

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

Hang Wang\*, Yunfeng Chen\*, Omar M. Saad\*, Wei Chen<sup>†</sup>, Yapo Abolé Serge Innocent Oboué\*, Liuqing Yang<sup>‡</sup>, Sergey Fomel<sup>§</sup> and Yangkang Chen<sup>§</sup>

\* *Key Laboratory of Geoscience Big Data and Deep Resource of Zhejiang Province*

*School of Earth Sciences, Zhejiang University*

*Hangzhou, Zhejiang, China, 310027*

*18328504171@163.com, yunfeng\_chen@zju.edu.cn, omar.saad@nriag.sci.eg,*

*obouesonofo1@gmail.com*

† *Key Laboratory of Exploration Technology for Oil and Gas Resources of Ministry of Education*

*Yangtze University*

*Wuhan, Hubei, 430100, China*

*chenwei.yangtze@gmail.com*

‡ *State Key Laboratory of Petroleum Resource and Prospecting*

*China University of Petroleum (Beijing)*

*Beijing, China, 102200*

*yangliuqingqin@163.com*

§ *Bureau of Economic Geology*

*The University of Texas at Austin*

*University Station, Box X, Austin, Texas 78713-8924*

*sergey.fomel@beg.utexas.edu, chenyk2016@gmail.com*

*Corresponding authors:* *Yunfeng Chen*, *yunfeng\_chen@zju.edu.cn* & *Yangkang Chen*,

*chenyk2016@gmail.com*

(December 15, 2021)

Running head: **slope and structural filtering**

## ABSTRACT

Local slope is an important attribute that can help distinguish seismic signals from noise. Based on 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 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 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 is an attribute of great importance in the seismic exploration community. Based on their 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 seismic inversion. Among these applications, 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 beam migration by compressing local slant stacks. They apply a new second-order structure tensor to estimate different slope fields at each data point. Based on these slope fields, the raw data can be decomposed into several components. Further, combined with 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 leads to a fast implementation in subsequent processing with almost no quality degradation. Wu et al. (2019) propose a simple convolutional network to simultaneously obtain the slope estimates, locate the faults and smooth noisy seismic data. They synthesize many 3D noisy datasets as the inputs of the network and designate the clean original versions of the datasets as the target outputs. By training these data sets, an effective network architecture can be established and applied to other general seismic data. This network is capable of producing more accurate processing results than 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 volumes. They first use the semblance based method to

estimate an approximate slope field and then, along the slope direction, a local window is designed to include those events with relatively 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 conventional approaches and find that this algorithm is more robust in noisy environments 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 for 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-transform domain filter. Then, the estimation of the local slope field can be treated as a non-linear inverse problem. By minimizing the residual components, the time-varying slope can be estimated iteratively. Schleicher et al. (2009) reduce the computational cost of seismic imaging tasks by utilizing the slope information of the seismic events. They add a correction term to the conventional linear plane-wave destruction method for a more precise slope estimation. Compared with nonlinear plane-wave destruction, the corrected method shows comparable performance. This improved version can produce stable results 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 structures, which is named as omni-directional plane-wave destruction (OPWD). When applied to predictive painting and event picking, the proposed method can produce better results than traditional plane-wave

destruction. Chen et al. (2013a) use a analytical approach to estimate the slope field, which avoids the need for iterative calculations. Compared to the conventional PWD method, the proposed algorithm can achieve similar results with much lower computational cost. 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 vertical structure by a linear iterative optimization framework. Wang et al. (2020) propose a non-stationary slope estimation framework to deal with complex seismic data, using 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 designed to be well suited to seismic data (Fomel, 2002; Chen et al., 2014; Geng et al., 2020), is also based on an accurate slope estimate. 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 mean or median operator applied in a specific local window that is driven by the local slope at the corresponding locations. For definitions and more information on mean and median filters, please refer to Zhou and Li (2018). Zhou and Li (2018) propose an algorithm to simultaneously suppress spike noise and reconstruct missing data, which uses robust slope estimation to assist the mean and median filters. Huang et al. (2020) combine the slope-based structural filter with an inversion frame to iteratively suppress erratic noise.

Compared to conventional median filtering, this proposed frame can handle stronger erratic noise and produce a better result with higher signal-to-noise ratio. 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 when applied to examples containing curved events. Further combined with a shaping regularization framework, this approach can produce much better denoising results than the conventional filters.

Considering the importance of the accuracy of the slope field and structural filtering, we present a useful Matlab code package to implement both operations. For noisy input data, we recommend first preprocessing the data with a conventional denoising method (such as the traditional mean or median filtering) to have a higher relative 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 curved events can be transformed to a flattened domain. The smoothing operator can be further applied along the flattened events to suppress the random noise. Compared to the other signal-processing/denoising code packages, the approach proposed here has significant advantages. First, the package is able to suppress the noise more accurately with the help of the pre-calculated slope field. That is because the slope field can provide necessary time-shift information to flatten the seismic events, which strengthens the coherence of the signal. Thereby, the output preserves signal better and causes less leakage while other methods handling curved events directly do not have such advantage. Second, this package has two individual functions, slope estimation and structural filtering. Although in this paper, the slope estimation serves as a middle step of the final structural filtering, the slope-estimation part can also be used as an individual tool for other purposes (not only for denoising),

such as assisting in the imaging and interpretation. Lastly, many parameters of this code package can be robustly set to default values for most cases. Only a few parameters need to be adjusted by the user. This leads to a straightforward implementation. 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 synthetic and field examples to show its performance on slope estimation and structural filtering.

## METHOD

### Plane wave destruction

The equation describing a 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,  $\sigma$  stands for the local slope,  $t$  and  $x$  represent the time and spatial dimensions. When the local slope  $\sigma$  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:

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

where  $U(w, x)$  and  $F(w)$  are the Fourier transforms of  $u(t, x)$  and  $f(t)$ , respectively. Furthermore, 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 multiplying by 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

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

Next, if we write a convolution between the Z-transform data  $[U(Z_t, 0), \dots, U(Z_t, N)]$  ( $N$  is the trace number) and a plane-wave destruction (PWD) filter  $[1, -Z_t^{-\sigma}]$ :

$$\begin{aligned} & \left[ \begin{array}{c} U(Z_t, 0), \dots, U(Z_t, N) \end{array} \right] \\ & * \\ & \left[ \begin{array}{c} 1, -Z_t^{-\sigma} \end{array} \right] , \quad (7) \\ & \downarrow \\ & \left[ \begin{array}{c} U(Z_t, 0), U(Z_t, 1) - U(Z_t, 0)Z_t^{-\sigma}, \dots, U(Z_t, N) - U(Z_t, N - 1)Z_t^{-\sigma}, U(Z_t, N) \end{array} \right] \end{aligned}$$

where  $*$  stands for the convolution operator. The filter  $[1, -Z_t^{-\sigma}]$  is known as the destruction operator. Based on 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 should tend to **0**.

As convolution in the space domain is equal to multiplication in Z-transform domain,

the aforementioned equation 7 can be rewritten in Z domain as:

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

where symbol  $\times$  represents the multiplication. The above two equations indicate that a plane wave can be well decomposed by the filter  $[1, -Z_t^{-\sigma}]$ . Rewriting it into the form Z transform gives:

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

This resulting filter can be applied to the plane wavefield as:

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

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

To further simplify the calculation, we approximate the linear phase delay operator  $Z_t^{-\sigma}$  with an allpass filter  $\frac{T(Z_t)}{T(1/Z_t)}$  that represents the Taylor approximation of factor  $e^{iw\sigma}$  (Fomel, 2002) and is a function with respect to local slope  $\sigma$ . Accordingly, the above equation can be rewritten as:

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

alternatively,

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

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

Here, we show detailed expressions of the filters  $T(Z_t)$  and  $T(1/Z_t)$  according to the Fomel (2002). Their Taylor expansion with third order accuracy is given by:

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

and

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

The coefficients of filters can be readily written into the time-domain form 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 destruction process in equation 12 can be implemented by convolving the filters with the traces at locations  $x+1$  and  $x$  (i.e.,  $\mathbf{U}_{x+1}$  and  $\mathbf{U}_x$ ).

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

and

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

Finally, the destruction residual  $r$  is obtained by calculating the difference between  $p_1$  and

$p_2$ , and is formulated as:

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

Ideally, the destruction residual should be close to zero, hence the optimal slope  $\sigma$  can be solved by minimizing this residual term.

### Estimating local slope

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

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

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

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

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

Equation 21 is solved with the shaping regularization framework (Fomel, 2007):

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

where  $\Delta\boldsymbol{\sigma}_n^m$  represents the updated value corresponding to the  $m_{th}$  linear iteration and  $\mathbf{S}$

is the shaping operator. Finally, the converged model is formulated as:

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

where  $\mathbf{H}$  is a triangle smoothing operator with smoothing radii of  $L_x$  and  $L_y$  (and  $L_z$  for 3D case) that links to the shaping operator  $\mathbf{S}$  via  $\mathbf{S} = \mathbf{H}\mathbf{H}^T$ .  $\lambda$  is a parameter controlling the relative scaling of the forward operator  $\mathbf{G}$ .

## Structural filtering

With an accurate slope estimate in the above subsection, one can flatten and smooth the noisy seismic data. The flattening operation is implemented in a local window to 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 integer  $l$  ( $l \in [-R, R]$ ) denotes the local flattened domain, and  $R$  is the half length of the local window. For a certain trace  $u(:, x)$ , its corresponding flattened gather is defined as  $u(:, x, :)$ , where the symbol  $:$  denotes all elements in this dimension. Note that the original trace  $u(:, x)$  is at the center of the local flattened gather corresponding to  $l = 0$ , i.e.,  $u(:, x, 0)$ . Each trace in 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  $\mathbf{u}'_x$  denotes the original 2D data  $u(:, x + l)$ . The operator  $\mathbf{F}_{i,j}$  predicts  $\mathbf{u}'_j$  from  $\mathbf{u}'_i$  and is designed according to the estimated slope value. Equations 24 and 25 denote the predictions from the right and left sides, respectively.

After getting the flattened data, a smoothing filter can be applied along the structural direction in the flattened domain. The filtered flattened gather  $u(:, x, :)$  is then averaged to obtain a smoothed trace that gives the final denoised result  $\hat{u}(:, x)$ . Compared to conventional smoothing filter, the proposed method makes full use of the slope information and is capable of minimizing the signal damage.

## CODE DESCRIPTION

### Code for 2D data

The main script ‘test\_2D\_SOF.m’ has three parts including generating the test data, local slope estimation and structural filtering. Note that one can replace the data generation step with any existing synthetic or field data 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 generates a synthetic wavelet. The input parameters are the dominate frequency  $f$ , the time interval  $dt$  and the duration  $tlength$ . The output is a discrete sequence of wavelet  $w$ . We design an anticline model shown in Figure 24a and is stored in variable  $t$ . By convolving  $w$  and  $t$ , we get the synthetic data saved in variable  $data$ . Then we add the random noise and obtain the noisy data  $dn$ .

#### 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 include the 2D seismic data  $din$ , the non-linear iteration number  $niter$ , the linear iteration number  $liter$ , the PWD accuracy  $order$  and the size of triangle smoothing operator  $rect$ . Other inputs are the regularization ( $eps\_dv, eps\_cg$ ) and tolerance ( $tol\_cg$ ) parameters used by the iterative optimization algorithm. 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 equation 21. The destruction operators in equations 15 and 16 are calculated by subfunctions ‘B3.m’ for third-order accuracy or ‘B5.m’ for fifth-order accuracy. Then, subfunction ‘str\_divne.m’ utilizes these operators to update value  $\Delta\sigma$  of slope field according to equations 22 and 23. The iterative shaping regularization method is implemented by subfunction ‘str\_conjgrad.m’.

### Part III: Flattening the events and smoothing the data

The function ‘str\_pwsSmooth\_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 to flatten all slanted events. The user can choose a parameter to properly controls the length  $R$  of the smoothing operator. A larger length leads to a stronger smoothing effect. The function 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 ‘str\_pwspray\_lop2d.m’ takes the same input parameters and outputs the flattened data with an 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. In addition, the subfunction ‘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 has a similar code structure as that of the 2D case except for the first part where it directly loads the prepared field data instead of generating the synthetic data. This script calls all functions in the ‘code3D’ folder. The descriptions of the main script and its related subfunctions are detailed below and the folder structure is shown in Figure 2.

### Part I: Loading the 3D data

This part simply loads the field 3D data ‘real3d.bin’ with complex geological structures and then plots the data.

### Part II: Estimating the local slope

We use the function ‘str\_dip3d.m’ to calculate the slope field of the 3D data with exactly the same parameters as those of 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  $rect$ , etc. Note that the  $rect$  has three elements for this three dimension case. The output differs from that of the 2D estimation function in that two slope fields are obtained, 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’. The two functions are responsible for the inline and crossline directions, respectively. The subfunction ‘str\_divne.m’ is the same as the 2D case.

