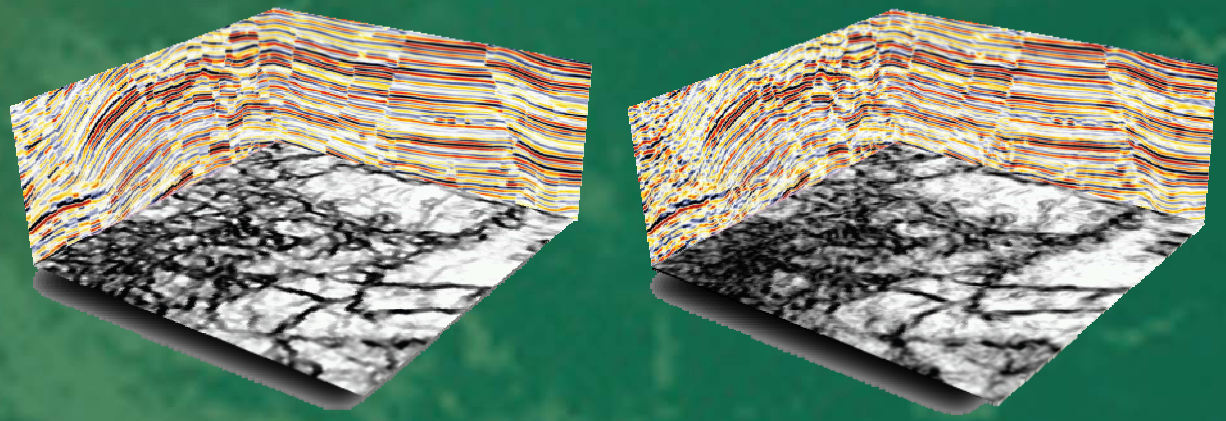
Fault Enhancement Filter

Friso Brouwer



Filtering Seismic Data for improved Fault Detection

Considerations

- A fault enhancing filter should:
  - **Sharpen** the faults
  - **Suppress non-fault discontinuities**, specifically non-geological discontinuities such as random noise and acquisition footprint
  - **Stable**: not create artifacts
  - Easy **tunable**
  - **Fast**

Considerations

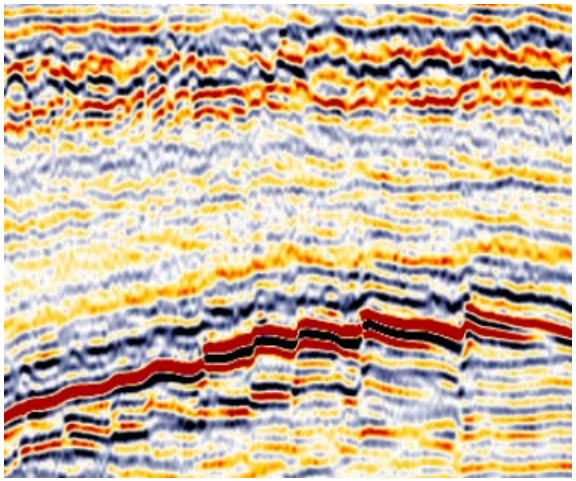
Two standard filters for data enhancement in OpendTect

- **Diffusion** type filter, using the '**position**' attribute
- **Median Dip Filter**, using the '**volume statistics**' attribute
- Because of their respective (dis)advantages we have to implement a combination of these filters to enhance faults

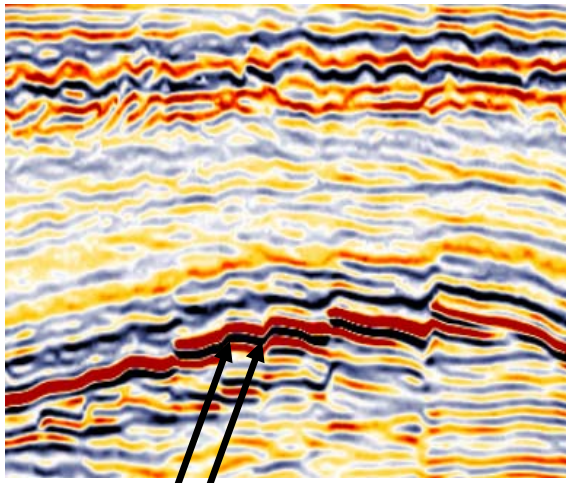
Method	Advantage	Disadvantage
Diffusion	**Sharpens faults** dramatically	Creates **discontinuity artifacts** at noise points
Median Dip Filter	**Eliminates random noise**, reducing background noise level in the fault attributes	**Smooth seismic** across fault boundaries

Considerations

Original

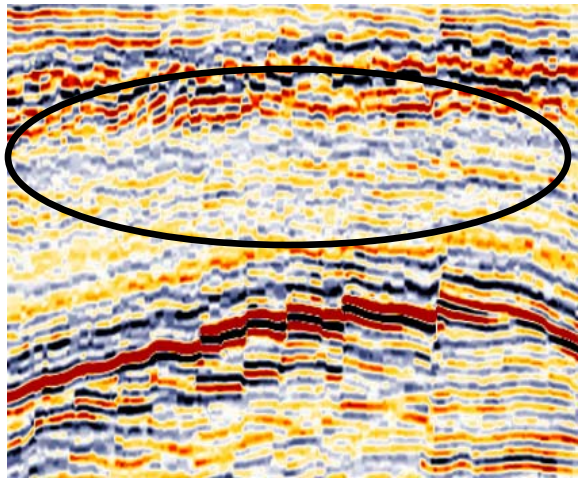


Median Dip Filter



Note that the reflectors are smoothed across the faults at the arrows.

