STRUCTURE-ORIENTED FILTERING OF POSTSTACK DATA – PROGRAM sof3d

Contents

Overview

Structure-oriented filtering provides a means of preserving signal parallel to structural dip while rejecting random and cross-cutting coherent noise. There are several techniques to preserve discontinuities across faults and stratigraphic edges. The simplest and perhaps the most common approach is to apply a median filter along structural dip. Fehmers and Höecker (2003) use an estimate of similarity (in their paper, they used chaos) to identify candidate edges; voxels that fall at one these locations are simply not filtered. Luo et al. (2006) follow Kuwahara et al. (1976) to choose the smoothest (in general non-centered) window in which to apply the filter; in general windows straddling an edge have more rapid lateral changes in amplitude and are not chosen. Program **sof3d** integrates both the Fehmers and Höecker (2003) and Luo et al. (2006) edge preservation workflows with some important differences. First, the measure of smoothness or the presence of an edge is provided through a previously computed similarity volume. Second, there is a relatively wide choice of filters including alpha-trimmed mean, lower-upper-middle, and principal component filters, with the mean and median filters being end members of the alpha-trimmed mean filter. It is a best practice to examine the data rejected by the filter to ensure that valuable signal has not been removed; sof3d therefore allows computation of the rejected noise as well as the filtered result.

Computation flow chart

The inputs to program **sof3d** include seismic amplitude (or other attribute to be smoothed such as velocity or impedance), the inline and crossline estimates of reflector dip computed from program **dip3d** and a measurement of similarity from program **similarity3d**. The inline and crossline estimates of dip may have been previously filtered using program **filter_dip_components**. Furthermore, the seismic amplitude data may have been subjected to a previous pass through structure-oriented filtering program **sof3d** or may have been spectrally balanced using program **spec_cmp.** The outputs include principal component- (also called Karhunen–Loève, or KL-) alpha-trimmed-mean-, LUM- or mean-filtered versions of the input seismic amplitude data.

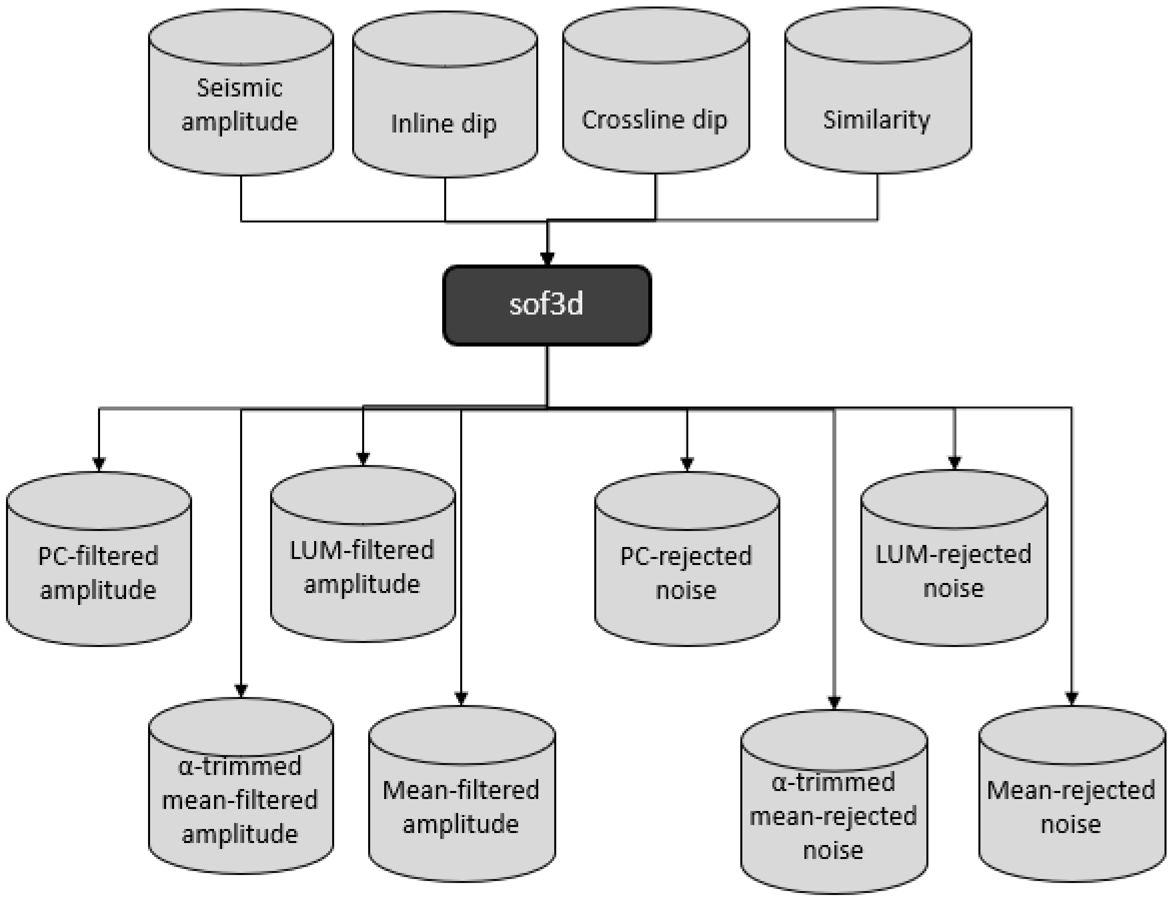

Seismic amplitude
Inline dip
Crossline dip
Similarity
sof3d
PC-filtered amplitude
LUM-filtered amplitude
PC-rejected noise
LUM-rejected noise
α-trimmed mean-filtered amplitude
Mean-filtered amplitude
α-trimmed mean-rejected noise
Mean-rejected noise

Figure 1.

Output file naming convention

Program **sof3d** will always generate the following output files:

Output file description	File name syntax
Program log information	sof3d_*unique_project_name_suffix*.log
Program error/completion information	sof3d_*unique_project_name_suffix*.err

where the values in red are defined by the program GUI. The errors we anticipated will be written to the **.err* file and be displayed in a pop-up window upon program termination. These errors, much of the input information, a description of intermediate variables, and any software trace-back errors will be contained in the **.log* file.

Depending on filters chosen, **sof3d** will also generate the following filtered files:

Output file description	File name syntax
Principal component (KL) filtered data	d_pc_filt_*unique_project_name_suffix*.H
Alpha-trimmed mean filtered data	d_alpha_trimmed_mean_filt_*unique_project_name_suffix*.H
Lower-Middle-Upper (LUM) filtered data	d_lum_filt_*unique_project_name_suffix*.H
Mean filtered data	d_mean_filt_*unique_project_name_suffix*.H

If you choose *Output rejected noise for each selected filter, you also get the fol*lowing files:

Output file description	File name syntax
Principal component (KL) rejected noise	d_pc_noise_*unique_project_name_suffix*.H
Alpha-trimmed mean rejected noise	d_alpha_trimmed_mean_noise_*unique_project_name_suffix*.H
Lower-Middle-Upper (LUM) rejected noise	d_lum_noise_*unique_project_name_suffix*.H

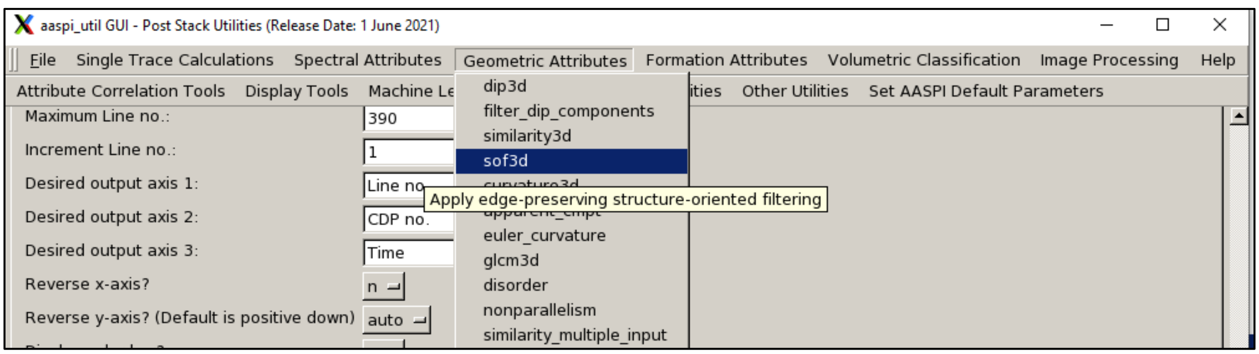
Mean rejected noise	d_mean_noise_*unique_project_name_suffix*.H

Finally, if you choose to apply sof3d to multiple bandpassed versions of the input data under the *Spectral balancing parameters* tab you will obtain this file:

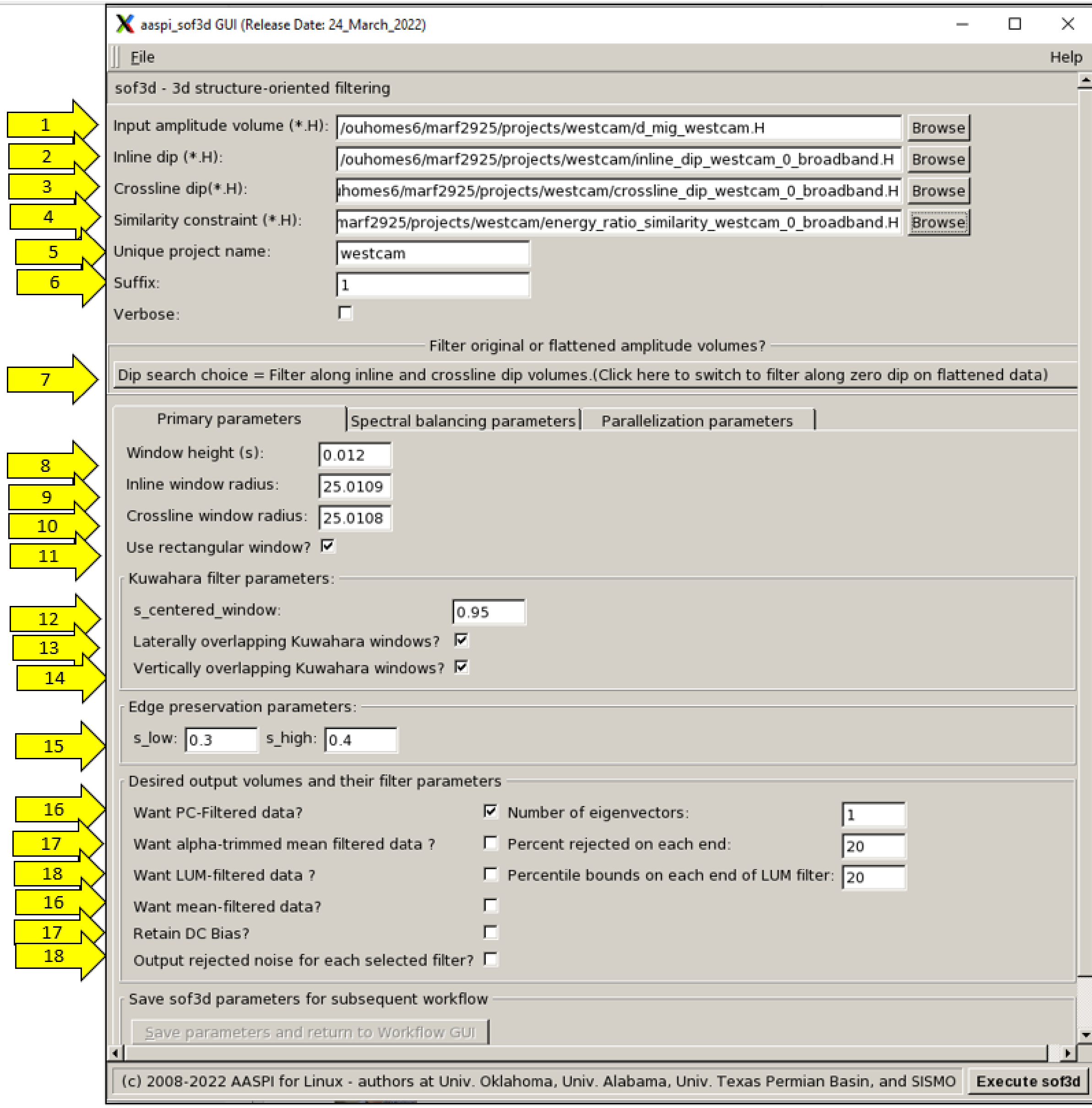
Output file description	File name syntax
Filter banks applied to the input data	sof_filterbanks_*unique_project_name_suffix*.H

Computing structure-oriented filtered data

Once we have volumetric estimates of dip and azimuth as well as a similarity/coherence attribute sensitive to edges, we can apply simple filters that reject random noise and preserve edges. The general name for this process is edge-preserving structure-oriented filtering. Program **sof3d** is found under the *Geometric Attributes* tab:



The following GUI appears:

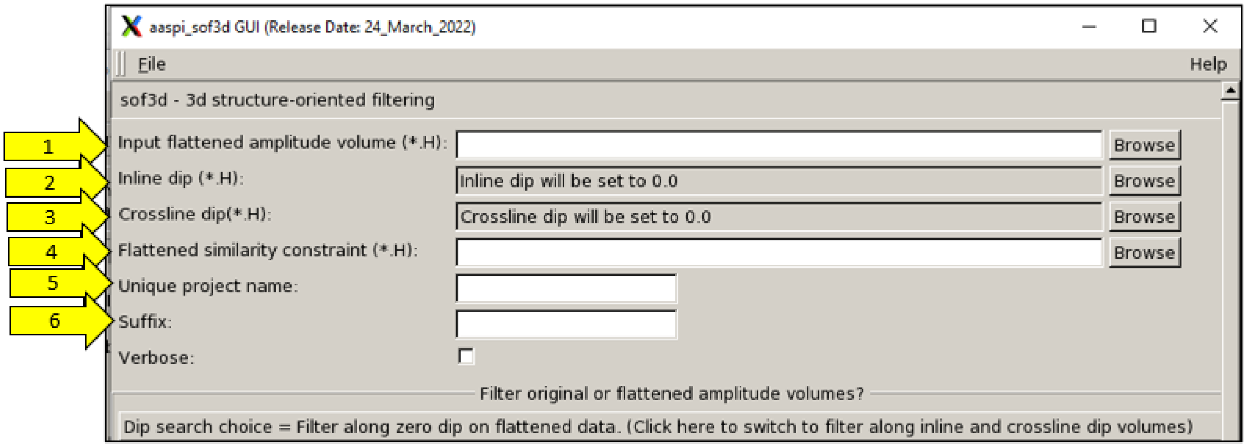


Input data volumes

First, select (1) the seismic amplitude volume to be filtered, which in this example is *d_mig_westcam.H*. Next, select the (2) inline and (3) crossline components of dip and (4) the similarity attribute volume generated in program **similarity3d** which will be used to control the edge preservation components of the filters. The (5) unique project name for this data volume is 'westcam. Because this is the first pass of structure-oriented filtering, (6) type '1' as the suffix. pass of filtering. Later, if I choose to cascade the filters, using the output of the first filter as the input to the second pass, I might set the suffix to be '2'.

Filtering along structural dip or parallel to a flattened horizon

In general, we want to apply structure-oriented filtering along structural dip which is the (7) default setting. However, in some cases you may wish to filter parallel to picked horizon in order to enhance gas hydrate bottom simulating reflectors, or to enhance and then subtract interbed multiples that are parallel to a picked multiple generator. In either of these scenarios, you need to first flatten both the seismic amplitude and similarity volumes on the picked horizon. Clicking the (7) Filter original or filtered amplitude volumes modifies the GUI slightly:



where now you are not only prompted to select flattened amplitude and similarity volumes but also prevented from entering inline and crossline dip volumes. Under this mode of operation, the algorithm will assume the desired inline and crossline dips are both zero, producing the desired result.

After setting these preliminary parameters, I need next to define the filter window size and shape.

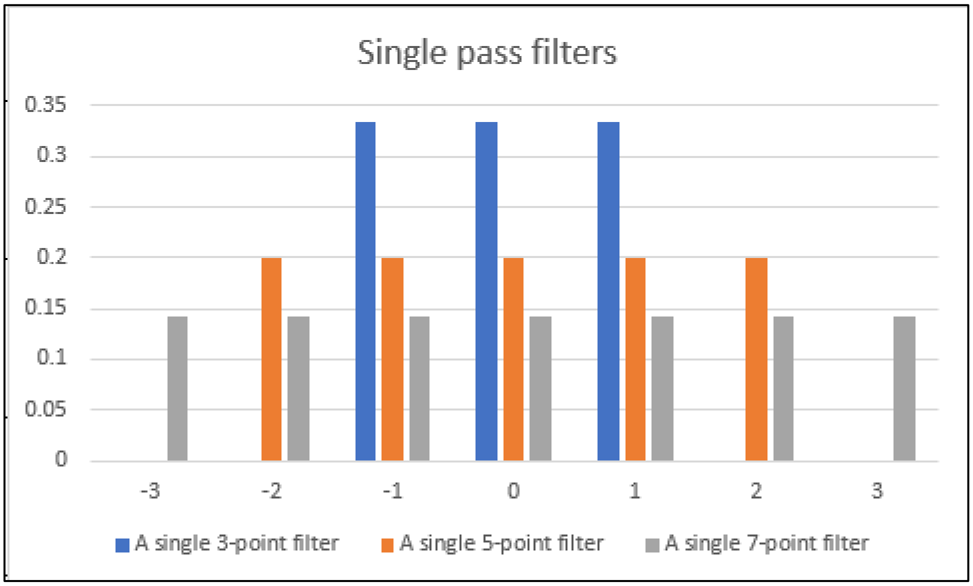
Analysis window size

The principal component (also called Kohonen-Loève or KL) filter uses the data that fall within a 3D analysis window defined by both lateral dimensions and a (8) vertical analysis window. In contrast, the mean, alpha-trimmed mean, and Lower-Middle-Upper (LUM) filters only use the interpolated data samples that fall along a dipping plane intersecting the center of the window. The lateral size of these windows is defined by a (9) inline and (10) crossline radius. Finally, (11) choose whether to use a rectangular or elliptical analysis window. In this example, the result is a 3×3=9-trace analysis window. The vertical analysis window spans 7 samples. Thus, for this example, the PC-filter exploits the patterns and the information content of 3×7=63 voxels while the other three filters use the information content of only 9 voxels.

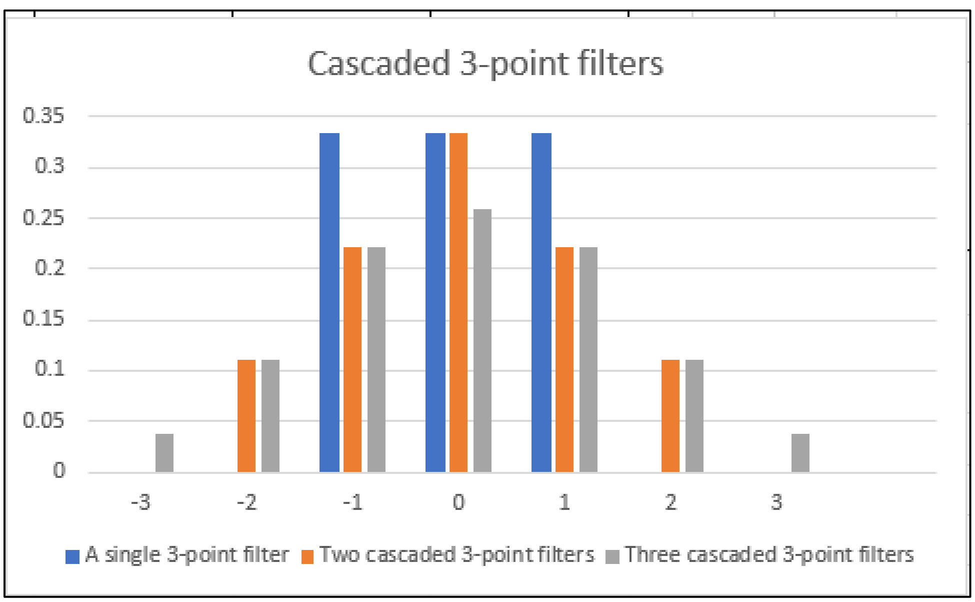
Both the computation time and the strength of the filter increase with increasing window size. For good quality data, it is more effective workflow to iteratively smooth using smaller windows rather than to double the window size in both directions. Such smaller windows not only follow curving reflectors better but also implicitly taper the filter towards the edges.

