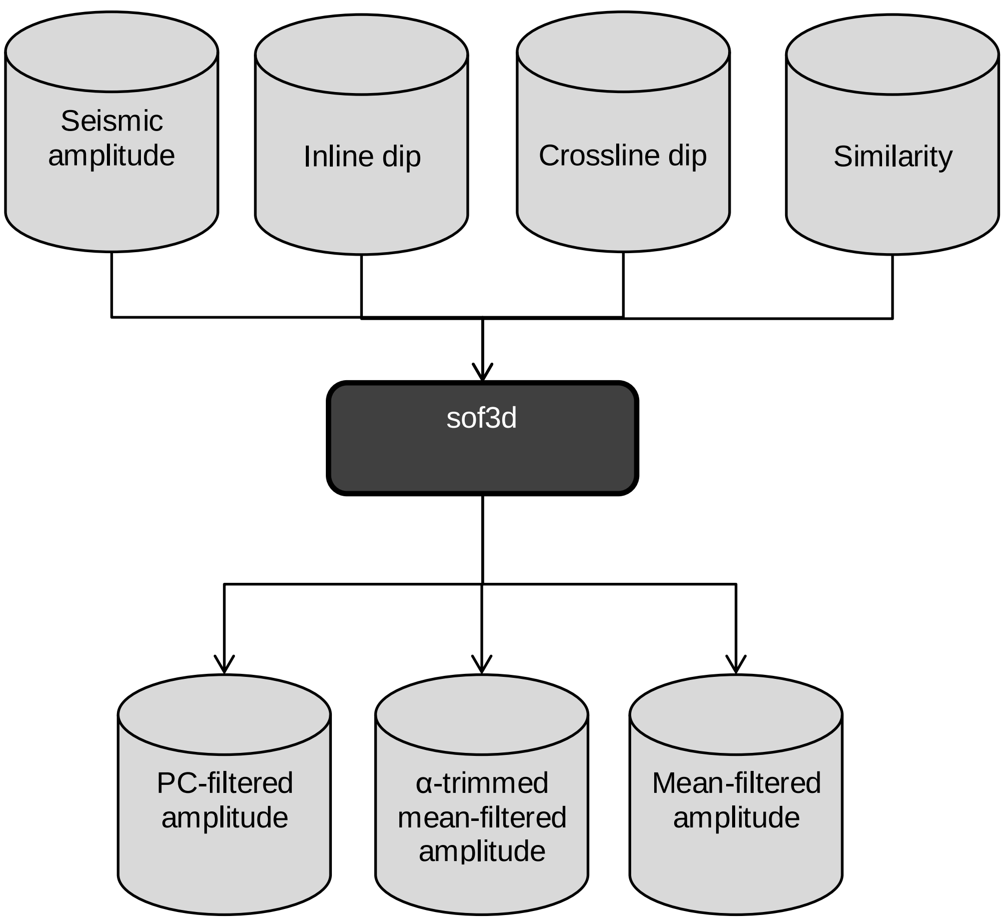
STRUCTURE-ORIENTED FILTERING OF POSTSTACK DATA – Program sof3d

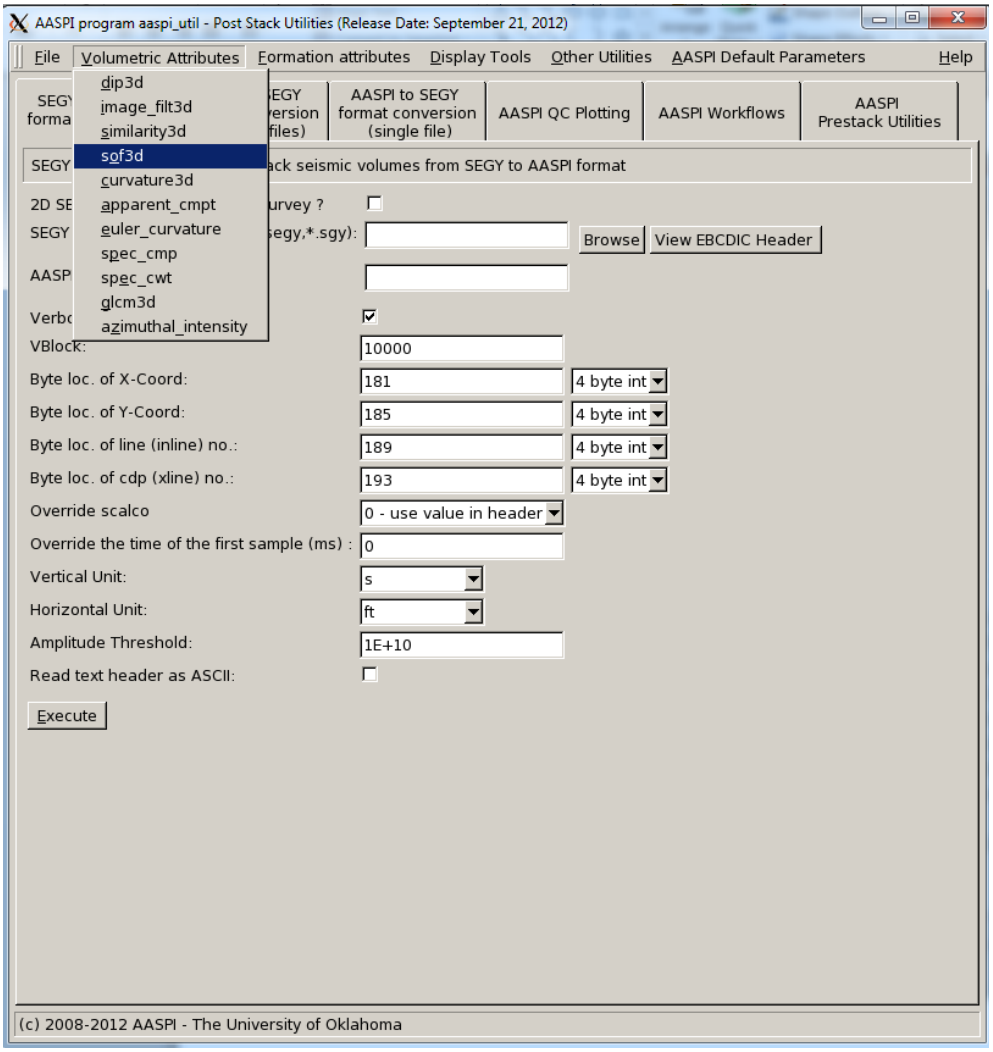
Computation flow chart

Program **sof3d** was significantly rewritten in April, 2011 to provide the application and blending of ‘anisotropic diffusion’ filtering described by Fehmers and Höecker (2003) as well as the Kuwahara filter (Kuwahara et al., 1976) described by Luo et al. (2006). 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 **image_filt3d**. 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-, or mean-filtered versions of the input seismic amplitude data.