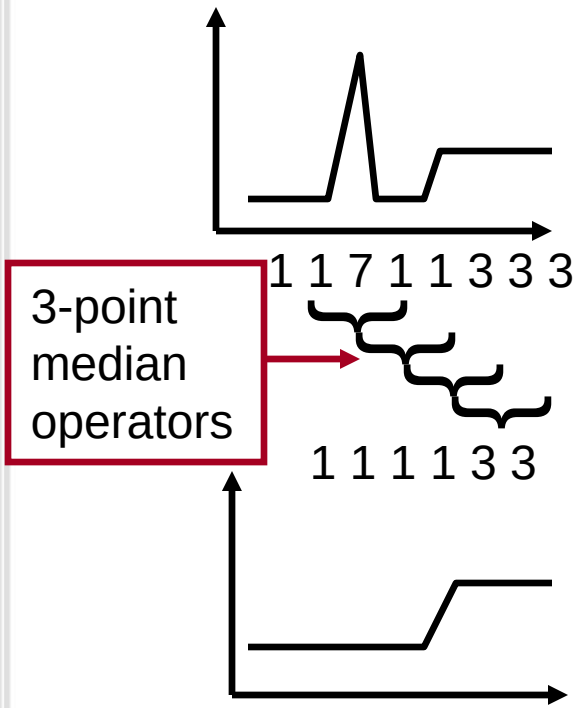
Diffusion



Note that the low amplitude zone with low S/N is very patchy.

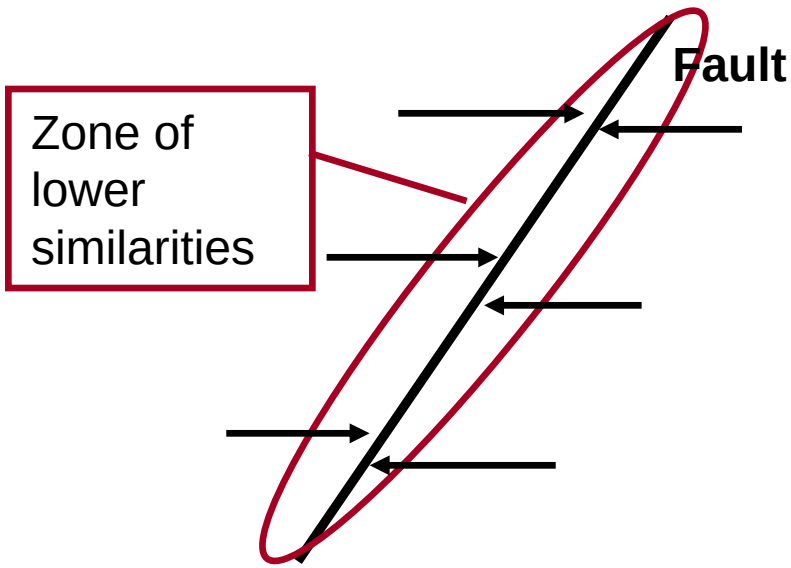
Principles of the filters

Median Dip Filter



Applying a N-point (here N=3) median operator noise spikes are removed, while trends are preserved

Diffusion filter



Amplitudes (information) are migrated toward the fault. Since this happens at both side of the fault, this results in a sharper contrast at the fault

Implementation: concept

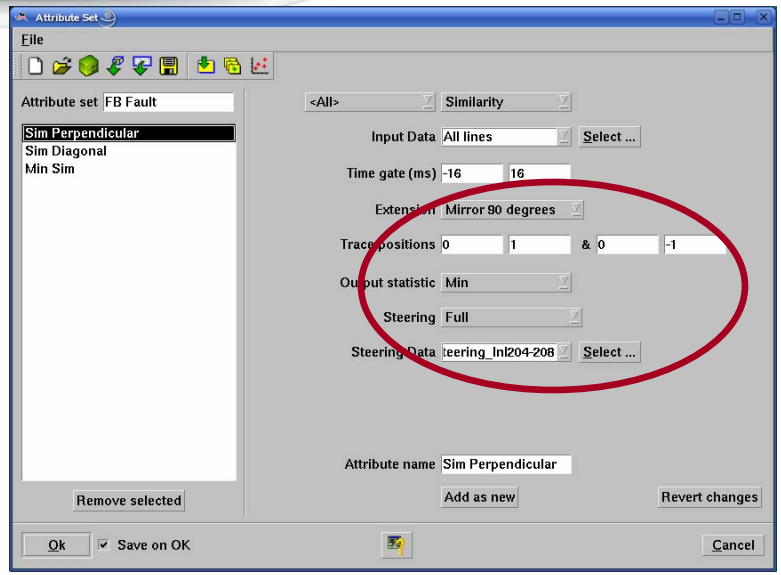
- A **cut-off** value is established for similarity (or another fault attribute)
  - Note that in a more advanced application of the filter one can make the cut-off **adaptive** to for example RMS amplitude, or another indicator for S/N ratio.
- **Below*** the cut-off the **diffusion** filter is applied
- **Above*** the cut-off the **MDF** filter is applied
  - Optionally: for a range around the cut-off value the diffusion and MDF filter can be mixed based on a linear weighing using the similarity as weighting factor. This ensures a smooth transition. This can be implemented using the mathematic attribute.