### Part III: Flattening the events and smoothing the data

The function ‘str\_pwsMOOTH\_lop3d.m’ is used to flatten and smooth the 3D data. The

inputs of the function include two slope fields and their corresponding smoothing lengths in the inline and crossline directions. Note that the subfunction ‘str\_pwspray\_lop3d.m’ is invoked to create 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 third-order accuracy to calculate the slope. Figure 3a is the pre-stack clean data with a size of  $256 \times 128$ . The sampling interval is 1 ms. The spatial interval used here is 50m. We add random noise with a variance of 0.2 to generate the noisy data (Figure 3b). We use the traditional mean filter as the preprocessing method, which outputs a preliminary denoised result for subsequent slope estimation. The estimated slope field of the preconditioned data indicates the right event dips (Figure 3c). According to the estimated slope field, we can flatten all traces and get the flattened data (Figure 4). The events are well aligned in all traces, which indicates the effectiveness of the flattening operation. Figures 5a-5d show comparisons between the structural filter and the traditional mean 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 well preserves the energy of the events. 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 noise panel also confirms the superiority of the proposed code package because its noise section shows significantly less signal leakage (Figure 5c).

Figure 6a is a post-stack clean data with complex structures (anticline and faults). Its

size is  $302 \times 214$ . The sampling interval is 1ms and the spatial interval is 20m. Figure 6b is the corresponding noisy data with a noise variance of 0.2. The estimated slope based on the preprocessed data is shown in Figure 6c, which correctly captures the shapes of curved 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. Although both of them slightly damage signals in the faulting areas, the plain smoothing filter causes more apparent 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. The comparison shows that performances of both methods degrade with the increase of the noise energy, and the proposed method always outperforms the conventional one.

The performance of the proposed package is further examined with a real data set, which contains a shot gather with a dimension of  $800 \times 61$  and is contaminated with random noise (Figure 10a). The sampling interval is 1 ms and spatial interval of it is 50m. 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. The comparison shows that the

structural filter is capable of better preserving the seismic energy and minimizing the signal damage. Additionally, the leaked energy in the removed noise section from the proposed method (Figure 12c) is much less than that from the conventional filter in Figure 12d (see the rectangles).

The second field data example is a post-stack data set with a size of  $201 \times 251$  (Figure 13a). Its time and spatial intervals are 1ms and 25m, respectively. 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 preserves more details in the signal than the simple smoothing filter, especially in the right flank of the anticline. The smoothing length used 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 final test is conducted with a 3D real data set taken from Chen et al. (2016), which has a size of  $101 \times 51 \times 10$  and a sampling interval of 1 ms. The spatial intervals in X and Y directions are 25m. 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 that 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 details of the denoised and removed noise sections, we extract the corresponding slice of the 5th inline in Figures 18a-18d. The denoising result from the structural filter (Figure 18a) clearly better preserves the signal than the traditional mean filter does in the area of complex structures indicated by the black rectangles (Figure 18b).

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

Aside from the random noise, erratic energy in seismic data also imposes difficulty in processing workflows. We test the potential application of the proposed method on data with impulse noise. The clean and noisy data are shown in Figures 19a and 19b. Figures 19c and 19d show the filtered sections by the proposed and conventional methods. Considering the characteristics of the impulse noise, we use a median filter instead of a smoothing filter to process the noisy data in both proposed and conventional methods. The test results show that the proposed method can still effectively suppress the impulse noise and obtain stronger events than the conventional one. The difference sections in Figures 19e and 19f also demonstrate that the proposed framework can better preserve complex structures.

To test the performance of the proposed code on even more complex structures, we use a data example containing the diffractions with conflicting dips (see the black arrows in Figure 20a) as the input of the package. Figures 20b and 20c are the denoised result and the removed noise, respectively. Some signal leakage are observed near the conflicting areas. That is because there are two dips in these areas, i.e., the dominant (relative strong and flat events) and the non-dominant (relative weak and steep events) dips. The slope estimation can only output the dominant dips due to the interference, which in turn degrades the performance of the structural filtering, as evidenced by the damage of the signal with non-dominant dips (see the black arrows in Figure 20c). Here, we propose a strategy (additional step) to alleviate this phenomenon. After obtaining the filtered signal and its noise section,

we apply the package again to the removed noise (Figure 20c) to extract leaked signals with non-dominant dips. Then, the extracted energy is added back to the original denoised result (Figure 20b) to obtain an updated image where the signal leakage phenomenon is effectively mitigated (Figures 21a and 21b).

To compare the performance of the proposed method with another commonly used coherency filter (an F-K method), we implement a denoising experiment using these two approaches. Figure 22a is the same post-stack data used in the above example. Figures 22b and 22c are the denoised results from the F-K and the proposed method, respectively. Note that the F-K method is implemented in a moving window with a size of  $32 \times 32$  and 7-points sliding step in both X and Y directions. The filtering result from F-K method shows fewer structure details compared to the proposed one, and still contains strong residual noise in the upper left corner. That is because, even within a local window, seismic data may still contain events with a variety of dips, some of which belong to dominant events with similar slopes whereas other (secondary) events have inconsistent slope information. Under this circumstance, the F-K can wrongly filter out some structure details and tend to only output the events with dominant dips due to its insufficiency of distinguishing complex dip variation with only a simple threshold parameter in the F-K domain. This phenomenon is illustrated in Figures 22a-22c (see the red rectangles), where events with positive slopes (upper-left to lower-right) are output with clear and strong amplitudes while fine-scale structures with the negative slopes (lower-left to upper-right) are weak or completely filtered out. In comparison, the proposed method is a totally local method (point-by-point) which does not involve a window operation, thus it can deal with very detailed spatial change of event slopes and cause much less damages to small-scale features (see red rectangles in Figures 22c). The removed noise in Figures 23a and 23b also shows the difference of leaked

structure details (see the red rectangle). It is also worth noting that the proposed method can better suppress the near vertical noise compared with the F-K filter (see blue rectangles in Figures 23a and 23b), which suggests the advantage of the slope-based structural filter over conventional approach when dealing with the non-random noise.

We further test the influence of the dip angle on the final denoised result. We keep the other parameters unchanged and only vary the dip angle of the synthetic event. Figures 24a-24c correspond to the denoised results from different data sets with a gradually increasing slope. The smoothing length is  $R = 2$ . Our test results show that the denoising performance degrades with an increase in structural dip (Figures 24a-24c) and more signal leakage is observed in the residual noise panels (Figure 24d-24f).

Additionally, we discuss the influence of the smoothing length  $R$ . For the same original noisy data presented in Figure 24a, we set  $R$  to 2,4,6 and assess the denoising performances of the proposed method. Figures 25a-25c show the denoised results, and Figures 25d-25f show the corresponding noise sections. The test results show that a larger smoothing length is beneficial to suppress random noise more effectively and output a cleaner image at the cost of producing more signal leakage (see the obvious leakage in the middle of Figure 25f).

For the convenience of users, we give recommended values of input parameters used in the slope estimation and structural filtering parts. Table 1 shows the details of these recommendations. In the slope estimation part, the parameters  $niter$ ,  $liter$ ,  $eps\_dv$ ,  $eps\_cg$ ,  $tol\_cg$  are used in the optimization algorithm for solving the slope field and can be set to the default values for most cases. The PWD filter order  $order$  has two options, 1 for third-order accuracy and 2 for fifth-order accuracy. The latter one is more accurate but is more computationally expensive. The size of triangle smoothing operator  $rect$  controls the

resolution and smoothness of the estimated slope field, i.e., a larger value corresponds to a low-resolution but smoother estimation whereas a smaller value results in high resolution but weak anti-noise ability. Generally, a value between 5-10 in each dimension is suitable for a common noisy data. As for the smoothing length  $ns$  in the structural filtering part, it has been tested in the above paragraph. The user can select the proper value according to the data quality. Same as the slope estimation part, the regularization parameter  $eps$  can also be set to the default value for most cases.

## CONCLUSION

We present an open-source Matlab code package for slope estimation and structural filtering, which can be used to process 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 2D and 3D cases have three main processing modules including input data preparation, slope field calculation, and structural filtering. Compared with the conventional mean filtering, the presented code package can effectively suppress the random noise while minimizing the signal leakage. The superior performance is mainly due to the use of the pre-calculated local slope field that helps flatten the slanted events, and thus, the smoothing filter can be applied along the structure direction. The synthetic and field data with slanted events demonstrate its superiority over the conventional method. We also discuss the impacts of some important parameters on the final smoothing results, which provides a quick guideline 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.

## **DATA AND MATERIALS AVAILABILITY**

Data associated with this research are available and can be obtained by contacting the corresponding author.

## **ACKNOWLEDGEMENTS**

The research is partially supported by the starting funds from Zhejiang University, sponsors of the Texas Consortium of Computational Seismology, research funding granted to Texas Seismological Network Project, the Open Fund of Key Laboratory of Exploration Technologies for Oil and Gas Resources (Yangtze University), Ministry of Education (Grant No. PI2021-02), and the National Natural Science Foundation of China (Grant No. 41804140).

## 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: *GEOPHYSICS*, **78**, no. 1, V1–V9.
- , 2013b, Omnidirectional plane-wave destruction: *Geophysics*, **78**, no. 5, V171–V179.
- 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 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 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 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.

## **LIST OF TABLES**

- 1 Free parameters and their recommended values.

## LIST OF FIGURES

- 1 The code structure of 2D slope estimation and structural filtering.
- 2 The code structure of 3D slope estimation and structural filtering.
- 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.
- 4 The flattened data of synthetic pre-stack 2D example.
- 5 The denoised results and removed noise of pre-stack synthetic 2D data. (a) The result by the proposed method ( $S/N=16.98dB$ ). (b) The result by the conventional smoothing filter without using the slope information ( $S/N=11.81dB$ ). (c) The removed noise of the proposed method. (d) The removed noise of the conventional smoothing filter.
- 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.
- 7 The flattened data of synthetic post-stack 2D example.
- 8 The denoised results and removed noise of synthetic post-stack 2D data. (a) The result by the proposed method ( $S/N=14.39dB$ ). (b) The result by the conventional smoothing filter without using the slope information ( $S/N=12.07dB$ ). (c) The removed noise of the proposed method. (d) The removed noise of the conventional smoothing filter.
- 9 The denoising performances of the conventional (red) and proposed (blue) methods with respect to different noise variances.
- 10 (a) The noisy pre-stack 2D field shot gather. (b) The local slope field of real pre-stack 2D data.
- 11 The flattened data of real pre-stack 2D example.

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.

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

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

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.

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.

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.

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.

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 ( $S/N=15.18dB$ ). (d) The denoised result by the conventional median filter without using the slope information ( $S/N=12.61dB$ ). (e) The removed noise of the proposed method. (f) The removed noise of the conventional

median filter without using the slope information.

20 The stacked data with conflicting dips. (a) Original noisy data. (b) The denoised result. (c) The removed noise. Note the signal leakage from the events with non-dominant dips in the conflicting areas.

21 The results after the additional step that applies the package to the removed noise from the first step to extract leaked signals with non-dominant dips. Then, the extracted signals are added back to the denoised result of the first step. (a) The denoised result after additional step. (b) The removed noise after additional step.

22 The denoising comparison of F-K and the proposed method. (a) The original noisy data. (b) The denoised result of F-K. (c) The denoised result of the proposed method. There is obvious residual noise in the F-K result (blue rectangle), and many of the structure details are harmed (red rectangles).

23 The removed noise of both F-K and the proposed methods. (a) The removed noise of F-K. (b) The removed noise of the proposed method. We can find more structure details in the former one (red rectangle).

24 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.

25 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)  $R = 6$ .

	parameter	default (recommended) value
slope estimation	niter	5-10
	liter	10-20
	order	1 (three-order accuracy) or 2 (five-order accuracy)
	eps_dv	0.01
	eps_cg	1
	tol_cg	0.000001
	rect	5-10 (suitable for most cases) larger—low resolution but smoother smaller—high resolution but more noisy
	verb	1
structural filtering	ns (r1, r2 in 3D case)	1-4 (depends on noise level) larger—less residual noise but more leakage smaller—more residual noise but less leakage
	order	1 (three-order accuracy) or 2 (five-order accuracy)
	eps	0.01

Table 1: Free parameters and their recommended values.

