# Double sparsity dictionary for seismic noise attenuation[a]

*Yangkang Chen*, Jianwei Ma† and Sergey Fomel**

## ABSTRACT

A key step in sparsifying signals is the choice of a sparsity-promoting dictionary. There are two basic approaches to design such a dictionary: the analytic approach and the learning-based approach. While the analytic approach enjoys the advantage of high efficiency, it lacks adaptivity to various data patterns. On the other hand, the learning-based approach can adaptively sparsify different datasets but has a heavier computational complexity and involves no prior-constraint pattern information for particular data. We propose a double sparsity dictionary (DSD) for seismic data in order to combine the benefits of both approaches. We provide two models to learn the DSD: the synthesis model and the analysis model. The synthesis model learns DSD in the data domain, and the analysis model learns DSD in the model domain. We give an example of the analysis model and propose to use the seislet transform and data-driven tight frame (DDTF) as the base transform and adaptive dictionary respectively in the DSD framework. DDTF obtains an extra structure regularization by learning dictionaries, while the seislet transform obtains a compensation for the transformation error caused by slope dependency. The proposed DSD aims to provide a sparser representation than the individual transform and dictionary and therefore can help achieve better performance in denoising applications. Although for the purpose of compression, the proposed DSD is less sparse than the seislet transform, it outperforms both seislet and DDTF in distinguishing signal and noise. Two simulated synthetic examples and three field data examples confirm a better denoising performance of the proposed approach.

## INTRODUCTION

Sparse approximations aim at representing the most information of the given data by a linear combination of pre-specified atom signals with sparse linear coefficients. Sparse approximation theory has been a rapidly evolving field, since many state-of-the-art signal and image processing tasks have been successfully handled with the concept of sparse representation, including image inpainting and restoration (Elad et al., 2005; Mairal et al., 2008, 2009; Quan et al., 2011; Cai et al., 2013), image denoising (Protter and Elad, 2009; Cai et al., 2013), data compression (Bryt, 2008), and

[a] Published in Geophysics, 81, no. 2, V17-V30, (2016)

blind source separation (Zibulevsky and Pearlmutter, 2001). The majority of known methods for sparse approximation of signals can be divided into two general categories: an analytic approach and a learning-based approach. The analytic approach refers to a fixed basis while the learning-based approach adaptively finds the required sparse basis by training. *Transform* is usually applied to refer to analytic approach while *dictionary* refers to learning-based approach. A number of sparsity promoting transforms with fixed basis functions have been proposed in the literature for handling different signal processing tasks, including wavelets (Sweldens, 1995; Mallat, 2009), curvelets (Candès et al., 2006; Ma and Plonka, 2010), contourlets (Do and Vetterli, 2005), shearlets (Labate et al., 2005), and bandelets (LePennec and Mallat, 2005), as well as the classic Radon transforms (Ibrahim and Sacchi, 2014b,a). The learning-based dictionaries usually utilize machine learning techniques to infer the dictionary. The advantage of this type of approach is finer-tuned dictionaries compared with the analytic approach. Tuned dictionaries can result in better performance in different applications. The downside of the learning-based approach is a higher computational cost because of a large number of iterations and redundant overlap computations for small patches. Different algorithms for learning dictionaries may result in different efficiencies and performances. Commonly used dictionary-training algorithms include principle component analysis (PCA) (Jolliffe, 2002; Vidal et al., 2005), the method of optimal directions (MOD) (Engan et al., 1999), and variants of singular-value decomposition (SVD) such as K-SVD (Aharon et al., 2006).

Over the past several decades, different types of fixed-basis sparsity-promoting transforms have been explored for seismic data processing applications, and promising results have been reported. Luo and Schuster (1992) applied a wavepacket transform to seismic data compression. Zhang and Ulrych (2003) developed a type of wavelet frame that takes into account the characteristics of seismic data both in time and space for denoising applications. Ioup and Ioup (1998) applied a wavelet transform to both random noise removal and data compression using soft thresholding in the wavelet domain. Du and Lines (2000) applied a multi-resolution property of wavelet transform to attenuate tube waves. Jafarpour et al. (2009) used a discrete cosine transform to obtain sparse representations of fields with distinct geologic features and to improve the solutions of traditional geophysical estimation problems.A number of researchers have reported successful applications of the curvelet transform in coherent and random noise attenuation thanks to the multi-scale directional property of the curvelet domain (Hennenfent and Herrmann, 2006; Wang et al., 2008; Neelamani et al., 2008, 2010). Following the compressive sensing theory (Donoho, 2006), curvelet transform has also been used to restore missing seismic data (Hennenfent et al., 2010; Naghizadeh and Sacchi, 2010). The fixed-basis sparse transforms enjoy efficiency benefits. However, they are not data-adaptive.

Fomel and Liu (2010) introduced a sparsity promoting transform which can be data adaptive, called the *seislet* transform. Following the lifting scheme used in the construction of second-generation wavelets, the seislet transform utilizes the spatial predictability property of seismic data to construct the predictive operator. Fomel and Liu (2010) used plane-wave destruction (Fomel, 2002) to aid the prediction process. Instead of using the plane-wave destruction algorithm, Liu and Fomel (2010) used differential offset-continuation to construct seislet transform for prestack reflection data. The offset continuation seislet transform can obtain better sparsity for conflicting-dip events in prestack data by using offset continuation instead of local slopes to connect different common offset gathers.

The learning-based dictionaries have not been widely used in seismic data processing until recent years. Kaplan et al. (2009) used a data-driven sparse-coding algorithm to adaptively learn basis functions in order to sparsely represent seismic data in the transform domain and performed improved denoising performance for both synthetic and field data examples. Based on a variational sparse-representation model, Beckouche and Ma (2014) proposed a denoising approach by adaptively learning dictionaries from noisy seismic data. Liang et al. (2014) applied a learning-based sparsity-promoting dictionary, called data-driven tight frame (DDTF) (Cai et al., 2013), as a sparse transform in the framework of the split inexact Uzawa algorithm to restore missing seismic data. The split inexact Uzawa algorithm was proposed by Zhang et al. (2011) and can be viewed as a generalization of the alternating direction of multiplier method (ADMM). While learning-based dictionaries can be more adaptive than fixed-basis transforms, no prior-constraint structural information is involved in the construction of these dictionaries. As seismic data have distinct structural patterns, the general learning-based dictionaries can be improved by incorporating structure information for achieving additional sparsity.

In this paper, we propose a double sparsity dictionary (DSD) for sparsifying seismic data. The basic concept of DSD is borrowed from the image processing literature (Rubinstein et al., 2010; Ophir et al., 2011). Rubinstein et al. (2010) initialized the basic principle of a sparsity model of the dictionary atoms over a base dictionary. Ophir et al. (2011) extended the method of Rubinstein et al. (2010) largely by grouping the analysis-domain data into bands. By bridging the gap between fixed-basis transforms and learning-based dictionaries, we propose a cascaded DSD framework, which aims at combining the benefits of both approaches. The proposed DSD framework offers an extra level of sparsity for representing seismic data. On the one hand, DSD compensates the weakness of fixed-basis transforms in sparsifying seismic data patterns and thus can be more robust. At the same time, DSD can relieve the dependence of learning-based dictionaries on large filtering-window overlap and large number of iterations and thus can be more efficient and effective. We define two models to construct DSD: a synthesis model and an analysis model; according to the learning domain of each model. We provide an example of the analysis model, in which we cascade the seislet transform and DDTF to form a DSD framework. In this framework, DDTF compensates for the dip dependence of the seislet transform, while the seislet transform adds structural regularization to DDTF. Thus, the proposed DSD can provide sparser representation of seismic data in order to implement thresholding-based denoising algorithms. We use two synthetic tests and three field data examples to test the performance of the proposed DSD framework in removing random noise. Our experiments show noticeably better performance using DSD-based thresholding than using either seislet or DDTF based thresholdings. For the presented examples, DSD also performs significantly better than the classic $f$-$x$ deconvolution, according to both SNR measurements and visual observations.

# THEORY

## Double sparsity dictionary

Sparse representation for a given signal $\mathbf{d}$ can be formulated as the following optimization problem (Rubinstein et al., 2010; Ophir et al., 2011):

$$\hat{\mathbf{m}} = \arg\min_{\mathbf{m}} \parallel \mathbf{m} \parallel_0, s.t. \parallel \mathbf{d} - \mathbf{A}\mathbf{m} \parallel_2 \leq \epsilon. \tag{1}$$

Here, $\mathbf{m}$ is the sparse representation of the observed signal $\mathbf{d}$, $\mathbf{A}$ is the sparsity-promoting transform (or dictionary), and $\hat{\mathbf{m}}$ is the estimated sparse representation. $\parallel \cdot \parallel_0$ and $\parallel \cdot \parallel_2$ denote the $L_0$ norm and $L_2$ norm of a vector, respectively, and $\epsilon$ is the error tolerance. The double sparsity model states that the sparsity promoting transform $\mathbf{A}$ can be implemented as a cascaded combination of two dictionaries:

$$\mathbf{A} = \mathbf{B}\mathbf{W}, \tag{2}$$

where $\mathbf{B}$ is a base analytic transform and $\mathbf{W}$ is an adaptive learning-based dictionary. The base dictionary can be a fixed-basis sparsity-promoting dictionary, while the adaptive dictionary can be adaptively learned over the compact representation provided by $\mathbf{B}$.

Compared with a fixed-basis dictionary, the double sparsity model as defined in equation 2 can provide more adaptability by modifying $\mathbf{W}$. It acts as an extension to the initial base dictionary $\mathbf{B}$. Compared with learning-based dictionary, the double sparsity model can be more efficient and stable because the data to be learned have a relatively more compact representation and thus the initial base transform provides a structure regularizer to the learning process. The double-sparsity transform domain coefficients $\mathbf{m}$ obtain two-level sparsity: base transform provides the first-level sparsity and the learning dictionary provides the second-level sparsity. In this respect, the philosophy of our work is also similar to the bandlets (LePennec and Mallat, 2005): rather than seeking the direct transform to get the ultimate sparsification of the input signals, one can first use an existing transform that does reasonably well in that respect, and then add another layer of processing that squeezes more over the already simplified signals (Ophir et al., 2011). In the next two sections, we introduce two models to learn the double sparsity dictionary (DSD) for seismic data: synthesis-based DSD and analysis-based DSD.

## Learning DSD in data and model domains - synthesis and analysis models

Provided that the learning-based dictionary is a tight frame such that $(\mathbf{W}^T)^T\mathbf{W}^T = \mathbf{W}\mathbf{W}^T = \mathbf{I}$, DSD can be learned by considering the following problem:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{m}} \frac{1}{2} \parallel \mathbf{d} - \mathbf{B}\mathbf{W}\mathbf{m} \parallel_2^2, \\ &s.t. \parallel \mathbf{m} \parallel_0 \leq t \quad and \quad \mathbf{W}\mathbf{W}^T = \mathbf{I}, \end{aligned} \tag{3}$$

where $t$ denotes the sparsity constrained coefficient.

Equation 3 can be solved alternatively by minimizing

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{m}} \frac{1}{2} \parallel \mathbf{d} - \mathbf{B}\mathbf{W}\mathbf{m} \parallel_2^2 + \lambda \parallel \mathbf{m} \parallel_0, \\ &s.t. \mathbf{W}\mathbf{W}^T = \mathbf{I}, \end{aligned} \tag{4}$$

where $\lambda$ denotes the damping factor, which is connected with the sparsity constrained coefficient $t$. Equation 4 is called *synthesis* model for DSD.

Assuming the base dictionary $\mathbf{B}$ is invertible, the synthesis-based DSD model as shown in equation 3 is equivalent to:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{m}} \frac{1}{2} \parallel \mathbf{B}^{-1}\mathbf{d} - \mathbf{W}\mathbf{m} \parallel_2^2 + \lambda \parallel \mathbf{m} \parallel_0 \\ &s.t. \mathbf{W}\mathbf{W}^T = \mathbf{I}, \end{aligned} \tag{5}$$

where $\mathbf{B}^{-1}$ denotes the forward base transform. Equation 5 suggests that DSD can also be learned in the model domain of the multi-scale decomposition operator $\mathbf{B}$ instead of in the data domain. Equation 5 is called *analysis* model for DSD.

The synthesis model (equation 4) offers more flexibility for learning DSD because there are many sparsity-promoting transforms that are not exactly invertible. The analysis model, on the other hand, offers more convenience for constructing DSD when an invertible sparsity-promoting base transform is available.

## Solving the analysis model by data-driven approach

In order to perform optimization in equations 4 and 5, we can employ the data-driven approach that was suggested previously by Cai et al. (2013). As an example, we only introduce the solver for equation 5.

The minimization can be solved by updating coefficients of vector $\mathbf{m}$ and the learning-based dictionary $\mathbf{W}$ alternately. We adopt the following algorithm for solving problem 5:
Input: Base dictionary $\mathbf{B}$, initial learning-based dictionary $\mathbf{W}^{(0)}$.