Iterative filtering using small windows vs single pass filtering using larger windows

The following figure shows the weights used for 1D single pass 3-, 5-, and 7-point centered mean filters:



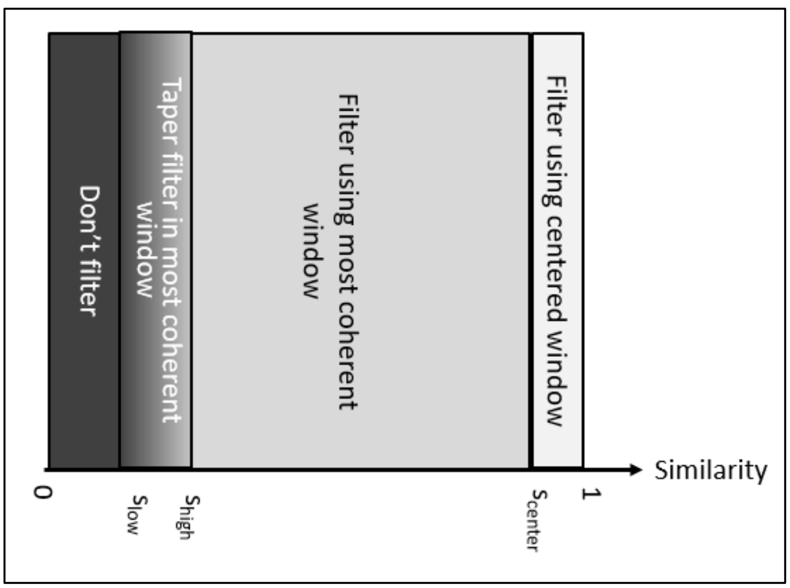
Note that the weights for each sample of a given filter are the same. In contrast, if we generate 3-, 5-, and 7-point filters, by the iterative application of a 3-point filter we obtain these (effectively tapered) weights:



For 2D filters, cascaded filters are computationally more efficient. For example, the cost of a mean filter applied to a 7×7 window is 49 additions and multiplications. The cost of a 3×3 filter is 9 additions and multiplications. If we cascade this filter 3 times, the total cost of applying a structurally adaptive, tapered 7×7 filter is 27, which is about half the computational effort of the single pass 7×7 filter. If the original dip estimation is noisy as it is here, we advise recomputing the dip using program **dip3d** before the 2nd pass of filtering.

Kuwahara filtering

For now, consider a simple mean (average) filter. In addition to other types of filters that will be described later in this documentation, there are several additional strategies for edge preservation while applying a filter along structure. The following diagram summarizes how we control these additional filters, based on an estimate of its continuity using a similarity (coherence) attribute volume:



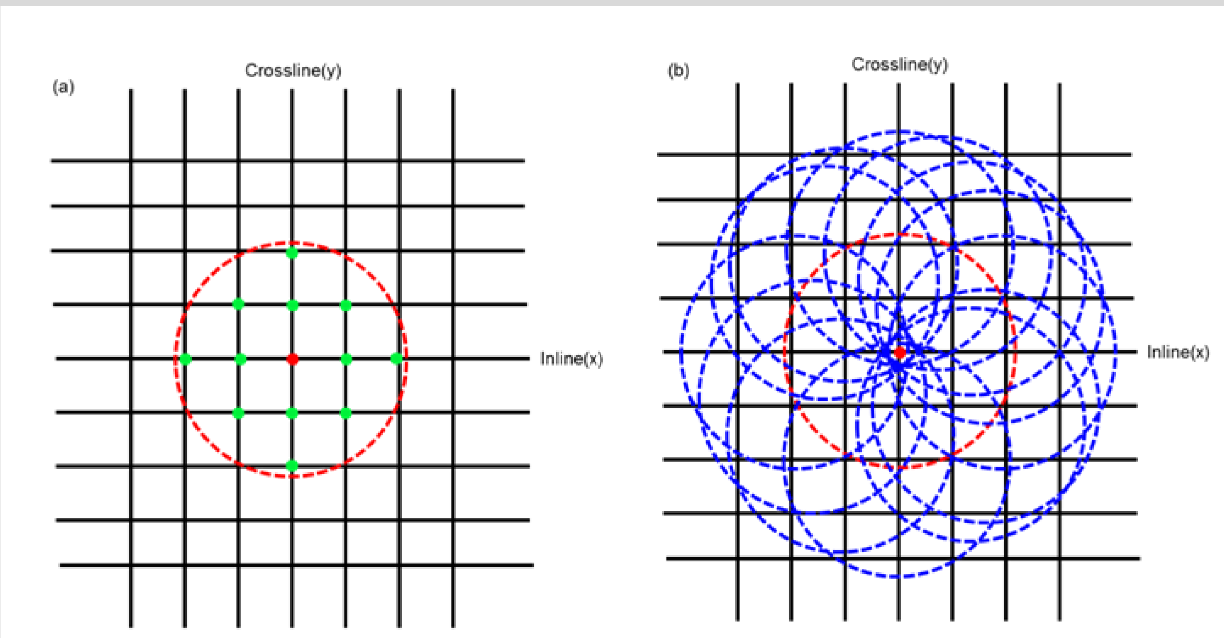
Luo et al. (2002) were the first to apply a Kuwahara filter to seismic data. Originally applied to by Kuwahara et al. (1976) to enhance the edges seen in medical images, Luo et al. (2002) examined the mean and standard deviation of a suite of overlapping windows of the same size that all included the analysis point. Rather than replace the value of the analysis point with the mean of the centered window, the Kuwahara filter replaces it with the mean of the (potentially non-centered) window that exhibits the smallest standard deviation. Marfurt (2006) extended Luo et al.'s (2006) concept to be structure-oriented and chose the "best" window to be the one that exhibited the highest value of coherence (similarity). The Kuwahara filter both smooths (by the application of the internal mean or other filter) and sharpens (by choosing a non-centered rather than centered window) the data. For this reason, the Kuwahara filter can organize the data in a way that result in seismic amplitude images that look blocked, or “patchy” in parts of the data where we expect it to be smooth. We therefore define (12) a threshold similarity value, s_{center}, above which the filter always uses a centered analysis window.

For mean, alpha-trimmed mean, and LUM filters that apply only to the plane of data intersecting the analysis point, we only need to examine (13) lateral shifting of the windows. However, for principal component filters we may wish to define a Kuwahara filter that searches over (14) vertically shifted window as well. For the example shown here, the mean, alpha-trimmed mean, and LUM filters examine the data in 9 overlapping 2D windows, whereas the principal component filter examines the data in 9×7=63 overlapping 3D windows. Through careful memory management, the cost of a 63 window Kuwahara multiwindow filter is only a few percent more expensive than a simple centered window filter.

Theory: Kuwahara windows

Programs **dip3d, sof3d**, **sof_prestack,** and **kuwahara3d** all use a modification of overlapping window parameter estimates introduced by Kuwahara et al. (1976) in medical imaging. The original idea is simple. If an analysis window contains five traces, then there is a total of five windows (a centered window and four adjacent, offset windows) that contain the analysis point. In Kuwahara et al.'s (1976) original work and Luo et al.'s (2002) edge-preserving smoothing algorithm, one calculates the mean and standard deviation of each window. That window which has the smallest standard deviation is hypothesized to be less noise contaminated. The mean of this window is then used as the output for the analysis point. Marfurt (2006) modified this approach for volumetric dip calculations where he used 3D rather than 2D overlapping windows. In program **sof3d** the "best" window is the one with the highest measure of similarity (e.g., semblance, Sobel filter, or energy ratio). In program **dip3d** using the GST estimate of dip the best window is the one with highest measure of planarity. In program **kuwahara3d** the best window is the one exhibiting the smallest coefficient of variation.

The lower left figure represents a 13-trace circular analysis window centered about the analysis point indicated by the red solid dot. Each of the traces represented by the green dots in the lower left figure form the center of an additional twelve 13-trace blue circular analysis windows shown in the lower right figure. For program **sof3d**, we have precomputed and saved the similarity of each of these 13 blue analysis windows using program **similarity3d**. We therefore read these precomputed data in, select the one with the highest similarity, and apply a mean, alpha-trimmed mean, Lower-Upper-Middle, or principal-component (Karhunen-Loève) filtered estimate of the signal. We then save the result at the (perhaps non-centered) analysis point to the filtered volume. Use of such laterally shifted windows helps avoid smoothing across faults. Use of vertically shifted analysis windows helps avoid smearing across angular unconformities.

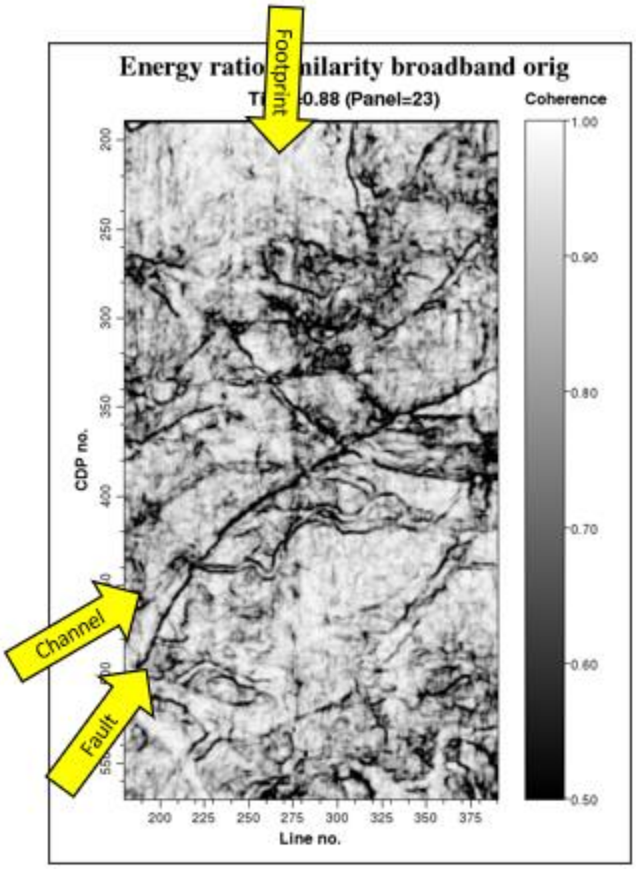


Edge preservation parameters

The next set of (15) and (16) edge-preservation parameters follow the work of Fehmers and Höecker (2003), who use an auxiliary attribute volume that controls where smoothing occurs. Although they do not explicitly state the attribute they use, their application is based on the gradient structure tensor estimates of vector dip, suggesting that they may have used chaos as their edge-sensitive attribute. In the AASPI software, we will use one of the similarity attributes to define the existence of an edge. For similarity values $s < s_{\mathrm{low}}$, we assume we have a strong edge, whereby the Kuwahara filtered data are assigned weights of $w=0.0$. If the value of the similarity attribute is greater than $s_{\mathrm{high}} < s < s_{\mathrm{centered}}$, and the Kuwahara filtered data are assigned weights of w=1.0. Finally, for values of similarity, $s_{\mathrm{low}} < s < s_{\mathrm{high}}$ we compute a weight $w=(s-s_{low})/(s_{high}-s_{low})$, and a compute the linearly weighted average of the Kuwahara filtered and unfiltered data $d_{out}=w*d_{filt}+(1-w)*d_{orig}$.

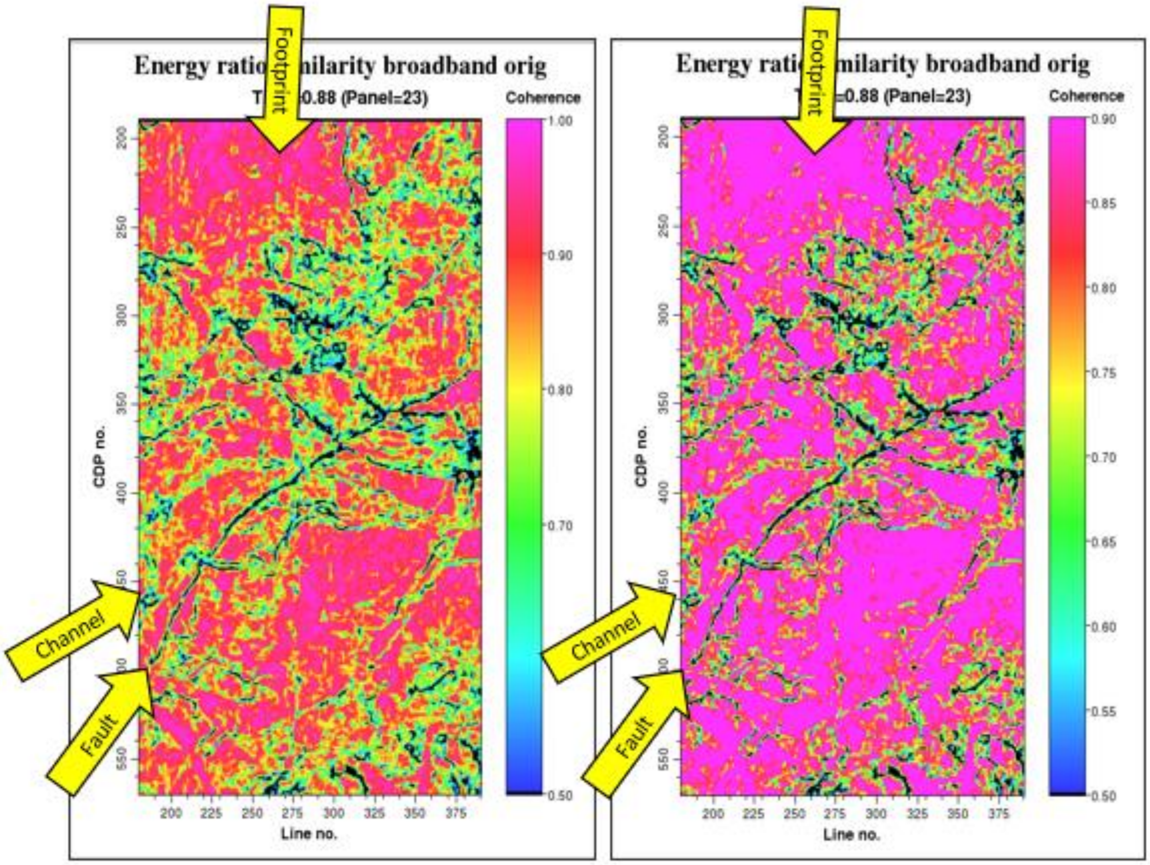
Estimating filter cutoff values s_{low}, s_{high}, and s_{center}

Most interpreters, particularly geologists, are surprised to learn that despite all the sophisticated mathematics, that many, if not most processing parameters are chosen subjectively. Such is the case with the parameters of s_{low}, s_{high}, and s_{center} . The figure below shows a time slice through a legacy seismic data volume acquired offshore Louisiana, USA. The energy ratio similarity time slice on the left is shown plotted using a conventional gray scale with the color bar ranging between 0.5 and 1.0 . Note that there are several faults and channels delineated by the data. There is also relatively weak, but annoying N-S trending acquisition footprint. Our goal is to choose the three parameters s_{low}, s_{high}, and s_{center} that preserve the low coherence geologic features of interest, suppress acquisition footprint, and enhance more subtle geologic features.



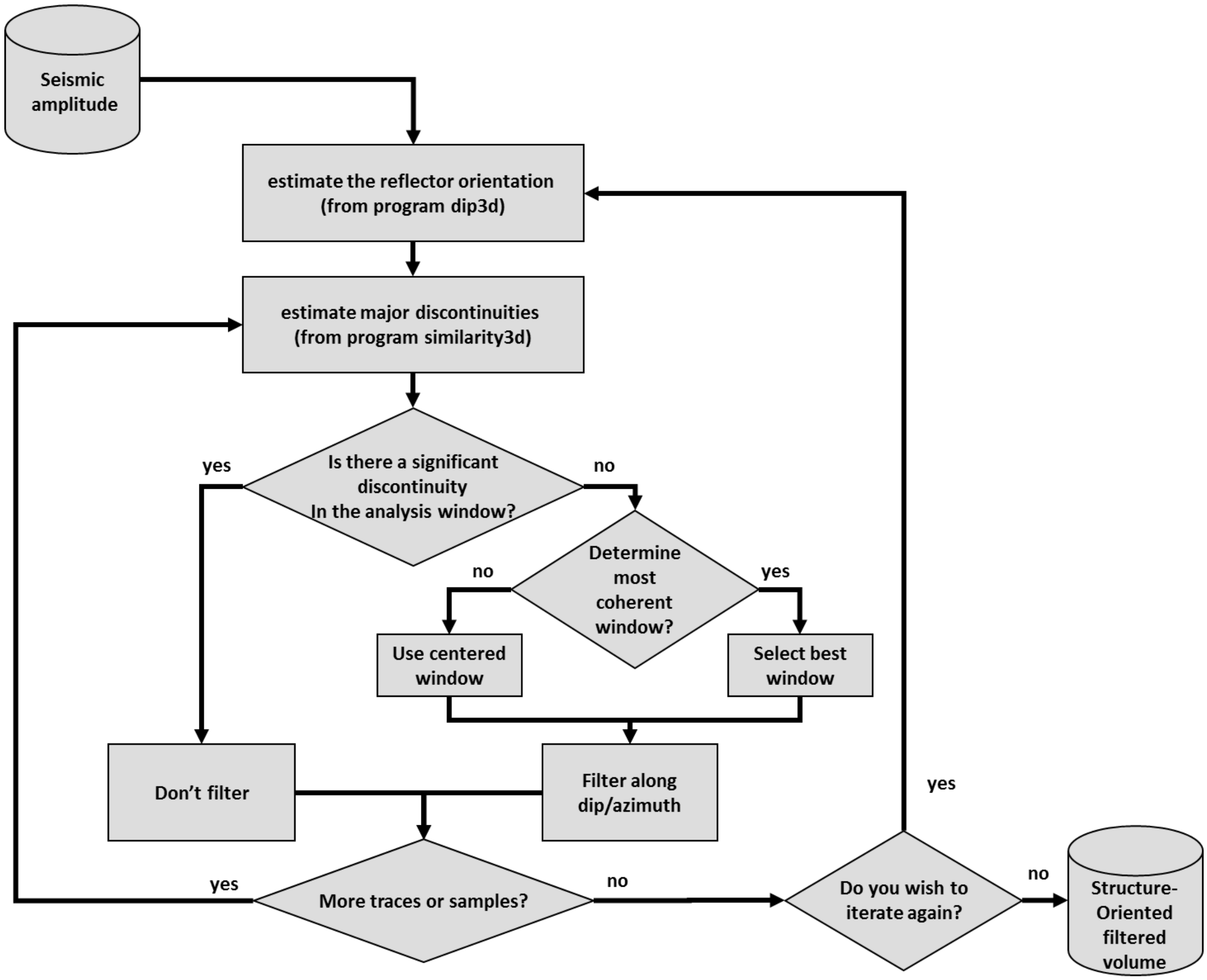
,

Although monochromatic color bars are superior in delineating edges, it does not help us in defining these three threshold values. I therefore choose a rainbow color bar and obtain the image on the lower left. By modifying the lower value of the color bar, I find that a value of 0.5 results in the stronger geologic edges appearing as blue. However, the I still see the weaker N-S trending footprint anomalies appearing as orange and yellow. I test several upper values of the color bar and find that by setting it to 0.9 that most of the footprint is relatively invisible and now appears as magenta.



With this simple exercise, I set my threshold values to be s_{low}=0.4 s_{high}=0.5, and s_{center}=0.9.

Davogustto (2011) summarizes the edge-preserving structure-oriented workflow using the following flow chart:



Desired output volumes and their filter parameters

The final section of the *Primary parameters* tab is to choose the filters to be applied. There are four basic filters: the (16) principal component filter, the (17) alpha-trimmed mean filter, the (18) lower-upper-middle (LUM) filter, and the (18) the mean filter. Note that a median filter is a subset of the alpha-trimmed mean filter when we set the rejected data on each side to be 50%. If we set the rejected data to be 0%, we obtain the mean filter. It is good practice to examine the noise that has been rejected by a chosen filter to assure that important geological features have not been significantly suppressed. To do so, simply place a check mark after (18) *Output rejected noise for each selected filter*. The filter parameters are defined in the following theory boxes.

Theory: Review of linear and nonlinear filters

Let's assume we have *J* voxels that fall within a 2D or 3D analysis window. There are several linear and nonlinear filters that can be applied.

The mean filter

The mean filter is the simplest, where the mean μ of *J* samples d_j is defined as:

$$\mu = \frac{1}{J}\sum_{j=1}^{J} d_j \ . \tag{1}$$

The mean filter is a smoothing filter and may not only smooth across faults but smooth in erroneous spikes into the output.

The median filter

The first step of the median filter is to sort the data vector, **d**, into a new vector **u** where $u_k \le u_{k+1}$:

$$\mathbf{u} = \mathrm{sort}\{d_1, d_2, \ldots, d_j, \ldots, d_{J-1}, d_J\}. \tag{2}$$

Then the median, *m*, is defined as:

$$m = u_{(J+1)/2} \ . \tag{3}$$

The median filter is an edge-preserving filter and will preserve changes in dips across faults. It also rejects erroneous spikes in the input data.