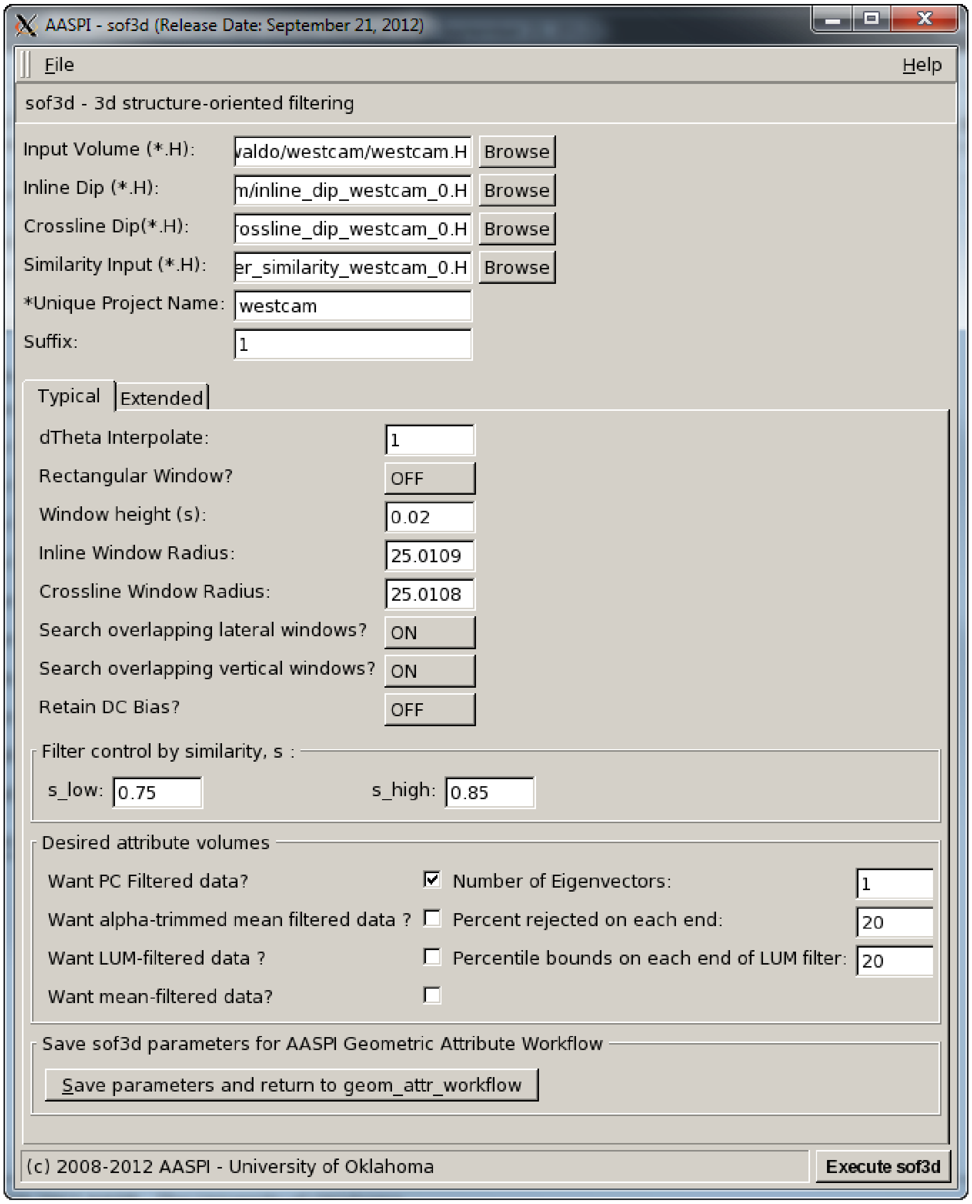


Computing structure-oriented filtered data

Once we have volumetric estimates of dip and azimuth 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 *Volumetric Attributes* tab:



The following GUI appears:



First, select your seismic amplitude volume to be filtered, which in this example is *westcam.H*. Next, select inline and crossline components of dip and the similarity attribute volume generated in program **similarity3d** which will be used as a filtering weight according to the similarity attribute response. As before we define the project name as 'westcam' and type '1' as the suffix, which will indicate that we have subjected the data to one pass of filtering. Each attribute volume option – PC-filtered (also called Kohonen-Loeve or KL filtered), alpha-trimmed mean filtered, LUM-filtered, and mean filtered – will generate a separate volume which can be compared. In this example we have chosen the PC-filtered data volume. As in all AASPI codes, program progress is echoed to the *xterm* from which **aaspi_util** was launched. The end of the print-out looks like this:

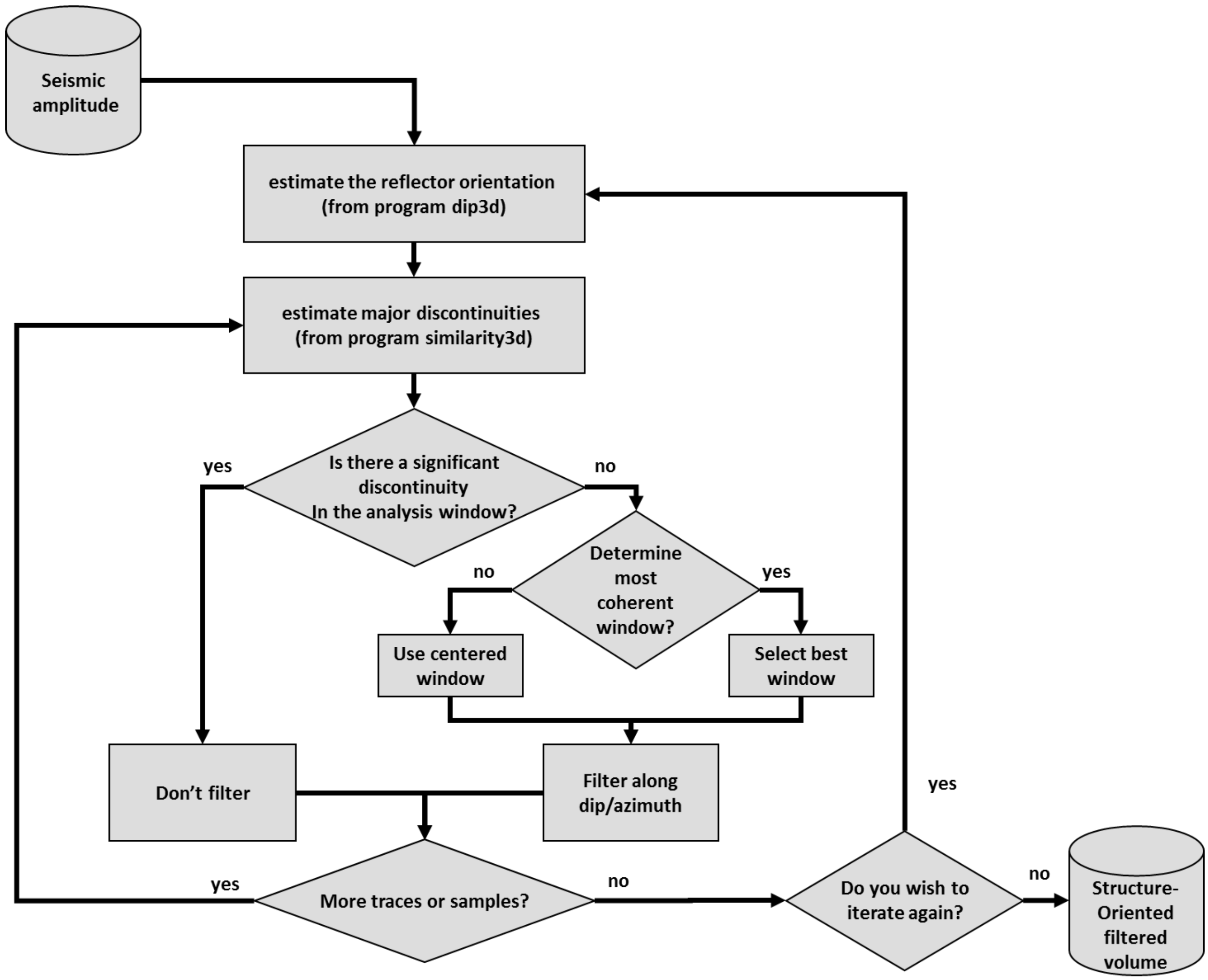
```
         0 : attempt to deallocate filter
         0 : filter      deallocated.
         0 : memory deallocated on master and slaves
         2 :end loop over lines
         2 number of traces processed:        1649
 process                             task      time (hr)  time/trace
  2:                   read and scale data      0.000        0.000
  2:          compute analytic filter bands     0.000        0.000
  2:                     send data via MPI      0.000        0.000
  2:                  receive data via MPI      0.000        0.000
  2:                  send results via MPI      0.000        0.000
  2:               receive results via MPI      0.000        0.000
  2:                   scan discrete dips       0.000        0.000
  2:                      interpolate dip       0.000        0.000
  2:                  calculate cov matrix      0.001        0.003
  2:                calculate eigenvectors      0.000        0.001
  2:                  project data onto v1      0.000        0.000
  2:                 write results to disk      0.000        0.000
  2:                            total time      0.008        0.018
         2 : attempt to deallocate d_coarse etc.
         2 : deallocated d_coarse etc.
         2 : memory deallocated on slaves
         2 : deallocate d_mean_filt etc.
         2 : d_mean_filt deallocated.
         2 : inline_dip  deallocated.
         2 : line_index  deallocated.
         2 : t_lag_interp deallocated.
         2 : attempt to deallocate filter
         2 : filter      deallocated.
         2 : memory deallocated on master and slaves
         1 :normal completion. routine sof3d
         0 :normal completion. routine sof3d
         7 :normal completion. routine sof3d
         2 :normal completion. routine sof3d
         8 :normal completion. routine sof3d
         6 :normal completion. routine sof3d
         3 :normal completion. routine sof3d
         4 :normal completion. routine sof3d
         5 :normal completion. routine sof3d
[kmarfurt@opal boonsville]$
```

If you type '*ls –ltr*' in the above *xterm*, you find the most recent files to be

