

WuMapPy

WuMapPy Documentation

Release 0.32

Lionel Darras, Philippe Marty & Quentin Vitale

Aug 01, 2019

TABLE OF CONTENTS

1	About WuMapPy	5
1.1	Introduction	5
1.2	Features	5
1.3	Main authors	5
1.4	License	6
2	Installation	7
2.1	Using pip	7
2.2	Building from sources	7
2.3	Dependencies	7
2.4	MSI installer	8
2.5	Uninstallation	9
3	Getting started	11
3.1	Starting WuMapPy	11
3.2	Getting help	12
3.3	GUI Settings	12
3.4	Opening data files	15
3.5	Processing data	20
3.6	Georeferencing	20
4	GUI Overview	23
4.1	Main Window	23
4.2	Dataset Window	25
4.3	Geographic set Window	28
5	General Operations	31
5.1	Dataset Informations	31
5.2	Transform	31
5.3	Math	32
5.4	Clip	32
5.5	Digitize	33
6	General Processing	35
6.1	Peak filtering	35
6.2	Thresholding	35
6.3	Zero-Mean Traversing	36
6.4	Median filtering	36
6.5	Festoon filtering	36
6.6	Regional trend filtering	39

6.7	Wallis filtering	40
6.8	Ploughing filtering	41
6.9	Constant destriping	41
6.10	Curve destriping	42
7	Magnetic Processing	45
7.1	Logarithmic transformation	45
7.2	Pole reduction	46
7.3	Continuation	46
7.4	Analytic signal	47
7.5	Equivalent stratum magnetic susceptibility	49
7.6	Gradient <-> Total field Conversion	49
7.7	Euler deconvolution	50
8	Georeferencing	53
9	Feedback & Contribute	55
10	Citing WuMapPy	57
11	Changelog	59
11.1	Version 0.32	59
11.2	Version 0.31	59
11.3	Version 0.30	59
11.4	Version 0.21	60
11.5	Version 0.20	60

ABOUT WUMAPPY

1.1 Introduction

WuMapPy is a project initiated in 2014 through cooperation between two units of the CNRS¹ (UMR5133-Archeorient and UMR7619-Metis). Since 2017, it is also developed by Geo-Heritage (a cooperation between UMR5133-Archeorient and Eveha International).

WuMapPy is an open source Python package that offers tools for sub-surface geophysical survey data processing, in the field of archaeology, geology, and other sub-surface applications.

It mainly focuses on ground surveys data and offers tools to process the data and create geophysical maps that can be imported to GIS softwares.

WuMapPy contains *general tools*, such as data *destaggering*, *destriping* and *method-specific tools*, such as *reduction to the pole* or *magnetic data continuation*.

WuMapPy builds a geophysical `DataSet` object composed by series of data in the format (X,Y,Z) with (X,Y) being the point position of the geophysical value Z, in order to process and/or display as maps of Z values.

This package is a GUI for the processing package `GeophPy`.

1.2 Features

- Building dataset from one or several data files.
- Displaying geophysical maps in 2-D or 3-D.
- Processing datasets with general or method-specific geophysical filters.
- Export processed datasets into georeferenced format.
- Compatibility with Python 3.x.

1.3 Main authors

- **Lionel DARRAS**

CNRS, UMR5133-Archeorient, Lyon, France

lionel.darras@mom.fr

- **Philippe MARTY**

¹ French National Center for Scientific Research

CNRS, UMR7619-Metis, Paris, France

philippe.marty@upmc.fr

- **Quentin VITALE**

Eveha International, Lyon, France

quentin.vitale@eveha.fr

1.4 License

WuMapPy is developed on a [GNU GPL v3](#) license :

Copyright (c) 2014-2019 by Lionel Darras, Philippe Marty, Quentin Vitale See AUTHORS for more details.

Some rights reserved.

Redistribution and use in source and binary forms of the software as well as documentation, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- The names of the contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE AND DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

INSTALLATION

You can install **WuMapPy** as a Python package or a stand-alone version. In the former case, a Python (3.x) installation is necessary to install the package.

2.1 Using pip

You can install **WuMapPy** directly from the [PyPI](#) repository using `pip`.

First, make sure you have an up-to-date version of `pip` using the command:

```
>>> pip install --upgrade pip
>>> or
>>> python -m pip install --upgrade pip
```

Then, install, upgrade (or uninstall) **WuMapPy** directly from [PyPI](#) repository using `pip` with these commands:

```
>>> pip install wumappy
>>> pip install --upgrade wumappy
>>> pip uninstall wumappy
```

You can also download the zip file “WuMapPy-vx.y” from the [PyPI](#) repository, and from the download folder use:

```
>>> pip install WuMapPy-vx.y.zip
```

2.2 Building from sources

Download the zip file “WuMapPy-vx.y” and unzip it. Go to the unzipped folder and run the install script with the following command:

```
>>> python setup.py install
>>> or
>>> python -m setup.py install
```

2.3 Dependencies

WuMapPy is a GUI for the `GeophPy` package, it requires:

Name	Version	Comment
GeophPy	3.2 (or greater)	
Qt binding for Python		PySide, Qt4, Pyside2 or Qt5
QT.py		
matplotlib		
Sphinx	1.4.3 (or greater)	

Tip: Failure on **Windows**

WuMapPy uses others Python modules and packages that should be automatically installed. However, the installation of these modules may failed on **Windows** (in the absence of a C++ compiler for instance).

They can be installed independently using the useful website: <http://www.lfd.uci.edu/~gohlke/pythonlibs/>. This website provides many popular scientific Python packages precompiled for **Windows** distributions.

To install a package independently:

1. Download the precompiled package sources corresponding to your Python version and **Windows** distribution (SomePackage-vx.y-cp3x-cp3xm_winxx.whl);

NumPy, a fundamental package needed for scientific computing with Python.

Numpy+MKL is linked to the [Intel® Math Kernel Library](#) and includes required DLLs in the numpy.core directory.

[numpy-1.14.6+mkl-cp27-cp27m-win32.whl](#)

[numpy-1.14.6+mkl-cp27-cp27m-win_amd64.whl](#)

[numpy-1.14.6+mkl-cp34-cp34m-win32.whl](#)

[numpy-1.14.6+mkl-cp34-cp34m-win_amd64.whl](#)

[numpy-1.14.6+mkl-cp35-cp35m-win32.whl](#)

[numpy-1.14.6+mkl-cp35-cp35m-win_amd64.whl](#)

[numov-1.14.6+mkl-cp36-cp36m-win32.whl](#)

2. In download folder, use a command prompt and install the package using `pip` with the name of the downloaded archive:

```
>>> python setup.py install SomePackage-vx.y-cp3x-cp3xm_winxx.whl
>>> or
>>> python -m setup.py install SomePackage-vx.y-cp3x-cp3xm_winxx.whl
```


3. Repeat the process for all packages which installation failed before re-installing **WuMapPy**.
-

2.4 MSI installer

You can install **WuMapPy** as a stand-alone software (no need of a Python distribution) using the msi installer.

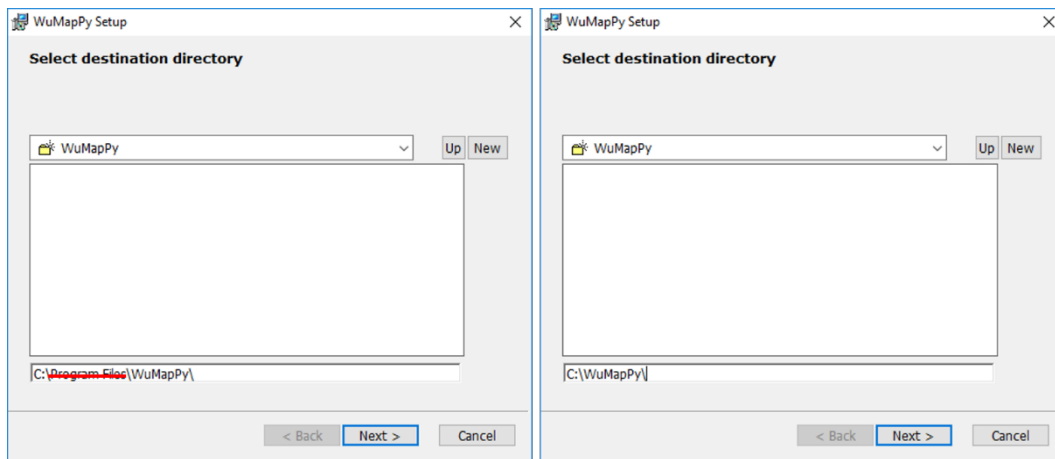
1. Download the “WuMapPy-vx.y-winxx.msi” installer that matches your OS system and double-click on it;

Nom

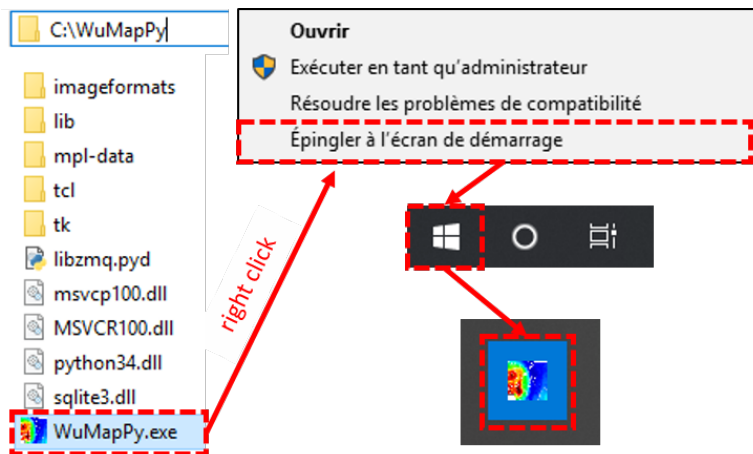
 WuMapPy-0.32-amd64.msi

2. Select the installation path

Warning: The installation path must contain **NO SPACE**. Especially not like the default **Windows Program Files**.



3. Add a **WuMapPy** shortcut to the **Windows** Start Menu by right-clicking on the WuMapPy.exe icon the installation folder and selecting *Pin shortcut to Start Menu*:



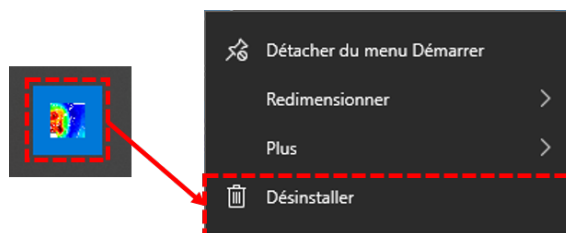
2.5 Uninstallation

The Python package can simply be uninstalled using pip:

```
>>> pip uninstall wumappy
>>> or
>>> python -m pip uninstall wumappy
```

Or, for the standalone version:

- by right-clicking on the **WuMapPy** shortcut and selecting uninstall



- via uninstall program provided by **Windows** (*Control Panel/Programs/Uninstall a program*).


Applications et fonctionnalités

[Gérer les fonctionnalités facultatives](#)

[Gérer les alias d'exécution d'application](#)

Effectuez des opérations de recherche, de tri et de filtrage par lecteur. Si vous voulez désinstaller ou déplacer une application, sélectionnez-la dans la liste.

Trier par : Nom ▾ Filtrer par : Tous les lecteurs ▾

	WuMapPy 0.32.0	850 Mo 08/10/2018
<div>Modifier</div>		<div>Désinstaller</div>

GETTING STARTED

3.1 Starting WuMapPy

The **WuMapPy** GUI is developed using a Qt binding for Python. It takes advantages of Qt.py so the software should be compatible with any of the available Qt binding for Python (PyQt4 or PySide and PyQt5 or PySide2), eventhough it has only be test with the PySide bindings so far.

To launch the software and display **WuMapPy**'s *Main Window*:

- for Python installation type wumappy as a command in any console:

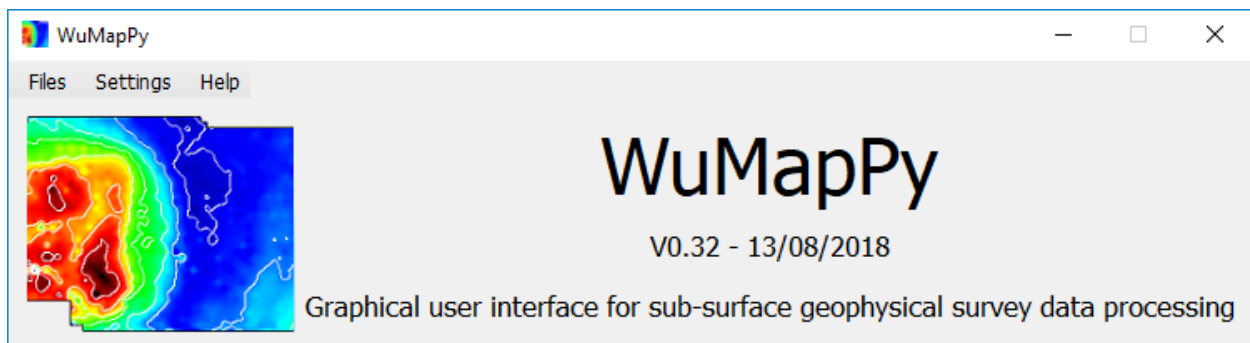
```
>>> wumappy
>>> WuMapPy
>>> python wumappy
>>> python -m wumappy
```

Note: Windows PATH System variable

On **Windows**, if typing wumappy in a command prompt does not bring up the program:

- check that the **PATH System variable** contains the Python scripts path `'..PythonXX\Scripts'`, (where XX depends on your Python version);
 - Add the path to the PATH variable if necessary;
 - then restart your command prompt and type wumappy.
-

- for .msi installation double-click on the **WuMapPy** shortcut (🖼️).

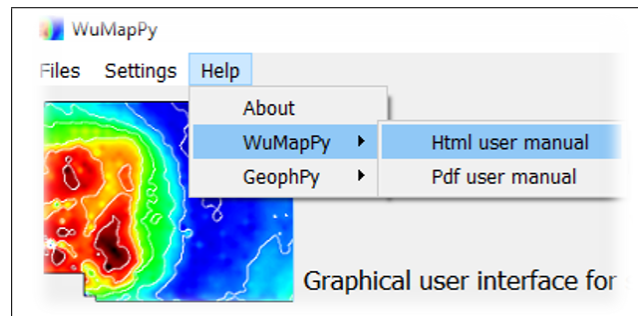


3.2 Getting help

3.2.1 Documentation

You can find access **WuMapPy** and **GeophPy**'s documentations in two formats (html or pdf) from **WuMapPy**'s *Main Window* help menu.

In this menu, you have also access to the **WuMapPy** and **GeophPy** versions number.



