

A Matlab code package for 2D/3D local slope estimation and structural filtering

Hang Wang, Yunfeng Chen, Omar M. Saad, Wei Chen, Yapo Abolé Serge Innocent Oboué, Liuqing Yang, Sergey Fomel and Yangkang Chen

ABSTRACT

Local slope is an important attribute that can help distinguish seismic signals from noise. Based on the optimal slope estimation, many filtering methods can be designed to enhance the signal-to-noise ratio (S/N) of noisy seismic data. We present an open-source Matlab code package for local slope estimation and the corresponding structural filtering. This package includes 2D and 3D examples with two main executable scripts and related sub-functions. All code files are in the Matlab format. In each main script, local slope is estimated based on the well-known plane wave destruction (PWD) algorithm. Then, the seismic data are transformed to the flattened domain by utilizing this slope information. Further, the smoothing operator can be effectively applied in the flattened domain. We introduce the theory and mathematics related to these programs, and present the synthetic and field data examples to show the usefulness of this open-source package. The results of both local slope estimation and structural filtering demonstrate that this package can be conveniently and effectively applied to the seismic signal analysis and denoising.

INTRODUCTION

Local slope attribute is of great importance in the seismic exploration community. Based on the slope information, seismic signals can be enhanced or separated from background noise. This parameter can also help analyze the velocity model from the pre-stack data and act as a regularization term to constrain the seismic inversion. Among these applications, the structural filtering is an effective method to extract signals from noisy data or to regularize the model to be geologically plausible.

There have been many methods developed for estimating the seismic dip or local slope (Silva et al., 2015; Wang et al., 2015; Silva et al., 2016). Hu et al. (2015) speed up the beam migration by compressing the local slant stacks. They apply a new second-order structure tensor to estimate different slope fields in each data point. Based on these slope fields, the raw data can be decomposed into several components. Further, combined with the matching pursuit (MP), the data can be compressed and sparsely represented by several coefficients, i.e., the local slopes and the MP parameters. Since this method requires less storage, it can be fast implemented in

the subsequent processing with almost no quality degradation. Wu et al. (2019) propose a simple convolution network to simultaneously obtain the slope estimation, locate the faults and smooth the noisy seismic data. They synthesize many 3D noisy data as the input of the network and the clean data as the output. By training these data sets, an effective network architecture can be established and applied to the other general seismic data. This network is capable of producing more accurate processing results than the conventional methods. Lou et al. (2019) combine the advantages of semblance based and gradient structure tensor (GST) based methods to analyze the slope of seismic data volume. They first use the semblance based method to estimate an approximated slope field and then, along the slope direction, a local window is designed to include those events with relative stable slope. Finally, the gradient structure tensor (GST) method is used to calculate the dominant slope. The test result of a real field example demonstrates the effectiveness of this method. Griffiths et al. (2020) use the time-frequency-based approach to estimate the slope of multicomponent seismic data. First, the Stockwell transform is applied to obtain the time-frequency data for calculating the time-varying slope. They compare the proposed method with the conventional ones and find that this algorithm is more robust in the noisy environment and enables a more accurate and faster estimation of the local slope, which will naturally lead to a more satisfactory imaging result.

In addition to the methods mentioned above, plane-wave destruction (PWD) is also a popular approach to the estimation of the local slope. The idea of plane-wave destruction is first proposed by Claerbout (1992). Fomel (2002) reformulate the PWD operator as a Z-domain filter. Then, the estimation of local slope field can be treated as a non-linear inverse problem. By minimizing the residual components, the time-varying slope can be solved iteratively. Schleicher et al. (2009) reduce the computing cost of the seismic imaging tasks by utilizing the slope information of the seismic events. They add a correction term to the conventional linear plane-wave destruction for a more precise slope estimation. Compared with the nonlinear plane-wave destruction, the corrected method shows a comparable performance. This improved version can produce stable result even in an extremely noisy environment. Chen et al. (2013b) design a circle-interpolation model with 2D delay filter to estimate the local slope of complex structure, which is named as omni-directional plane-wave destruction (OPWD). When applied to the predictive painting and event picking, the proposed method can produce better results than the traditional plane-wave destruction. Chen et al. (2013a) use a analytical approach to estimate the slope field, which avoids the iterative calculations. Compared to the conventional PWD method, the proposed algorithm can achieve similar results with much less computational cost. Besides, this method can also be easily extended to 3D cases. Li et al. (2017) propose to use the local dip instead of the local slope as the parameter of plane-wave destruction. It can effectively handle the vertical structure by a linear iterative optimization framework. Wang et al. (2020) propose a non-stationary slope estimation framework to deal with the complex seismic data, which has different smoothing radii for varying data complexities.

With the help of the accurate slope estimation, many seismic data processing

(Porsani et al., 2010; Hellman, 2014; Swindeman and Fomel, 2015; Zhang et al., 2015; Hellman and Boyer, 2016), interpretation (Wang et al., 2015) and imaging tasks (Khoshnavaz et al., 2016; Stovas and Fomel, 2016) can be effectively implemented. It is particularly worth mentioning that a new transform called seislet, which is more applicable to the seismic data (Fomel, 2002; Chen et al., 2014; Geng et al., 2020), is also based on an accurate slope estimation. The applications of local slope are not limited to those aforementioned problems. Another noteworthy application of local slope is structural filtering (Fehmers and Hocker, 2003; Yang et al., 2014; Fan et al., 2016; Zhang et al., 2016; Gan et al., 2016; Kim et al., 2017; Xue et al., 2019; Xu et al., 2019; Zhou and He, 2019). In short, slope-based structural filtering is a means or median operator applied in a specific local window that is decided by the local slope at the corresponding locations. Zhou and Li (2018) propose an algorithm to simultaneously suppress the spike noise and reconstruct the missing data, which uses the robust slope estimation to assist the means and median filters. Huang et al. (2020) combine the slope-based structural filter with an inversion frame to iteratively suppress the erratic noise. Compared to the conventional median filtering, this proposed frame can handle the stronger erratic noise and produce a better result. Chen et al. (2020) design a space-varying structural-oriented median filter to alleviate the problem caused by the imprecise slope estimation of noisy data. By adaptively changing the filter lengths, this operator has stable performance in the examples with curve events. Further combined with a shaping regularization framework, it can produce much better denoising results than the conventional filters.

Considering the importance of the accurate slope field and the structural filtering, we present a useful Matlab code package to implement them. For the input noisy data, it is first preprocessed by a conventional denoising method (such as the traditional means or median filtering) to have a relative higher signal to noise ratio (S/N), which is beneficial to the following slope estimation. Then, the PWD method is used to calculate a precise slope field. Based on this field, the raw noisy data with curve events can be transformed to a flattened domain. The smoothing operator can be further applied along the flattened events to suppress the random noise. In this paper, we first review the basic algorithms of the code package, then introduce the demonstration scripts and the related subfunctions. Finally, we use some synthetic and field examples to show its performance on slope estimation and structural filtering.

METHOD

Plane wave destruction

The equation describing the plane wave can be expressed as:

$$\frac{\partial u}{\partial x} + \sigma \frac{\partial u}{\partial t} = 0, \quad (1)$$

where u is the wave field, σ stands for the local slope, t and x represent the time and offset dimensions. When the local slope σ remains unchanged, the solution for above

equation has a general form:

$$u(t, x) = f(t - \sigma x), \quad (2)$$

where $f(t)$ is the waveform function.

Transforming equation 2 into the frequency domain, we can obtain a new form of solution:

$$U(w, x) = F(w)e^{iw\sigma x} = U(w, 0)e^{iw\sigma x}, \quad (3)$$

where $U(w, x)$ and $F(w)$ are the Fourier transforms of $u(t, x)$ and $f(t)$, respectively. Further, a prediction formula based on equation 3 can be derived as:

$$U(w, x) = U(w, x - 1)e^{iw\sigma}. \quad (4)$$

This formula means that a trace can be predicted from its adjacent trace by multiplying the factor $e^{iw\sigma}$. If expressed in the Z-transform domain with $Z_t^{-\sigma} = e^{iw\sigma}$, equation 4 can be rewritten as :

$$U(Z_t, x) = U(Z_t, x - 1)Z_t^{-\sigma}, \quad (5)$$

or

$$U(Z_t, x) - U(Z_t, x - 1)Z_t^{-\sigma} = 0. \quad (6)$$

Further, if we make a convolution between the Z-transform data $[U(Z_t, 0), \dots, U(Z_t, N)]$ (N is the trace number) and a prediction filter $[1, -Z_t^{-\sigma}]$:

$$\begin{bmatrix} U(Z_t, 0), \dots, U(Z_t, N) \end{bmatrix} * \begin{bmatrix} 1, -Z_t^{-\sigma} \end{bmatrix} \downarrow \begin{bmatrix} U(Z_t, 0), U(Z_t, 1) - U(Z_t, 0)Z_t^{-\sigma}, \dots, U(Z_t, N) - U(Z_t, N-1)Z_t^{-\sigma}, U(Z_t, N) \end{bmatrix}, \quad (7)$$

where $*$ stands for the convolution operator. The filter $[1, -Z_t^{-\sigma}]$ is known as the destruction operator. Based on the equation 6, all the elements in $[U(Z_t, 0), U(Z_t, 1) - U(Z_t, 0)Z_t^{-\sigma}, \dots, U(Z_t, N) - U(Z_t, N-1)Z_t^{-\sigma}, U(Z_t, N)]$ except for the first one $U(Z_t, 0)$ and the last one $U(Z_t, N)$ are zeros. Thus, the convolution result is approximate to 0.

As the convolution in space domain is equal to the multiplication in Z-transform domain, the aforementioned equation 7 can be rewritten in Z domain as:

$$\begin{aligned} & U(Z_t, 0) + U(Z_t, 1)Z_x^1 \dots + U(Z_t, N)Z_x^N \\ & \times \\ & 1 - Z_t^{-\sigma}Z_x \\ & \downarrow \\ & U(Z_t, 0) + (U(Z_t, 1) - U(Z_t, 0)Z_t^{-\sigma})Z_x^1 + \dots + \\ & (U(Z_t, N) - U(Z_t, N-1)Z_t^{-\sigma})Z_x^N + U(Z_t, N)Z_x^{N+1}, \\ & \downarrow \\ & U(Z_t, 0) + 0 + \dots + 0 + U(Z_t, N)Z_x^{N+1} \\ & \approx \\ & 0 \end{aligned} \quad (8)$$

where \times is the symbol of multiplication. The above two equations indicate that the filter $[1, -Z_t^{-\sigma}]$ can well destruct the plane wave, and its Z-transform form, similar to equation 8, is given as:

$$Q(Z_t, Z_x) = 1 - Z_x Z_t^{-\sigma}, \quad (9)$$

and can be used as:

$$Q(Z_t, Z_x)U(Z_t, Z_x) = (1 - Z_x Z_t^{-\sigma})U(Z_t, Z_x) = 0, \quad (10)$$

where $U(Z_t, Z_x)$ is the Z transform of wavefield.

Further, for calculation convenience, we let the $Z_t^{-\sigma} = \frac{T(Z_t)}{T(1/Z_t)}$ where $T(Z_t)/T(1/Z_t)$ is the Taylor approximation of factor $e^{iw\sigma}$ (Fomel, 2002) and is the function with respect to local slope σ , and then the above equation can be formed as:

$$Q(Z_t, Z_x)U(Z_t, Z_x) = (1 - Z_x \frac{T(Z_t)}{T(1/Z_t)})U(Z_t, Z_x) = 0, \quad (11)$$

or

$$P(Z_t, Z_x)U(Z_t, Z_x) = (T(1/Z_t) - Z_x T(Z_t))U(Z_t, Z_x) = 0, \quad (12)$$

where $P(Z_t, Z_x) = T(1/Z_t) - Z_x T(Z_t)$ is another form of PWD filter $Q(Z_t, Z_x)$.

Here, we list the detailed expressions of the filters $T(Z_t)$ and $T(1/Z_t)$ according to the Fomel (2002). Note that the Taylor approximation can have three or five-order accuracy, we use the three-order one as an example:

$$T(Z_t) = \frac{(1-\sigma)(2-\sigma)}{12} Z_t^{-1} + \frac{(2+\sigma)(2-\sigma)}{6} + \frac{(1+\sigma)(2+\sigma)}{12} Z_t, \quad (13)$$

and

$$T(1/Z_t) = \frac{(1-\sigma)(2-\sigma)}{12} Z_t + \frac{(2+\sigma)(2-\sigma)}{6} + \frac{(1+\sigma)(2+\sigma)}{12} Z_t^{-1}. \quad (14)$$

Let the time-domain forms of filter $T(Z_t)$ and $T(1/Z_t)$ as:

$$\mathbf{T} = \left[\frac{(1-\sigma)(2-\sigma)}{12}, \frac{(2+\sigma)(2-\sigma)}{6}, \frac{(1+\sigma)(2+\sigma)}{12} \right], \quad (15)$$

and

$$\mathbf{T}_{-1} = \left[\frac{(1+\sigma)(2+\sigma)}{12}, \frac{(2+\sigma)(2-\sigma)}{6}, \frac{(1-\sigma)(2-\sigma)}{12} \right], \quad (16)$$

the implementation of the destruction process in equation 12 can be expressed as: for the traces at locations $x+1$ and x (i.e., \mathbf{U}_{x+1} and \mathbf{U}_x), we use the filters \mathbf{T} and \mathbf{T}_{-1} to convolve with the two traces.

$$p_1 = U_x(t-1)T(3) + U_x(t)T(2) + U_x(t+1)T(1), \quad (17)$$

and

$$\begin{aligned} p_2 &= U_{x+1}(t-1)T_{-1}(3) + U_{x+1}(t)T_{-1}(2) + U_{x+1}(t+1)T_{-1}(1) \\ &= U_{x+1}(t-1)T(1) + U_{x+1}(t)T(2) + U_{x+1}(t+1)T(3) \end{aligned} \quad (18)$$

Then, by subtracting the convolution result p_1 from p_2 , we can get the destruction residual r , which should be close to zero:

$$\begin{aligned} r &= p_2 - p_1 \\ &= (U_{x+1}(t-1) - U_x(t+1))T(1) + \\ &\quad (U_{x+1}(t) - U_x(t))T(2) + \\ &\quad (U_{x+1}(t+1) - U_x(t-1))T(3) \end{aligned} \quad (19)$$

By minimizing the residual r , the corresponding slope σ can be solved.