The α-trimmed mean filter

The α-trimmed filter is an extension of the median filter. First, the algorithm sorts the data in ascending order as in equation 2. Then one defines a fraction (usually defined as a percentage) of the data that falls within the range

$$0 \le \alpha \le \frac{1}{2} \ . \tag{4}$$

The filter rejects αJ "outliers" on each end of the data vector and computes the mean of the values of u_j with indices $1+\alpha J \le j \le J-\alpha)(J-1)$:

$$u_{\alpha-trim} = \frac{1}{J-2\alpha(J-1)} \sum_{j=1+\alpha(J-1)}^{J-\alpha(J-1)} u_j \ . \tag{5}$$

The alpha-trimmed mean filter thus rejects outliers and smooths the remaining values. As such it may still smooth changes in dip across faults.

The Lower-Upper-Middle (LUM) filter

The LUM filter is the default filter in **filter_dip_components** and acts in the following manner:

$$u_{LUM} = \mathrm{median}\left(u_{1+\alpha(J-1)}, u^*, u_{J-\alpha(J-1)}\right) = \begin{cases} u_{1+\alpha(J-1)} & u^* < u_{1+\alpha(J-1)} \\ u_{J-\alpha(J-1)} & u^* > u_{J-\alpha(J-1)} \ \ 0 \le \alpha \le 0.5 \\ u^* & otherwise \end{cases} \tag{6}$$

Like the alpha-trimmed mean filter, the LUM filter rejects high and low amplitude "outliers". Instead of taking the mean of the remaining samples, it compares the dip value at the center of the analysis window u* to the upper and lower percentiles. If u* falls beyond these percentiles, it clips the value to the upper or lower percentile; otherwise, it leaves the value alone. In this manner, the LUM filter preserves detailed variation, but rejects erroneous values.

Theory: An overview of principal components

Principal components provide a means of identifying a consistent amplitude pattern (think "signal") that repeats, sample by sample, within an analysis window. For ease of visualization let's examine a (very large) 21x21 inline by crossline patch of seismic amplitude extracted parallel to dip and azimuth. Such a patch forms a 441 long "sample vectors" of the seismic amplitude data (Kirlin and Done, 1999). In order to best see the pattern of the signal through the incoherent noise, we need to examine more than one sample vector. In satellite imagery, we might take multiple snapshots of a fixed patch of the earth over several days. The "amplitude" of the snapshot will change due to different illumination at 9 AM, 12 noon and 5 PM. Likewise, the ground surface itself may be partially obscured by clouds, the location of which may appear to be random at each satellite pass over our patch of earth. The underlying spatial pattern – rivers, roads, forest and prairie will remain fixed. In principle each snapshot should be correlated to all the others.

Sample Vectors

The covariance matrix is constructed from a suite of sample vectors. In **sof3d**, the "vectors" take the form of a suite of *2K+1 M*-trace maps parallel to structure, centered about the analysis point. The objective is to identify and then preserve vertically consistent patterns across the 2*K*+1 local amplitude maps. To achieve this objective, one needs to compute a covariance matrix.

The Covariance Matrix

The covariance matrix, **C**, is constructed by comparing each sample vector to itself and all its neighbors**.** AASPI applications use not only the 2*K+1 M*-trace sample vectors through the original seismic amplitude, **d**, but also an additional *2K+1* sample vectors through its Hilbert transform, $\mathbf{d}^{\mathbf{H}}$:

$$C_{mn} = \sum_{k=-K}^{K}\left[d(t_k, x_m, y_m)d(t_k, x_n, y_n) + d^H(t_k, x_m, y_m)d^H(t_k, x_n, y_n)\right]. \qquad (7)$$

These additional (90^0-phase rotated) sample vectors fall in the same window and thus do not modify the vertical resolution. However, they ameliorate areas of low signal-to-noise ratio about zero crossings, where the original absolute amplitude is smallest, but also where the corresponding Hilbert transform is largest.

Eigenvectors and Eigenvectors

Any matrix can be decomposed into eigenvalues and eigenvectors (Kirlin and Done, 1999). The covariance matrix described by equation 7 will be real, square, and symmetric, resulting in non-negative eigenvalues. For square matrices, one can write:

$$\sum_{m=1}^{M} C_{nm} v_m^{(k)} = \lambda_k v_n^{(k)}, \qquad (8)$$

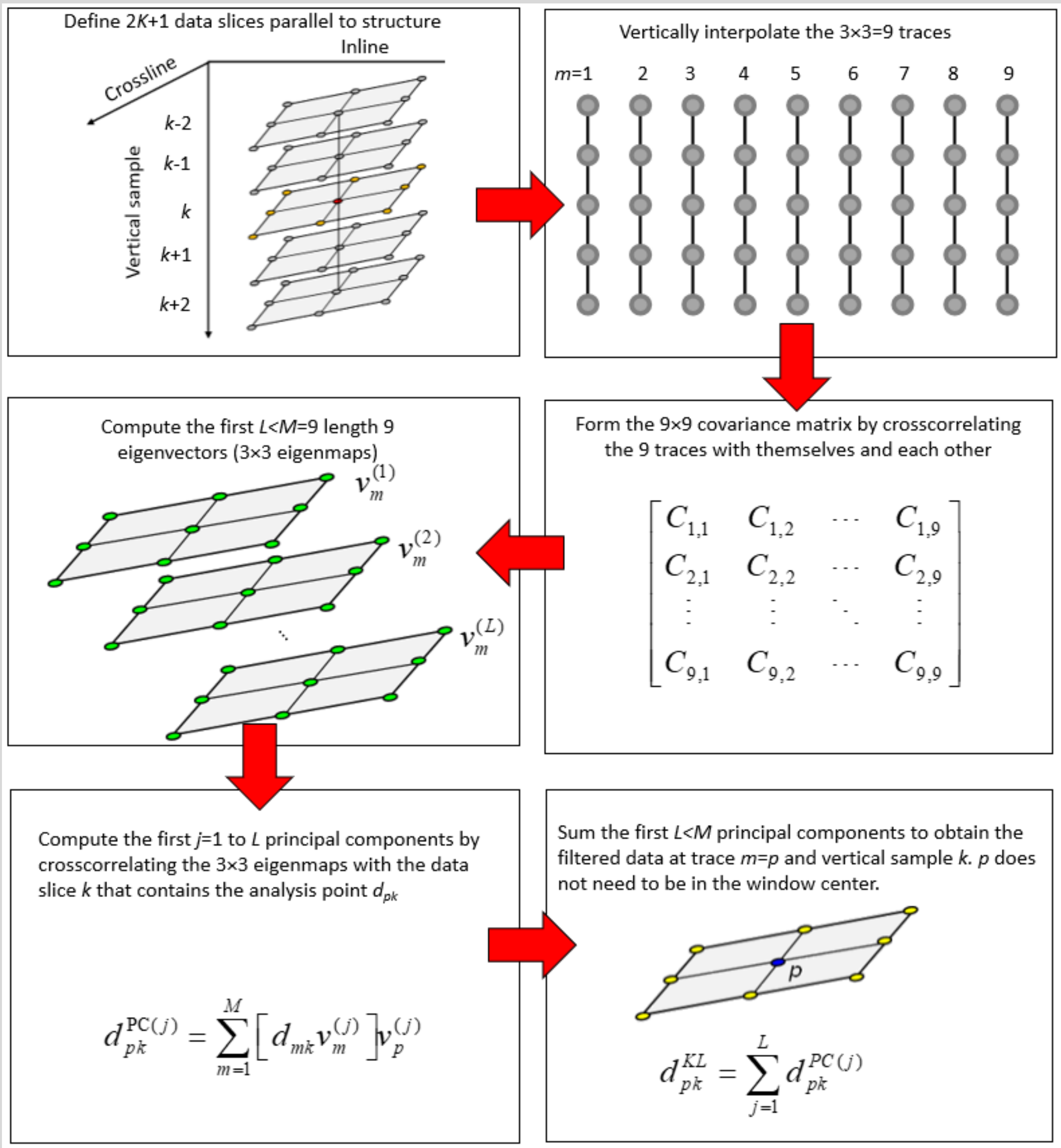
where λ_k is the k^{th} eigenvalue and $v_n^{(k)}$ is the corresponding unit-length eigenvector, or "eigenmap" of the data patterns.

Principal Components

Principal components are scaled versions of the eigenvectors. For the j^{th} principal component the scale factor is the inner product or correlation of the j^{th} eigenvector with a sample vector (*M*-trace data slice) that contains the analysis point (t_0, x_n, y_n) at the center of the window:

$$d_{PC}^{(j)}(t_0, x_n, y_n) = \sum_{m=1}^{M}\left[v^{(j)}(x_m, y_m)d(t_0, x_m, y_m)\right]v^{(j)}(x_n, y_n). \qquad (9)$$

3D analysis windows, covariance matrices, and principal components

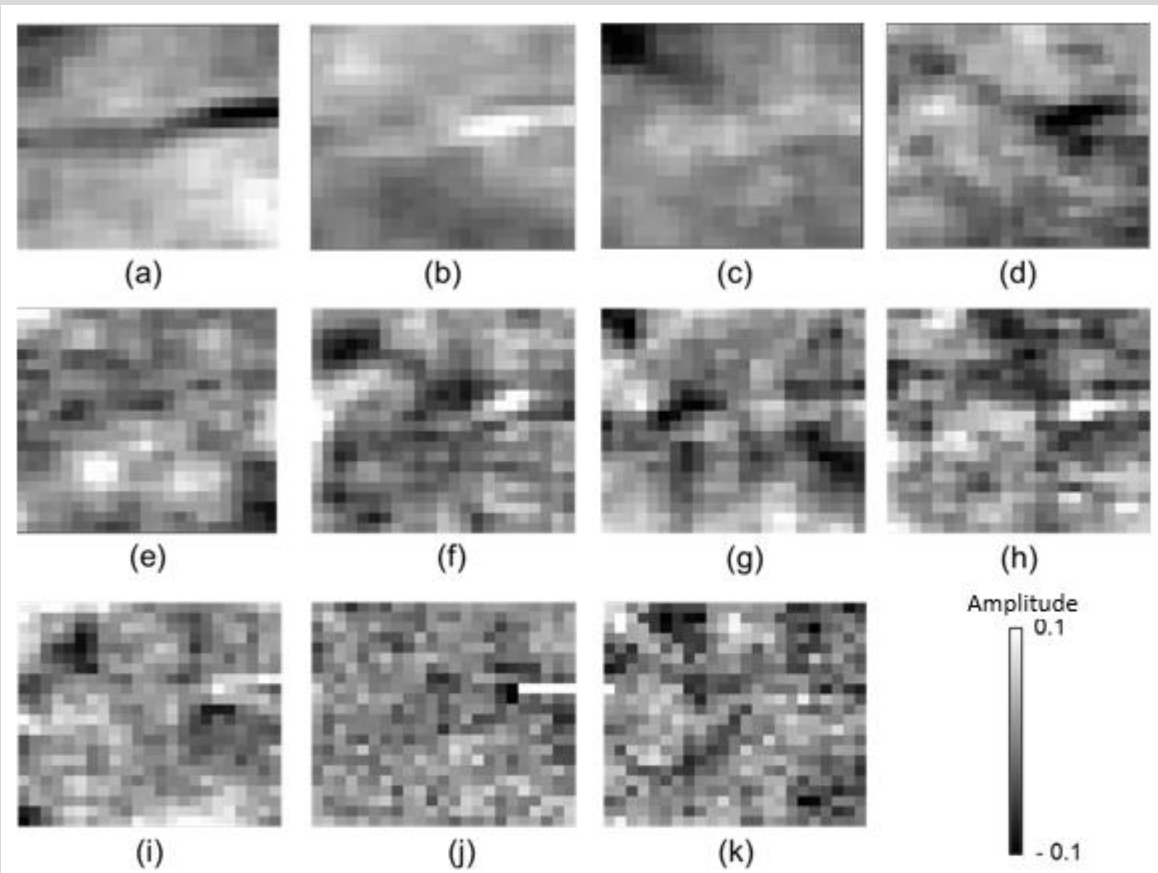


Cartoon showing structure-oriented filtering applied to an offset- or azimuth-limited stacked data volume along structural dip using a centered analysis window about the red analysis point. In this example there are 3 crosslines by 3 lines resulting a length M=9 “sample vector” for each interpolated dipping horizon slice at time k. These sample vectors are cross-correlated and averaged from k=-K to k=+K (K=2) time samples resulting in a 9×9 covariance matrix described by equation 7. The first L length 9 eigenvectors represent 3×3 “maps” that best represent the lateral variation of amplitude within the analysis window. These eigenmaps are cross-correlated with the sample vector at time k to compute a suite of L principal components. One or more of these components are then summed to form the filtered data at the analysis point at trace $m=p$ and vertical sample k.

Theory: An example of eigenvectors and principal components

In his MS thesis, Davogustto (2011) illustrates the concepts of patterns seen in seismic data using 11 21 trace by 21 trace sample vectors (K=5 and M=441 in equation 7) oriented along structure. Vertical changes in the wavelet amplitude from peaks, troughs, and zero-crossings exhibit a similar spatial pattern but different brightness, not unlike the varying brightness of the satellite images at different times of the day discussed in the previous box. The m^{th} row and n^{th} column C_{mn} is the cross-correlation from $K=-5$ to $K=+5$ of the n^{th} trace with the m^{th} trace in the analysis window.

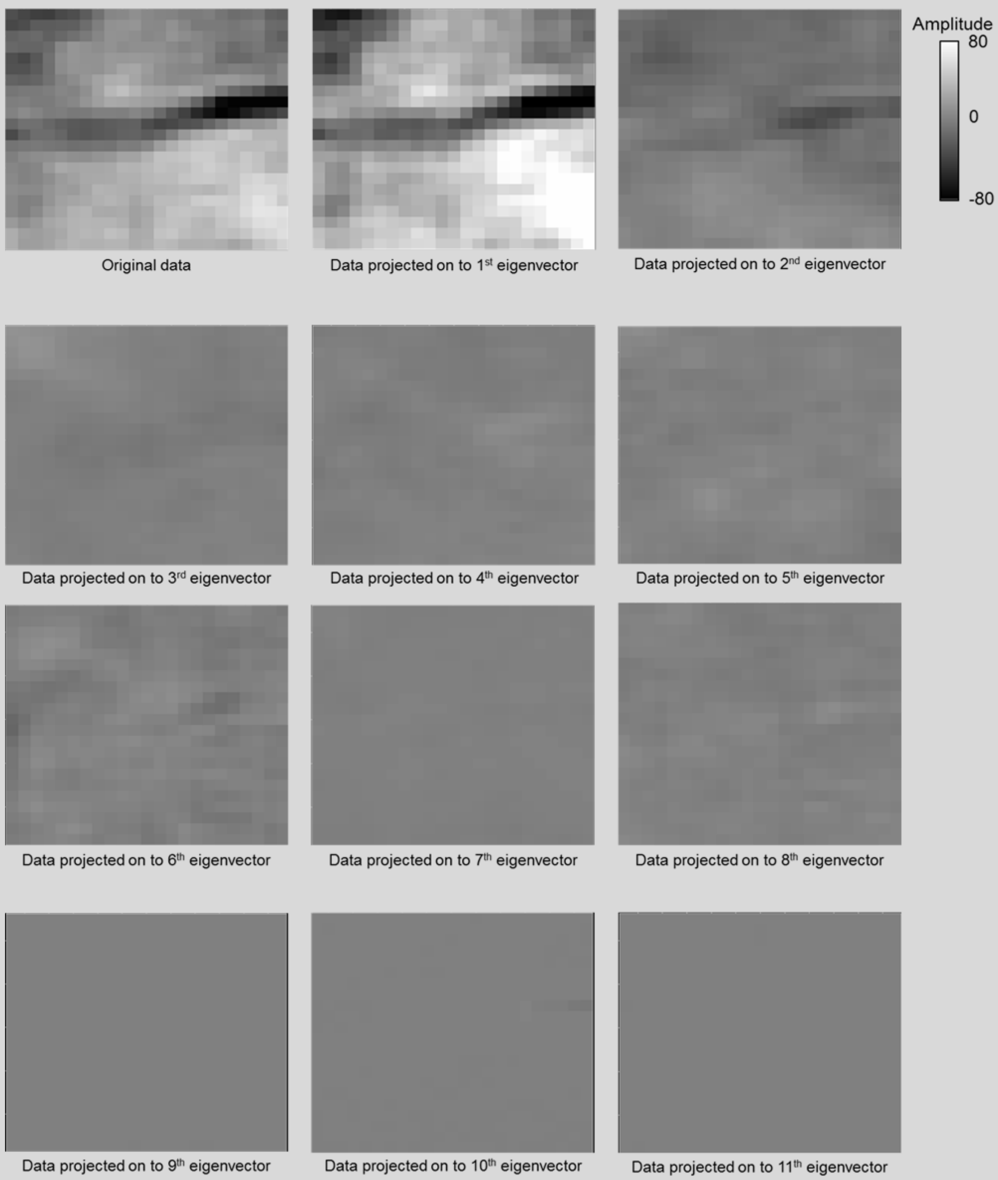
Unfortunately, we were not prescient enough to keep the original 11 sample vectors. However, we did keep images of the first 11 of the 441 eigenvectors of the 441×441 covariance matrix **C:**



By definition, the first eigenvector $\mathbf{v}^{(1)}$ shown in (a) best represents the variability of the seismic amplitude of the 11 sample vectors (Kirlin and Done, 1998). In this example, the first four eigenvectors or "eigenmaps" represent relatively smoothly varying reflectivity. In contrast eigenvectors 9, 10, and 11 as shown in the subfigures (i), (j), and (k) are more random and represent noise in the data.

An example of eigenvectors and principal components (continued)

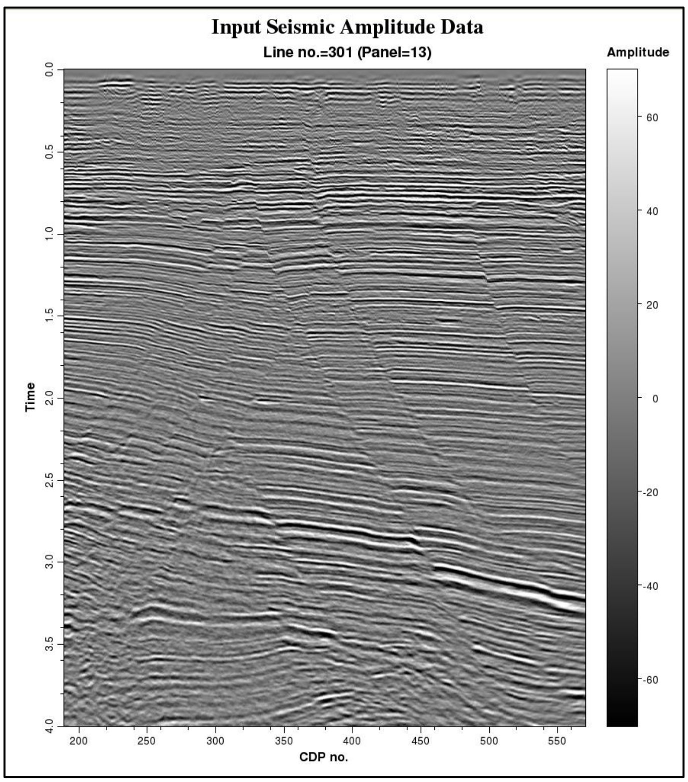
Principal component analysis has entered the seismic processing world from many directions and therefor has many names including eigenstructure, eigenvalue-eigenvector, singular value decomposition (SVD), and Karhunen–Loève transform analysis, causing unnecessary confusion. The eigenvectors $\mathbf{v}^{(j)}$ of the covariance matrix **C** are by construction unit length and orthogonal, such that they can form a basis function much like those used in the k_x-k_y transform. Using equation 9, the first 11 of the total 441 "principal components" of the mapped data *u(t=0,x,y)* along the horizon slice are obtained by cross-correlating *u(t=0,x,y)* with $v^{(j)}(x,y)$ (where *j* varies between 1 and 441) are shown in the figure below. Note that most of the amplitude is represented by the first two eigenvectors.