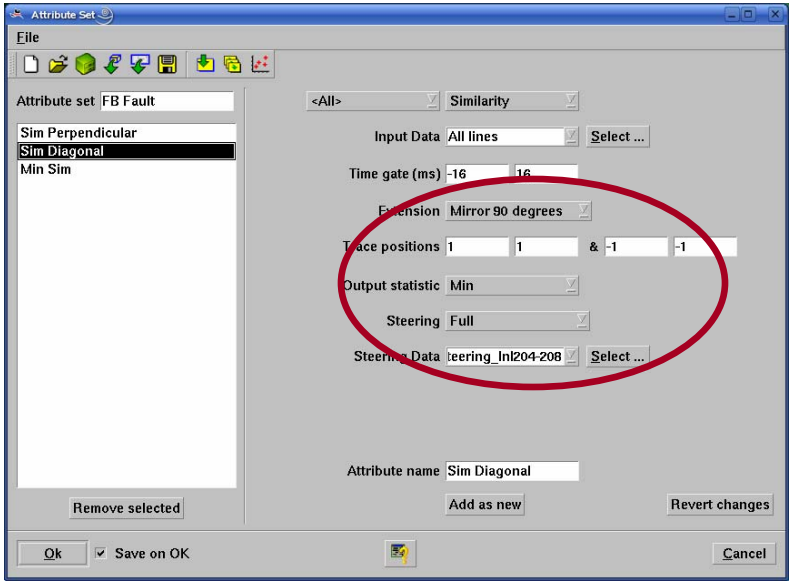
*) Assuming fault attributes where **low** value indicate faults

Implementation: Similarity

- First we build the **fault attribute**, in this example **similarity**
- We assume that we have a Steering Cube – this is the dGB method to correct attributes for dip. Please read the following pdf for more information on how to create a steering cube
- Experiment with the parameters for similarity, especially time and step-out to get the optimum attribute.
- Multiple similarity attributes can be combined using the mathematics attribute
- Often it is a good idea to build one similarity attribute focusing on the inline/crossline directions, one focusing on the diagonal directions and combine these 2 similarity attributes with a minimum operator in the mathematics attribute. This will give the optimum compromise between speed and accuracy. **See the following slides for the practical implementation**

Implementation: Similarity





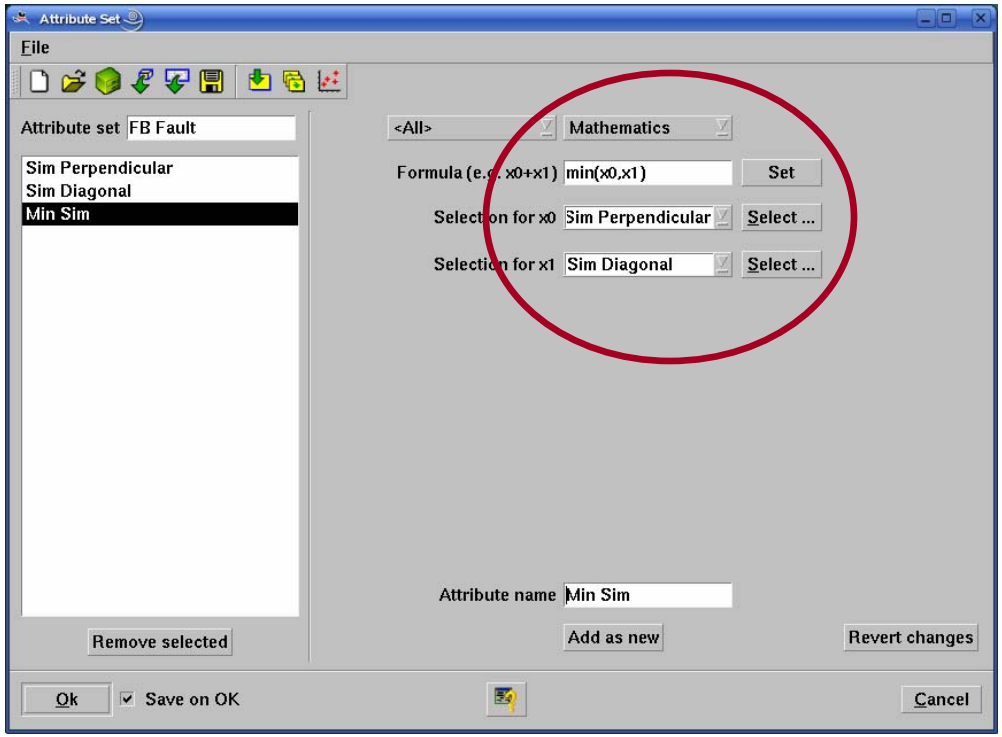
For the perpendicular direction: choose option> **"& 90 degrees"**, with trace pair **(0,1)x(0,-1)** (relative step-out) and the **min** (minimum) operator to select which similarity (0,1)x(0,-1), or (1,0)x(-1,0) will be the output.

For the diagonal direction: choose option> **"& 90 degrees"**, with trace pair **(1,1)x(-1,-1)** (relative step-out) and the **min** (minimum) operator to select which similarity (1,1)x(-1,-1), or (1,-1)x(-1,1) will be the output..

Note that for faults you always should choose the **minimum output** for similarity! Use the **median filtered steering cube**.