Estimating local slope

To solve equation 12 and obtain the slope estimation, Fomel (2002) reformulates the equation 12 as a nonlinear inverse problem that can be expressed as follows:

$$\mathbf{P}(\boldsymbol{\sigma})\mathbf{u} \approx \mathbf{0}, \quad (20)$$

where $\mathbf{P}(\boldsymbol{\sigma})$ denotes the 2D PWD filter $P(Z_t, Z_x)$ with the σ varying throughout the whole section. As \mathbf{P} is a nonlinear convolution operator, the equation should be further linearized by the following iterative form:

$$\mathbf{P}(\boldsymbol{\sigma}_n)\mathbf{u} + \mathbf{G}\Delta\boldsymbol{\sigma}_n \approx \mathbf{0}, \quad (21)$$

where $\mathbf{G} = \mathbf{P}'(\boldsymbol{\sigma}_n)\mathbf{u}$ is the Jacobian matrix. $\Delta\boldsymbol{\sigma}_n$ is the update value of n_{th} non-linear iteration.

We solve equation 21 by using the shaping regularization framework (Fomel, 2007):

$$\Delta\boldsymbol{\sigma}_n^m = \mathbf{S}[\Delta\boldsymbol{\sigma}_n^{m-1} + \mathbf{G}^T(\mathbf{P}(\boldsymbol{\sigma}_n)\mathbf{u} - \mathbf{G}\Delta\boldsymbol{\sigma}_n^{m-1})], \quad (22)$$

where $\Delta\boldsymbol{\sigma}_n^m$ represents the updated value corresponding to the m_{th} linear iteration. \mathbf{S} is the shaping operator. Further, the converged model is formulated as follows:

$$\Delta\boldsymbol{\sigma}_n = \mathbf{H}[\lambda^2\mathbf{I} + \mathbf{H}^T(\mathbf{G}^T\mathbf{G} - \lambda^2\mathbf{I})\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{G}^T\mathbf{P}(\boldsymbol{\sigma}_n)\mathbf{u}, \quad (23)$$

where \mathbf{H} is the triangle smoothing operator with the radii L_x, L_y (and L_z for 3D case), which has the relationship $\mathbf{S} = \mathbf{H}\mathbf{H}^T$ with shaping operator \mathbf{S} . λ is a flexible parameter controlling the relative scaling of the forward operator \mathbf{G} .

Structural filtering

After obtaining the accurate slope estimation in the above subsection, one can flatten and smooth the noisy seismic data based on it. The flattening operation is implemented in the local window and will create a local flattened domain. That being said, if we have a 2D raw data $u(t, x)$ with size $T \times X$, the flattened data will have the form

$u(t, x, l)$ with size $T \times X \times (2R + 1)$, where the integer l ($l \in [-R, R]$) denotes the local flattened domain. R is the half length of the local window. For a certain trace $u(:, x)$, it can derive a corresponding flattened gather $u(:, x, :)$, where the symbol $:$ denotes all the elements in this dimension. Note that the original trace $u(:, x)$ is at the center of the local flattened gather and thus, corresponds to $l = 0$, i.e., $u(:, x, 0)$. Each trace in the gather $u(:, x, :)$ can be defined by the following prediction formulas according to Fomel (2010):

$$\mathbf{u}_l = \mathbf{F}_{x+1,x} \mathbf{F}_{x+2,x+1} \dots \mathbf{F}_{x+l,x+l-1} \mathbf{u}'_{x+l}, \quad (24)$$

and

$$\mathbf{u}_{-l} = \mathbf{F}_{x-1,x} \mathbf{F}_{x-2,x-1} \dots \mathbf{F}_{x-l,x-l-1} \mathbf{u}'_{x-l}, \quad (25)$$

where \mathbf{u}_l is the vector form of $u(:, x, l)$ and \mathbf{u}'_x stands for the $u(:, x + l)$, i.e., the original 2D data. $\mathbf{F}_{i,j}$ represents the prediction operator that predicts \mathbf{u}'_j from \mathbf{u}'_i . Note that this operator is closely related to the above estimated slope. Equations 24 and 25 denote the predictions from the right and left sides, respectively.

After getting the flattened data, the smoothing filter can be applied along the structural direction in the flattened domain. One flattened gather $u(:, x, :)$ corresponds to one smoothed trace which will be put in the corresponding position of the denoised result $\hat{u}(:, x)$. Compared to the conventional smoothing filter, the proposed one makes full use of the slope information and thus can significantly reduce the signal damage.

CODE DESCRIPTION

Code for 2D data

The name of the main script is ‘test_2D_SOF.m’ that has three parts including generating the test data, local slope estimation and the structural filtering. Note that one can replace the data generation part with existing synthetic or field data file to test their own dataset. This script calls all functions in the ‘code2D’ folder. The detailed descriptions of these three parts and their related functions are listed below and in Figure 1.

Part I: Generating the test data

This part includes a function called ‘str_ricker.m’ that can generate the synthetic wavelet. The input parameters are the dominate frequency f , the time interval dt and the duration $tlength$. The output is the discrete sequence of the wavelet w . We design an anticline model shown in Figure 20a stored in variable t . By convolving w and t , we get the synthetic model in $data$. Then we add the random noise and obtain the noisy data in dn . Note that the ‘MyColormaps.mat’ is a colormap file used to plot the figures.

Part II: Estimating the local slope

In this part, we first use the lateral smoothing operator to pre-process the noisy data to obtain $dtemp$. We then use the function ‘str_dip2d.m’ to calculate the slope field. Its important inputs are the 2D seismic data din , the non-linear iteration number $niter$, the linear iteration number $liter$, PWD accuracy $order$ and size of triangle smoothing operator $rect$. The output is the 2D slope field. The subfunction ‘str_conv_allpass.m’ is used to calculate the $\mathbf{P}(\sigma)\mathbf{u}$ in variable $u2$ and its Jacobian matrix \mathbf{G} in variable $u1$. This procedure corresponds to the equation 21. The destruction operator in equation 15 and 16 are calculated by its subfunctions ‘B3.m’ for the three-order accuracy or ‘B5.m’ for the five-order accuracy. Then, ‘str_divne.m’ utilizes them to output the update value $\Delta\sigma$ of slope field, which is related to the equations 22 and 23. The iterative shaping regularization method is implemented by its subfunction ‘str_conjgrad.m’.

Part III: Flattening the events and smoothing the data

The function ‘str_pwsMOOTH_lop2d.m’ combining the flattening and smoothing procedures is used in this part. The slope field obtained from the last part is used as the input of the function to flatten the slanting events. The user can choose a proper parameter that controls the length R of the smoothing operator. A larger length corresponds to a stronger smoothing effect. Its inputs are the noisy data dn , the estimated slope dip , the smoothing length ns corresponding to R , the PWD order $order$ and the regularization parameter eps . Its subfunction, the ‘str_pwspray_lop2d.m’ with the same input parameters, is the one outputting the flattened data with the additional flattened dimension. This function corresponds to equations 24 and 25. After getting the flattened data, smoothing operator along the flattened direction is applied to produce the final denoised data. Besides, the ‘str_snr.m’ is used to calculate the S/N of the noisy or the denoised data to evaluate the performance of the presented package.

Code for 3D data

The main script of the 3D example is ‘test_3D_SOF.m’ which also has three parts like the 2D script. However, the first part of it directly loads the prepared field data instead of generating the synthetic data. This script calls all the functions in the ‘code3D’ folder. The descriptions of the main script and its related subfunctions are detailed as follows. The structure of this folder is shown in Figure 2.

Part I: Loading the 3D data

This part simply loads the field 3D data ‘real3d.mat’ with complex geological structures and plots the image of it.

Part II: Estimating the local slope

We use the function ‘str_dip3d.m’ to calculate the slope field of the 3D data. Its input parameters are the same as the 2D case, i.e., the 3D seismic data din , the non-linear iteration number $niter$, the linear iteration number $liter$, PWD accuracy $order$

and size of triangle smoothing operator *rect*. Note that this *rect* has three elements for the three dimensions while the ‘str_dip2d.m’ only has two elements. The output difference from the 2D estimation function is that this function outputs two slope fields, i.e., one for the inline direction dip_i and the other for the crossline direction dip_x . The functions used to calculate the $\mathbf{P}(\sigma)\mathbf{u}$ in variable $u2$ and its Jacobian matrix \mathbf{G} in variable $u1$ are ‘str_conv_allpass_i.m’ and ‘str_conv_allpass_x.m’. As the 3D data has two dip fields in both inline and crossline directions, there are two subfunctions for it. The other subfunction ‘str_divne.m’ is the same as the 2D case.

Part III: Flattening the events and smoothing the data

The function ‘str_pwsingle_lop3d.m’ is used to flatten and smooth the 3D data in this part. The two slope fields along with the two corresponding smoothing lengths in the inline and crossline directions are input to the function to respectively control the flattened result and smoothing strength. Note that the ‘str_pwspray_lop3d.m’ is the subfunction that forms the additional flattened dimension.

EXAMPLES

We use five examples to demonstrate the effectiveness of the presented open-source package, including two synthetic and three real data examples. In all these examples, we use the destruction operator with three-order accuracy to calculate the slope. Figure 3a is the pre-stack clean data with size of 256×128 . The sampling rate is 1 ms. We add random noise to it to generate the noisy data (Figure 3b). The noise variance is 0.2. We use the traditional means filter as the preprocessing method, which outputs the roughly denoised result for the subsequent slope estimation. The estimated slope field is shown in Figure 3c. The slope estimation of the preconditioned data can produce a useful slope map that indicates the right event dips. According to the estimated slope field, we can flatten the traces and get the flattened data in Figure 4. From this diagram, we can find that, in the flatten domain, the events in all 5 traces are in the same time position. Figures 5a-5d are the comparison between the structural filter and the traditional means filter without using the slope information. Figure 5a is the denoised result of the structural filter, and Figure 5b is the one from the conventional smoothing filter. We find that the former one has obviously less residual noise than the latter one, and preserves the energy of the events well. Their S/Ns are 16.98 dB and 11.81 dB, respectively. Note that both of them use the same smoothing length of $R = 2$. The corresponding removed noise also confirms the superiority of the proposed code package because its noise section (Figure 5c) has much less signal leakage.

Figure 6a is a post-stack clean data with complex structures (anticline and faults). Its size is 302×214 . The sampling rate is 1ms. Figure 6b is the corresponding noisy data with 0.2 noise variance. The estimated slope based on the preprocessed data is shown in Figure 6c, which correctly indicates the shapes of curve events. Based on the estimated slope information, the flattened data can be obtained in Figure 7. Figures

8a and 8b are the filtered synthetic data, respectively. The smoothing lengths are all chosen as $R = 3$. It is clear that the result obtained from the proposed approach contains less residual noise than that from the conventional smoothing filter. Figures 8c and 8d correspond to the removed noise by these two methods. One can easily find that, although both of them slightly damage signals in the faulting areas, the plain smoothing filter causes much more obvious signal leakage near the complex structures. Their corresponding S/Ns are 14.39 dB and 12.07 dB, respectively.

To investigate the denoising performance of the proposed method under different noise environments, we gradually increase the noise variance from 0.2 to 1 in the above post-stack model and keep all parameters unchanged, i.e., $R = 2$ and three-order accuracy. Figure 9 is the corresponding output S/Ns for the proposed (blue) and conventional (red) methods. From the comparison, we find that the performances of both methods will degrade with the increase of the noise energy, and the proposed method always outperforms the conventional one.

Figure 10a is a field shot gather containing random noise with size of 800×61 . The sampling rate is 1 ms. Figures 10b and 11 are the estimated slope and the corresponding flattened data, respectively. The two sides of Figure 10b correspond separately to the upward (blue) and downward (red) events. In this example, we use $R = 1$. Figures 12a and 12b are the denoised results from the structural filter and the conventional smoothing filter. From the comparison, we find that the event energy of the former one is obviously stronger than the latter one, which indicates less signal damage. Additionally, the leaked energy in the removed noise section from the proposed method (Figure 12c) is much less than the conventional filter in Figure 12d (see the rectangles).