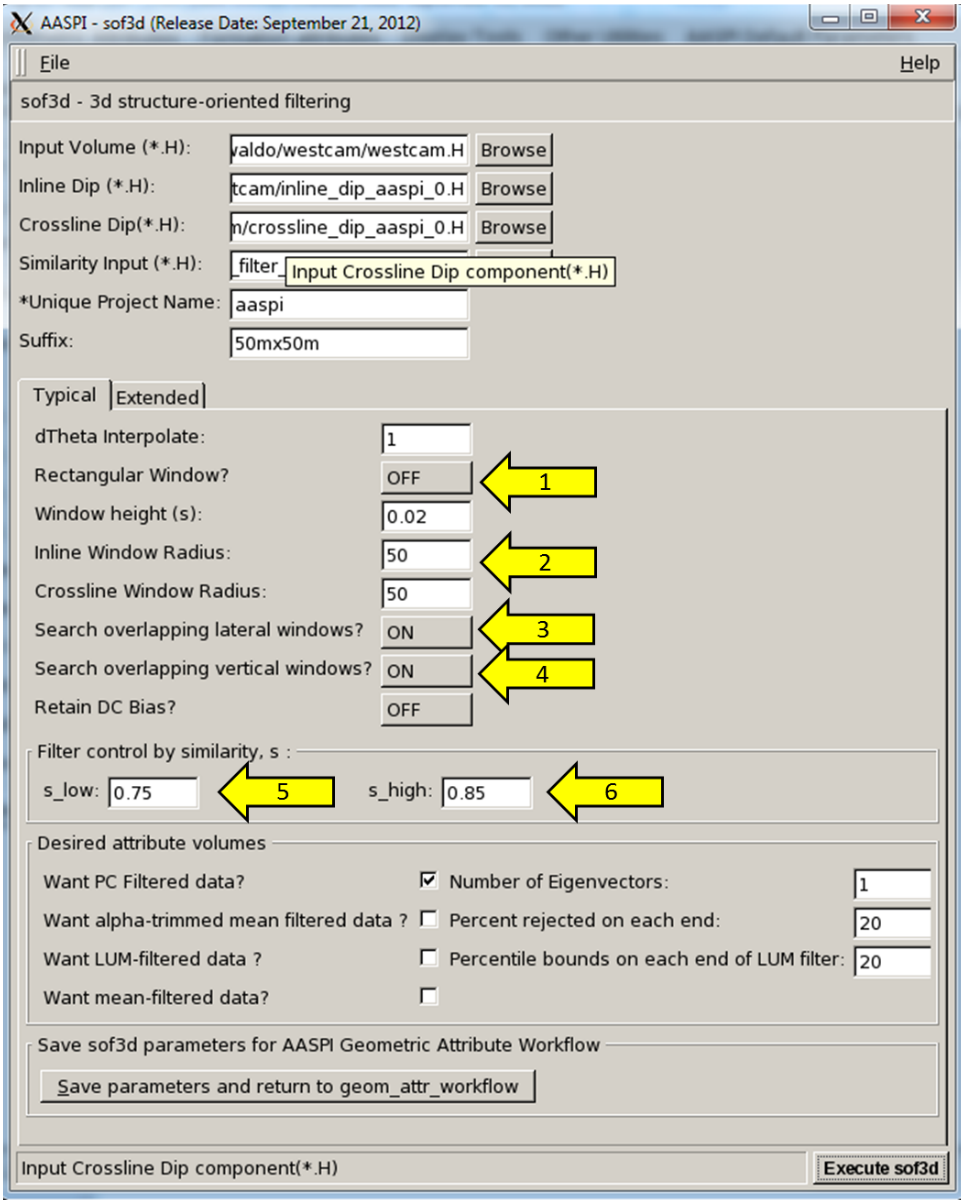
```
-rw-r--r-- 1 oswaldo aaspi      36 Oct 16 18:08 live_processor_list
-rw-r--r-- 1 oswaldo aaspi    2204 Oct 16 18:09 filter_coeff_westcam_1.H@@
-rw-r--r-- 1 oswaldo aaspi    2380 Oct 16 18:09 filter_coeff_westcam_1.H
-rw-r--r-- 1 oswaldo aaspi    2205 Oct 16 18:09 d_pc_filt_westcam_1.H@@
-rw-r--r-- 1 oswaldo aaspi    2391 Oct 16 18:09 d_pc_filt_westcam_1.H
-rw-r--r-- 1 oswaldo aaspi   47439 Oct 16 18:10 sof3d_westcam_1.out
[oswaldo@jade westcam]$
```

Which shows that the output of program **sof3d** was called *d_pc_filt_westcam_1.H* . You may wish to run TWO iterations of structure-oriented filtering. To do so, return to your **sof3d** GUI, and use the browser to find the file *d_pc_filt_westcam_1.H* . Let's use a suffix of 'pc_2' to indicate the results are from 2 passes of structure-oriented filtering. Use the same dip calculation and Kuwahara window (though you can rerun program **dip3d** on the file *d_pc_filt_westcam_1.H)*.

The main workflow for the structure oriented filter is described in Davogustto and Marfurt (2011):



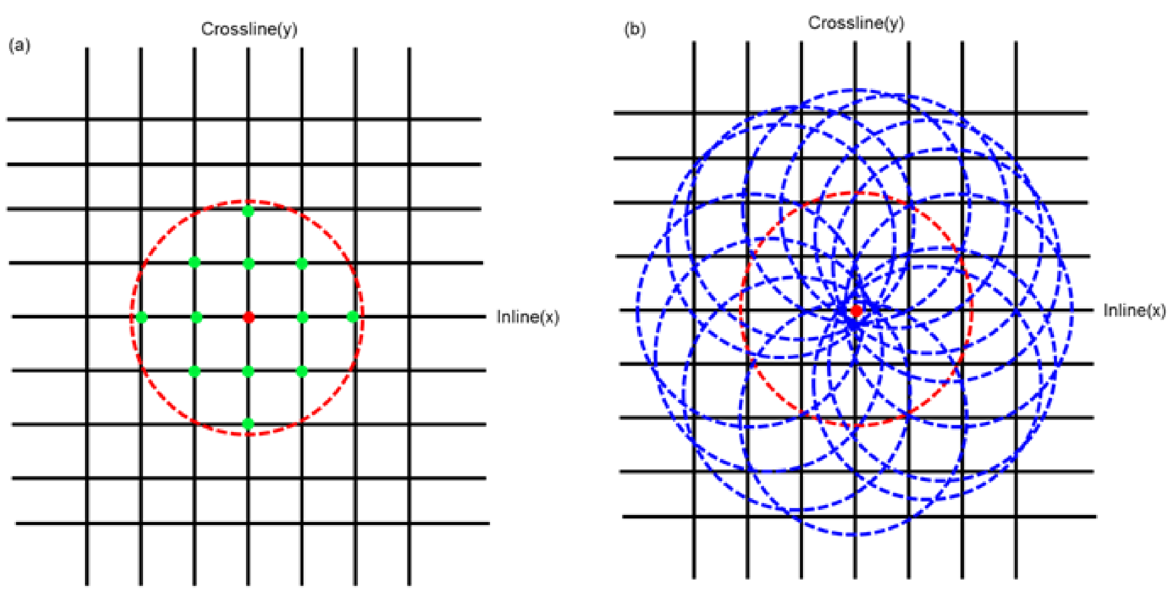
Let's explain the advanced parameters in more detail:



The program default is to use a circular search window that for equal inline and crossline spacing will contain 5 traces. These parameters can be changed by (arrow 1) selecting *rectangular* window analysis (where 3x3=9 traces fall within the smallest window) and/or by (arrow 2) increasing the inline and crossline radii to define and elliptical or rectangular window of the desired size. In this case, we've increased the size to be a 50 m radius window containing 13 traces. Both the computation time and the strength of the filter increase with increasing window size. For good quality data, a more effective workflow is to iteratively smooth using smaller windows rather than 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. 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.