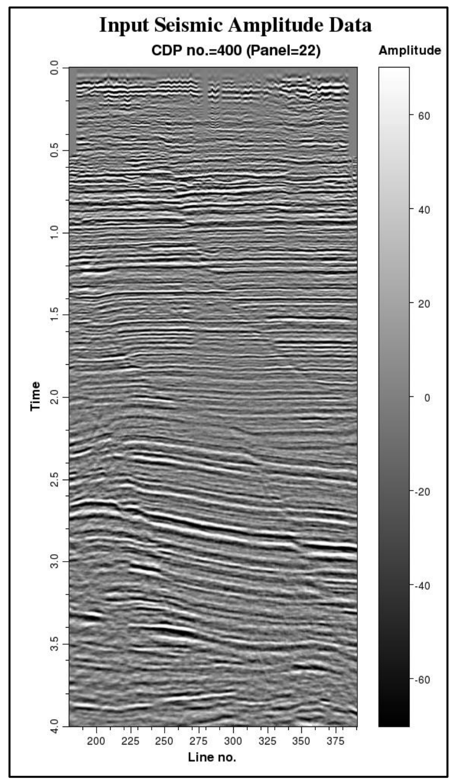


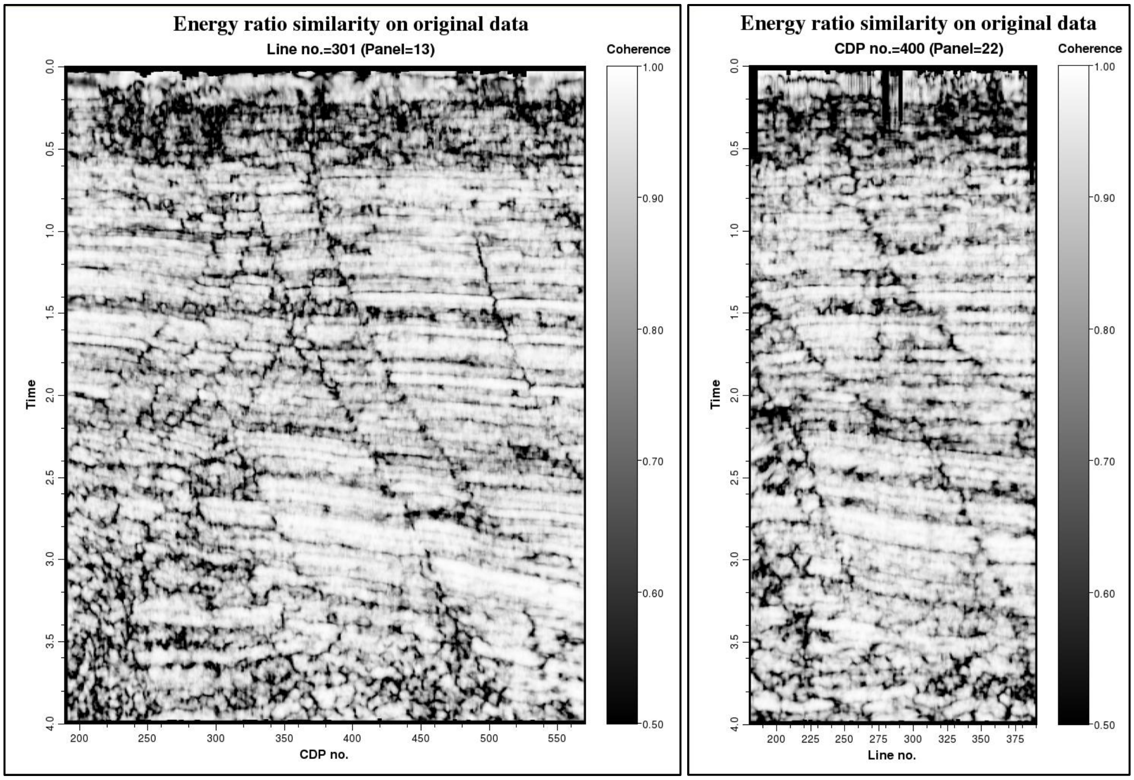
Example: SOF applied to a legacy Gulf of Mexico survey

The westcam survey (for West Cameron Island, LA, USA) was acquired in the early 1990s using a narrow azimuth streamer acquisition. The survey exhibits channels, listric faults, bright spots, and fairly strong acquisition footprint in the sail (inline) direction. Because of the moderate data quality, it provides a good data set to test alternative structure-oriented filtering strategies. For each filter option, I will provide a vertical and time slice of the filtered data, of the rejected noise, and of the coherence volume computed from the filtered data. Recall from Sheffield and Payne (2008) that vertical slices in the crossline direction will exhibit a greater degree of acquisition footprint.

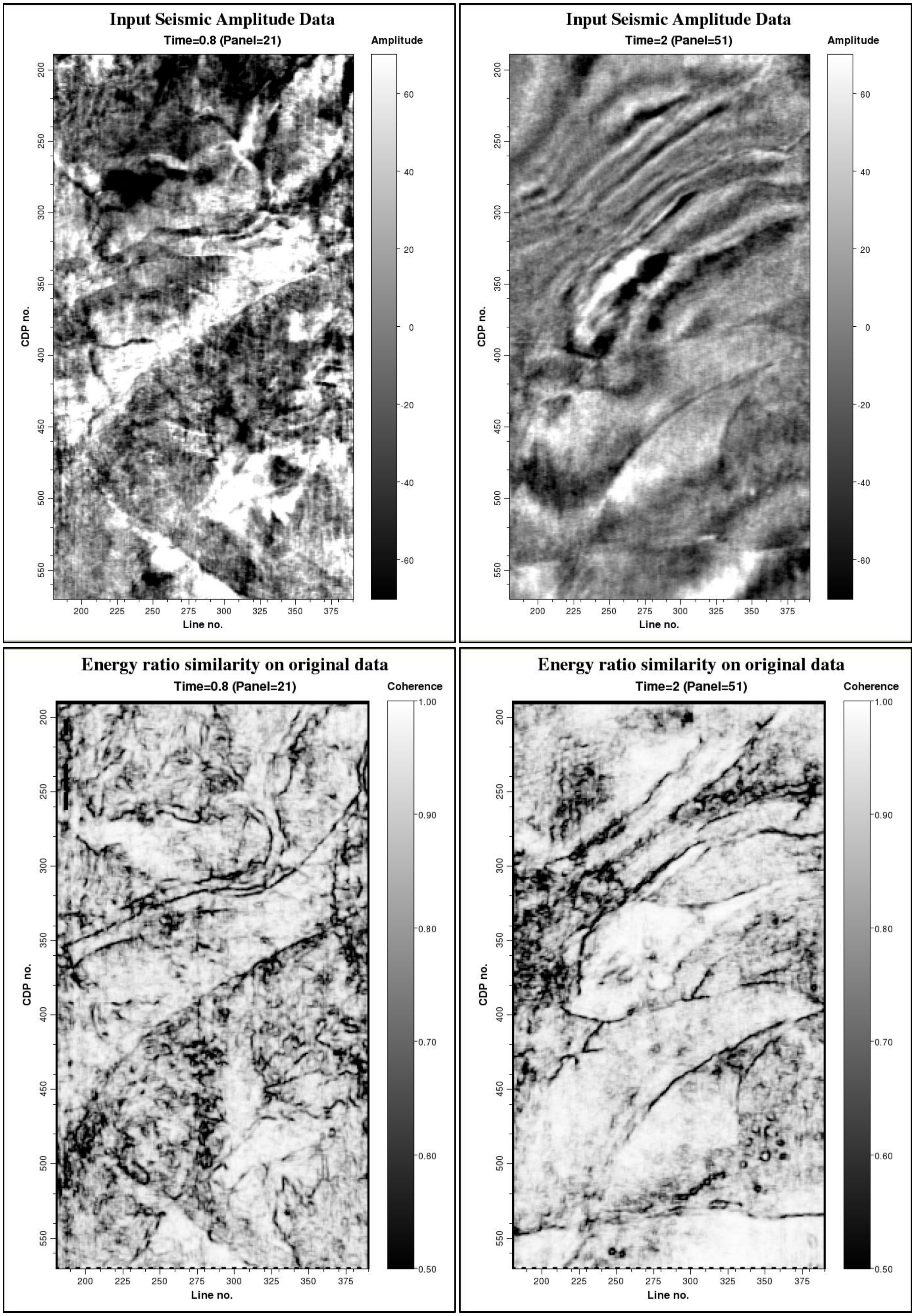
The original data





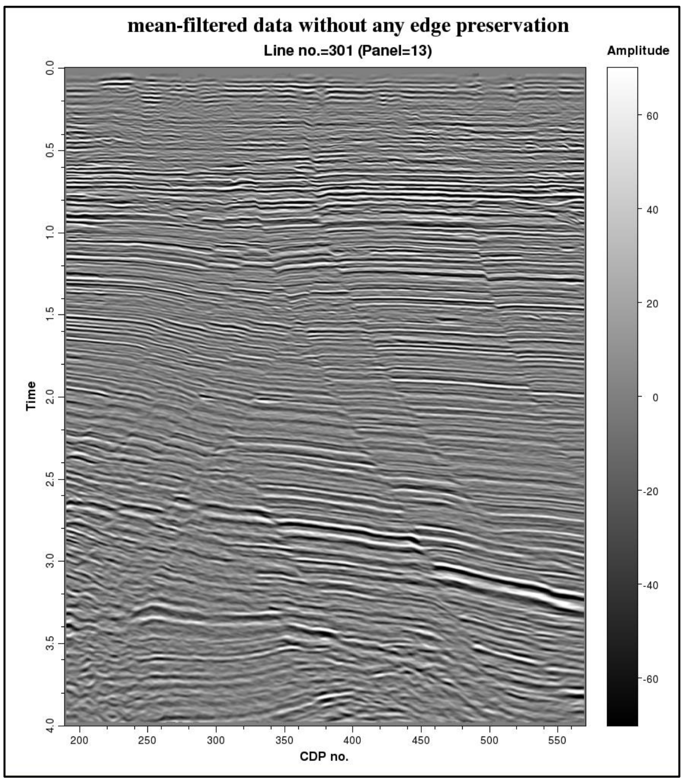


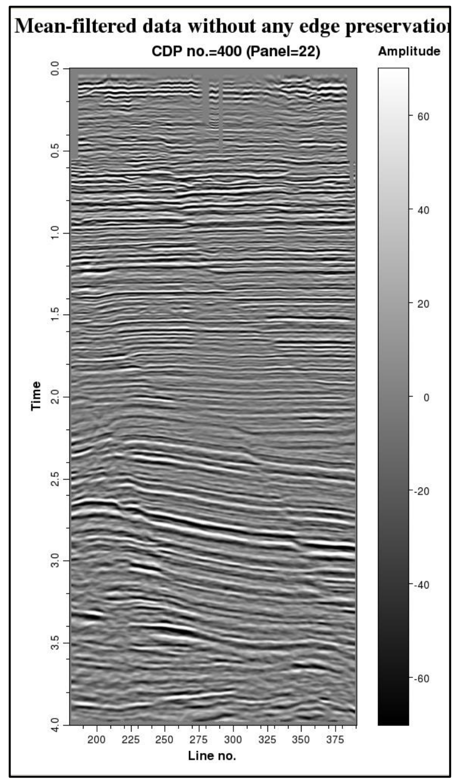
Energy ratio similarity on original data
Line no.=301 (Panel=13)
Coherence
Time
CDP no.
Energy ratio similarity on original data
CDP no.=400 (Panel=22)
Coherence
Time
Line no.

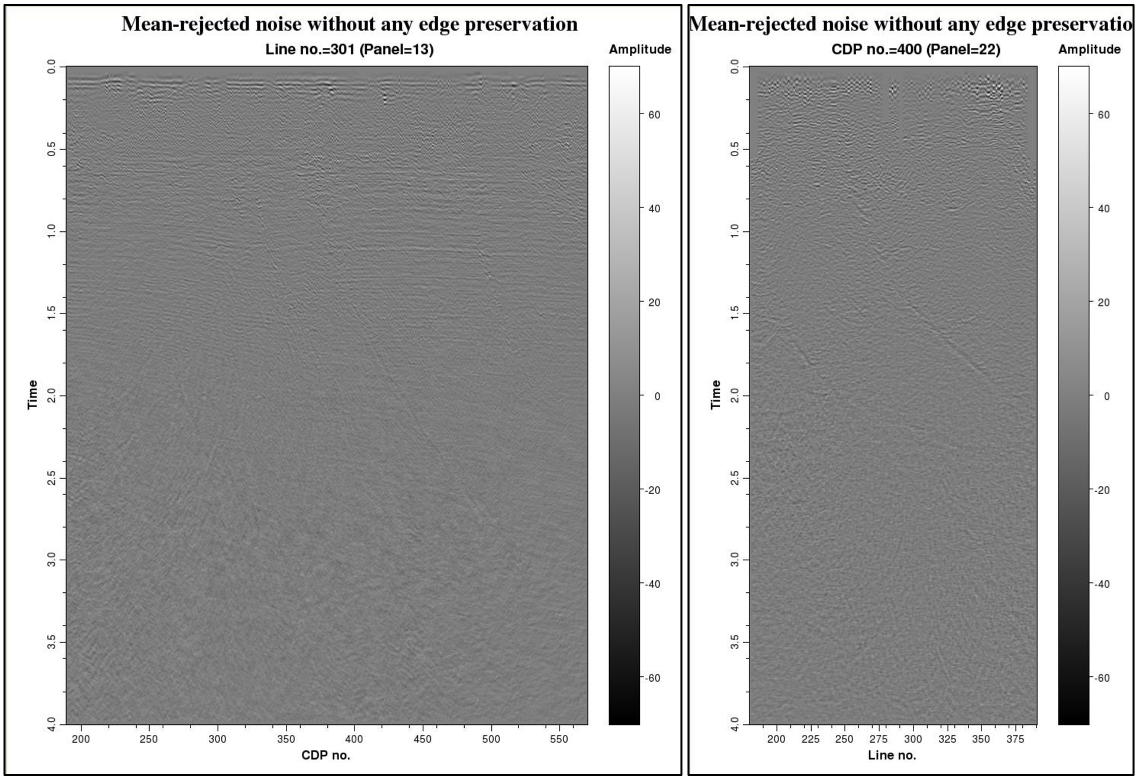

Input Seismic Amplitude Data
Time=0.8 (Panel=21)
Amplitude
Input Seismic Amplitude Data
Time=2 (Panel=51)
Amplitude
CDP no.
Line no.
Energy ratio similarity on original data
Time=0.8 (Panel=21)
Coherence
Energy ratio similarity on original data
Time=2 (Panel=51)
Coherence
CDP no.
Line no.

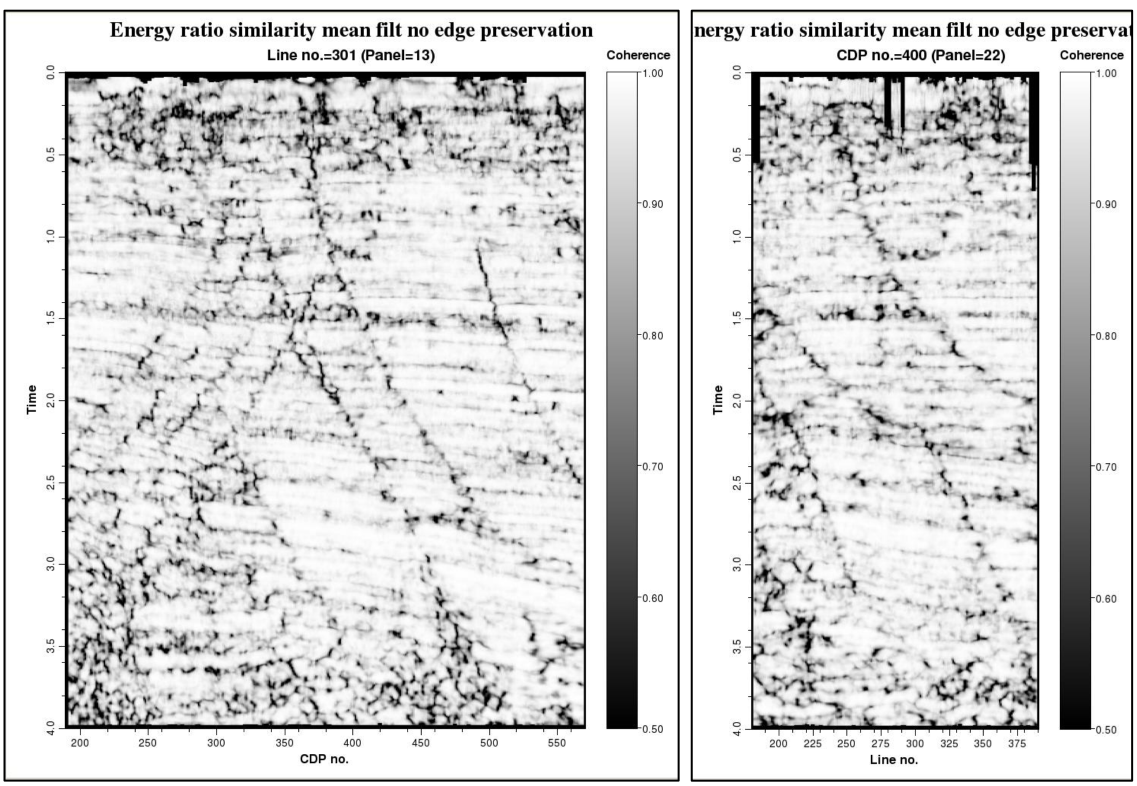
Mean filtering without edge preservation of any kind $s_{low}=0.0$, $s_{high}=0.0$, $s_{center}=0.0$

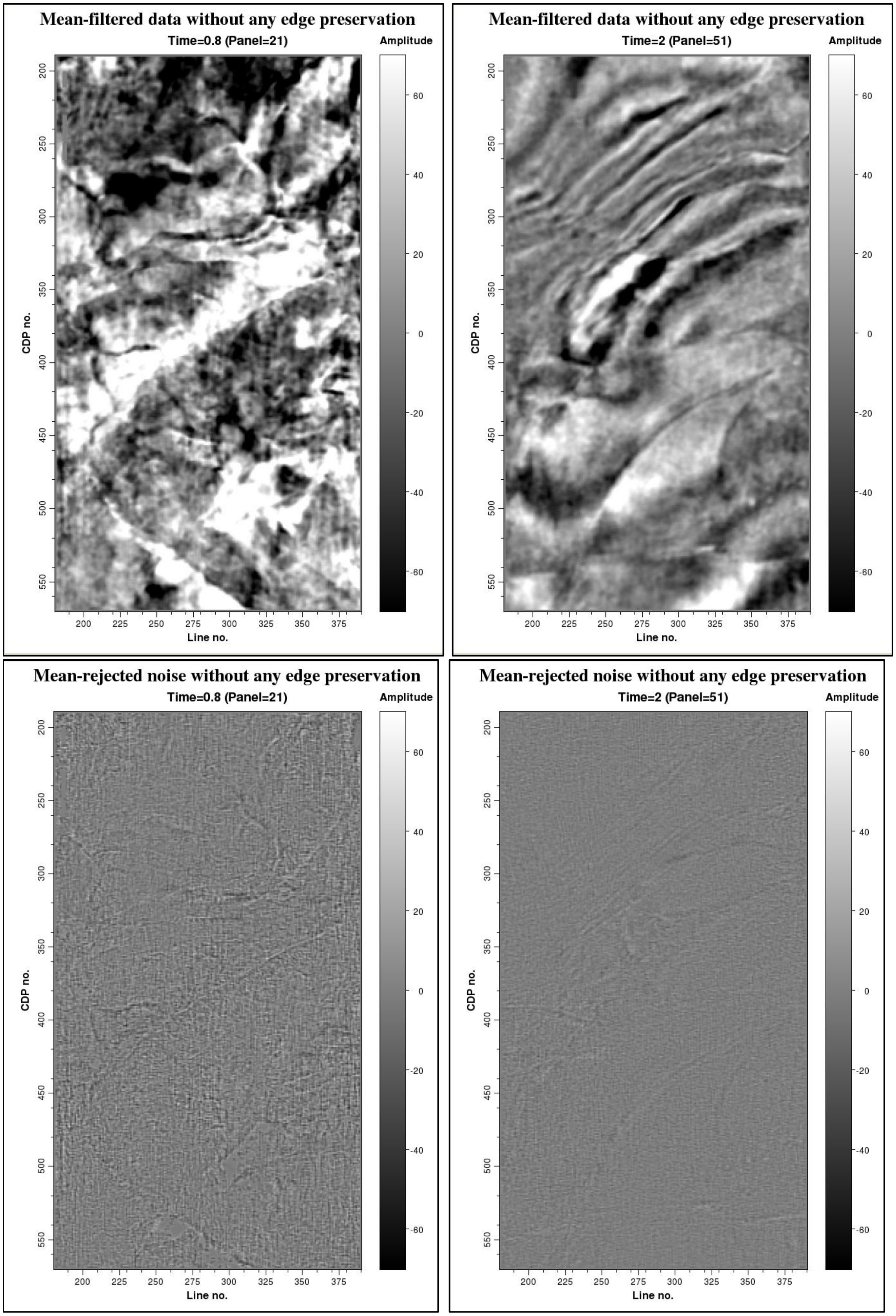
Now, let's apply a simple filter along structure, but not worrying about edges of any kind. I do not recommend this filter, but rather use it (like Fehmers and Höecker did), that edge preservation is important. To replicate the following images, set $s_{low}=s_{high}=s_{center}=0.0$ and place a checkmark in front of *Want mean-filtered data* and in front of *Output rejected noise for each selected filter*.