On **Linux** operating system, to opening these documentations with the best application, you need, before starting **WuMapPy** application, to write the applications full names to use with html and pdf documents in the “config” file:

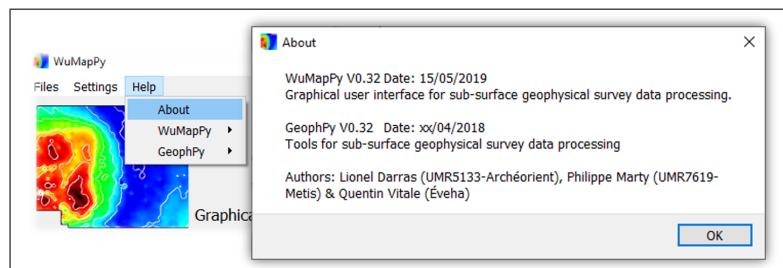
```
>>> [MISC]
      html_viewer = none
      pdf_viewer = none
```

with ‘none’, the default string, the application launches only the file full name (“.../WuMapPy.pdf” for example), and the operating system define which application to execute to opening the documentation file.

Note: The “config” file is saved in “~/wumappy” on **Windows**, or “~/wumappy” on **Linux** and **Mac OS**, where ‘~’ is the user directory.

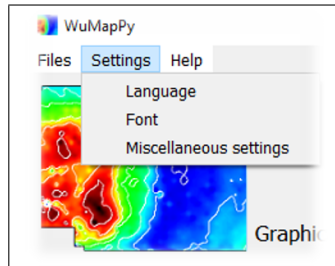
3.2.2 Package version

You can access **WuMapPy**'s version by selecting about in the the main window's help menu.



3.3 GUI Settings

You can change the entire GUI *Language*, *Font* and others *Miscellaneous settings* from **WuMapPy**'s *Settings* menu.



3.3.1 Language

The language descriptions are presents as *.lng* dictionary files in the `~/wumappy/resources` directory where `~` is the user home directory. The default builtin language for the Graphic User Interface is English, and it's the only one generated by the code and saved as the *english.lng*.

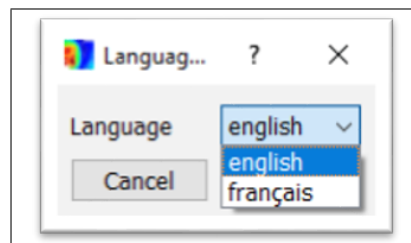
This file as the follwing format:

```
>>> english
FILES_ID      Files
SETTINGS_ID   Settings
FONT_ID       Font
MISCSETTINGS_ID Miscellaneous Settings
HELP_ID       Help
ABOUT_ID     About
```

The first line is the language name, the first column is object identifier (menu name, group box name etc.) and the second the displayed translation.

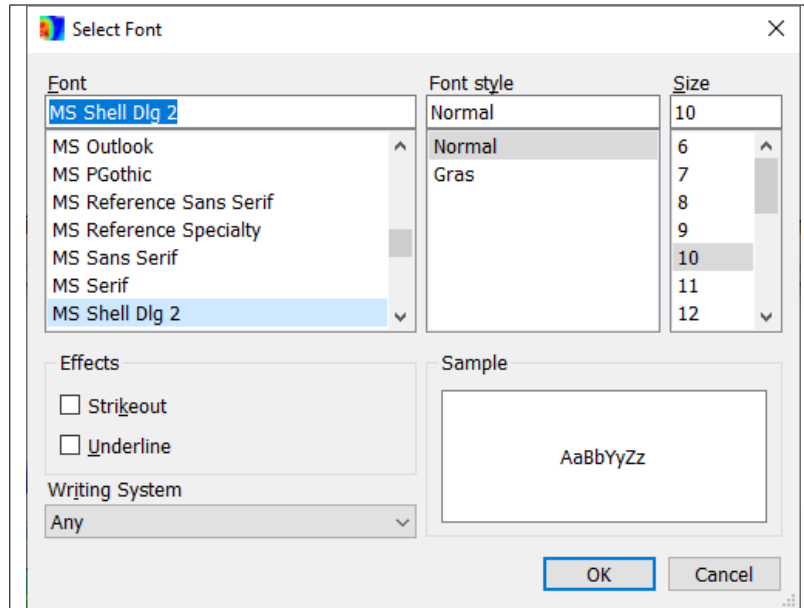
You can easily add and use an other language file: duplicate the *english.lng* file as *otherlanguage.lng*, change the language name in the first row and modify the second row with the correct translation in the new language.

The language files in the `~/wumappy/resources` directory (like *french.lng* or *spanish.lng* etc.) are automatically detected by **WuMapPy** and become available languages in the *Settings* → *Language* menu.

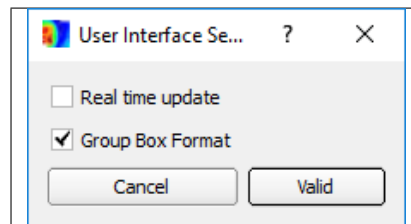


3.3.2 Font

The font type and the font size used in the majority of the windows and dialog boxes can be modified in the *Settings* → *Font* menu.



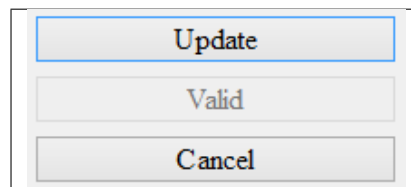
3.3.3 Miscellaneous settings



With this dialog box, it is possible to

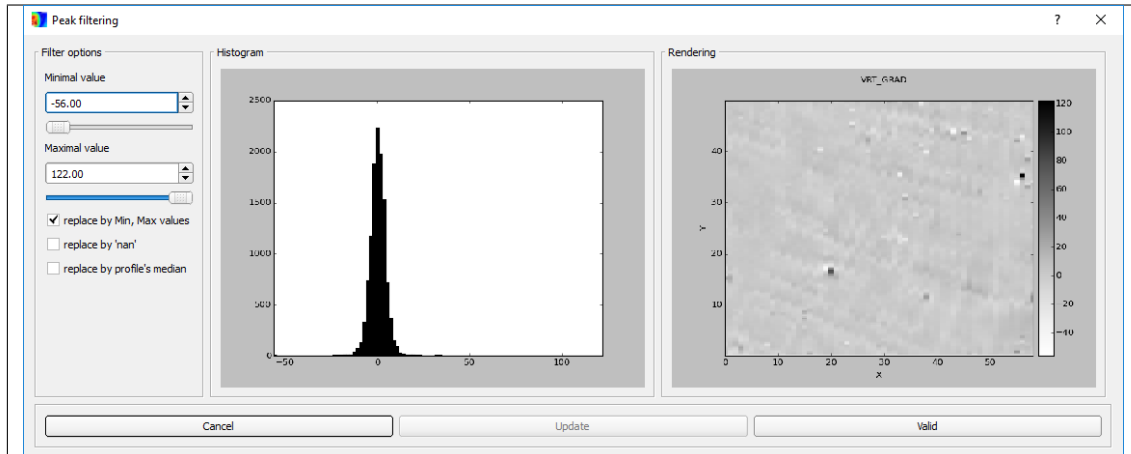
- Check/uncheck the flag of *Real time update* after a modification in a dialog box.

If the flag is unchecked, an *Update* button will be displayed in the dialog box and the map *Rendering* tab will be updated only if this *Update* button is clicked.

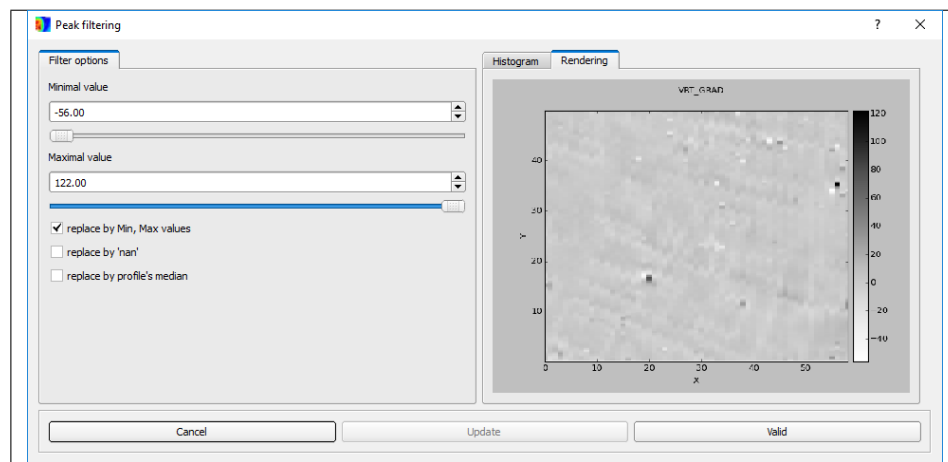


- Check/uncheck the flag of *Group Box Format* display style.

If the flag is unchecked all dialog boxes' tabs will be displayed in the same windows.



If the flag is checked, options will be displayed in different tabs.



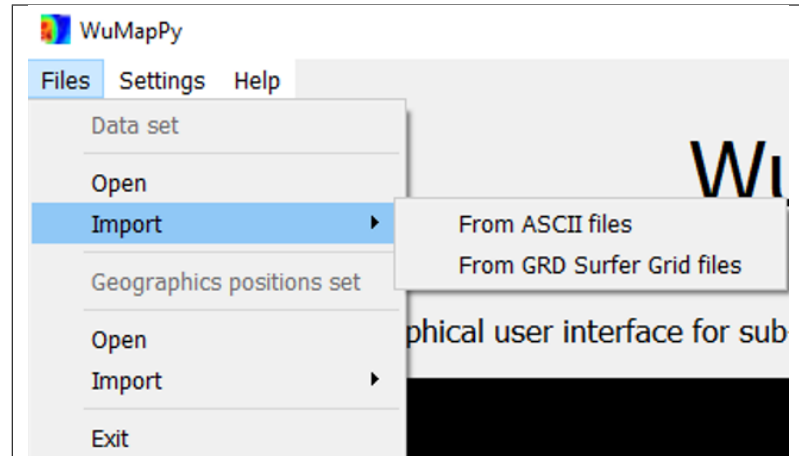
Note: Group Box Format

If the *Group Box Format* is disabled, the dialog boxes with many different tabs may be hard to use.

3.4 Opening data files

For now, **WuMapPy** reads three different types of data file formats:

- *netCDF files* (used as **WuMapPy**'s internal file system) ;
- *delimiter-separated ASCII files*;
- *Surfer Grid files*.



3.4.1 netCDF files

WuMapPy natively uses a **netCD** style format to store the processed dataset. You can re-open a previously processed data using the *open menu* in the main window *Files* menu.

Note: **WuMapPy's netCD format**

... To be explicated ...

3.4.2 Surfer grid

WuMapPy manages **Golden Software Surfer** grid files (**Surfer 7** binary grids, **Surfer 6** binary grids and **Surfer 6** ASCII grids).

Use the *Files* → *Import* → *From GRD Surfer Grid files* menu to import a grid. The grid type is automatically determined from the *.grd* file itself.

Note:

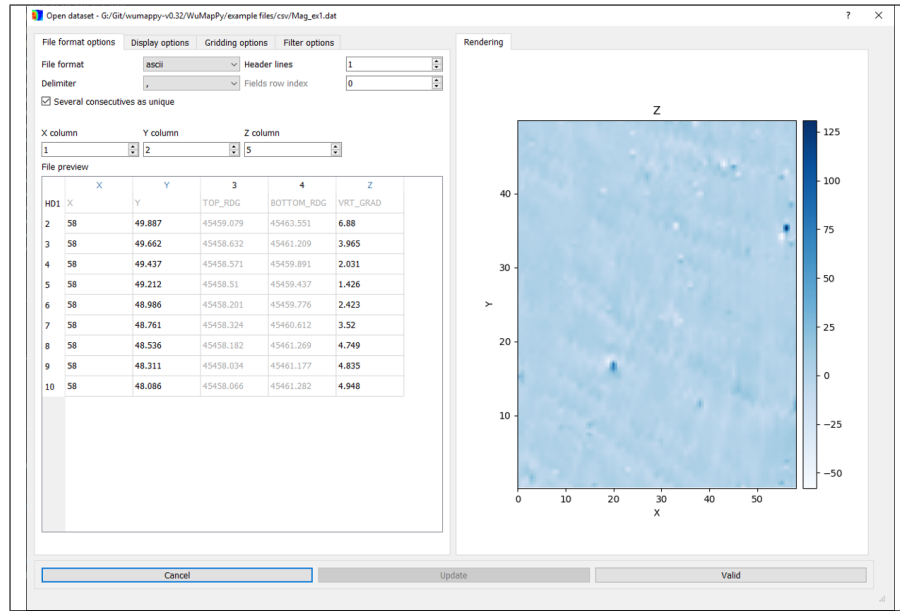
From Surfer's website: "Surfer Grid files contain grid nodes or grid cells at x,y locations, and these nodes/cells have specific z values associated with them. The grid files Surfer supports have evenly-spaced grid nodes/cells, though there are grid files that have uneven spacing. The x,y limits in map units, the grid spacing, and null values are all defined in the grid file."

3.4.3 Ascii files

You can import, display and pre-process data in comma-separated values files (CSV), or any other delimiter-separated values files format, from the *Files* → *Import* → *From ASCII files* menu.

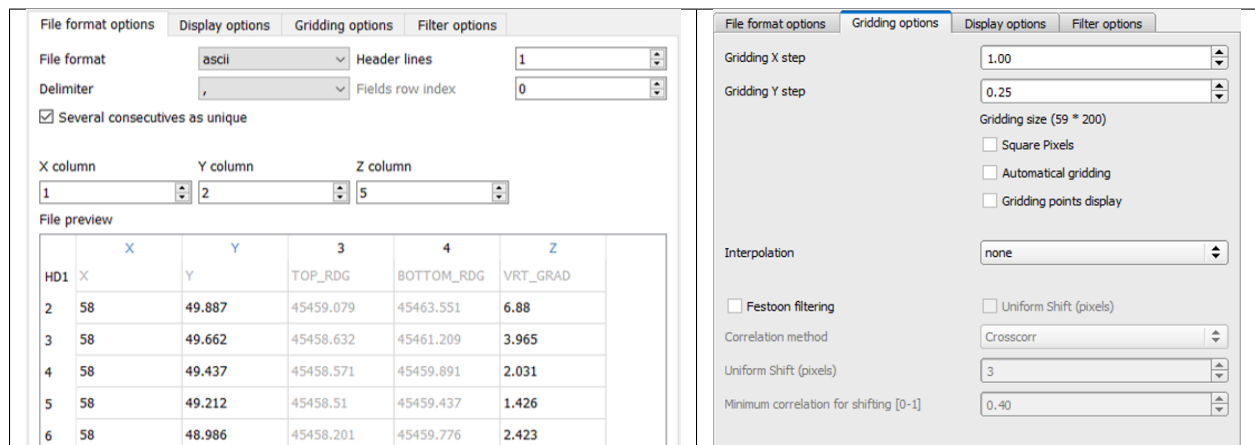
A file preview allows you to select:

- the file delimiter;
- the columns for X, Y and Z;
- and the number of header lines to ignore.