1. Transform data from data domain to model domain according to
$$\mathbf{b} = \mathbf{B}^{-1}\mathbf{d} \,. \tag{6}$$
2. **for** $k = 0, 1, \cdots, K-1$:
   (a) Fix the learning-based dictionary $\mathbf{W}^{(k)}$, estimate the double-sparsity coefficients vector $\mathbf{m}^{(k)}$ by
$$\mathbf{m}^{(k)} = \arg\min_{\mathbf{m}} \frac{1}{2} \parallel \left(\mathbf{W}^{(k)}\right)^T \mathbf{b} - \mathbf{m} \parallel_2^2 + \lambda \parallel \mathbf{m} \parallel_0 \,. \tag{7}$$
   (b) Given the double-sparsity coefficients vector $\mathbf{m}^{(k)}$, update the learning-based dictionary $\mathbf{W}^{(k+1)}$:
$$\begin{aligned} \mathbf{W}^{(k+1)} &= \arg\min_{\mathbf{W}} \frac{1}{2} \parallel \mathbf{W}^T\mathbf{b} - \mathbf{m}^{(k)} \parallel_2^2, \\ &s.t. \mathbf{W}\mathbf{W}^T = \mathbf{I}. \end{aligned} \tag{8}$$

   **end for**

After $K$ iterations, the DSD coefficients are obtained and the observed data become sparsely represented by DSD. It is known that minimization 7 has a unique solution $\hat{\mathbf{m}}$ provided by applying a hard thresholding operator on the coefficient vector $\left(\mathbf{W}^{(k)}\right)^T \mathbf{b}$. The minimization 8 can be implemented by using a SVD-based optimization approach (Cai et al., 2013).

Assuming the size of the coefficients domain is $N_1 \times N_2$, and $N = N_1 N_2$, let a filter mapping $\mathbf{F_a} : \mathcal{R}^N \rightarrow \mathcal{R}^N$ be the block-wise Toeplitz matrix representing the convolution operator with a finitely supported 2D filter $\mathbf{a}$ under the Newmann boundary condition. The learning based dictionary $\mathbf{W} \in \mathcal{R}^{N\times Np}$ can be defined as
$$\mathbf{W} = \left[\mathbf{F}_{\mathbf{a}_1}, \mathbf{F}_{\mathbf{a}_2}, \cdots, \mathbf{F}_{\mathbf{a}_p}\right] . \tag{9}$$
Each $\mathbf{a}_i$ is a 2D filter associated with a tight frame and the columns of $\mathbf{W}$ form a tight frame for $\mathcal{R}^N$. $p$ denotes the number of filters. The patch size discussed in the following examples corresponds to the size of each $\mathbf{a}_i$.

Liang et al. (2014) give an example of using spline wavelets for the initial $\left(\mathbf{W}^{(0)}\right)^T$ and the finally learned dictionary $\left(\mathbf{W}^{(K)}\right)^T$. Following Liang et al. (2014), we also choose spline wavelets for the initial $\mathbf{W}$.

## Seislet transform based DSD

The analysis model can be used only if the base transform is invertible and can be more convenient to straightforwardly cascade two types of transforms: the base transform and the learning-based dictionary. In this paper, we use the seislet transform

(Fomel and Liu, 2010) as the base transform and DDTF (Cai et al., 2013) as the learning-based dictionary to construct the proposed DSD framework, following the aforementioned algorithm. The DDTF refers exactly to the process of solving equations 7 and 8. In the seislet based DSD, the seislet dictionary $\mathbf{B}$ is substituted by the combined dictionary $\mathbf{BW}$. The effective atoms we obtain are thus interpolated version of the atoms in the learning-based dictionary DDTF, interpolated by the seislet transform process. The seislet based DSD thus enjoys the structure-oriented multi-scale capabilities of the seislet transform while adding information to it specific to the training domain. It should also be mentioned that because of the initial seislet decomposition, each sub-band in the transform domain can be treated separately and thus can has its own sub-dictionary (Ophir et al., 2011).

We train the filters patch by patch of the whole dataset. We apply the maximally overlapping patches (i.e., the two neighbor patches only shifted by one column) for the training. This creates a richness in the training data and leads to a shift-invariance for the dictionary. In the DDTF, for each patch, the training filer is orthogonal, but the full transform matrix $\mathbf{W}$ (each column is a training filter on patches) is redundant. Because of the base seislet transform, the DDTF trains the dictionary from a multi-scale seislet domain instead of the single scale domain of the original data. Although we follow the same strategy as Ophir et al. (2011), we utilize different tools for the two main components in the multi-scale learning framework. The DDTF used in the proposed method is much faster than the K-SVD method used by Ophir et al. (2011), and the seislet transform used in the proposed method is more appropriate for seismic data than the wavelets used by Ophir et al. (2011). In K-SVD, there exists a large number of patches for training and the components of the dictionary are updating individually (indicating many SVD decompositions). However, the DDTF method updates all the components (columns of $\mathbf{W}$) by one SVD decomposition (Yu et al., 2015), thus is more efficient. Besides, because of the learning efficiency in DDTF, the DDTF based approaches can be applicable to high-dimensional denoising and interpolation problems. The seislet transform, however, is superior than traditional wavelet transform for compressing seismic data because it utilizes local slope to construct the transform. The superior performance of seislet transform over state-of-the-art transforms in denoising, interpolation, and deblending of seismic data have been demonstrated in a number of references (Fomel and Liu, 2010; Liu and Fomel, 2010; Chen et al., 2014; Gan et al., 2015). Thus, the combination between the seislet transform and DDTF through the DSD framework seems very appealing.

The cascaded DSD framework, combined with thresholding-based denoising, can be briefly summarized as follows:

1. Apply the seislet transform to noisy seismic data.
2. Apply the DDTF to each band in the seislet domain to form DSD domain.
3. Perform thresholding in the DSD domain.
4. Apply the adjoint DDTF to the thresholded data in DSD domain.

5. Apply the inverse seislet transform to recover the denoised data.

There is an inner iterative process in the above steps when applying the DDTF. It refers to applying the DDTF by the data-driven learning process following equations 6 to 8. The iterative process acts as a training step for the DDTF, and the trained dictionary **W** are finally used to transform the seislet subband data into the double-sparsity domain and used for the adjoint DDTF step in step 4. Figure 1 shows a demonstration of denoising processes of the proposed DSD. It also shows the seislet transform domain and DSD transform domain before and after thresholding.

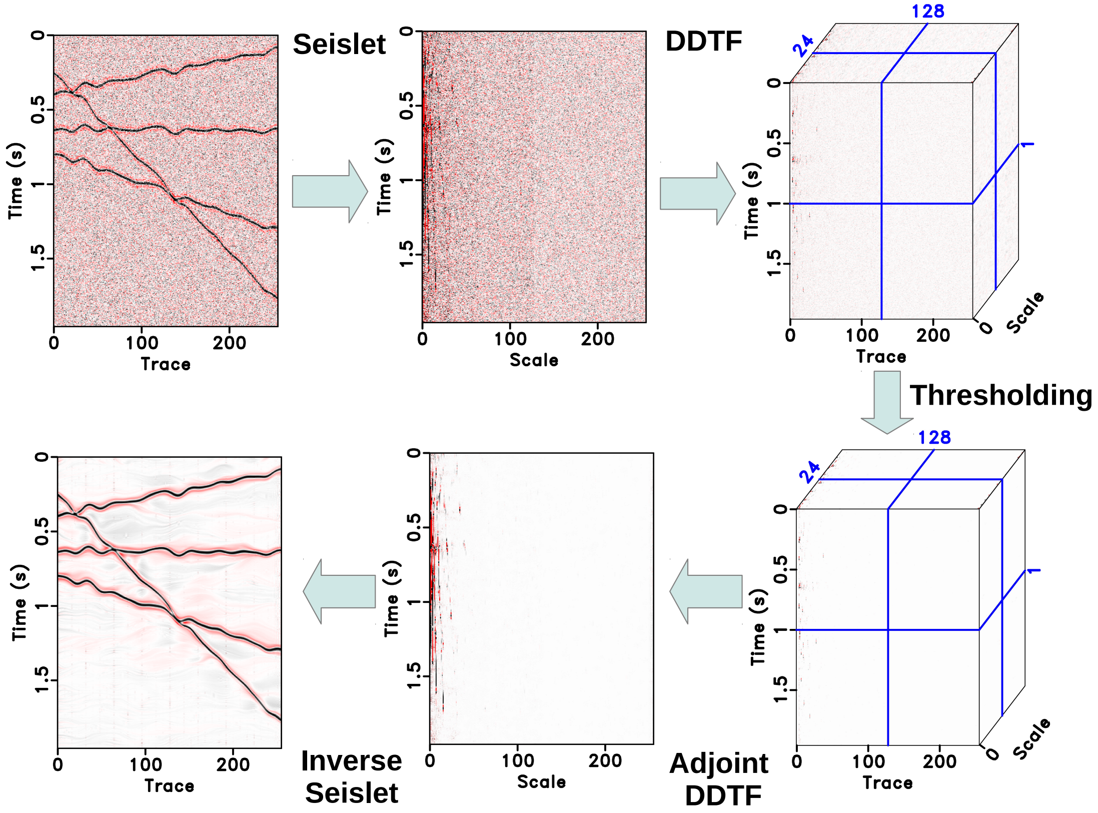


Figure 1: Demonstration of the DSD-based denoising processes.

# EXAMPLES

In this section, we use two synthetic examples and two field data examples to test the denoising performance of the proposed DSD based thresholding. In order to numerically test the performance, and to quantify the comparison for all the examples, we simulate noisy data by adding Gaussian white noise.

We apply the following simple model for synthesizing noisy data with Gaussian white noise:

$$\mathbf{u} = \mathbf{d} + \epsilon(\sigma), \tag{10}$$

where $\mathbf{u}$ denotes the simulated noisy seismic data, $\mathbf{d}$ denotes the clean data, and $\epsilon(\sigma)$ denotes a Gaussian white noise with zero mean and standard deviation $\sigma$. The clean data is recovered by applying a simple thresholding operator in the DSD domain.

In order to numerically test the denoising performance, we define the criterion for comparison as signal-to-noise ratio (SNR):

$$SNR = 10\log_{10}\frac{\|\mathbf{d}\|_2^2}{\|\mathbf{d}-\hat{\mathbf{d}}\|_2^2}, \tag{11}$$

where $\hat{\mathbf{d}}$ is the denoised signal.

## Synthetic examples

The first synthetic example contains four linear events with different dips. The steepest event crosses the other three events. In order to test the denoising performance in the presence of non-linear events, we added some continuous random statics to the data by shifting each trace from a linear-events profile with different shifts. The clean seismic profile is shown in Figure 2a and the noisy profile with Gaussian white noise added is shown in Figure 2b.

In order to compare the sparsity in compressing seismic data, we plot the reconstruction error curves with respect to the number of selected largest coefficients in Figure 3. The reconstruction error is defined as $E = \| \mathbf{d} - \mathbf{A}\mathbf{m}_N \|_2^2$, where $\mathbf{m}_N$ denotes the largest $N$ coefficients in the transform domain. The faster the reconstruction errors decay, the sparser the corresponding transform domain is. Because both DDTF and DSD are redundant transforms, in order to compare the sparsity in terms of compression fairly, we select a different number of largest coefficients to reconstruct the seismic data and plot the corresponding calculated reconstruction errors. The comparison shows that the reconstruction error curve of the seislet transform decays fastest, followed by DSD and DDTF. For other examples in the paper, the comparison of sparsity in terms of compression behaved similarly. Thus, we observe that, in terms of compression performance, the seislet transform behaves best, the proposed DSD behaves the second best, while the traditional DDTF behaves worst. Whether the number of the largest coefficients or the percentage of the largest coefficients should be put on the horizontal label is important. Because the space of transform domain might not have the same dimensions for different transforms, we chose to put the number instead of percentage on the horizontal label. When comparing the sparsity in terms of compression, what we really care about is that the least number of coefficients can represent the most information.

In order to compare the sparsity in terms of denoising performance for three different transforms, we compare the best denoised results using both visual observation and measured SNR. We achieve the best denoising performance by modifying the threshold parameter and by comparing the SNR and general denoising performance, while fixing other parameters that might affect the construction of these transforms. Instead of using a fixed threshold value, we use a more flexible percentile thresholding strategy, which means that we apply the inverse transform by using a specific percentage of the transform domain coefficients sorted by amplitude. The threshold required by the soft- or hard-thresholding in the transform domain is chosen according to the percentage of coefficients we want to preserve. The optimum percentage of coefficients for different approaches might not be the same, therefore we need to set the optimum percentage for all the approaches by trying different percentage to obtain the highest signal-to-noise ratio in order to be fair for all the methods. This strategy was used previously by Chen et al. (2014).

Using different percentage coefficients, we can obtain different denoised results for different thresholding approaches. In this example, we use $7 \times 7$ patch size for the tight frames and 30 iterations for updating the tight frames and the coefficients. Table 1 shows SNRs using different percentage coefficients for three different thresholding approaches. As evident from Table 1, the seislet thresholding obtains its optimal performance when 5% coefficients are used. The DDTF thresholding obtains its optimal performance when 4% coefficients are used. However, the DSD thresholding only needs 2% coefficients to obtain its optimal performance. The best denoised results are shown in the first row of Figure 4. Their corresponding noise sections are shown in the second row of Figure 4.

