

Introduction



Mirone is a MATLAB-based framework tool that allows the display and manipulation of a large number of grid formats through its interface with the GDAL library. Its main purpose is to provide users with an unusual easy-to-use graphical interface to the more commonly used programs of the GMT package. In addition it offers also a large number of tools that are particularly focused to the fields of geophysics and Earth Sciences. Among them, the user can find tools to do multibeam mission planning, elastic deformation studies, tsunami propagation modeling, IGRF computations and magnetic Parker inversions, Euler rotations and Euler poles computations, plate tectonic reconstructions, seismicity analysis and focal mechanism plotting, advanced image processing tools, etc... The high quality mapping and cartographic capabilities that made GMT known worldwide is guaranteed through the Mirone's capability of automatically generating GMT cshell scripts and dos batch files.

As mentioned, Mirone is written using the MATLAB programming language. MATLAB is a very powerful tool for software development. The language is very easy to learn and contains an endless collection of mathematical functions that highly facilitates the task of writing complicated code, including Graphical Users Interfaces. However, it also suffers from important limitations. Speed and specially memory consumption (voracity) are the main ones.

To circumvent the above constraints almost all heavy computations involving matrix data are done in Mirone with the help of external code written in C and compiled as MEX files. Since those MEX files use the scalar programming type and use single precision or short integer variables whenever that is all that is needed to not compromise the result accuracy, their memory consumption is reduced to the minimum necessary while the running speed is that of a compiled code.

Although Mirone is written in MATLAB, a stand-alone version to run under Windows is also provided. This version was originally a bit less efficient than the native MATLAB code but since it is compiled with the Intel compiler it became actually sensibly faster than the MATLAB version (not so much wonder, since the latter gets permanently fatter with Java shit). The compiled version has its own advantages. Namely, we can drag-and-drop files onto the Mirone desktop icon to open them. In addition, it even supports file associations but in order that this works the user must set the environmental variable **MIRONE_HOME** with the path to the root Mirone's installation directory (this should be taken care of by the installer).

Installation

First, although not absolutely mandatory it is highly recommendable that you install GMT. This is as difficult as to download the setup [HERE](#) and click next to all options. Mirone is distributed in both source code and a Windows compiled stand-alone version. The condition is that you have MATLAB 6.5 version (or higher). To run it from MATLAB, unzip the source code to wherever directory you want. Within MATLAB, change to that directory and write "mirone" (without the ") on the command prompt. This, however, is a wasteful way of doing things. The point is that MATLAB with its Java interface consumes a lot of computer memory and has a lousy memory management capacity. A better way **if you have R13** is if you start MATLAB with the -nojvm option. This saves a lot of computer resources. See the instructions on the startup.m file that you find on the Mirone's installation directory for how to create a desktop icon to do this automatically. Nothing of the previously said implies dependency to a particular Operating System, so Mirone runs equally in Windows or Unix/Linux or Mac OS X. However, in order to run it in non-Windows platforms, a bunch of MEX files have to be compiled for that system. The [GDAL](#) and [OpenCV](#) libraries must be installed. The *MEX* files can be built from the files that are installed under the "mex" directory of the root's Mirone directory. Regarding instructions on how to compile them under other OSs the most I can help you right now is, see the makefile.mexs file and adapt it to your environment. There are Linux and Mac OSX versions on the Mirone download site but they are by no means subject to the same maintenance care as the Windows version.

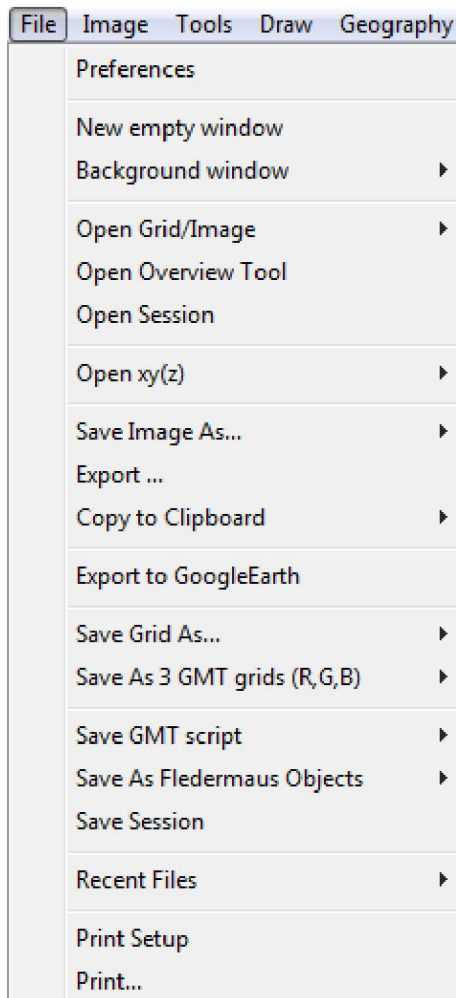
On the other hand, the compiled version works only in Windows (or under Wine in Linux or Crossover in OSX). It is provided as a [setup](#) where you only have to say "yes" to most of the questions. At the end you will have one more icon on your desktop. You drag-and-drop on the icon most of the file types that Mirone recognizes and it will open them at once.

If you have installed GMT using the above link you will have available the [GMT](#) coastlines files up to high definition. There is still one higher lever (5 times higher) called the full definition file. Since it occupies about 40 Megs it is not distributed by default and if you want to install it, you have to do it yourself. To do so, download it either from official GMT site or from [here](#) and uncompress the file into the ../share/coast of the GMT installation directory. If, on the other hand, you did not install GMT than you will not have any coastlines available (Datasets->Draw coastlines->... show up grayed). In that case it is still possible to install only the netCDF coastlines files and make them available to Mirone use. You need than to download those files from the [GMT](#) site, install them in a directory of your choice (and please, don't use the ultimate stupid "c:\Documents and Settings\My ... location but if you do, don't bother me with questions why something is not working). Next read and follow instruction on the file "coastline.conf" locate in the Mirone's "gmt_userdir" directory.

Starting on version 1.4 you can run the Mirone with Matlab version from anywhere (that is, not obligatory to run it from the Mirone installation directory) as long as you do the following:

1. Add the Mirone installation dir to the Matlab path
2. Add the Mirone installation dir to the Windows path

File

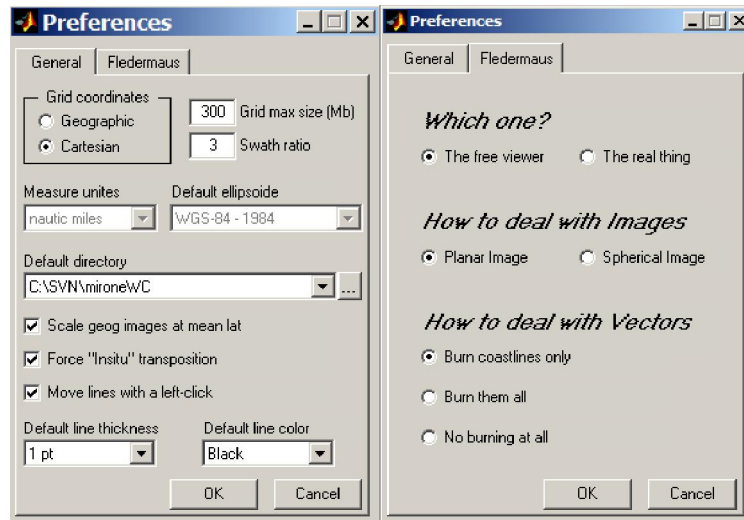


This menu provides entries that deal with most of input/output operations. "Most" because there are some grid tools that have their own loading entries. On the other hand, with the stand-alone version, the easiest way of loading a file (be it grid, image or vector data) is to simply drag-and-drop the file on the Mirone icon.

Preferences

Upon start the user is presented with Mirone in its basic form which is a simple bar with menus only. Activities are initiated by using those menus. Although Mirone comes with pre-defined defaults it is very important that the user understand what they mean and do. The first time the program is used the user is strongly advised to hit the button with hammers icon. That opens the preferences window. The meanings of those settings are as follow:

The grid/image type of coordinates. Select Geographic (default) if you are working with a grid whose coordinates are longitude and latitude. Select Cartesian otherwise. This is an important setting because some operations, like measuring, depend on knowing if we are dealing with geographic or Cartesian coordinates. When a grid is read, Mirone tries to guess what coordinates it has and although it guesses correctly most of times, it may fail.



- *Grid Max size* - In order to do all grid manipulations Mirone stores the original grid in the computer's memory (regardless of their type they are always stored as single precision). This is the maximum size in Mb that a grid can have and be held on the computer's memory. This does not mean that larger grids/images cannot be processed. It only means that if they are larger than this value, they will be re-read from disk whenever necessity demands. It is up to you (based of course on the computer capabilities) to decide on a reasonable value for this parameter. Grid size is computed in the following way: $n_rows \times n_columns \times 4 / (1024 \times 1024)$. Note: Mirone does most of its heavy computation with external MEX files that use single precision arithmetic's. However, some operations may still be done with MATLAB operators which imply double precision (in this case it takes twice as much memory). When you are using image files, however, this rule does not apply anymore. In fact there is no obvious rule because you may be using compressed or uncompressed images.
- *Swath ratio* - When doing a multibeam planning one always has to know the swath width coverage as a function of the water depth. When you start the first track, a question will be made if you accept this value or want to change it. Subsequent tracks won't ask it again, so if you want to change it in the middle of a planning session, you have to come here.
- *Measure unites* - When computing distances that's the result is presented in the chosen unit. Available options are: nautical miles, kilometers, meters and user (it means I don't know what are these).
- *Default ellipsoid* - Mirone computes distances on an ellipsoidal Earth. Among the various ellipsoids you will also find one for Mars. Do not forget to select this when working with MOLA grids. Besides the default ellipsoids an alternative mechanism is provided to supply different ellipsoids via an external file. In the '*data*' subdirectory there is one example file named *ellipsoids_example.txt* that shows how such a file should be written. To make this file active, rename it to *ellipsoids.txt* and it will then take precedence over the built-in default ellipsoids.
- *Working directory* - To reduce the cumbersome task of browsing through the directory tree every time you need to save a file, the program uses a default working directory. You may either write the full path here or press the side button and select one within the selecting window that will open. Besides this mechanism, the program also remembers the last directory used to open a file and will use it as a first guess for the next time you need to open another file.

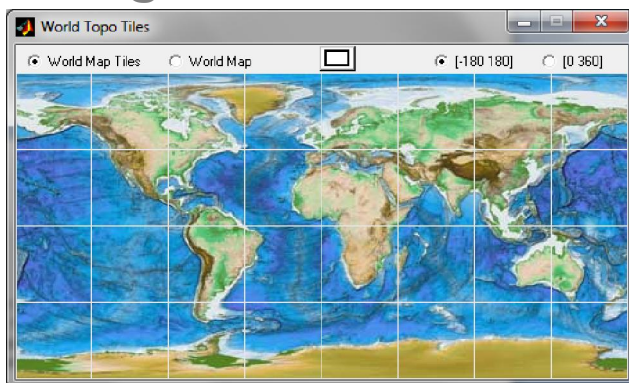
- *Scale geog images at mean lat* - Since 1 degree of longitude and latitude do not cover the same arc length at Earth surface, isometric plotting of geographical grids squeezes the image vertically. Scaling the Y's axis to the cosine of the mean lat minimizes this effect. However, if the image has a large latitudinal extent the image will have a nearly one-to-one relation at the center but will show an important distortion at top and bottom.
- *Force insitu transposition* - This is an option to help importing large grids. Importing grids implies a conversion that uses matrix transposition. This operation is fast if we make a copy of the importing grid. However, this requires twice the grid size on memory. If you do not have enough memory to import a large grid, use this option that does the transposition *insitu*. That is, it uses only one time the grid size in memory. The price to pay, however, is in speed because it runs about 10 times slower.
- *Move lines with a left-click* - Controls what mouse selection is used to move polylines/patches. If checked lines are moved with a left click drag-n-drop. Though easier to operate this has often annoying side effects (move the polyline when in fact we wanted to edit one of its vertexes). Uncheck if you want to use a Shift-click left mouse button or click both left and right mouse buttons to move the line. A bit more cumbersome, but safer.
- *Default line thickness* - Use this default value for drawing lines, circles or other graphical elements.
- *Default line color* - Use this default value for drawing lines, circles or other graphical elements (such as text strings).

On the second tab – Fledermaus – one selects the behavior when creating Fledermaus objects by clicking one eye icon of the Mirone bar. As mentioned in [Save as Fledermaus object](#), we can choose between planar or spherical objects (useful to visualize global grids). The other options under *How to deal with vectors* control if line elements are transferred as vectors, in which case they have an altitude attribute that allows them to intersect the surface, or if they are rasterized into the image.