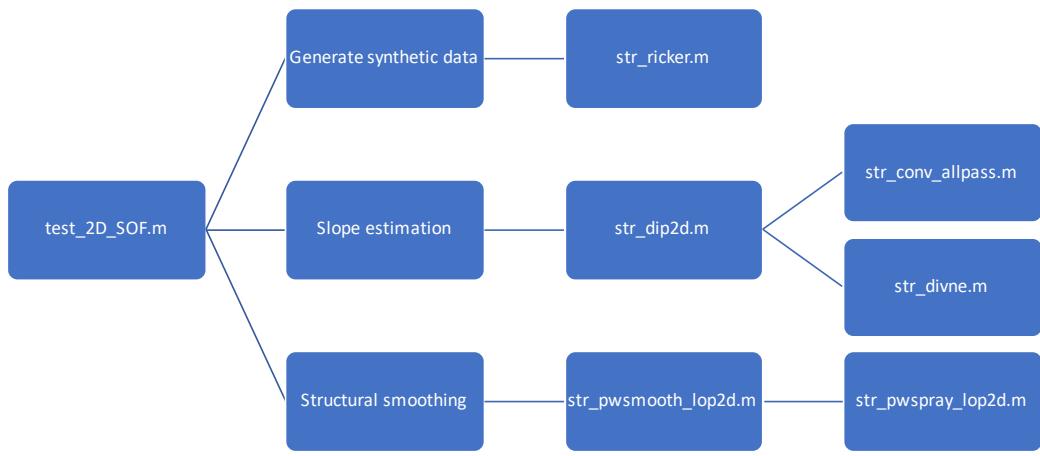


Figure 1: The code structure of 2D slope estimation and structural filtering.

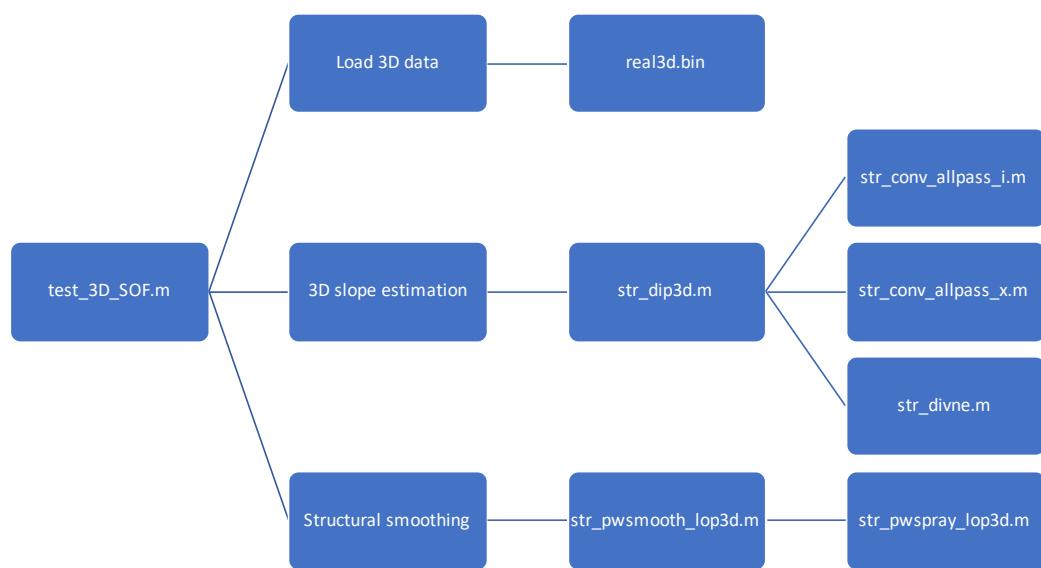


Figure 2: The code structure of 3D slope estimation and structural filtering.

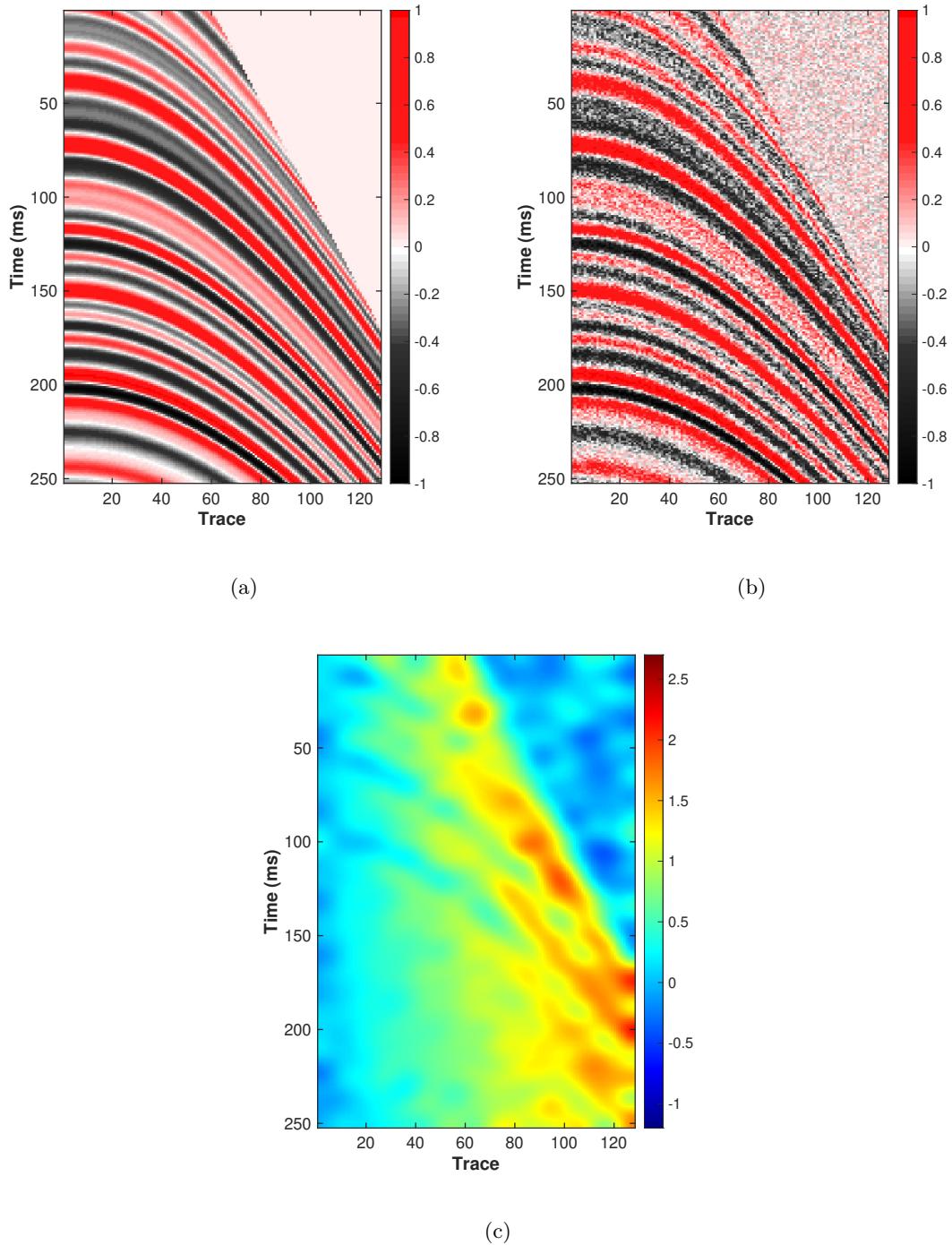


Figure 3: The synthetic pre-stack 2D data. (a) The clean data. (b) The noisy data with random noise (variance=0.2). (c) The local slope field of synthetic pre-stack 2D data. Note that it is calculated based on the preprocessed data.

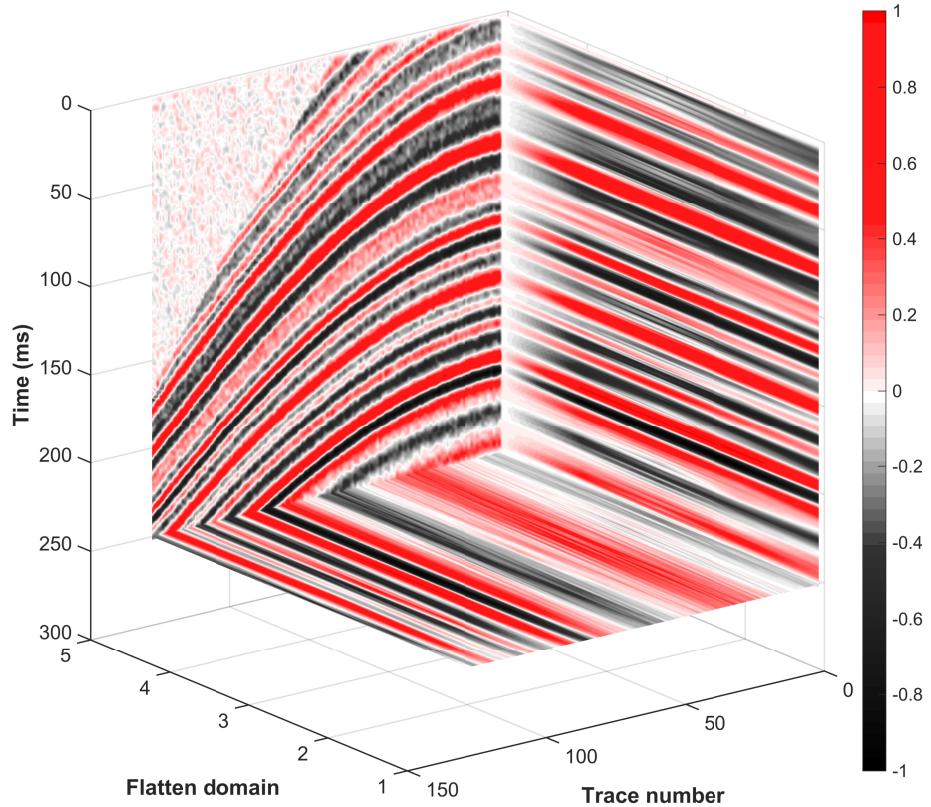


Figure 4: The flattened data of synthetic pre-stack 2D example.

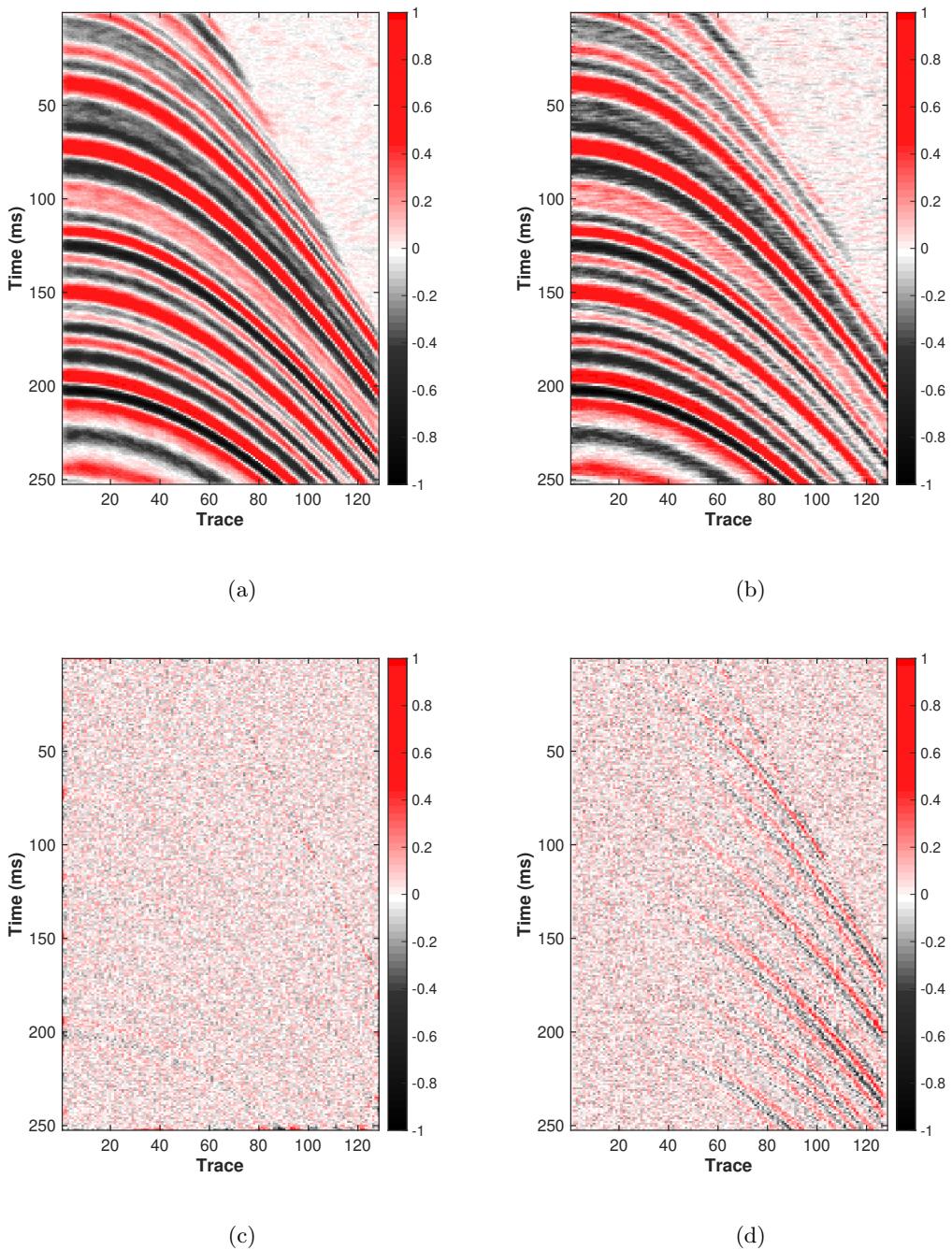


Figure 5: The denoised results and removed noise of pre-stack synthetic 2D data. (a) The result by the proposed method ( $S/N=16.98\text{dB}$ ). (b) The result by the conventional smoothing filter without using the slope information ( $S/N=11.81\text{dB}$ ). (c) The removed noise of the proposed method.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