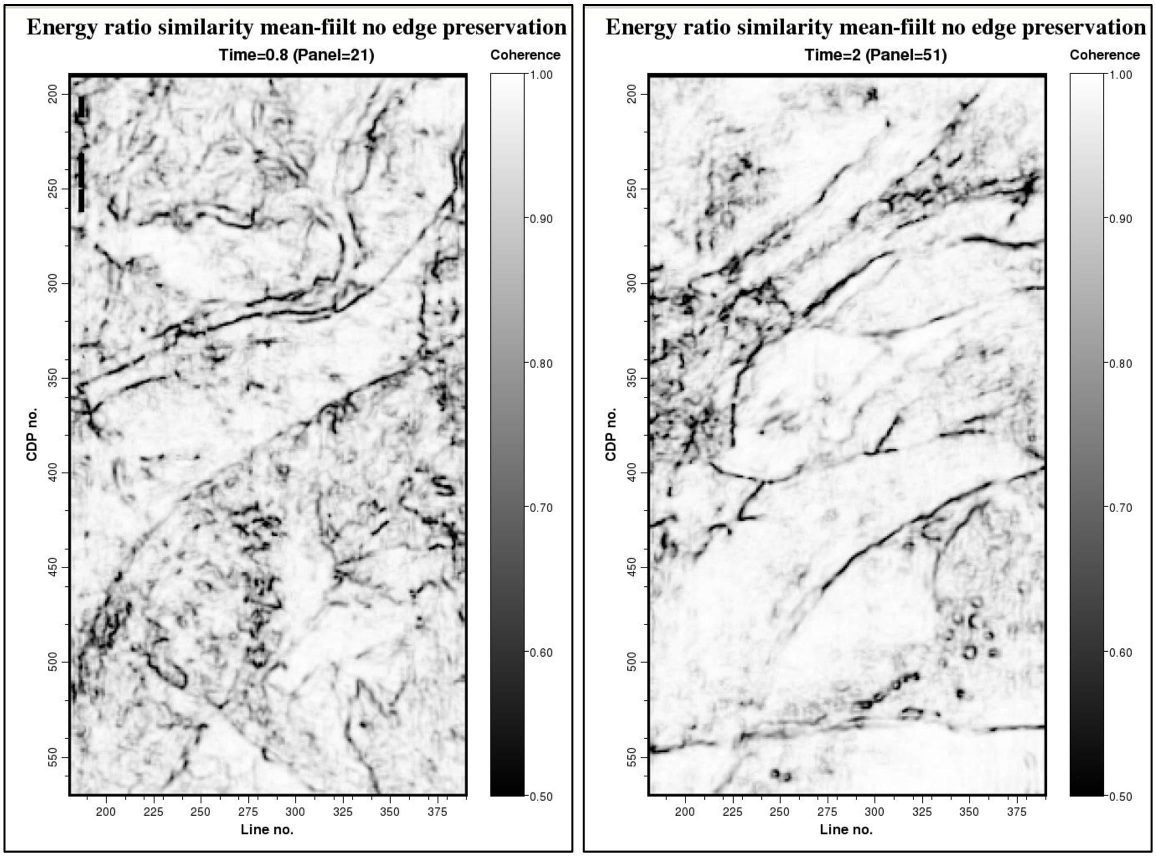




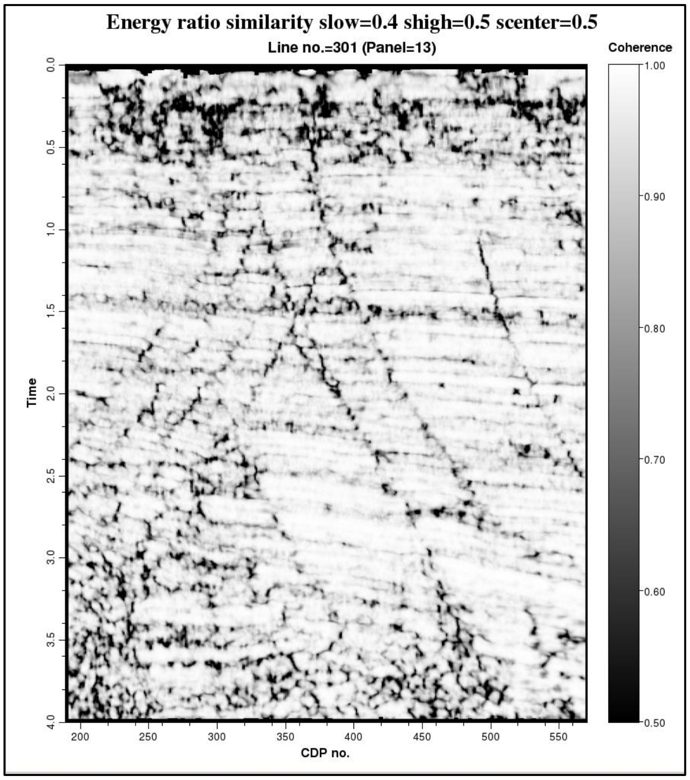

Mean-rejected noise without any edge preservation
Line no.=301 (Panel=13)
Amplitude
Time
CDP no.
Mean-rejected noise without any edge preservatio
CDP no.=400 (Panel=22)
Amplitude
Time
Line no.

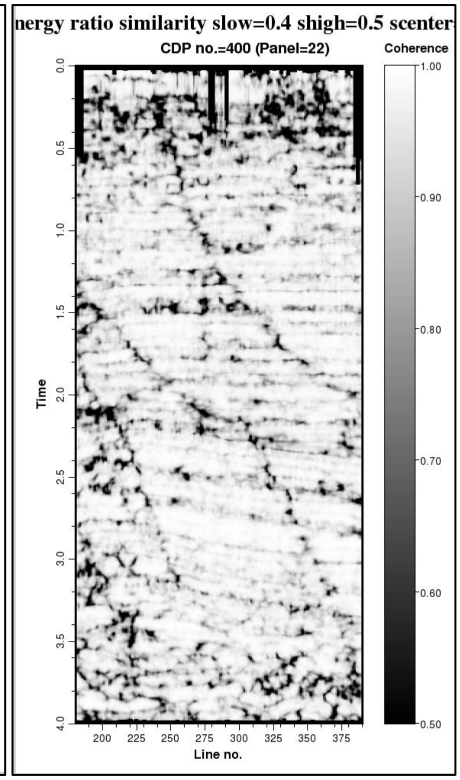

Energy ratio similarity mean filt no edge preservation
Line no.=301 (Panel=13)
Coherence
Time
CDP no.
nergy ratio similarity mean filt no edge preservat
CDP no.=400 (Panel=22)
Coherence
Time
Line no.

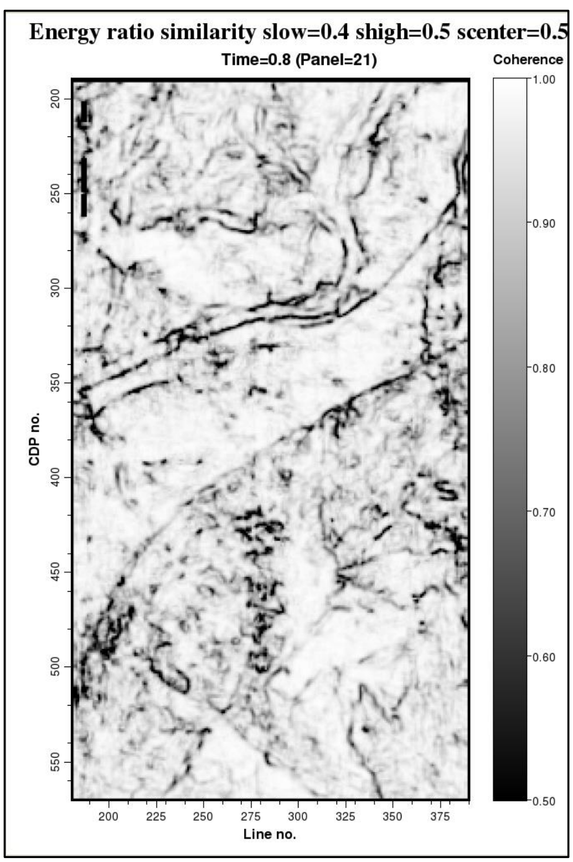

Mean-filtered data without any edge preservation
Time=0.8 (Panel=21)
Amplitude
CDP no.
Line no.
Mean-filtered data without any edge preservation
Time=2 (Panel=51)
Amplitude
CDP no.
Line no.
Mean-rejected noise without any edge preservation
Time=0.8 (Panel=21)
Amplitude
CDP no.
Line no.
Mean-rejected noise without any edge preservation
Time=2 (Panel=51)
Amplitude
CDP no.
Line no.

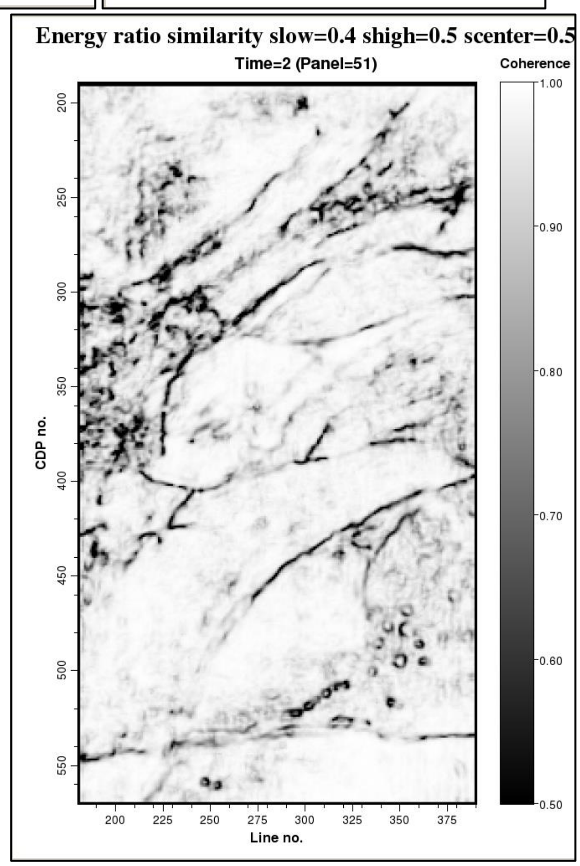

Energy ratio similarity mean-fiilt no edge preservation
Time=0.8 (Panel=21)
Coherence
CDP no.
Line no.
Energy ratio similarity mean-fiilt no edge preservation
Time=2 (Panel=51)
Coherence
CDP no.
Line no.

Mean filtering using the Fehmers and Höecker's (2003) workflow

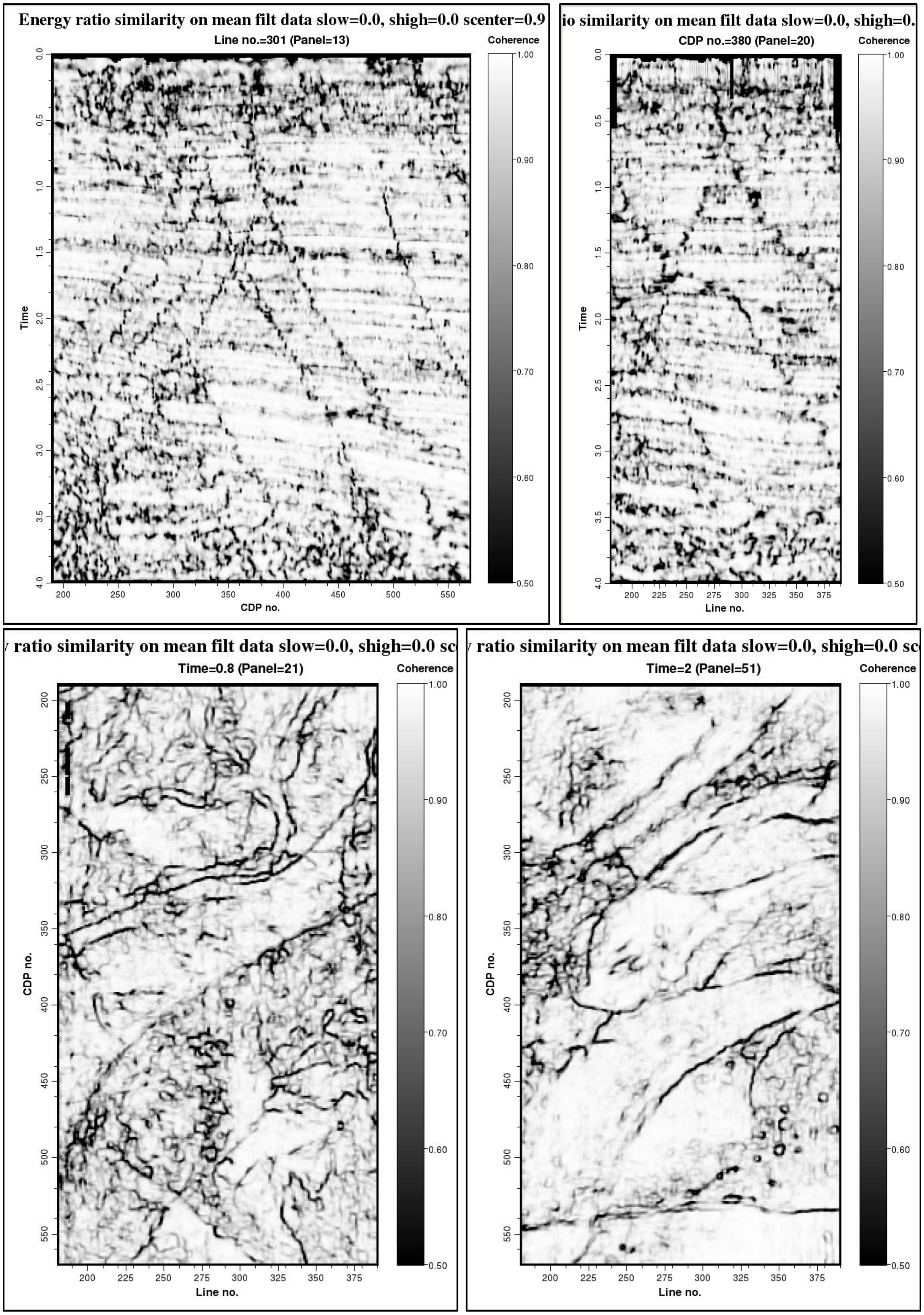






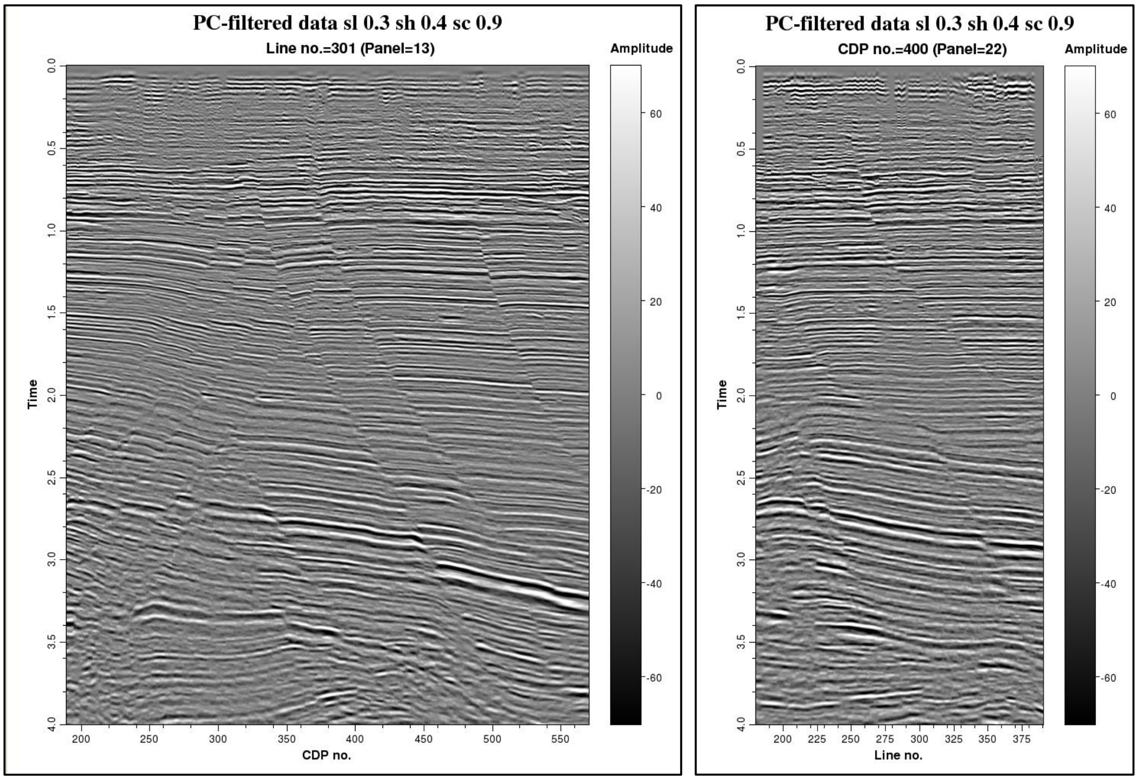


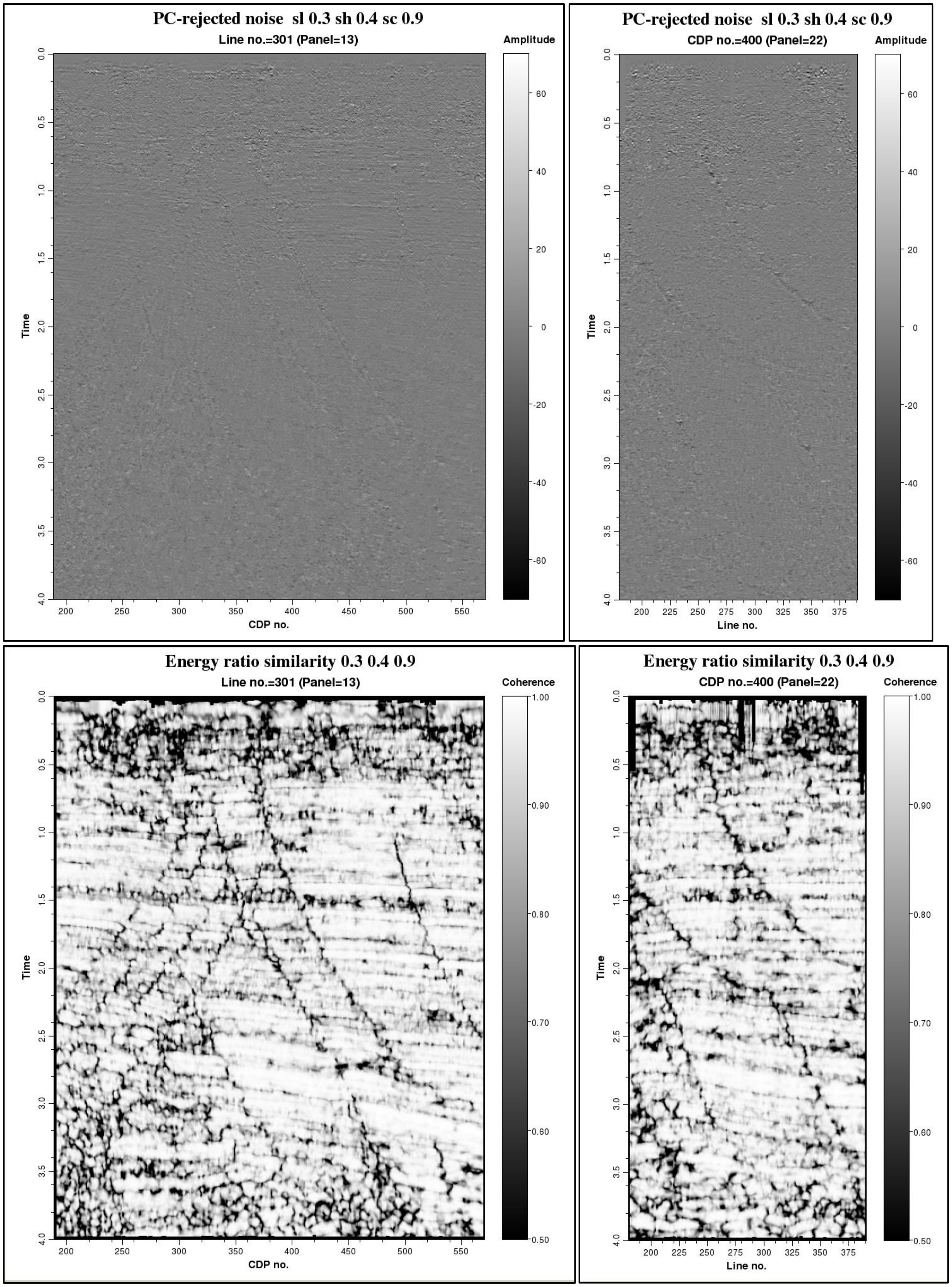
Mean filtering using Luo et al.'s (2006) Kuwahara filter workflow

