IMPROVING DIP ESTIMATES – PROGRAM filter_dip_components

Contents

Motivation

An accurate estimate of dip is critical to the computation of geometric attributes. Attributes such as coherence, amplitude gradients, gray level co-occurrence matrices, and principal component, mean, median structure-oriented filters are computed along structural dip. Curvature and aberrancy are computed *from* structural dip such that these two attributes are particularly sensitive to spikes in the dip estimate volumes. Program **filter_dip_components** provides a means to minimize these effects, resulting in more stable estimates.

Definition of dip components, dip magnitude and dip azimuth

As the name implies, the inline_dip is an apparent dip along the inline axis whose value is positive down in the increasing CDP Number direction. The crossline_dip is an apparent dip along the crossline axis whose value is positive down in the increasing Line Number direction. These apparent dip components are measured in degrees from the horizontal. If the seismic amplitude data are depth migrated or stretched to depth from a time-migrated volume, these dip angles represent those seen in the subsurface seismic image. If the seismic amplitude data are time migrated, the data are internally (and crudely) converted to depth using a single constant conversion velocity provided as input to the program **dip3d** GUI.

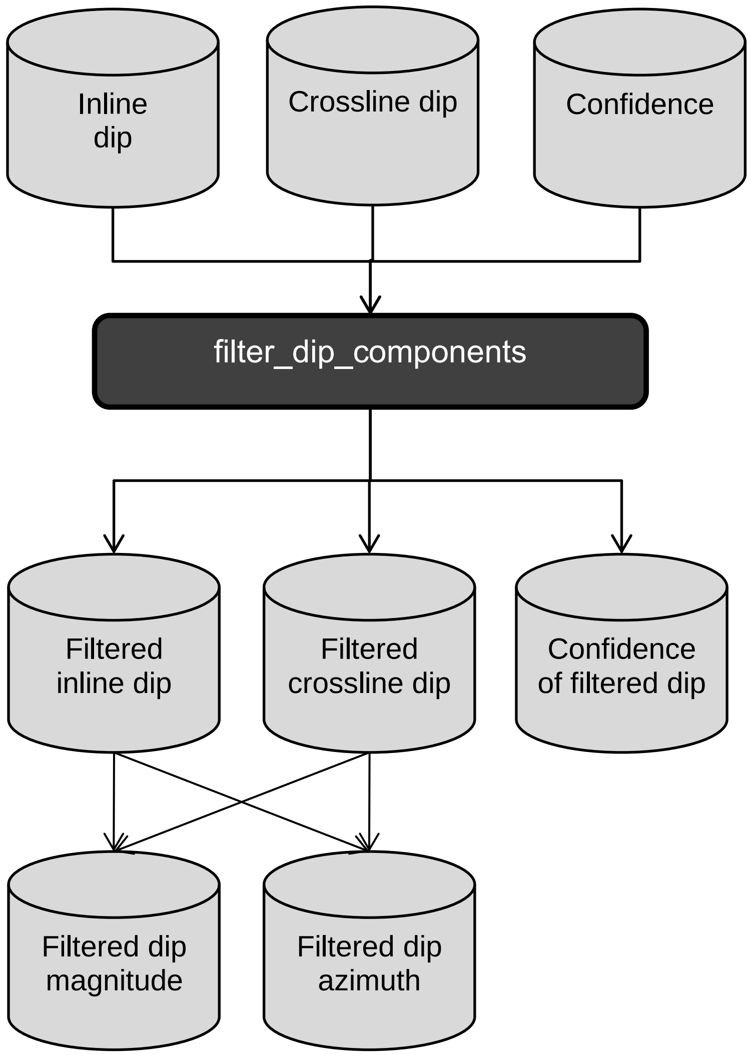
Like all other AASPI programs, the dip azimuth ranges between -180° and +180° and is defined clockwise from North, where North is at 0°, East at 90°, West at -90°, and South at ±180°. The discontinuity in the azimuth about ±180° is best addressed through the use of a cyclical color bar. Avoiding the discontinuity in flatting or generating stratal slices requires interpolating inline and crossline (continuous) apparent dip components using both the dip azimuth and dip magnitude volumes as input to programs **vector_flatten** or **vector_stratal_slice**.

The dip magnitude is displayed in degrees measured from the horizontal. Internally, the computation is obtained by converting the apparent dip components in degrees to apparent dip components in m/m or ft/ft, computing the square root of the sum of the squares and converting back to degrees, using the same velocity value (or model) as used in computing the apparent dip components.

Computation flow chart

The input to program **filter_dip_components** is generally computed from program **dip3d** and includes estimates of the inline and crossline dip components as well as the confidence of these estimates. The confidence computed in program **dip3d** is simply the semblance along dip of the analytic traces that fall within the (potentially uncentered) Kuwahara window used in the computation. Program **filter_dip_components** can be run iteratively, whereby the output can be used as input for the next iteration.

Geometric attributes: Program **filter_dip_components**



Output file naming convention

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

Output file description	**File name syntax**
Program log information	filter_dip_components_*unique_project_name_suffix*.log
Program error/completion information	Filter_dip_components_*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 the filters selected, **filter_dip_components** will also generate one or more of the following output volume triplets:

Geometric attributes: Program **filter_dip_components**

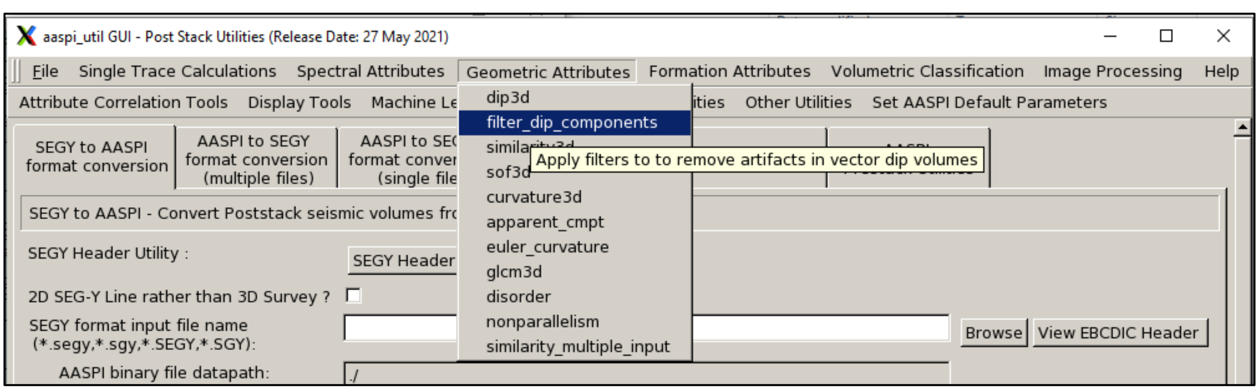
Output file description	**File name syntax**
Inline mean-filtered component of vector dip	inline_dip_mean_filt_*unique_project_name_suffix*.H
Crossline mean-filtered component of vector dip	crossline_dip_mean_filt_*unique_project_name_suffix*.H
Confidence of the mean-filtered dip estimate	conf_mean_filt_*unique_project_name_suffix.*H
Inline alpha-trimmed mean-filtered component of vector dip	inline_dip_alpha_trimmed_mean_filt_*unique_project_name_suffix*.H
Crossline alpha-trimmed mean-filtered component of vector dip	crossline_dip_alpha_trimmed_mean_filt_*unique_project_name_suffix*.H
Confidence of the alpha-trimmed mean filtered dip estimate	conf_alpha-trimmed_mean_filt_*unique_project_name_suffix*.H
Inline MSMTM-filtered component of vector dip	inline_dip_msmtm_filt_*unique_project_name_suffix*.H
Crossline MSMTM-filtered component of vector dip	crossline_dip_msmtm_filt_*unique_project_name_suffix*.H
Confidence of the MSMTM-filtered dip estimate	conf_msmtm_filt_*unique_project_name_suffix*.H
Inline LUM-filtered component of vector dip	inline_dip_lum_filt_*unique_project_name_suffix*.H
Crossline LUM-filtered component of vector dip	crossline_dip_lum_filt_*unique_project_name_suffix*.H

If selected, you can also output the following optional filtered dip azimuth and dip magnitude volumes:

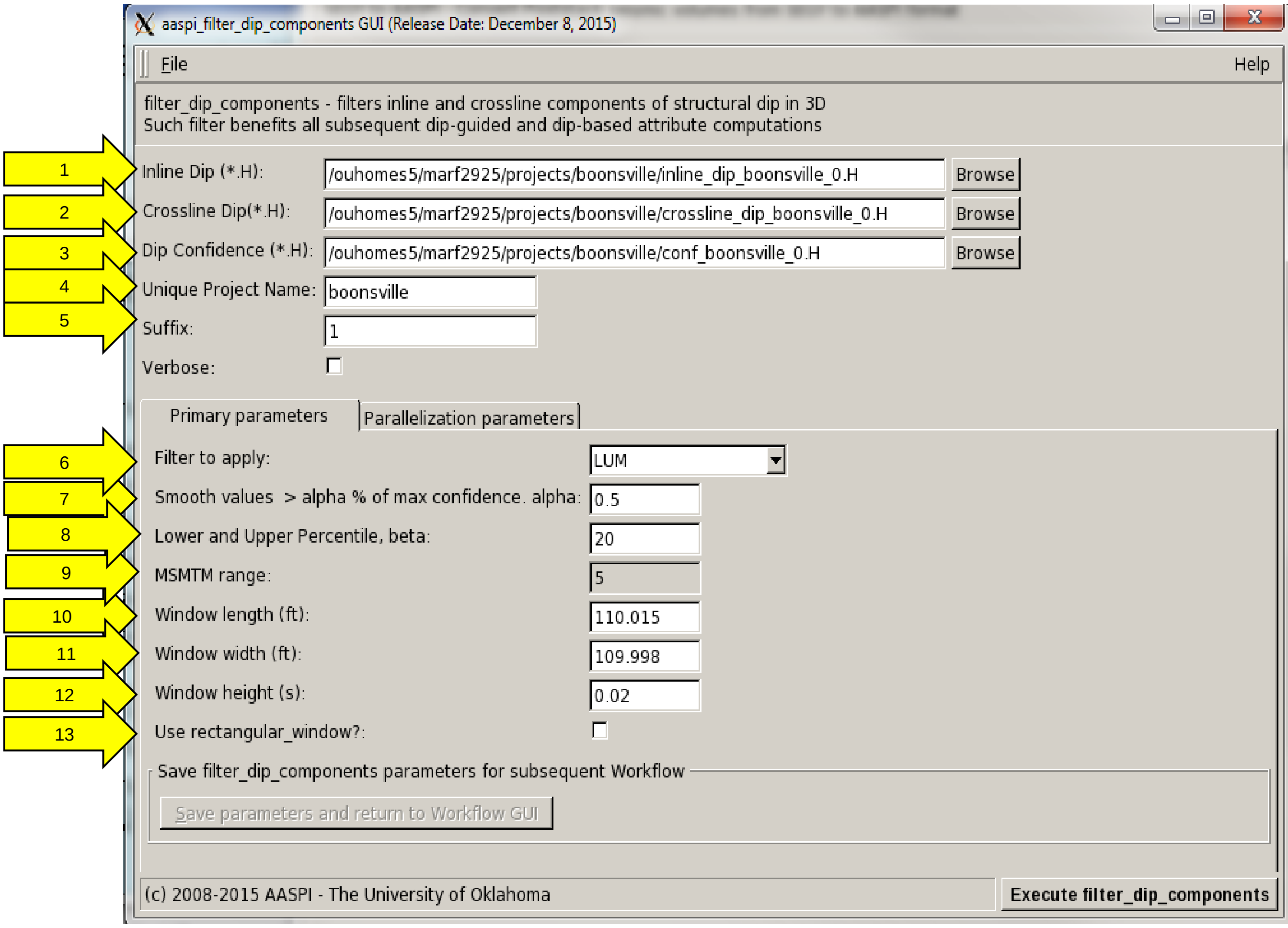
Output file description	**File name syntax**
Dip azimuth mean-filtered vector dip	dip_azimuth_mean_filt_*unique_project_name_suffix*.H
Dip magnitude mean-filtered vector dip	dip_magnitude_mean_filt_*unique_project_name_suffix*.H
Dip azimuth alpha-trimmed mean-filtered vector dip	dip_azimuth_alpha_trimmed_mean_filt_*unique_project_name_suffix*.H
Dip magnitude alpha-trimmed mean-filtered vector dip	dip_magnitude_alpha_trimmed_mean_filt_*unique_project_name_suffix*.H
Dip azimuth MSMTM-filtered vector dip	dip_azimuth_msmtm_filt_*unique_project_name_suffix*.H
Dip magnitude MSMTM-filtered vector dip	dip_magnitude_msmtm_filt_*unique_project_name_suffix*.H
Dip azimuth LUM-filtered vector dip	dip_azimuth_lum_filt_*unique_project_name_suffix*.H
Dip magnitude LUM-filtered vector dip	dip_magnitude_lum_filt_*unique_project_name_suffix*.H