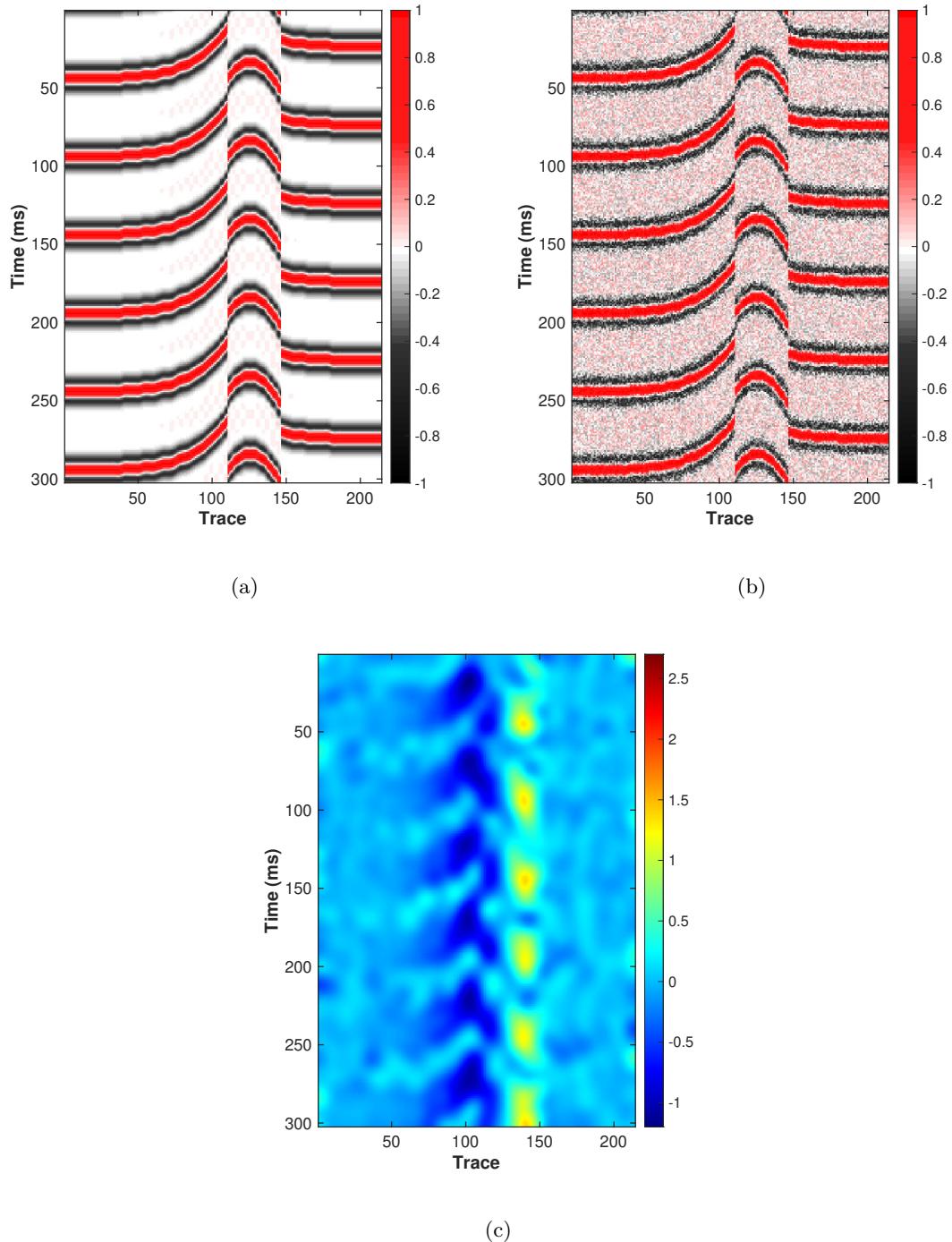


Figure 6: The synthetic post-stack 2D data. (a) The clean data. (b) The noisy data with random noise (variance=0.2). (c) The local slope field of synthetic post-stack 2D data. Note that it is calculated based on the preprocessed data.

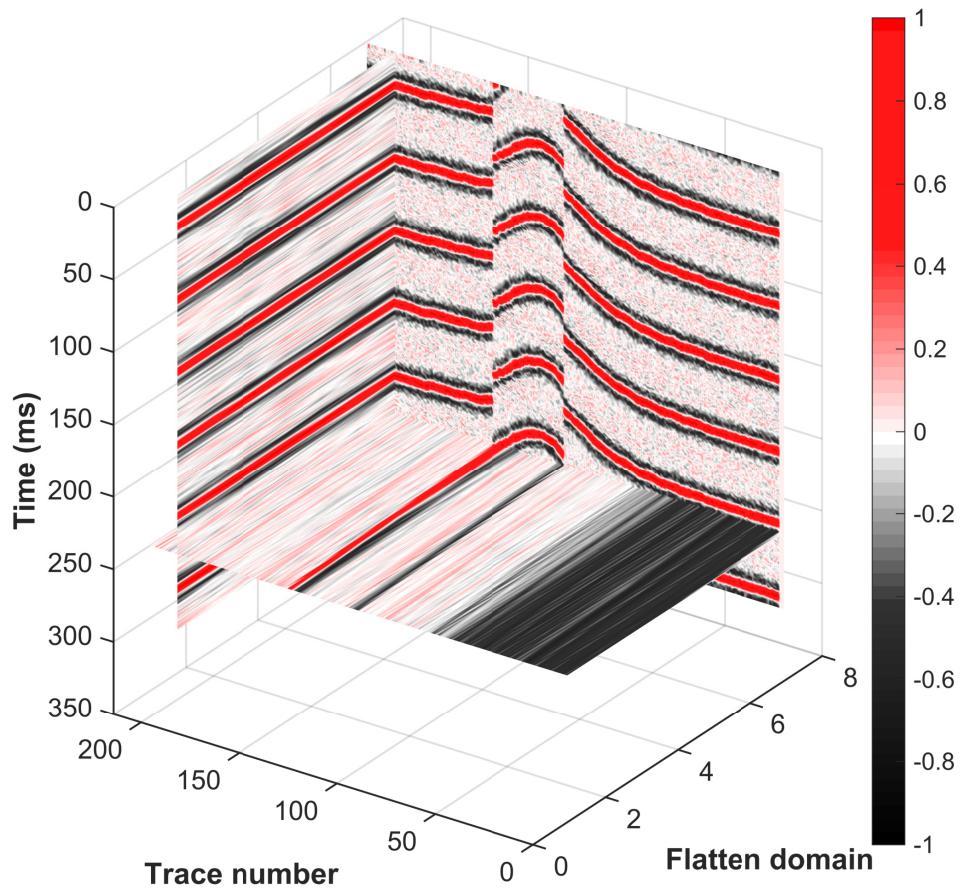


Figure 7: The flattened data of synthetic post-stack 2D example.

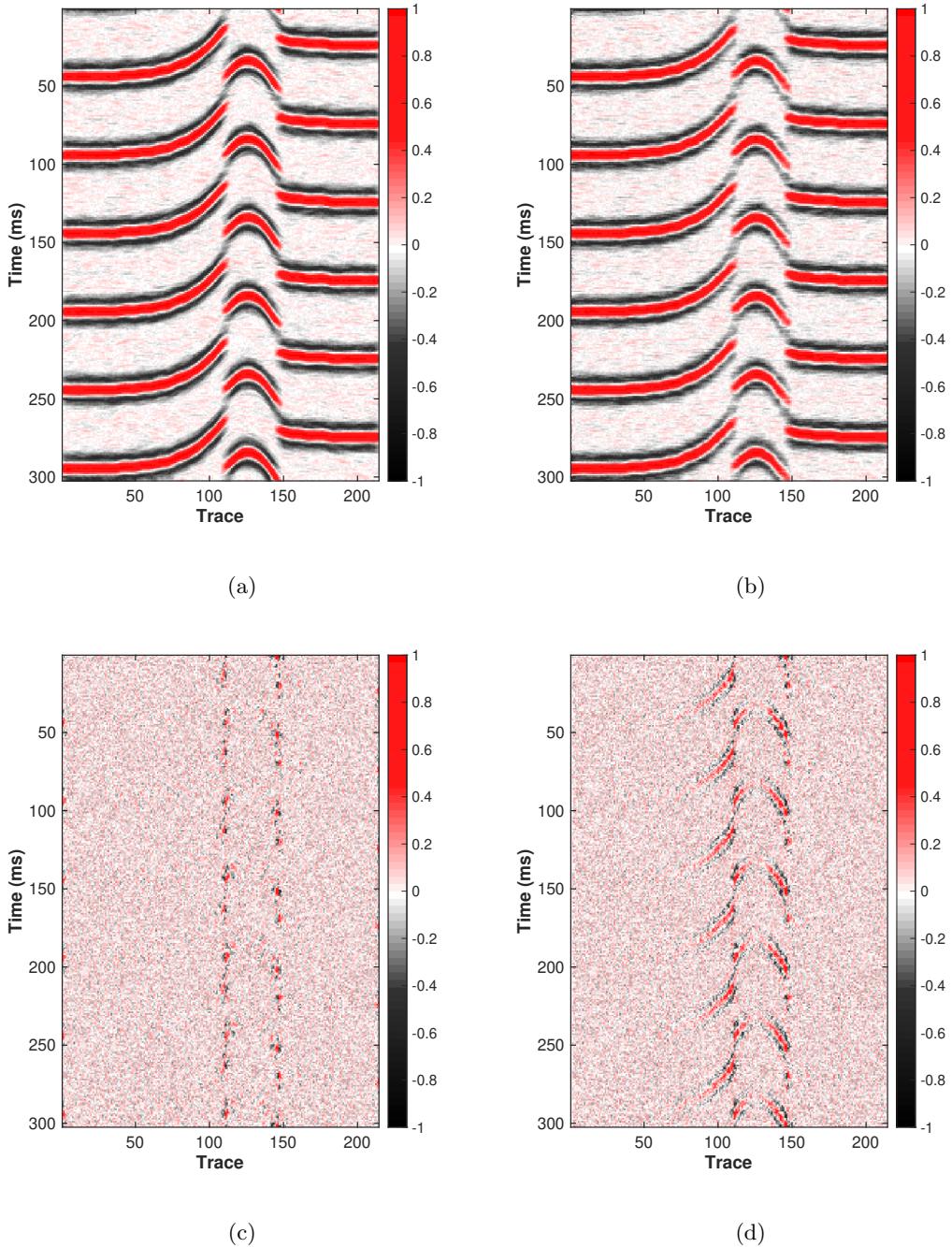


Figure 8: The denoised results and removed noise of synthetic post-stack 2D data. (a) The result by the proposed method ( $S/N=14.39\text{dB}$ ). (b) The result by the conventional smoothing filter without using the slope information ( $S/N=12.07\text{dB}$ ). (c) The removed noise of the proposed method.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

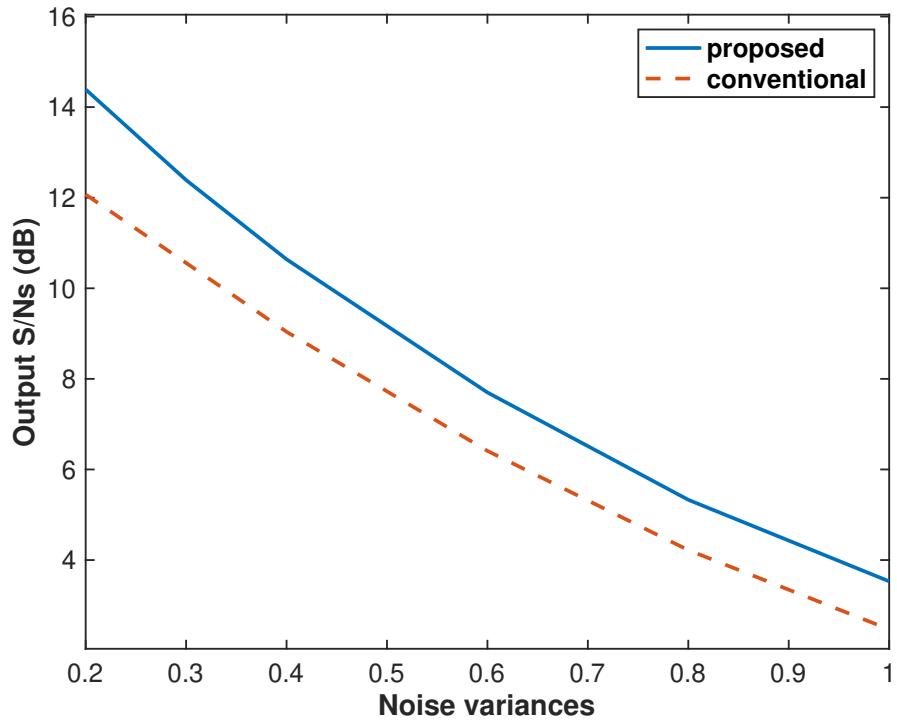


Figure 9: The denoising performances of the conventional (red) and proposed (blue) methods with respect to different noise variances.