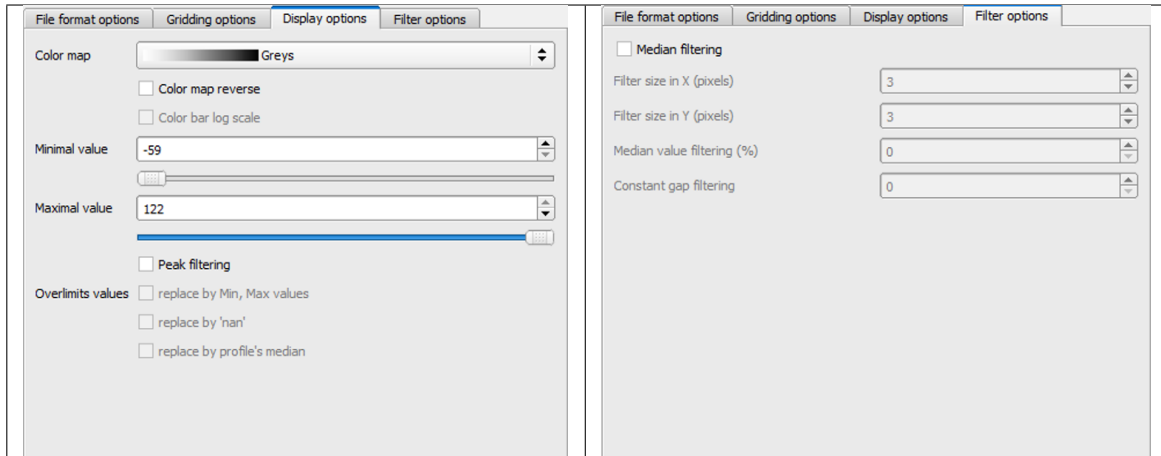


The *Update* button allows you to preview the effect of the filter before an actual validation when the *real time update* option is switched off.

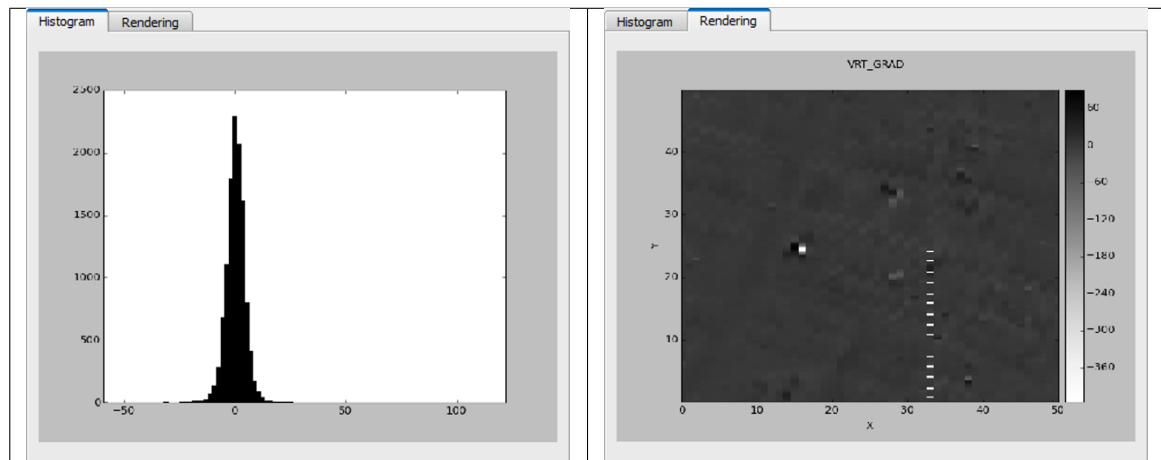
- File format, Gridding options and Festoon filter tabs.



- Display options and Median filter options tabs.

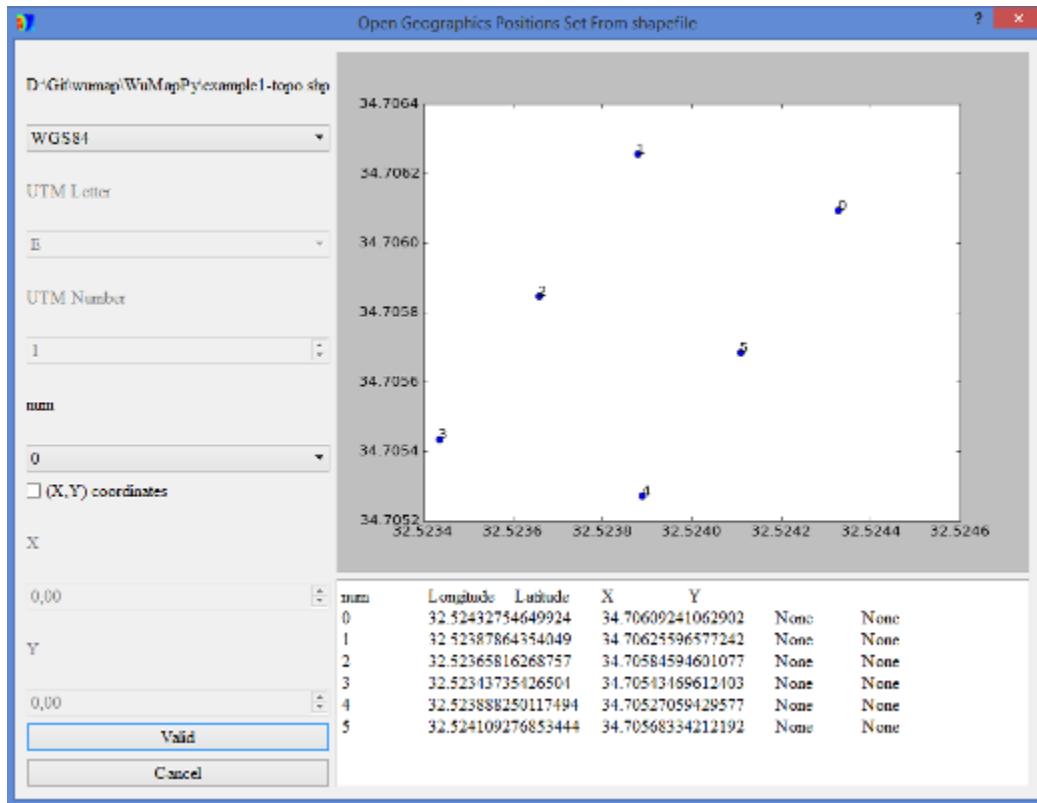


- Histogram and dataset Display tab.



3.4.4 Open Geoposition Set

WuMapPy uses Geoposition Set for data georeferencing. You can open and edit them in the GUI. Geoposition Sets are simply a set of Ground Control Points (GCPs) associated with a dataset. They can be imported from an ascii (.csv) file or a shapefile (.shp).



(for more informations, cf. **GeophPy** documentation)

Note: Geoposition Set ascii format

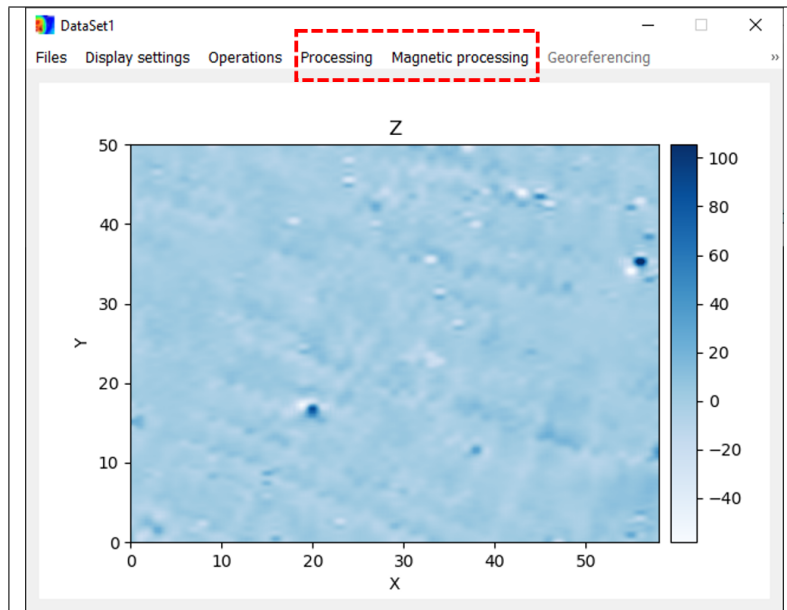
A simple semicolon-separated values with following informations, see the example in the `example files` directory.

```
>>> # Example with GCPs in GPS and local coordinates
UTM
1;745038.191;4656005.727;150;0
2;745068.172;4656045.663;150;50
3;745028.43;4656076.057;100;50
4;744988.466;4656105.978;50;50
5;744998.428;4656036.093;100;0
```

```
>>> # Example with GCPs in GPS coordinates only
WGS84
1;66.84617533;37.74956917;;
2;66.84649517;37.7489535;;
3;66.8472475;37.74972867;;
4;66.84689417;37.7491385;;
5;66.84691867;37.7491025;;
6;66.84689083;37.74980083;;
8;66.84720583;37.74922933;;
12;66.84760417;37.74967583;;
13;66.8478755;37.74908767;;
14;66.84794333;37.7491175;;
15;66.84815417;37.74870267;;
```

3.5 Processing data

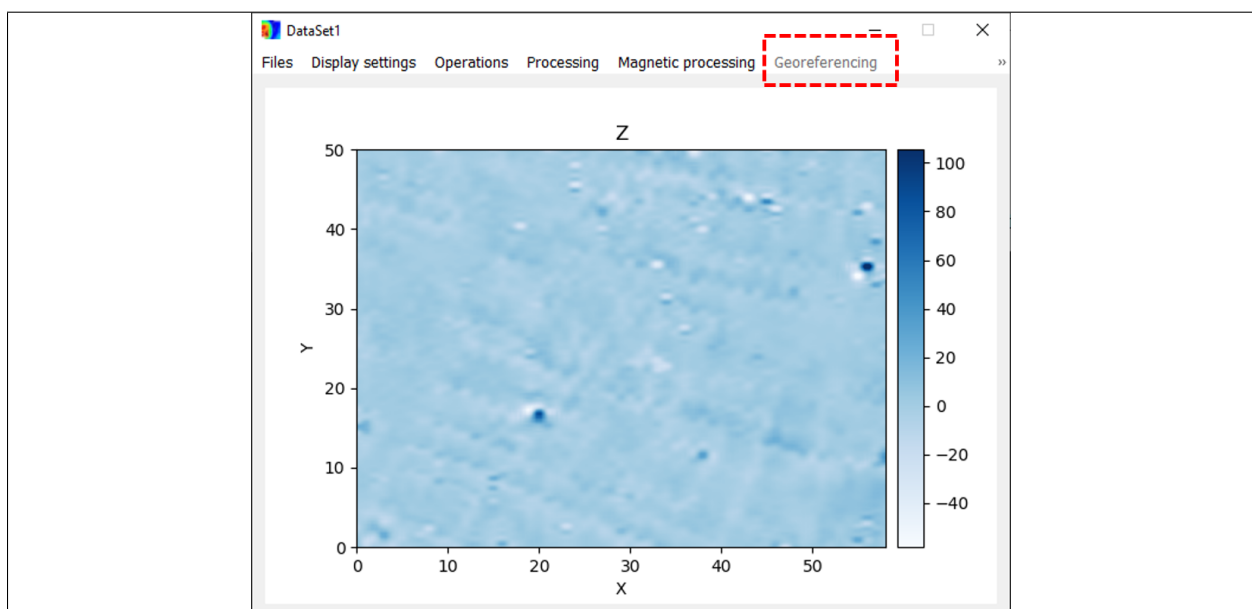
All the different available processing step can be applied to dataset from the *Dataset Window*.



See *General Processing*, *General Processing* and *Magnetic Processing* for a list of the different processing steps available.

3.6 Georeferencing

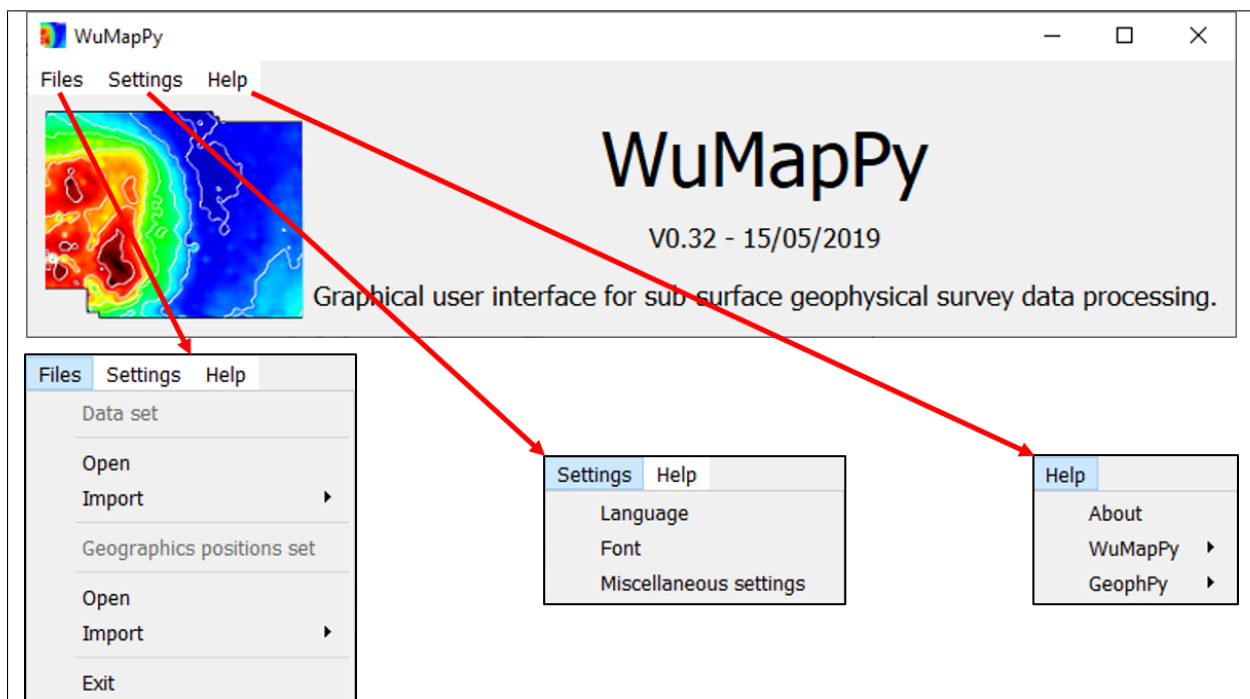
Georeferencing based on a Geoposition Set can be done from the *Dataset Window* if the corresponding Geoposition Set has been loaded.



See *Georeferencing* for more details.