Invoking the filter_dip_components GUI

In the **aaspi_util** GUI, under the *Geometric Attributes* tab, choose program **filter_dip_components.**



The following window appears:



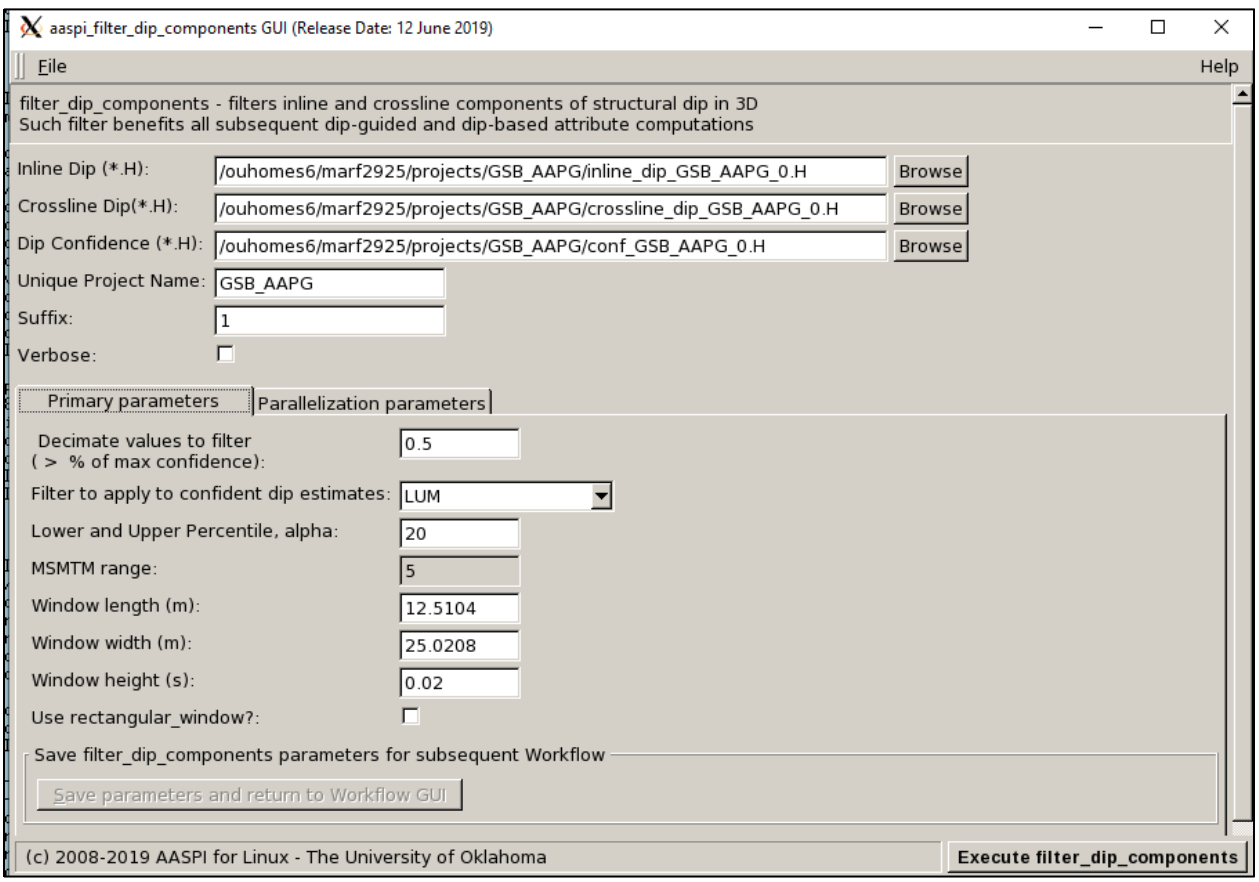
filter_dip_components has three input files: (1) inline and (2) crossline components of dip and the (3) confidence (analytic semblance) of the estimate. There will be three output files – the filtered inline and crossline components of dip and an updated confidence estimate. I've set the *Suffix* to be '1' indicating that this is the first pass of filtering. The possible filters at present include LUM (lower-upper-middle), MSMTM (multistage median-based modified trimmed mean), *median* and *mean* filters. al-Dossary and Marfurt (2007) show the applicability of LUM and MSMTM filters.

Among the parameters, (7) the confidence, *alpha*, is active for all the filters in the list; *alpha* does not work on the values of dip as in an alpha-trimmed mean filter, but rather on the confidence estimate. For the default value of *alpha=0.5*, the values that fall within the analysis window are sorted according to their confidence. If the confidence falls below *alpha=0.5* of the most confident estimate of dip, we reject it. For those values for which we are quite confident, we take the selected filtered value as our output. The default window size consists of the neighboring traces and samples, in this case +/-25m and +/- 0.02 s.