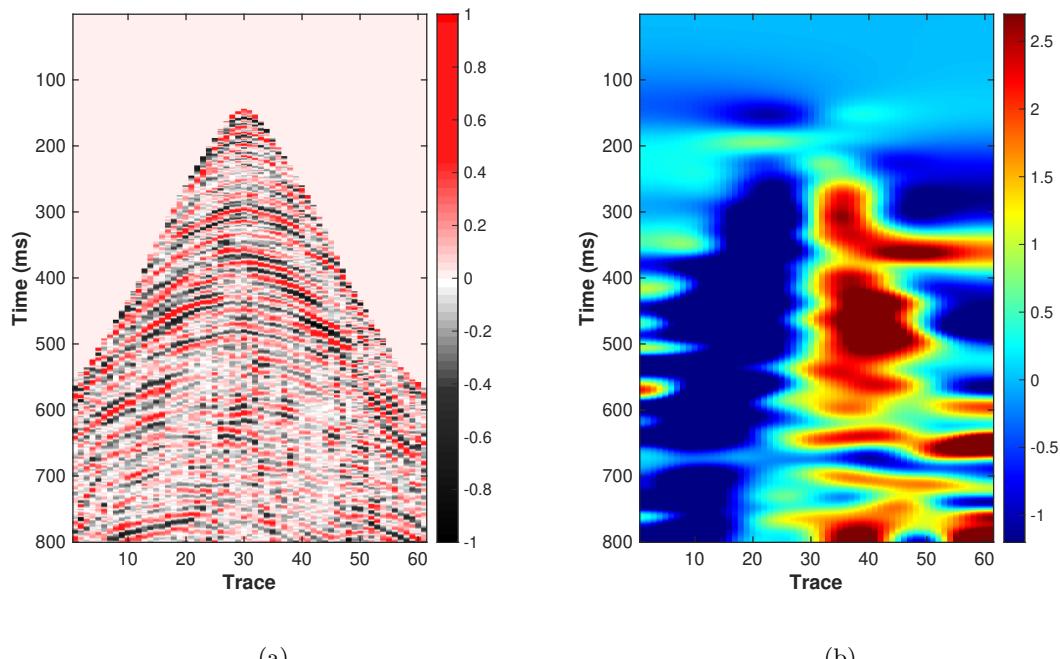


Figure 10: (a) The noisy pre-stack 2D field shot gather. (b) The local slope field of real pre-stack 2D data.

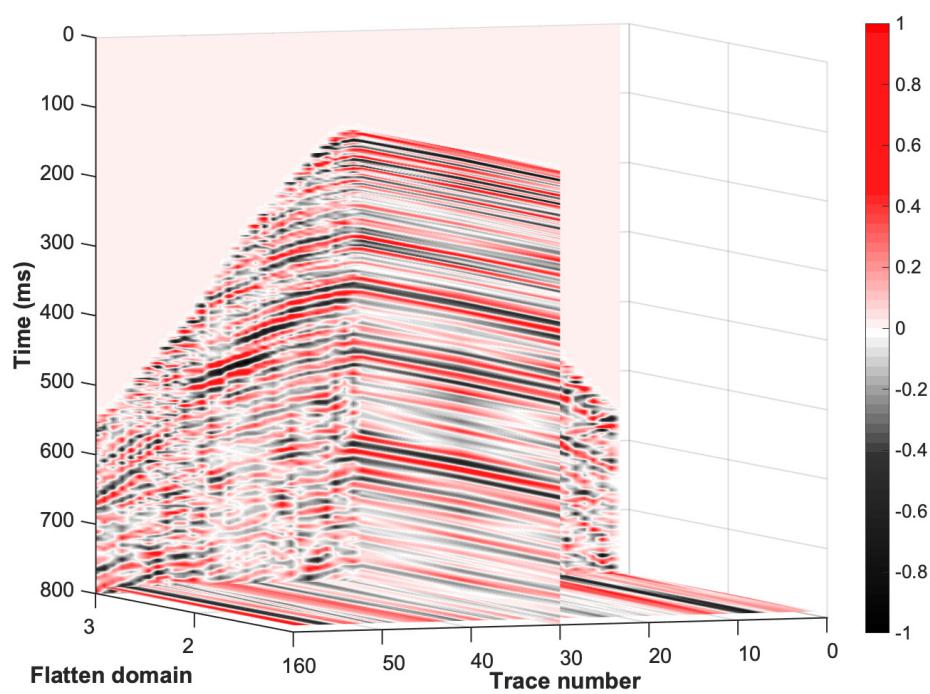


Figure 11: The flattened data of real pre-stack 2D example.

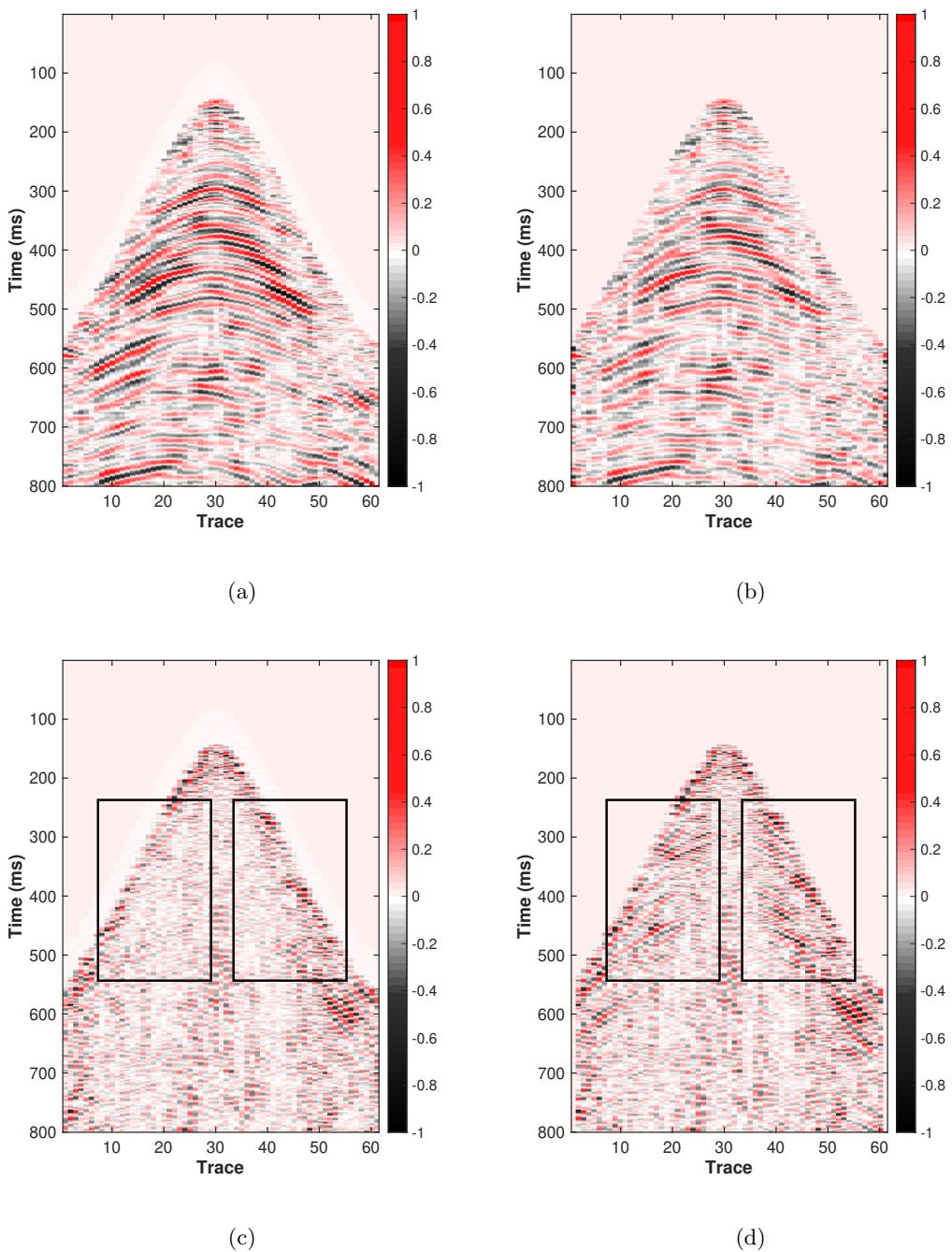
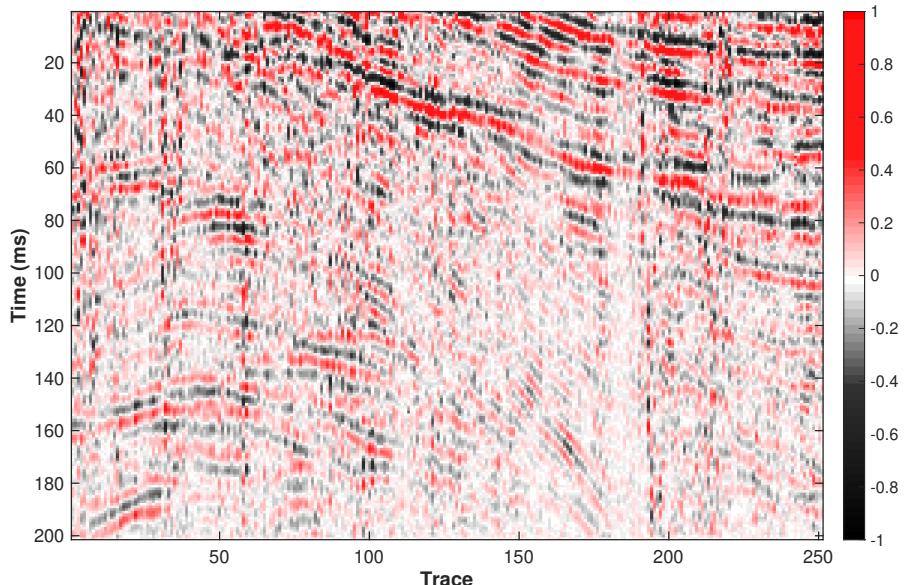
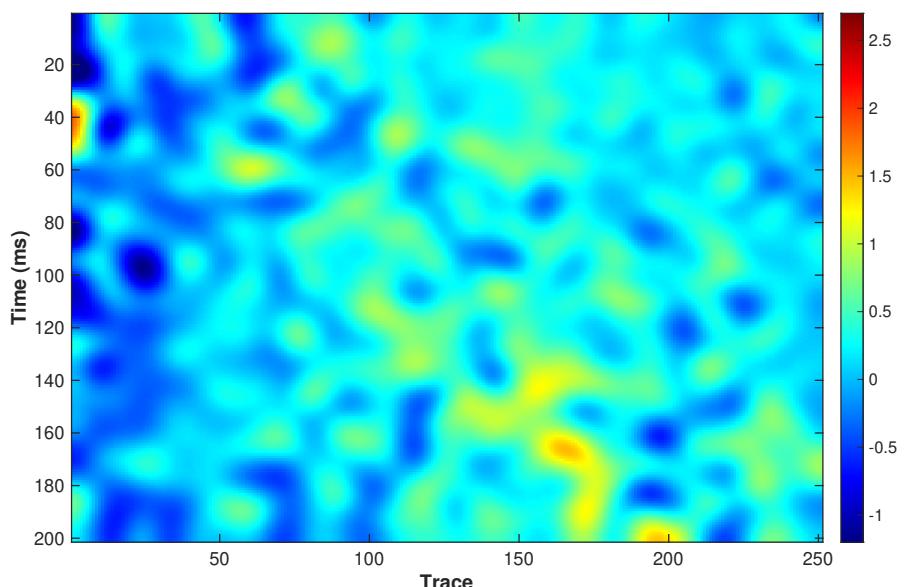


Figure 12: The denoised results and removed noise of real pre-stack 2D data. (a) The result by the proposed method. (b) The result by the conventional smoothing filter without using the slope information. (c) The removed noise of the proposed method. (d) The removed noise of the conventional smoothing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



(a)



(b)