New empty window

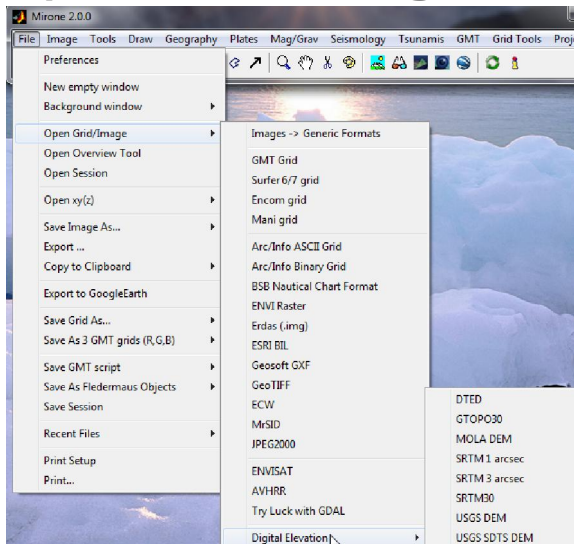
Opens a new empty Mirone window from where you can open another file. You can obtain the same result by clicking on the first icon of Mirone's Tool bar. This is preferable than to open a new Mirone session (for instance, by clicking on the Mirone icon) since it's faster and uses less memory. It is also the mandatory way for opening files that can be [linked](#), [drape](#) or [retro-illuminated](#).

Background window



You can either select a tile of the world map, the whole world or a rectangular region by clicking on the button with the rectangle icon. You can also select a blank map, for which case the corner coordinates must be provided. This option is the “poor man choice”, use it when you need a background map to define a region to work with.

Open Grid/Image

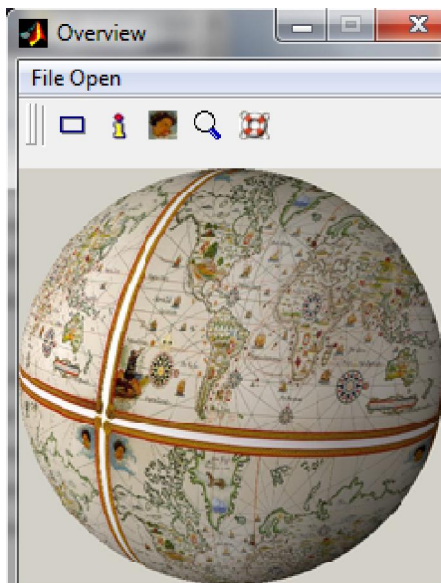


By default Mirone automatically detects and reads the large majority of the formats that it recognizes. However, you may need to specifically instruct it to read a particular format. For that select the type of file you want to open from the available options. Those options are a large subset of the GDAL supported formats. Excluded from GDAL formats are those that deal with satellite images (they are complicated to manipulate and potentially very big). It is also possible to read MOLA (Mars grids) files. Surfer 6 and 7 binary grids are detected automatically. Inside the *Digital Elevation* you will find DTED, GTOPO30, GeoTIFF DEM, MOLA DEM, SRTM 1, 3 & 30, USGS DEM, USGS SDTS DEM. The *Generic Formats* are the common image formats (png, jpeg, tiff, etc...)

Images with a .raw or .bin extension accessible via the *Images->Generic Formats* input dialog have a special interface to load them. It is assumed that files of this type are *raw* binary, eventual multi-band, file formats. By *raw* we mean an image file without any header information on its, contents, size, etc... On those cases it is the user responsibility to provide the missing information and we do that using the interface below. Each edit box has its own tooltip help that displays when pointed with the cursor.

Open Overview Tool

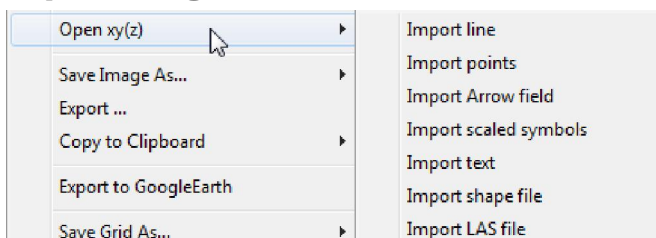
This tool is meant to use with very large grids that otherwise would use most of available memory or not be load at all. The selected grid is sub-sampled, during the reading stage, in order to obtain an overview of about 200 rows/ columns. This way, arbitrarily large grids may be previewed, but the quality of the preview will degrade with grid size. Click on the rectangle icon to draw a rectangle on the image. Double clicking the rectangle lets you edit it. A precise control of the rectangle size is available by a right-click and selecting "rectangle-limits". Selecting "Crop Grid" extracts the region inside the rectangle at the grid's full resolution.



Open Session

Open a previously saved session (see below the [Save Session](#)).

Open xy(z)



Imports a generic ASCII file that can be or not a multi-segment file. Multi-segment files (in a GMT sense) are files that start by a line with the '>' character and each segment is separated by one of such lines. The multi-segment files allow more information to be transmitted after the

">" character. Namely, If first line in file is of the form:

'>U_N_I_K' plot a single line NaN separated (Useful when file has many segments)

'>ARROW' plot an arrow field

'>-:' swap 1st and 2nd columns (assumed as (y,x) -> (x,y))

'>CLOSE' plot patches instead of lines (independently of pline being closed or not)

'>VIMAGE' tell Fledermaus to plot a scene with a VIMAGE. In this case the VIMAGE keyword must be followed, in the same line, by the full name of the image to be plotted vertically. The next two lines in file must contain the first and last point of a line that will determine the image's horizontal extent. Note that this will show only when file is visualized in Fledermaus.

Further from the keywords just described, the segment separator line may contain other type of information. One is explained in [Import scaled symbols](#) and regards the control of line/symbol properties. The other is information considered pertinent to be displayed via a right-click on the element and selection the corresponding "info" option. This information is stored after the '>' character as an text string. The information available for the [Magnetic isochrons](#) work exactly this way. Example: "> Anomaly 5 Eurasia/NAm" (without the quotes).

Import line

Prompts for a file name of an x,y ASCII file that will be drawn as a line on the current figure. That line will be given the same properties as the natively drawn lines.

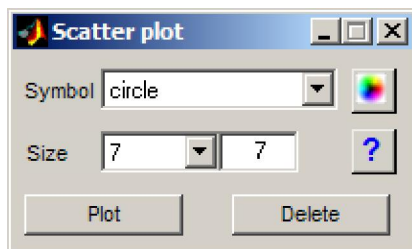
Import points

Prompts for a file name of an x,y ASCII file that will be drawn as individual points on the current figure. Points will be given the same properties as the natively inserted symbols.

Import Arrow field

Prompts for a file with 4 columns. First two contain (x,y) location of the arrow base and the other two columns should contain the U and V components. For the time being, not much control on the arrow size is provided. Arrows are scaled automatically.

Import scaled symbols



Prompts for a filename of three or more columns file. The number of columns (again, minimum is 3) the plot result. With 3 cols, the third is used to attribute symbol color. That is done by scaling Z values to the current colormap. Symbol type and size is selected in the next figure. If the file has four columns, fourth column contains symbol size in points. If 7 columns then column 5-to-7 should contain the RGB color in the interval [0 255] or normalized [0 1].

One other manner of controlling the symbol plotting is to use multi-segment files and the `-W`, `-G` and `-S` option switches after the `>` character. `-G`, `-W` and `-S` determines the fill color and outline type in a similar (although not so extent) way as the corresponding switch of GMT's `psxy` program.

-G Select color for filling the symbol. Specify the grey shade (0–255) or color (r/g/b, each in range 0–255). Example `-G255/0/0` fill with red.

-W Set pen attributes for lines. Valid options are `-W1,38/130/255` `-W3` or `-W100/255/255`. If omitted, line thickness is 1 point. If omitted, line color is the default line color.

-S Select among `'+o*xsd^v><ph'` MATLAB code for symbols. For example `-Sv` plots inverted triangles.

If you select this from a Mirone window with a previously loaded grid/image, country plots will be clipped to the region delimited by the image with an automatic map projection if necessary. If, however, you call it from within an empty Mirone figure, data BoundingBox is used to create a background region image to hold the data limits.

You can also load a file that is a text file list. That is, a file that has the full name of a data file, one per line.

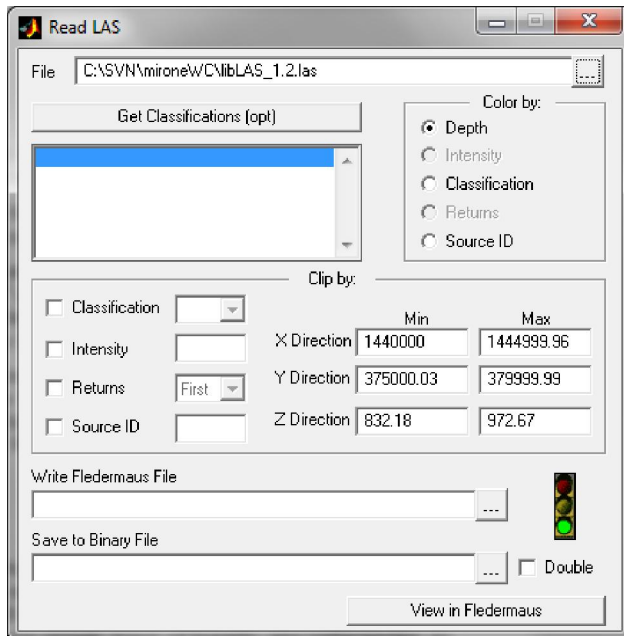
Import text

Import text from an ASCII file. That file must have the x,y coordinates in the first two columns, followed by the text.

Import shape file

Load an ESRI shape file. It currently parses only "polygon", "point" and "arc" attributes. Shapefiles can be loaded from an empty Mirone window. In that case, data BoundingBox is used to create a background region image to hold the data plotting.

Import LAS file



This option calls a helper window to assist what to do with the LIDAR data. Since these files are potentially very big and Matlab is a horror to manipulate large datasets, and its 3D viewing capabilities are simply pathetic, the tool assist in parsing by parameters and lets user save a binary file or see the point cloud in Fledermaus.

Save Image As...

Saves the current display image into one of the common image formats (e.g. jpg, tiff, bmp, etc...) and less common such GeoTiff, Erdas Imagine, Envi .hdr Labeled, ESRI .hdr Labeled and JPEG2000. It is also possible to save a "world file" using the "Save .tfw file" option. World files are simple ways of linearly geo-referencing an image. They consist of 6-line ASCII file such as

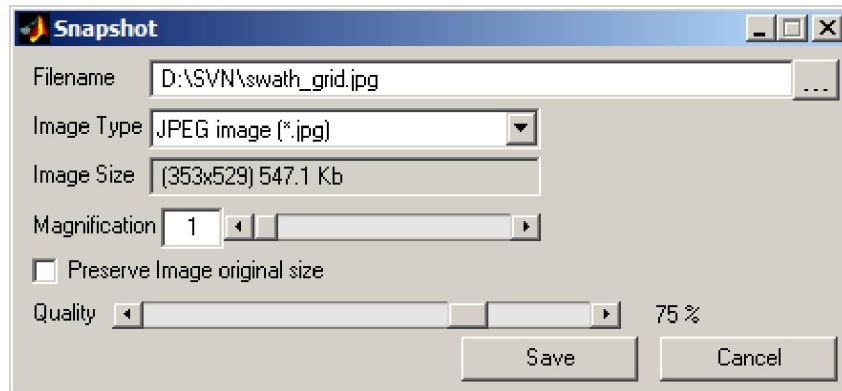
```
2.4384
0.0000
0.0000
-2.4384
441794.4342
5094101.4520
```

These are the 6 coefficients of a three-by-three transformation matrix. They indicate that a pixel is 2.4384 meters square, and that the center of the upper left pixel has an easting (x coordinate) of 441794.4342 and a northing (y coordinate) of 5094101.4520. For fans of the matrix:

$$x = 2.4384 * \text{column} + 0.0 * \text{row} + 441794.4342$$
$$y = (-2.4384) * \text{row} + 0.0 * \text{column} + 5094101.4520$$

The second and third values are zero because there is no rotation or skew. The convention is that a world file is named by jamming a w into the extension. For example, o47122d3.jpg might have a o47122d3.jgw file and o47122d3.tif might have a o47122d3.tfw file.

Be aware that with this option, graphic elements (e.g. lines) are not saved. Only the image is (but see [Export](#)).



Export

Does a screen capture and save it into one of the common image formats. This differs from [Save Image As...](#) where graphic elements (e.g. lines) are not saved. The image size/resolution is selected in the export window shown above.

Copy to Clipboard

This option, available only on windows, copies the whole image contents and places it on the Windows clipboard. Pasting is obtained with the classical Ctrl-v.

There are two sub-options: *Image only* & *Image and frame*. As their labels indicate, they differ in fact that the second capture the image and the frame (that is, the coordinates annotations) and the first not.