If you have selected the LUM filter, then the (8) *beta* value becomes active. If we set *beta* to be 50%, the result will be the same as using the median filter, where as if we set it to 0%, the result will be as if we had not filtered the data. If we set *beta* to be between 0 – 50%, for example 20%, then values which fall between 20 – 80 % of the confidence estimate will be kept. Values that fall below 20% of the confidence estimate will be set to the lower threshold 20% confidence value, and values that fall above 80% of the confidence estimate will be set to the upper threshold 80% confidence value. Values that fall below our lower threshold and above our upper threshold will be clipped.

The MSMTM (Multistage median-based modified trimmed mean) filter is able to preserve detail, meaning it acts as an edge preserving filter, a lineament preserving filter and can smoothen noise. The MSMTM is a modified trimmed mean (MTM) filter that implements a multistage median filter (MSM). A data sample's value is kept if it lies in the range of [*m – q, m + q*], where *m* is calculated using a MSM filter and *q* is a user defined range. Larger values of *q* result in some smearing of lineaments through higher amplitude "noise" areas, while smaller values of *q* better preserve narrow lineaments. For further discussion, please refer to al-Dossary and Marfurt (2007).

The *Parallelization parameters* panel only asks for the list of nodes and the number of processors per node:



Like all AASPI codes, click *Execute* and intermediate information will be printed in the xterm from which **aaspi_util** was launched:

```
  8:data preloaded
  6:data preloaded
  7:data preloaded
 0: first_line,current_line,last_line,ETA      105     110     201     0.003 h
 0: first_line,current_line,last_line,ETA      105     120     201     0.003 h
 0: first_line,current_line,last_line,ETA      105     130     201     0.002 h
 0: first_line,current_line,last_line,ETA      105     140     201     0.002 h
 0: first_line,current_line,last_line,ETA      105     150     201     0.002 h
 0: first_line,current_line,last_line,ETA      105     160     201     0.001 h
 0: first_line,current_line,last_line,ETA      105     170     201     0.001 h
 0: first_line,current_line,last_line,ETA      105     180     201     0.001 h
 0: first_line,current_line,last_line,ETA      105     190     201     0.000 h
 0: first_line,current_line,last_line,ETA      105     200     201     0.000 h
          1 :end loop over lines
          1 number of traces processed:         1649
 process                                  task       time (hr)  time/trace (s)
  1:                               read  data        0.000          0.000
  1:                        send data via MPI        0.000          0.000
  1:                     receive data via MPI        0.000          0.000
  1:                     send results via MPI        0.000          0.000
  1:                  receive results via MPI        0.000          0.000
  1:                     calculate attributes        0.000          0.000
  1:                    write results to disk        0.000          0.000
  1:                               total time        0.004          0.008
          1 : memory residing only on slaves deallocated
          1 : attempt to deallocate p_out
          1 : attempt to deallocate q_out
          1 : attempt to deallocate conf_out
          1 : attempt to deallocate line_index
line_index deallocated
in_memory deallocated
lag_interp deallocated
t_lag_interp deallocated
t_lag_interp,start_cdp,end_cdp  deallocated
          1 : shared arrays residing on both master and slave deallocated
          8 :end loop over lines
          8 number of traces processed:         1552
 process                                  task       time (hr)  time/trace (s)
  8:                               read  data        0.000          0.000
  8:                        send data via MPI        0.000          0.000
  8:                     receive data via MPI        0.000          0.001
```

Once the job is completed, typing *ls –ltr* at the terminal prompt shows that the following files were created:

```
-rw-r--r--  1 kmarfurt aaspi      31 Aug  3 16:10 live_processor_list
-rw-r--r--  1 kmarfurt aaspi    1921 Aug  3 16:10 inline_dip_median_filt_boonsville_1.H@@
-rw-r--r--  1 kmarfurt aaspi    2987 Aug  3 16:10 inline_dip_median_filt_boonsville_1.H
-rw-r--r--  1 kmarfurt aaspi    1927 Aug  3 16:10 dip_magnitude_median_filt_boonsville_1.H@@
-rw-r--r--  1 kmarfurt aaspi    3023 Aug  3 16:10 dip_magnitude_median_filt_boonsville_1.H
-rw-r--r--  1 kmarfurt aaspi    1925 Aug  3 16:10 dip_azimuth_median_filt_boonsville_1.H@@
-rw-r--r--  1 kmarfurt aaspi    3040 Aug  3 16:10 dip_azimuth_median_filt_boonsville_1.H
-rw-r--r--  1 kmarfurt aaspi    1927 Aug  3 16:10 crossline_dip_median_filt_boonsville_1.H@@
-rw-r--r--  1 kmarfurt aaspi    3005 Aug  3 16:10 crossline_dip_median_filt_boonsville_1.H
-rw-r--r--  1 kmarfurt aaspi    1909 Aug  3 16:10 conf_median_filt_boonsville_1.H@@
-rw-r--r--  1 kmarfurt aaspi    2776 Aug  3 16:10 conf_median_filt_boonsville_1.H
-rw-r--r--  1 kmarfurt aaspi   22535 Aug  3 16:11 image_filt3d_boonsville_1.out
[kmarfurt@opal boonsville]$
```

Note that we have created filtered versions of the inline dip and crossline dip components. The part of the name *median_filt* denotes the kind of filter that was applied. Had we applied a LUM filter, we would see *lum_filt* instead. Program **filter_dip_components** also generates new versions of dip magnitude and dip azimuth computed from the filtered dip component volumes.