GUI OVERVIEW

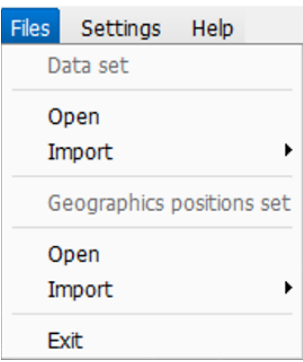
4.1 Main Window



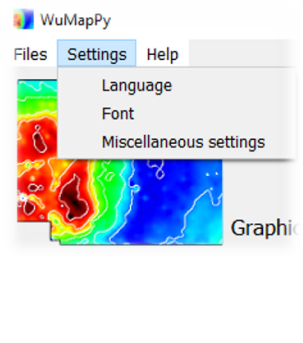
From the main window you can access the *Files*, *Settings* and *Help* menus. From there you can:

- Open processed data or *Import new data from ascii files*.
- Import or *Open Geoposition Set* files.
- Change the GUI *Language*, *Font* and others *Miscellaneous settings*.
- Access the GUI *Help and documentation*.

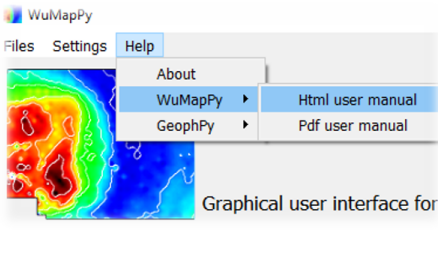
4.1.1 Files

<p>From the file menu you can:</p> <ul style="list-style-type: none"> • Open a Data Set file (.nc, NetCDF); • <i>Import data from (X,Y,Z) ascii files;</i> • Open a Geographic Positions Set (.netcdf); • Import a shapefile (.shp). 	
---	--

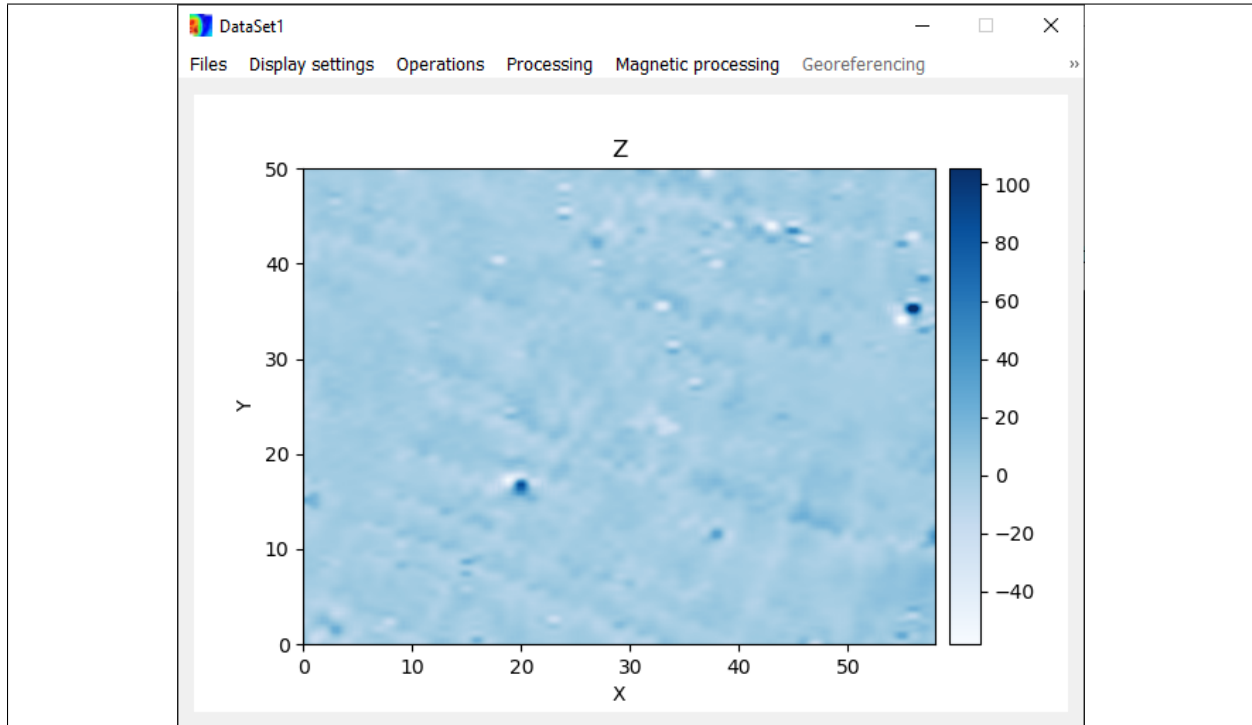
4.1.2 Settings

<p>From the file menu you can:</p> <ul style="list-style-type: none"> • Change the GUI Language; • Change the GUI Font; • Change others Miscellaneous Settings: <ul style="list-style-type: none"> – GUI auto update – GUI GroubBox/Tab layout 	
---	---

4.1.3 Help

<p>From the file menu you can:</p> <ul style="list-style-type: none"> • WuMapPy and GeophPy versions number; • WuMapPy and GeophPy documentations in html or pdf. 	
--	--

4.2 Dataset Window



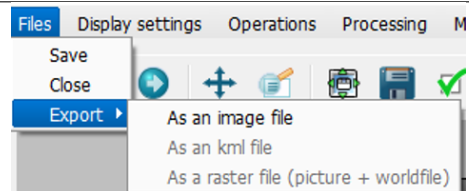
Once opened, a DataSet is displayed in a window with a menu bar that contains the different available options:

- *Files* (save and export the data)
- *Display Settings*
- *Processing* (general processings)
- *Magnetic Processing*
- *Georeferencing*
- *Miscellaneous settings*

4.2.1 Files

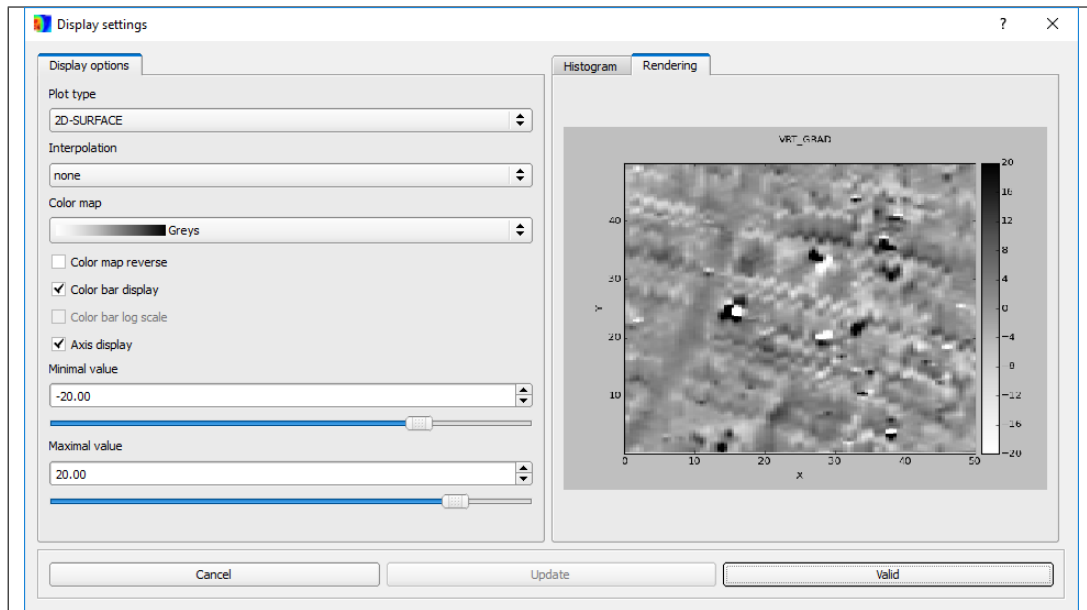
From the *Files* menu you can:

- Close the current dataset;
- Save the dataset in a netcdf file format ;
- Export the dataset in several formats.



4.2.2 Display Settings

With this menu, it's possible to changes the DataSet display options (colormap, axis, value limits, ...)

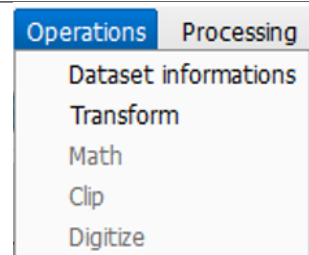


4.2.3 Operations

This menu gives access to all the *General Operations* on datasets available in WuMapPy:

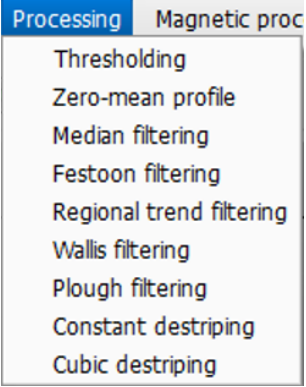
From the Operations menu you can:

- Get the dataset informations;
- Transform the data geometry;
- Apply basic math to the dataset;
- Clip and Digitize the dataset;



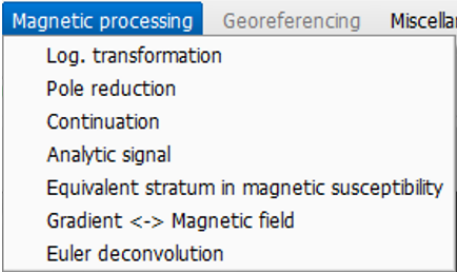
4.2.4 Processing

This menu gives access to all the *General Processing* available in WuMapPy.

<p>From the Processing menu you can:</p> <ul style="list-style-type: none"> • <i>Peak filtering;</i> • <i>Thresholding;</i> • <i>Zero-Mean Traversing;</i> • <i>Median filtering;</i> • <i>Festoon filtering;</i> • <i>Regional trend filtering;</i> • <i>Wallis filtering;</i> • <i>Ploughing filtering;</i> • <i>Constant destriping;</i> • <i>Curve destriping;</i> 	
---	--

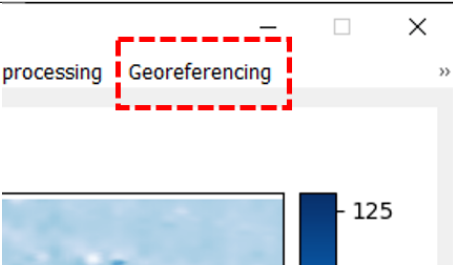
4.2.5 Magnetic Processing

This menu gives access to all the *Magnetic Processing* available in WuMapPy:

<p>From the Magnetic processing menu you can:</p> <ul style="list-style-type: none"> • <i>Logarithmic transformation;</i> • <i>Pole reduction;</i> • <i>Continuation;</i> • <i>Analytic signal;</i> • <i>Equivalent stratum magnetic susceptibility;</i> • <i>Gradient <-> Total field Conversion;</i> • <i>Euler deconvolution;</i> 	
--	---

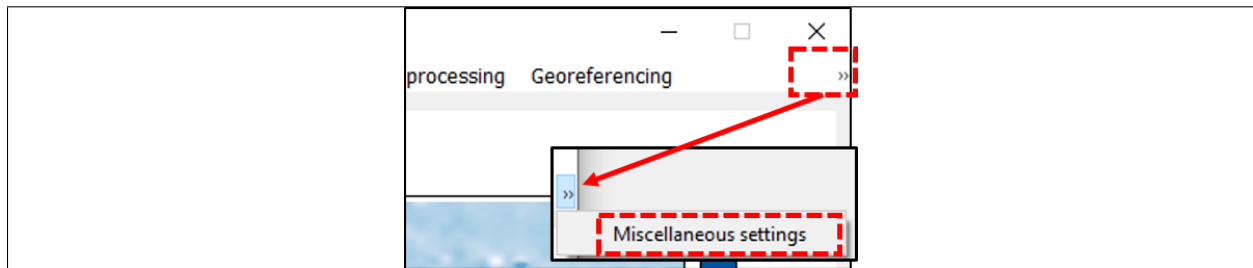
4.2.6 Georeferencing

Dataset georeferencing using a set of Ground Control Points (GCPs). Available only if a Geographic Positions Set is opened and displayed.

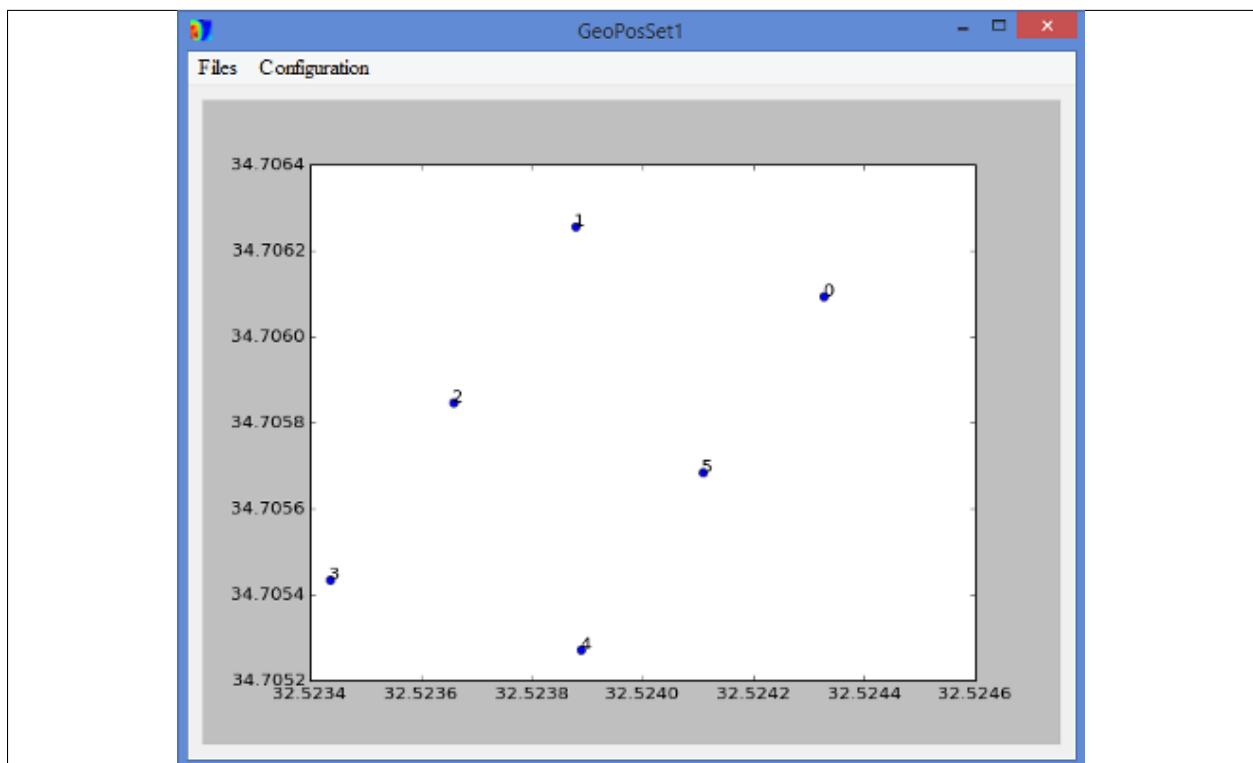
<p>From the georeferencing menu you can:</p> <ul style="list-style-type: none"> • georeference a dataset with GCPs. 	
---	--

4.2.7 Miscellaneous Settings

It is a simple duplicate of the Main window's *Miscellaneous settings*.