The screen captures behaves differently when a color palette is displayed. If the palette is of the "Show Inside" type, no attempt is made to reproduce the displayed configuration and the palette will end somewhere inside the image and close to its right side. For "Sow At side" palettes, the capture tries to maintain what is in the screen but the end proportions are not exactly the same.

Image capture is done at 300 DPI, which makes this the most easily and quality prone way of inserting figures in Word documents.

The "Copy-active-line" and "Paste line" are used to copy-paste line elements between two different figures. In order to be able copy a line, this must be selected by putting it in edition mode (a double-click on the line). Next you can do a Ctrl-C and after a Ctrl-V on the second image. Note that this is a copy with coordinates operation, so the line will be pasted on exactly the same coordinates that it had on the first figure. BONUS: once you have copied the line with the Ctrl-C you can past it in any text editor (please, anything but Notepad) on in EXCEL.

Export to GoogleEarth

When the displayed image is referenced using geographical coordinates one can export the image plus its other graphical contents into a .kmz file (a .kmz is just a .kml file zipped and

changed the extension name). Due to its potential large size, coastlines are not exported. Furthermore Google Earth, at smaller scales display its own coastlines, which are very low quality as compared with the GMT ones used by Mirone.

The Datasets plate boundaries, earthquakes and volcanoes have special extra information associated with them.

To save space Mirone tries to be clever in picking the most economic file format used to save images. Images derived from grids with NaNs will have those values set to transparent.

Save Grid As...

There are currently 8 format possibilities to save a grid:

1. GMT grid – the default new netCDF coards compliant GMT format
2. Surfer 6 grid – Golden software grid binary format (ancient Surfer 6 format)
3. Encom grid – The Encom (www.encom.com.au) grid format used by ModelVision
4. GeoTiff
5. Erdas imagine
6. Envi .hdr Labeled – (www.itvis.com/envi)
7. ESRI .hdr Labeled
8. JPEG2000

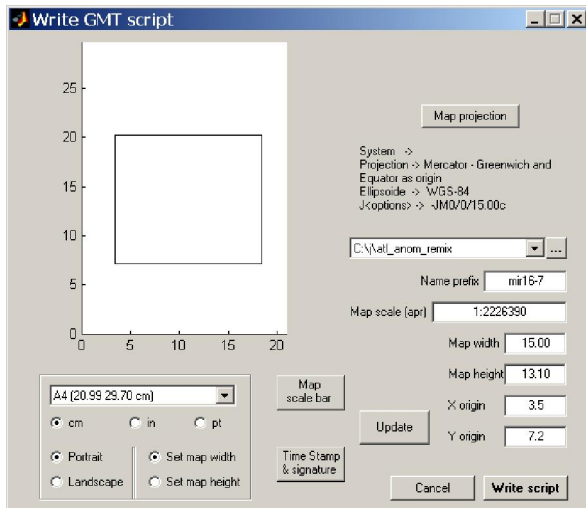
Note that although GeoTiff as well as JPEG2000 are more known to store images one can use them to store floating point grids as is the case here. Floating point GeoTiffs are also known by GeoTiff DEM.

Save As 3 GMT grids (R,G,B)

Decompose the color image into 3 components, Red, Green & Blue and save each of these components into a GMT grid file. The output name is used as a template to which the `_r`, `_g` & `_b` strings are appended in order to identify the individual color components. This option is extremely useful with GMT4 version, for it will allow transforming any image into GMT grids. You can then (for example) drape satellite images over DTMs, use the alternate illumination algorithms, etc... There are two options on how to save. *Image only* will save only the image and ignore all graphic elements that may exist in the figure. It will also create grids that have the same number of rows and columns as the original file (be it a grid or an image). *Screen capture* saves the image, and all existing graphic elements, by doing a screen capture.

Save GMT script

cshe or dos *batch* files generation. In this window it is possible to fine tuning the map details like its size and position on the page as well as the map projection. This tool searches systematically all elements and guesses how they were produced. It then decides what is the best way to reproduce the display using only GMT commands. For example, if the image displayed derives from a GMT grid and it was illuminated with the shading method used by *grdgradient* the script lines will generate an illumination grid from the original GMT grid and use *grdimage* with the `-I` option. If that is not the case then a screen capture is executed and the image saved as three R,G,B grids. Those three grids are then utilized by *grdimage* to rebuild a color image. Line elements are written on disk and later used by *psxy*. However, coastlines, national borders and rivers or contour lines are not saved. Instead, appropriate calls to *pscoast* or *grdcontour* are inserted into the script. To



change the map size, double click over the rectangle and that click and drag. Every time you edit the rectangle is necessary to hit the "Update" button. The script (or the batch file) and the eventual auxiliary files are written into the directory declared below the Map projection settings. The script (or batch file) is abundantly commented which makes it easy to educated GMT users to change it further if they are not fully satisfied with the code generated result. To run it, just cd to the directory it was saved and type its name.

Save as Fledermaus object

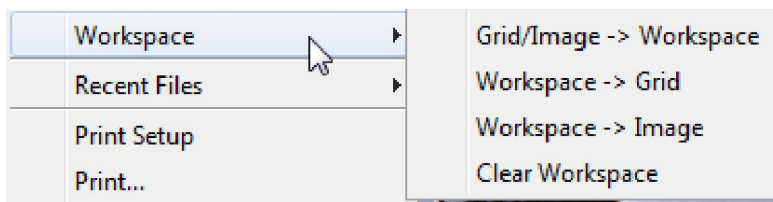
Grids (with whatever they contain) may be saved as .sd IVS objects. These objects may be viewed with the iview3d free Fledermaus viewer. Fledermaus is a fantastic 3D software (probably the best) and has a free viewer that can be downloaded from www.ivs.unb.ca/products/iview3d/. If you have it installed you can either use this option to create a .sd file and separately run it with the viewer or click the icon "eye" in the Tool bar which will do both (but not save the .sd file). Note that large grids produce large .sd files so if you do not have that much RAM (or you want to save the .sd for later use) it may be better to use this save option.

There are two types of Fleder files and their labels are quite obvious for what they mean: *Spherical Fledermaus Obj* and *Planar Fledermaus Obj*

Save Session

Saves the current state of your work, including all (well, most of all) graphic (drawing) elements into a matlab .mat file. Selecting [Open Session](#) resumes the saved session. Very useful for backups. Notice that the grid file is not stored but only the information needed to reconstruct the image. So if the original grid is not found you cannot resume your session. This functionality is not entirely up-to-date with the latest added options (e.g. drawing Euler pole circles).

Workspace



This option and the others in this group will only show up in the MATLAB running version. It allows exporting variables into the MATLAB's workspace that you can manipulate using the full power of the MATLAB

engine.

Grid/Image -> Workspace

Depending on what you are exporting, a different group of variables is generated. When exporting from a loaded grid, the set of variables is: *Z* an array of doubles containing the grid data; *img* the image derived from *Z*; *head* a 1x10 row vector with [*x_min x_max y_min y_max z_min z_max reg x_inc y_inc grd_format*]; *X* & *Y* row vectors with the grid's *x* & *y* coordinates; *FigHandle* the figure handle and *is_grid* which will be 1 to indicate that we have a grid. When exporting from a loaded image, the variables are: *img* the image; *X* & *Y* row vectors with [1:n_columns] and [1:n_rows] respectively; and *is_grid* which will be 0 to indicate that we have an image. Use your MATLAB skills to do whatever you want with either *Z* and/or *img* but normally you should not mess around with the other variables. NOTE: you can export two grids sets. In order to do it you must have open two Mirone windows, each containing a loaded grid.

Workspace -> Grid/Image

Import the *Z* variable, after whatever manipulations you have submitted it, and launch it in a new Mirone window.

Workspace -> Image

Import the *img* variable, after whatever manipulations you have submitted it, and launch it in a new Mirone window.

Recent Files

Maintains a list with the last 14 opened files. This is a very convenient way of loading files that are located somewhere in the disk.

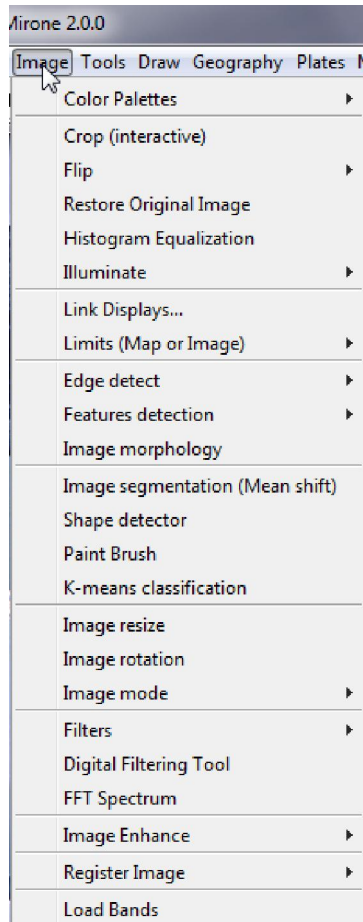
Print Setup

Configure your printer to receive output from Mirone.

Print

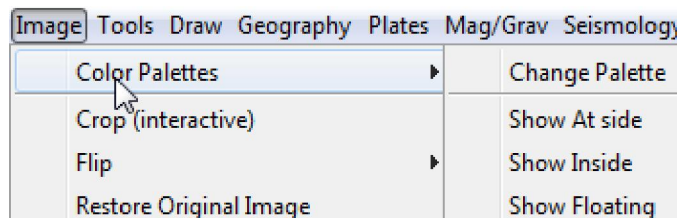
Spend lots of money in color cartridges.

Image

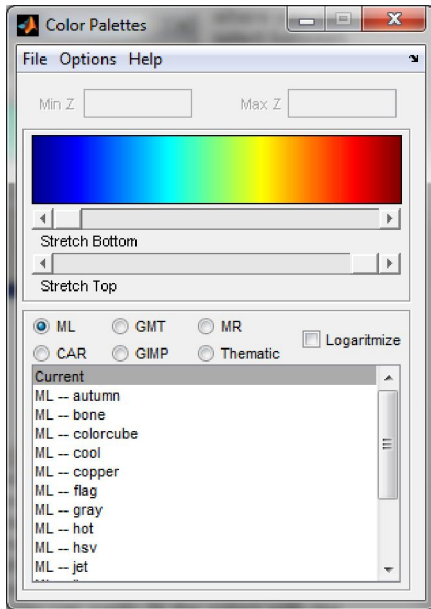


This menu provides entries that deal with several aspects of image processing operations.

Color Palettes



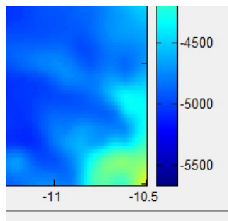
Change Palette



Opens a window tool from where you can select between more than 100 color tables. Every selection takes place immediately. Every time that a new color schema is selected the base image is automatically updated. Furthermore, all palettes may be changed by a double click over the color bar. That inserts color markers. Drag these markers to modify the color map. A second double click deletes the marker. The *Min Z* and *Max Z* fields may be used to impose a fixed color map between those values. The *Logarithmize* check box converts between linear (default) and natural logarithm color maps. External GMT color palettes may also be imported or exported via the *File* menu. Note: If you have previously drawn contours into your image, you can easily fit the colors with the contours.

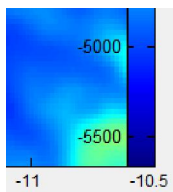
The *Thematic* group hosts dedicated palettes for specific displays. Namely, for magnetic anomalies, separation of bathymetry and topography, for bathymetry or topography only.

Show At side



Appends the current palette on the right (outer) side of the currently displayed image as the example shown in the side figure.

Show Inside



Shows the current palette on the right (inside) side of the currently displayed image.

Show Floating

Creates a new separate figure containing the palette. This figure is a bit too wide because Matlab doesn't seem to allow figures thinner than 104 pixels (???)

Crop (interactive)

Crops an image inside a specified rectangle. After selecting this option the pointer changes into a cross indicating that you are in the drawing mode. Press the left mouse button and drag to define the crop rectangle. Finish by pressing the left mouse button again. Note, this is crude and little controlled way of doing a cropping operation. The best way is to draw a rectangle on the image. Calmly decide on its size and position and then inquire its properties (by a right-click on the rectangle borders) for "Crop Image".

Flip

Flips the image UP-Down or Left-Right. Note: coordinated (georeferenced) images are displayed with origin at lower left corner and plain images with origin in upper left corner. Some operations like referencing or draping require knowing where to flip or not one image with respect to the other. Although I try to keep trace of this, sometimes I get caught in an unexpected case, and the result is upside-down. Use this option if that happens.

Restore original image

Restores the original image. Use this option when you have been playing around with the color palette tool or histogram equalization or illumination and want to start over.