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:

$$\mu = \frac{1}{J}\sum_{j=1}^{J} d_j \ . \quad (1)$$

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 $u_k \le u_{k+1}$:

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

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

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

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

$$0 \le \alpha \le \frac{1}{2} \ . \quad (4)$$

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+αJ ≤j≤J-α)(J-1)*:

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

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

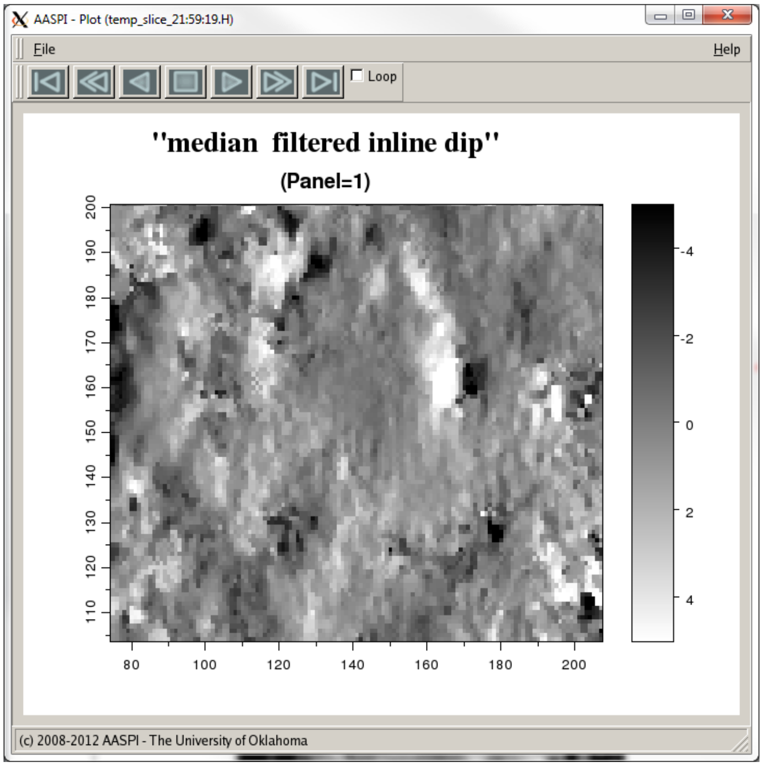
The Lower-Upper-Median (LUM) filter:

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

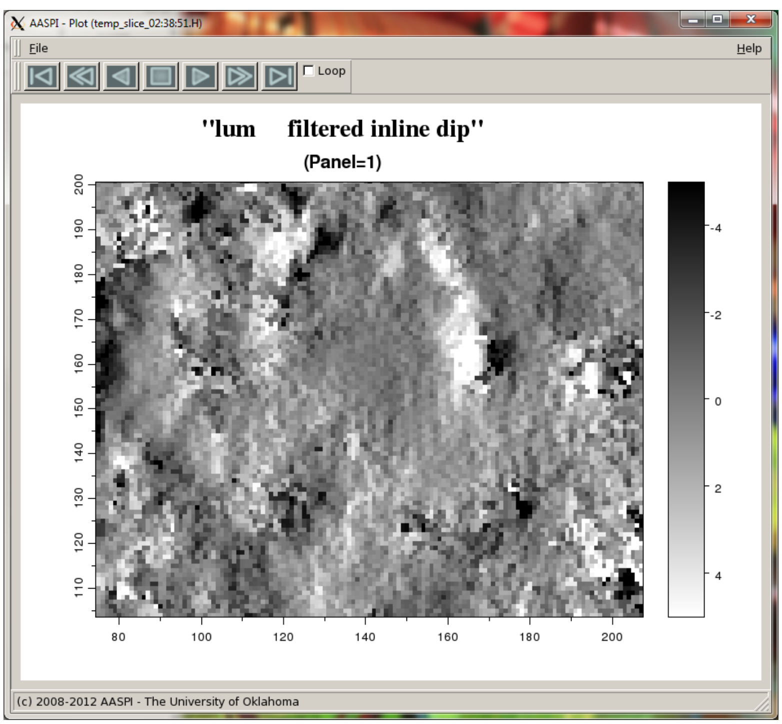
$$u_{LUM} = \mathrm{median}\left(u_{1+\alpha(J-1)}, u^*, u_{J-\alpha(J-1)}\right) = \begin{cases} u_{1+\alpha(J-1)} & u^* < u_{1+\alpha(J-1)} \\ u_{J-\alpha(J-1)} & u^* > u_{J-\alpha(J-1)} \ 0 \le \alpha \le 0.5 \\ u^* & otherwise \end{cases} \quad (6)$$

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.

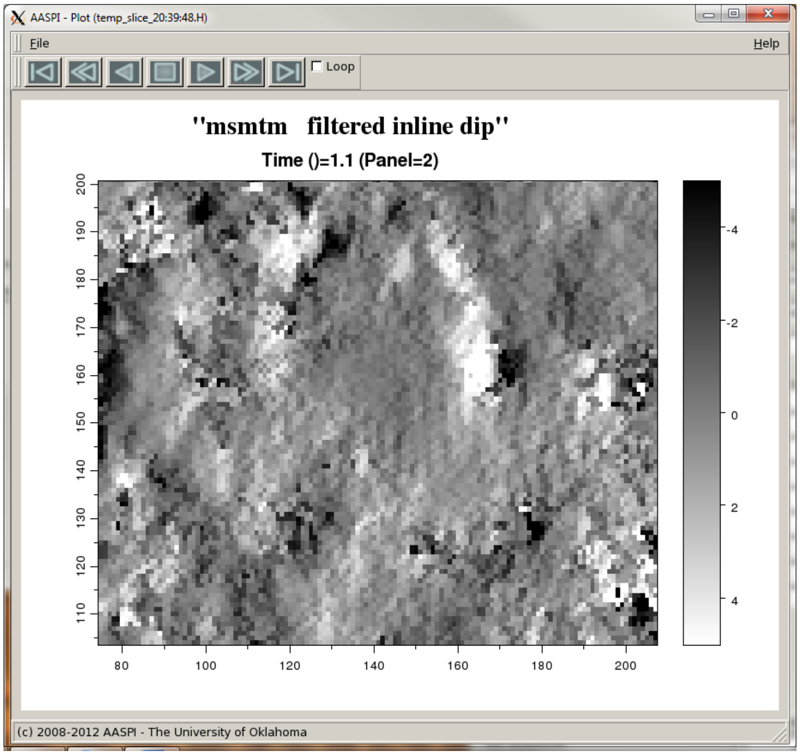
The results of the median filter look like this (time slice, t = 1.1 sec):