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

—

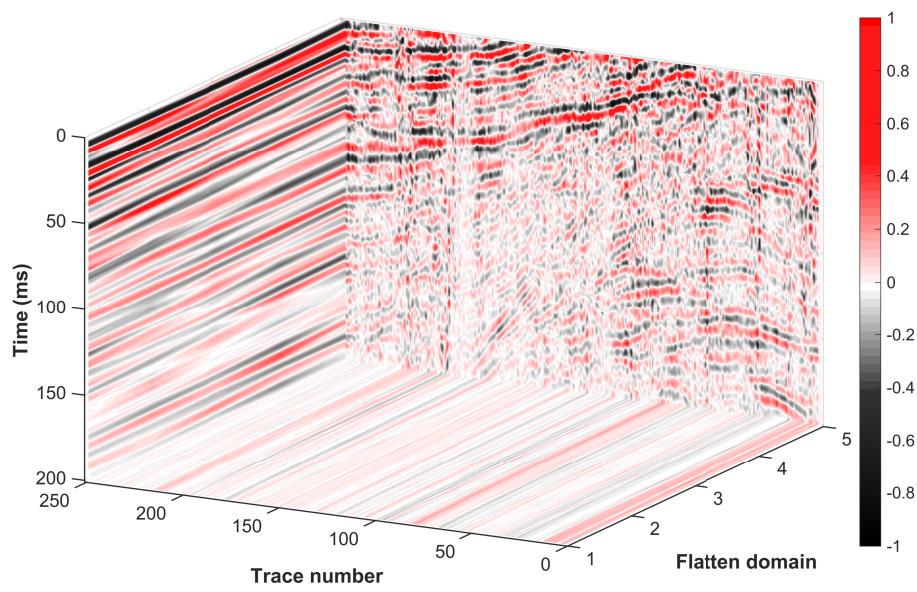
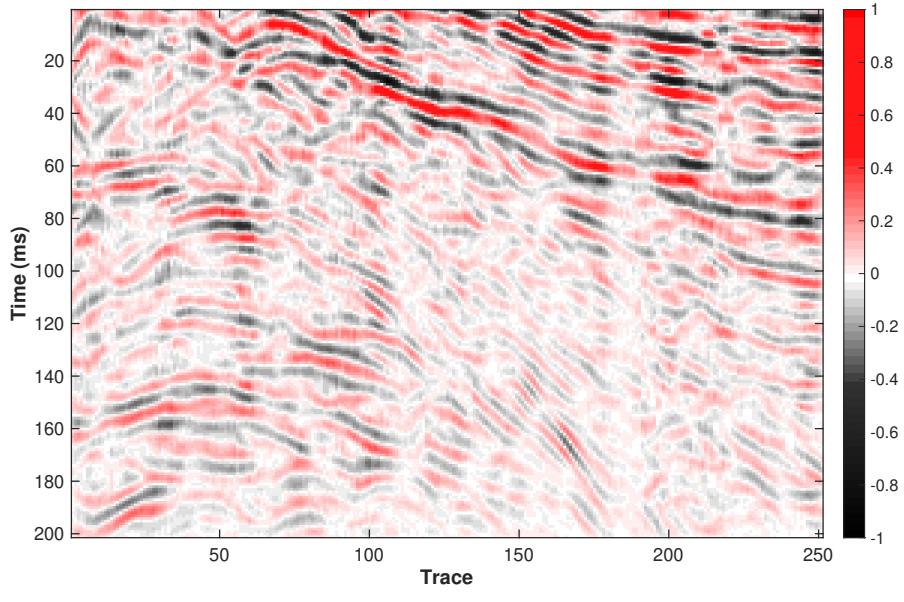
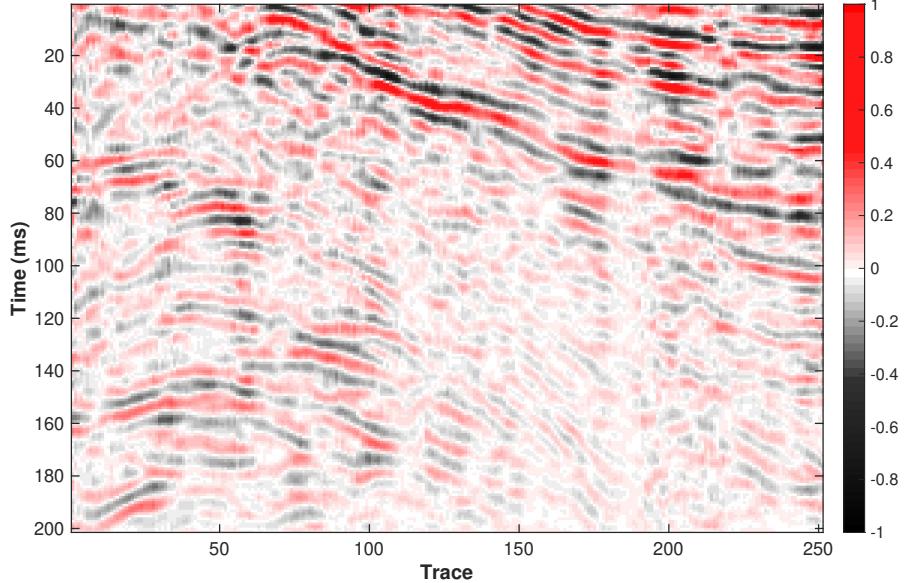


Figure 14: The flattened data of real post-stack 2D example.

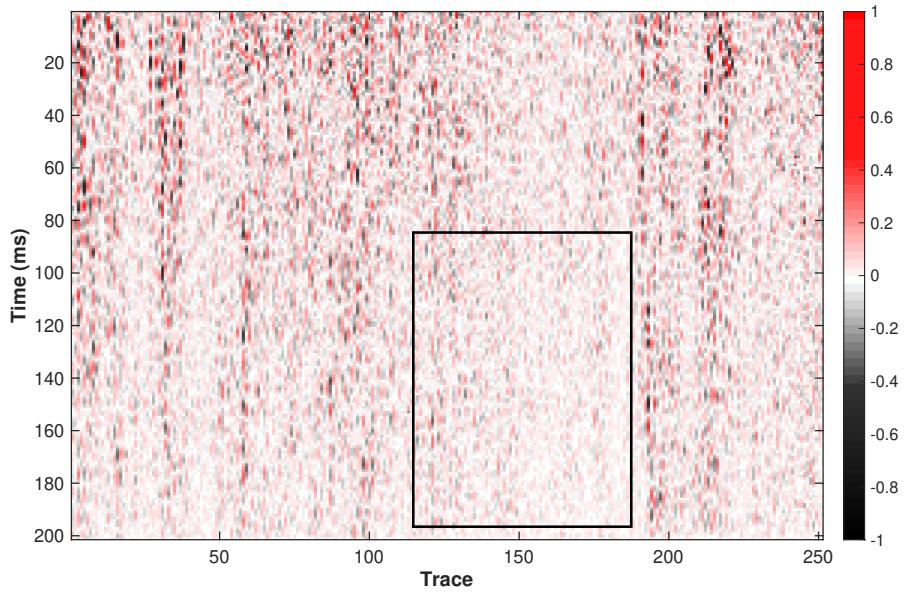


(a)

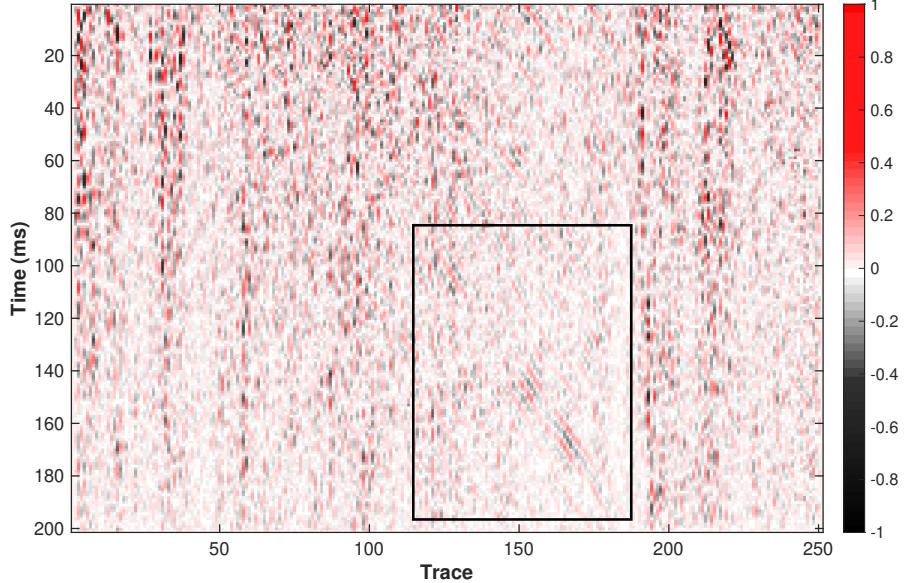


(b)

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.

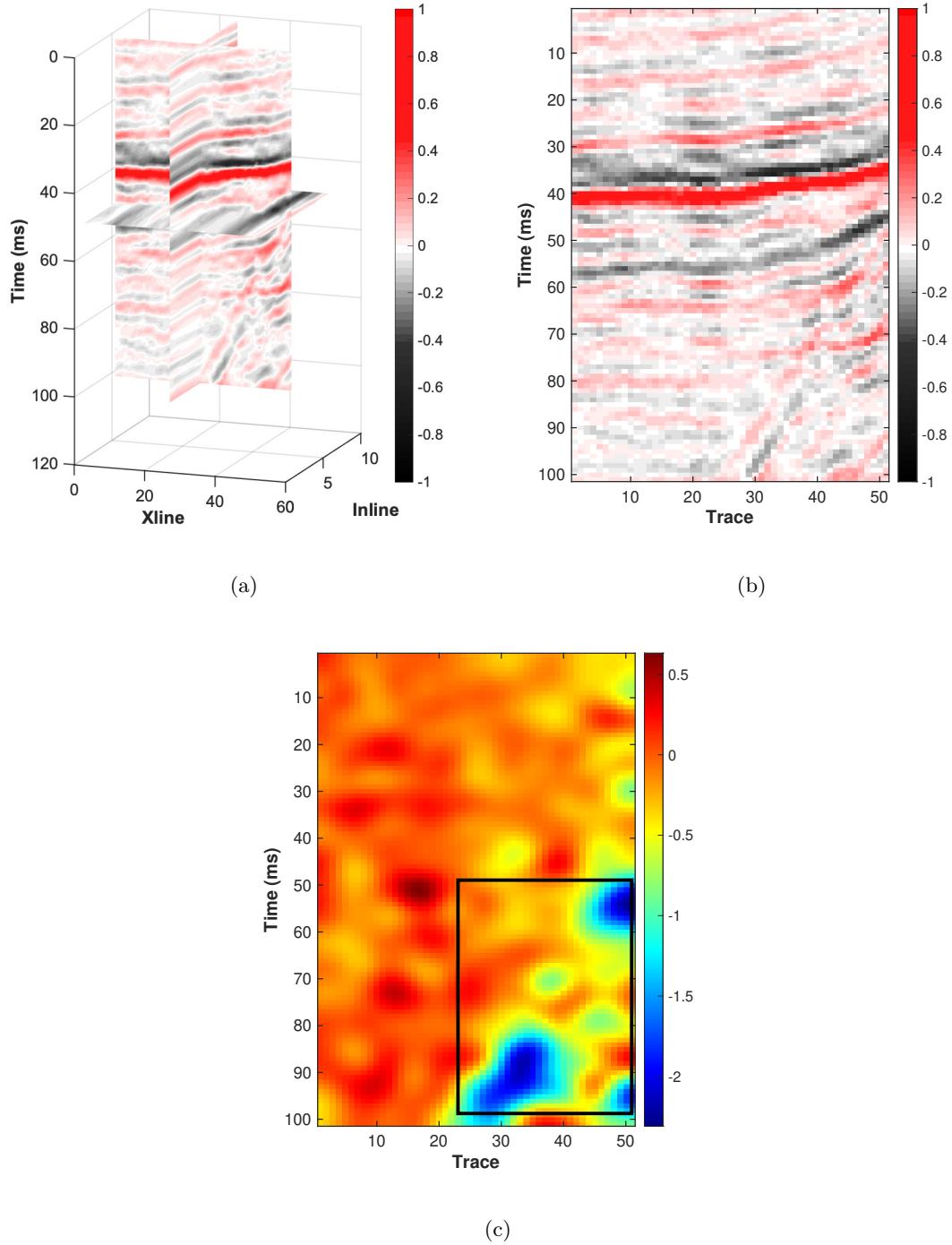


Figure 17: The field 3D data. (a) The 3D plot. (b) The slice in the 5th inline. (c) The local slope field of the 5th inline slice. It is calculated based on the raw data. The black rectangle indicates the location of the large-dip area.