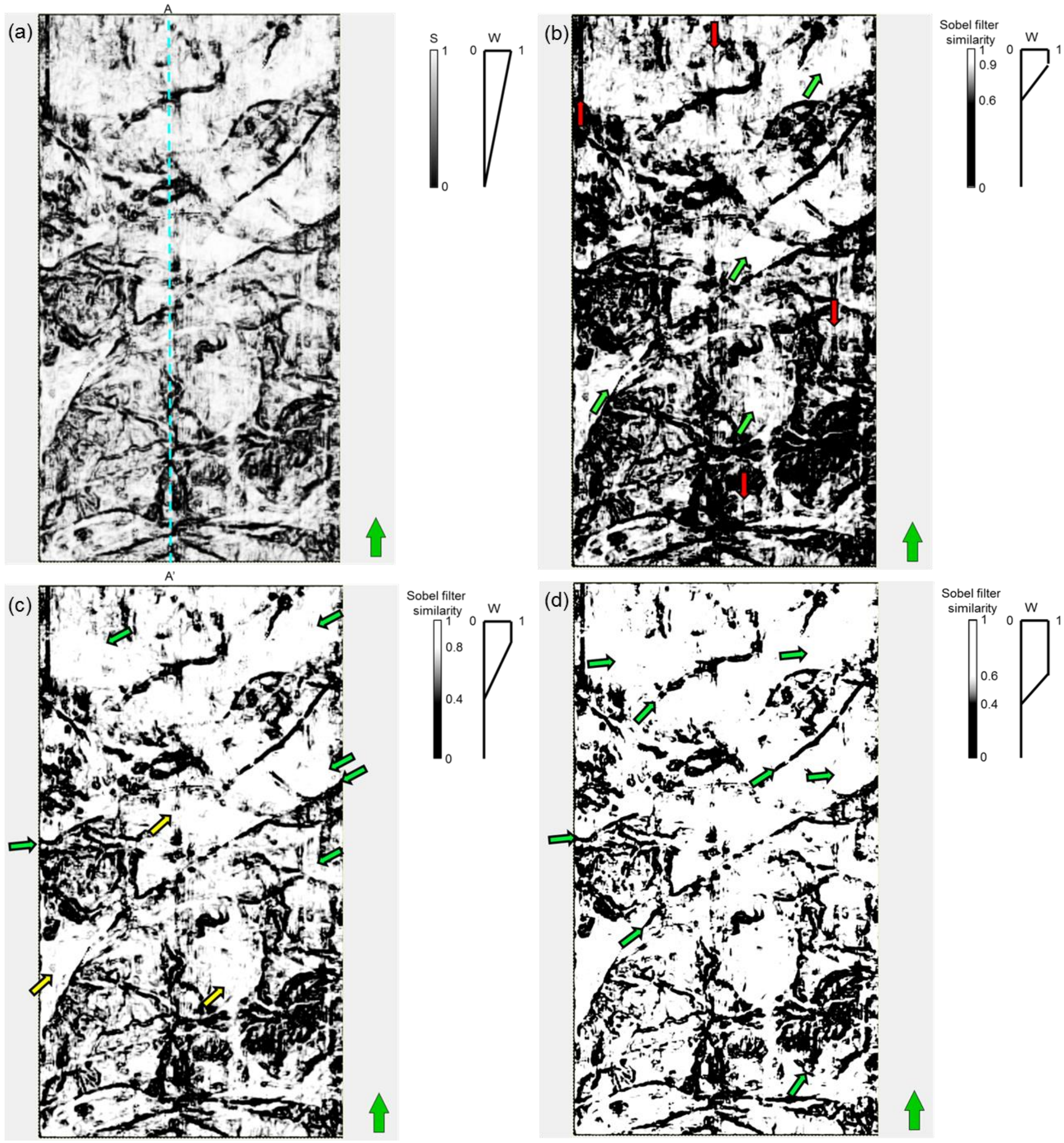
Marfurt (2006) built on Luo et al.'s (2002) Kuwahara algorithm to implement a robust volumetric dip and azimuth calculation that avoided smearing of faults, fractures and other discontinuities using an overlapping window method. This technique along with the seismic data input can be used to implement volumetric filters based on mean, median, α-trimmed mean or principal component algorithms (see box below in this chapter for an overview of PC filtering). Rather than using a centered analysis window, the algorithm uses the most coherent window containing each analysis point, hence enhancing the lateral resolution near discontinuities and reducing both random and coherent noise (Marfurt, 2006).

In the figure below (a) 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 (b) form the center of their own 13-trace analysis windows. Each of these overlapping analysis windows also contains the trace represented by the red dot. The original Kuwahara et al. (1979) algorithm estimated the mean and standard deviation of the data in each window. The window having the smallest standard deviation was declared to best represent the signal; the mean of this window was then assigned to be the filtered data at the output (typically uncentered) analysis point. Marfurt (2006) applied this same approach to 3D seismic data using a simple extension. Rather than use the standard deviation, he computed the dip-steered coherence in 3D overlapping windows. After selecting the window with the highest coherence, he then computed either the mean, alpha-trimmed mean, or principal-component filtered estimate of the signal and assigned the result to the filtered volume at the analysis point. Use of such (arrow 3) laterally shifted windows helps avoid smoothing across faults. Use of (arrow 4) vertically shifted analysis windows helps avoid smearing across angular unconformities.



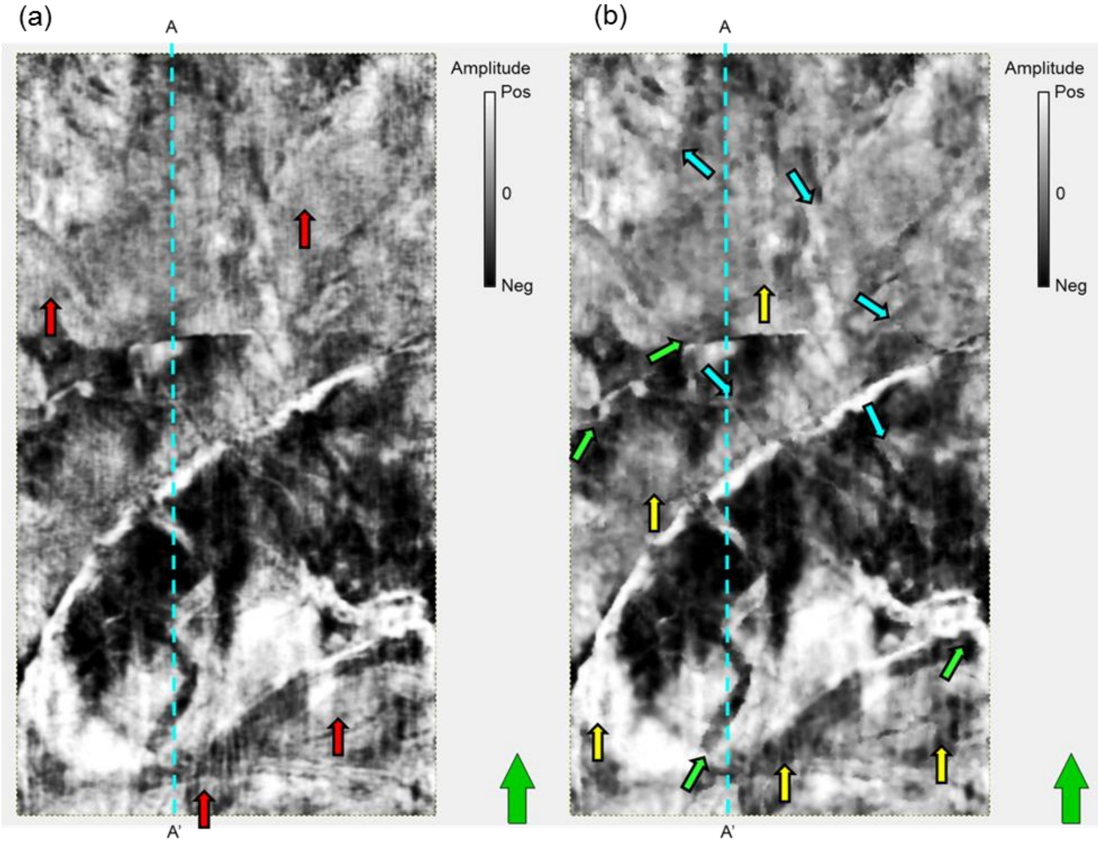
For simplicity, the flow chart shown earlier indicates a simple *Don't filter* vs. *Filter along dip/azimuth* branch. In the current implementation of program **sof3d** we've implemented components of the Fehmer's and Hoecker (2003) workflow. If the value of the similarity attribute at the analysis point falls *below* the threshold indicated by (arrow 5), $s<s_{low}$, no filtering takes place and the filtered data are assigned weights of $w=0.0$. If the value of the similarity attribute at the analysis point falls *above* the threshold indicated by (arrow 6), $s>s_{high}$, the filtered data are assigned weights of $w=1.0$ such that the filtered data replaces the original data on output. If the value of the similarity attribute at the analysis point falls *between* the two values indicated by (arrows 5 and 6), the weights of the filtered data are $w=(s-s_{low})/(s_{high}-s_{low})$, and a linearly weighted average of the filtered and unfiltered data $d_{out}=w*d_{filt}+(1-w)*d_{orig}$, takes place.

The figure below shows a suite of images illustrating the interactive workflow used to define smoothing weights, w, in structure-oriented filtering for a time slice at t=0.76 s through the Sobel filter similarity data volume, s, computed from the westcam data set. Use the color bar on your work station display to choose appropriate values of s_{low} and s_{high}. Specifically, set two color 'ramp' points to be s_{low} and s_{high}. Then set the color to be white *if* $s>s_{high}$, black if $s<s_{low}$ and shades of gray if $s_{low}<s<s_{high}$. The resulting image will be the weights applied to the filtered data on output such that all black discontinuities will be preserved and all white areas will be filtered.