Figure 13a is a post-stack field example with size of 201×251 . Its time interval is 1ms. This data contains the stratigraphic overlap and anticline. Figure 13b is its slope estimation, and Figure 14 is the 3D data in the flatten domain. Figures 15a and 15b are the denoised results from the two methods. The proposed method apparently preserves the signal details better than the simple smoothing filter, especially in the right flank of the anticline. The smoothing length here is $R = 2$. We also highlight the difference of the leaked signals by the rectangles in their removed noise sections (Figures 16a and 16b).

The 3D raw data that we use here is taken from the 3D dataset in Chen et al. (2016) with a size of $101 \times 51 \times 10$ and a sampling rate of 1 ms. Figure 17a is the 3D plot of the data where we can see complex structures at the corner. Figure 17b is the 5th inline slice of the 3D data which contains random noise. Figure 17c is the local slope field of the 5th inline. As the noise energy in this data is not so strong, we omit the preprocessing procedure and directly use the raw data to estimate the slope. Despite the presence of noise in the original data, a smooth slope can still be estimated, especially in the region with large dips (see the black rectangle). To better show the details of the denoised and removed noise sections, we extract the corresponding slice figures of the 5th inline in Figures 18a-18d. Figures 18a and 18b are the denoised results of the structural filter and the traditional means filter,

respectively. It is very clear that the latter one causes more damages to the signals in the area with complex structures. They are indicated by the black rectangles. The difference sections in Figures 18c and 18d also confirm the better performance of the structural filtering. Note that these two algorithms both use the same smoothing length of $R = 2$.

DISCUSSION

We also test the potential application of the proposed method on the data with impulse noise. The clean and noisy data are shown in Figures 19a and 19b. Figures 19c and 19d are the filtered sections by the proposed and conventional methods. Note that, for better handling of the impulse noise, we replace the smoothing filter by the median filter and apply it to the flattened data. We can find that the proposed method can still effectively suppress the impulse noise and get stronger events than the conventional one. The difference sections in Figures 19e and 19f also show that the proposed framework can better preserve the complex structures.

We further test the influence of the dip angle on the final denoised result. We keep the other parameters unchanged and change the dip angle of the synthetic event. Figures 20a-20c correspond to the denoised results from different data sets with gradually increasing slopes and the same noise level. The smoothing length is $R = 2$. It is very clear that when geology structure is gentler, the denoising performance is better. With the increase of the event dip, the signal leakage becomes more obvious. The noise sections in Figures 20d-20f also confirm this phenomenon.

Additionally, we discuss the influence of the smoothing length R . For the same original noisy data as Figure 20a, we set R to 2,4,6 and compare the denoising performance of the proposed method. Figures 21a-21c show the denoised results, and Figures 21d-21f show the corresponding noise sections. From the comparison, we find that a larger smoothing length is beneficial to suppress random noise more effectively and output a cleaner result at the cost of producing more signal leakage (see the obvious leakage in the middle of Figure 21f).

CONCLUSION

We present an open-source Matlab code package for the slope estimation and structural filtering, which can be used in both 2D and 3D seismic data. We first illustrate the mathematical theory of the code framework. Then, we describe the contents of the main scripts in detail. Both the 2D and 3D cases have three parts in their scripts. The first one is to load or generate the input raw data. The second is to calculate the local slope field. The last part is to apply the slope information to the structural filtering. Compared with the simple means filtering, the presented code package can effectively suppress the random noise while causing less signal leakage. The reason why it has such an advantage is that the pre-calculated local slope field can help

flatten the slanted events, and thus, the smoothing filter can be applied along the structure direction. The synthetic and field data with slanted events all demonstrate its superiority over the conventional method. We also discuss the impacts of some important parameters on the final smoothing results, which gives the suitable circumstances to use this method. Due to the standalone and easy-to-follow features, the presented open-source package could have a wide impact in many seismic processing, imaging, and interpretation tasks.

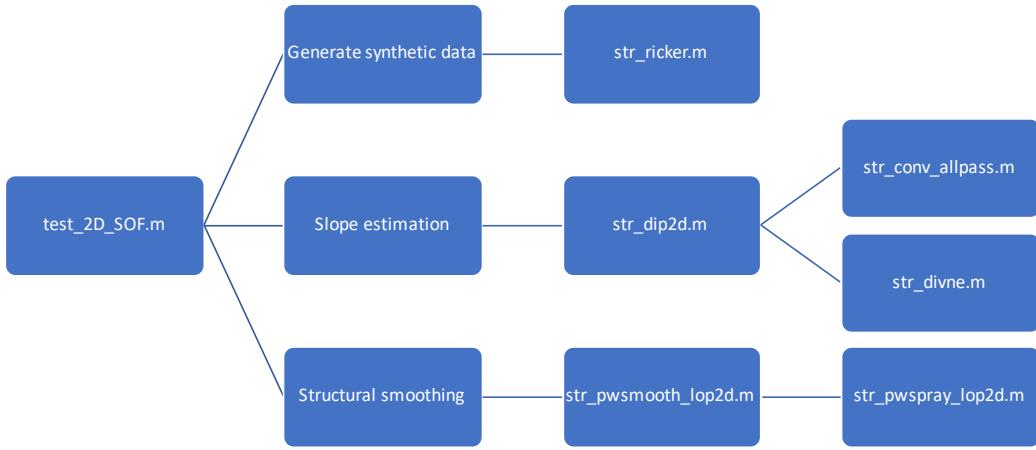


Figure 1: The code structure of 2D slope estimation and structural filtering.

REFERENCES

- Chen, Y., S. Fomel, and J. Hu, 2014, Iterative deblending of simultaneous-source seismic data using seislet-domain shaping regularization: *Geophysics*, **79**, no. 5, V179–V189.
- Chen, Y., W. Huang, D. Zhang, and W. Chen, 2016, An open-source matlab code package for improved rank-reduction 3d seismic data denoising and reconstruction: *Computers & Geosciences*, **95**, 59–66.
- Chen, Y., S. Zu, Y. Wang, and X. Chen, 2020, Deblending of simultaneous source data using a structure-oriented space-varying median filter: *Geophysical Journal International*, **222**, no. 3, 1805–1823.
- Chen, Z., S. Fomel, and W. Lu, 2013a, Accelerated plane-wave destruction: *GEO-PHYSICS*, **78**, no. 1, V1–V9.
- , 2013b, Omnidirectional plane-wave destruction: *Geophysics*, **78**, no. 5, V171–V179.

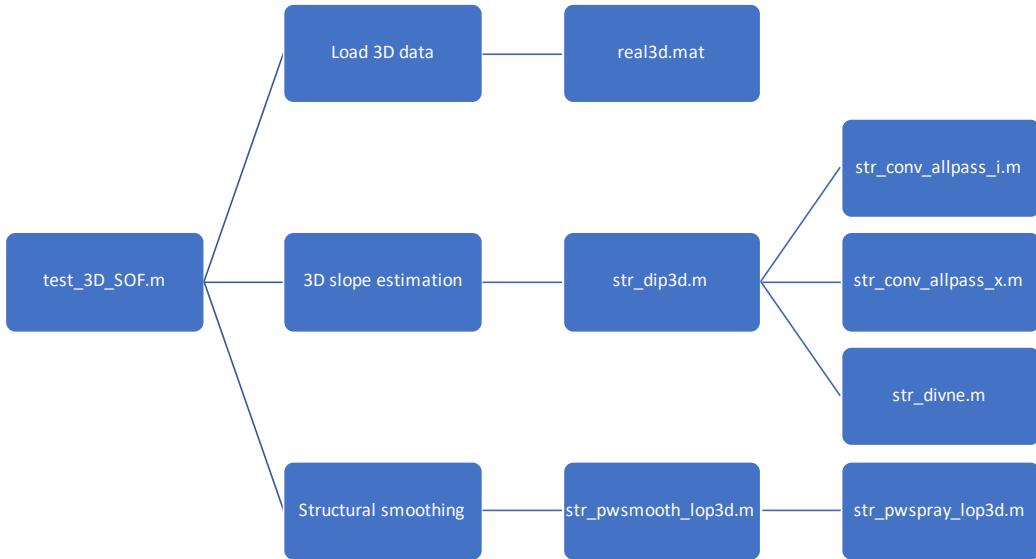


Figure 2: The code structure of 3D slope estimation and structural filtering.

- Claerbout, J. F., 1992, Earth soundings analysis: Processing versus inversion: Blackwell Scientific Publications, Inc.
- Fan, J.-W., Z.-C. Li, K. Zhang, M. Zhang, and X.-T. Liu, 2016, Multisource least-squares reverse-time migration with structure-oriented filtering: *Applied Geophysics*, **13**, no. 3, 491–499.
- Fehmers, G., and C. Hocker, 2003, Fast structural interpretation with structure-oriented filtering: *Geophysics*, **68**, no. 4, 1286–1293.
- Fomel, S., 2002, Applications of plane-wave destruction filters: *Geophysics*, **67**, no. 6, 1946–1960.
- , 2007, Shaping regularization in geophysical-estimation problems: *Geophysics*, **72**, R29–R36.
- , 2010, Predictive painting of 3d seismic volumes: *Geophysics*, **75**, A25–A30.
- Gan, S., S. Wang, Y. Chen, X. Chen, and K. Xiang, 2016, Separation of simultaneous sources using a structural-oriented median filter in the flattened dimension: *Computers & Geosciences*, **86**, 46–54.
- Geng, Z., X. Wu, S. Fomel, and Y. Chen, 2020, Relative time seislet transform: *Geophysics*, **85**, no. 2, V223–V232.
- Griffiths, M. P., A. J.-M. Pugin, and D. Motazedian, 2020, Estimating local slope in the time-frequency domain: Velocity-independent seismic imaging in the near surface: *Geophysics*, **85**, no. 5, U99–U107.
- Hellman, K., 2014, Simultaneous estimation of crs parameters with multi-dimensional local slopes: 84th Annual International Meeting, SEG, Expanded Abstracts, 5183.
- Hellman, K., and S. Boyer, 2016, A local slope approach to multidimensional trace

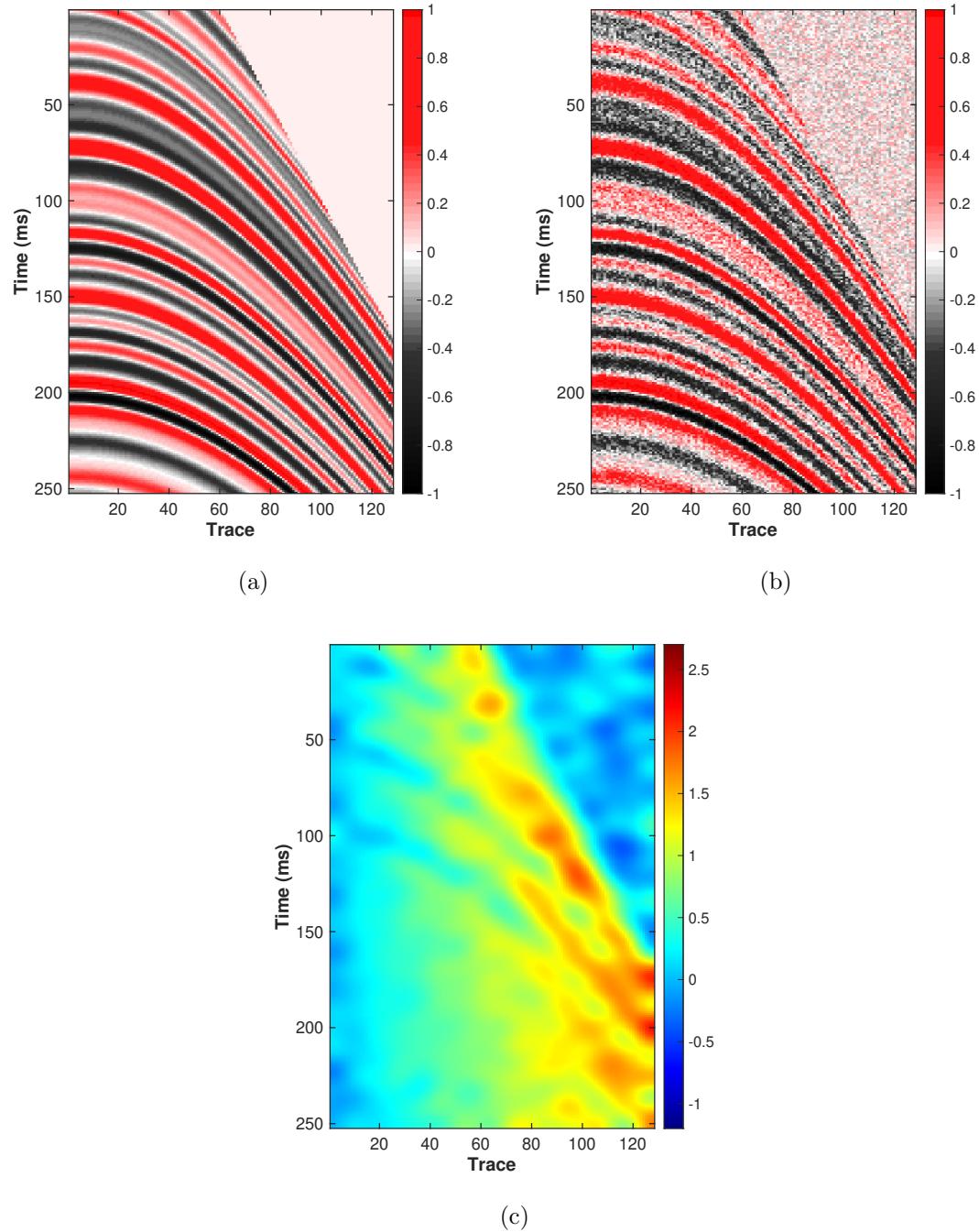


Figure 3: The synthetic pre-stack 2D data. (a) The clean data. (b) The noisy data with random noise (variance=0.2). (c) The local slope field of synthetic pre-stack 2D data. Note that it is calculated based on the preprocessed data.