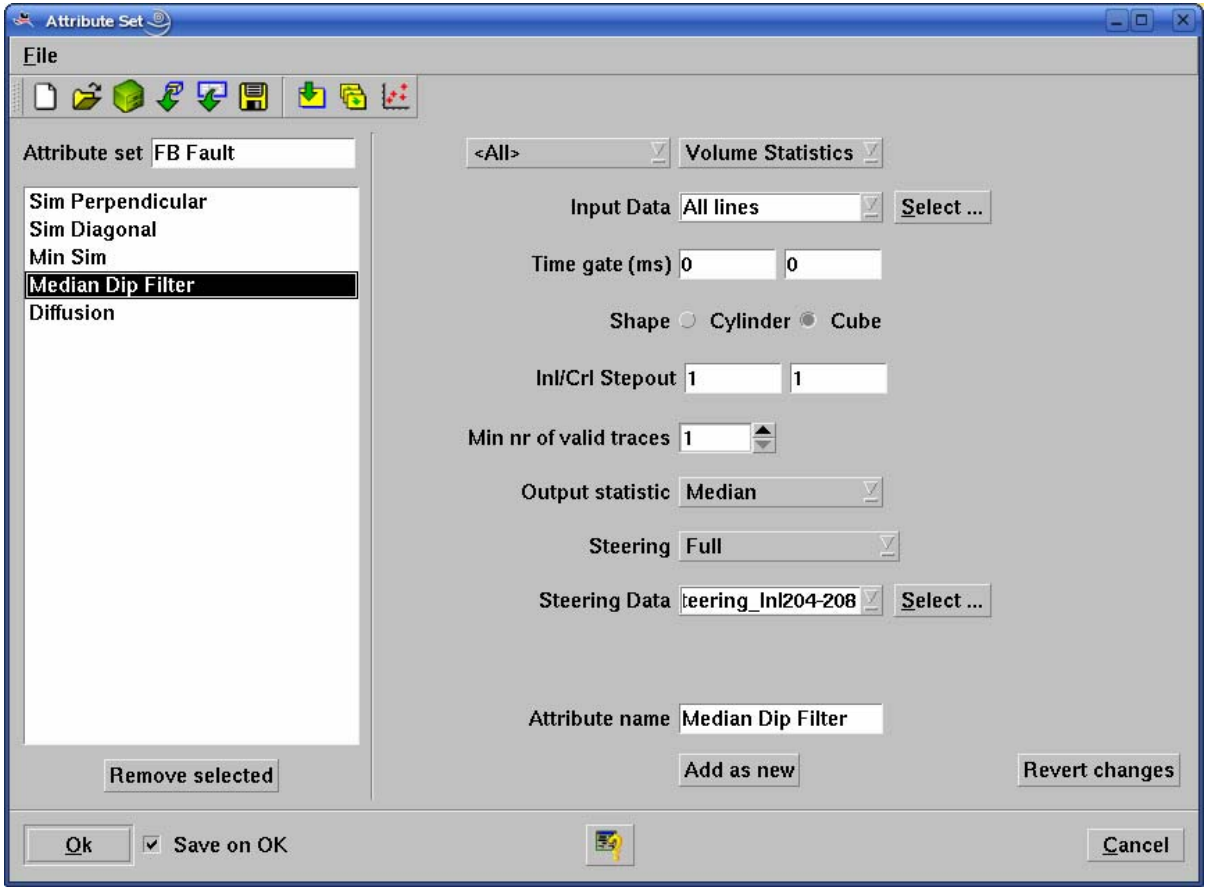
Implementation: Similarity

- The two similarity attributes are combined using the minimum operator in the mathematics attribute.
- The exact syntax of mathematics equations can be found in the OpendTect help



Implementation: MDF Filter

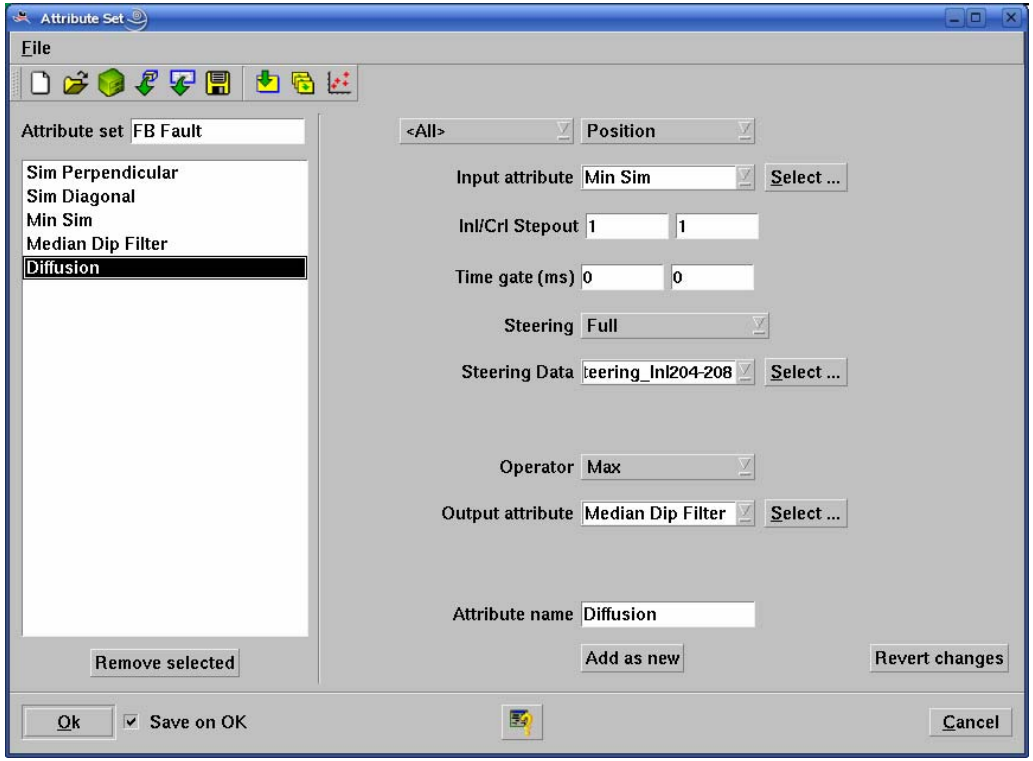
The Median Dip Filter implemented in the attribute window