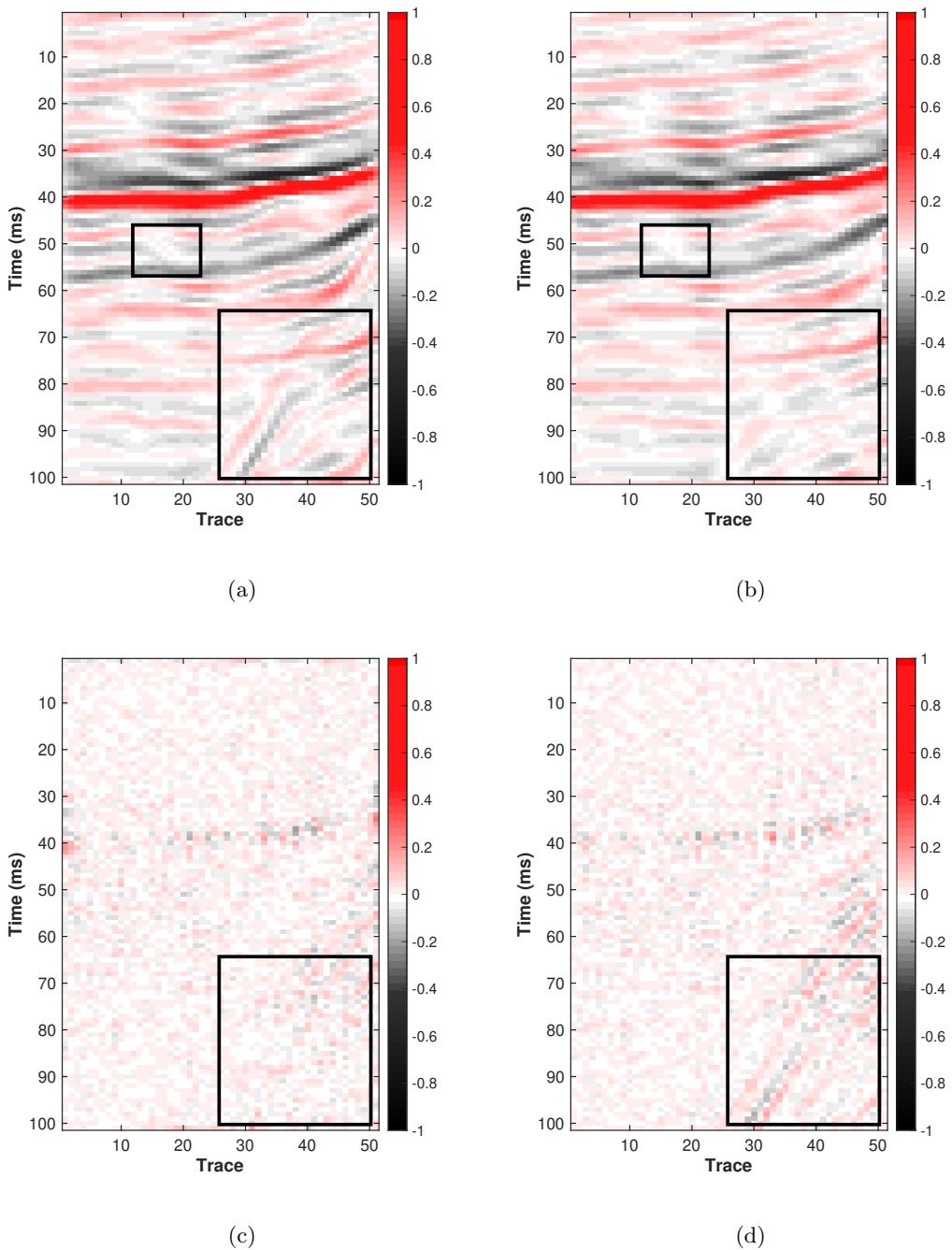


Figure 18: The denoised results and removed noise of the 5th inline slice in field 3D data.

(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.

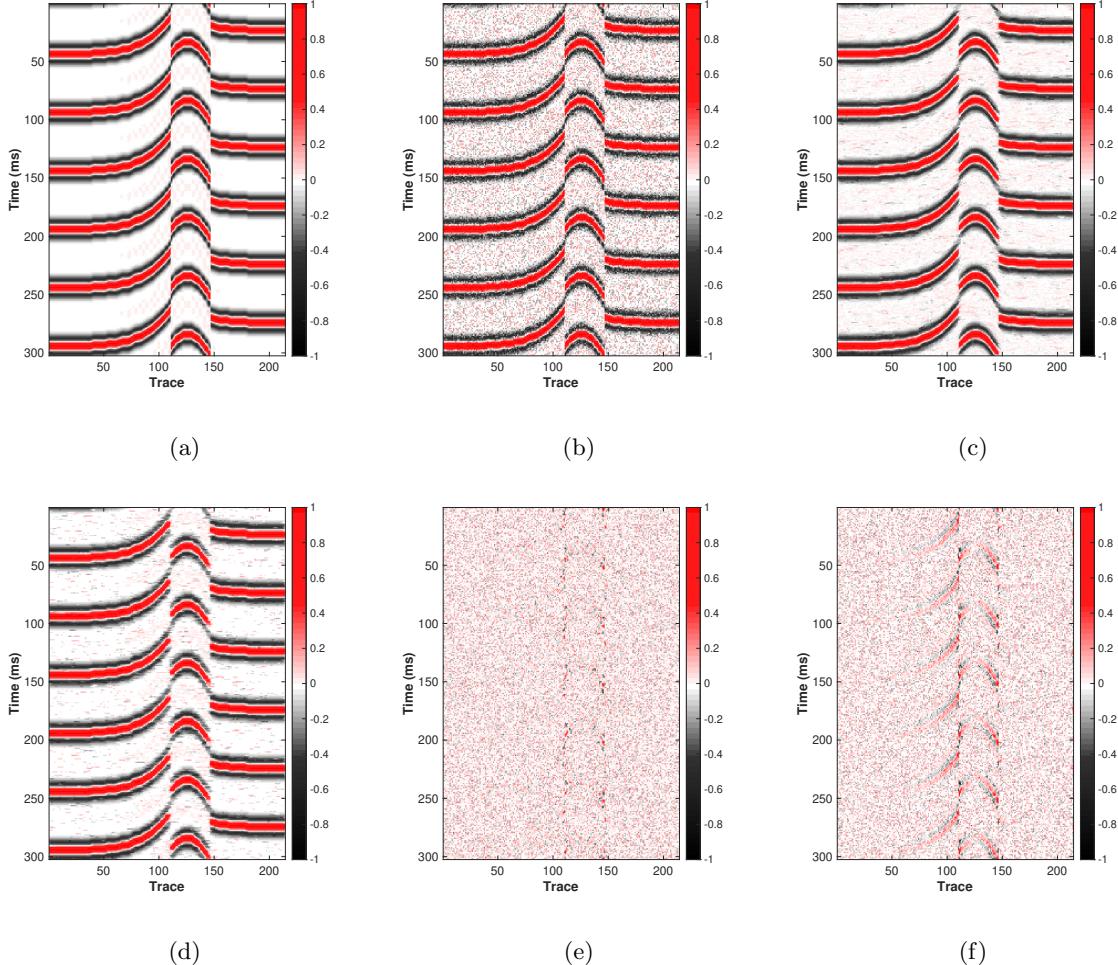
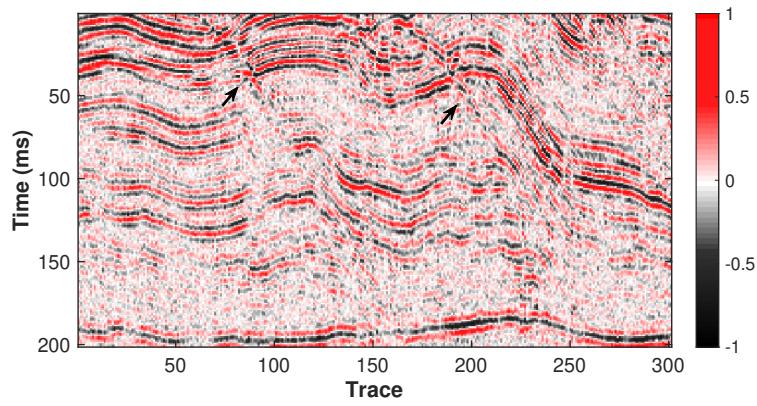
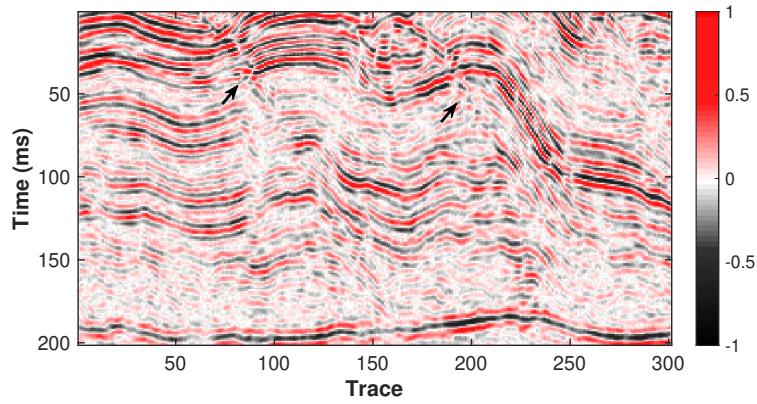


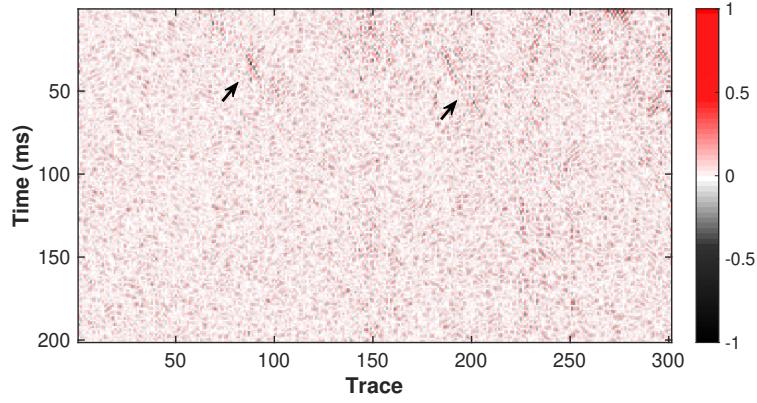
Figure 19: The test for impulse noise. (a) The clean data. (b) The noisy data with impulse noise. (c) The denoised result by the proposed method ( $S/N=15.18\text{dB}$ ). (d) The denoised result by the conventional median filter without using the slope information ( $S/N=12.61\text{dB}$ ). (e) The removed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



(a)



(b)



(c)

Figure 20: The stacked data with conflicting dips. (a) Original noisy data. (b) The denoised result. (c) The removed noise. Note the signal leakage from the events with non-dominant dips in the conflicting areas.

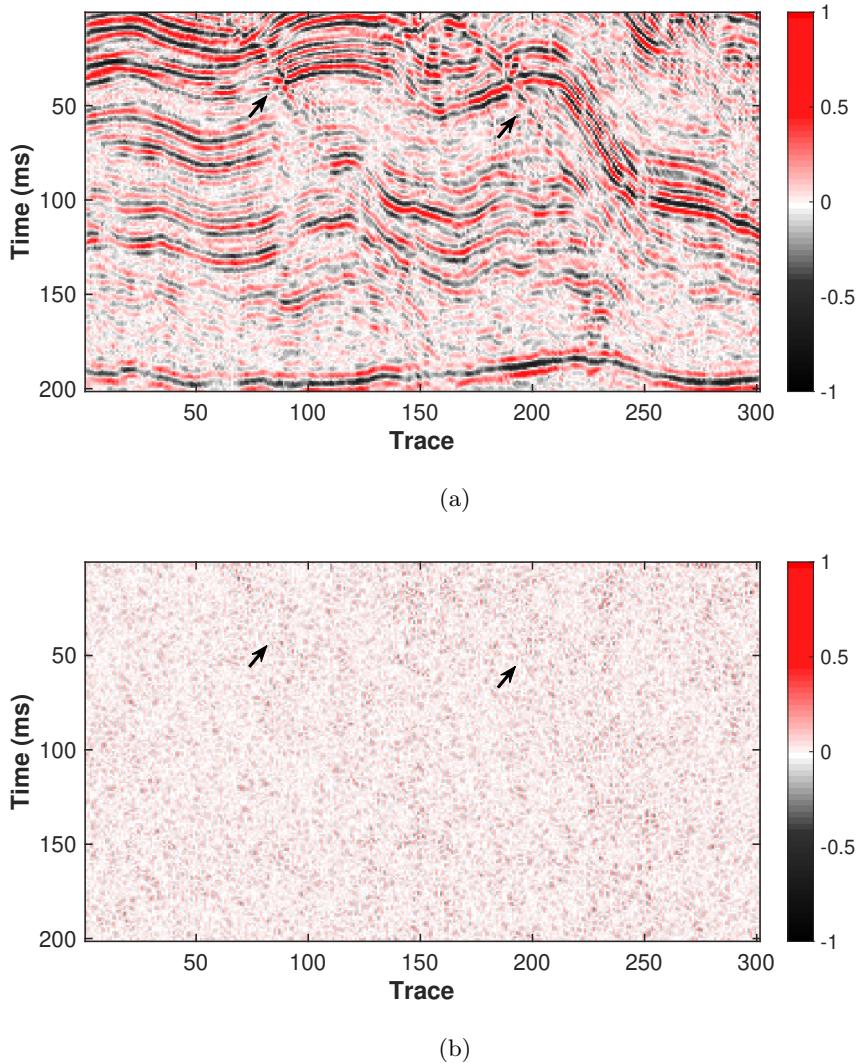


Figure 21: The results after the additional step that applies the package to the removed noise from the first step to extract leaked signals with non-dominant dips. Then, the extracted signals are added back to the denoised result of the first step. (a) The denoised result after additional step. (b) The removed noise after additional step.