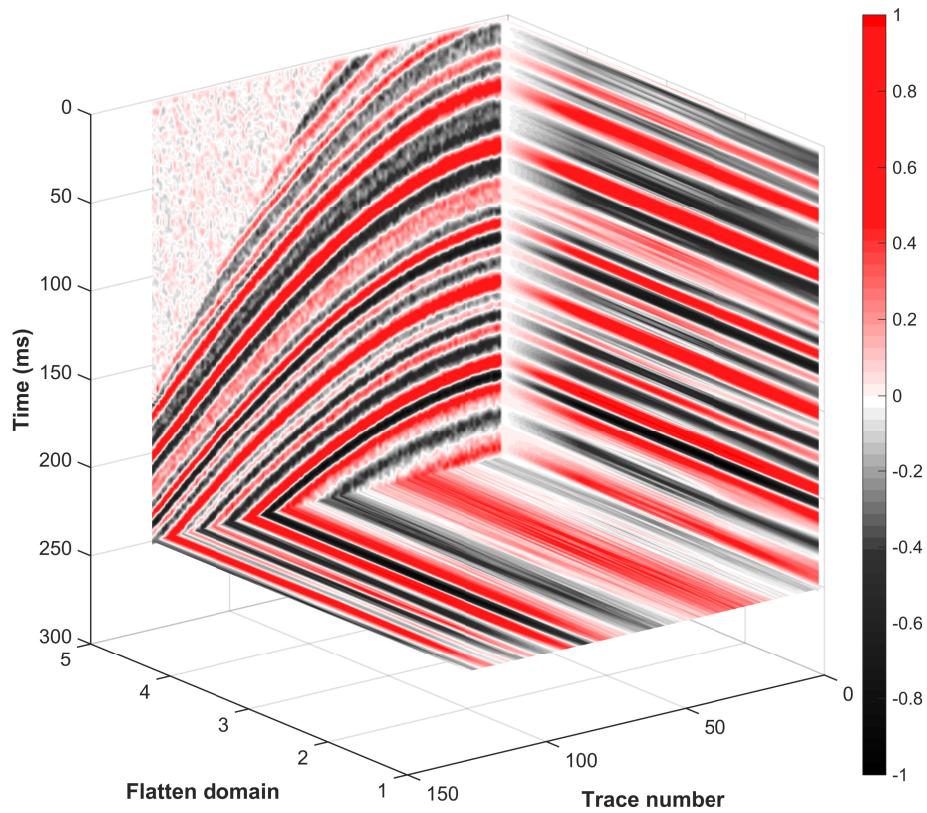


Figure 4: The flattened data of synthetic pre-stack 2D example.

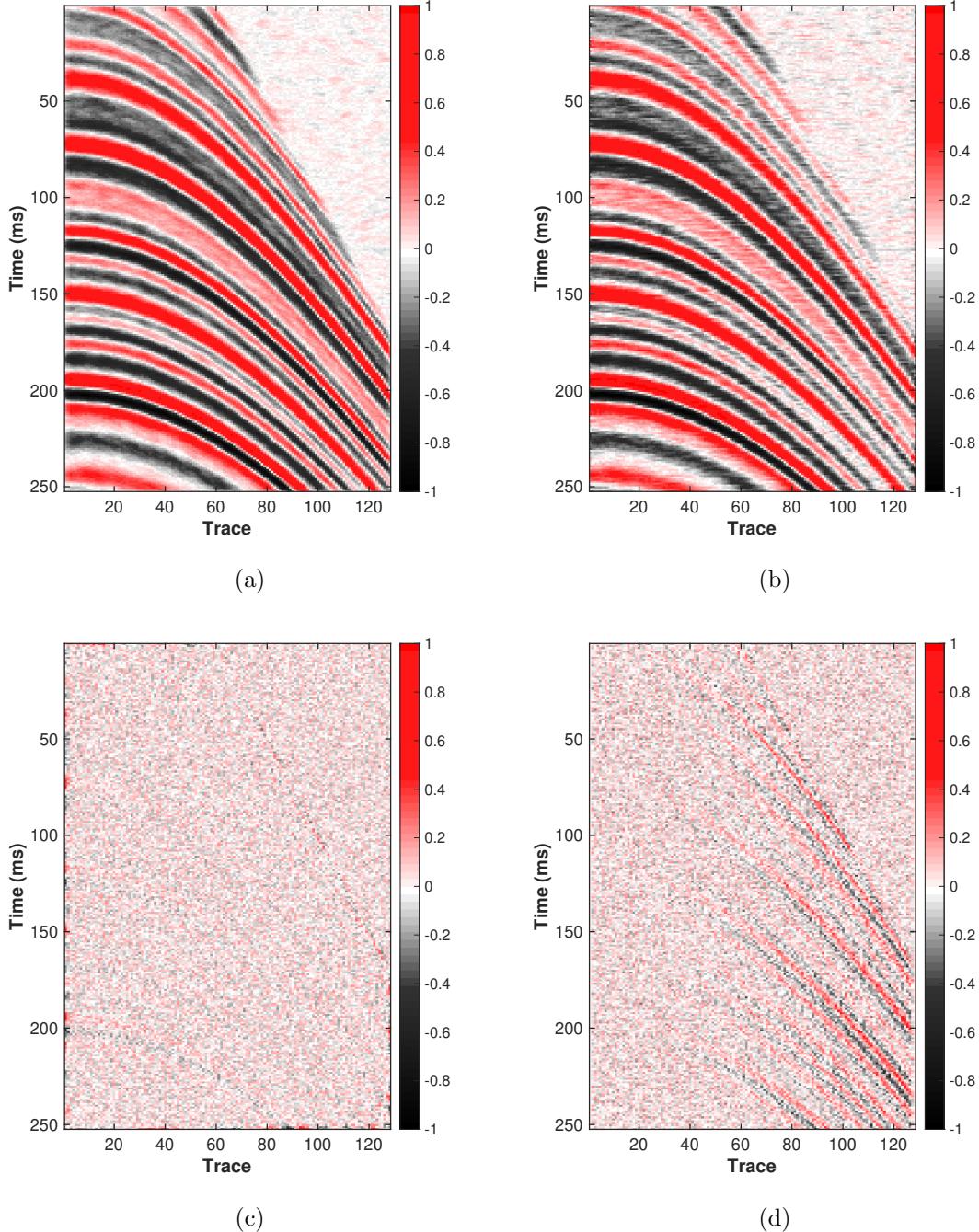


Figure 5: The denoised results and removed noise of pre-stack synthetic 2D data. (a) The result by the proposed method ($S/N=16.98\text{dB}$). (b) The result by the conventional smoothing filter without using the slope information ($S/N=11.81\text{dB}$). (c) The removed noise of the proposed method. (d) The removed noise of the conventional smoothing filter.

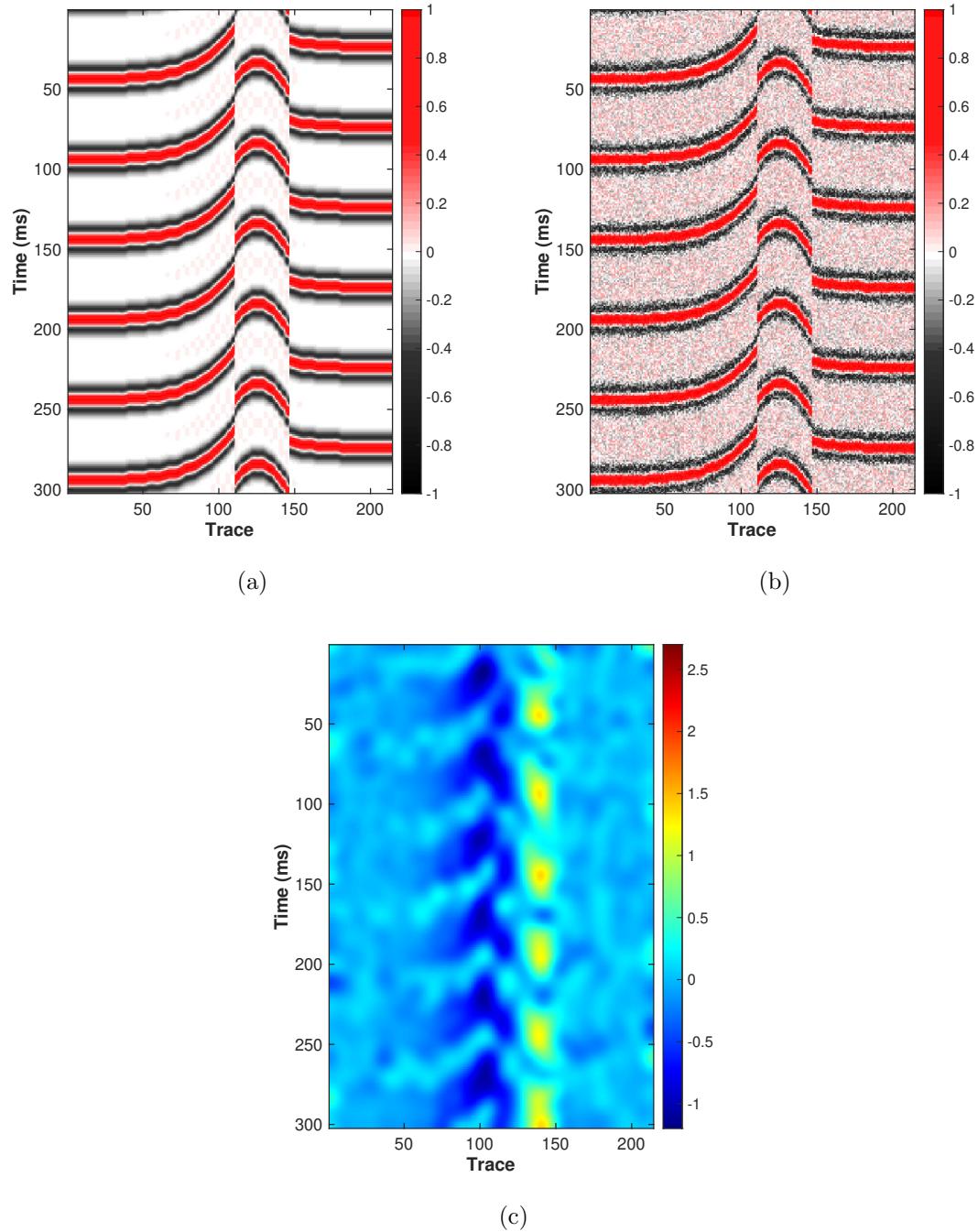


Figure 6: The synthetic post-stack 2D data. (a) The clean data. (b) The noisy data with random noise (variance=0.2). (c) The local slope field of synthetic post-stack 2D data. Note that it is calculated based on the preprocessed data.

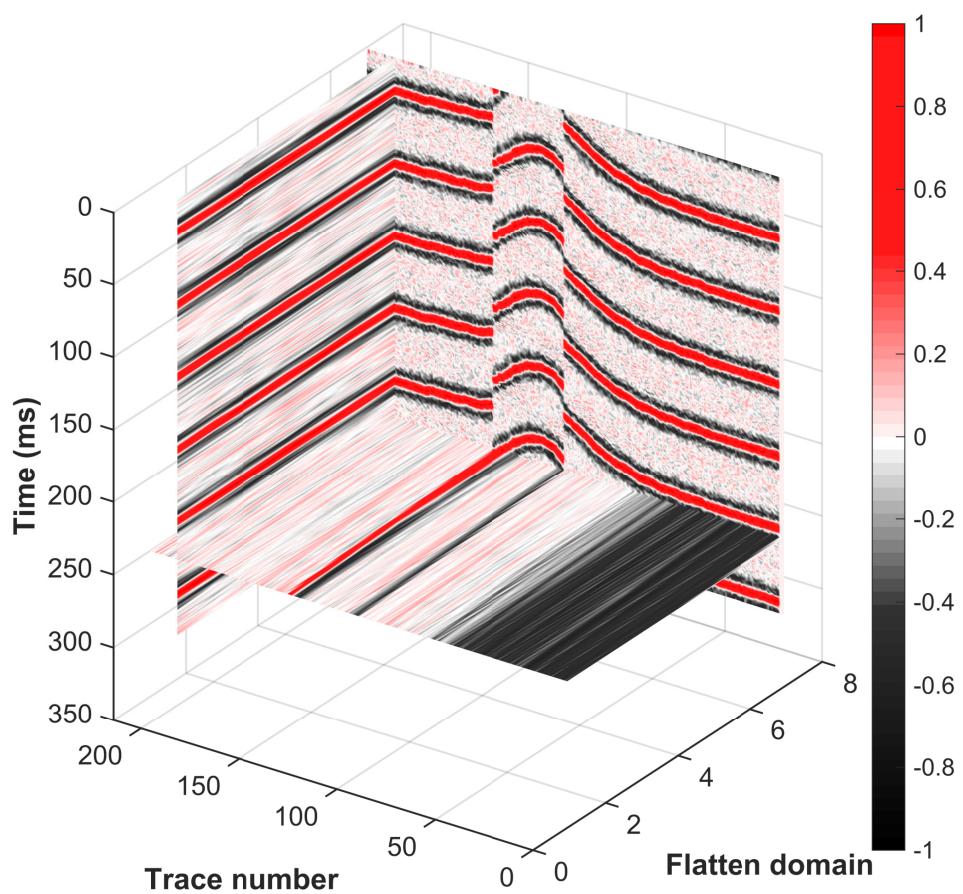


Figure 7: The flattened data of synthetic post-stack 2D example.