Here we see what the result of the LUM filter looks like (time slice, t = 1.1 sec):



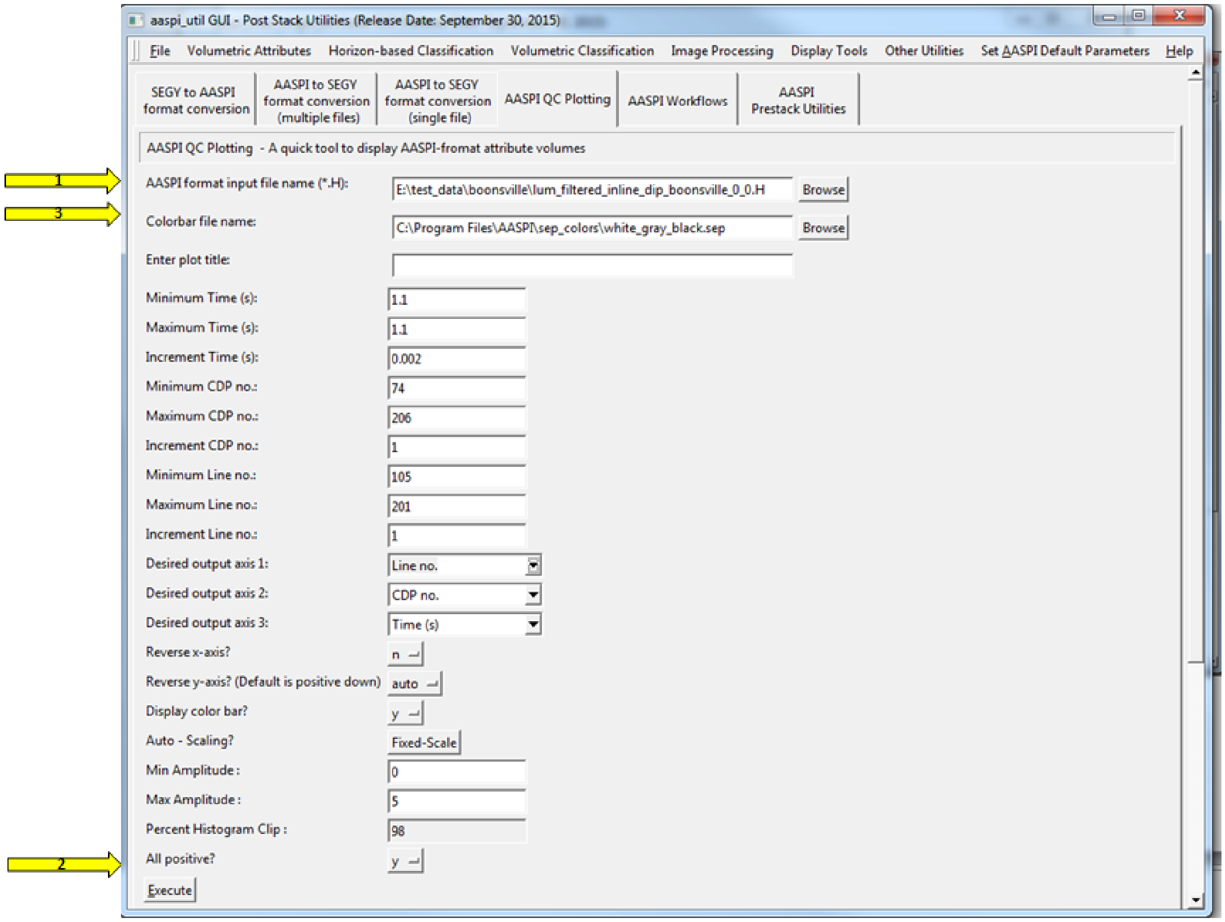
And here is the result of the MSMTM filter with *q* = 4 (time slice, t = 1.1 sec)



We note that the median filtered image is overall less noisy and smoother, with a little less N-S acquisition footprint. However, it also has somewhat lower resolution than the input image shown previously. In comparison to the median filter, the LUM filtered image shows more acquisition footprint, but it has enhanced the collapse features as well. The MSMTM filter improves in regards to the footprint and shows better details near the collapse features.

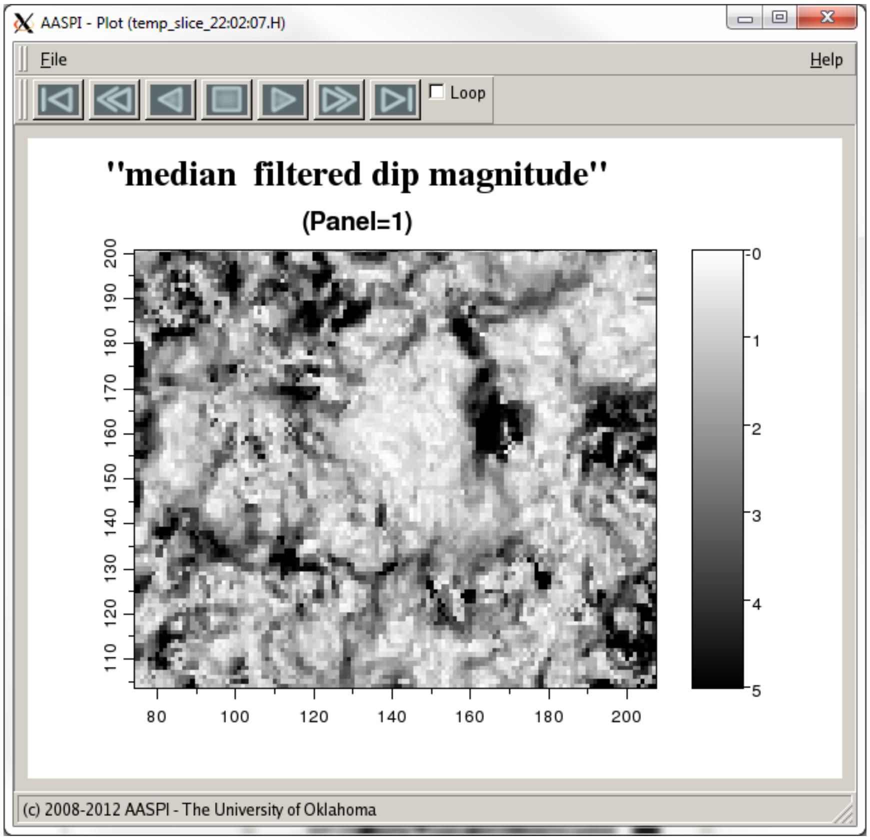
Let's now plot the filtered dip magnitude. Return to the main ***aaspi_util*** GUI and select the tab titled *'AASPI QC Plotting'*:

Geometric attributes: Program **filter_dip_components**



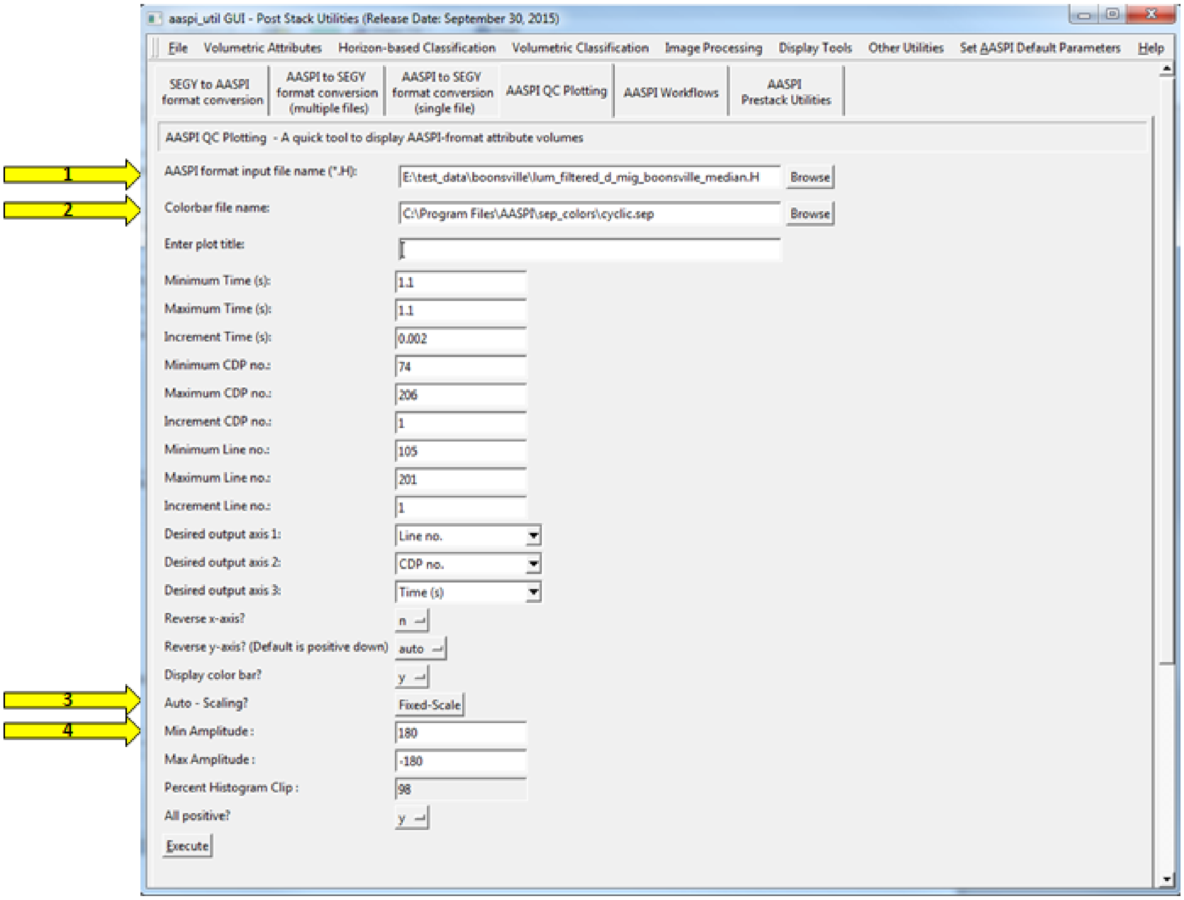
First select the file *dip_magnitude_median_filt_boonsville_1.H to* plot. Dip magnitude will be strictly positive, so we will want to (2) set the *All positive?* option to *y*. Finally (3) we will want to plot dip magnitude against a white-gray-black colorbar so that flat dips appear as white. The resulting image looks like this (see next page):