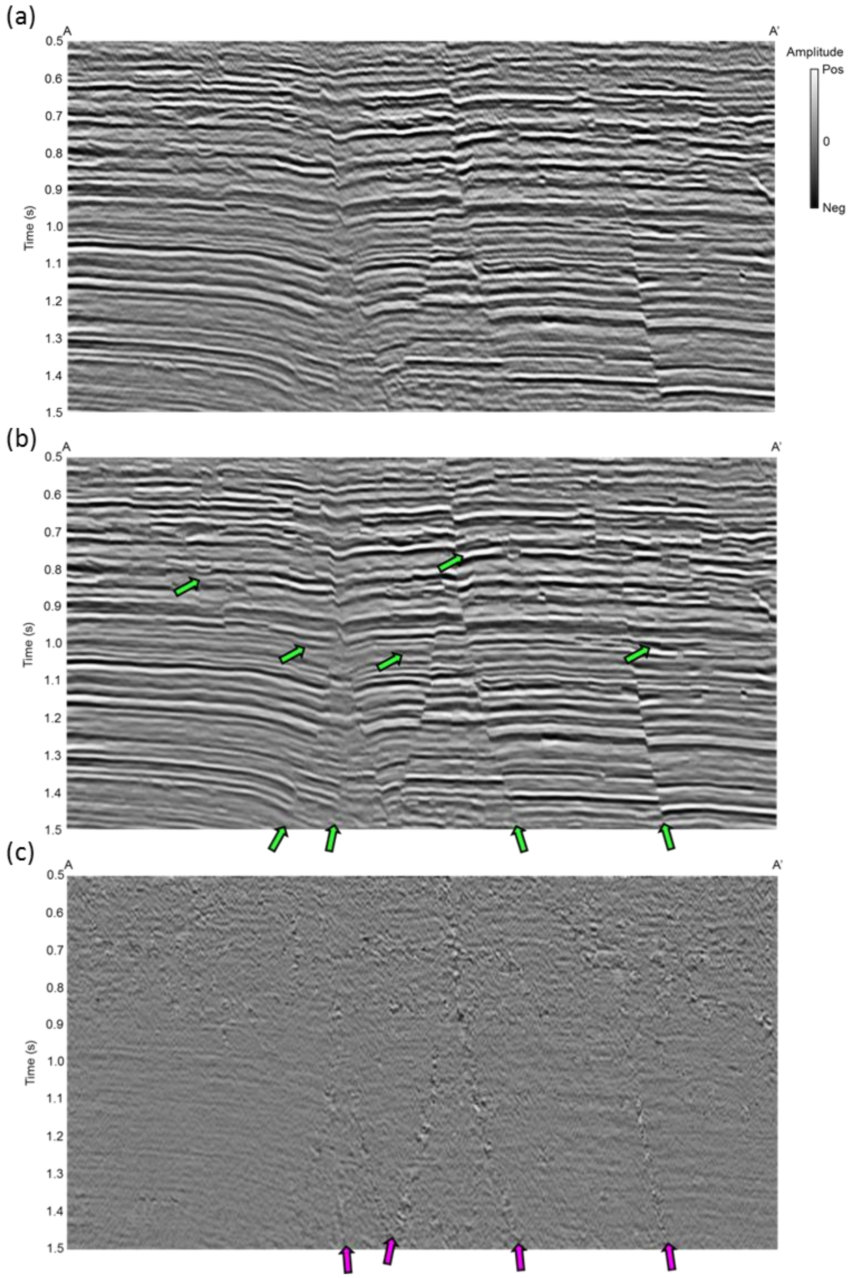


In the image above (a) shows the color bar applied to the similarity values, *s*, and weights, *w*=*s*. This is our normal display of similarity, *s*. By modifying the threshold values for *s* we increase or decrease the smoothing weights thereby changing the aggressiveness of the filter. In (b) we adjust the colorbar to enhance the footprint noise (red arrows). Our filter would thus unfortunately *preserve* footprint. Green arrows indicate white areas where the filter will be more aggressive and remove incoherent noise. Yellow arrows in (c) indicate stratigraphic features that might be smoothed due to their relatively higher similarity weights, *w*. In general, the Kuwahara filtering option can avoid such smoothing across the discontinuity. Green arrows indicate features that will be preserved or areas where footprint will be removed. The values of *w* in (d) are perhaps the optimal setting for this data volume. Green arrows indicate areas where the values of *w* in (d) shows an improvement over the values in (c) both in preserving fault discontinuities and suppressing footprint discontinuities.

Let's see what structure-oriented filtering has done to our seismic amplitude data.



The image above shows a time slice at *t*=0.76 s through (a) the original (unfiltered) seismic amplitude volume and (b) the output from program **sof3d** using one pass of principal component filtering for the Westcam data set. Red arrows in (a) indicate footprint contamination. The amplitude of the footprint in (b) was diminished by the filter but there are still remnants visible after the first filtering iteration (yellow arrows). Green arrows indicate areas where edges of geologic features are sharper.

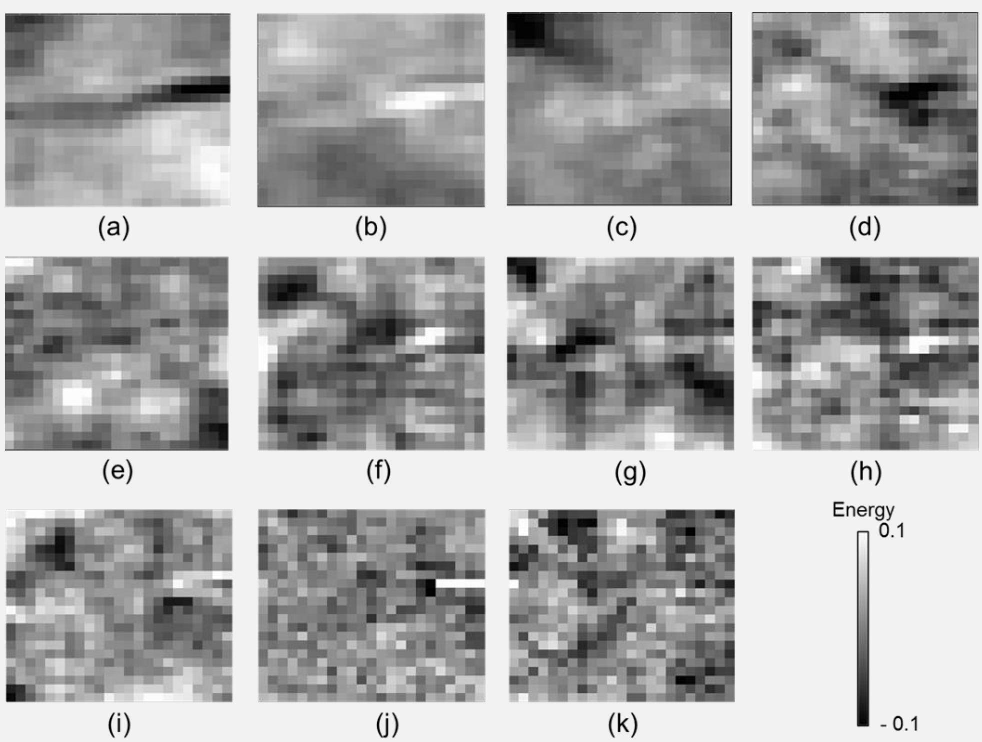


The figure above shows a vertical slice along AA' through the (a) original data, (b) filtered data and (c) noise, respectively. The data were filtered using the 'optimal' values of s_{low} and s_{high} discussed above, and a suite of 13 overlapping (Kuwahara) windows each of which contains 13 traces. Program **sof3d** does a good job in removing the laterally variable footprint modulation through the entire section (green arrows), as well as fault plane reflections (magenta arrows) that conflict with the dominant dipping reflectors. An improvement in the reflector coherence due to the overlapping Kuwahara window technique is observed.

An overview of principal components (part 1)

Principal components provide a means of identifying an amplitude pattern 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 see the pattern 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.