Even for the best results using seislet and DDTF thresholdings, there is still a small amount of residual signal left in the noise sections. The recently proposed local orthogonalization of signal and noise (Chen and Fomel, 2015) may help address this problem. The local orthogonalization can be used in combination with any denoising approach to recover lost coherent signals in the removed noise section. The seislet thresholding algorithm sometimes suffers from this over-smoothing problem. The slight artifacts as shown in Figure 4a are caused by the over-smoothing issue. Figure 4c shows much better performance in terms of removing the artifacts caused in the seislet transform. Thus, another conclusion of this example is that the DSD can also help improve the result of the traditional seislet transform. Figures 4g, 4h, and 4i show the error sections (difference between the denoised data and the true clean data) of the first example using different approaches. We observe that the DSD thresholding can obtain the least denoising error, which further confirms that the DSD thresholding can preserve more useful energy. For simplicity, we only use a 1% increment when scanning the threshold percentage parameters.

The first example demonstrates that the sparsest transform in terms of compression may not be the most efficient for removing random noise. The DSD can achieve a better performance for distinguishing signal and noise. Choosing controlling parameters in order to estimate the best denoised results requires some trial and error in practice. Therefore, from the second synthetic example to the last field data example, we only show the best results for each approach, and calculate the numerical SNRs as a quantitative reference to evaluate the performance.

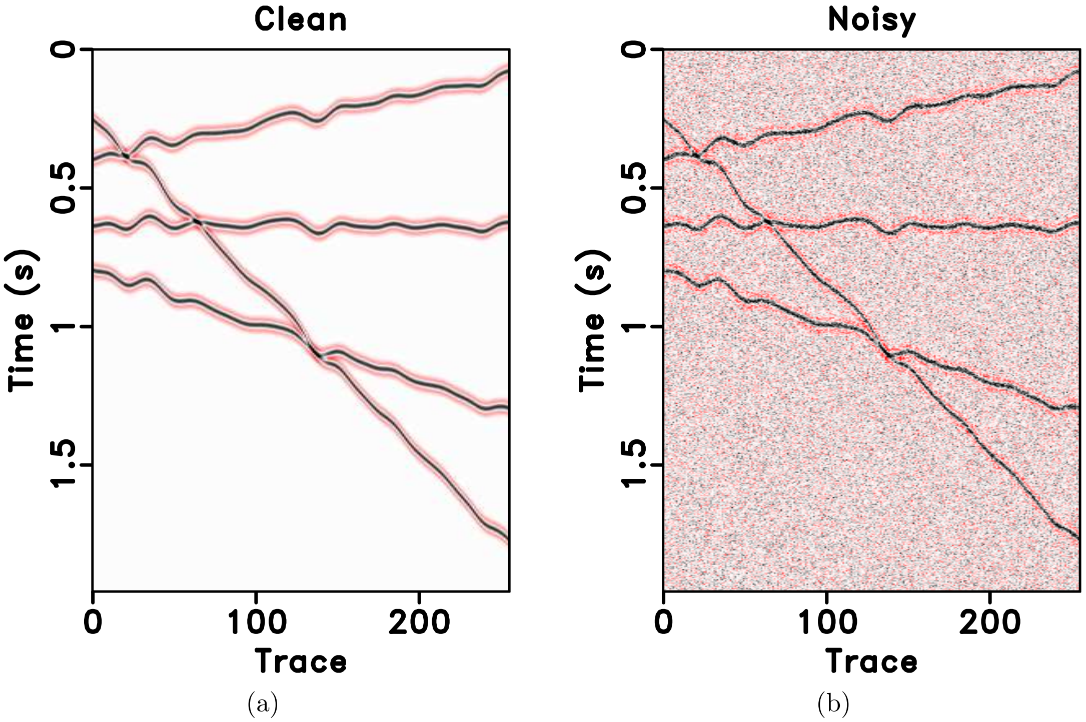


Figure 2: (a) Clean data. (b) Noisy data.

Figure 5 shows a comparison of basis functions of different transforms by randomly selecting a number of significant coefficients in the transform domain, and then transforming the data back to the time-space domain. As can be seen from the comparison, both seislet and DSD have well constructed basis functions following the slope of seismic events. DDTF, however, does not show any structural pattern from the basis functions. DDTF and DSD have better adaptivity than seislet because the seislet basis shows a stronger dependency on the initial slope estimation. In can be observed that the DSD basis (Figure 5c) only shows coherent energy around the linear-events regions, while the seislet basis (Figure 5a) shows coherent energy nearly everywhere because most regions have zero local slope and such zero local slope will cause coherent basis because of a stronger dip dependency of seislet over DSD. The apparent randomness of the basis of DSD lays between that of seislet and DDTF, which indicates a better adaptivity of DSD to the input dataset. When observing Figure 5, we should focus on both the randomness and the coherency of the basis, because randomness and coherency represent the slope dependency and data adaptivity, respectively. The DSD obtains a compromise between the two features. Figure 6 shows a comparison of reconstruction performance by selecting 150 maximum coefficients in different transform domains. It is clear that the same number of coefficients in the seislet transform can recover the most amount of useful reflection energy, which is followed by DSD and DDTF, respectively. Figure 6 is also consistent with the result shown in Figure 3 in that though the DSD can obtain the best performance in separating useful signals and random noise, it is less effective than

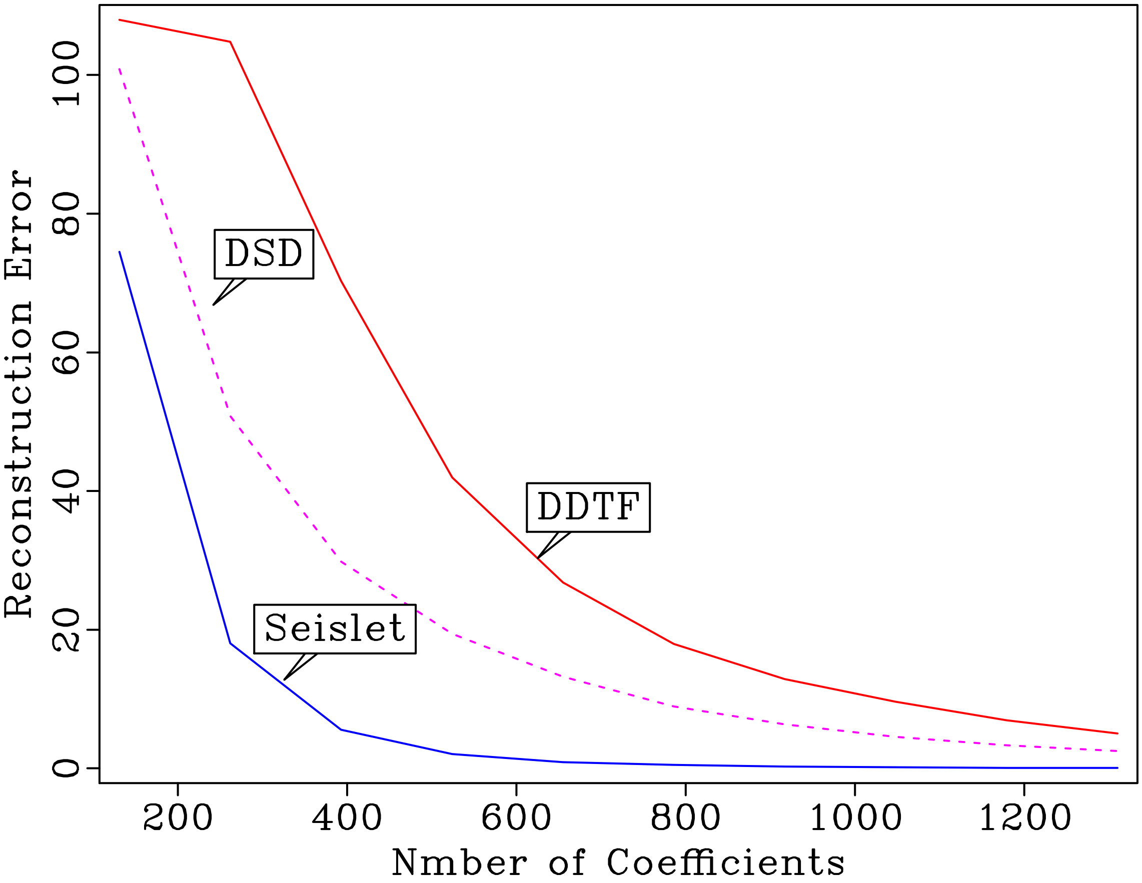
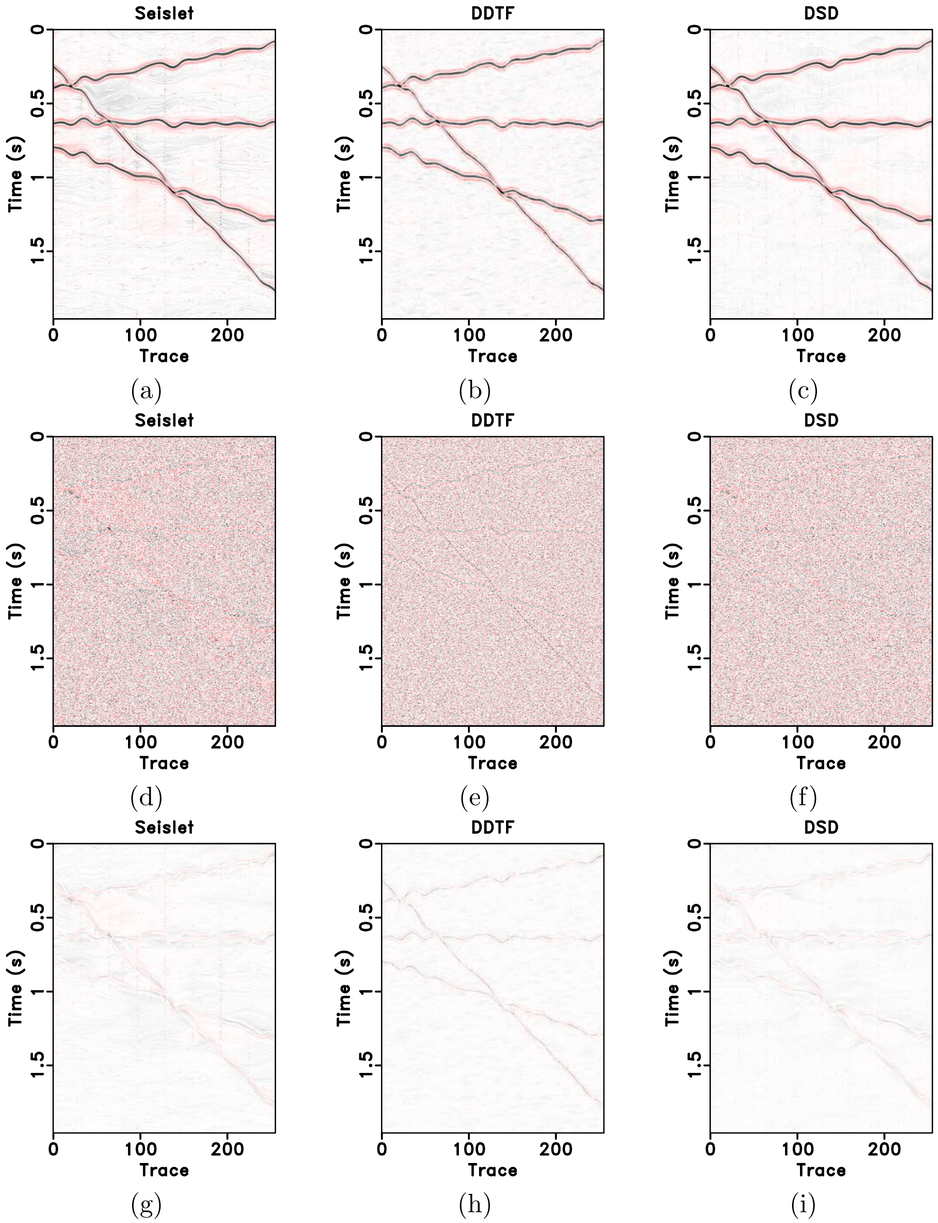


Figure 3: Sparsity comparison in terms of compression.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