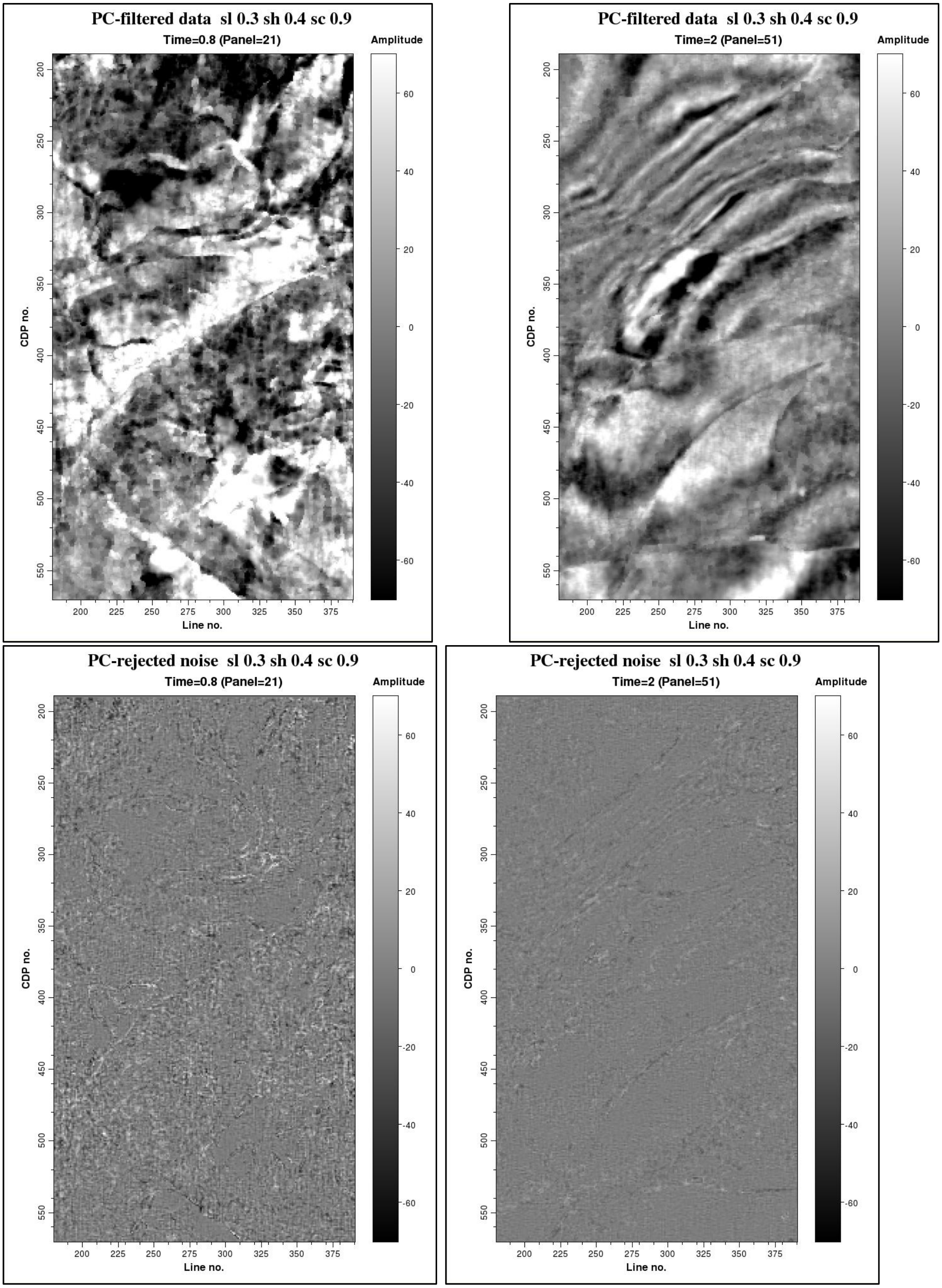


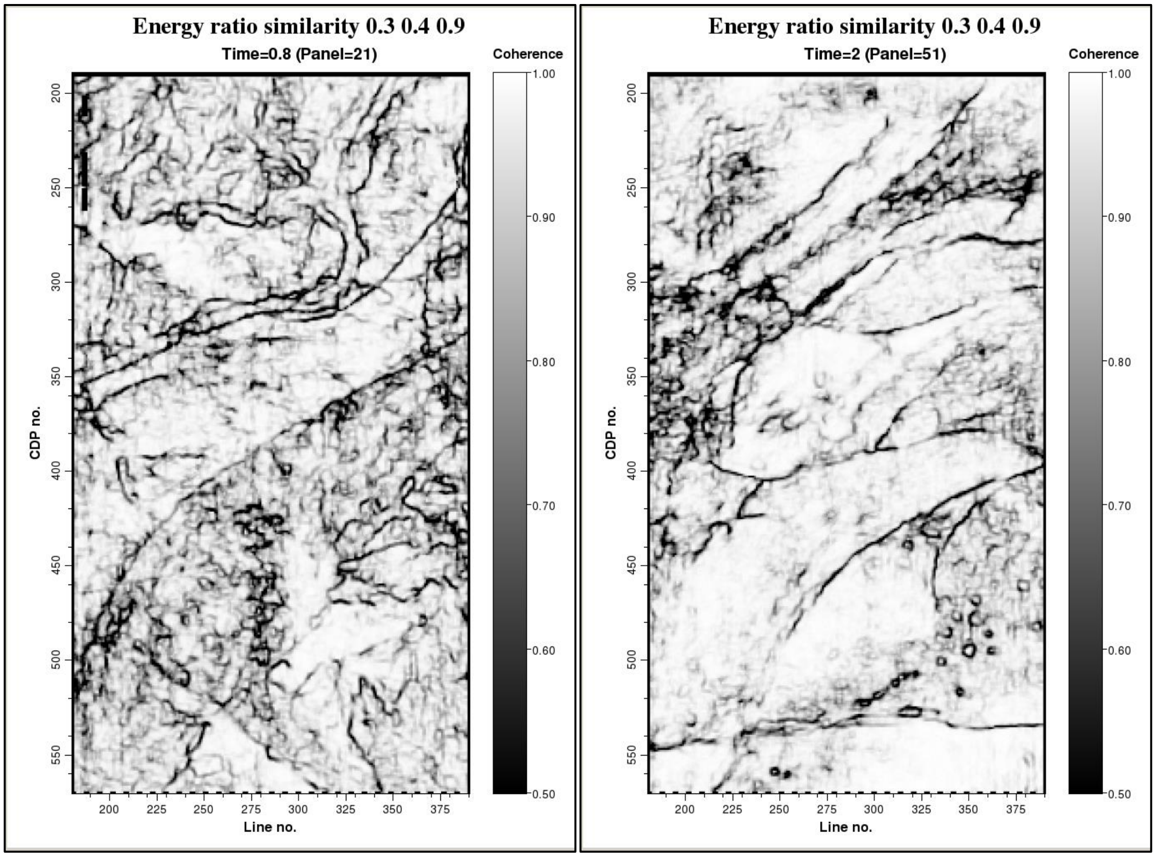
Mean filtering using both workflows

Principle component filtering ***s_{low}=0.3, s_{high}=0.4, s_{center}=0.9***

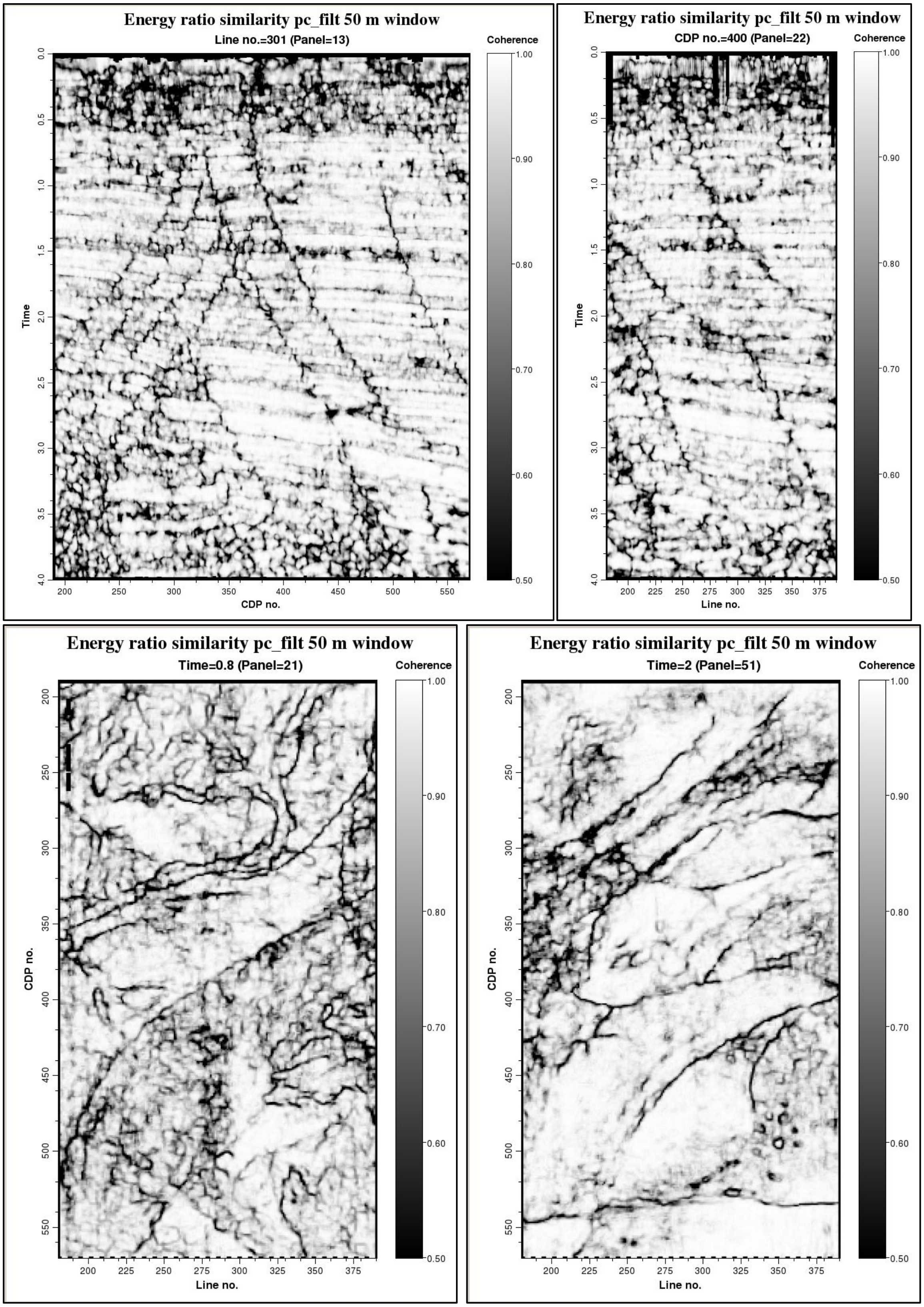



PC-rejected noise sl 0.3 sh 0.4 sc 0.9
Line no.=301 (Panel=13)
Amplitude
Time
CDP no.
PC-rejected noise sl 0.3 sh 0.4 sc 0.9
CDP no.=400 (Panel=22)
Amplitude
Time
Line no.
Energy ratio similarity 0.3 0.4 0.9
Line no.=301 (Panel=13)
Coherence
Time
CDP no.
Energy ratio similarity 0.3 0.4 0.9
CDP no.=400 (Panel=22)
Coherence
Time
Line no.

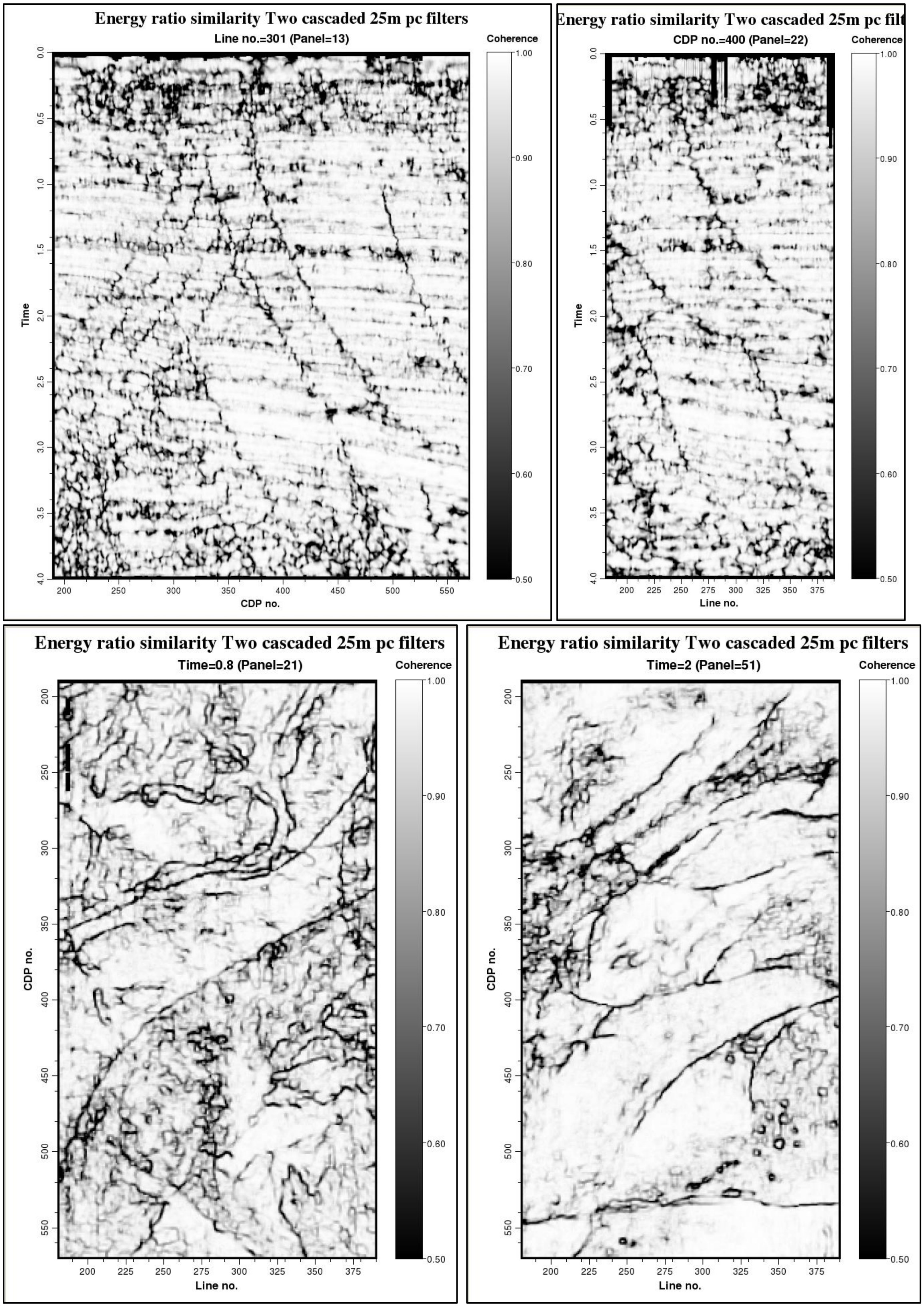

PC-filtered data sl 0.3 sh 0.4 sc 0.9
Time=0.8 (Panel=21)
Amplitude
CDP no.
Line no.
PC-filtered data sl 0.3 sh 0.4 sc 0.9
Time=2 (Panel=51)
Amplitude
CDP no.
Line no.
PC-rejected noise sl 0.3 sh 0.4 sc 0.9
Time=0.8 (Panel=21)
Amplitude
CDP no.
Line no.
PC-rejected noise sl 0.3 sh 0.4 sc 0.9
Time=2 (Panel=51)
Amplitude
CDP no.
Line no.



Using a single 50 m principal component circular filter


Energy ratio similarity pc_filt 50 m window
Line no.=301 (Panel=13)
Coherence
Time
CDP no.
Energy ratio similarity pc_filt 50 m window
CDP no.=400 (Panel=22)
Coherence
Time
Line no.
Energy ratio similarity pc_filt 50 m window
Time=0.8 (Panel=21)
Coherence
CDP no.
Line no.
Energy ratio similarity pc_filt 50 m window
Time=2 (Panel=51)
Coherence
CDP no.
Line no.

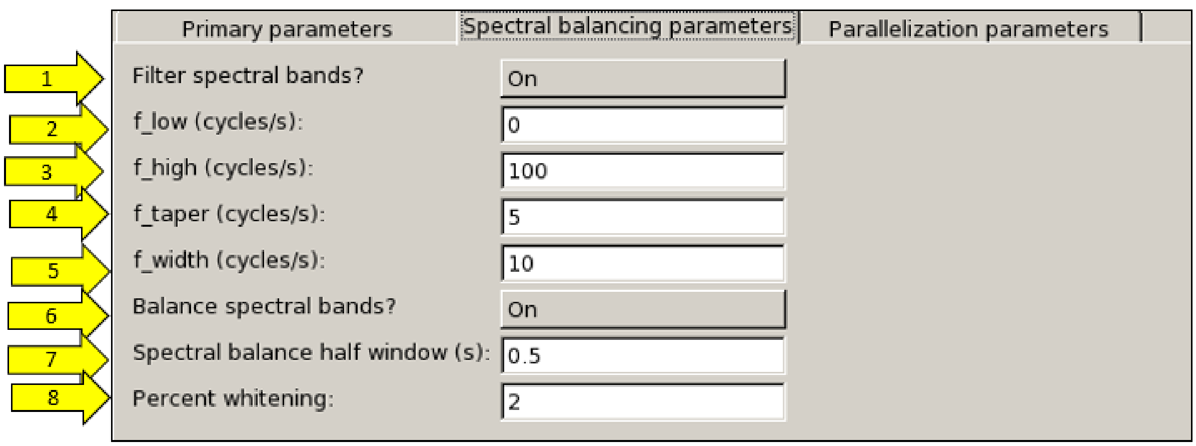
Using two cascaded 25 m principal component circular filters



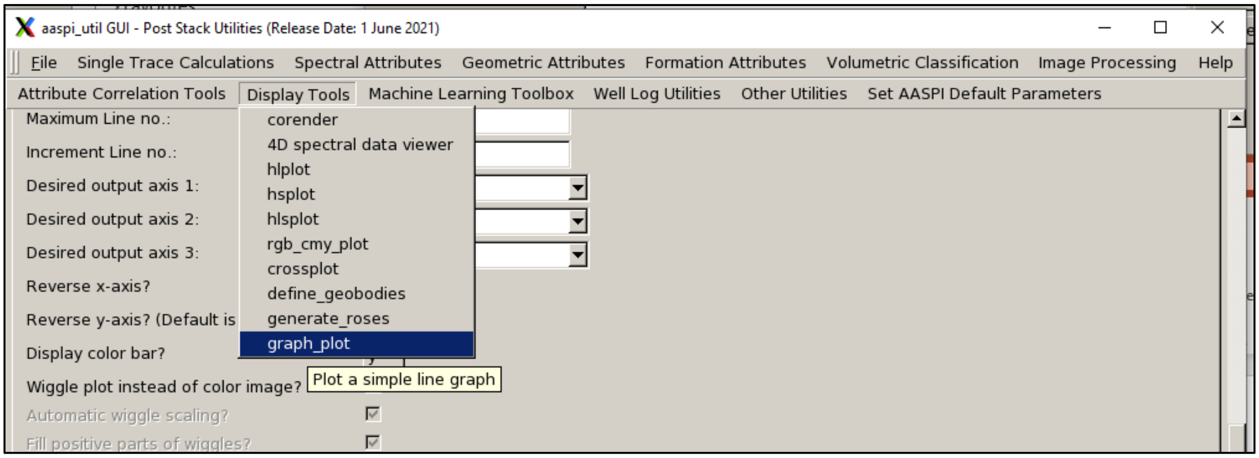
Spectral balancing parameters and structure-oriented filtering of band-pass filtered data

The default structure-oriented filter is all-pass in the frequency domain, ranging from 0 Hz to Nyquist. Helmore (2009) proposed a slightly different workflow that used a single dip-azimuth computation from the broad-band data but applied structure-oriented filtering to a suite of band-pass filtered version of the seismic amplitude data. As in conventional single-trace spectral balancing, each output passband could be boosted to a common output level if desired.