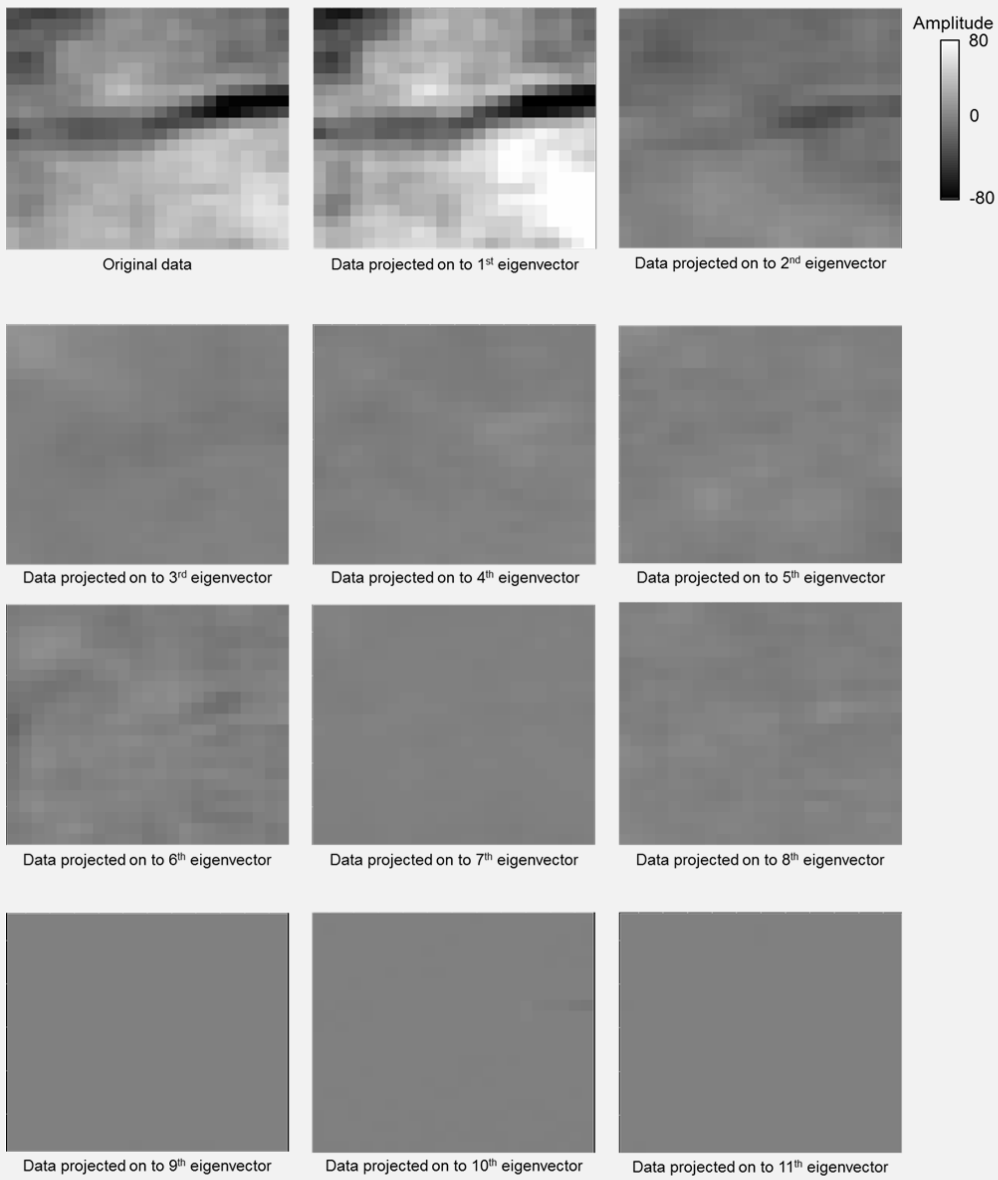
Let's examine 11 "sample snapshots" or "sample vectors" of our seismic patch. Clearly, we could reacquire our survey 11 times with different acquisition parameters to randomize our noise. More simply, we can assume that our seismic wavelet within the ± 5 sample window is sampling the same reflectivity. Changes in the wavelet (peak, trough, zero-crossings) are not unlike the satellite images at different times of the day. The figure below shows these 11 images. If K=,5 cross-correlation from K=-5 to K=$+5$ of the n^{th} trace with the m^{th} trace forms the mn^{th} element of the 441 by 441 covariance matrix **C**.



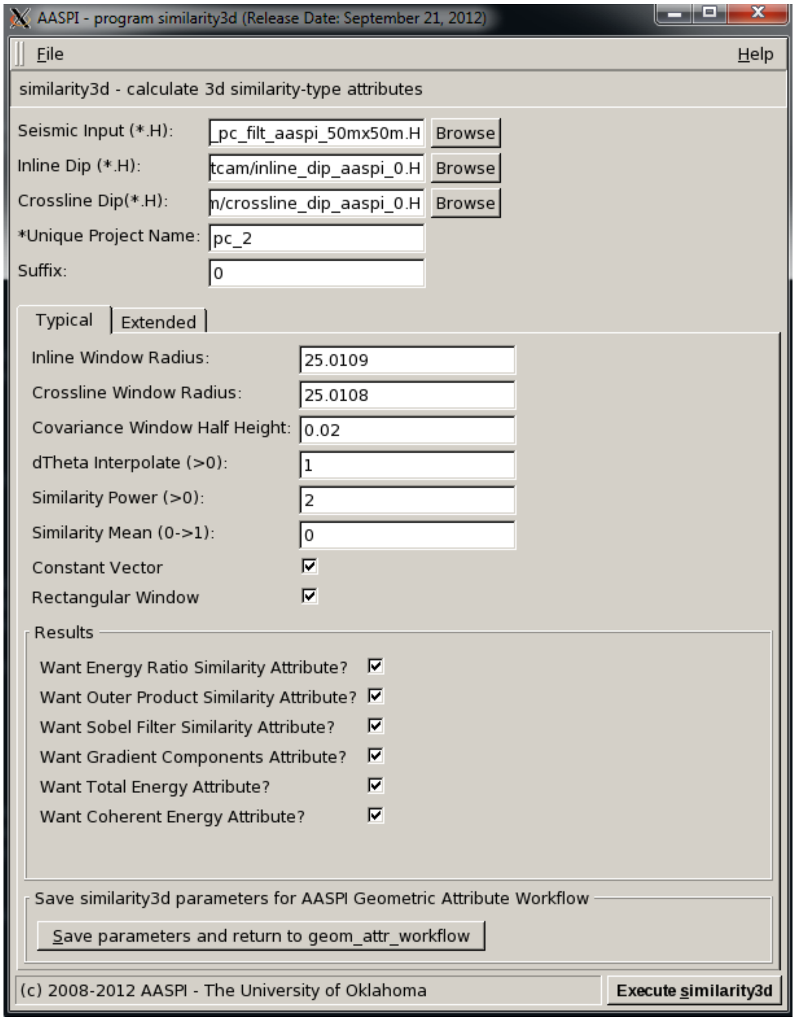
By definition, the first principal component, also called the first eigenvector (a), represents the variability of the data, and for moderate amplitude noise, best represents our consistent reflectivity pattern (Kirlin and Done, 1998) as shown in the figure above.

An overview of principal components (part 2)

For better or worse, principal component analysis has entered the seismic processing world from many directions, rendering the additional names of eigenstructure, eigenvalue-eigenvector, singular value decomposition (SVD), and Karhunen–Loève transform analysis, causing unnecessary confusion. The eigenvectors $\mathbf{v_m}$ of the covariance matrix $\mathbf{C}$ are by construction unit length and orthogonal, such that they can form a basis function as we did with the k_x-k_y transform. 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 $v_m(x,y)$ (where m varies between 1 and 11) are shown in the figure below. Note that most of the amplitude is represented by the first two eigenvectors.

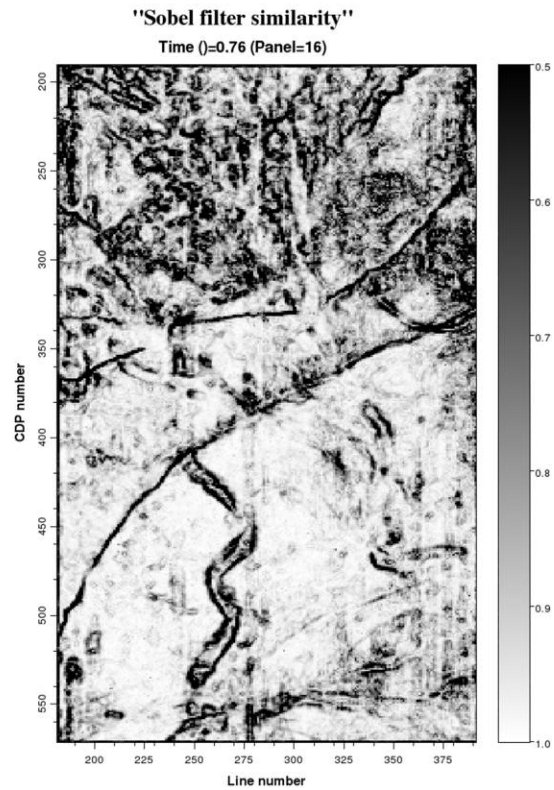


To evaluate the impact of structure-oriented filtering on coherence, pull up the **similarity3d** GUI and rerun it using the filtered outputs:



Where the input is the filtered data and we now use a suffix of 'pc_2' to indicate that the attributes have been computed from data that have undergone two passes of filtering. Click *Execute* and then display the results using the *SEP Viewer* tab of the **aaspi_util** GUI:

Before sof3d_workflow

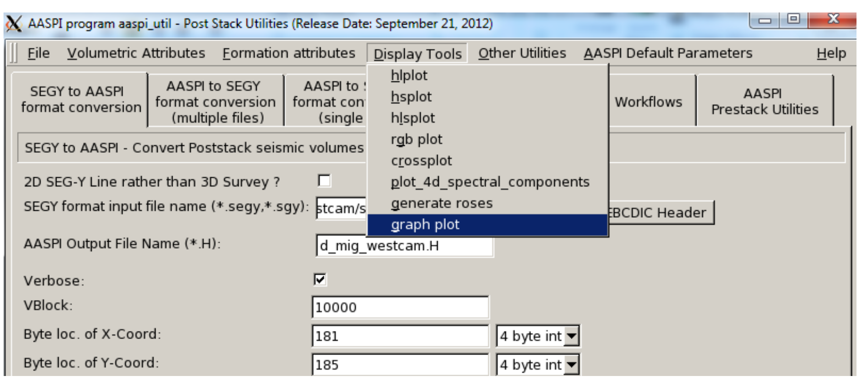


After sof3d workflow

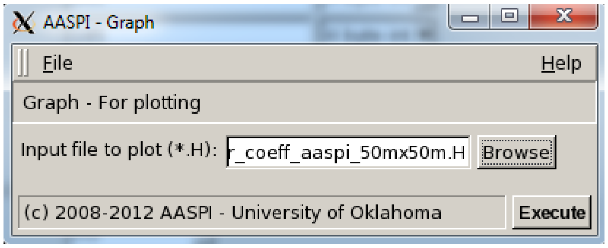
Note how the background incoherent events (either seismic or geologic 'noise') have been suppressed while the edges in this images are somewhat sharper on the Sobel filter similarity computed from the PC-filtered data on the filtered data shown on the right than from that computed from the unfiltered data shown on the left.

Structure-oriented filtering of band-pass filtered windows

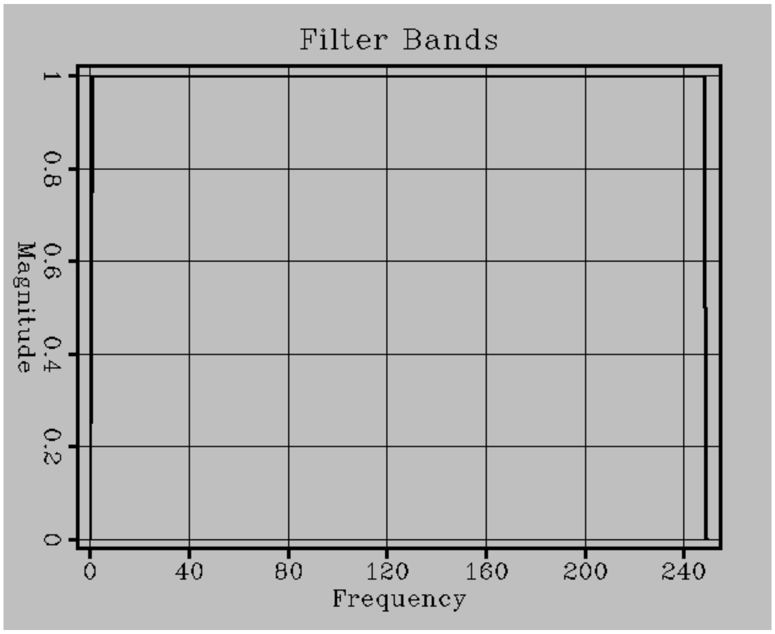
You may have noted an additional file in the output called filter_bands_westcam_1.H . We can plot the contents of this file by clicking the **Graph plot** utility.



Choosing the name filter_bands_boonsville_2.H in the Graph window

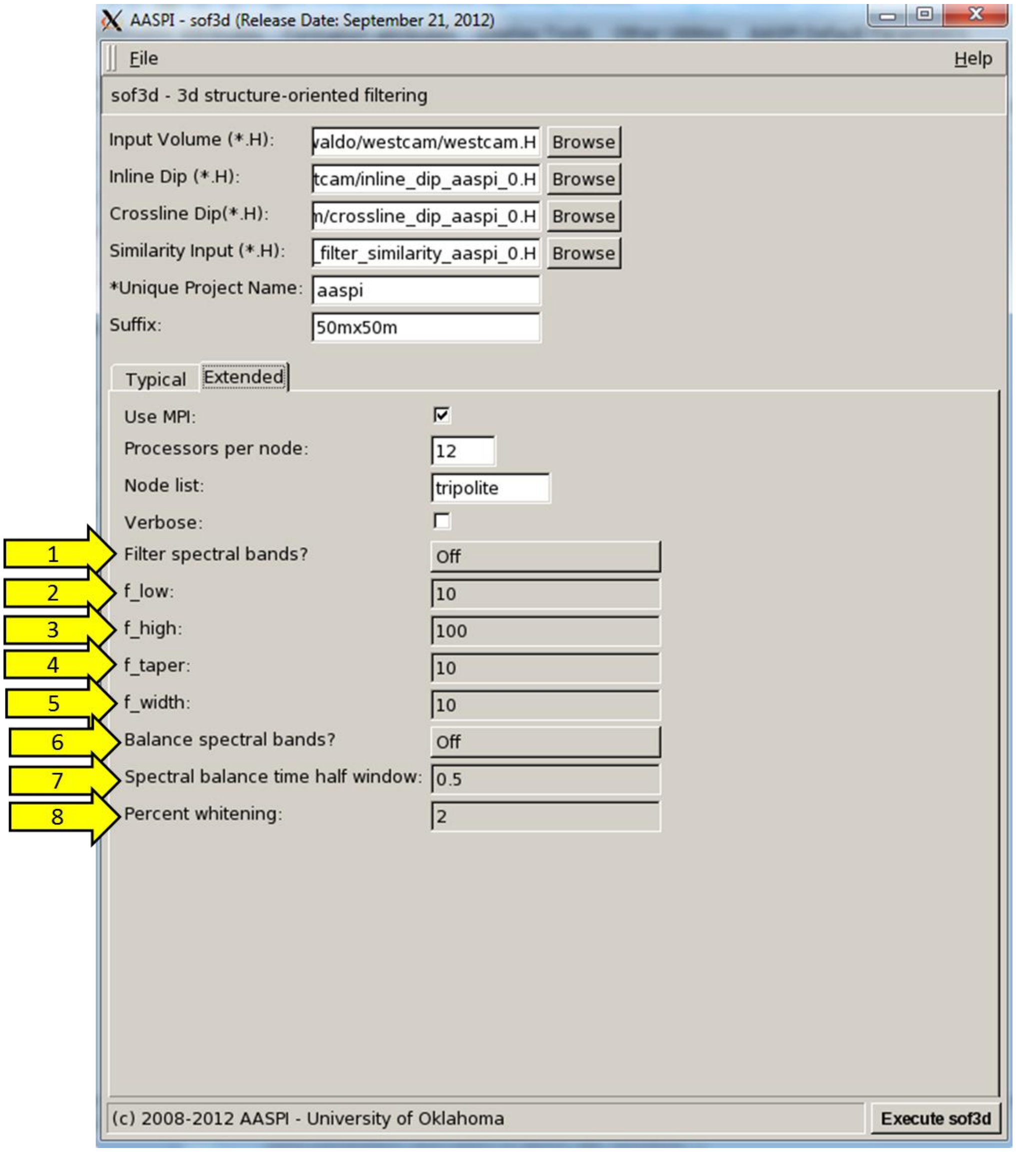


results in the following image:



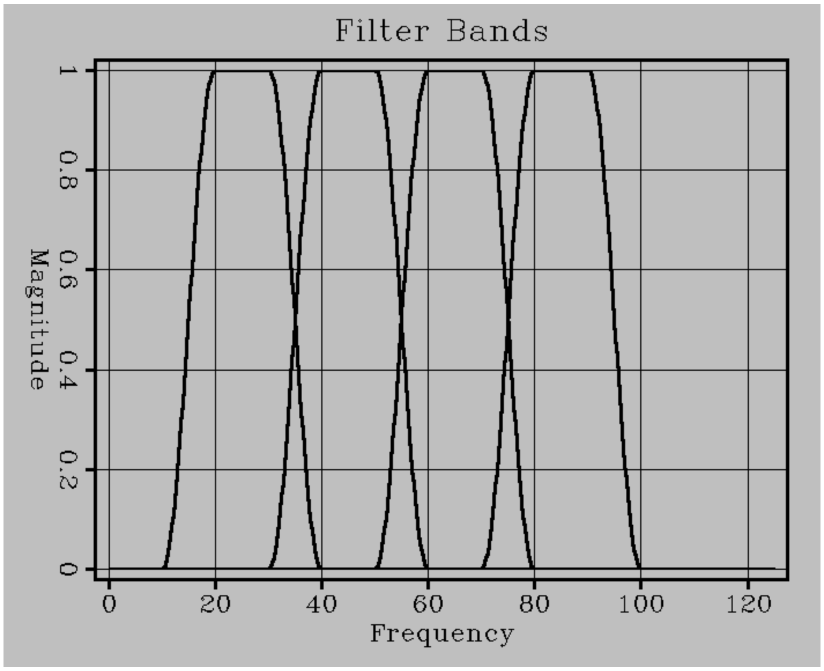
The default structure-oriented filter is all-pass in the frequency domain, ranging from 0 Hz to Nyquist. Helmore (2009) proposed a slightly different workflow that used a single dip-azimuth computation from the broad-band data, but applied structure-oriented filtering to a suite of band-pass filtered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