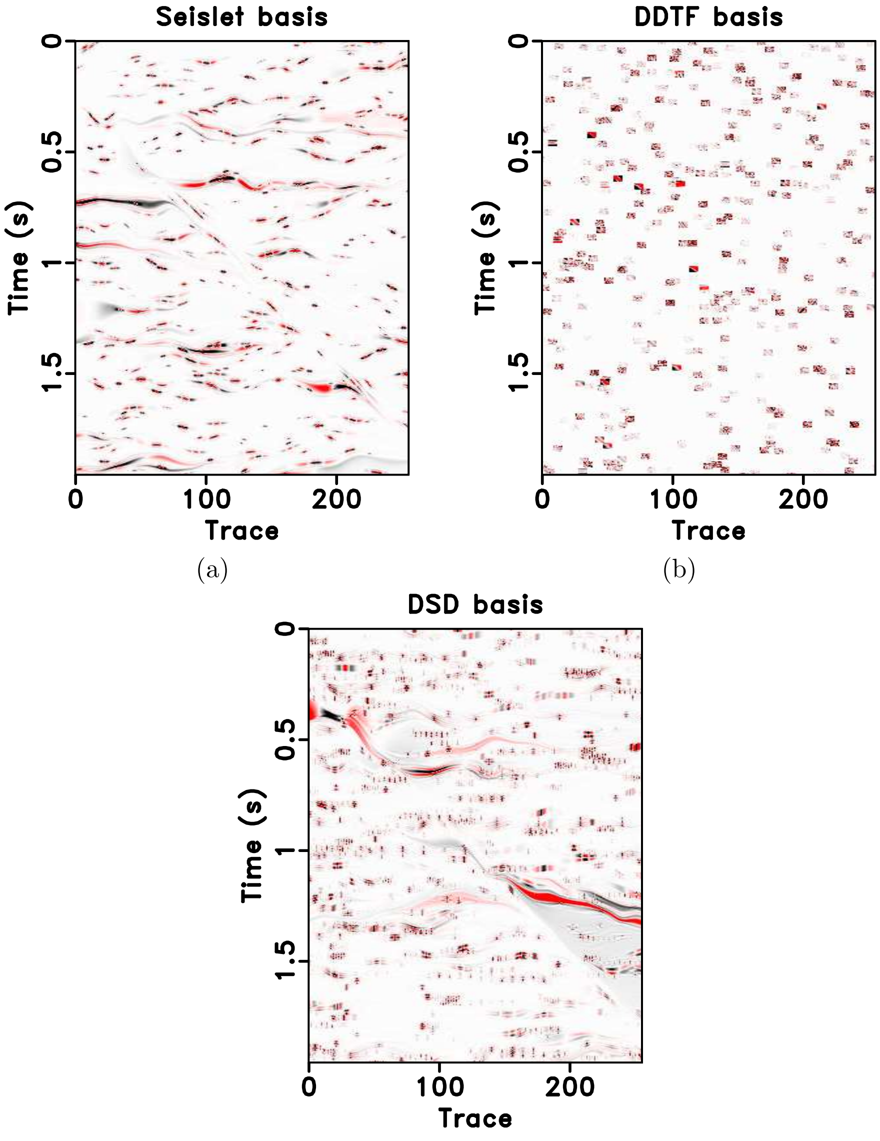
Figure 4: First row: denoised sections using different approaches. (a) Seislet thresholding. (b) DDTF thresholding. (c) DSD thresholding. Second row: noise sections using different approaches. (d) Seislet thresholding. (e) DDTF thresholding. (f) DSD thresholding. Third row: error sections using different approaches. (g) Seislet thresholding. (h) DDTF thresholding. (i) DSD thresholding. The best results are selected for each method.

the seislet transform for compressing the seismic data. Figure 7 shows a comparison of reconstruction performance by selecting 0.1% of the most significant coefficients in different transform domains. Because both DDTF and DSD are redundant, the actual numbers of coefficients in different transform domains are different. For example, in this case, the number of selected coefficients in the seislet domain is 131, but the numbers of selected coefficients in the DDTF and DSD domains are both 6422. It can be seen that in this case the reconstruction using DSD recovers the most information, which is also consistent with Table 1. Figures 6 and 7 also indicate a phenomenon that the largest coefficients in the seislet and DSD domains mainly reconstruct the low-frequency part of the input seismic data, which can be explained by the fact that the low-frequency components are easier for seislet-based transforms to compress.

In the second example, we use a Sigmoid synthetic model (Claerbout, 2010). The clean and noisy data are shown in Figures 8a and 8b, respectively. After using soft-thresholding in the seislet domain, DDTF domain, and DSD domain, and selecting the best parameters for each case, we obtain three denoised results, shown in Figure 9. In this example, we also use $7 \times 7$ patch size for the tight frames and use 30 iterations for updating the tight frames and the DSD coefficients. The three corresponding noise sections are shown in Figure 10. The comparison of SNRs for this example is shown in the first row in Table 2. Thanks to selecting optimal parameters for each method, these denoised results are all acceptable considering both the preservation of useful signals and removal of random noise. However, we observe that the DSD-based thresholding can get better results in that it leaves practically no coherent useful signal in the noise section (Figure 10c) and its calculated SNR is the highest among all the three approaches. In this example, we used 20 % coefficients for seislet transform, 16 % coefficients for DDTF, and 10 % coefficients for DSD. Figures 8, 9, and 10 have the same scaling in order to make the comparison between different methods fair. What is interesting about this example is that Figures 10a and 10c show the fault (as expected), but Figure 10c shows all reflectors nearly equally. This observation is reasonable because DDTF thresholding does not follow the structure of seismic events, thus the denoising process causes equal damage to all events. On the other hand, both seislet thresholding and DSD thresholding include the structure information when constructing the transform, and, therefore, cause smaller damage for events with smoothly changing slope. The damage to the fault regions using traditional denoising approaches is a common problem in seismic data processing. Similar results are also shown by Liu and Fomel (2010). As we can see from this comparison, the denoised result using DSD thresholding can improve the preservation of structural edge information.

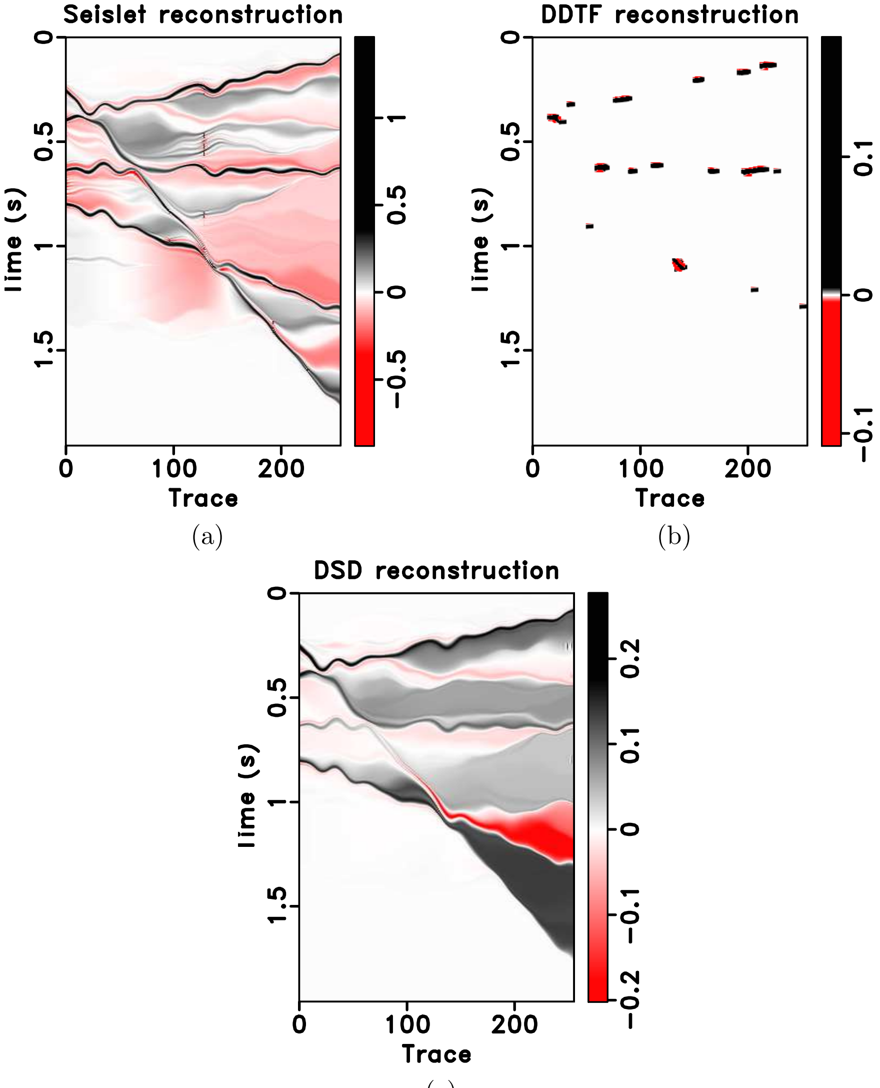
## Field data examples

For field data tests, we first use a relatively simple 2-D seismic image, shown in Figure 11a. The simulated noisy data with Gaussian white noise is shown in Figure 11b. The denoised results from the three different approaches specified previously are shown



(a) (b) (c)

Figure 5: Comparison of basis functions by randomly selecting the same number of coefficients in the transform domain. (a) Basis functions of the seislet transform. (b) Basis functions of the DDTF. (c) Basis functions of the DSD.

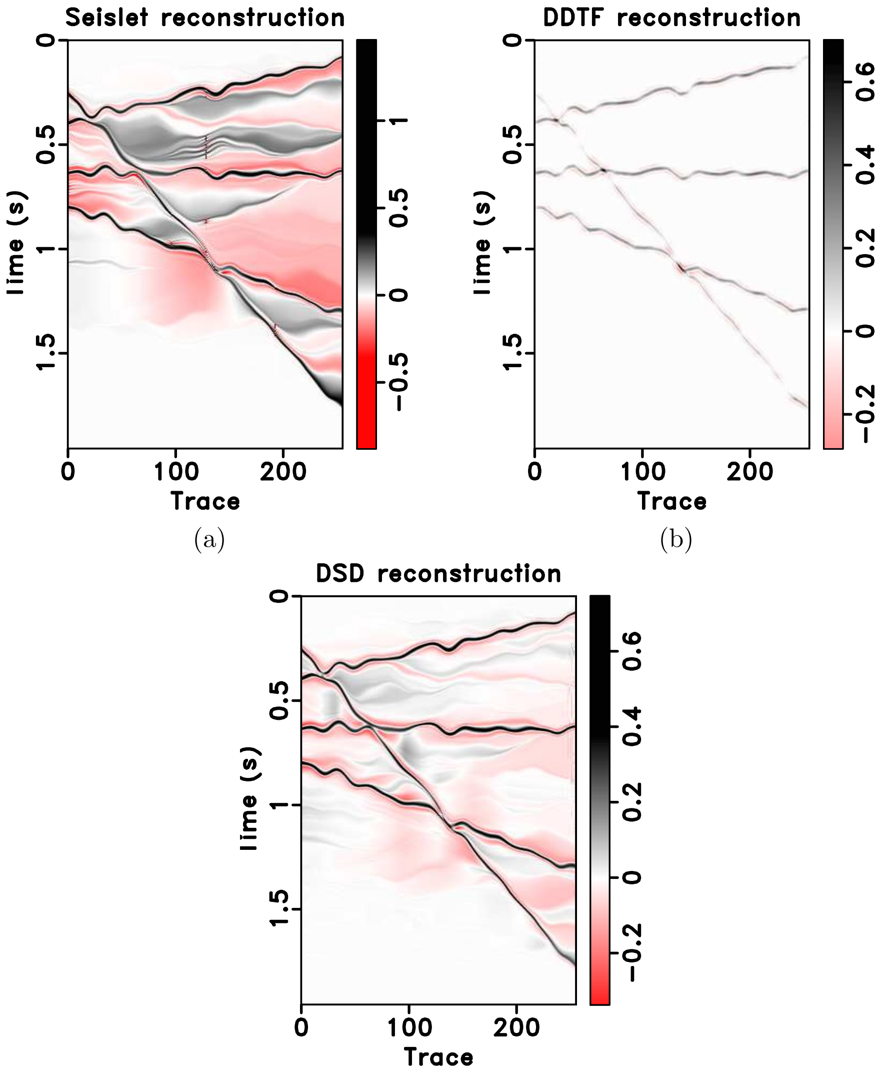


(a)

(b)

(c)

Figure 6: Comparison of reconstruction performance by selecting 150 maximum coefficients in the transform domain. (a) Reconstruction using the seislet transform. (b) Reconstruction using the DDTF. (c) Reconstruction using the DSD.

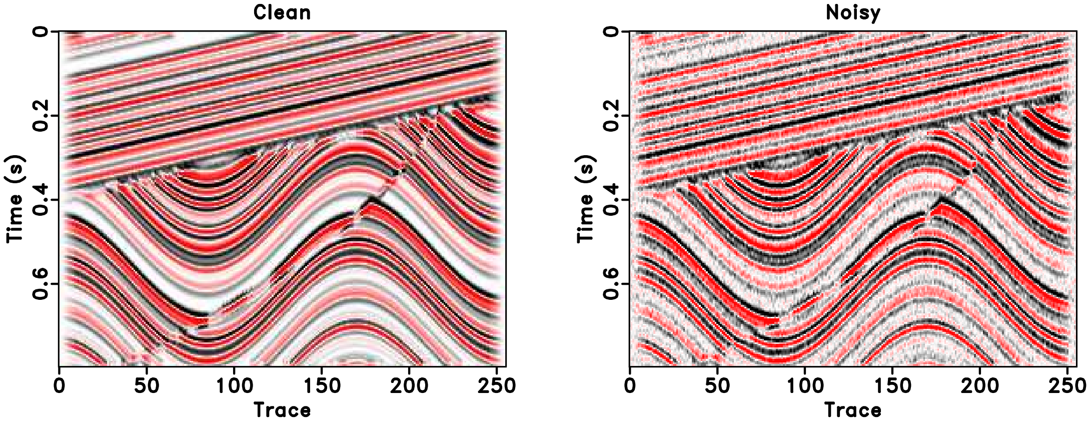


(a)

(b)