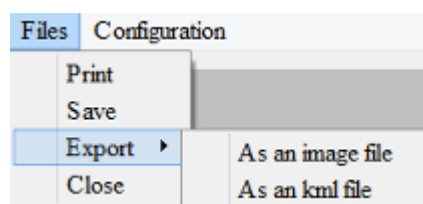


4.3 Geographic set Window

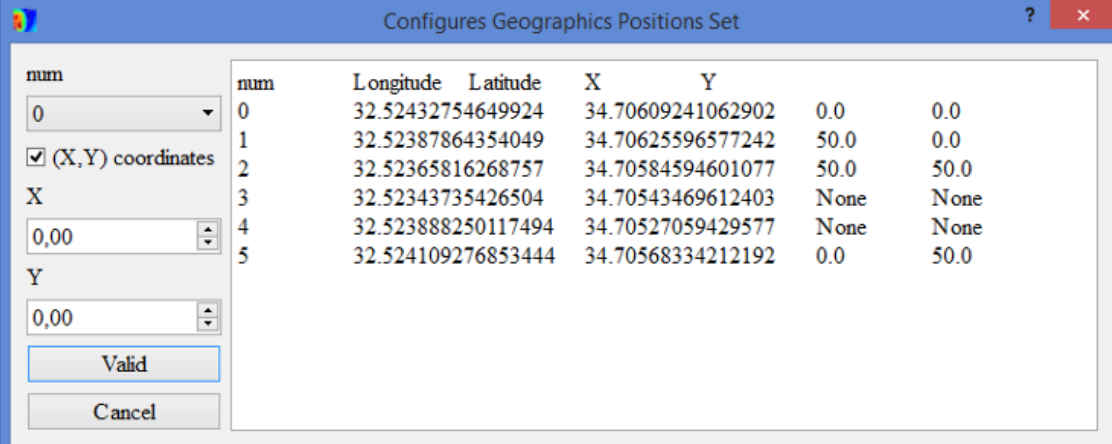


Once opened, a Geographic Positions Set is displayed in a window with a menubar that contains the different available options:

- **Files** To save and export the Geographic Positions Set



- **Configuration** To edit the Geographic Positions Set



num	Longitude	Latitude	X	Y
0	32.52432754649924	34.70609241062902	0.0	0.0
1	32.52387864354049	34.70625596577242	50.0	0.0
2	32.52365816268757	34.70584594601077	50.0	50.0
3	32.52343735426504	34.70543469612403	None	None
4	32.523888250117494	34.70527059429577	None	None
5	32.524109276853444	34.70568334212192	0.0	50.0

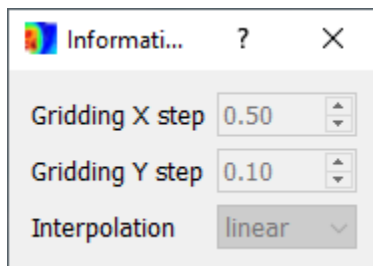
GENERAL OPERATIONS

General operations on a dataset.

Note: For more informations about this processing, see the GeophPy documentation.

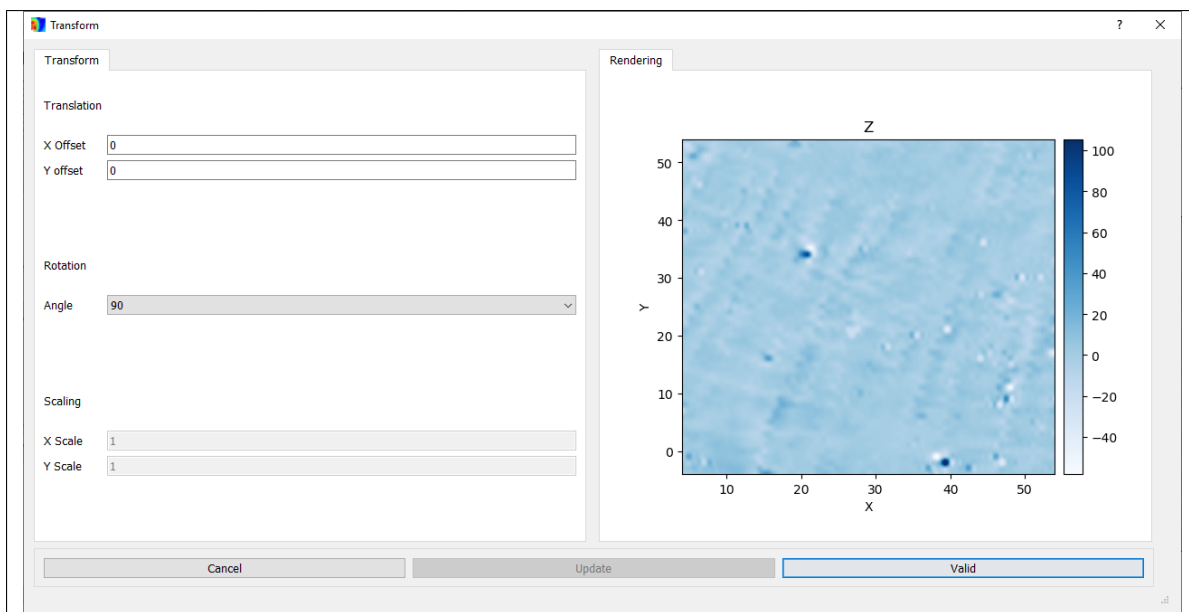
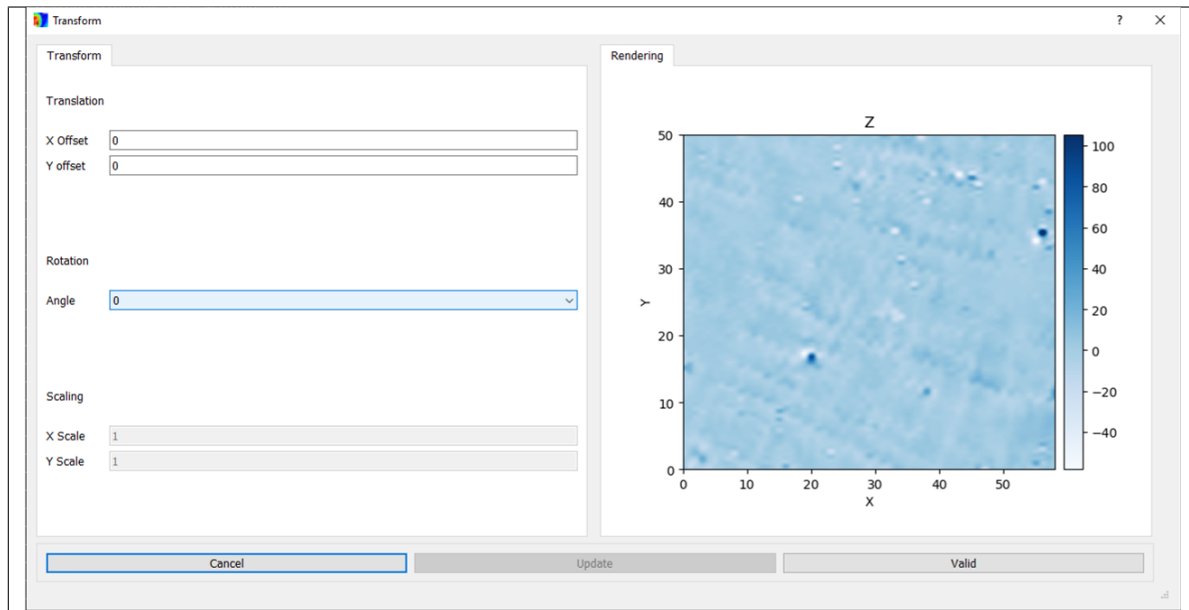
5.1 Dataset Informations

Show basic informations on the dataset.



5.2 Transform

Provides the basic geometrical operations on the dataset such as translation and rotation.



5.3 Math

Basic mathematical operation operations on the dataset such as addition or multiplication by a constant.

... Not Yet Available ...

5.4 Clip

... Not Yet Available ...

5.5 Digitize

... Not Yet Available ...

GENERAL PROCESSING

General processings, not bound to a specific survey method.

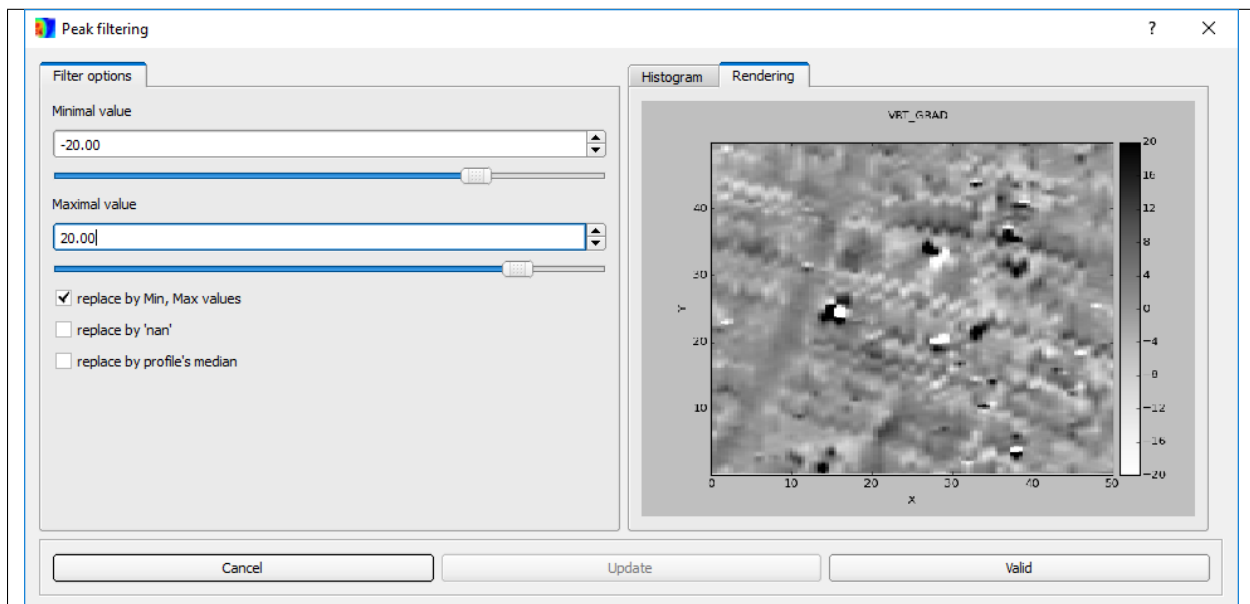
Note: For more informations about this processing, see the GeophPy documentation.

6.1 Peak filtering

... Not Yet Available ...

6.2 Thresholding

Data thresholding for values outside of the [*Minimal value*, *Maximal value*] interval.



Note: For more details, see the GeophPy package documentation.

6.3 Zero-Mean Traversing

Acquisition profile detrending by removing the mean or median.

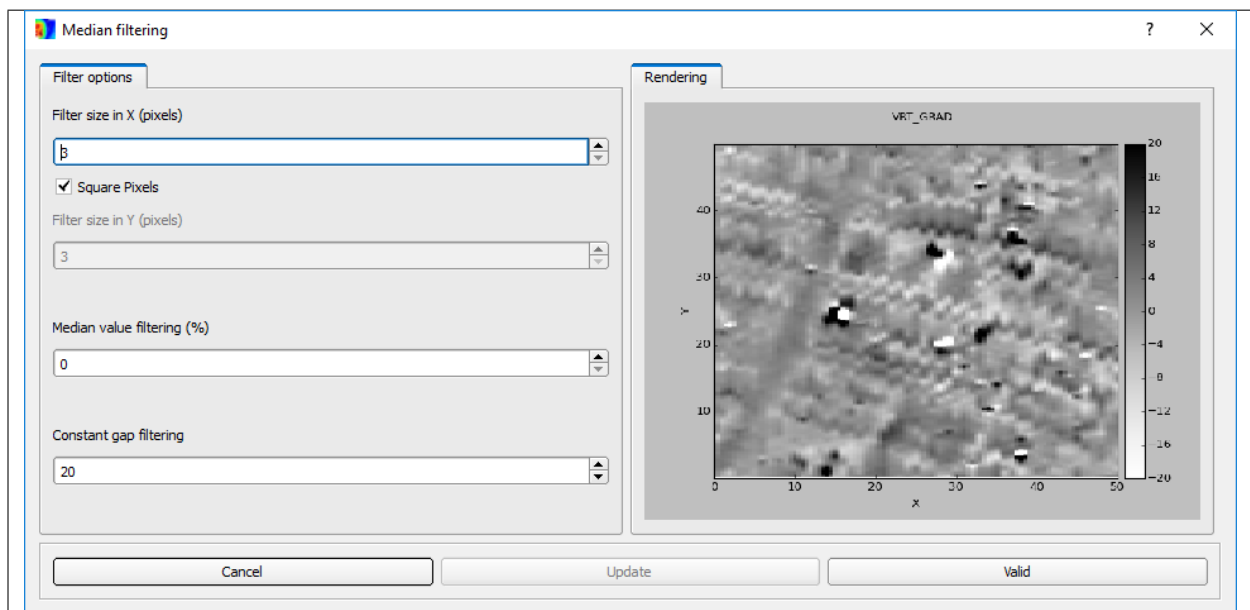
Subtracts the mean (or median) of each traverse (profile) in the dataset.

Note: This filter is strictly equivalent to the *constant destriping filter* in configuration ‘mono’ sensor, using ‘additive’ destriping method and $N_{prof}=0$:

... To Be Illustrated ...