- Use **Volume Statistics** Attribute
- Input: Seismic
- Choose time gate [0,0]
- Step-out: [1,1] (using the MDF alone one can choose arbitrary large step-outs, in this application limit to 1 trace step-out!)
- Use Full Steering with the **Median Dip Filtered Steering**

Implementation: Diffusion Filter

The Diffusion Filter implemented in the attribute window



- Use **Position Attribute**
- Input: a **fault indicating attribute** - here the Min Sim
- Choose time gate [0,0]
- Step-out: [1,1] (as the MDF)
- Choose the **statistic** for selection of the output position such that the point with “no-fault” is selected – using **similarity** as **input** attribute this is **max**.
- As output attribute one can choose the filtered or the original seismic. When the final filter is a combination with the MDF filter as in this example, choose as output attribute the MDF filter

The Position attribute

- The **input attribute** evaluated at multiple points in a subvolume.
- A search for the location of the minimum, maximum, or median value of the input attribute is conducted.
- **At this location** the second (output) attribute is evaluated. This will be the output value of the position attribute

Numerical example (2D)

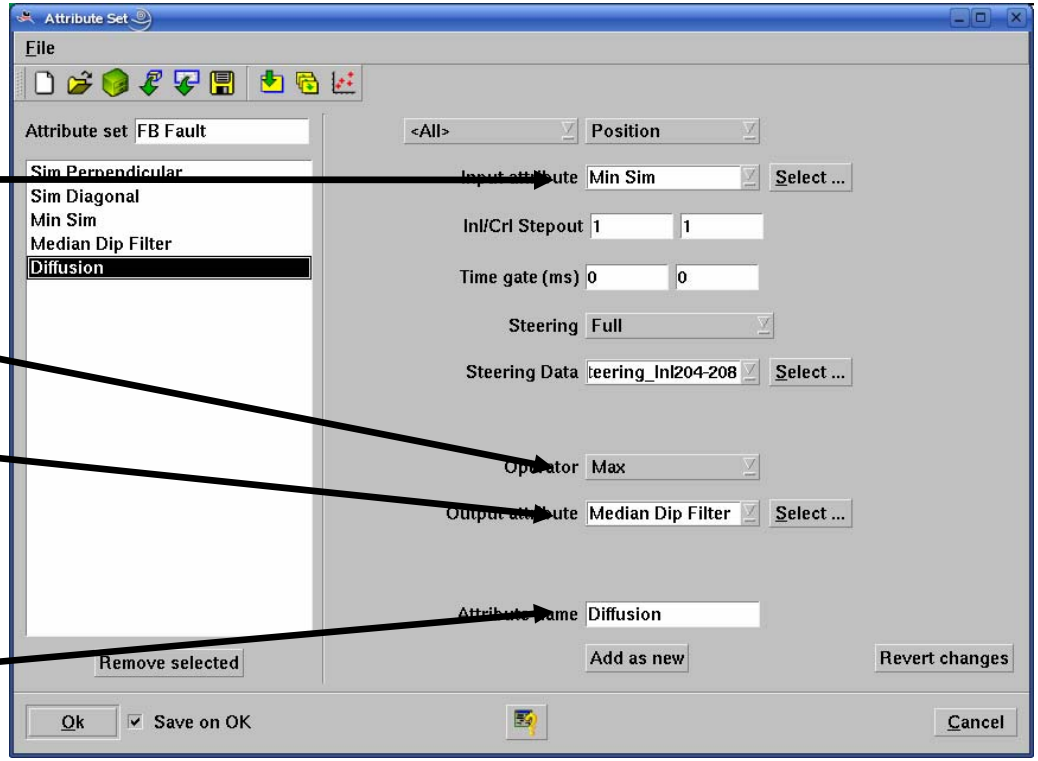
Input attribute: **0.9**, 0.6, 0.4

Statistic: Max

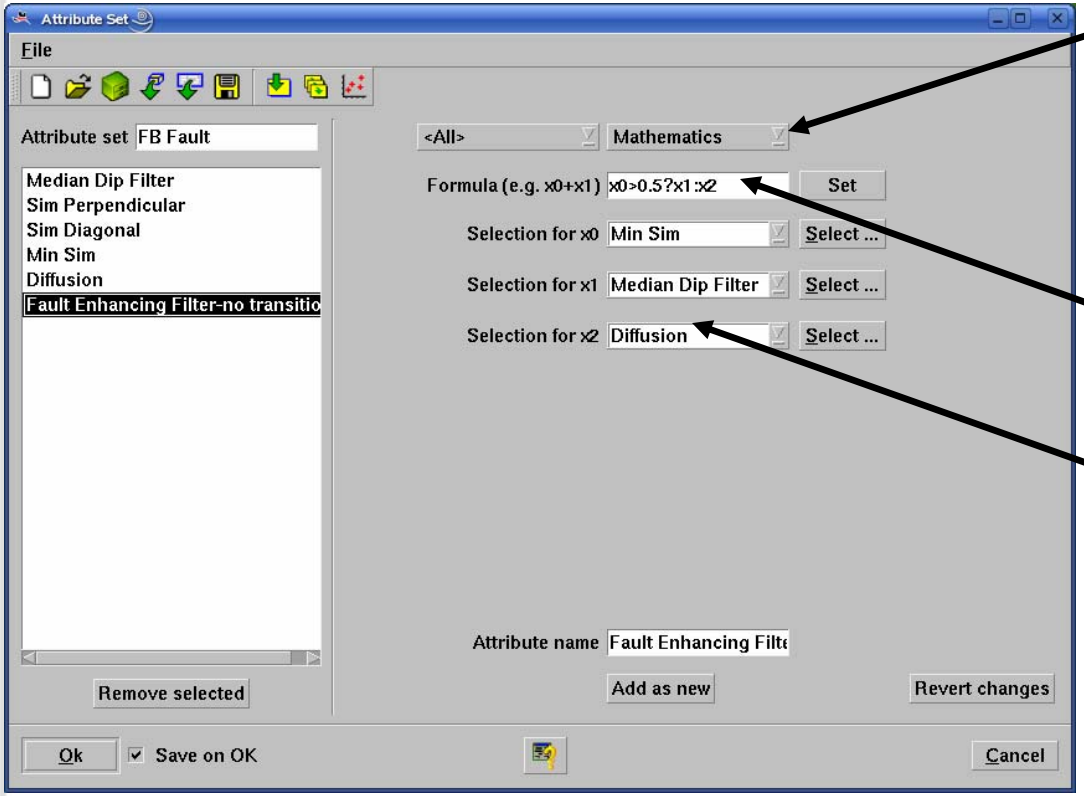
Location of Max: **1**, 0, 0

Output attribute: **4**, 8, 3

Value of the output attribute at the location of the maximum: 4 ; this is the output of the Diffusion attribute

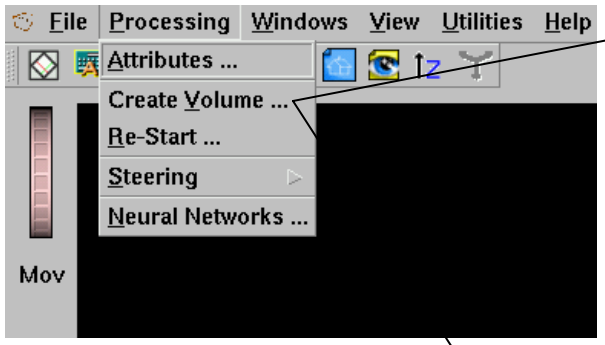


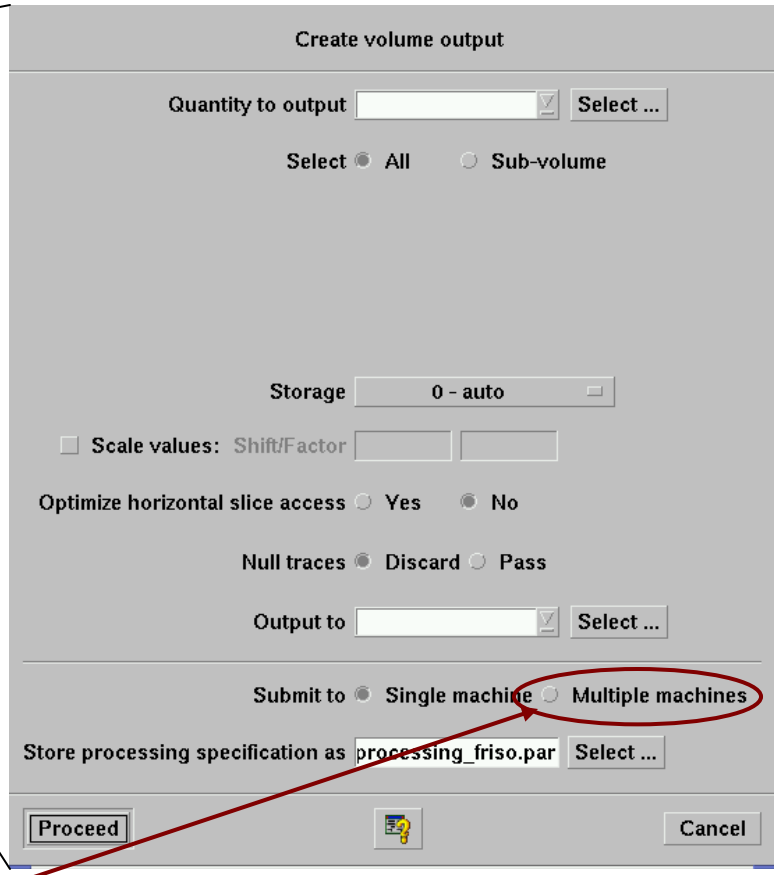
Implementation: combining the filters using a cut-off



- Use the mathematics attribute
- Create an if-then-else expression with the similarity or other fault attribute as cut-off.
- In OD syntax (condition?output1:output2)
- With similarity as fault indictor this reads: **if** similarity>0.5 **then** output Median Dip Filter **else** output Diffusion Filter
- At this point it is a good idea to output the intermediate result as a **volume** –see the next slide how