In our 2011 release we implemented some of Helmore's (2009) concepts. Reexamining the **sof3d** GUI,

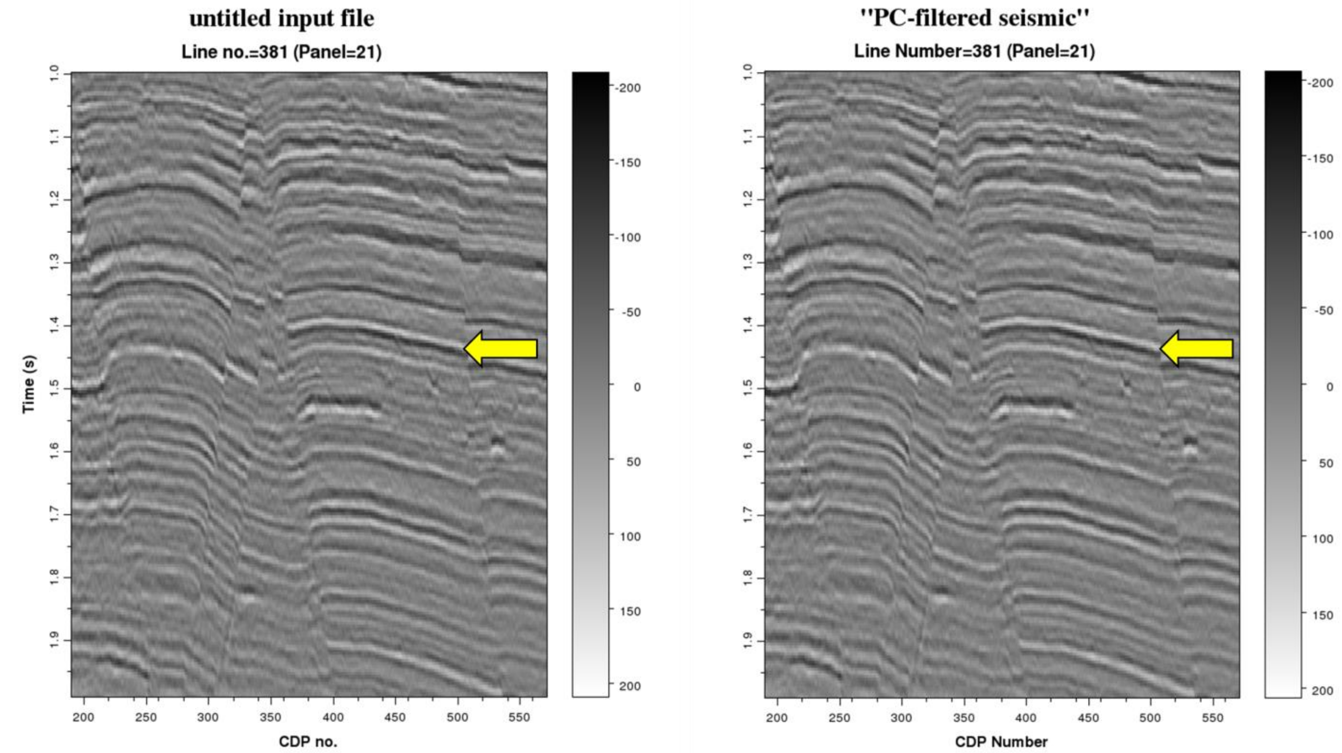


In the *'Extended'* tab, first, click the button (1) to turn the *Filter spectral bands?* option on. The default values of low and high frequencies, frequency taper for each band, and width of the untapered portion of each band (2)-(5) above are reasonable for the Westcam survey. If you wish to spectrally 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.

Reinvoking the Graph Display tool with the name filter_bands_ westcam_balanced.H displays the actual filter bands defined by the parameters in the previous GUI:



Arrows indicate two locations where the spectral balancing has improved the vertical discrimination of previously hidden reflection events.

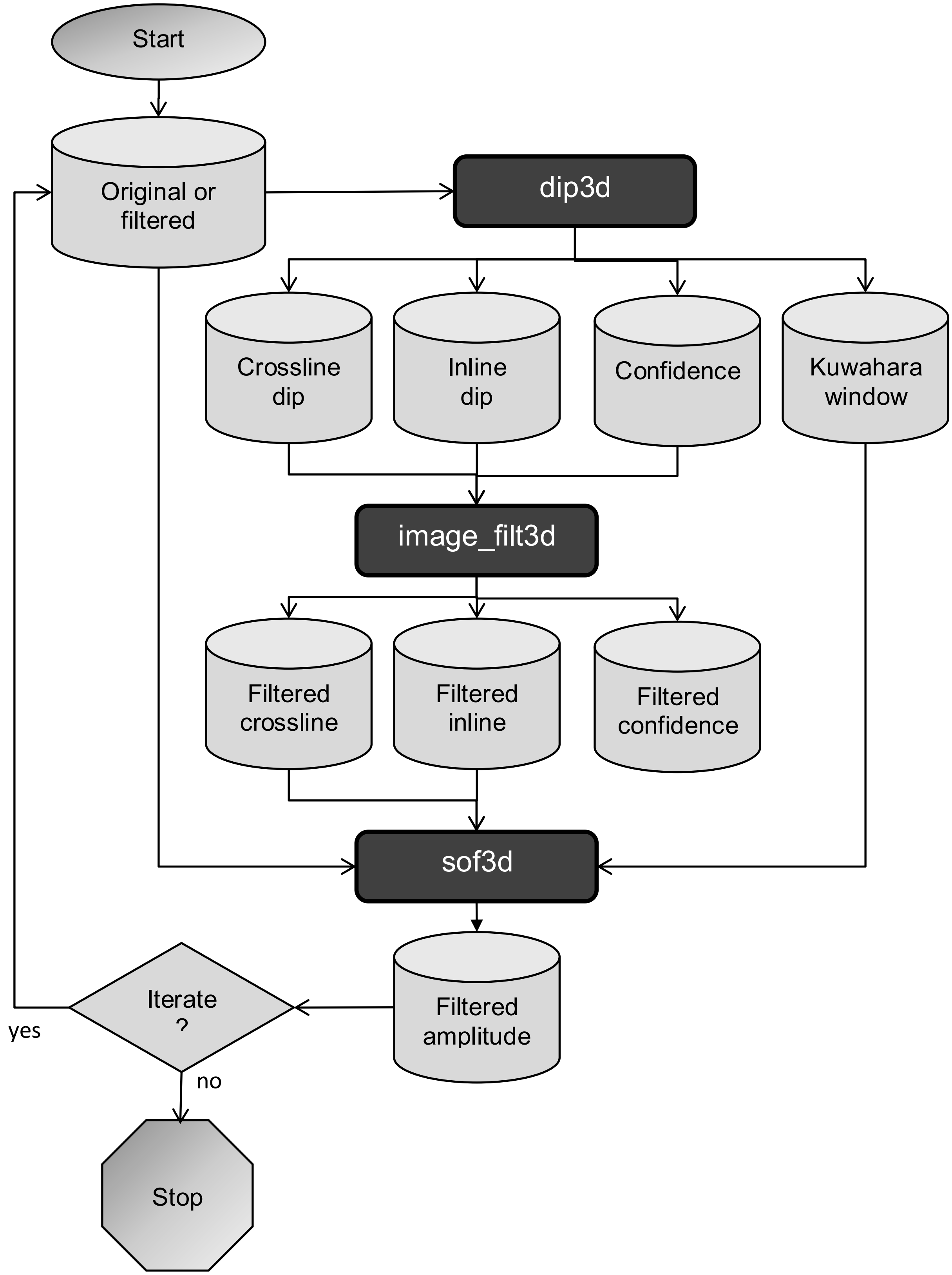


Original Data

Spectrally-balanced data.
Two passes of PC SOF 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:

Volumetric Attributes-sof3d



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., and K. J. Marfurt, 2011, Footprint suppression applied to legacy seismic data volumes: to appear in the GCSSEPM 31st Annual Bob. F. Perkins Research Conference.

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.