6.4 Median filtering

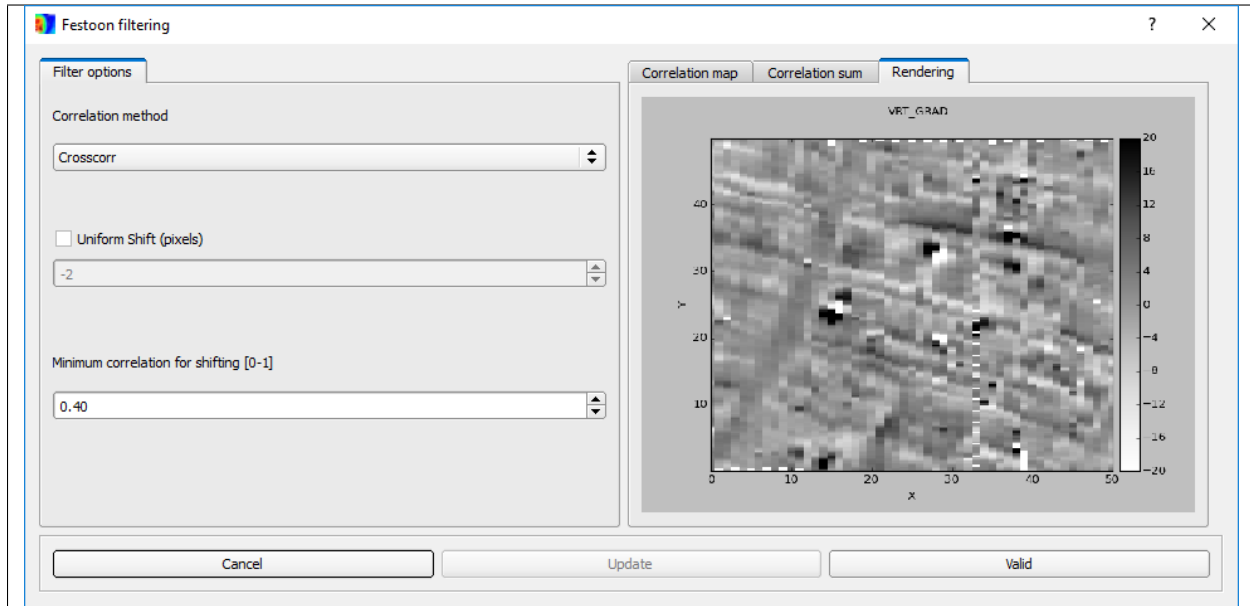
Classic median (salt-and-pepper) filter.



Note: For more details, see the GeophPy package documentation.

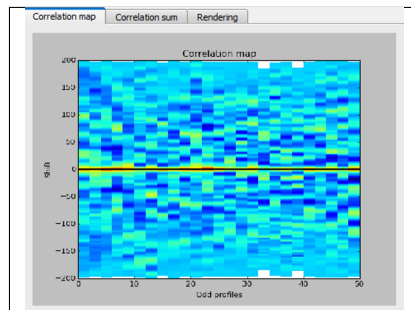
6.5 Festoon filtering

The festoon filter (destagger filter) reduces the positioning error along the survey profiles that result in a festoon-like effect.

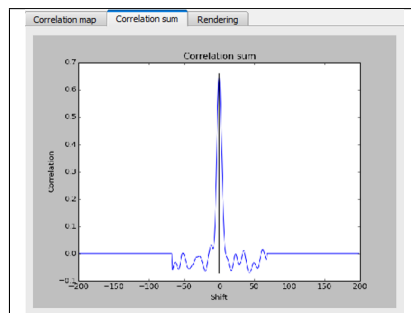


An optimum shift is estimated based on the correlation of a particular profile and the mean of its surrounding profiles. The filter's windows display 3 different tabs :

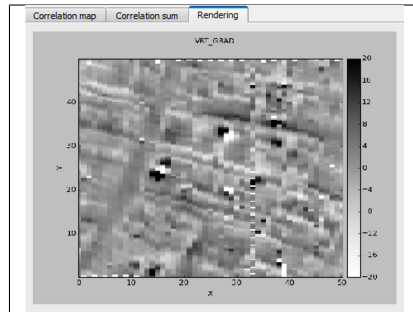
- **The correlation map,**



- **The correlation sum profile,**

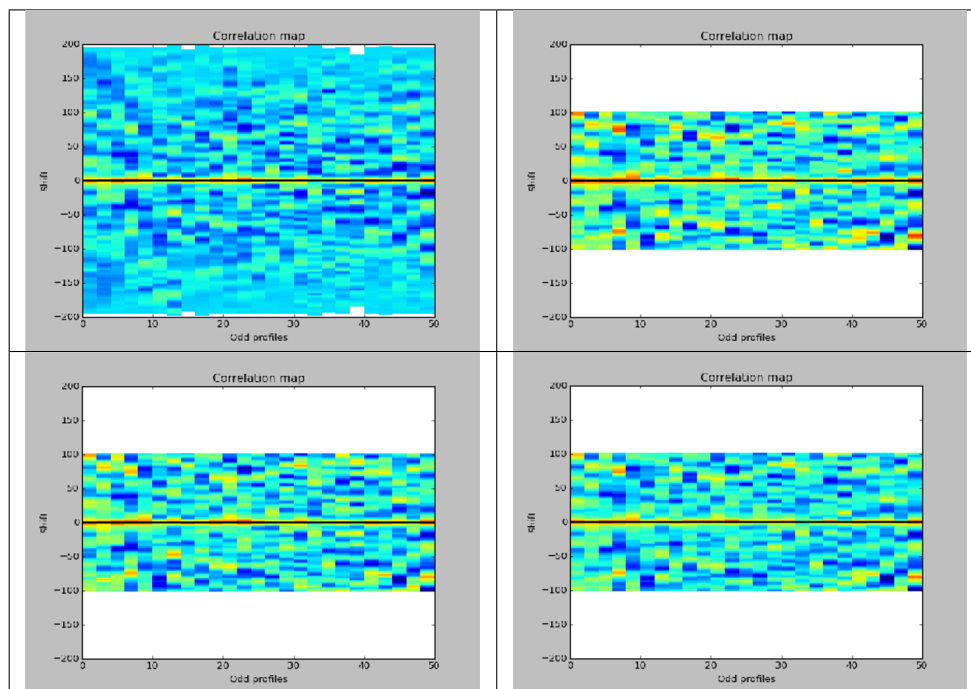


- **The filtered data.**



Different options are available:

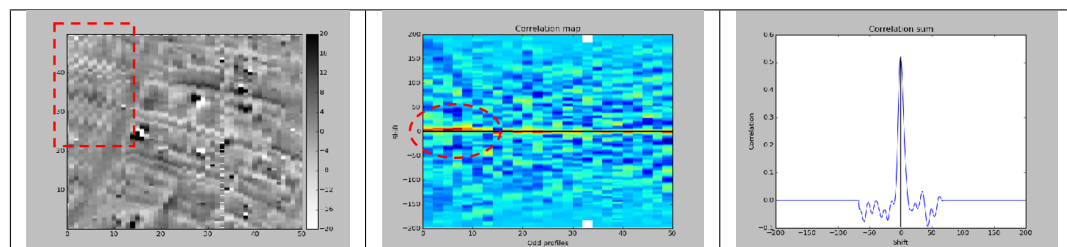
- **Method** for correlation calculation (Cross-correlation or Pearson and Spearman or Kendall correlation):



Due to the extensive computation time, Pearson, Spearman and Kendall correlation method are not computed over the whole shift domain.

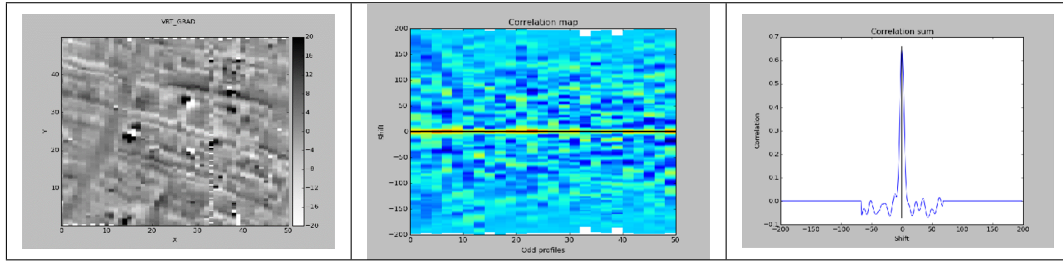
The usage of Cross-correlation is hence recommended.

- **Uniform** shift throughout the data:



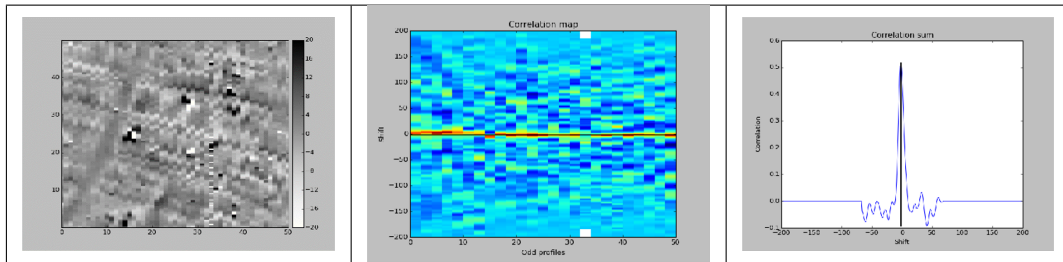
Return the best average shift for the dataset (based on the correlation sum off the dataset). Can be problematic when the position error is not regular over the dataset.

- **Non uniform** shift (different for each profile):



Return the best shift for each profile of the dataset (based on the correlation map).

- and required **minimum correlation** value:

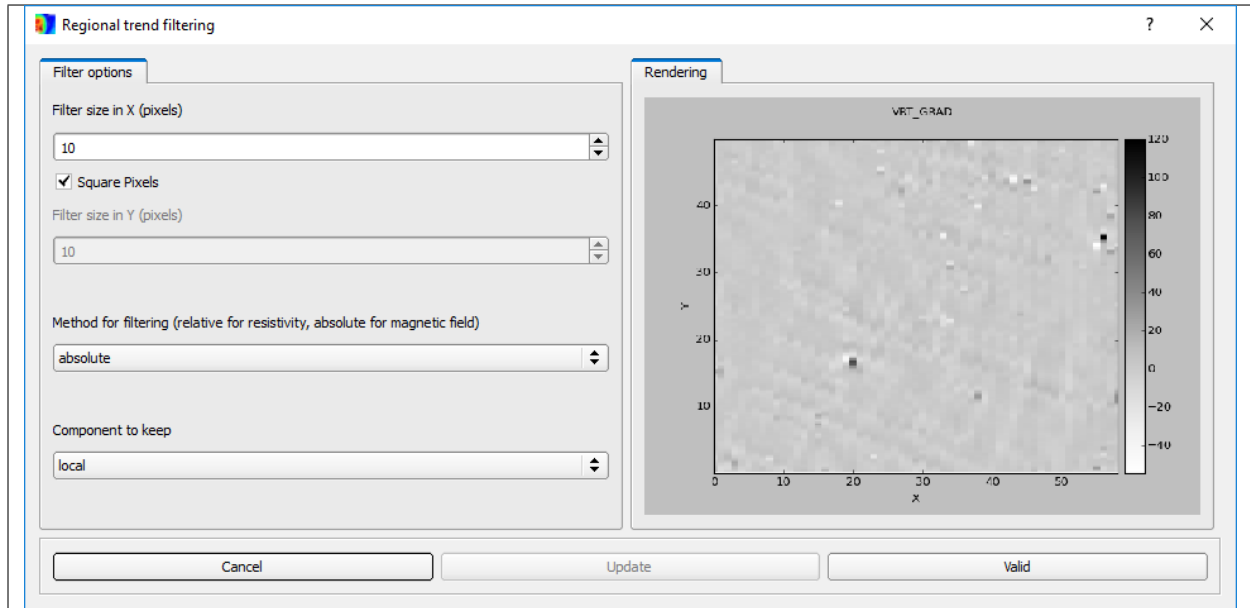


Prevents shifting profiles if correlation value is too low, here is an example for 1 (i.e. no shift allowed).

Note: For more details, see the GeophPy package documentation.

6.6 Regional trend filtering

Remove the background (or regional response) from a dataset to enhance the features of interest.

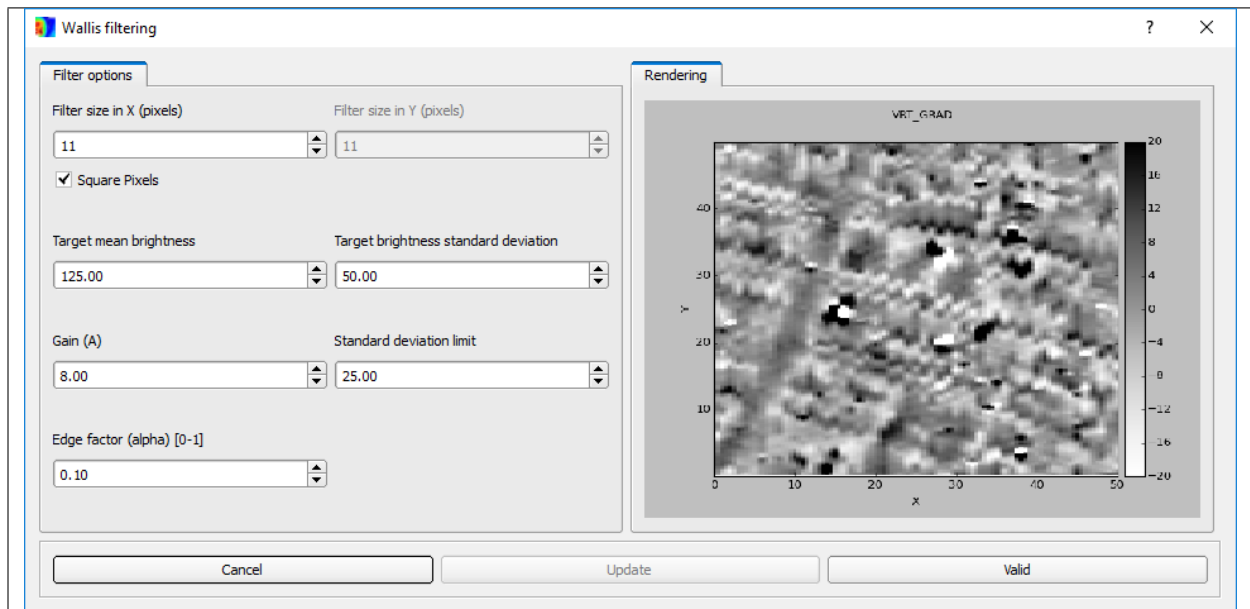


Note: For more details, see the GeophPy package documentation.

6.7 Wallis filtering

The Wallis filter is a locally adaptative contrast enhancement filter.

It is based on the local statistical properties of sub-window in the image. It adjusts brightness values (grayscale image) in the local window so that the local mean and standard deviation match target values.



Note: For more details, see the GeophPy package documentation.

6.8 Ploughing filtering

Directional filter.