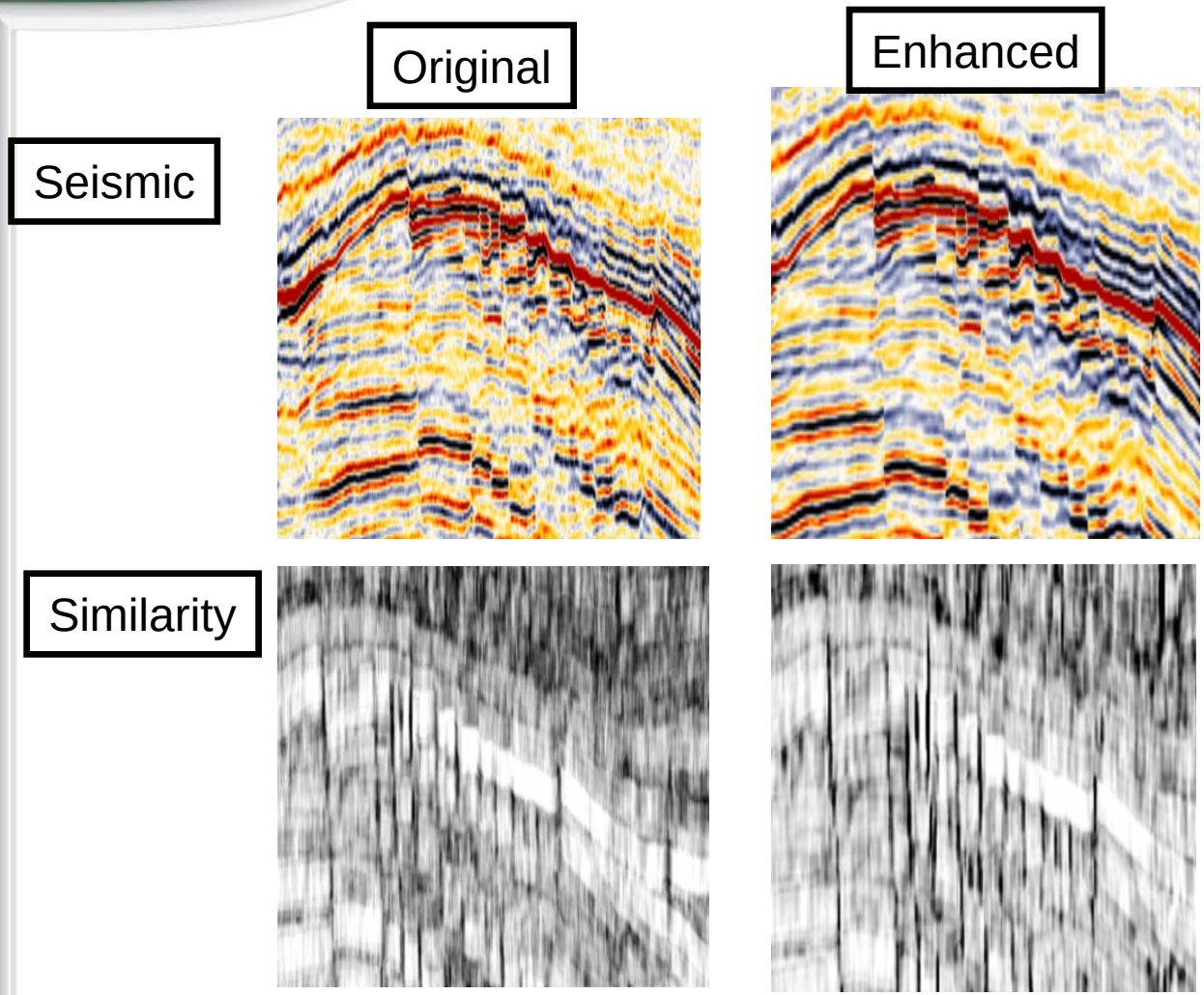
Implementation: apply to volume





OpendTect supports multi-platform **distributed** computing. The system needs to be setup properly to use this scheme. See Administrator's manual.

Enhanced Fault Attribute



- Apply the **fault attribute** (similarity) to the **Fault Enhanced Seismic** volume
- How to build this similarity attribute is explained earlier.
- Output this as a volume

Implementation: iterate

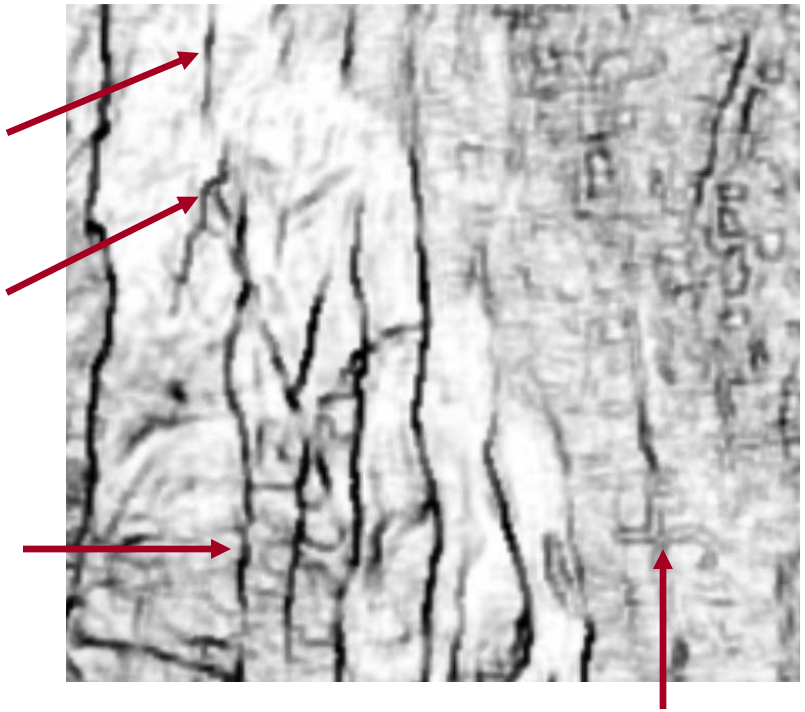
- In most applications one iteration is enough. Except if there is very bad initial data quality.
- The same steps and attributes as in iteration 1 can be used for the second iteration. However, input data and cut-offs should be updated
- Data for a second iteration
  - Use the **enhanced seismic** and **enhanced similarity**.
  - Optionally a new steering cube on the enhanced seismic can be calculated
- In a second iteration it is often better not to apply a median filter, this will smooth the result very much.
  - For the diffusion filter, use the fault enhanced seismic (iteration1) as output attribute
  - For the final output, experiment for finding a good cut-off value for applying the diffusion filter
  - If the cut-off for applying the diffusion filter is not met, output the fault enhanced seismic (iteration1)

