictionaries and decrease the effectiveness of MCA, which would reduce the accuracy of the signal and noise separation (Starck et al., 2004; Bruckstein et al., 2009). Finally, an extension of the method in 3D is straightforward and would benefit from a better description of the seismic wavefield.
CONCLUSION

We proposed a new sparsity promoting method for removing coherent noise from seismic data. In this method, a dictionary is learned from the data and divided into two subdictionaries: one describing the morphology of the signal and the other one describing the morphology of the noise. Then, both subdictionaries are used to separate the noise from the signal via MCA. For that step, the subdictionaries are optimal because they have been specifically trained to provide a very sparse representation of the data. Unlike conventional DL based methods, the proposed method can remove coherent noise. In addition, this method does not require a manual search for optimal transforms that may sparsify the signal and the noise, in contrast to existing MCA-based denoising methods. We used the proposed method to remove the barnacle and steering device-related noise from two field data examples and compared the results with those of FX-Decon, FX-Cadzow and the curvelet-based denoising method. The proposed method provided the best results for removing the steering device-related noise. For removing the barnacle-related noise, the proposed method performed as good as FX-Decon and better than FX-Cadzow and the curvelet-based denoising method.

ACKNOWLEDGMENTS

We acknowledge the SLIM Institute that provided us with the software of the curvelet based denoising method. We also thank R. Rubinstein who made available the K-SVD software that was used in the proposed method. We would like to express our gratitude to the G&E team of PGS for the fruitful discussions on the subject, and we thank PGS for permission to publish this paper. Finally, we wish to thank the assistant editor J. Etgen, the associate editor I. Moore, the reviewers M. Aharchaou, N. Kazemi, and J. Ma, for valuable and constructive comments.

REFERENCES


Learning and analysis of the data morphology


Sacchi, M. D., 2009, FX singular spectrum analysis: Presented at the CSEG CWLS Convention.