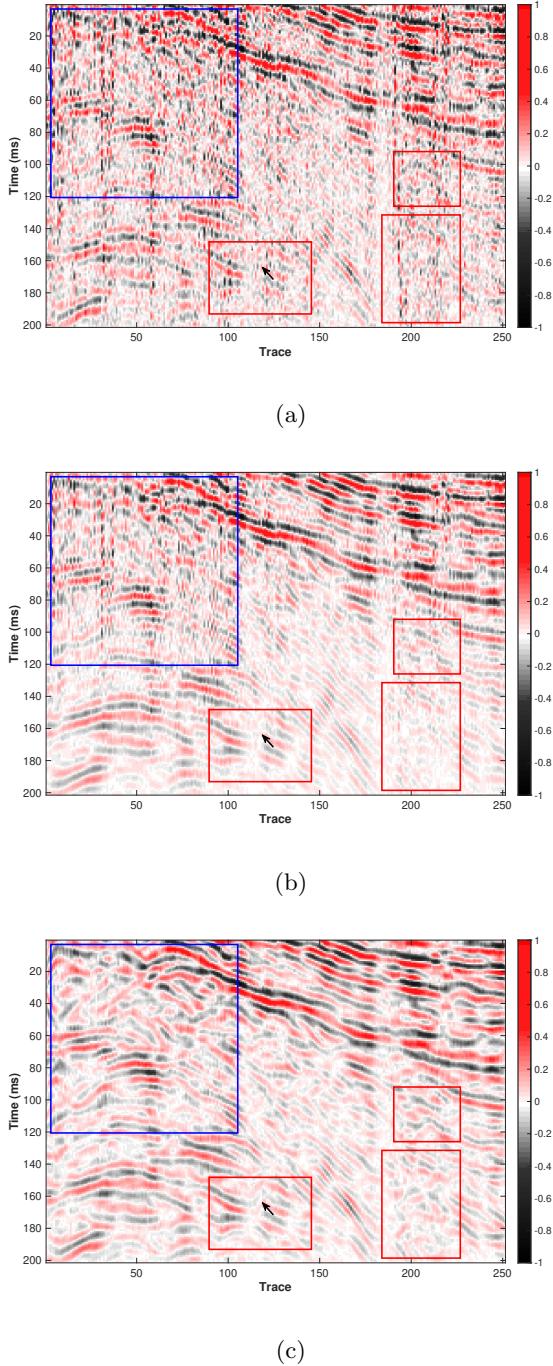
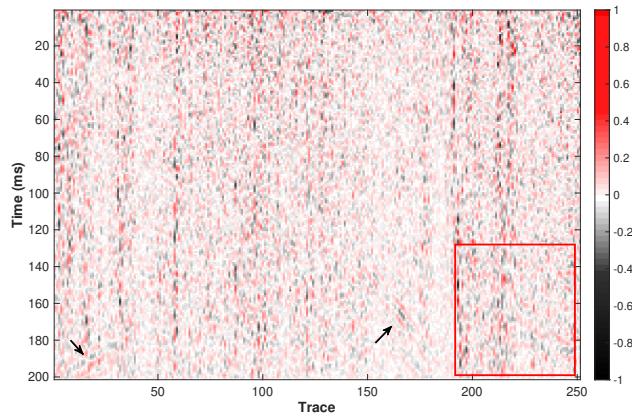
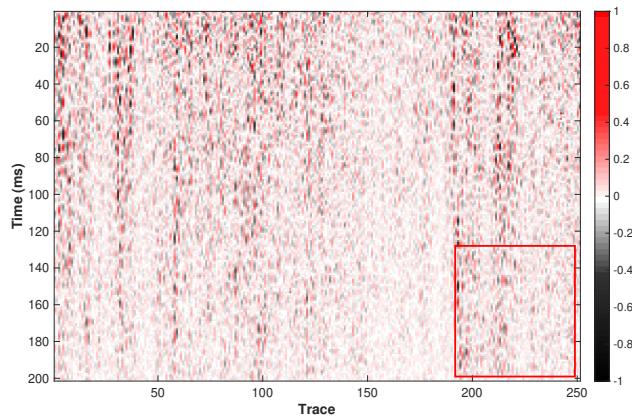


Figure 22: The denoising comparison of F-K and the proposed method. (a) The original noisy data. (b) The denoised result of F-K. (c) The denoised result of the proposed method. There is obvious residual noise in the F-K result (blue rectangle), and many of the structure details are harmed (red rectangles).



(a)



(b)

Figure 23: The removed noise of both F-K and the proposed methods. (a) The removed noise of F-K. (b) The removed noise of the proposed method. We can find more structure details in the former one (red rectangle).

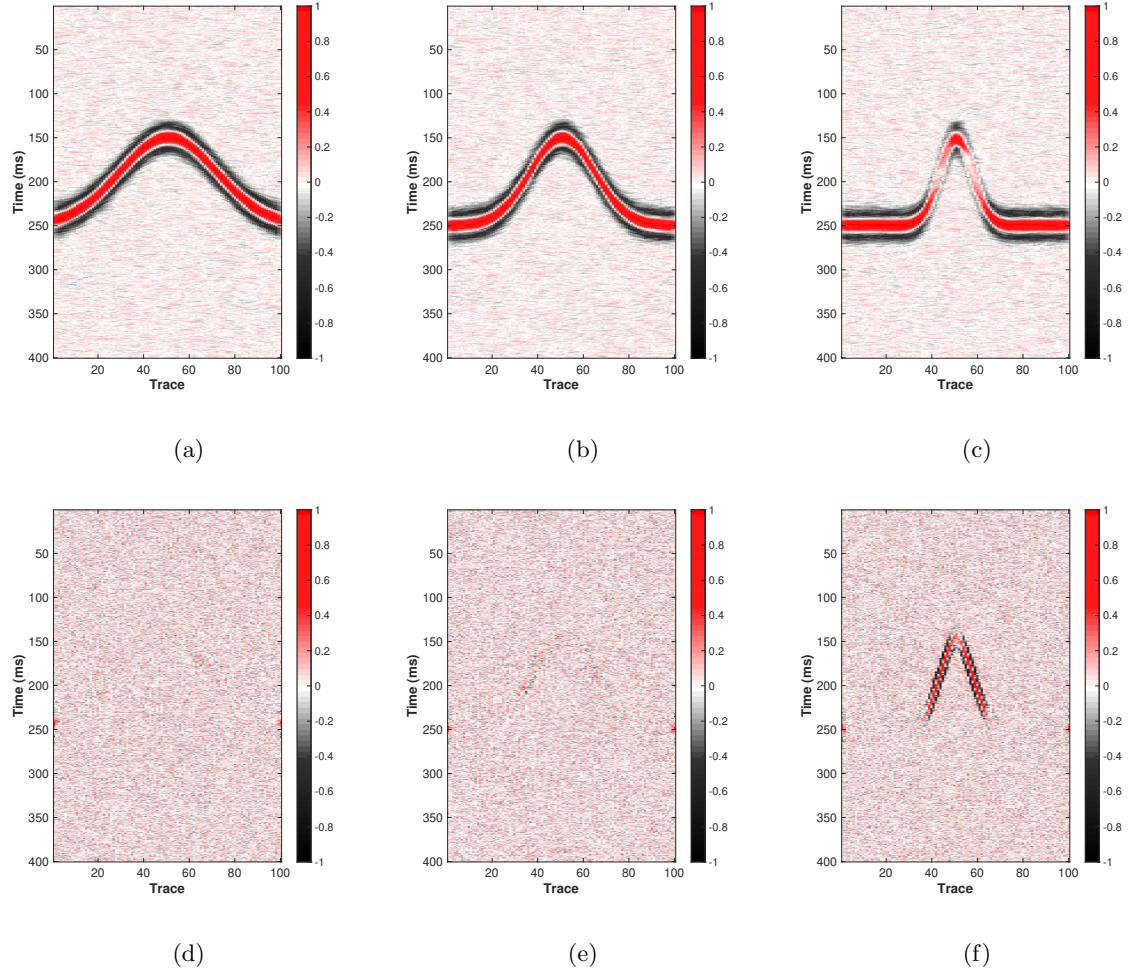


Figure 24: The denoised results with (a) gentle, (b) moderate and (c) steep events, and the removed noise from the data with (d) gentle, (e) moderate and (f) steep events.

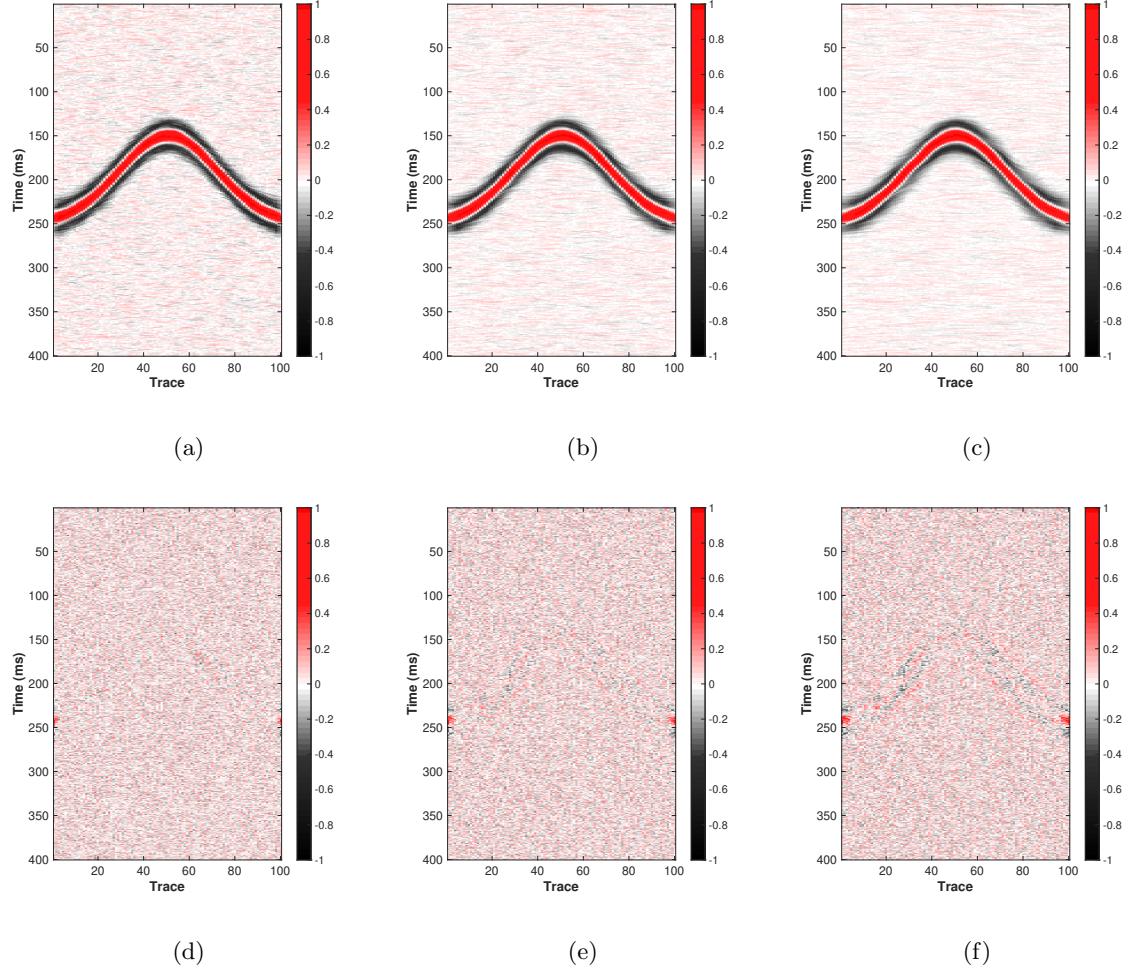


Figure 25: The denoised results by the smoothing lengths (a)  $R = 2$ , (b)  $R = 4$  and (c)  $R = 6$ , and the removed noise by the smoothing lengths (d)  $R = 2$ , (e)  $R = 4$  and (f)  $R = 6$ .