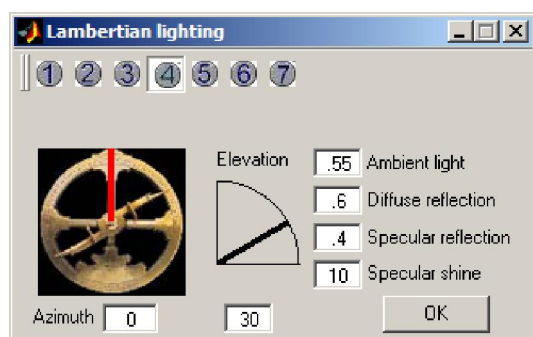
Histogram Equalization

Enhances the contrast of images by transforming the values in an intensity image, or the values in the color map of an indexed image, so that the histogram of the output image approximately matches a flat histogram. This operation works pretty well for black and white images but give somewhat idiot results for color images. The reason relies on the fact that for color images each color band is independently equalized and then a color image is rebuilt. I know that is not the way the thing should be done, but the correct operation (which is followed in **Region-Of-Interest**) is very slow and memory consuming. When an image is equalized, the original image is saved in memory and selecting this operation again restores that original image. This can be used for undoing other image operations (like illuminations).

Illuminate

GMT *grdgradient*
False color
Anaglyph

This sub-menu provides alternate schemas to do image illumination. A window tool allows selection of inclination and azimuth of the light source. Among the available options, the **False Color** lets you illuminate a GMT grid along three different directions and compose the final image with each illuminated direction in the three-color channels (R, G and B). This is useful for enhancing structures perpendicular to each of the selected direction. Algorithms 1 – 4



are the same as the ones used in the GMT program *grdgradient*. Illumination type 4 has an inherent scale factor that is computed from data and for which there is currently no user side control. Due to this, the illumination is sometimes “exaggerated” and not so nice. Other times, however, it provides the best effect illumination of all available options.

Anaglyph - Produces an image that, when seen with red-blue glasses, gives a 3D perception view. It might take a little while for large grids since all

computations are done with pure MATLAB code.

Link Displays

When you have more than one figure displaying images that share, even if partially, the same zone it is interesting to be able to swap the displays. Thus we can, for example, see

what is on the other image at the exact location where of the mouse current position. To do this we link the two displays. This option opens a small figure where you can select the other display which will be linked to the first one. To make it work one only has to right-click on one of the linked figures and the other one will show up in its place. On Mirone compiled version the user must have the second display from a figure created using the [New empty window](#) option or the first icon in the main bar, otherwise the figures reside on separate memory spaces and are not visible to each other.

Drape

For having access to this option you need to have two Mirone windows opened and the second must have been opened using the [New empty window](#) option. This is crucial for one need to know where to drape. Then, the image of the second opened window can be draped on the first one. If the first image results from a grid you can still elaborate (complicate) more by applying an illumination. That illumination will change the Hue (from the HSV model) of the first image. Finally, this image can be saved with the [Save As 3 GMT grids \(R,G,B\)](#) option. Imagine that the first image was obtained from loading a GMT grid DTM model and the second represents a satellite image (or aerial photograph) of the same (not necessarily exactly equal) region. At the end you will have the GMT grid and the draping image split into three RGB grids. You can then use GMT to drape the three grids version of your photo onto the DTM. This option is also useful if you want to drape an image on a DTM before computing a Fledermaus object. A good example is to drape a backscatter image (preferable pos-processed with the ROI options) over a multi-beam bathymetry.

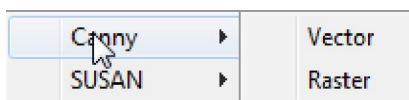
Retro-Illuminate

Like the *Drape* option, you need also to have a second Window opened using the [New empty window](#) option and load a grid. You can than illuminate the second grid with the reflectance computed from the first grid. Note that grids don't need to be of the same size or resolution nor cover exactly the same region. Overlapping zone is computed and the reflectance array is resized to fit the needs of second grid.

Limits (Map or Image)

When you have a non referenced image (a scanned image for example), use this to set the image coordinate limits. A better way of doing this is provided by the option [Register Image](#) available by right mouse clicking onto rectangle elements. A second option is offered to just change the displayed map extends.

Edge detect



Detect edges in images using two different algorithms. Choose between getting the end result as an mask image or in vector form.

Canny

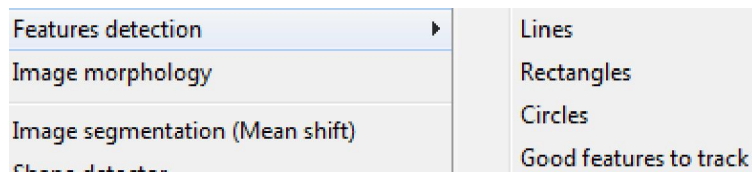
Run an edge detection algorithm (the well known – for the ones who know it - canny algorithm) over the displayed image and plots the result on two optional ways. The *Vector* option will

present the edge detected solution as polylines that may be saved as ASCII files. This is more than the traditional edge detecting operations do. The *Raster* option displays the edge detection as a black-and-white image that can later be used as a mask file. Note that the edge detecting is carried out over the image after it has been converted to a gray image and not on the grid that it derives from. So instead of letting the code do the conversion to a gray scale you can do it yourself and eventually shade that image. This way you have much better control on what is going to be edge detected.

SUSAN

This works in a similar way as the [Canny](#) but using a different and very performing algorithm known as SUSAN. For details on this algorithm, see users.fmrib.ox.ac.uk/~steve/susan

Features detection



Attempt to detect objects with these forms. The word “attempt” is used because it fails in several instances.

Lines

Detect straight lines on image using the Hough transform. Here we use a composition of MATLAB code to compute the Hough transform, since it works better than the OpenCV one.

Rectangles

Detect rectangular shapes on image.

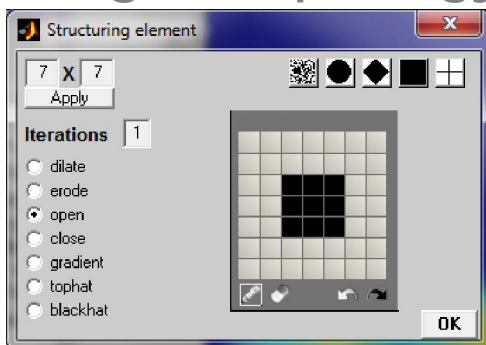
Circles

Finds circles on image using Hough transform. Since this operation works only on gray scale images, color images are converted to gray before applying operation. In order to reduce the number of false detection, the image is firstly subjected to a pair of Erode/Dilation processing. This is all carried out with cvlib_mex MEX wrapper to the OpenCV library.

Good features to track

Determines strong corners (big eigenvalues) on image.

Image morphology



Perform the morphological operations indicated.

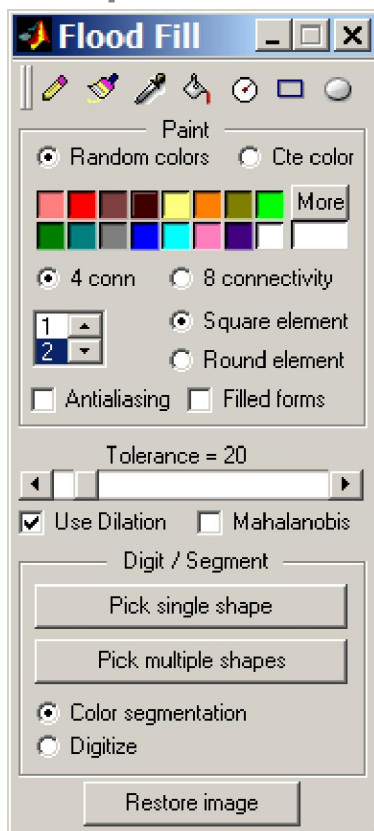
- Gradient – Dilated image minus Eroded image.
- Open – Erode image first and dilate after.
- Close – Dilate image first and erode after.
- Top-hat – Original image minus opened image
- Bottom-hat – Closed image minus original image.

Use the graph window to design the structural element or select among the default ones. Left-click adds the pixel and Ctrl-Left-click des-selects it.

Image segmentation (mean shift)

Another tool to do image segmentation. It uses a technique called "Mean Shift Segmentation". The authors refer to a mean-shift segmentation algorithm presented in "[Mean Shift: A Robust Approach Toward Feature Space Analysis](#)" by Comaniciu and Meer. An implementation of this paper in C++ and code availability exists under the name EDISON. The EDISON system contains the image segmentation/edge preserving filtering algorithm described in that paper. More information about EDISON (and the original codes) can be found here: www.caip.rutgers.edu/riul/research/code/EDISON/index.html

Shape detector



This tool can be used just for painting, in which case it works like the Paint Brush tool presented right below or in a much powerful mode: Color Segmentation. For the floodfill painting, every left-click paints a group of connected pixels. (See below the 'connectivity' and 'tolerance'). A right click stops the painting.

For all other operations a left-click starts the painting, which continues for every mouse movement and stops with any mouse click.

- Specifying connectivity: By default, it uses 4-connected neighbors but you can select 8-connected in the appropriate radio buttons
- Specifying tolerance: Tolerance is used only when the doing painting operations (when using the bucket icon) The "Tolerance" slider selects the maximal lower/higher brightness/color difference between the currently 'hit' pixel and its neighbors. Pixels within the hit pixel \pm |tolerance| are considered to belong to the repainted domain.

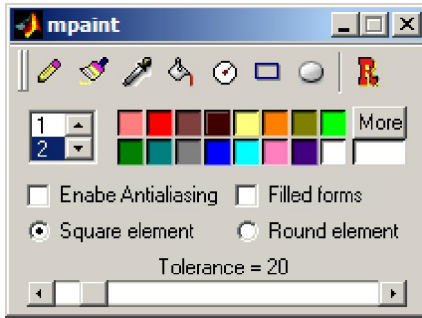
The second mode in which this tool is more unique is in doing color segmentation or automatic digitalization. With color segmentation one can extract a single object in the image or all objects that share the same color \pm half the tolerance selected with the *Tolerance* slider. To select either one or the

other use the indicated push buttons. The object selection is achieved with one left-click over the area of interest. Right after a new image will show up with the selected object and its exteriors in black.

The *Digitize* radio button selects the digitalization mode. Now instead of extracting an object from the image we are going to find its border limits that will be overlaid as a vector which can be saved in a file in the normal way. The *Use Dilation* checkbox indicates that the dilation morphological operation is applied before the limits detection. This normally increases the quality of the digitalization, but you can try without it.

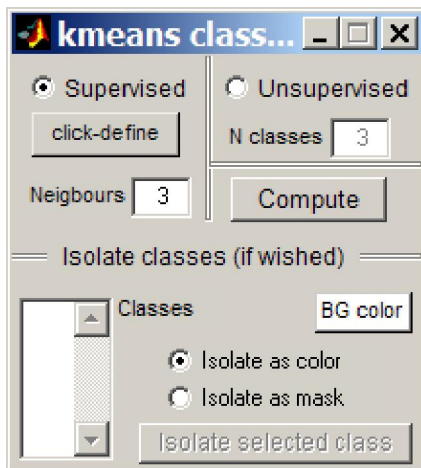
Paint Brush

A fast Paint GUI that works similarly to the Windows Paint program. It differs from that well known program in the way how mouse clicks works. Here, a first left-mouse click starts the



operations, which continue for every movement and stop with a right-click. For a better description on how it works, see above for *Shape Detector*

K-means classification

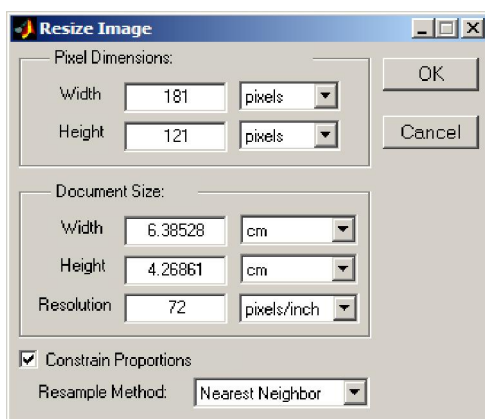


Common image classification procedures can be broken down into two broad subdivisions based on the method used:

- In a **supervised classification**, the user identifies homogeneous representative samples in an image of the different characteristic types (information classes) of interest. These samples are referred to as training areas. The selection of these training areas is accomplished by left-clicks on the image. Each click selects a seed color used to "train" the computer to recognize similar areas for each class. With large images its better to not exaggerate on the number of training areas for this operation is memory demanding.
- In an **unsupervised classification**, the process is reversed with respect to the supervised form. Color/gray classes are grouped first, based solely on the numerical information in the data. The "N classes" edit box selects the number of classes that the algorithm will find

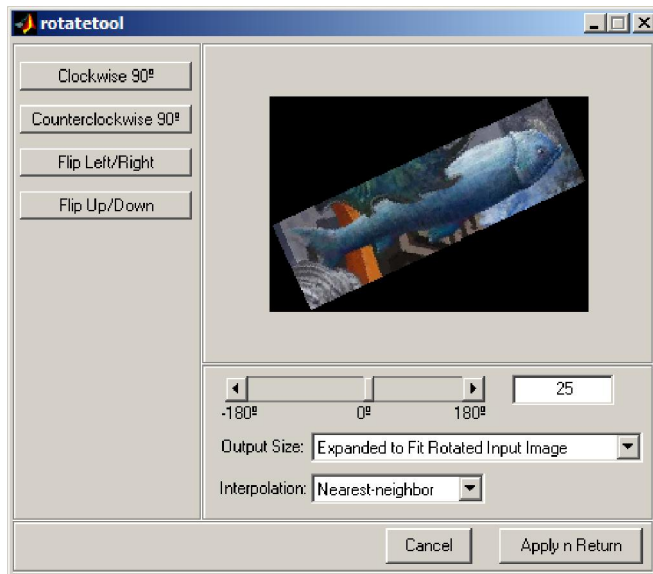
Programs, called clustering algorithms, are used to determine the natural (statistical) groupings or structures in the data. Here, the algorithm used in one known as K-means. This algorithm uses an iterative approach that minimizes the sum of distances from each object to its cluster centroid, over all clusters. This algorithm moves objects between clusters until the sum cannot be decreased further. The result is a set of clusters that are as compact and well-separated as possible. The lower part of the figure (below the *Isolate classes*) can be used to extract one or several computed classes, either as color images on mask files.