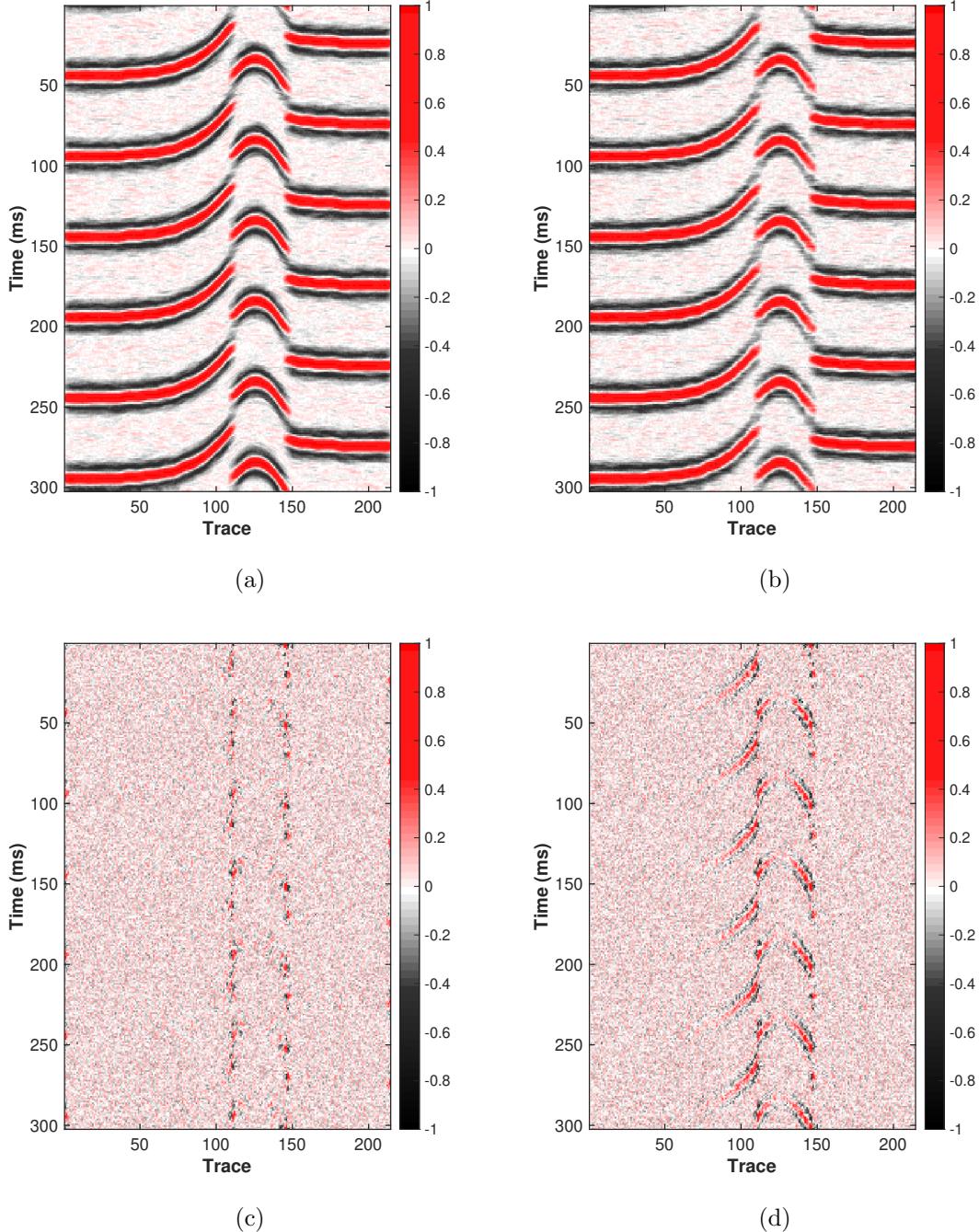


Figure 8: The denoised results and removed noise of synthetic post-stack 2D data. (a) The result by the proposed method ($S/N=14.39\text{dB}$). (b) The result by the conventional smoothing filter without using the slope information ($S/N=12.07\text{dB}$). (c) The removed noise of the proposed method. (d) The removed noise of the conventional smoothing filter.

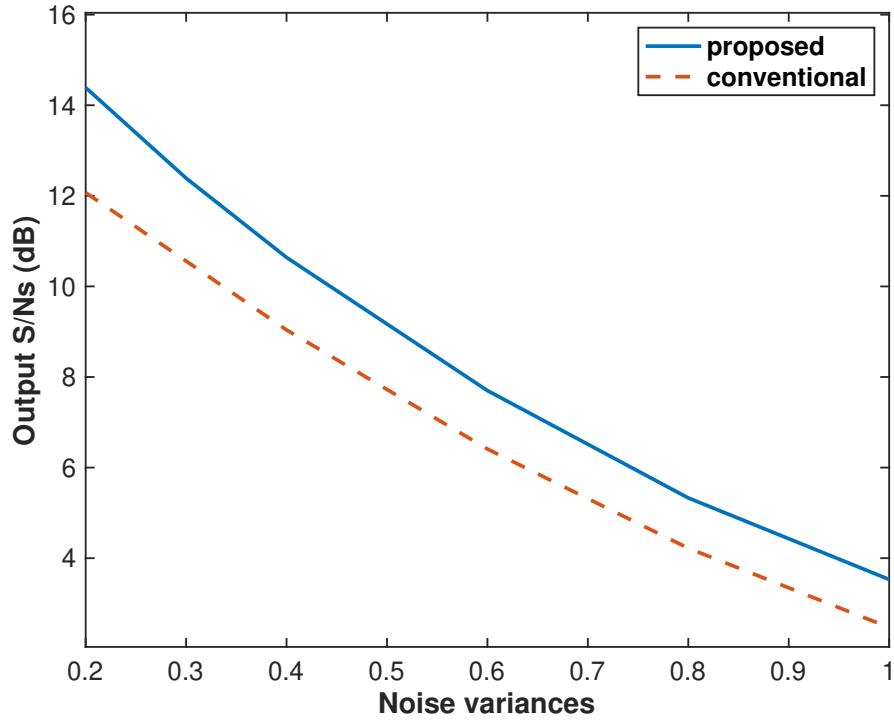


Figure 9: The denoising performances of the conventional (red) and proposed (blue) methods with respect to different noise variances.

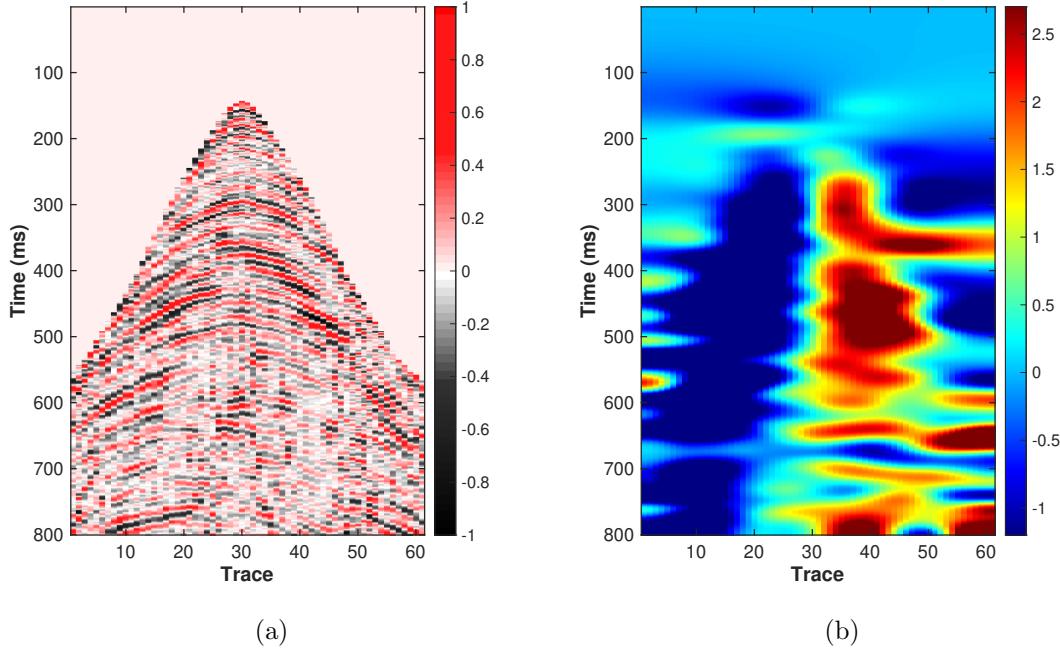


Figure 10: (a) The noisy pre-stack 2D field shot gather. (b) The local slope field of real pre-stack 2D data.

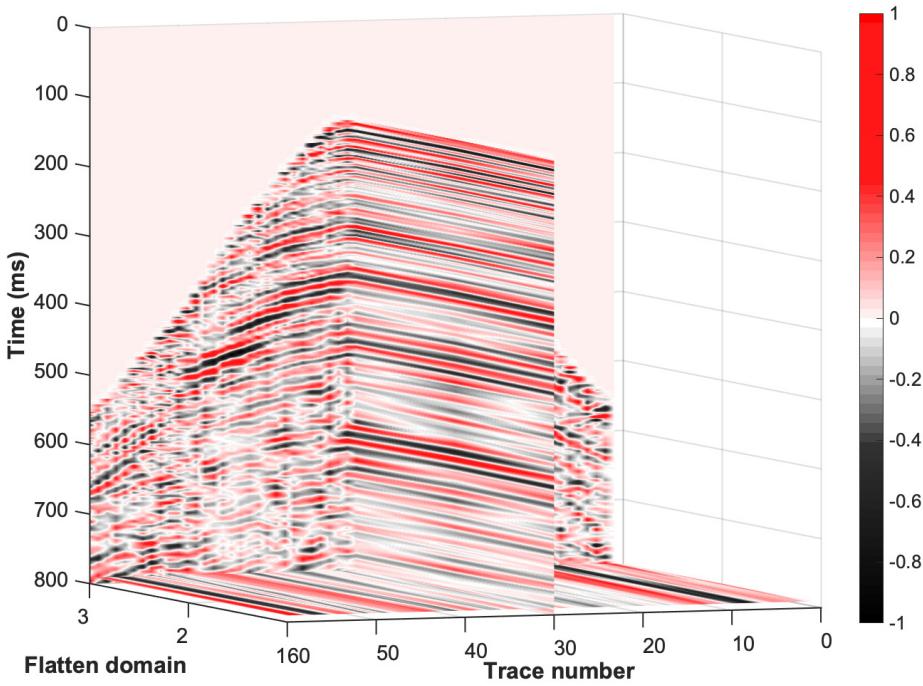


Figure 11: The flattened data of real pre-stack 2D example.