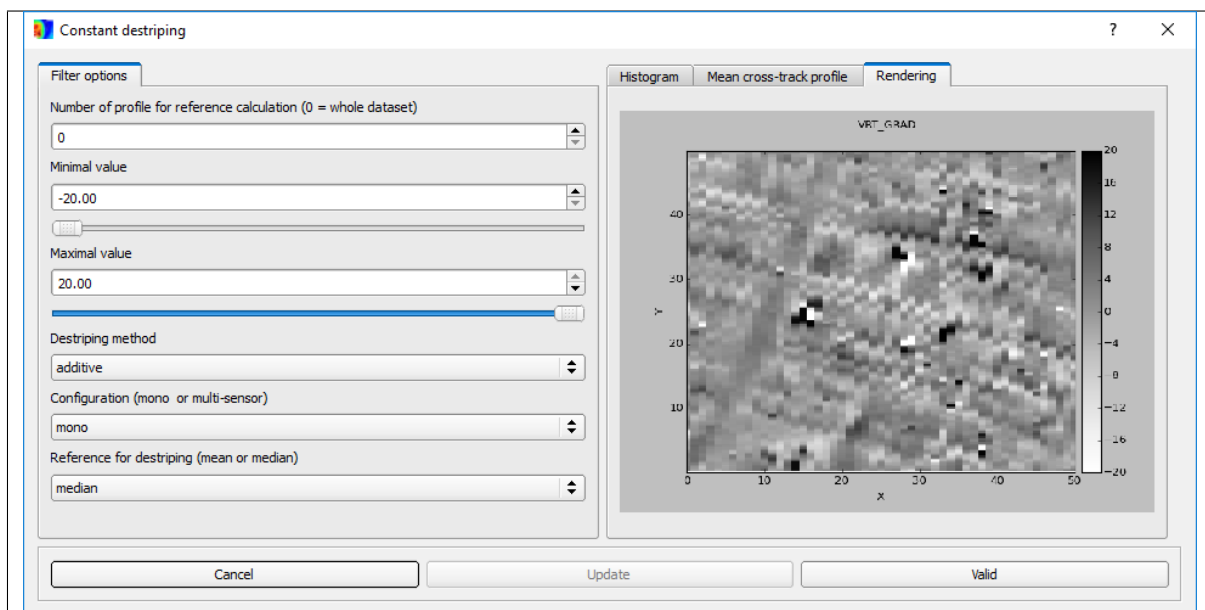
Apply a directional filter to reduce agricultural ploughing effect in the dataset (or any other directional feature).

... To Be Completed ...

6.9 Constant destriping

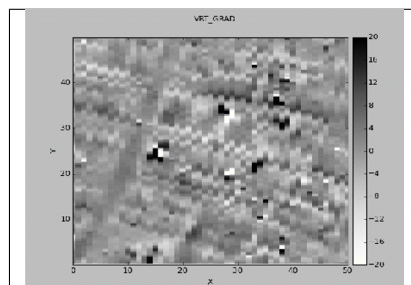
Acquisition profile detrending by removing a constant value.

Remove from the dataset the strip noise effect arising from profile-to-profile differences in sensor height, orientation, drift or sensitivity (multi-sensors array). Constant destriping is done using Moment Matching method.

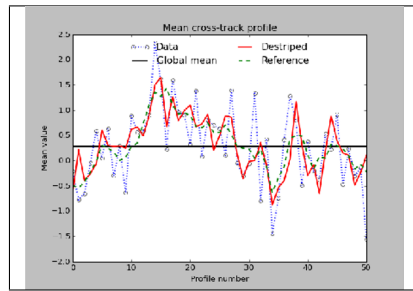


The filter's windows display 3 different tabs:

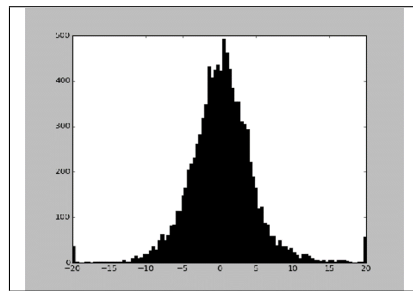
- **The filtered dataset**



- The mean cross-track profile



- The dataset histogram.

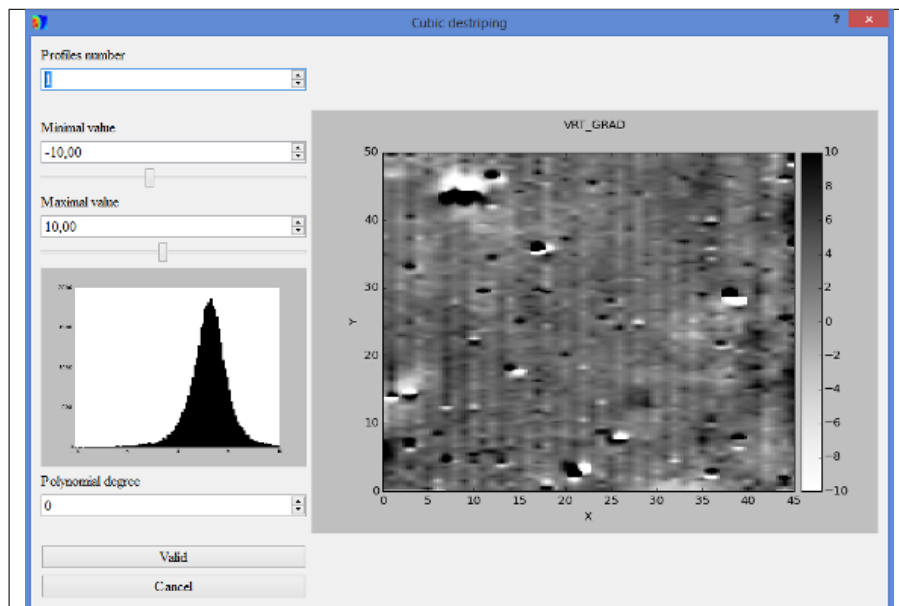


Note: For more details, see the GeophPy package documentation.

6.10 Curve destriping

Acquisition profile detrending by removing a polynomial fit.

Remove from the dataset the strip noise effect by fitting and subtracting a polynomial curve to each profile on the dataset.



Note: For more details, see the GeophPy package documentation.

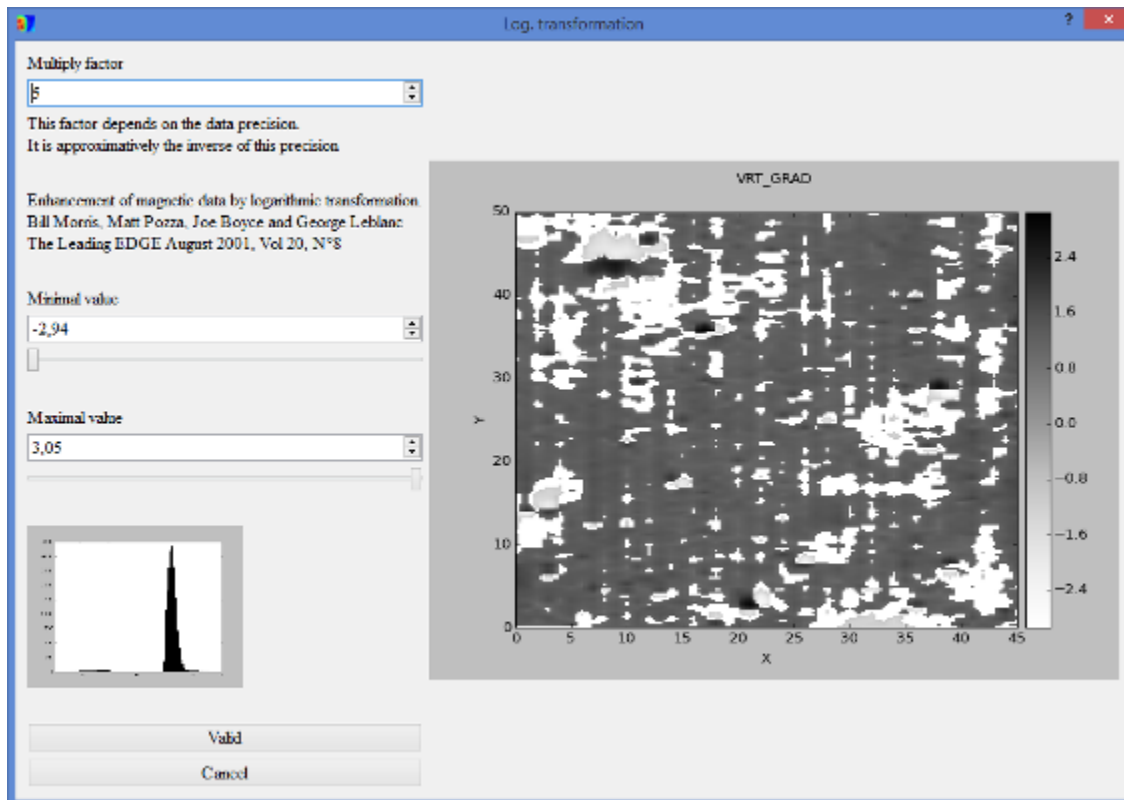
MAGNETIC PROCESSING

Processing specific to magnetic survey data.

7.1 Logarithmic transformation

The logarithmic transformation is contrast enhancement filter.

Originally used for geological magnetic data, it enhances information present in the data at low-amplitude values while preserving the relative amplitude information via logarithmic transformation procedure.

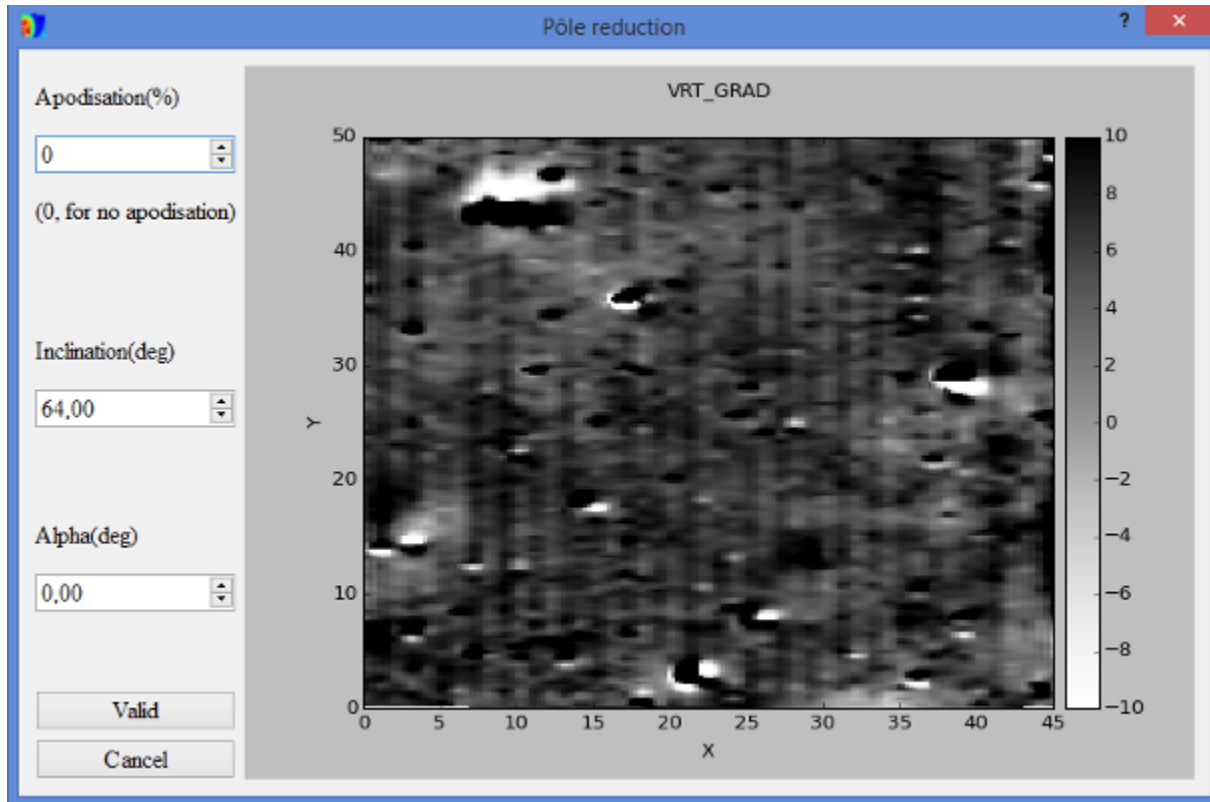


Note: For more informations about this processing, see the GeophPy documentation.

7.2 Pole reduction

Classic reduction to the pole.

The reduction to the magnetic pole is a way to facilitate magnetic data interpretation and comparison. It simulates the anomaly that would be measured at the north magnetic pole (inclination of the magnetic field is maximum, i.e. vertical).

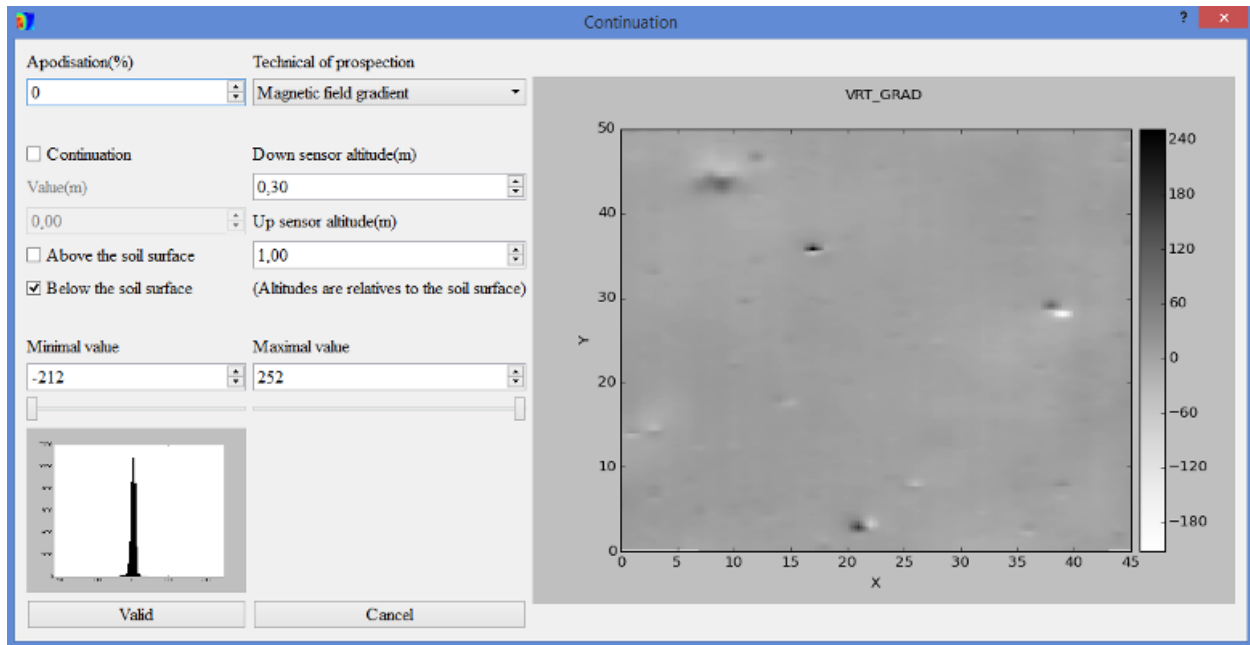


Note: For more informations about this processing, see the GeophPy documentation.

7.3 Continuation

Upward or downward continuation of potential field data (magnetic or gravimetric).

The filter computes the data that would be measured at an upper (*upward continuation*) or lower (*downward continuation*) survey altitude. The computation is done in the spectral (frequency) domain using the Fast Fourier Transform.