Filtering Seismic Volumes: Parameters

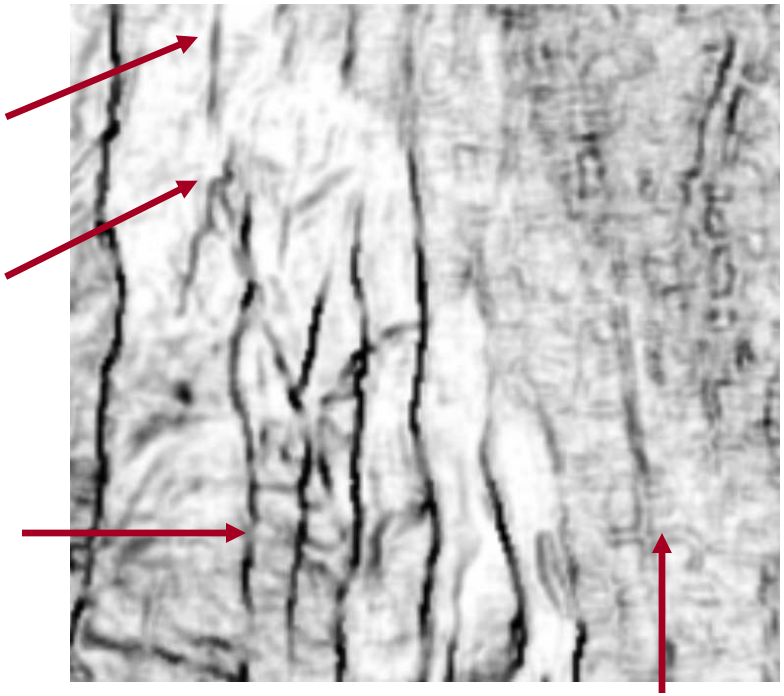
- How to set the parameters for similarity is treated in part 4 of this presentation
- In all filters use:
  - Time window: 0 to 0 ms
  - Step-out: 1
- The only real parameter to establish is the cut-off value to combine MDF and diffusion filter. Criteria
  - Similarity of the filtered volume should have suppressed “noisy” similarity due to random noise, acquisition footprint, etc.
  - Similarity of the filtered volume should have enhanced and sharpened faults

Effect of cut-off value

Cut-off=0.65 (more diffusion filter)



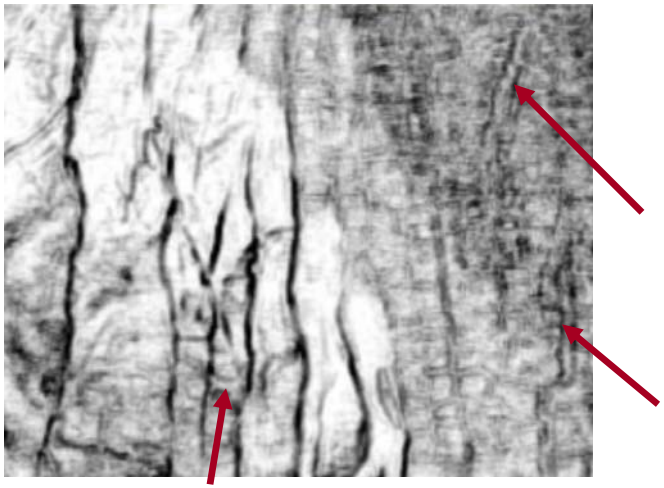
Cut-off=0.35 (more MDF filter)

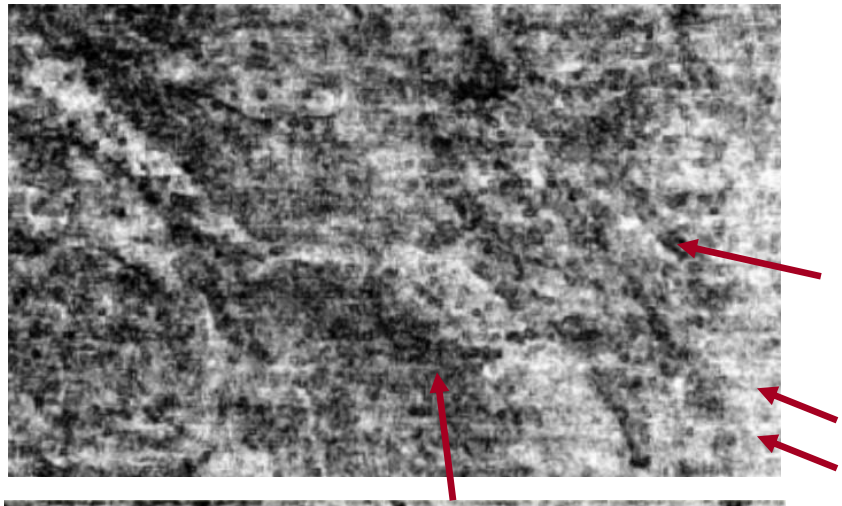


- Higher cut-off:
  - Weaker faults are imaged
  - More fault continuity
  - More no-fault noise (acquisition footprint, noise-spikes sharpened by diffusion, etc)

Result: Similarity before and after filter

Original





Filtered