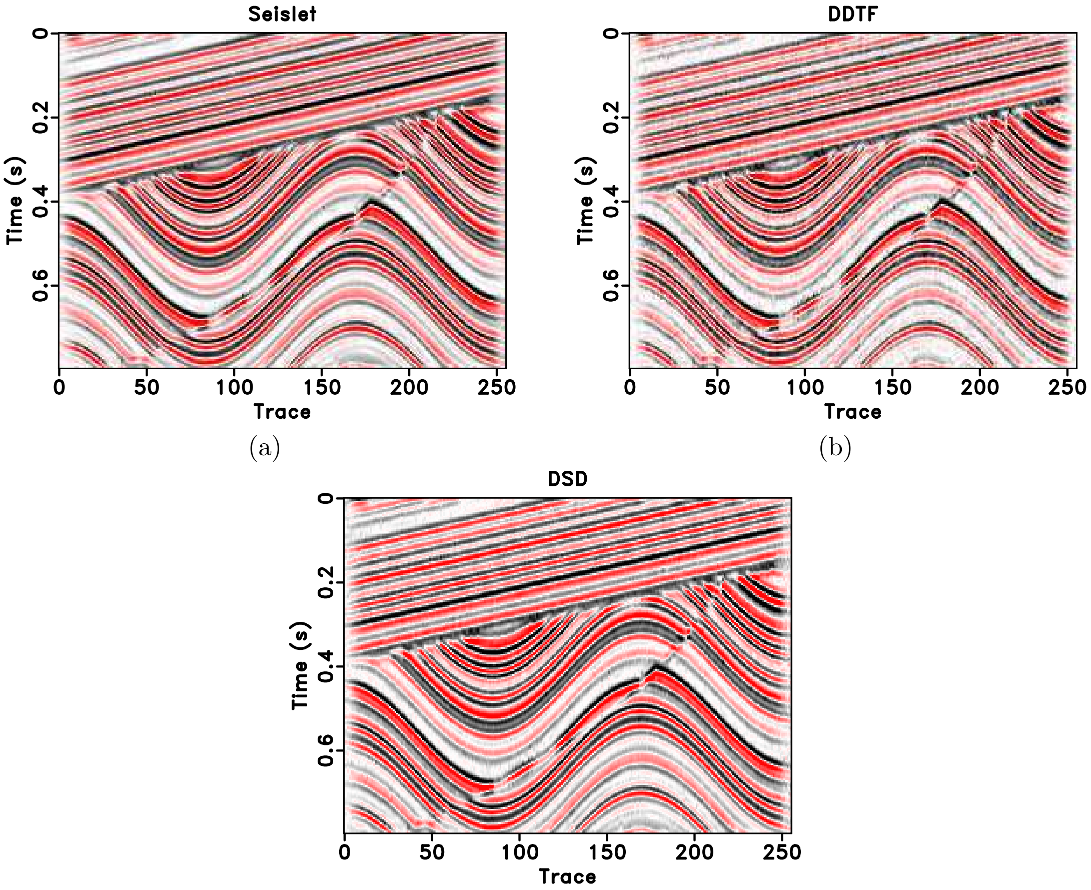
(c)

Figure 7: Comparison of reconstruction performance by selecting 0.1% largest coefficients in the transform domain. Note that the number of selected coefficients in different transform domains are different because of the redundant nature of both DDTF and DSD. There are 131 coefficients selected in the seislet domain. There are 6422 coefficients selected in both DDTF and DSD domains. (a) Reconstruction using the seislet transform. (b) Reconstruction using the DDTF. (c) Reconstruction using the DSD.



(a) (b)

Figure 8: (a) Clean data. (b) Noisy data.



(a) (b)

(c)

Figure 9: Denoised sections using different approaches. (a) Seislet thresholding. (b) DDTF thresholding. (c) DSD thresholding. The best results are selected for each method.

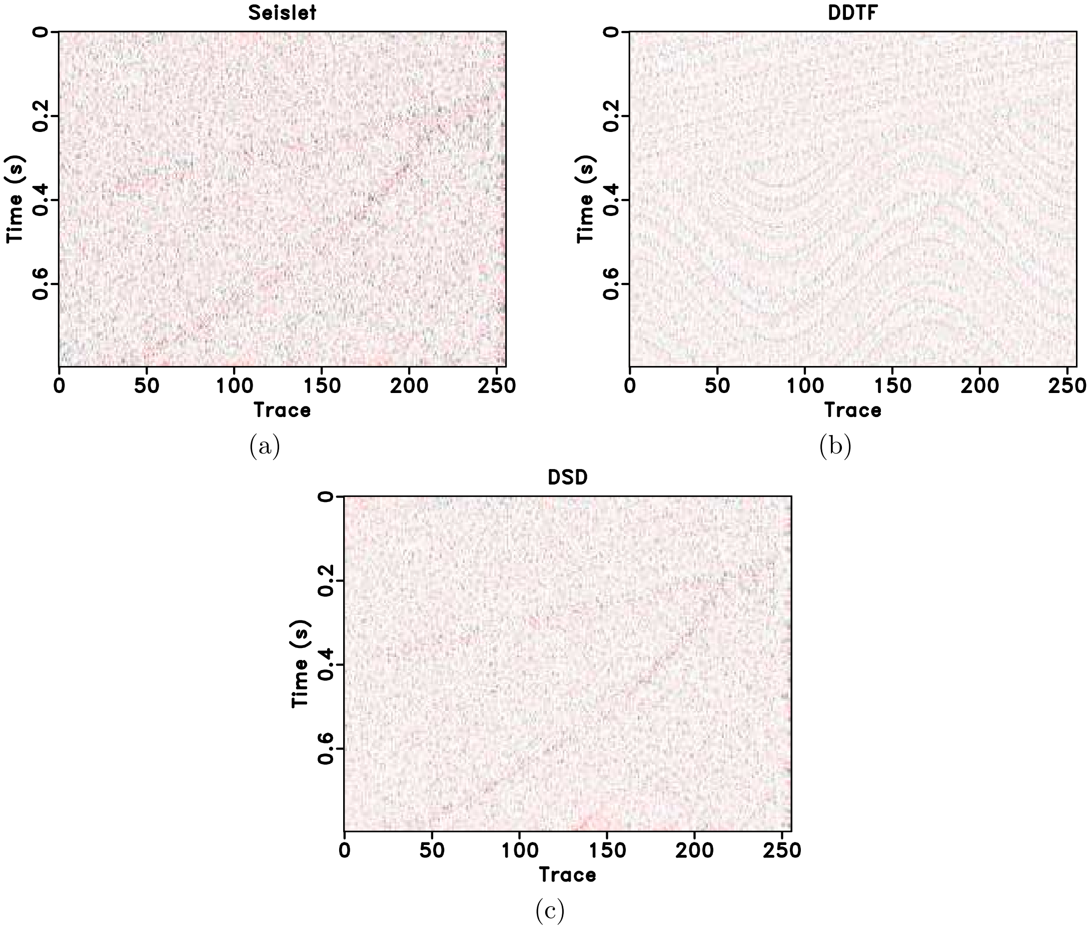


Figure 10: Noise sections using different approaches. (a) Seislet thresholding. (b) DDTF thresholding. (c) DSD thresholding. The best results are selected for each method.

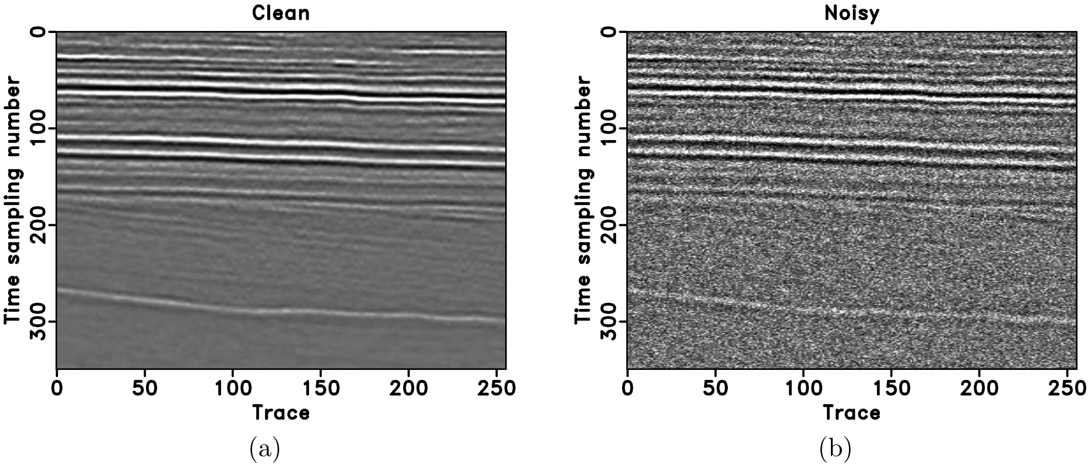
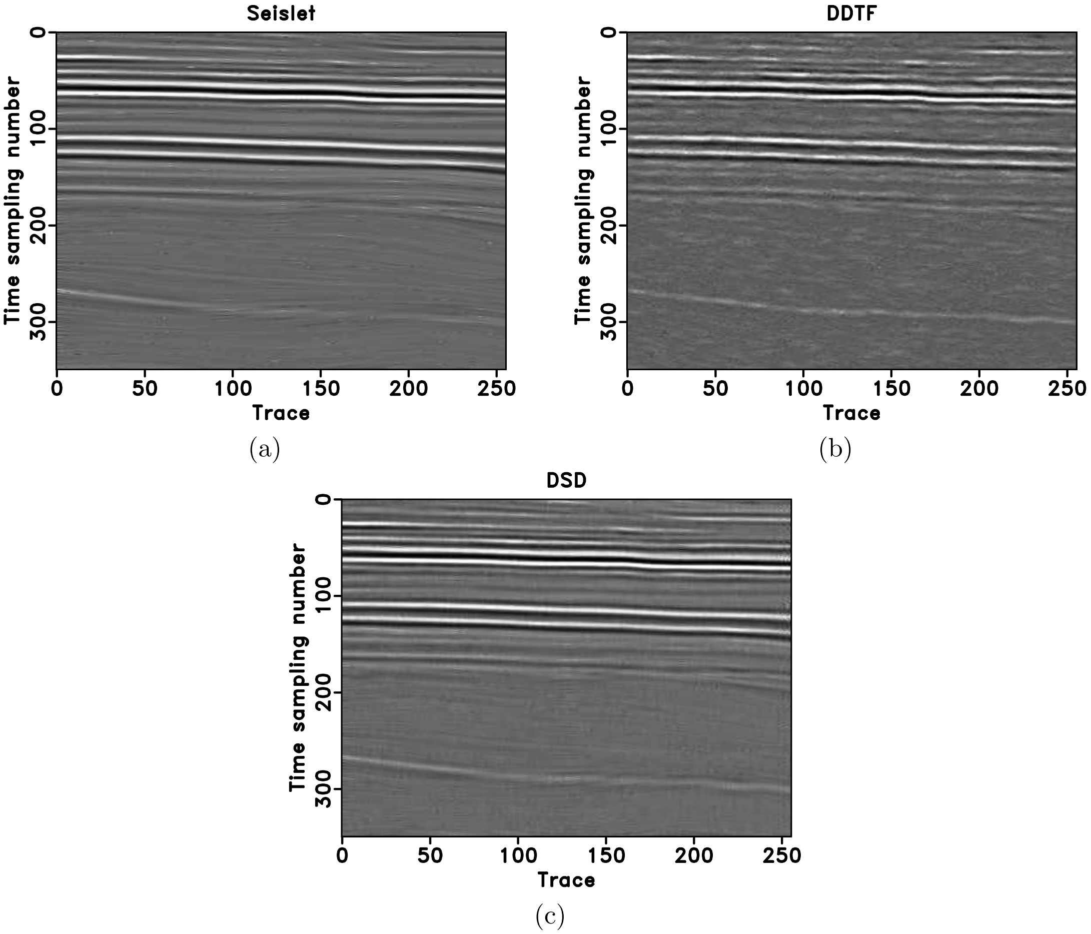


Figure 11: (a) Clean data. (b) Noisy data.



(a)

(b)

(c)

Figure 12: Denoised sections using different approaches. (a) Seislet thresholding. (b) DDTF thresholding. (c) DSD thresholding. The best results are selected for each method.