Image resize



Resize the image using a Photoshop like front-end window. Sure you all know how to use it. The resizing is performed by the OpenCV lib, which is a guaranty of quality and reasonable memory consumption.

Image rotation



User can specify angle, interpolation method, and output image size using the GUI and see a preview image of the results. Vertical flipping (turns the image upside down) and horizontal flipping (replaces the image with a mirror image of the original) are available as well.

Image mode

Convert between RGB (true color) and indexed images:

- RGB truecolor – converts and indexed image to RGB
- 8-bit color – converts RGB image into an indexed image by a process known as color quantization. The end result of this operation is of variable quality depending on the image complexity.
- Gray scale – convert to 8 bits gray scale images
- Black and White – convert to 2 bits, either black or white images
- Original Image – revert to the original image that was conveniently backup before.

Filters

Smooth (SUSAN)

Median (3x3)

Adaptive Median (7x7)

STD (3x3)

Min (3x3)

Max (3x3)

Range (3x3)

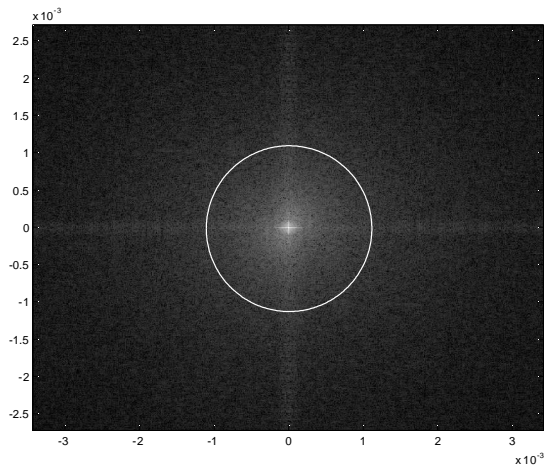
3x3 window

This section is still incipient and contains only a couple of filtering options:

- Smooth (SUSAN) - SUSAN nonlinear noise reduction reduces noise, ideally without blurring the underlying image. For details on this technique see users.fmrib.ox.ac.uk/~steve/susan
- Median (3x3) - Reduces noise in the image by replacing each pixel with the median of the neighboring pixel values over a 3x3 window
- Adaptive Median – Uses an algorithm taken from the Digital Image Processing Using Matlab book. It is a filter capable of adapting its behavior depending on the characteristics of the image in the area being filtered.
- STD – Computes a local standard deviation of image where each output pixel contains the standard deviation value of the 3-by-3 neighborhood around the corresponding pixel in the input image.

- Min – Retains the minimum of the 9 values inside the 3x3 kernel neighborhood around the corresponding pixel in the input image.
- Max – Same as above, but retains the maximum.
- Range STD – Computes a local range of image where each output pixel contains the range value (maximum value - minimum value) of the 3-by-3 neighborhood around the corresponding pixel in the input image.

FFT Spectrum



Interactive FFT filtering. When this option is selected the *FFT Stuff* control figure pops up. Here hit the button that says *Power Spectrum* and a new figure, like the example on the left, with the image's power spectrum is displayed. Next one can draw either a circle or a polygon. The area enclosed by that line will be used to either retain the power values, case in which we are doing a Low-pass filtering or the converse (high-pass filtering). To access to choice of Low or High pass filtering simply right-click on the plotted line and select the desired option from the menu the meanwhile was shown. In case of polygonal lines, the filtering options area little bit more buried inside the "ROI Crop tools" sub-

menu.

Digital Filtering Tool

All linear convolution filters compute weighted averages of the neighboring input image pixels. The only differences between the various linear convolution filters are the size and shape of the neighborhood, and the specific weights used.

User Defined Filters

There are two types of user defined filters, Low-pass Filters and General User-defined. With these filters, you can specify the height and width of the filter neighborhood.

Low-pass Filters - A low-pass filter removes the high frequency noise with the resulting output being a smoother grid. There are three user-defined low-pass filters. Each of these three filters allows you to specify the size of the neighborhood. The width and height of the filter neighborhood must both be positive, odd numbers.

General User-defined Filter - The General User-defined (MxN) linear filter allows you to specify the height and width of the filter neighborhood and any combination of weights. The table in the center part of the window displays the neighborhood size, based on the number of Rows and Cols, along with the weights for each pixel node in the neighborhood. Click in a cell in the table to change the node's weight.

Predefined Filters

The predefined filters have a large collection of 3×3 filters.

- Low-pass Filters are also known as smoothing or blurring filters. These filters remove the high frequency variation.

- High-pass Filters are also known as sharpening or crispening filters. They have the opposite effect of blurring. They tend to remove the background variation and emphasize the local details.
- Order 1 Derivative Filters are used to find horizontal and vertical edges.
- Order 2 Derivative Filters are another set of edge enhancement filters.
- Shift and Difference Filters are the two simplest horizontal and vertical differential operators.
- Gradient Directional Filters compute and return the directional derivatives in each of the eight compass directions.
- Embossing Filters identify and enhance edges aligned in one of the eight compass directions.

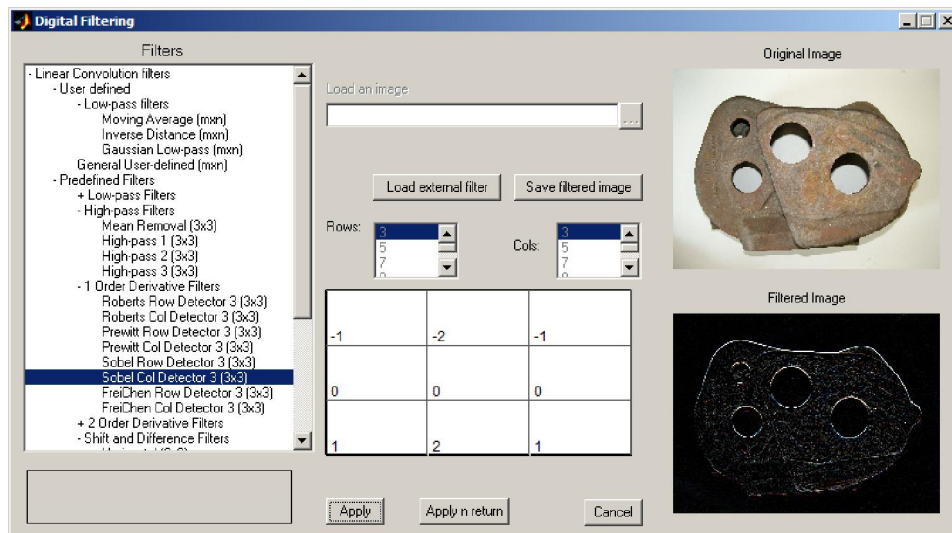


Image Enhance

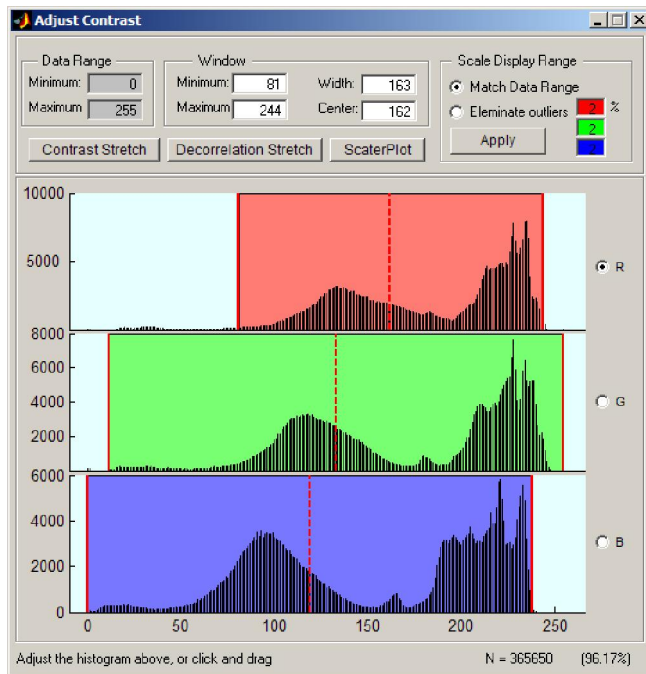
Image Enhance	▶	1 - Indexed and RGB
Register Image	▶	2 - Indexed only
		Image Color Editor (Indexed and RGB)

Indexed and RGB

Sophisticated tool to do image enhancing manipulations in either indexed or RGB (true color) images.

- Adjust the contrast and brightness of the displayed image by manipulating the window over the histogram in the Adjust Contrast tool. The tool provides several ways that you can modify the size and position of the window interactively:
- By grabbing one of the vertical sides of edges of the window and dragging it. You can also change the position of the window by grabbing the center line and dragging the window to the right or left.
- By specifying the size and position of the window in the Minimum and Maximum fields. You can also define these values by clicking the dropper button associated with these fields. Position this cursor over the pixel in the image that you want to be the minimum (or maximum) value and click the mouse button.
- By specifying the size and position of the window in the Width and Center fields.

- By automatically scaling the display range to match the image data range. For example, with the pout.tif image, if you select the Match data range option, the window changes from the default display range (0 to 255) to the data range of the image (74 to 224).
- By automatically trimming outliers at the top and bottom of the image data range. If you select the Eliminate outliers option, the Adjust Contrast tool removes the top 1% and the bottom 1%, but you can specify other percentages. When you specify a percentage, the Adjust Contrast tool applies half the percentage to the top and half to the bottom.



To see what the effect of changes made using one of the possibilities above hit the *Contrast Stretch* button. Another way to enhance the true-color composite is to use a decorrelation stretch, which enhances color separation across highly correlated channels. Use *Decorrelation Stretch* button to perform the decorrelation stretch. You can have a pretty good idea of what happened if you plot the *Scatter Plot* before and after the decorrelation stretch.

Indexed only

Another tool to perform image enhancements but that works only with gray or indexed color images.

This tool enables you to explore the effects of adjusting intensity values and applying gamma correction to images. It also enables you to perform histogram equalization, which modifies the image so that its histogram is approximately flat. The tool displays the histograms of the original and modified images.

When you select "Histogram Equalization", the histogram of the original image is equalized and the resulting image and histogram are displayed. The intensity transformation calculated in the equalization routine is shown in the transformation axes. The x-axis represents input pixel intensity and the y-axis represents output pixel intensity.

When you select "Intensity Adjustment", a number of other controls appear. These controls enable you to increase and decrease the brightness, contrast, and gamma correction. You can specify gamma explicitly using the edit box under the transformation axes. You can also make any adjustment you want directly using the controls on the transformation curve. The red and green control points directly set the parameters which are passed to the IMADJUST command. The yellow point enables you to graphically adjust gamma. Click on the blue line and drag it to move all of the control points simultaneously.