- interpolation: 86th Annual International Meeting, SEG, Expanded Abstracts, 5654.
- Hu, H., Y. Liu, A. Osen, and Y. Zheng, 2015, Compression of local slant stacks by the estimation of multiple local slopes and the matching pursuit decomposition: *Geophysics*, **80**, no. 6, WD175–WD187.
- Huang, G., M. Bai, Q. Zhao, W. Chen, and Y. Chen, 2020, Erratic noise suppression using iterative structure-oriented space-varying median filtering with sparsity constraint: *Geophysical Prospecting*, **69**, no. 1, 101–121.
- Khoshnavaz, M. J., A. Bóna, A. Dzunic, K. Ung, and M. Urosevic, 2016, Oriented prestack time migration using local slopes and predictive painting in the common-source domain for planar reflectors: *Geophysics*, **81**, S409–S418.
- Kim, B., J. Byun, and S. J. Seol, 2017, Efficient Structure-Oriented Filter-Edge Preserving (SOF-EP) Method using the Corner Response: *Geophysics and Geophysical Exploration*, **20**, no. 3, 176–184.
- Li, H., W. Yang, and X. Yong, 2017, A robust local dip estimation method based on plane-wave destruction: 87th Annual International Meeting, SEG, Expanded Abstracts, 1913–1917.
- Lou, Y., B. Zhang, T. Lin, N. Liu, H. Wu, R. Liu, and D. Cao, 2019, Accurate seismic dip and azimuth estimation using semblance dip guided structure tensor analysis: *Geophysics*, **84**, no. 5, O103–O112.
- Porsani, M. J., M. G. Silva, P. E. M. Melo, and B. Ursin, 2010, An adaptive local-slope svd filtering approach to enhance events on seismic sections: 80th Annual

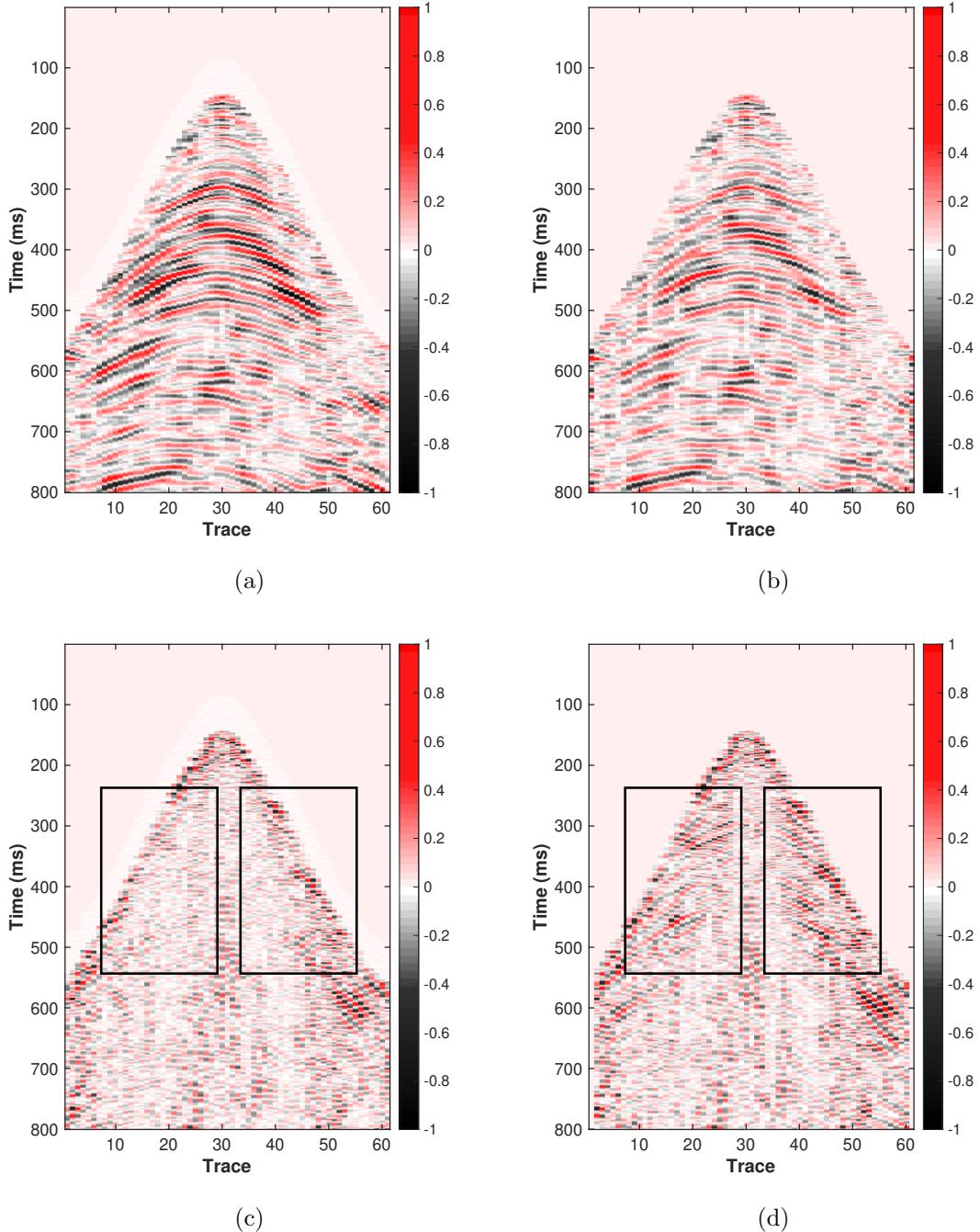


Figure 12: The denoised results and removed noise of real pre-stack 2D data. (a) The result by the proposed method. (b) The result by the conventional smoothing filter without using the slope information. (c) The removed noise of the proposed method. (d) The removed noise of the conventional smoothing filter without using the slope information.

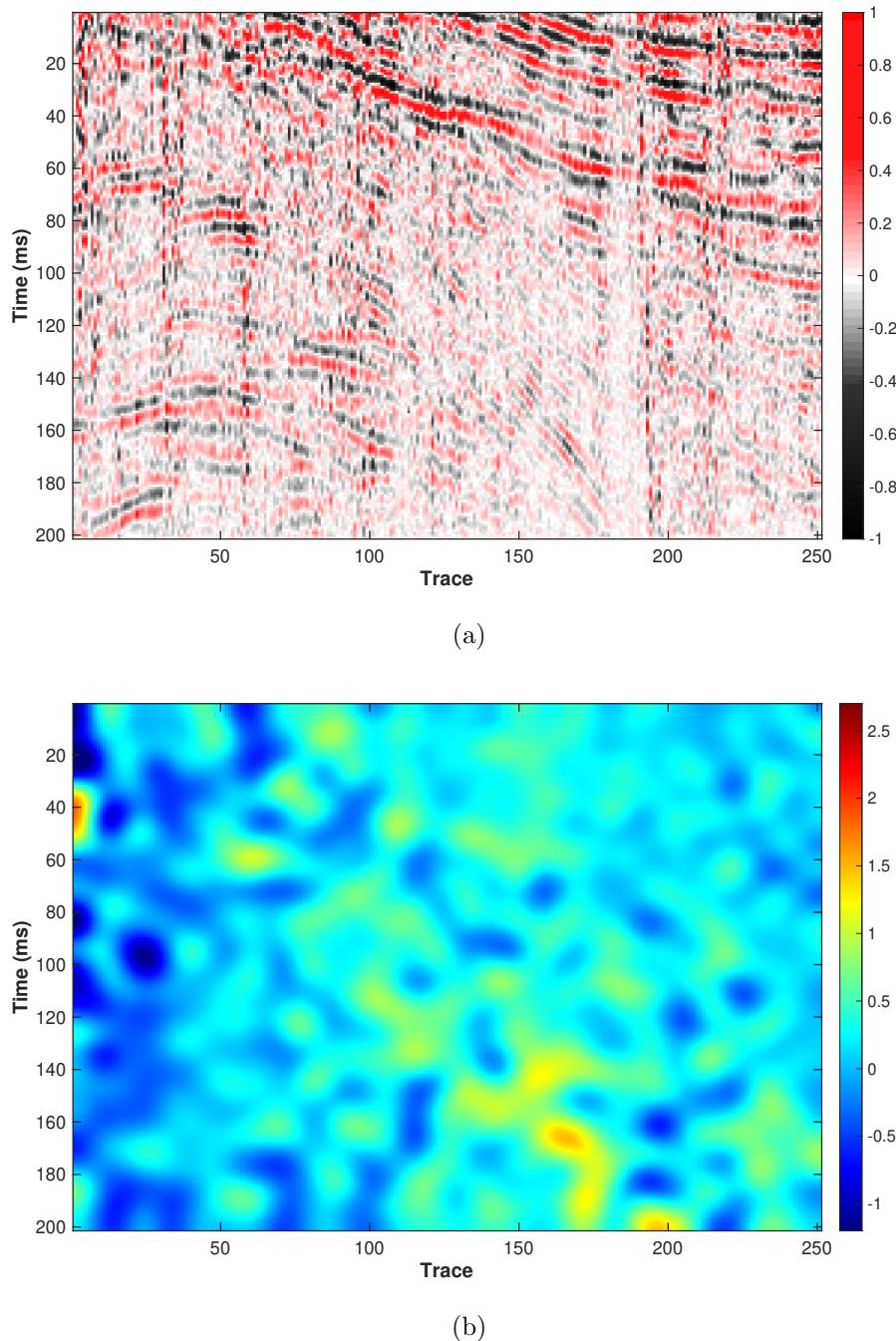


Figure 13: (a) The noisy post-stack 2D field record. (b) The local slope field of real post-stack 2D data.

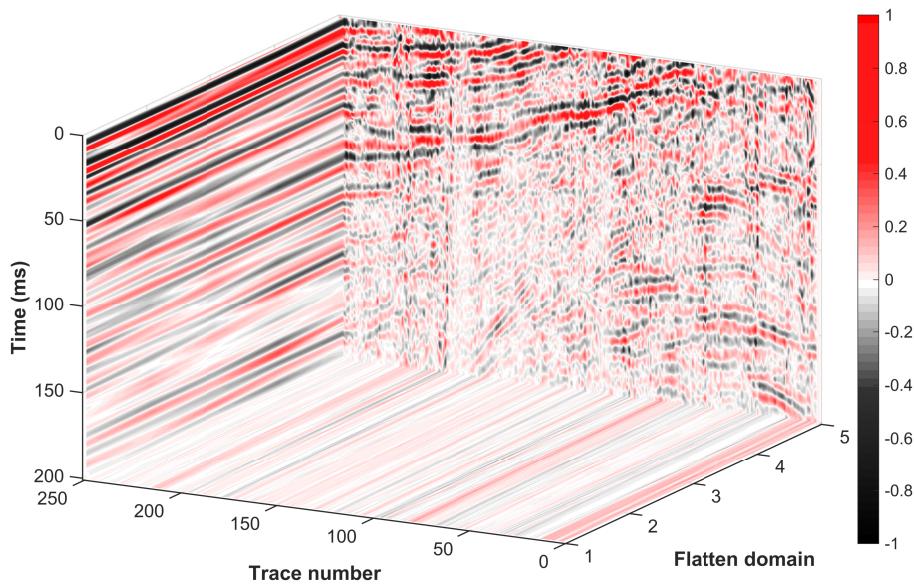


Figure 14: The flattened data of real post-stack 2D example.

- International Meeting, SEG, Expanded Abstracts, 4453.
- Schleicher, J., J. C. Costa, L. T. Santos, A. Novais, and M. Tygel, 2009, On the estimation of local slopes: *Geophysics*, **74**, no. 4, P25–P33.
- Silva, M. G., M. J. Porsani, and B. Ursin, 2016, Automatic data extrapolation to zero offset along local slope: *Geophysics*, **81**, U1–U12.
- Silva, M. G. D., M. J. Porsani, and B. Ursin, 2015, Recursive stack to zero offset along local slope: 85th Annual International Meeting, SEG, Expanded Abstracts.
- Stovas, A., and S. Fomel, 2016, Mapping of moveout attributes using local slopes: *Geophysical Prospecting*, **64**, no. 1, 31–37.
- Swindeman, R., and S. Fomel, 2015, Seismic data interpolation using plane-wave shaping regularization: 85th Annual International Meeting, SEG, Expanded Abstracts, 5634.
- Wang, H., G. Huang, and Y. Chen, 2020, Robust nonstationary local slope estimation: *IEEE Transactions on Geoscience and Remote Sensing*, **PP**, 1–9.
- Wang, Y., W. Lu, and P. Zhang, 2015, An improved coherence algorithm with robust local slope estimation: *Journal of Applied Geophysics*, **114**, 146 – 157.
- Wu, X., L. Liang, Y. Shi, Z. Geng, and S. Fomel, 2019, Deep learning for local seismic image processing: Fault detection, structure-oriented smoothing with edge-preserving, and slope estimation by using a single convolutional neural network: 89th Annual International Meeting, SEG, Expanded Abstracts, 2222–2226.
- Xu, Y., S. Cao, X. Pan, W. Liu, and H. Chen, 2019, Random noise attenuation using a structure-oriented adaptive singular value decomposition: *Acta Geophysica*, **67**, no. 4, 1091–1106.
- Xue, Z., H. Zhang, Y. Zhao, and S. Fomel, 2019, Pattern-guided dip estimation with