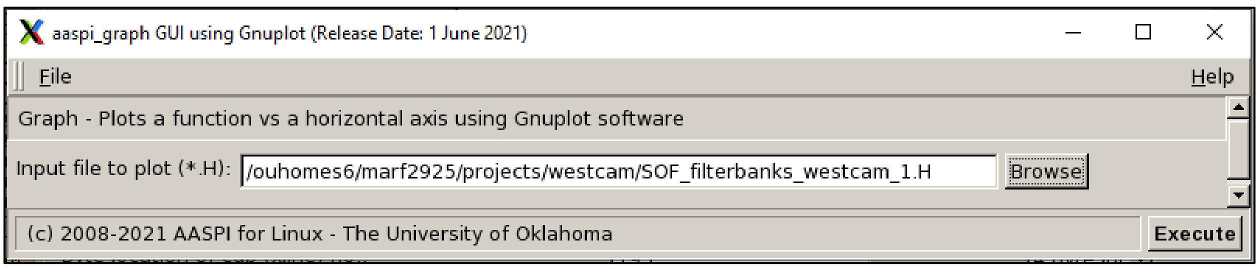
In our 2011 release we implemented some of Helmore's (2009) concepts, which are found under the *Spectral balancing parameters* tab:



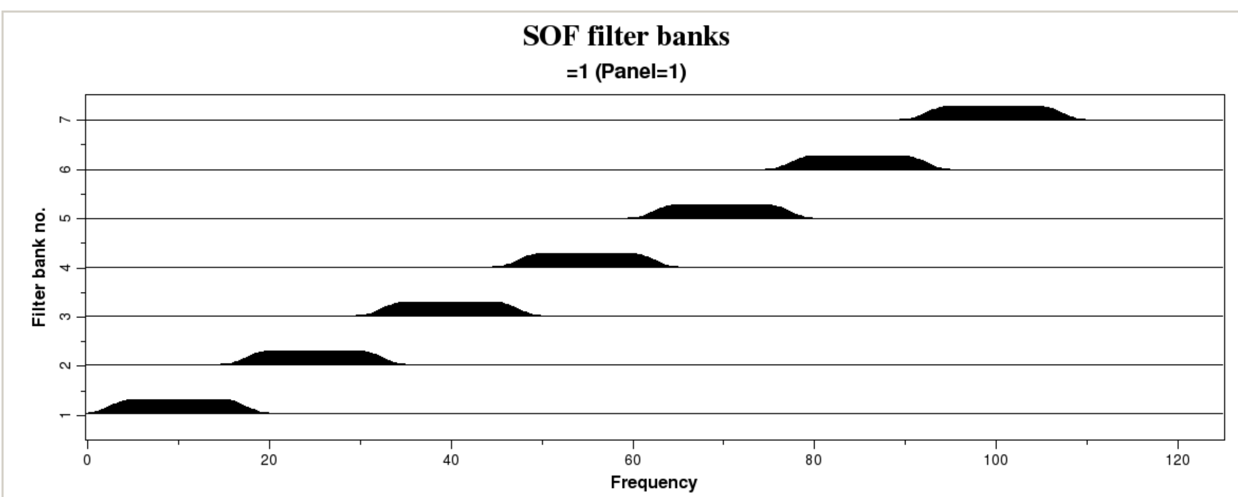
If you wish to apply structure-oriented filtering to one bandpassed version of the data at a time and sum the results to generate the output, (1) click to turn the *Filter spectral bands?* option on. The above values of low and high frequencies, frequency taper for each band range from 5 Hz to Nyquist and width of the untapered portion of each band (2)-(5) above are reasonable for the Westcam survey. The Tukey tapers for each filter bank overlap, reconstructing the original data if added together. If you wish to spectrally balance the output, click (6) to turn the balancing option on. The defaults are to (7) use a half-window of 0.5 s and (8) a percent whitening of 2%. Smaller percentages may further increase high frequencies while smaller windows may better balance lower amplitude events. However, high frequency noise may also be balanced.
If these options are activated, there will be an additional file containing the sof filter banks. To plot that file, invoke the **graph_plot** utility found under the *Display Tools* tab:



Choosing the name filter_bands_boonsville_1.H in the Graph window,

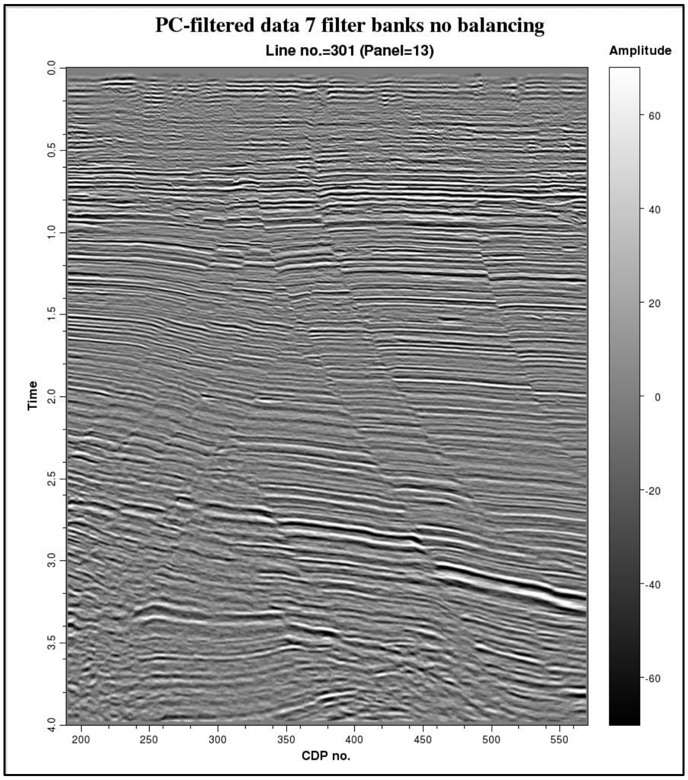


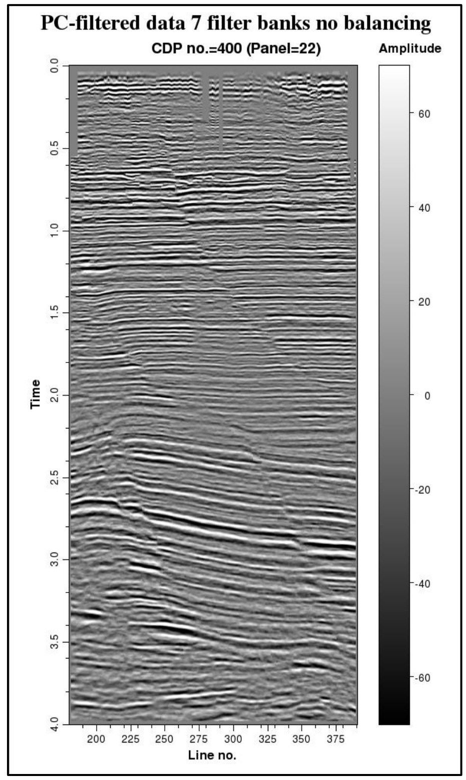
results in the following image:

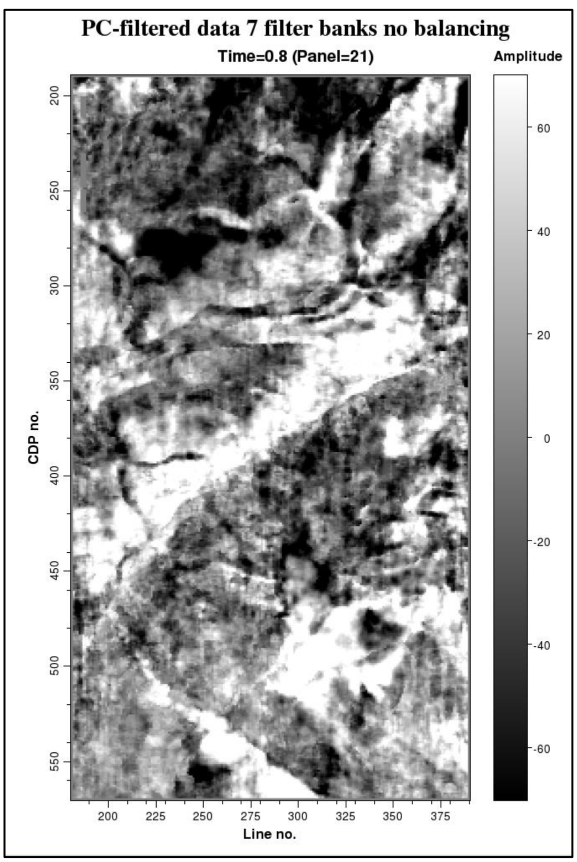


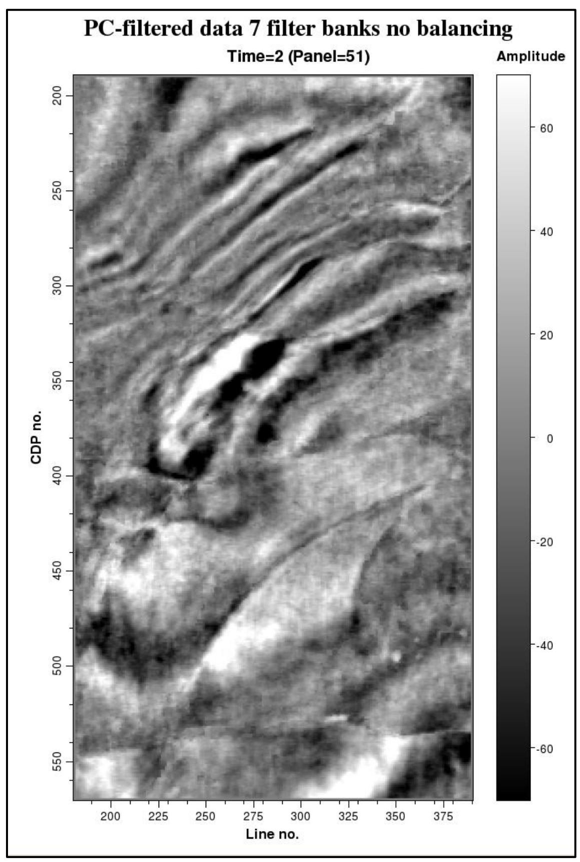
where filter bank #1 is flat between 5 and 15 Hz with a taper that ramps up from 0 to 4 Hz and ramps down from 15 to 20 Hz. Filter bank #2 is shifted over by 15 Hz. The seven filter banks span the requested data range of 0 to 100 Hz.

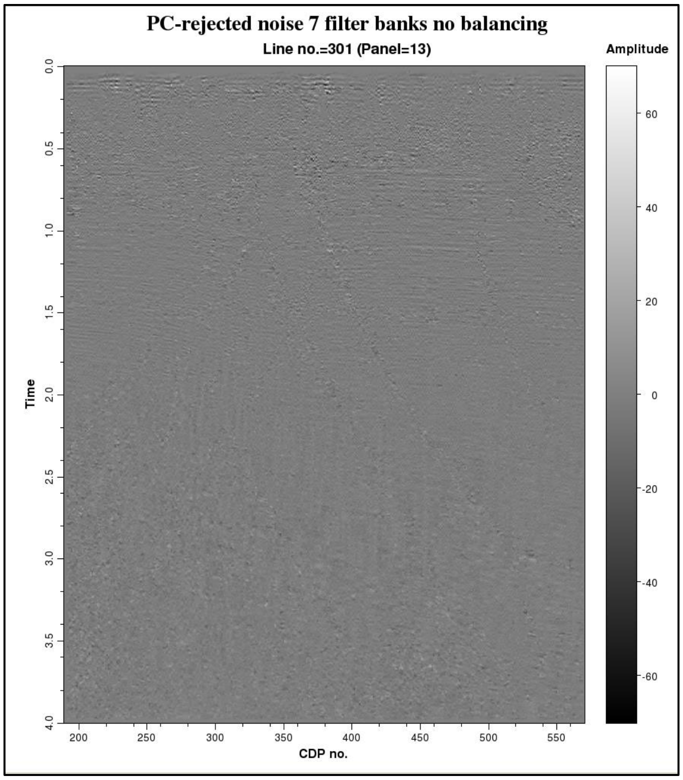
Principal** compon**ent filtering with seven filter banks without spectral balancing

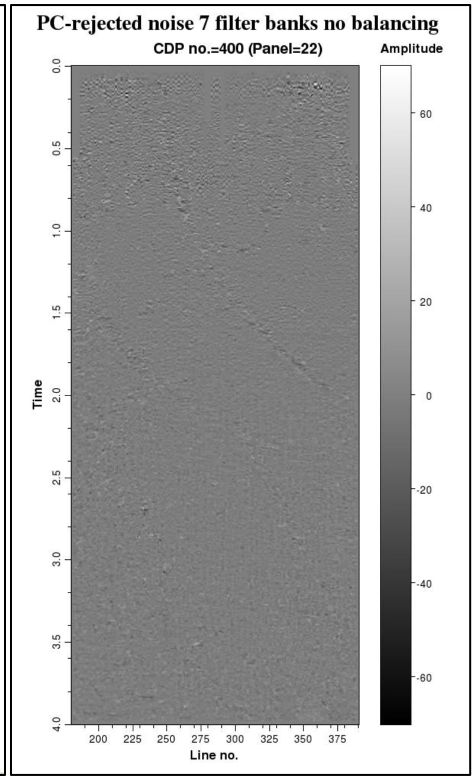


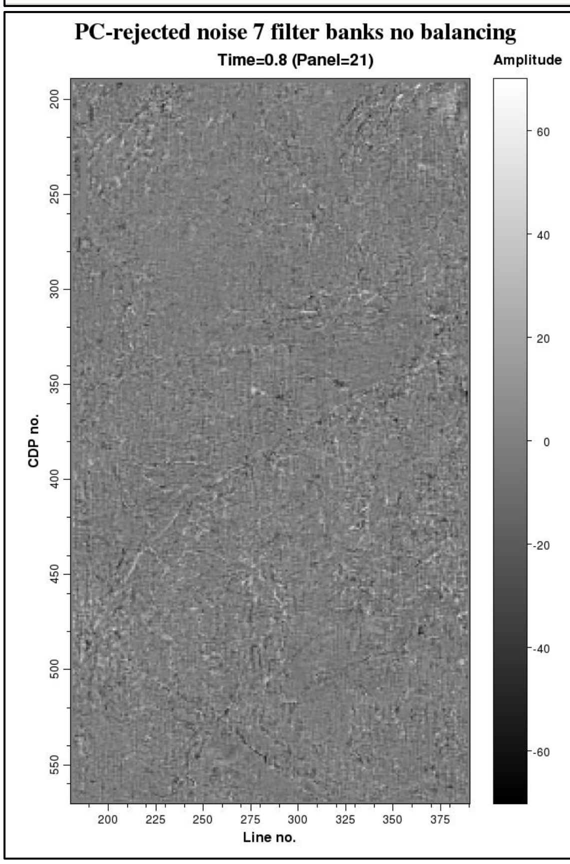


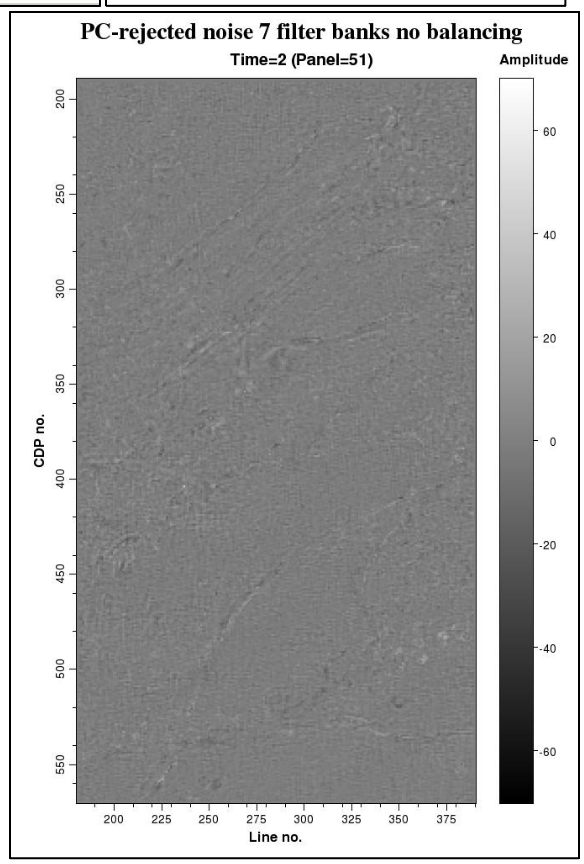


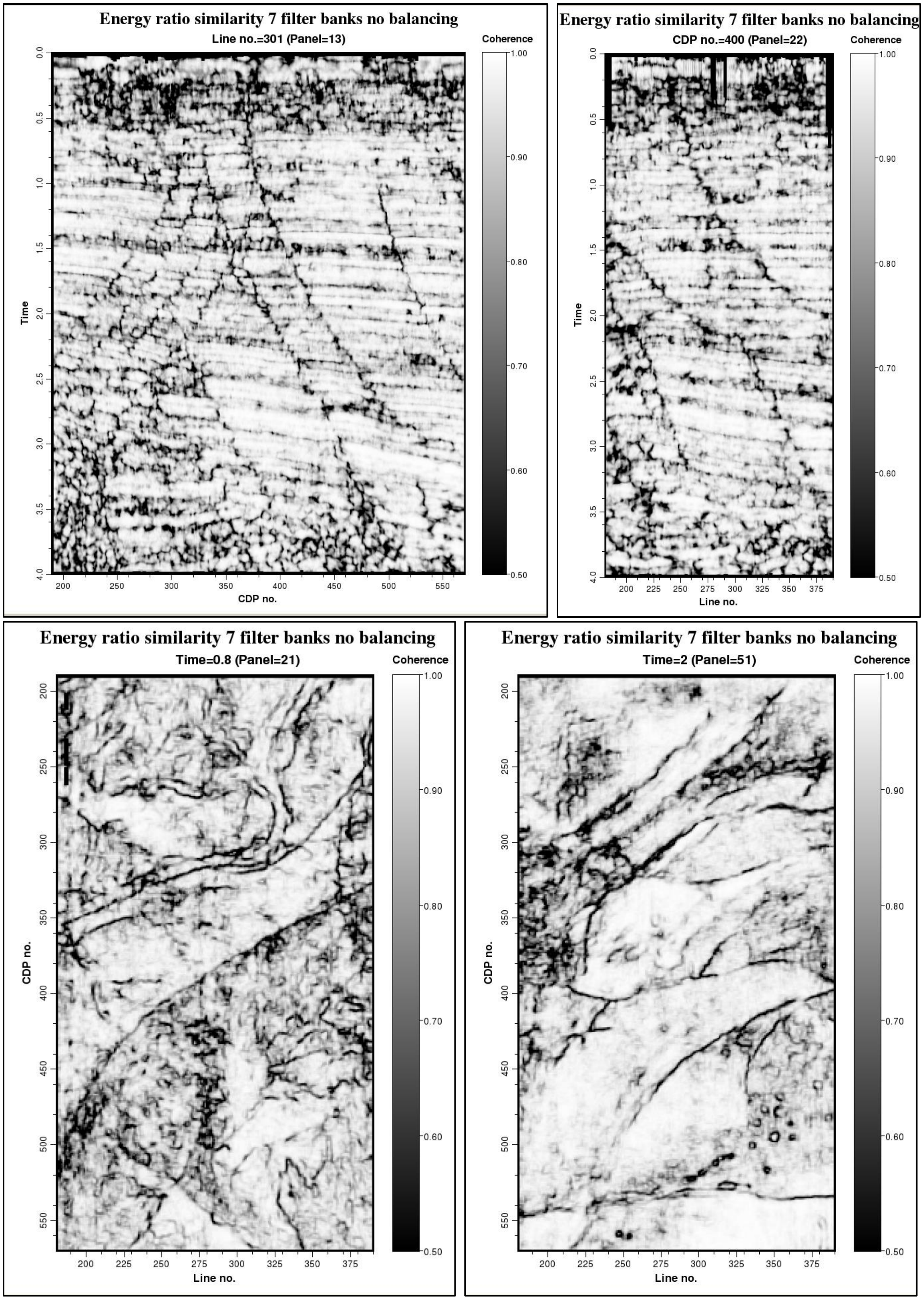



PC-rejected noise 7 filter banks no balancing
Line no.=301 (Panel=13)
Amplitude
Time
CDP no.
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0
200 250 300 350 400 450 500 550
60 40 20 0 -20 -40 -60