Geometric attributes: Program **filter_dip_components**



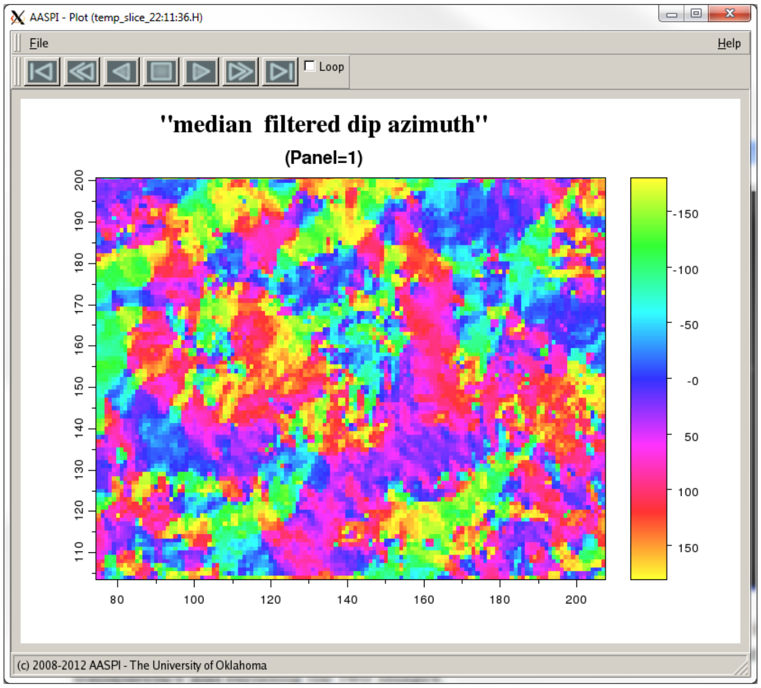
The very steep dip (black areas) corresponds to collapse features. Let's now plot the dip azimuth. Our *AASPI QC Plotting* GUI looks like following image (see next page):

Geometric attributes: Program **filter_dip_components**

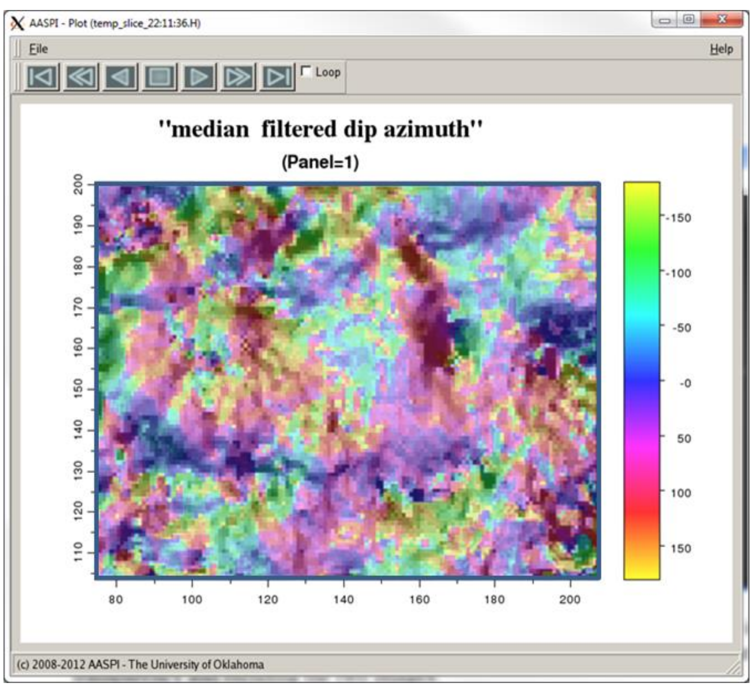


Enter the (1) file name *dip_azimuth_median_filt_boonsville_1.H* as the *AASPI Input*. Then (2) choose the cyclic.sep color bar so that -180^0 will plot with the same color as +180^0 (yellow for this colorbar). (3) Turn the *Auto – Scaling?* to be *Off* to turn off the histogram scaling and instead use explicit clipping. The ranges of these attributes are (4) -180^0 to +180^0.The result will look like the following image (see next page):

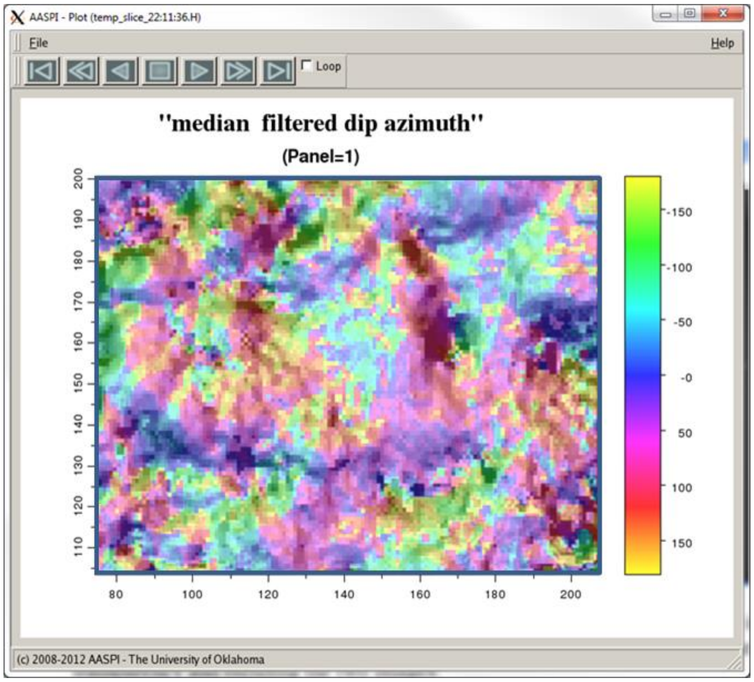
Geometric attributes: Program **filter_dip_components**



A drawback of dip azimuth is that it is meaningless when the dip magnitude is very close to 0.0 (the white areas in the dip magnitude image). We can better visualize these areas by using transparency and blending the two images:



However, this image is somewhat disappointing in that the areas of greater dip where the dip azimuth estimates are accurate are now blackened out. We can ameliorate this problem by plotting the dip magnitude against a black-gray-white color bar, thereby rendering the strong dip-magnitude areas more pastel:



However, the image is still less than ideal.

References

al-Dossary, S., and K. J. Marfurt, 2007, Lineament-preserving filtering: Geophysics, **72**, P1-P8.

Corrao, A., M. Fervari, and M. Galbiati, 2011, Hewett Plattendolomite: Reservoir Characterization by Resolution Enhanced Seismic data: GCSSEPM 31st Annual Bob. F. Perkins Research Conference on Seismic attributes – New views on seismic imaging: Their use in exploration and production, 66-99.