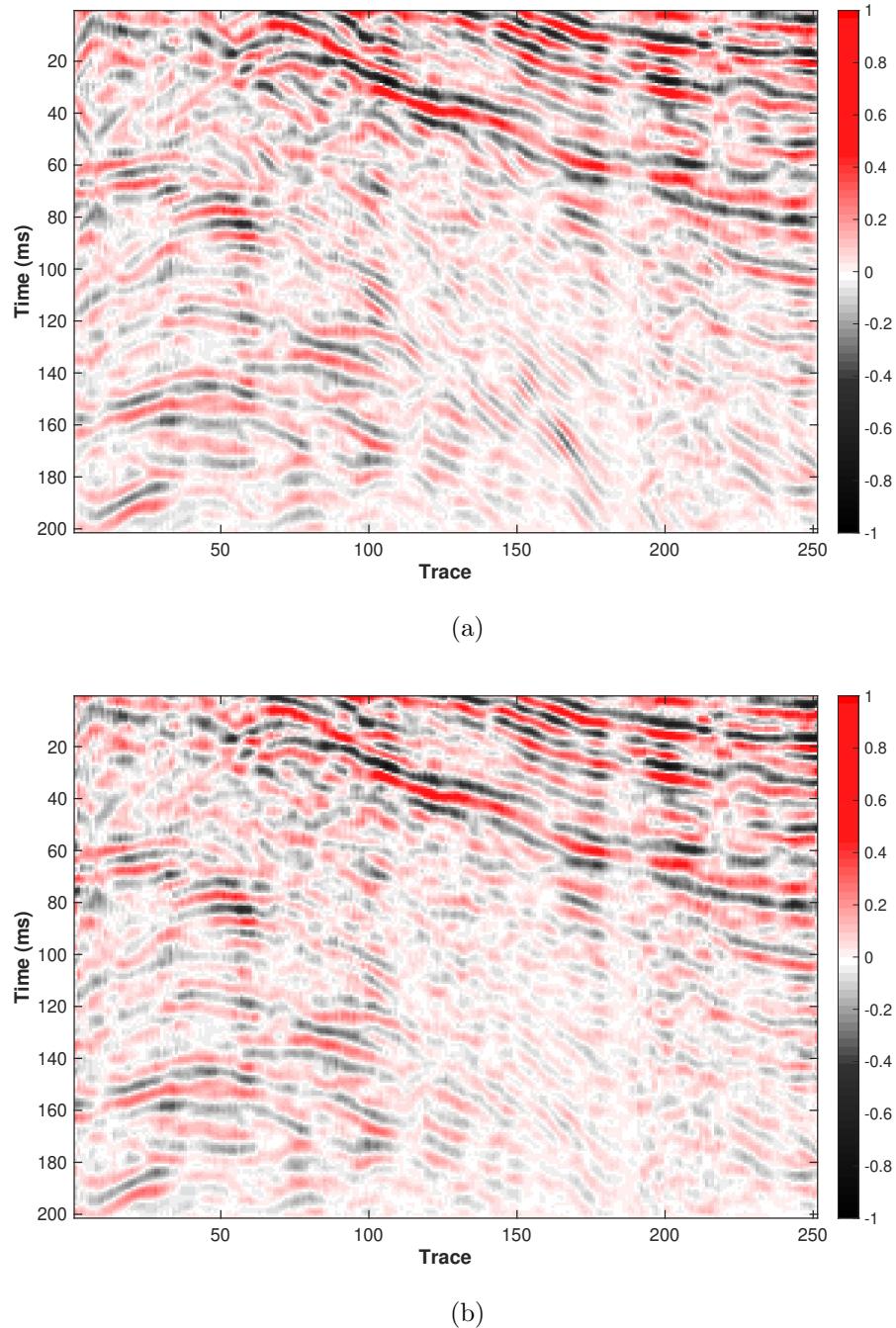
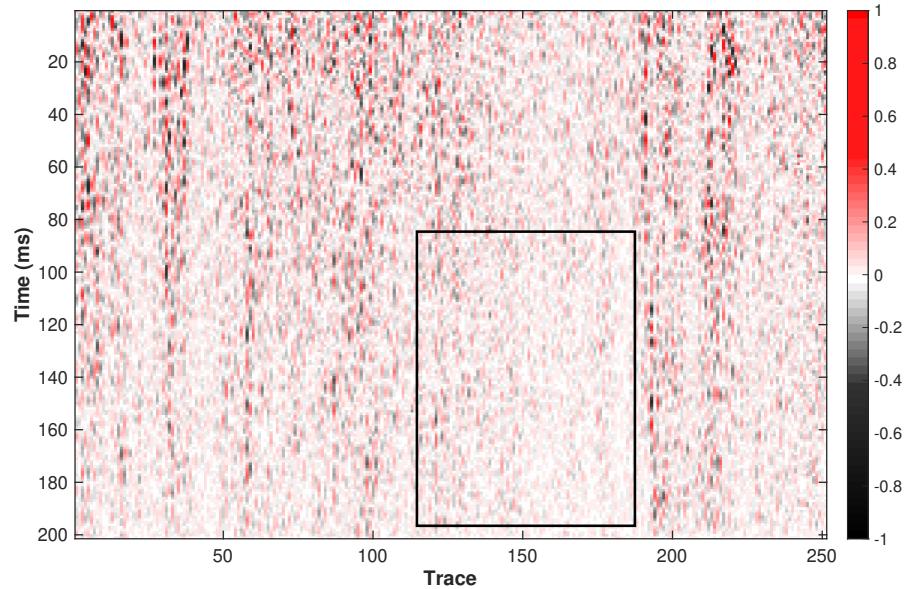
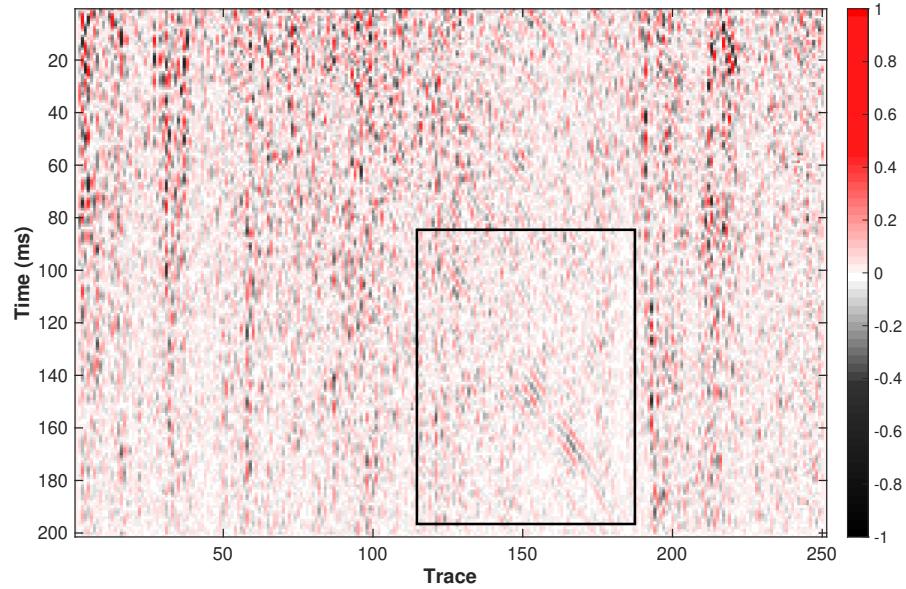


Figure 15: The denoised results of real post-stack 2D data. (a) The result by the proposed method. (b) The result by the conventional smoothing filter without the help of slope information.



(a)



(b)

Figure 16: The removed noise of real post-stack 2D data. (a) The removed noise of the proposed method. (b) The removed noise of the conventional smoothing filter without the help of slope information.

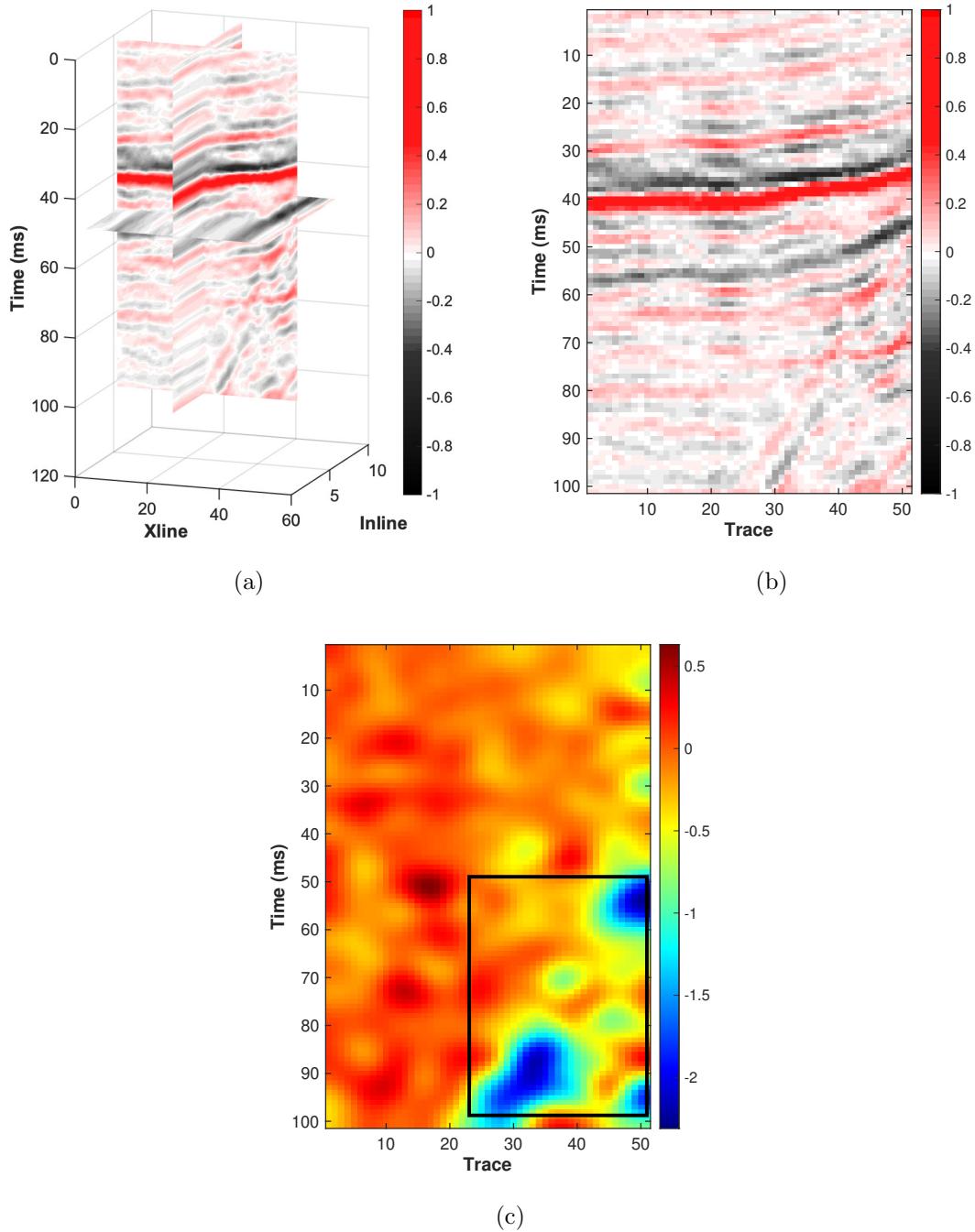


Figure 17: The field 3D data. (a) The 3D plot. (b) The slice in the 5th inline. (c) The local slope field of the 5th inline slice. It is calculated based on the raw data. The black rectangle indicates the location of the large-dip area.

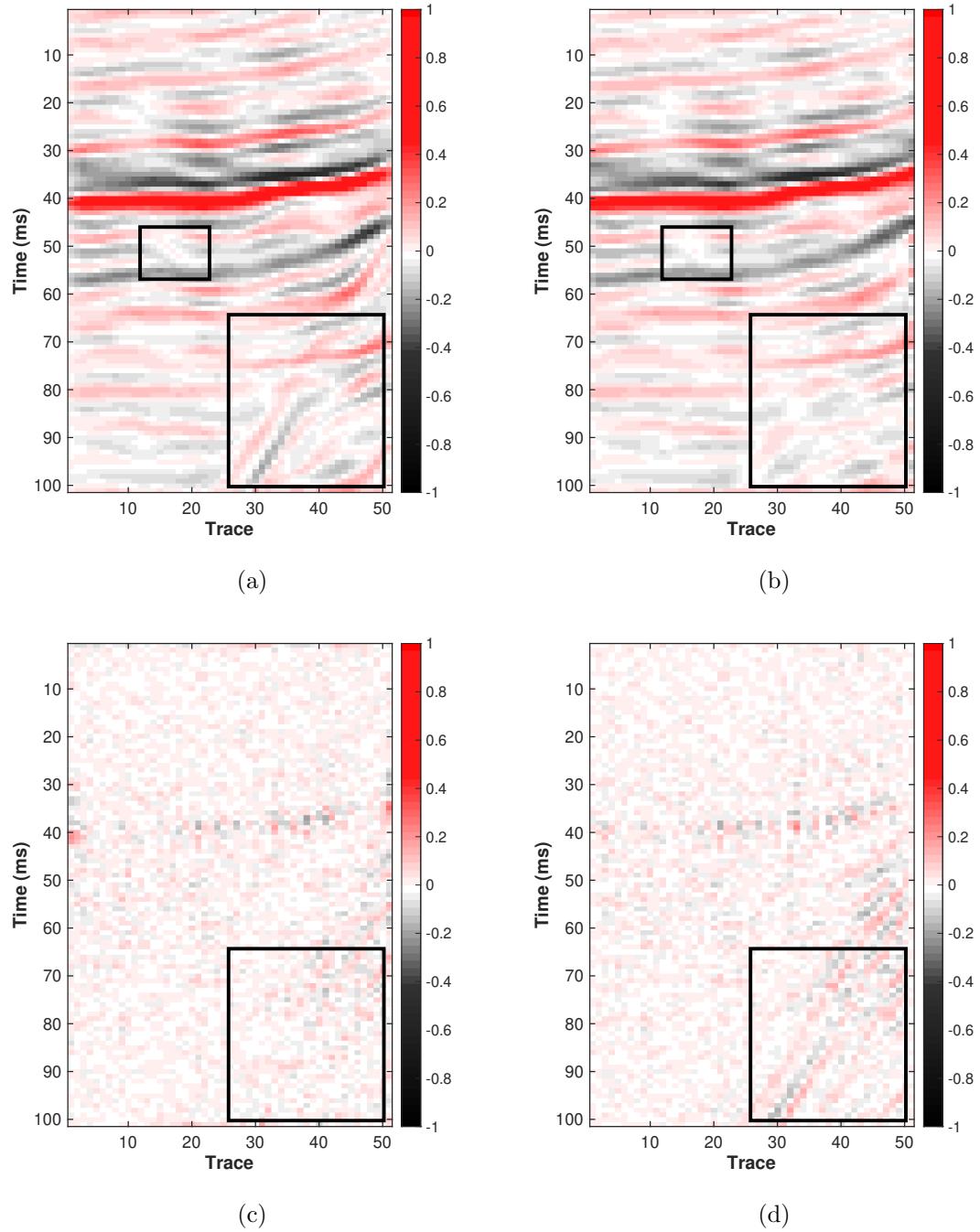


Figure 18: The denoised results and removed noise of the 5th inline slice in field 3D data. (a) The result by the proposed method. (b) The result by the conventional smoothing filter without using slope information. (c) The removed noise of the proposed method. (d) The removed noise of the conventional smoothing filter.