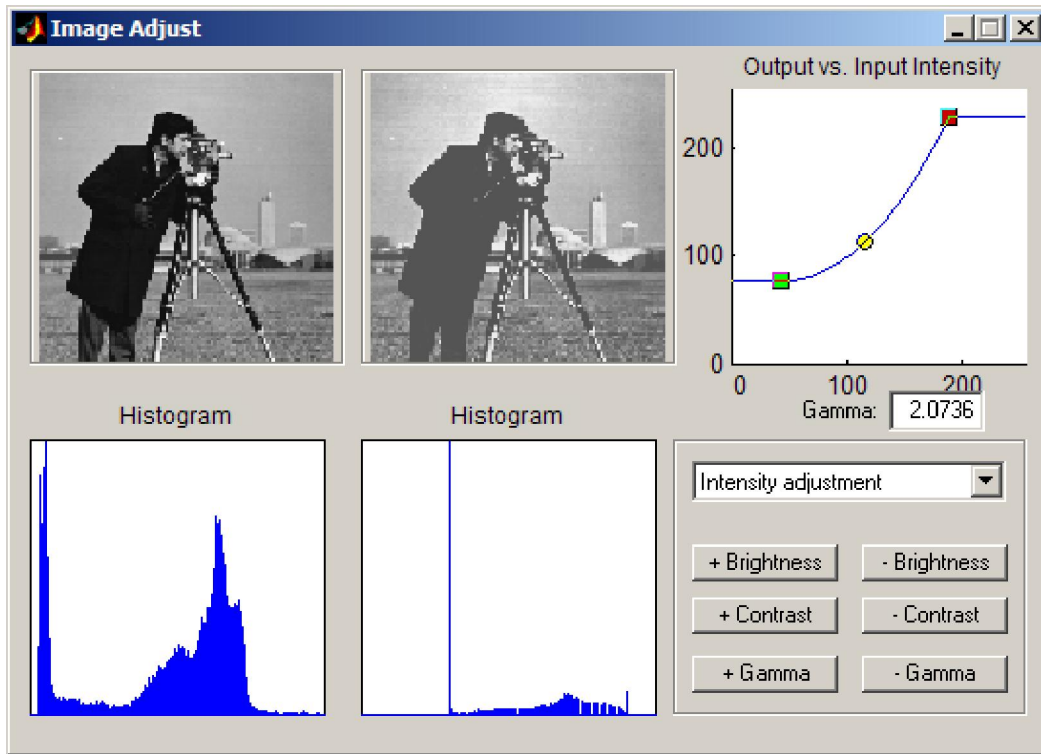
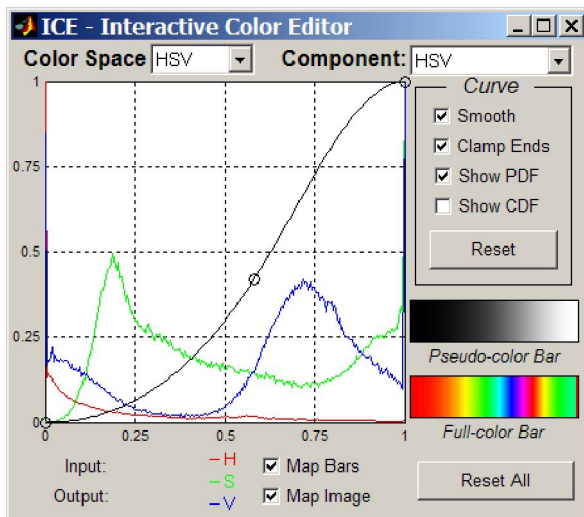


Image Color Editor



The ICE color editor is a modification of a tool with same name that comes with the *Digital Image Processing Using Matlab* book. It has been modified in order to allow interactive selection of the color model and by replacing many chunks of MATLAB code by equivalent OpenCV MEX calls, resulting in a dramatic speed improvement. It transforms an image's color components based on interactively specified mapping functions. The input is the current image, which can be an RGB or monochrome image to be transformed by interactively specified mappings.

ICE displays one popup menu selectable mapping function at a time. Each image component is mapped by a dedicated curve

(e.g., R, G, or B) and then by an all-component curve (e.g., RGB). Each curve's control points are depicted as circles that can be moved, added, or deleted with a two- or three-button mouse:

Mouse Button Editing Operation

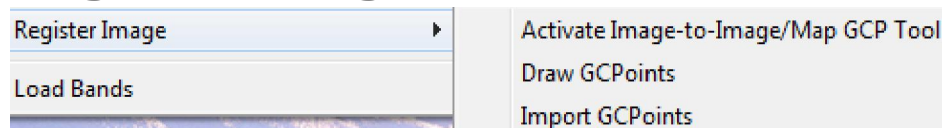
Left Move control point by pressing and dragging.

Middle Add and position a control point by pressing
 and dragging. (Optionally Shift-Left)

Right Delete a control point. (Optionally Control-Left)

Checkboxes determine how mapping functions are computed, whether the input image and reference pseudo- and full-color bars are mapped, and the displayed reference curve information (e.g., PDF):

Register Image

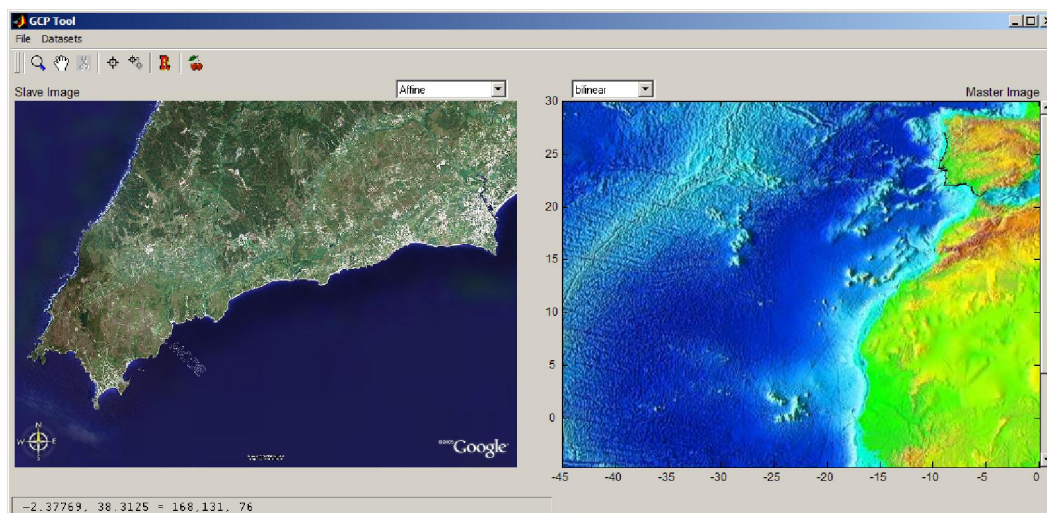


Registering or referencing images can be done using several different procedures. This menu is the entry point for the three more elaborate methods. A simpler and more approximate method is described at [Limits \(Map or Image\)](#)

Activate Image-to-Image/Map GCP Tool

This tool is used to georegister or georeferencer an image by registering an image to a previously registered Master Image and creating a new "georeferenced" image. It works in two steps:

1. Load the reference (registered) image and select this option on Image Menu. Inspecting the Image menu, you will see a check mark on it.
2. Load the image that is going to be registered (the Slave Image) and the Mirone window transmutes into something like the example below.

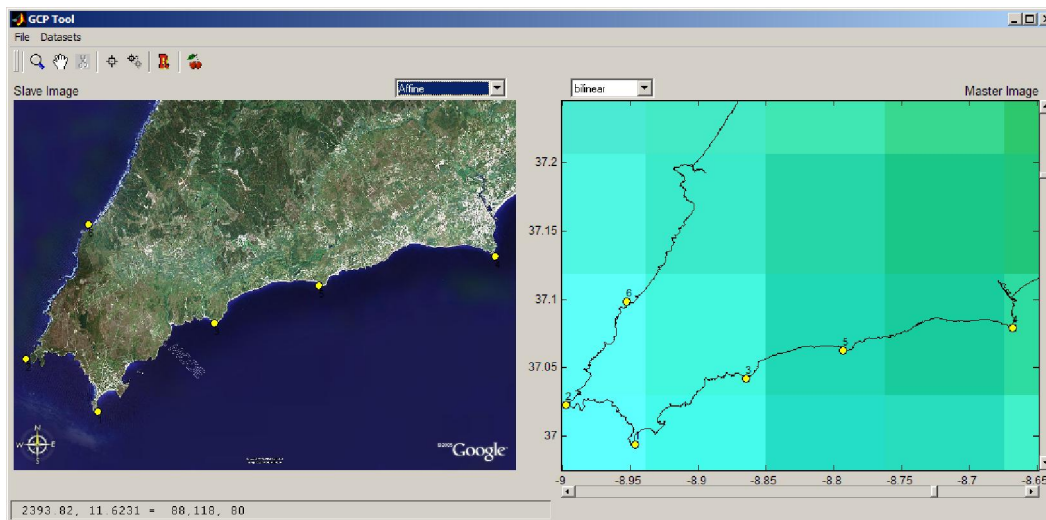


We are now going to zoom in on the Master Image, where we previously plotted a chunk of full definition coastlines and draw some ground control points (GCPs). Adding GCP is done on the following way:

1. Hit the button with the square in the cross icon (the fourth button) and the cursor will turn into a cross

2. Select distinctive common features on both the Slave and Master Images and click on the Slave Image (left image) followed by a click on Master Image (right image) over the same feature. This will add a pair of yellow dots, one on each image.
3. Repeat this process until you have a “reasonable” number of GCP pairs. To stop, double-click somewhere on one of the images

Next figure shows an example of what may look a referencing session



Right click over one of the yellow dots pops up a menu where we can select to show/hide the GCP pair numbers, delete a chosen pair of GCPs, to save them and, specially, to show a table with the points position and the error residue. Scroll this table to identify potentially bad located GCP pairs. If you have such a situation you can either kill the pair or simple adjust their position by double-clicking the specific point and click-and-drag to move.

When you are satisfied hit the **R** button, wait a bit and... that is it. You now have a georeferenced image.

To control the algorithm used in the image registration and how to interpolate the user can select between several options available in two popups located at the figure top center (which display Affine & bilinear in current example). Full documentation of those options is a bit long and interested people may consult it by reading the doc of MATLAB's *cp2tform* and *imtransform*, available for example at

www.mathworks.com/access/helpdesk/help/toolbox/images/cp2tform.html

www.mathworks.com/access/helpdesk/help/toolbox/images/imtransform.html

On version 1.4 it was added also the possibility to do the registration using a MEX of GDAL program *gdalwarp* (Matlab solutions, inexplicably, just fail too much). A final word to the “cherry button”. It's tooltip says 'Automatic Registration' and in fact that is what it tries to do. If you have two photos of the same area, one registered (for example an ortophoto) and the other not there are pretty good chances that this option automagically references your target image. The thing is done using the fantastic SIFT algorithm.

Draw GCPoints

An alternative method of registering images is when we know the coordinates of specific ground points that can be easily identified on an image such as house corners, cross roads, etc... In such cases the registering processes is achieved by plotting the GCPs on the image

and assigning the coordinates to them. This is what this option allows. When selected the cursor changes into a cross and every left-click adds one more vertex of a polyline. The inserting step ends with a right-click. Another right-click on the line or any of the inserted points displays a menu with these options:

1. Set reference Points
2. Show GCPs residuals
3. Register Image
4. Change registration method
5. Delete GCPs
6. Show GCPs numbers
7. Hide connection line
8. Show connection line

The next step is to assign the ground coordinates to points. Selecting option 1 opens a table like this one

	Unreg Points - X	Unreg Points - Y	Reg Points - X	Reg Points - Y
1	98	432		
2	155	285		
3	298	159		
4	452	152		
5	508	214		
6	555	350		
7	400	357		

The first two columns contain the image pixel coordinates (non-referenced coordinates). Columns 3 and 4 must now be filled with the georeferenced coordinates. Once that is done to all points we can check the goodness of the referencing process by using option 2 - *Show GCPs residuals*. Analyzing the residuals one can decide that some points need a position adjustment or even deletion. The first case is done in the normal way of double-click the target point to pass to edition mode and than click-and-drag. To delete a point, while in edition mode click on it one to make it the active point and that hit the keyboard "r" to remove that point. One satisfied use the *Register Image* to do it says.

The option 4 - *Change registration method* – opens a small window where one can select among the different warping methods used in the registration. See references at the end of [Activate Image-to-Image/Map GCP Tool](#) to get further information on those methods.



The meaning of points 5 to 8 is obvious and does not need any further explanation.