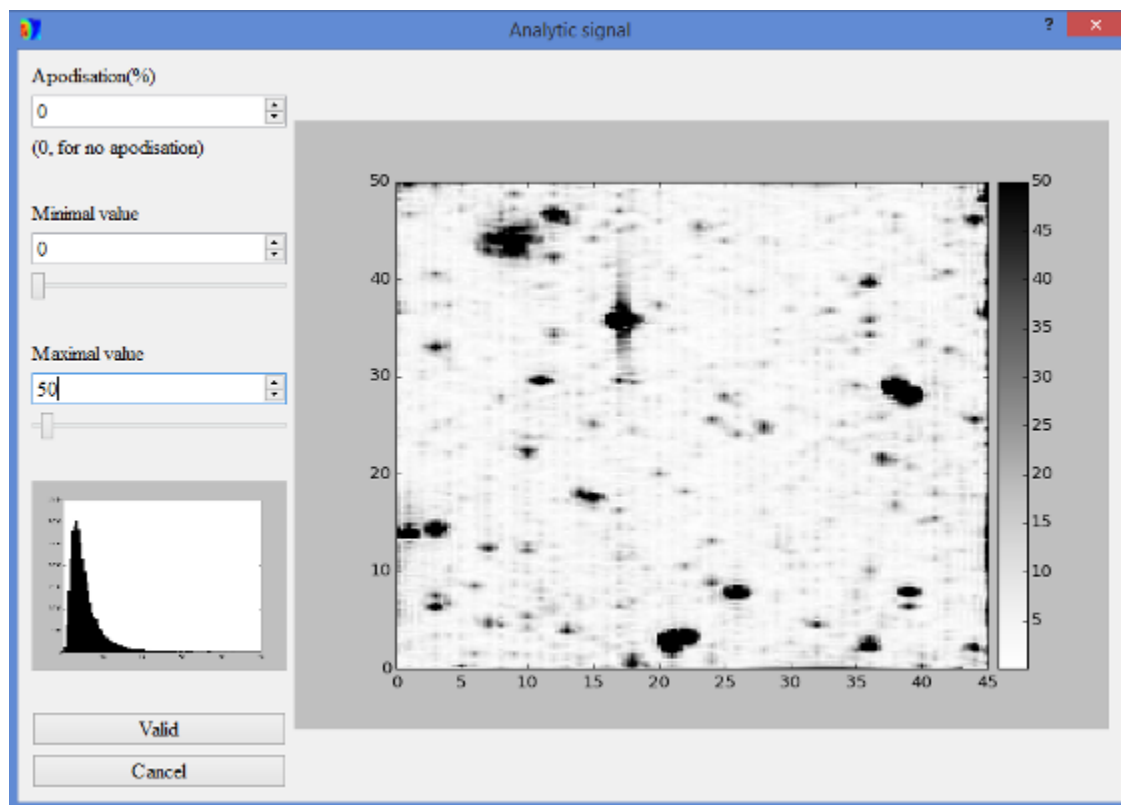


Note: For more informations about this processing, see the GeophPy documentation.

7.4 Analytic signal

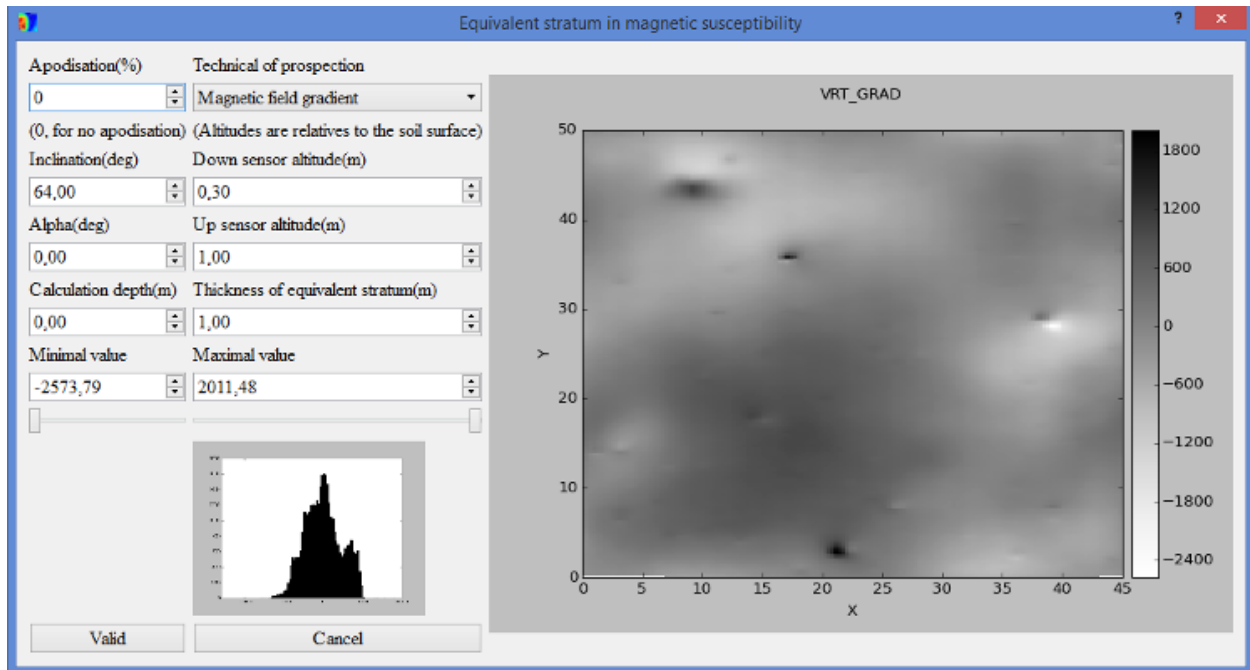
Computes the 3-D Analytic Signal.

The Analytic Signal (also known as the total gradient magnitude or energy envelope) is a way to ease magnetic source characterization independently from the direction of its magnetization.



Note: For more informations about this processing, see the GeophPy documentation.

7.5 Equivalent stratum magnetic susceptibility

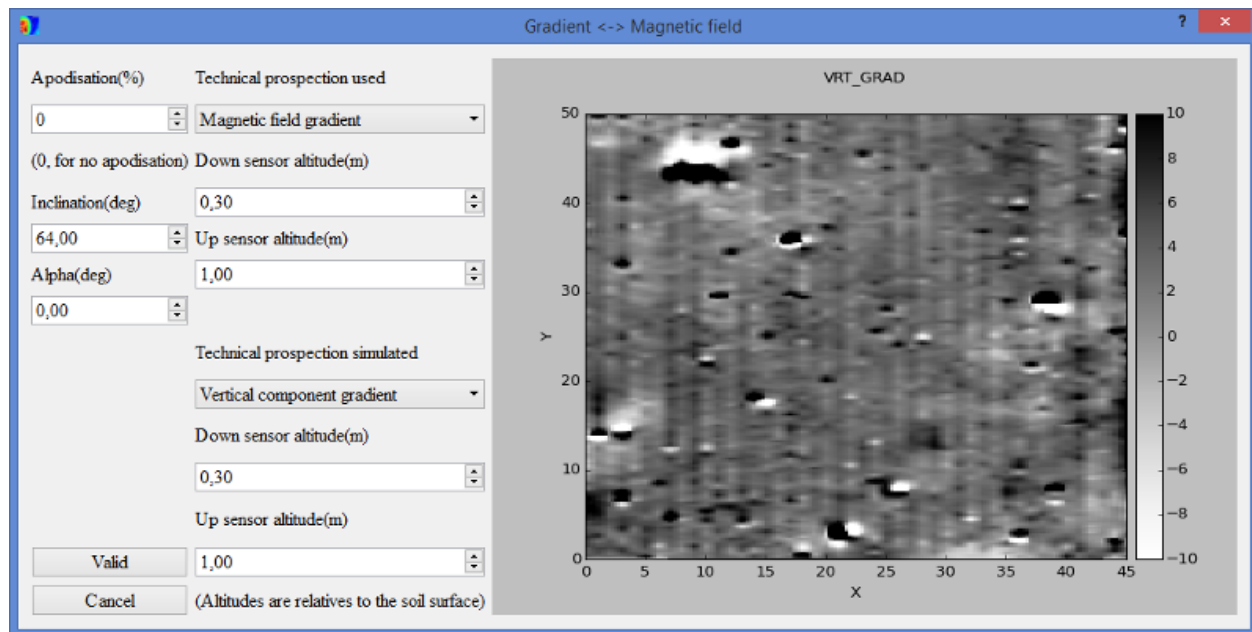


Note: For more informations about this processing, see the GeophPy documentation.

7.6 Gradient <-> Total field Conversion

Conversion between the different sensor's configurations.

Magnetic data are all derived from the same potential and thereby contains in the same information, making theoretical conversion from one sensor configuration to another is possible.

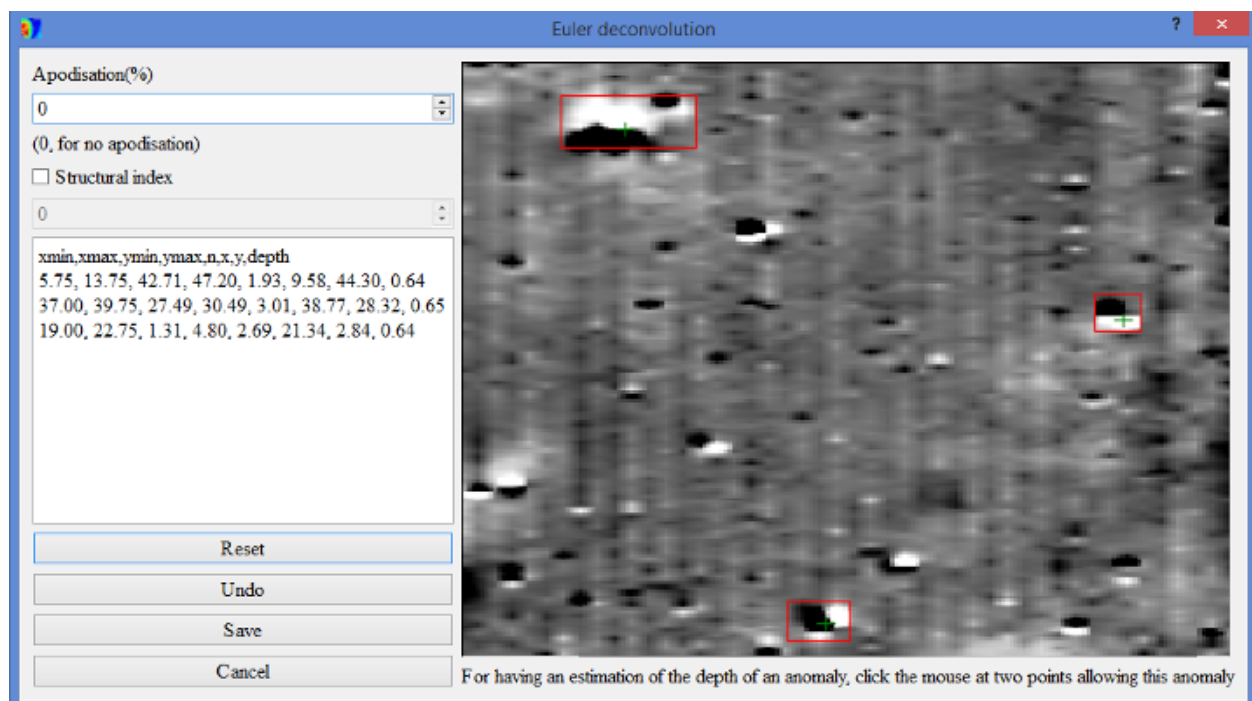


Note: For more informations about this processing, see the GeophPy documentation.

7.7 Euler deconvolution

Classic Euler deconvolution.

Euler deconvolution is a method to estimate the depth of magnetic sources that do not required reduced-to-the-pole data.



The “Undo” Button allow to cancel the last action. After having calculated Euler deconvolution for several zones, it’s possible to save these data in a “csv” file (with ‘;’ as delimiter) by clicking on the “Save” button.

Note: For more informations about this processing, see the GeophPy documentation.

GEOREFERENCING

Dataset georeferencing using a set of Ground Control Points (GCPs).

The GCPs can be edited using this dialog box. Available only if a Geographic Positions Set is opened and displayed.

num	Longitude	Latitude	X	Y
0	32.52432754649924	34.70609241062902	0.0	0.0
1	32.52387864354049	34.70625596577242	50.0	0.0
2	32.52365816268757	34.70584594601077	50.0	50.0
3	32.52343735426504	34.70543469612403	None	None
4	32.523888250117494	34.70527059429577	None	None
5	32.524109276853444	34.70568334212192	0.0	50.0

Note: For more informations about georeferencing, see the GeophPy documentation.

FEEDBACK & CONTRIBUTE

Your feedback is more than welcome.

Write email to lionel.darras@mom.fr, philippe.marty@upmc.fr or quentin.vitale@eveha.fr

CITING WUMAPPY

To cite this software : “Marty, P., Darras, L. (2015). WuMapPy. Graphical User Interface for sub-surface geophysical survey data processing (version x.y) [software]. Available at <https://pypi.python.org/pypi/WuMapPy>.”

CHANGELOG

11.1 Version 0.32

Released on 2019-08-01.

- HTML documentation theme changed to Read the Docs theme.
- Added windows resizing possibility.
- Added windows resizing possibility.
- Fixed compatibility with Python 3.6 (and superior).
- Fixed recent matplotlib version compatibility
- Added automatic sub-windowing for Euler deconvolution.
- Added import/export from/to Surfer grids.
- Added directional filtering (plough filter).
- Added the possibility to change the number of levels for '2D-CONTOUR' and '2D-CONTOURF' plot type.
- Added setmin/setmax possibility to limit data range for correlation calculation in festoon filter.
- Added zero-mean profile filter.
- Change import ascii dialog box style.
- Added row and column span possibility for widgets in the filters dialogboxes.
- Added csv file preview in ascii import dialogbox.

11.2 Version 0.31

Released on 2018-01-02.

- Fixed WuMapPy pip installation issues and updated documentation.

11.3 Version 0.30

Released on 2017-12-01.

- Updated WuMapPy documentation.
- Fixed Georeferencing GUI.

- Fixed Euler deconvolution GUI.
- Implemented Wallis filter.
- Added Mean cross-track profile plot for destripping filters.
- Added options for constant destripping filter.
- Added non uniform shift for Festoon.
- Updated Peak Filtering GUI.
- Fixed cancel button behavior in the import from ASCII files menu.
- Added color map preview in Display settings menu.
- Fixed reading delimited file issues.
- Fixed GUI group boxes' names.

11.4 Version 0.21

Released on 2016-06-30.

- Graphical User Interface Theme changed to 'Cleanlooks'.
- Text Font modified.
- Horizontal Layout TabWidget added.
- Design of all dialogbox modified.
- Bug correction for the Histogram in none real time update.
- Matplotlib navigation toolbar (zoom, pan, save) added.

11.5 Version 0.20

Released on 2015-09-10.

- Initial version.