PC-rejected noise 7 filter banks no balancing
CDP no.=400 (Panel=22)
Amplitude
Time
Line no.
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0
200 225 250 275 300 325 350 375
60 40 20 0 -20 -40 -60

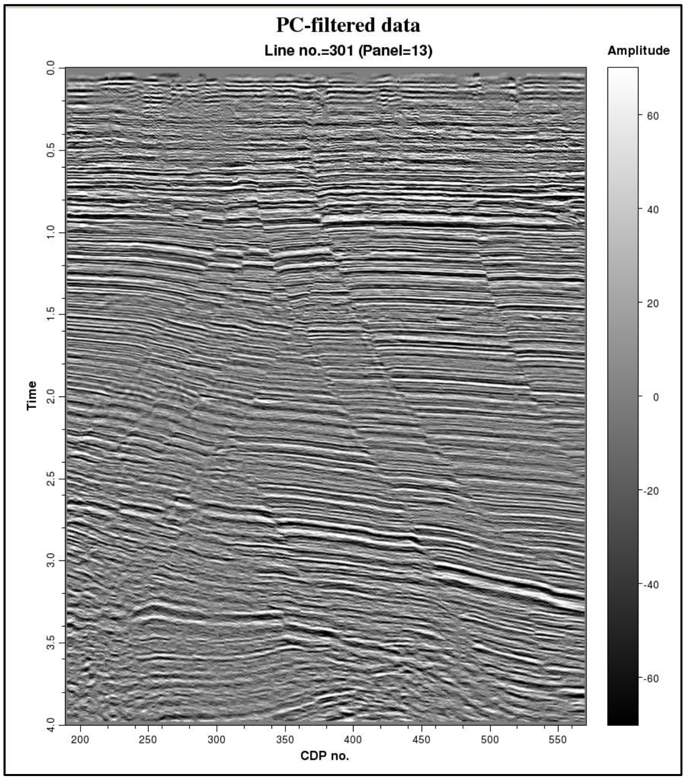

PC-rejected noise 7 filter banks no balancing
Time=0.8 (Panel=21)
Amplitude
CDP no.
Line no.
200 250 300 350 400 450 500 550
200 225 250 275 300 325 350 375
60 40 20 0 -20 -40 -60

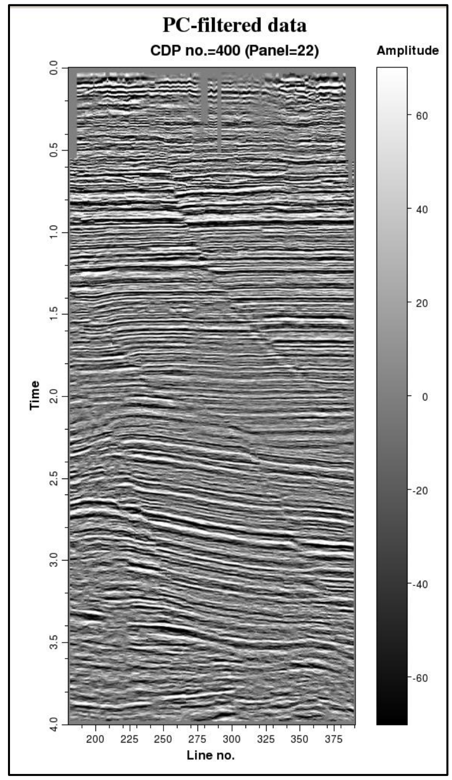

PC-rejected noise 7 filter banks no balancing
Time=2 (Panel=51)
Amplitude
CDP no.
Line no.
200 250 300 350 400 450 500 550
200 225 250 275 300 325 350 375
60 40 20 0 -20 -40 -60

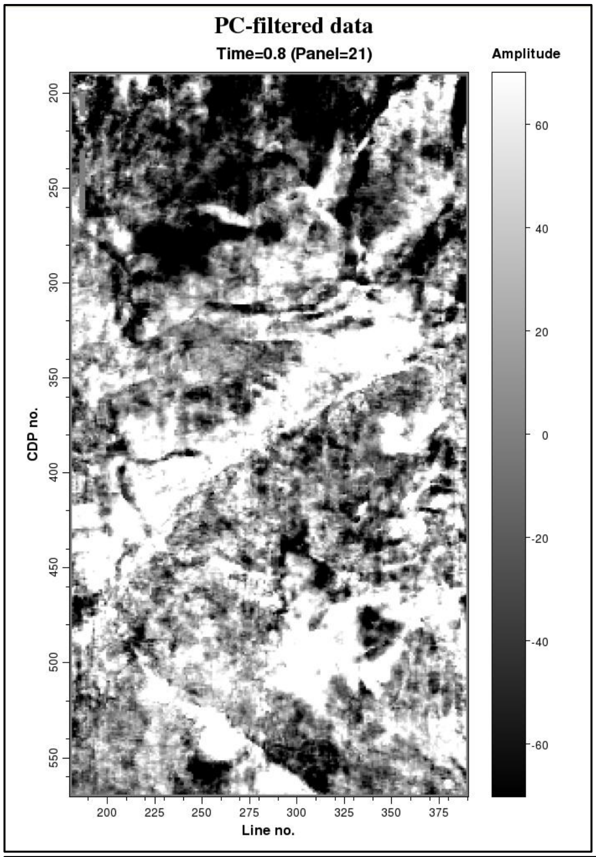

Energy ratio similarity 7 filter banks no balancing
Line no.=301 (Panel=13)
Coherence
Time
CDP no.
Energy ratio similarity 7 filter banks no balancing
CDP no.=400 (Panel=22)
Coherence
Time
Line no.
Energy ratio similarity 7 filter banks no balancing
Time=0.8 (Panel=21)
Coherence
CDP no.
Line no.
Energy ratio similarity 7 filter banks no balancing
Time=2 (Panel=51)
Coherence
CDP no.
Line no.

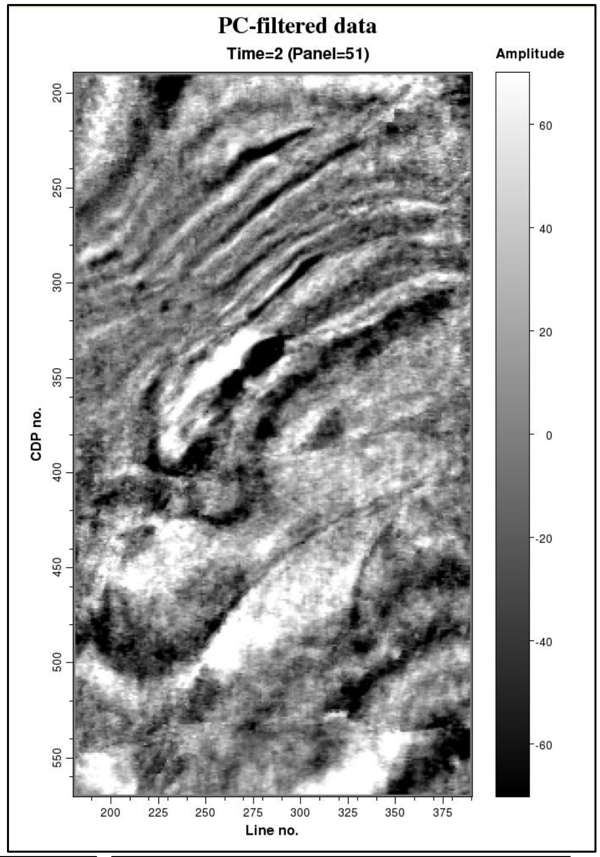
Principal component filtering with seven filter banks with spectral balancing

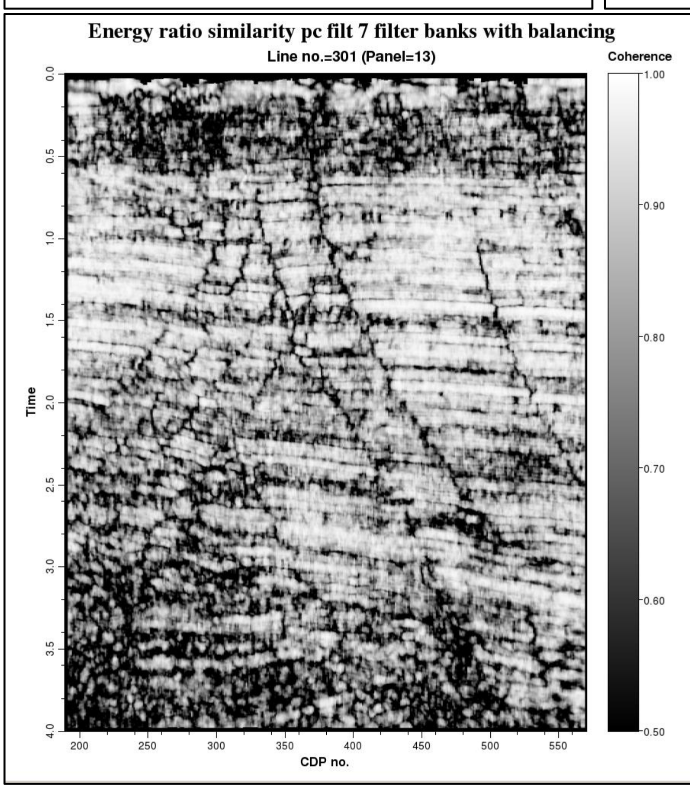
The results with spectral balancing look like this:

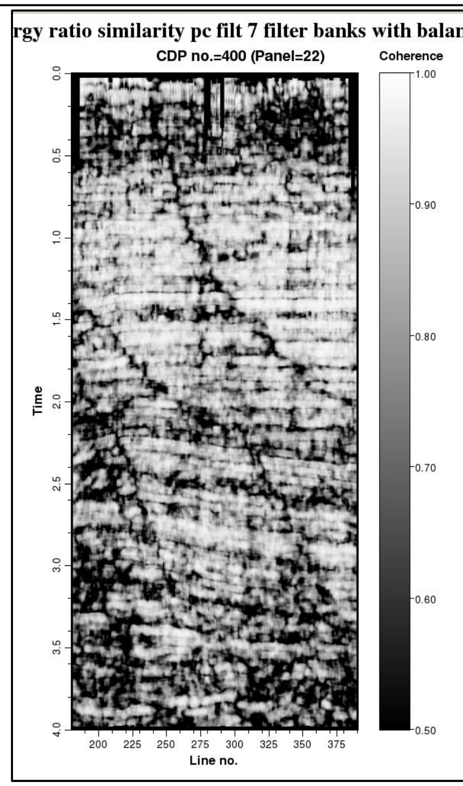


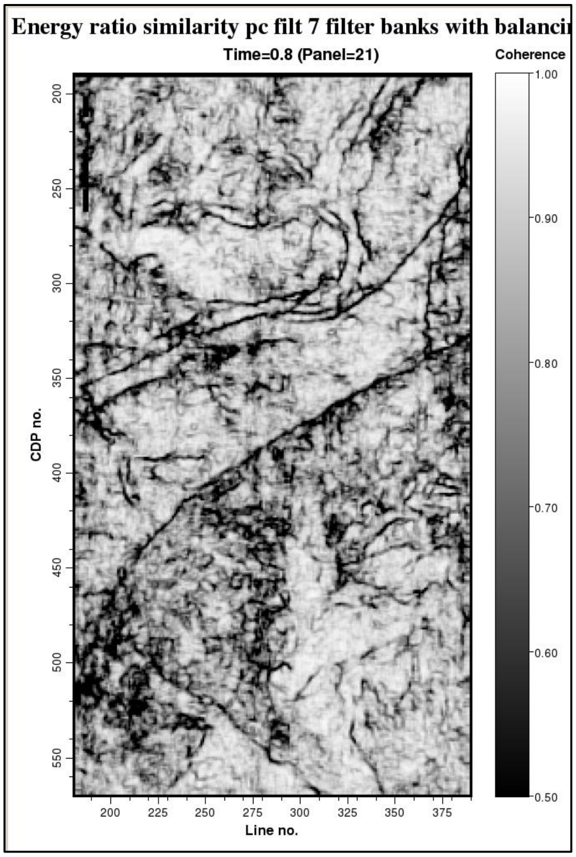


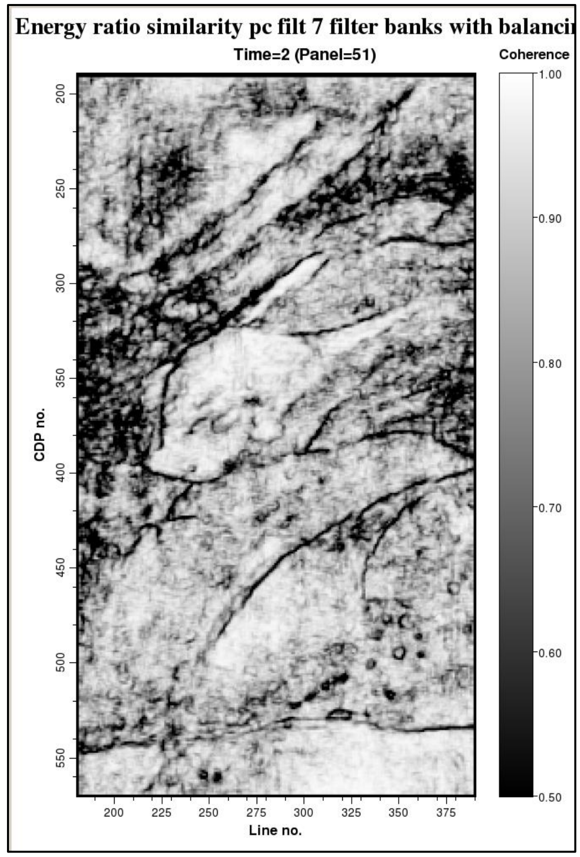

PC-filtered data
Time=0.8 (Panel=21)
Amplitude
CDP no.
Line no.


PC-filtered data
Time=2 (Panel=51)
Amplitude
CDP no.
Line no.


Energy ratio similarity pc filt 7 filter banks with balancing
Line no.=301 (Panel=13)
Coherence
Time
CDP no.


rgy ratio similarity pc filt 7 filter banks with balan
CDP no.=400 (Panel=22)
Coherence
Time
Line no.



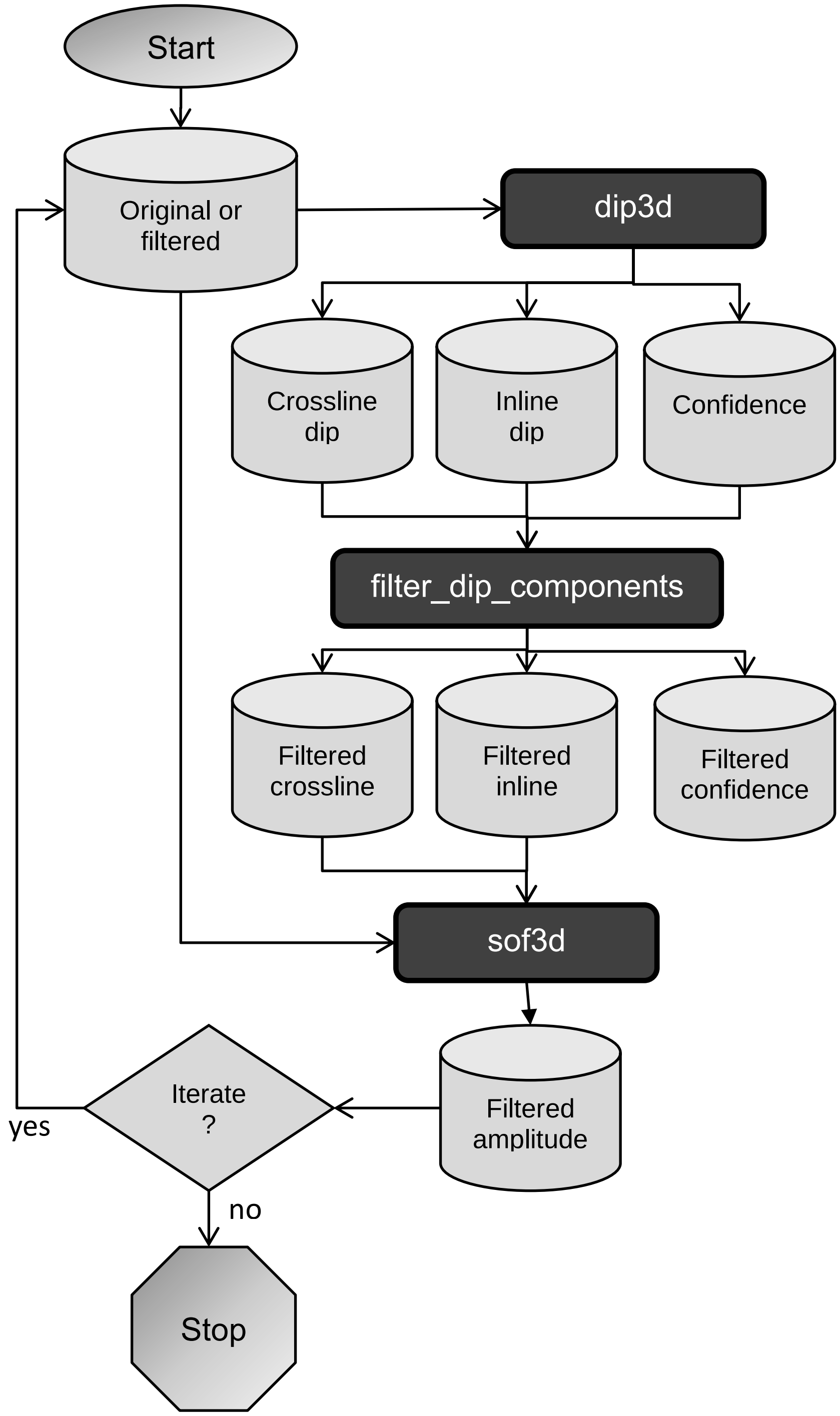


Observations on spectral balancing

For this example, there appears to be significant uplift in spectrally balancing the data as part of the structure-oriented filtering workflow. However, be warned that the spectral balancing in this application is computed trace-by-trace, such that the relative amplitudes may not be preserved. If your goal is to invert for impedance or otherwise quantitatively assess the amplitude, it is safer to perform spectral balancing using programs **spec_cwt** or **spec_cmp**. In these two programs, a single time-variant spectral balancing operator is applied to the entire survey, thereby assuring amplitude preservations. In this workflow, you would then first apply **sof3d** with multiple filter banks and follow it with spectral balancing using **spec_cwt** or **spec_cmp**.

A more rigorous iterative implementation of cascaded structure-oriented filtering

Although we can reuse the output of program **sof3d** as input to a 2nd iteration of **sof3d**, for noisy data it is better to feed this output back into program **dip3d** and repeat the process. In this manner the dips are updated to represent the improved fidelity of the filtered data. Such a workflow looks like the following and is found under ***AASPI_util*** >*Workflows > AASPI Iterative Structure-Oriented Filtering Workflow* tab:



References

Al-Bannagi, M., K. Fang, P. G. Kelamis, and G. S. Douglass, 2005, Acquisition footprint suppression via the truncated SVD technique: Case studies from Saudi Arabia: The Leading Edge, **24**, 832-834.

Davogustto, O., 2011, Removing footprint from legacy seismic data volumes: MS Thesis, The University of Oklahoma. http://mcee.ou.edu/aaspi/AASPI_Theses/2011_AASPI_Theses/Oswaldo_Davogustto_MS_Thesis_2011.pdf

Fehmers, G., F. W. Höecker, 2003, Fast structural interpretation with structure-oriented filtering: Geophysics, **68,** 1286-1293.

Helmore, S., 2009, Dealing with the noise – Improving seismic whitening and seismic inversion workflows using frequency split structurally oriented filters: 78th Annual International Meeting of the SEG, Expanded Abstracts, 3367-3371.

Kirlin, R. L., W. J. Done, 1999, Covariance Analysis for Seismic Signal Processing: Geophysical Developments No.8, Society of Exploration Geophysicists.

Kuwahara, M., K. Hachimura, S. Eiho, and M. Kinoshita, 1976, Digital processing of biomedical images: Plenum Press, 187–203.

Luo, Y., S. al-Dossary, and M. Marhoon, 2002, Edge-preserving smoothing and applications: The Leading Edge, **21**, 136–158.

Marfurt, K. J., 2006, Robust estimates of reflector dip and azimuth: Geophysics, **71**, 29–40.

Sheffield, T. M., and B. A. Payne, 2008, Geovolume visualization and interpretation: What makes a useful visualization seismic attribute?: 78th Annual International Meeting of the SEG, Expanded Abstracts, 849–853.