Plot in Memory GCPs

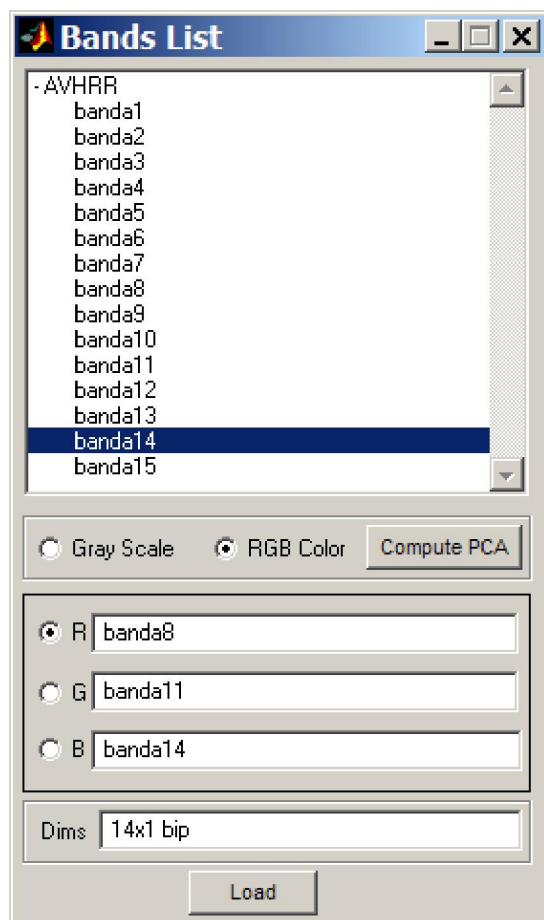
This option appears only when the loaded file is not georeferenced but has imbedded a list of GCPs. Note that are cases of files having both of them but in such situations this option doesn't show up. When we have that GCPs list, referencing the image works pretty much like described in [Register Image \(Draw GCP points\)](#) with the further simplification that the control points are already there. Note also that although it's possible to move the control points position, these eventual editing is ignored when referencing the image.

When the number of control points is greater than 10, a polynomial warping method is set as default. However, this might not always be the best option. It's up to you to decide/try.

Import GCPoints

Import ground control points from an ASCII file. This file must have 4 columns, of which the first two contain the image's column & row indices and the other 2 columns have the destination X & Y projected coordinates. After loading the file the *modus operandi* is the same as with the [Plot in Memory GCPs](#)

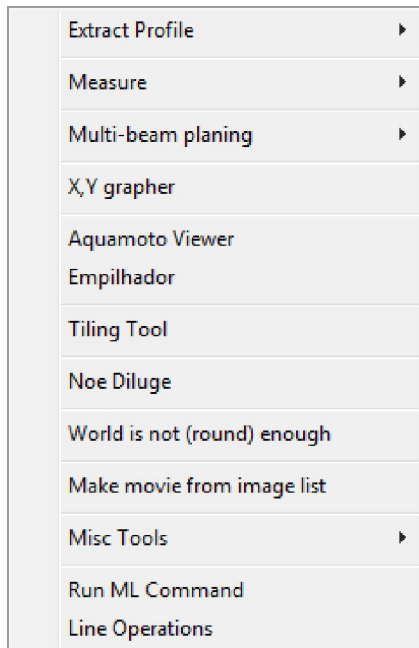
Load Bands



Satellite images are often composed of several bands taken at different spectral windows. If you have loaded one of those multi-band formats (on limit, the following is applicable to RGB images as well), you can choose to see only one band or to compose a false color image from the available bands. The figure below shows an example of a composition of bands 8-11-14 of an ENVISAT image. Note that the fact that a band shows on the *Bands List* figure does not imply that it loaded on memory. To save it, upon loading multi-band files only a few bands are stored in memory. As a consequence, for several band combinations it may be necessary to read them from disk, which takes a little longer.

The *Compute PCA* button computes the Principal Component Analysis decomposition of the current image. Interested readers should seek information about this methodology elsewhere (e.g. Google) or the DIPUM book from where this code was derived.

Tools



Housing of several generic tools that do not fit well in the other more thematic categories.

Extract Profile

Left clicks will plot a polyline, a right click remove the last entered point, double click to stop. As result a new window will pop up with an x,y plot containing the profile extracted from the grid along the drawn track. This corresponds to an interactive version of the *grdtrack* GMT program. The profile thus obtained can be saved to disk. Note that the x,y profile coordinates are obtained from linear interpolation and by consequence on geographic grids they may differ reasonably from the great circle arcs that reflect the shortest distance on a sphere between the end points. Note, this is crude and poorly controlled way of doing the profiling operation. The best way is to draw a line or polyline on the image. Calmly decide on its position and then inquire its properties (by a right-click on the rectangle borders) for "Extract Profile". A second mode to do the grid profiling is the "Dynamic" mode. Here, the profiling is done each time the user moves the mouse and the resulting curve is overlaid on the image. A double-click ends the dynamic profiling.

Measure

This is a measure tool that allows measuring distances (in units selected in **Preferences**), azimuths in degrees (cw from north) and area (in km²). Start by a left clicking and proceed with as many left clicks as you want. A double-left click adds the last point and shows up the results. Right clicks remove the previously entered point. The *Area per color* option displays the area occupied by each image intensity value in a table. This works only on indexed images since for RGB images there are just too many colors to perform this computation. The same information displayed in the table is also saved the sub-directory tmp of the Mirones installation dir, under the name of: *area_per_color.dat*.

Multi-beam planning

This tool is meant to help the planning of multi-beam surveys. Various systems have different swath widths coverage that depends on the number of beams of the particular system and the water depth. For simplicity the bottom coverage can be assumed to be a relation of the water depth. Hence, a "swath-width / water depth ratio" of 2 means that on a 4 km water column the swath will have a width of 8 km. The facilities provided are:

- Start a new track. In this mode left-clicks add new points; right-click or pressing the delete or back space remove previous points. Pressing RETURN, ENTER or middle button click ends the track and asks for a file name to save the track on disk.
- Imports a track from a (x,y) file on disk.
- Edit a track. In this mode the first left-click selects the track to be edited. The second selects the closest vertex that can now be moved and place into a new position by another left-click. However, while in the edit mode, pressing the "r", DELETE or BACKSPACE keys will remove the vertex; pressing "i" will insert a new vertex; pressing the "b" key will break the track in two. RETURN or left-click stops the edit mode.
- Delete a track. Use left-click to select the track and right-click to confirm.
- Save the track on disk. Use left-click to select the track and right-click to confirm.
- Compute the track length in Nautical Miles. Computations are made using great circle distances and spherical Earth approximation.

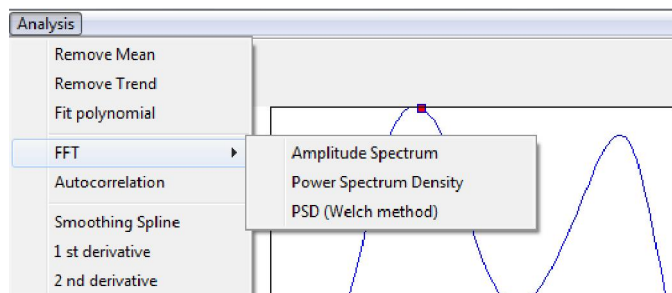
Region-Of-Interest

Use left mouse clicks to draw a closed polygon. A double left click ends the polygon drawing. Then you can select from a list of image processing operations that will be applied only inside the ROI polygon. Those operations work pretty well for grey images but regarding color images they may be very slow. That happens because the whole color image has to be transformed into an intensity space (YIQ in current case) and back to RGB, and those are slow operations and VERY memory demanding as far as Matlab is respected. So, even if you start with an indexed image you'll end up with a true color (RGB) image.

X,Y grapher

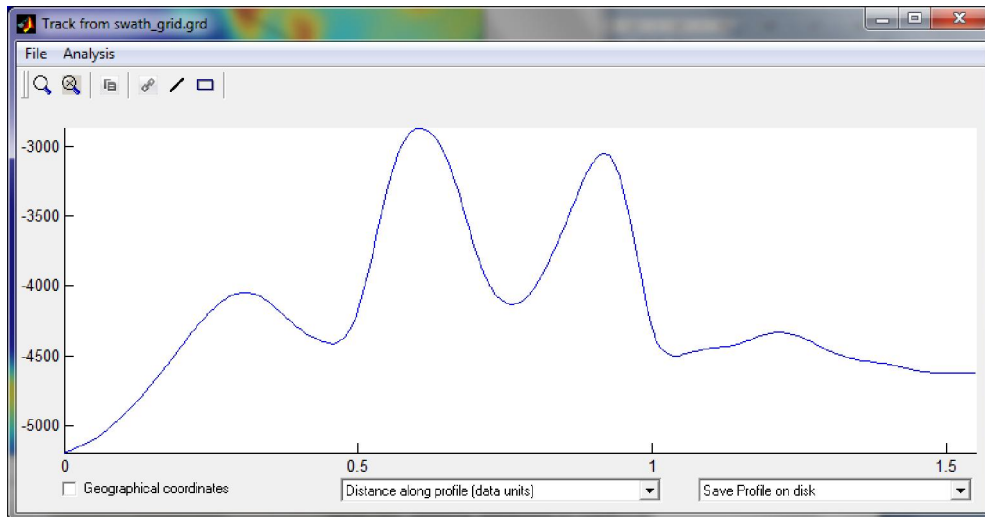
Calls the X,Y tool in a empty mode. Use this if you want to load one or several x,y files. For multi-column files an auxiliary window will popup to help selecting which columns to display. Otherwise this tool is automatically called for example when extracting a grid profile. When the coordinates are geographic and option check box allows plotting distances in Cartesian coordinates.

When displaying extracted profiles, the 'link' icon is very convenient to know where on grid is located a certain point on the profile. Just click the on the profile and a red marker will be plotted, as well as black dot on the corresponding grid position.

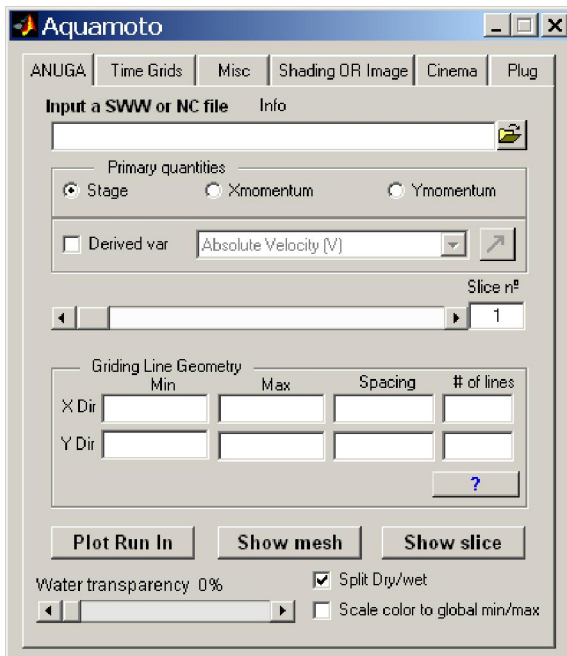


The icon with the line allows, by click-and-drag on the graph window, to calculate the slope dynamically. If it happens that the profile represents a magnetic anomaly, then an estimation of the depth to the source is carried on using the Spector & Grant method.

The icon with the rectangle (which is used to draw a rectangle) restricts the analysis to a limited X interval. This is useful when, for example after downloading tide gauge data, spectral analysis is performed. The “Analysis” menu offers several traditional methods to do quick time series processing as shown in the side image.



Aquamoto Viewer



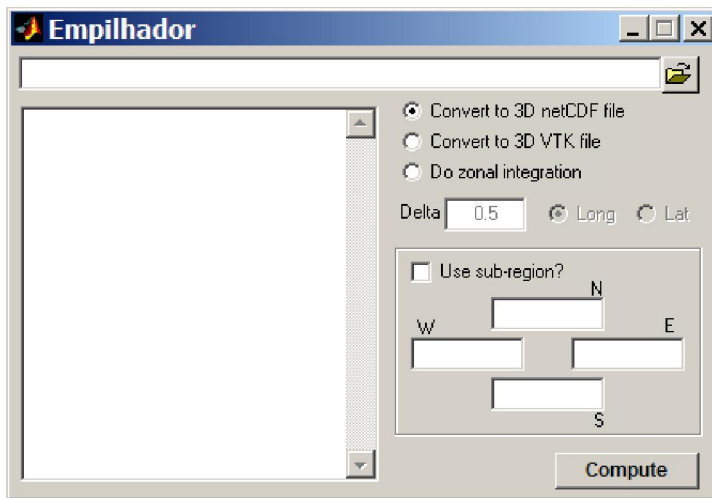
This is a large beast tool. On total it has a comparable number of code lines as Mirone itself. Originally its purpose was a specialized tool to visualize a particular netCDF format file produced by the Australian tsunami modeling software ANUGA, the .sww files. Since the .sww format is not in the form of a regular grid (it is a triangulation mesh) a linear interpolation is carried out in order to prepare data for viewing.

However, a more common usefulness of this tool is to use it as a generic netCDF viewer and manipulation tool for 3D netCDF grids. Viewing a particular slice in the grid is easily accomplished with the slider control. “Vertical” interpolations or movie maker are equally doable as well as all of Mirone options.

The last tab “Plug” displays one single large square pushbutton with the label “Run Plugin function”. That button calls the *aquaPlugin.m*

file that sits in the *utils* subdirectory. The intention is to execute specialized user defined routines. The current content of this file is directed towards satellite (SST and chlorophyll) computations.