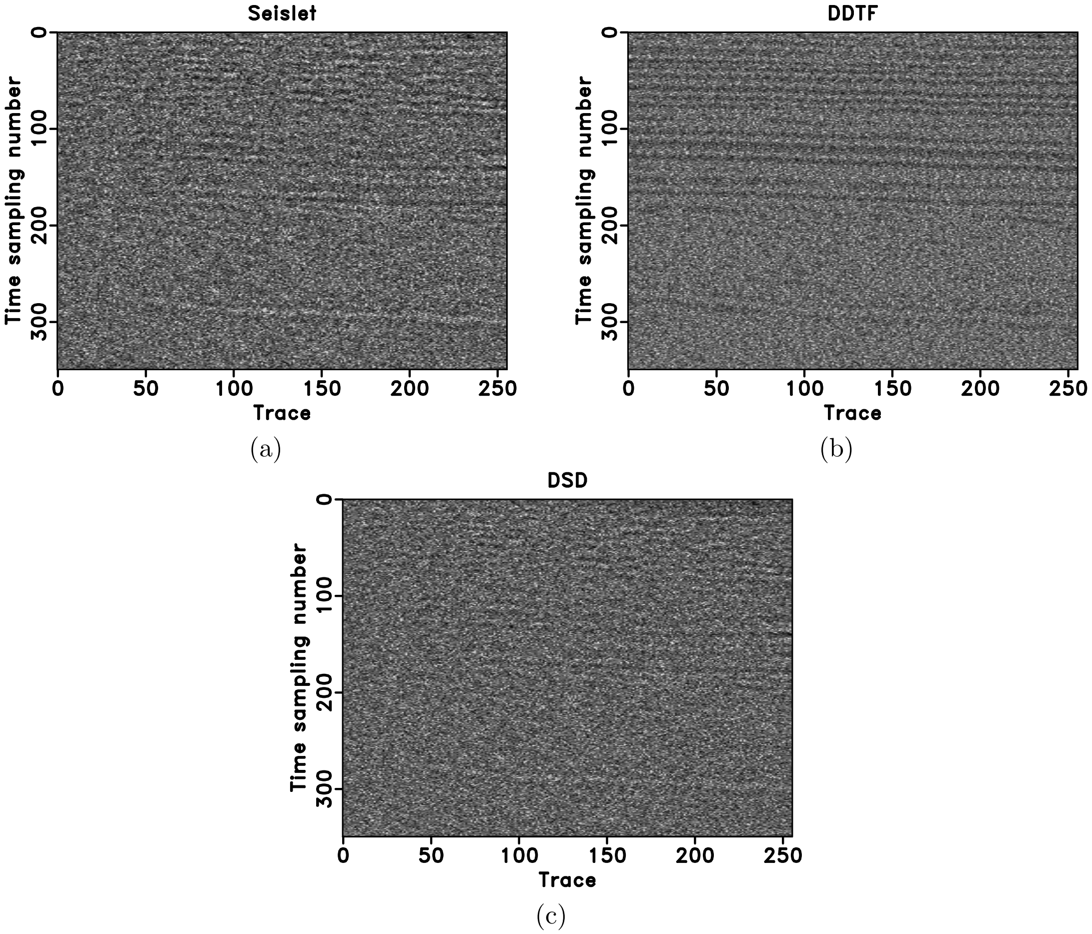
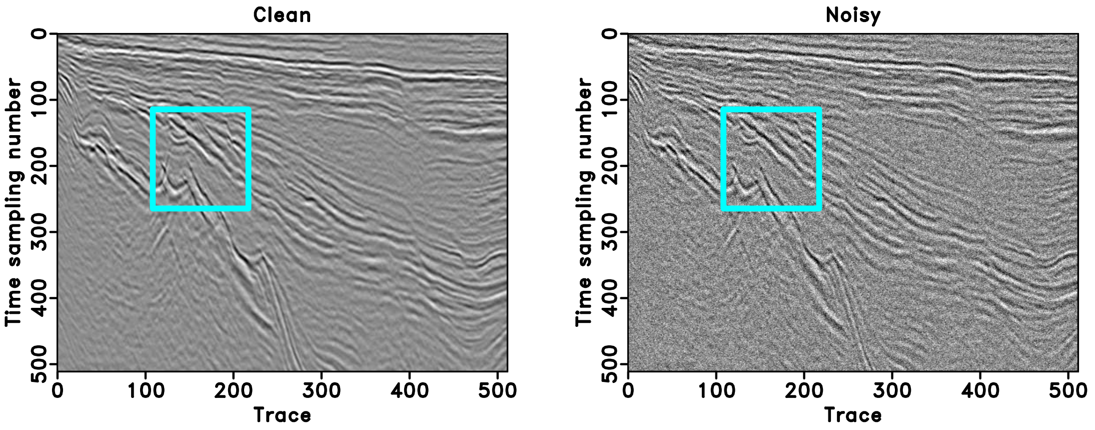


Figure 13: Noise sections using different approaches. (a) Seislet thresholding. (b) DDTF thresholding. (c) DSD thresholding. The best results are selected for each method.

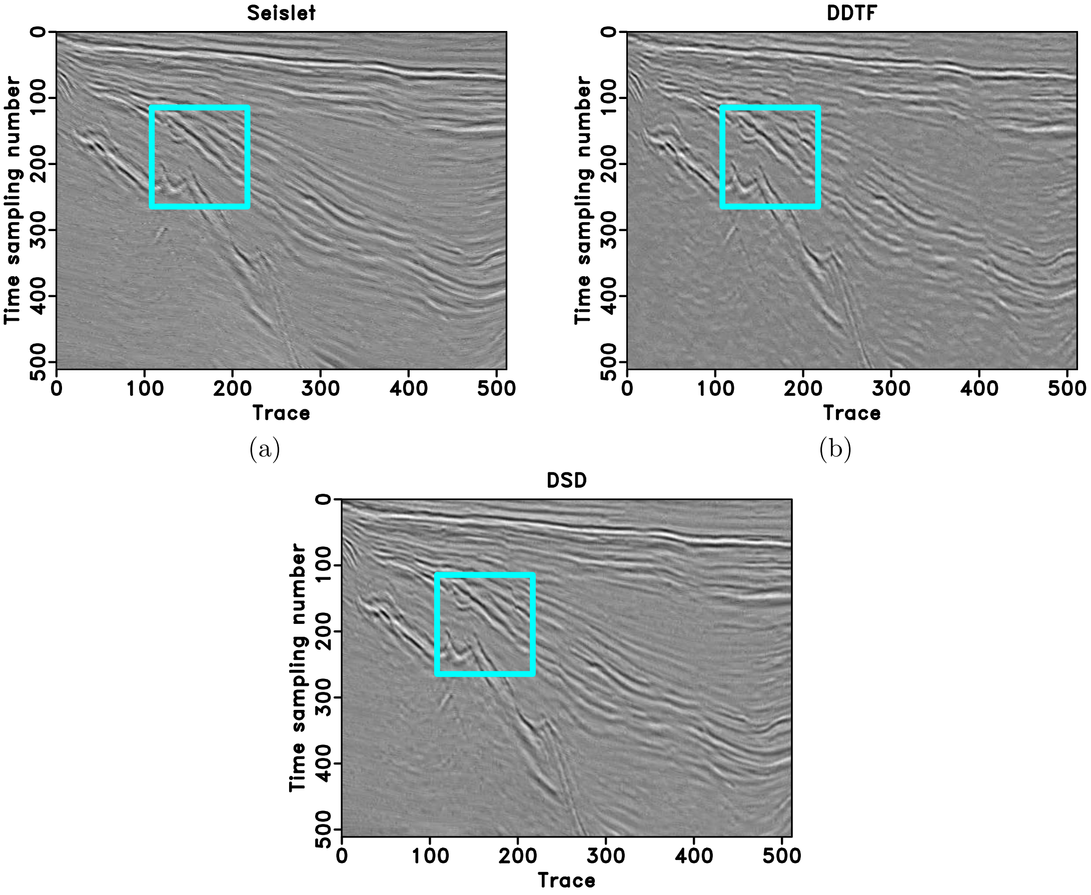
in Figure 12. In this example, we use 7 × 7 patch size for the tight frames and run 30 iterations for updating the tight frames and the coefficients. The corresponding three noise sections are shown in Figure 13. From the comparison of denoised results, we conclude that, while all of the three methods can obtain acceptable results, the proposed DSD-based thresholding produces the cleanest image (Figure 12c) with the least amount of useful energy left in the noise section (Figure 13c). The seislet transform based thresholding causes some damage to signal events around time sample 300. The DDTF based thresholding leaves some residual random noise, which comes from the fact that DDTF might automatically learn the behavior of random noise and thus become sensitive to random noise. In this example, we carefully selected parameters for each method in order to obtain the best performance. Specifically, we use 4 % coefficients in the seislet domain, 8 % coefficients in the DDTF domain, and 3 % coefficients in the DSD domain. The comparison of SNRs is shown in the third row in Table 2. The SNR comparison confirms the previous observation that DSD-based thresholding can obtain the highest SNR.



(a) (b)

Figure 14: (a) Clean data. (b) Noisy data. The frame boxes are zoomed for detailed comparisons in Figure 17.

Our second field data example is a more complicated case. Figures 14a and 14b show the clean data and simulated noisy data, respectively. In this example, we use 7 × 7 patch size for the tight frames and run 30 iterations for updating the tight frames and the DSD coefficients. Because of the complex signal structure, seislet transform fails to sparsify the data well. On the other hand, because of strong noise, some noise gets treated as useful signal and gets incorporated into the dictionary when constructing DDTF. Therefore, DDTF also fails to obtain good denoising result for this dataset. The proposed DSD compensates for the weaknesses of both the seislet transform and DDTF and thus achieves a better result. Figure 15 shows the optimal denoised results after using thresholding in the seislet domain with 17 % coefficients, thresholding in the DDTF domain with 11 % coefficients, thresholding in the DSD domain with 7 % coefficients. The corresponding noise sections are shown in Figure 16. From observing the denoised results and noise sections, we conclude that

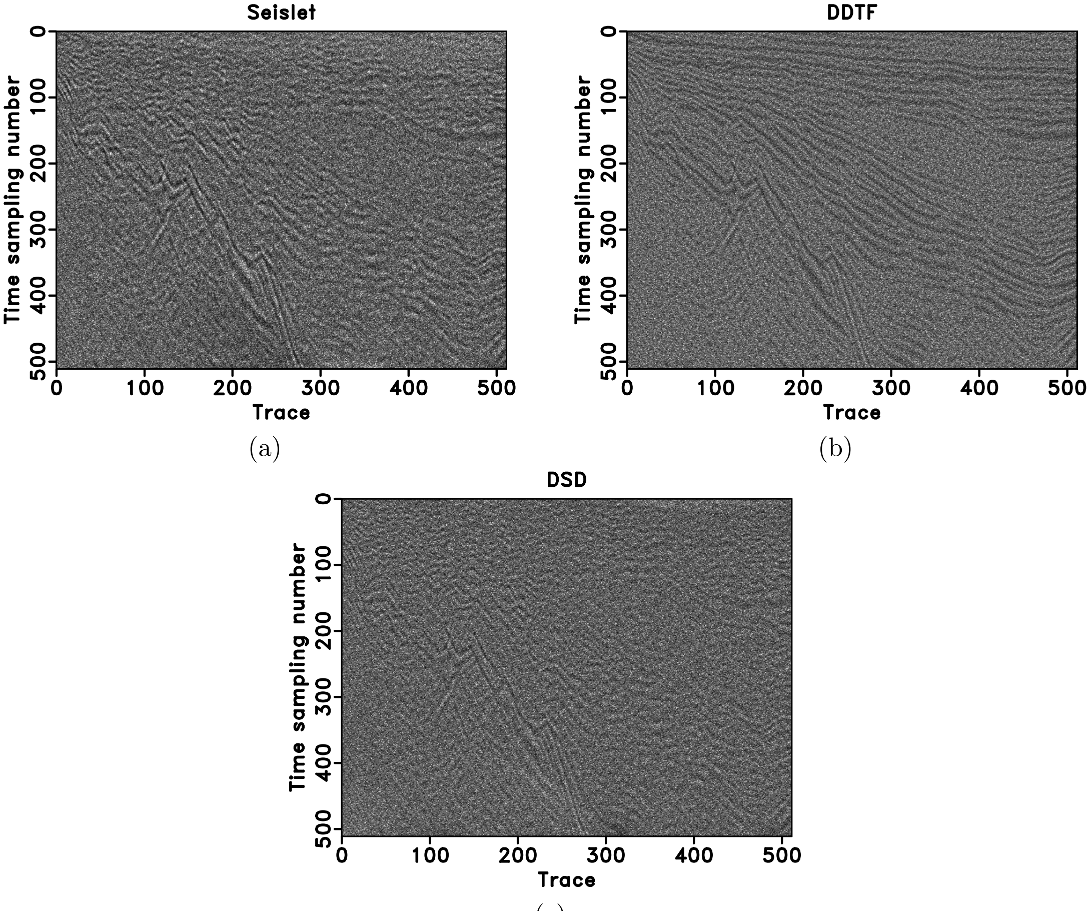


(a)

(b)

(c)

Figure 15: Denoised sections using different approaches. (a) Seislet thresholding. (b) DDTF thresholding. (c) DSD thresholding. The best results are selected for each method. The frame boxes are zoomed for detailed comparisons in Figure 17.



(a)

(b)

(c)

Figure 16: Noise sections using different approaches. (a) Seislet thresholding. (b) DDTF thresholding. (c) DSD thresholding. The best results are selected for each method.