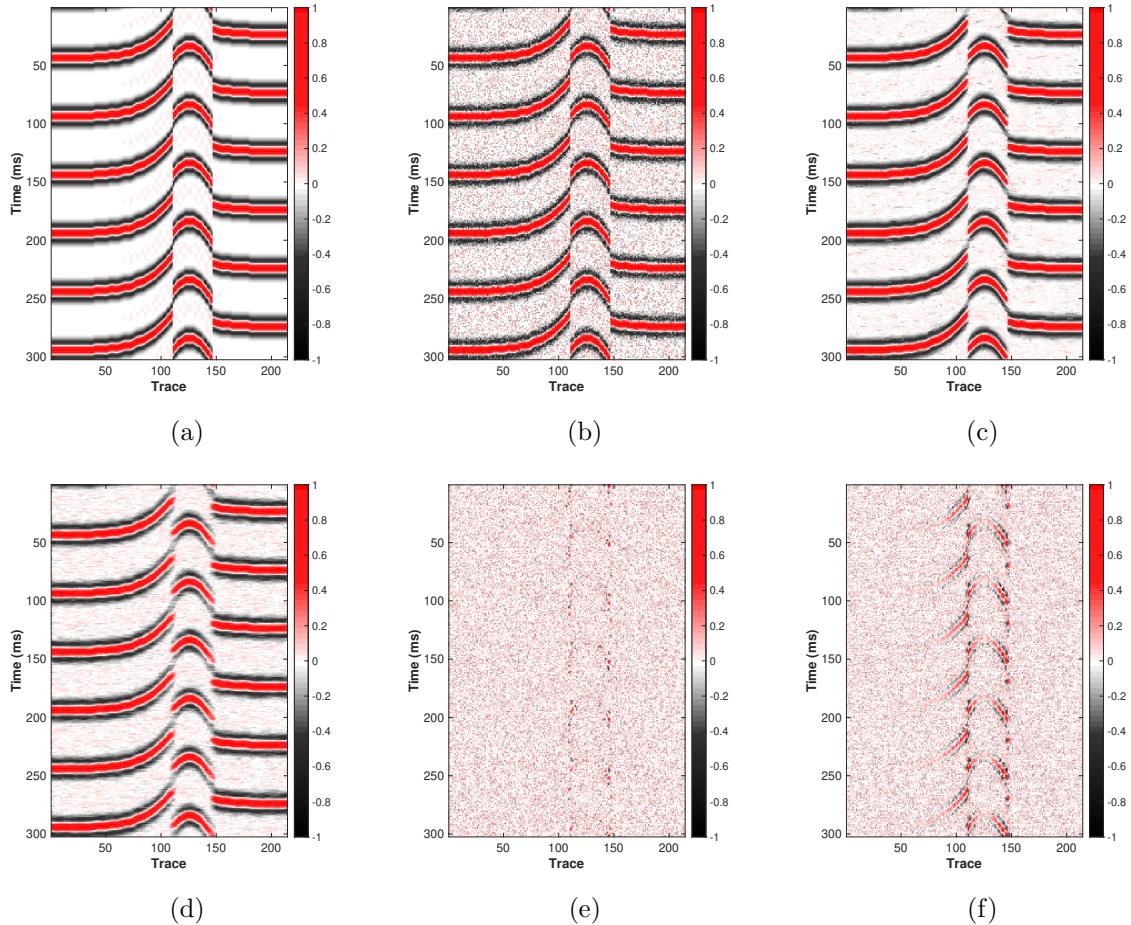


Figure 19: The test for impulse noise. (a) The clean data. (b) The noisy data with impulse noise. (c) The denoised result by the proposed method ($S/N=15.18\text{dB}$). (d) The denoised result by the conventional smoothing filter without using the slope information ($S/N=12.61\text{dB}$). (e) The removed noise of the proposed method. (f) The removed noise of the conventional smoothing filter without using the slope information.

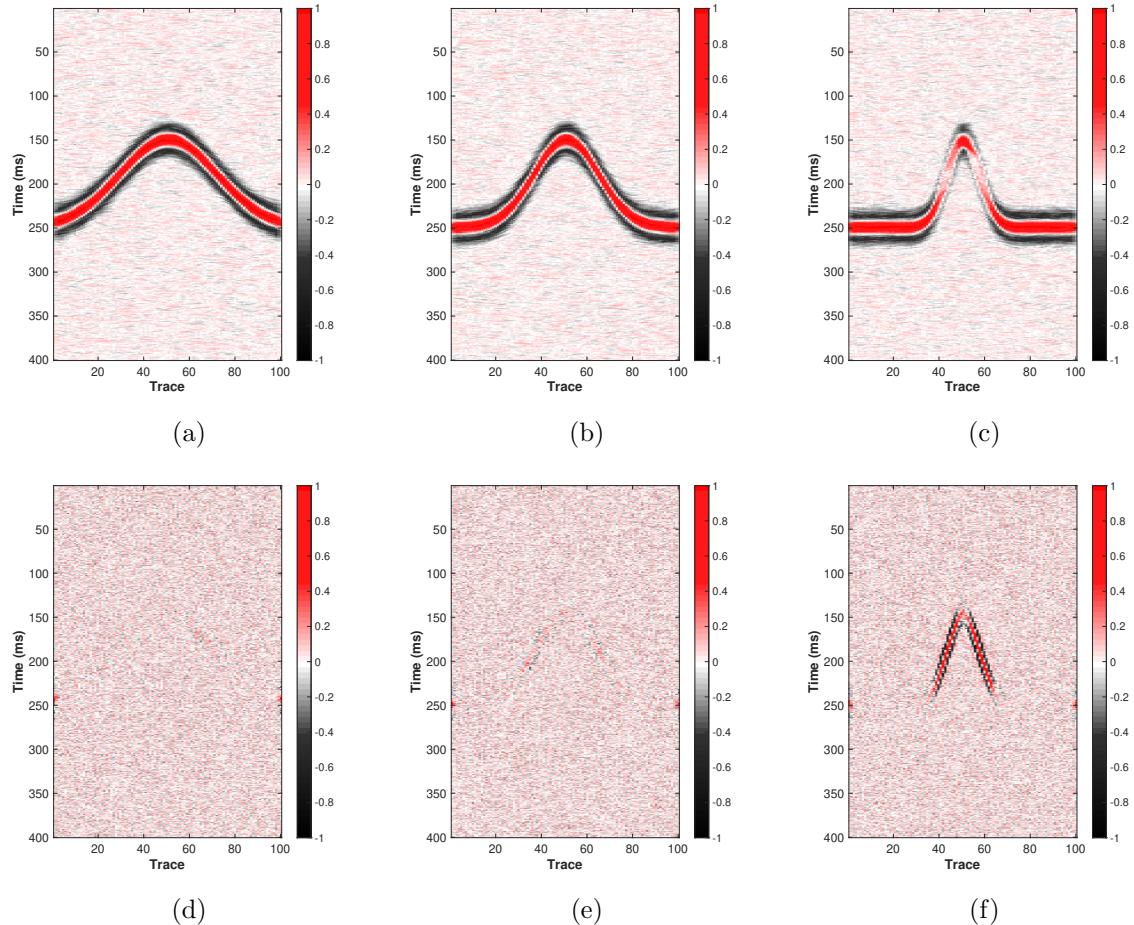


Figure 20: The denoised results with (a) gentle, (b) moderate and (c) steep events, and the removed noise from the data with (d) gentle, (e) moderate and (f) steep events.

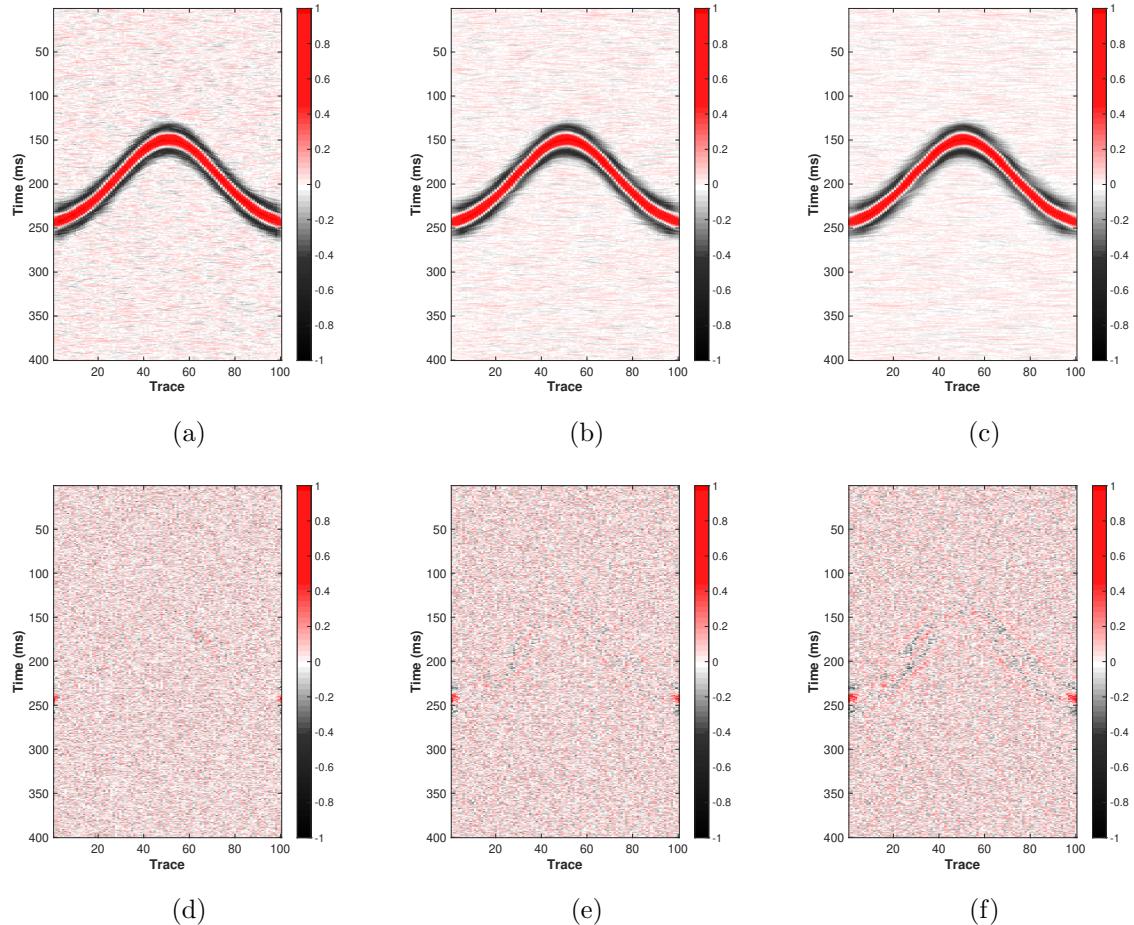


Figure 21: The denoised results by the smoothing lengths (a) $R = 2$, (b) $R = 4$ and (c) $R = 6$, and the removed noise by the smoothing lengths (d) $R = 2$, (e) $R = 4$ and (f) $R = 6$.

- plane-wave destruction filters: *Geophysical Prospecting*, **67**, no. 7, 1798–1810.
- Yang, L., W. Dian, L. Cai, L. Dian-Mi, and Z. Peng, 2014, Structure-oriented filtering and fault detection based on nonstationary similarity: *Chinese Journal of Geophysics-Chinese Edition*, **57**, no. 4, 1177–1187.
- Zhang, B., T. Lin, S. Guo, O. E. Davogustto, and K. J. Marfurt, 2016, Noise suppression of time-migrated gathers using prestack structure-oriented filtering: *Interpretation*, **4**, no. 2, SG19–SG29.
- Zhang, P., W. Lu, and Y. Zhang, 2015, Velocity analysis with local event slopes related probability density function: *Journal of Applied Geophysics*, **123**, 177 – 187.
- Zhou, Y., and J. He, 2019, Flattening the Seismic Data for Optimal Noise Attenuation: *IEEE Geoscience and Remote Sensing Letters*, **16**, no. 3, 487–491.
- Zhou, Y., and S. Li, 2018, Simultaneous deblending and interpolation using structure-oriented filters: *Journal of Applied Geophysics*, **150**, 230–243.