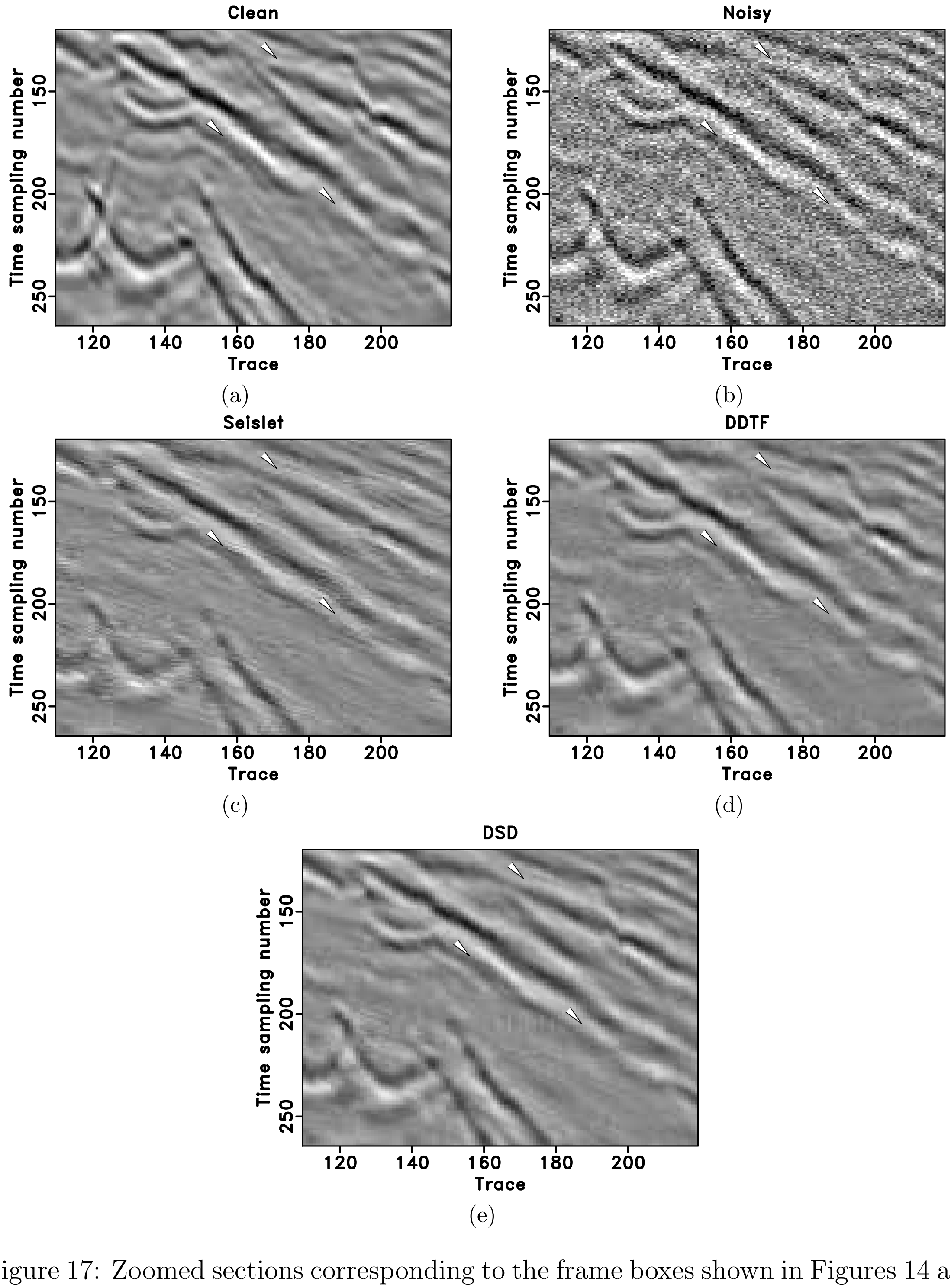


Figure 17: Zoomed sections corresponding to the frame boxes shown in Figures 14 and 15. (a) Clean data. (b) Noisy data. (c) Seislet thresholding. (d) DDTF thresholding. (e) DSD thresholding.

thresholding in the DSD domain obtained the cleanest image and with the smallest amount of useful coherent signals lost in the noise sections. The calculated SNRs are shown in the Field 2 row in Table 2. The SNR comparison confirms the observation that thresholding in DSD domain can get the highest SNR. In this example, we also tried different parameters in order to obtain the best result for each approach. The denoising performance using DSD is noticeably better than the other two approaches, in terms of both SNR measures and visual observation.

To see the differences more clearly, we zoomed parts of the denoised sections and show them in Figure 17. The zoomed sections correspond to the portions pointed out by the frame boxes in Figures 14 and 15. We can observe more clearly a better performance of the proposed DSD thresholding in comparison with the other two approaches in that it gets a cleaner image and preserves more of the useful signal.

Our third field data example is a noisy seismic dataset, shown in Figure 18. The denoised sections using different denoising approaches are shown in Figure 19. In addition to the three thresholding-based approaches, we also demonstrate the denoising performance using the classic $f-x$ deconvolution (Gulunay, 1986; Galbraith, 1991; Chen and Ma, 2014). For $f-x$ deconvolution, we use $50 \times 50$ windows with 50% overlap between different local windows. The prediction length is 6 points. These parameters have been tuned for obtaining the best denoising performance. In this example, we use $7 \times 7$ patch size for the tight frames and 30 iterations for updating the tight frames and the coefficients in DDTF and DSD. Correspondingly, Figure 20 shows the noise sections using four approaches. Since for the real case, we do not know the true answer, we are not able to numerically compare the performance of each approach as we did in the previous examples. Instead, we can only judge the performance by observation. The denoised section using $f-x$ deconvolution and its corresponding noise section are shown in Figures 19c and 20c, respectively. The denoising results show that the DSD thresholding removes the most noise while preserving the most significant details of the original seismic data.

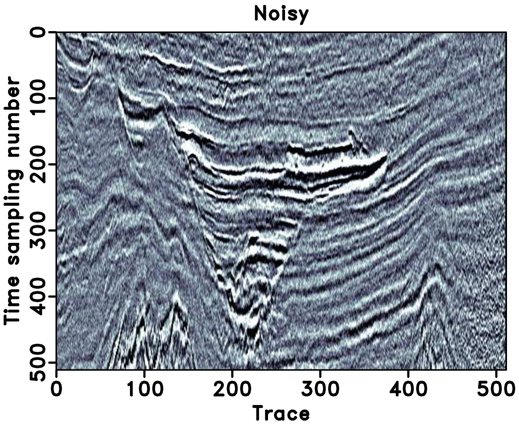


Figure 18: Real noisy seismic data.

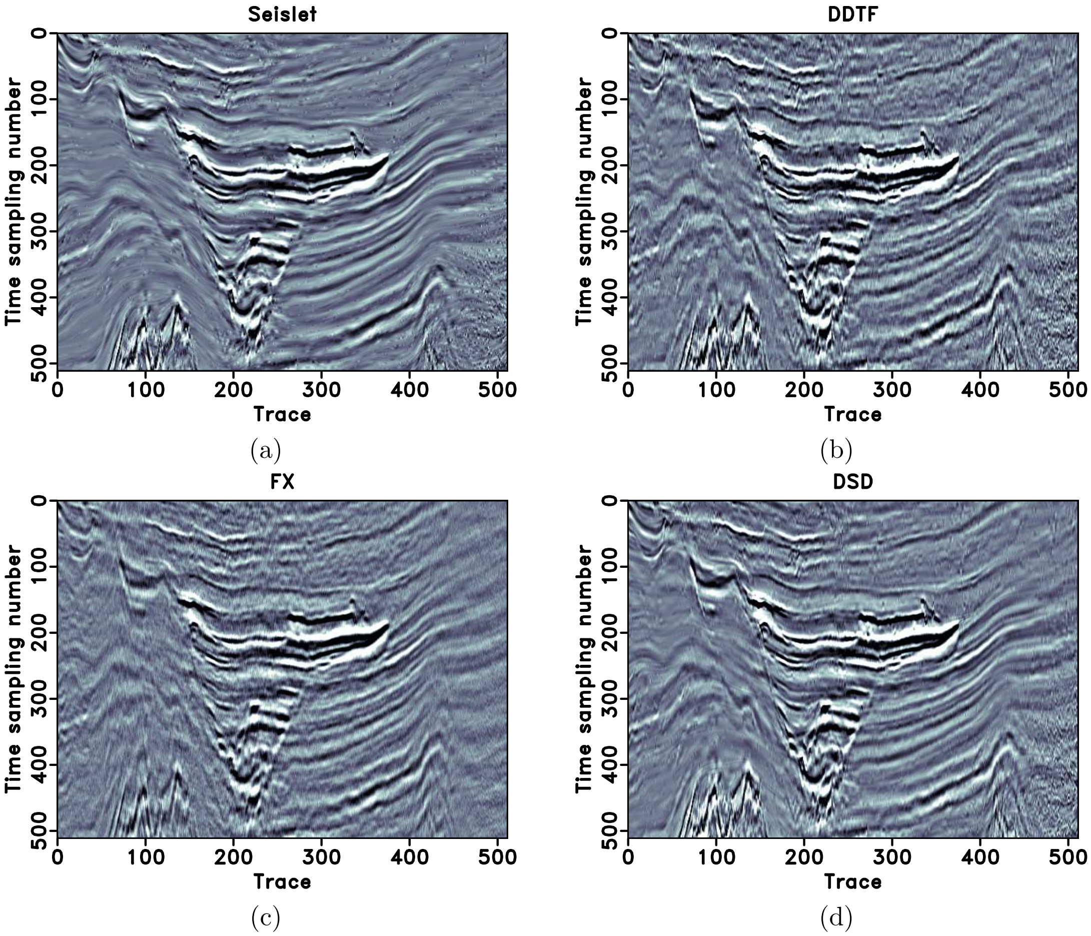
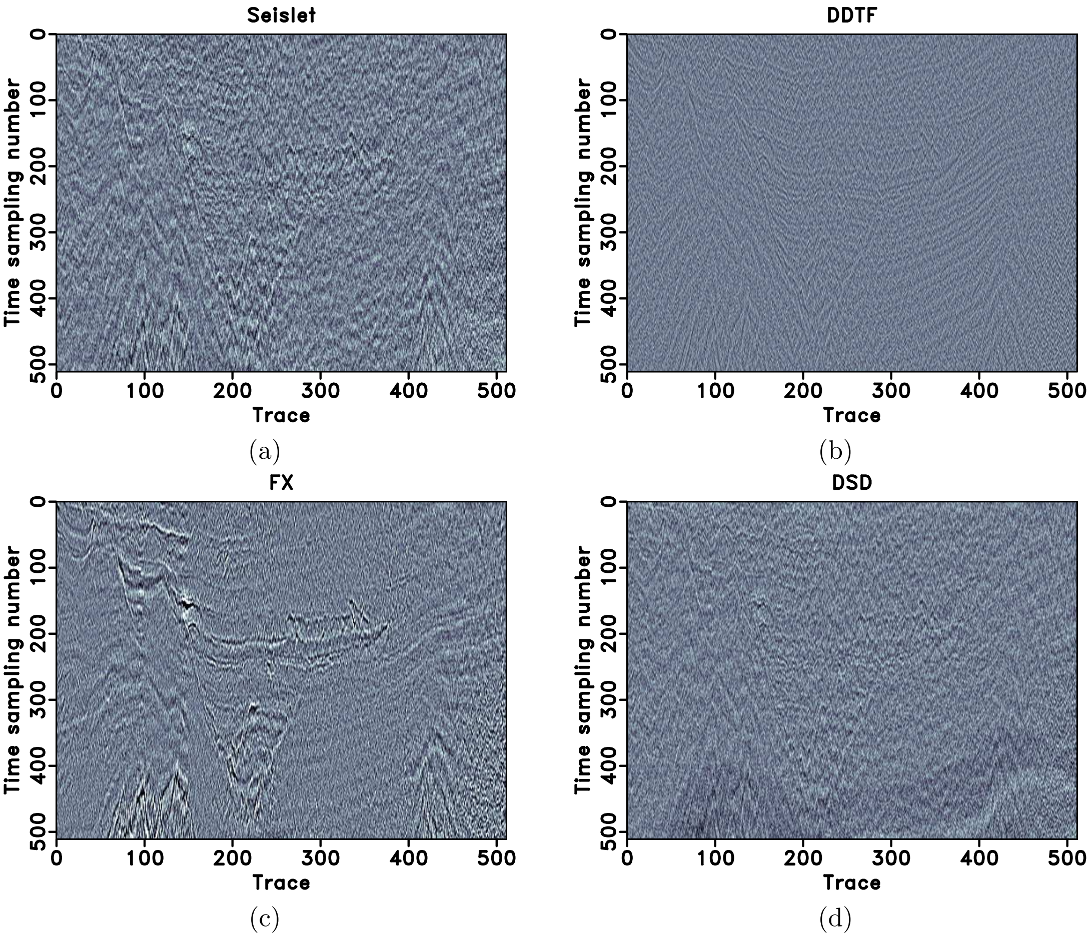


Figure 19: Denoised sections using different approaches. (a) Seislet thresholding. (b) DDTF thresholding. (c) $f-x$ deconvolution. (d) DSD thresholding. The best results are selected for each method.

| Percentage | Seislet | DDTF | DSD |
|---|---|---|---|
| 1 % | 12.67 | 16.35 | 19.25 |
| 2 % | 13.10 | 17.34 | **21.32** |
| 3 % | 13.31 | 18.32 | 20.65 |
| 4 % | 15.03 | **18.78** | 19.74 |
| 5 % | **15.63** | 18.56 | 19.32 |
| 10 % | 14.84 | 16.32 | 15.12 |
| 15 % | 13.33 | 12.85 | 11.54 |

Table 1: Comparison of SNR in dB using different approaches for first example (the original SNR is -7.31 dB).



(a) (b) (c) (d)

Figure 20: Noise sections using different approaches. (a) Seislet thresholding. (b) DDTF thresholding. (c) $f-x$ deconvolution. (d) DSD thresholding. The best results are selected for each method.

| Models | Original | Seislet | DDTF | DSD |
|---|---|---|---|---|
| Synthetic 2 | 20.04 | 33.67 (20%) | 29.43 (16%) | **37.25** (10%) |
| Field 1 | -3.015 | 23.12 (4%) | 18.04 (8%) | **26.34** (3%) |
| Field 2 | -0.94 | 15.54 (17%) | 18.23 (11%) | **21.97** (7%) |

Table 2: Comparison of SNR in dB using different approaches. The percentage corresponds to the optimum percentage of coefficients in the transform domain that can obtain the presented SNRs.

# DISCUSSION

The proposed DSD method attempts to create an optimal combination of the fixed-basis transform and learning-based dictionary, in order to enhance the performance of each individual type of transform and to overcome the disadvantages of each individual transform. The seislet transform used in this paper is data-adaptive but depends on the estimation of local slopes of seismic events. An incorrect estimation of local slopes may make the transform domain less sparse. The DDTF approach used in this paper may not be optimally efficient when large overlap of filtering windows and large number of iterations are used, because DDTF is sensitive to small-scale random noise and tends to learn the property of random noise when small window overlap or small number of iterations are used. The proposed hybrid framework can relieve the dependency of the seislet transform on slope estimation because the thresholding used for denoising is no longer solely relying on the sparsity of the seislet domain and instead is relying on the DSD domain. Because of the initial seislet transform, the transform domain becomes more compactly supported and distinguishes signal and noise better. DDTF applied after the seislet transform becomes less sensitive to noise and can achieve better performance when learning the behavior of useful signals. As a result, the window overlap and iteration number can both be smaller.

The proposed DSD approach follows the main idea of the multi-scale dictionary learning algorithm proposed by Ophir et al. (2011). The difference between our method and the previous approach is that Ophir et al. (2011) use K-SVD in the wavelet domain, while our method uses DDTF in the seislet domain. We use different tools but keep the same strategy. The seislet transform is more suitable than wavelet transform for seismic data. The DDTF is much faster than K-SVD for the dictionary learning stage. Ophir et al. (2011) split the data (actually split the subband coefficients) into overlapping patches, and then apply K-SVD on each patch. Analogously, we split the seislet subband data into overlapping patches, and apply DDTF on each patch. There is a control on how much overlapping is used. In general, it is advantageous to apply the maximally overlapping patches (i.e., only shift one column/row between the neighbor patches) for the training. This creates the ”richness”/”redundancy” in the training data that generates a shift-invariance for the dictionary. Such shift-invariant wavelets are helpful for denoising, which has been demonstrated in Coifman and Donoho (1995). In the DDTF, for each patch, the training filer is of tight frame or orthogonal, but the full transform matrix **W** is redundant (each column is a training filter on patches).

The double sparsity is more a concept than a specific type of transform. The seislet transform and DDTF used in this paper can be replaced by other combinations of a fixed-basis transform and a learning-based dictionary to achieve an overall performance which is better than that of each of the individual transforms. The DSD approach is also not limited to the application to random noise attenuation. Other possible applications include seismic data interpolation and regularization, simultaneous-source deblending, and sparsity-based seismic inversion.

Both the noisy synthetic data and the first two field data examples shown in this paper we simulated by adding Gaussian white noise. The advantage of using simulated data is that we know the true answer and can numerically calculate SNR in order to better compare and understand the properties of each transform-domain thresholding-based denoising approach. Although the Gaussian white noise is not always appropriate for simulating real random noise in field data, we use it here for simplicity. Changing the Gaussian white noise to colored noise and then performing similar tests is straightforward.

The fair comparison of different denoising approaches is always a difficult problem. In this paper, we only focus on the coefficient percentage for transform domain thresholding and fix other parameters, which mainly include the patch size of the tight frames and the number of iterations to update the tight frames.

## CONCLUSIONS

We have proposed a novel approach to compressing seismic data in order to perform transform-domain thresholding-based random noise attenuation. By considering the benefits of both analytic and learning-based sparsity-promoting transforms, we propose a DSD framework for compensating for the weaknesses of both approaches and sparsifying seismic data for random noise attenuation. We propose two models to construct the DSD framework. One model is based on learning in the data domain, and is called the synthesis-based DSD; the other model is based on learning in the model domain, and is called the analysis-based DSD. We select the seislet transform as the analytic-basis transform and DDTF as the learning-based sparse dictionary to construct a cascaded DSD framework based on the analysis-based DSD. Synthetic and field data examples with simulated random noise and one field data example with real noise demonstrate a superior performance of the proposed DSD in application to random noise attenuation of seismic reflection data.

## ACKNOWLEDGMENTS

We would like to thank Mauricio Sacchi, Jeffrey Shragge, Ian Moore, David Halliday, and three anonymous reviewers for constructive suggestions. The paper was prepared in the Madagascar open-source platform (Fomel et al., 2013), and all the examples are reproducible. This work is partially supported by the Natural Science Fund of China(grant NOs.:91330108, 41374121, and 61327013), the fundamental Research Funds for the Central Universities (grant NO.: HIT.BRETIV.201314), and the Program for New Century Excellent Talents in University (grant NO.: NCET-11-0804), and the Texas Consortium for Computational Seismology (TCCS).

# REFERENCES

Aharon, M., M. Elad, and A. M. Bruckstein, 2006, The K-SVD: An algorithm for designing of evercomplete dictionaries for sparse representation: IEEE Transactions on Signal Processing, **54**, 4311–4322.

Beckouche, S., and J. Ma, 2014, Simultaneous dictionary learning and denoising for seismic data: Geophysics, **79**, A27–A31.

Bryt, O., 2008, Compression of facial images using the K-SVD algorithm: Journal of Visual Communication and Image Representation, **19**, 270–283.

Cai, J., S. Huang, H. Ji, Z. Shen, and G. Ye, 2013, Data-driven tight frame construction and image denoising: Applied and Computational Harmonic Analysis, doi: 10.1016/j.acha.2013.10.001.

Candès, E. J., L. Demanet, D. L. Donoho, and L. Ying, 2006, Fast discrete curvelet transforms: SIAM, Multiscale Modeling and Simulation, **5**, 861–899.

Chen, Y., and S. Fomel, 2015, Random noise attenuation using local signal-and-noise orthogonalization: Geophysics, **80**, WD1–WD9.

Chen, Y., S. Fomel, and J. Hu, 2014, Iterative deblending of simultaneous-source seismic data using seislet-domain shaping regularization: Geophysics, **79**, V179–V189.

Chen, Y., and J. Ma, 2014, Random noise attenuation by *f-x* empirical mode decomposition predictive filtering: Geophysics, **79**, V81–V91.

Claerbout, J. F., 2010, Basic earth imaging: Stanford Exploration Project, http://sep.stanford.edu/prof/.

Coifman, R. R., and D. L. Donoho, 1995, Translation-invariant de-noising: Wavelets and Statistics, **103**, 125–150.

Do, M. N., and M. Vetterli, 2005, The contourlet transform: An efficient directional multiresolution image representation: IEEE Transactions on Image Processing, **14**, 2091–2106.

Donoho, D. L., 2006, Compressed sensing: IEEE transactions on Information Theory, **52**, 1289–1306.

Du, B., and L. R. Lines, 2000, Attenuating coherent noise by wavelet transform: Exploration Geophysics, **31**, 353–358.

Elad, M., J. L. Starck, P. Querre, and D. L. Donoho, 2005, Simultaneous cartoon and texture image inpainting using morphological component analysis (mca): Applied and Computational Harmonic Analysis, **19**, 340–358.

Engan, K., S. O. Aase, and J. H. Husoy, 1999, Method of optimal directions for frame design: IEEE International Conference on Acoustics, Speech, and Signal Processing, **5**, 2443–2446.

Fomel, S., 2002, Application of plane-wave destruction filters: Geophysics, **67**, 1946–1960.

Fomel, S., and Y. Liu, 2010, Seislet transform and seislet frame: Geophysics, **75**, V25–V38.

Fomel, S., P. Sava, I. Vlad, Y. Liu, and V. Bashkardin, 2013, Madagascar open-source software project: Journal of Open Research Software, **1**, e8.

Galbraith, M., 1991, Random noise attenuation by F-X prediction: A tutorial: 61st

Annual International Meeting, SEG, Expanded Abstracts, 1428–1431.

Gan, S., S. Wang, Y. Chen, Y. Zhang, and Z. Jin, 2015, Dealiased seismic data interpolation using seislet transform with low-frequency constraint: IEEE Geoscience and remote sensing letters, **12**, 2150–2154.

Gulunay, N., 1986, Fx decon and complex Wiener prediction filter: 56th Annual International Meeting, SEG, Expanded Abstracts, Session: POS2.10.

Hennenfent, G., L. Fenelon, and F. Herrmann, 2010, Nonequispaced curvelet transform for seismic data reconstruction: A sparsity-promoting approach: Geophysics, **75**, WB203–WB210.

Hennenfent, G., and F. Herrmann, 2006, Seismic denoising with nonunformly sampled curvelets: Computing in Science & Engineering, **8**, 16–25.

Ibrahim, A., and M. D. Sacchi, 2014a, Eliminating blending noise using fast apex shifted hyperbolic Radon transform: 76th Annual International Conference and Exhibition, EAGE, Extended Abstracts.

———, 2014b, Simultaneous source separation using a robust Radon transform: Geophysics, **79**, V1–V11.

Ioup, J. W., and G. E. Ioup, 1998, Noise removal and compression using a wavelet transform: 68th Annual International Meeting, SEG, Expanded Abstracts, 1076–1079.

Jafarpour, B., V. K. Goyal, D. B. McLaughlin, and W. T. Freeman, 2009, Transform-domain sparsity regularization for inverse problems in geosciences: Geophysics, **74**, R69–R83.

Jolliffe, I. T., 2002, Principal component analysis, 2nd ed.: New York: Springer.

Kaplan, S. T., M. D. Sacchi, and T. J. Ulrych, 2009, Sparse coding for data-driven coherent and incoherent noise attenuation: 79th Annual International Meeting, SEG, Expanded Abstracts, 3327–3331.

Labate, D., W. Lim, G. Kutyniok, and G. Weiss, 2005, Sparse multidimensional representation using shearlets: Proceedings of the SPIE, **5914**, 254–262.

LePennec, E., and S. Mallat, 2005, Sparse geometric image representations with bandlets: IEEE Transactions on Image Processing, **14**, 423–438.

Liang, J., J. Ma, and X. Zhang, 2014, Seismic data restoration via data-driven tight frame: Geophysics, **79**, V65–V74.

Liu, Y., and S. Fomel, 2010, OC-seislet: Seislet transform construction with differential offset continuation: Geophysics, **75**, WB235–WB245.

Luo, Y., and G. T. Schuster, 1992, Wave packet transform and data compression: 62nd Annual International Meeting, SEG, Expanded Abstracts, 1187–1190.

Ma, J., and G. Plonka, 2010, The curvelet transform: IEEE signal processing magazine, **27**, 118–133.

Mairal, J., F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, 2009, Non-local sparse models for image restoration: IEEE 12th International Conference on Computer Vision.

Mairal, J., G. Sapiro, and M. Elad, 2008, Learning multiscale sparse representations for image and video restoration: SIAM Multiscale Modeling Simulation, **7**, 214–241.

Mallat, S. G., 2009, A wavelet tour of signal processing: The sparse way: Academic Press.

Naghizadeh, M., and M. D. Sacchi, 2010, Beyond alias hierarchical scale curvelet interpolation of regularly and irregularly sampled seismic data: Geophysics, **75**, WB189WB202.

Neelamani, R., A. Baumstein, and W. S. Ross, 2010, Adaptive subtraction using complex-valued curvelet transforms: Geophysics, **75**, V51–V60.

Neelamani, R., A. I. Baumstein, D. G. Gillard, M. T. Hadidi, and W. L. Soroka, 2008, Coherent and random noise attenuation using the curvelet transform: The Leading Edge, **27**, 240–248.

Ophir, B., M. Lustig, and M. Elad, 2011, Multi-scale dictionary learning using wavelets: IEEE Journal of Selected Topics in Signal Processing, **5**, 1014–1024.

Protter, M., and M. Elad, 2009, Image sequence denoising via sparse and redundant representations: IEEE Trans Image Process, **18**, 27–35.

Quan, Y., T. Zhu, J. M. Harris, R. M. Burnstad, and S. E. Zarantonello, 2011, Image integration with learned dictionaries and application to seismic monitoring: SEG expanded abstracts: 81st Annual international meeting, 4217–4222.

Rubinstein, R., M. Zibulevsky, and M. Elad, 2010, Double sparsity: learning sparse dictionaries for sparse signal approximation: IEEE Transaction on signal processing, **58**, 1553–1564.

Sweldens, W., 1995, Lifting scheme: A new philosophy in biorthogonal wavelet constructions: Wavelet applications in signal and image processing iii: Proceedings of SPIE 2569, 68–79.

Vidal, R., Y. Ma, and S. Sastry, 2005, Generalized principal component analysis (gpca): IEEE Transactions on pattern analysis and machine intelligence, **27**, 1945–1959.

Wang, D., R. Saab, O. Yilmaz, and F. J. Herrmann, 2008, Bayesian wavefield separation by transform-domain sparsity promotion: Geophysics, **73**, A33–A38.

Yu, S., J. Ma, X. Zhang, and M. Sacchi, 2015, Denoising and interpolation of high-dimensional seismic data by learning tight frame: Geophysics, **80**, V119–V132.

Zhang, R., and T. Ulrych, 2003, Physical wavelet frame denoising: Geophysics, **68**, 225–231.

Zhang, X., M. Burger, and S. Osher, 2011, A unified primal-dual algorithm framework based on Bregman iteration: Journal of Scientific Computing, **46**, 20–46.

Zibulevsky, M., and B. A. Pearlmutter, 2001, Blind source separation by sparse decomposition in a signal dictionary: Neural computation, **13**, 863–882.