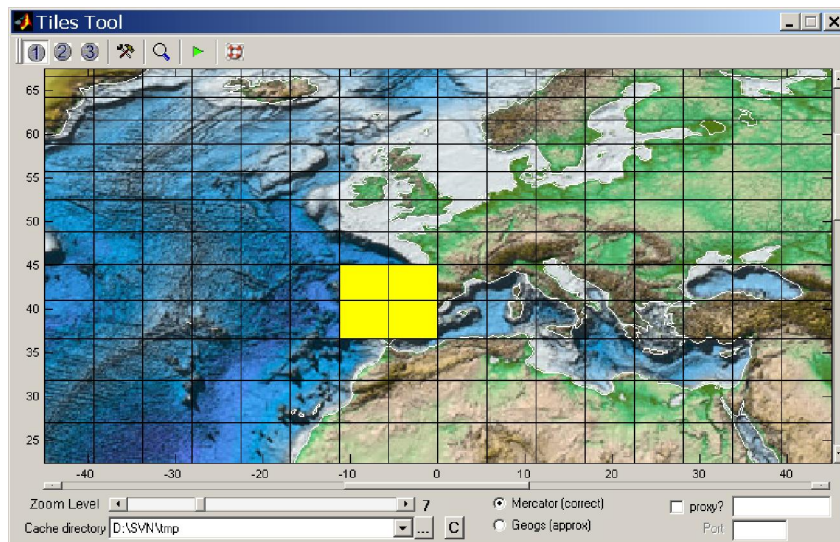
Empilhador



Use this tool to build a 3D netCDF grid from a pack of other grids. The list of grids is provided in the form of an ASCII file list that is loaded through the first edit box (or browse for it). HDF files with SST or Chlorophyll from NOAA satellites were the main targets of this tool, but one can use any other of the Mirone recognized formats. The 3D packed file can alternatively be save in one of VTK varieties and loded after on 3D viewers like the ParaView. The grids list file must contain the relative or full filename in the first column and, optionally,

a second column with a number that will be used as the layer (in the 3D file) number. If not provided, layers are numbered from 1 to n grids. If the first character on the grid list (1st column) is the '@' than the loading is paused with a message taken from whatever follows the @. Use this if you are composing from files located on DVDs. Example, the line "@ put cd 2" stops the file stacking and lets you change DVDs.

Tiling Tool

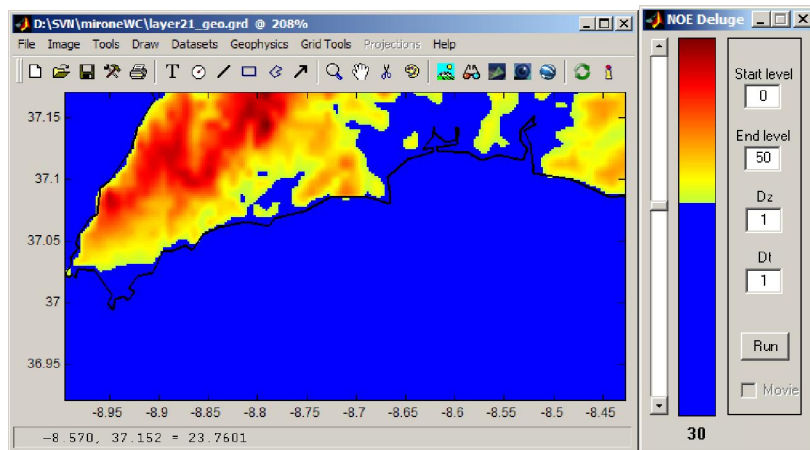


With this tool you can download and mosaic geo-referenced images from the Microsoft Virtual Earth servers. The figure presents you a more complete help where you can learn how to use a cache directory, take profit of an eventual World Wins installation cache, get image from Mars, etc... Google images could be retrieved as well but the guys from Google don't consider that legal.

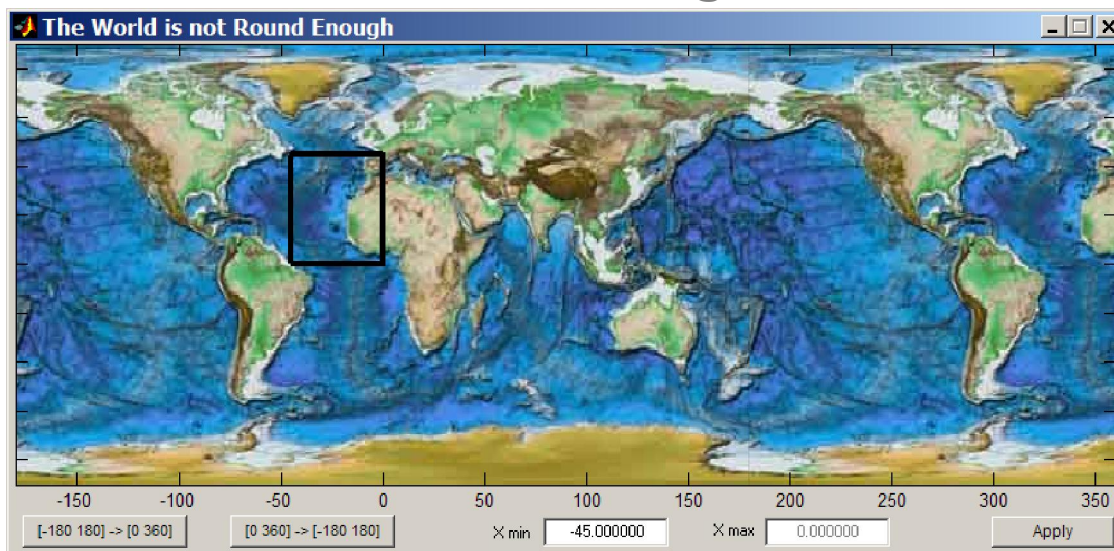
Noe deluge

The "Noe deluge" is a little utility to simulate the effects of the sea level variation. Certainly, if the grid data in use is coarse (and they often are) what you get is just a play (true for

other apps that this same thing). It only works if the grid in use has both positive and negative elevation in meters.



World is not (round) enough

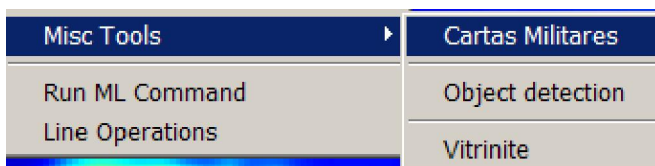


This is GUI interface to allow users to change between [-180 180] <--> [0 360] geographical longitudes. The rectangle is draggable and by moving it the user can translate the image longitude coordinates (for whatever use this might have).

Make movie from image list

An option to make movies in Mirone is to provide a text file containing the names of individual images that will be the movie frames. The file must contain the full name of each image, one per line. Common image formats (.jpg, .tif, .png, .gif) are accepted as input and the output is a movie in AVI format. Warning: obviously that all images must have exactly the same size, but that is not checked upon reading.

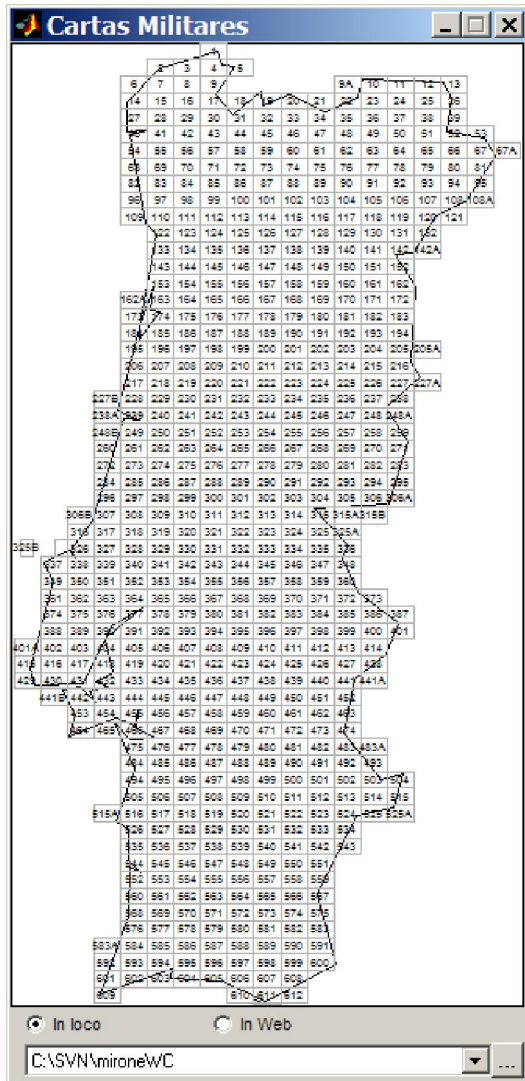
Misc Tools



Miscellaneous tools. This is intended as a repository of tools that do not

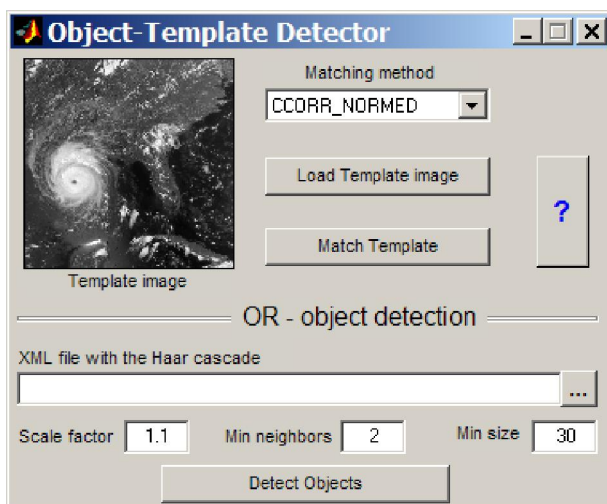
obviously fall in a category easy to group with others.

Cartas Militares



The *Cartas Militares* is a tool to load and automatically georeference the Portuguese raster maps with the same name. This tool can read the files from disk or download them from Internet (e.g. from the *Cartografia* section of the site www.mi4x4.com). No need to have (better, do not use) any .tfw file. The use is extremely easy, set the default directory (bottom edit box) to where the files reside and click on the selected quadrangle. If the map corresponding to the clicked rectangle is not found in the selected directory, a big red text warning is plotted during one second.

Object detection

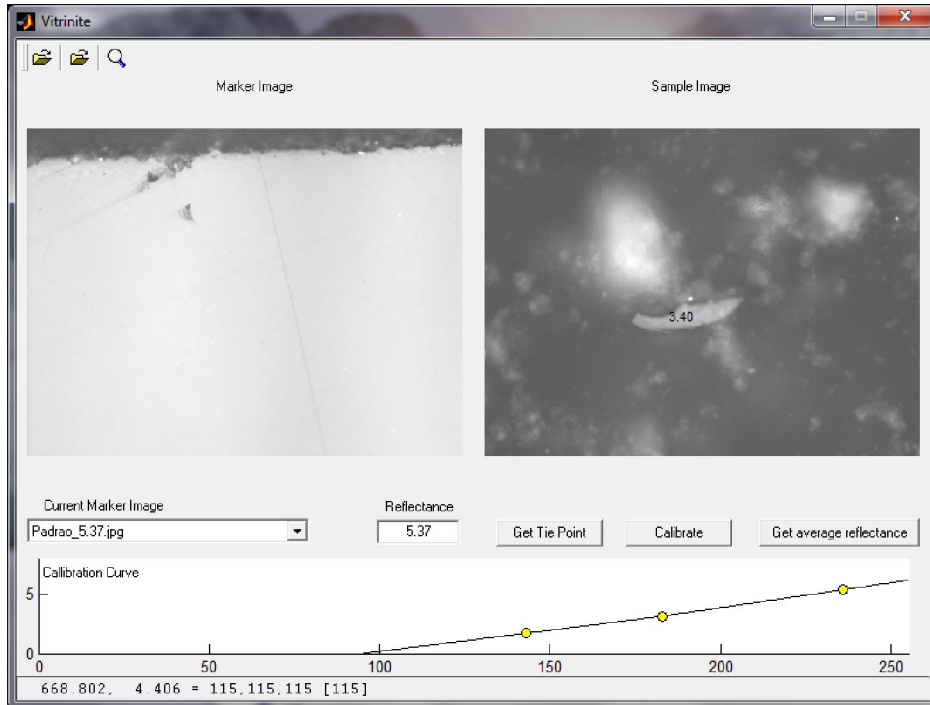


Object detection – Is a tool to perform object detection on an image. It can do the easier *Template matching* operation, where given an image it finds all places in the image that match a given subimage (also called the template). Typically, the template is much smaller than the image. The second half of the figure lets the user select the much far from trivial problem of detecting features using a trained Haar cascade. Mirone comes with no such file (they tend to be big) but interested users may get them

from the OpenCV distribution in its *examples* directory.

Vitrinite

Vitrinite – Is a tool that some geologists pay expensively on commercial products to get a relation between the color of small carbon incrustations and the age of the sample.



Run ML Command

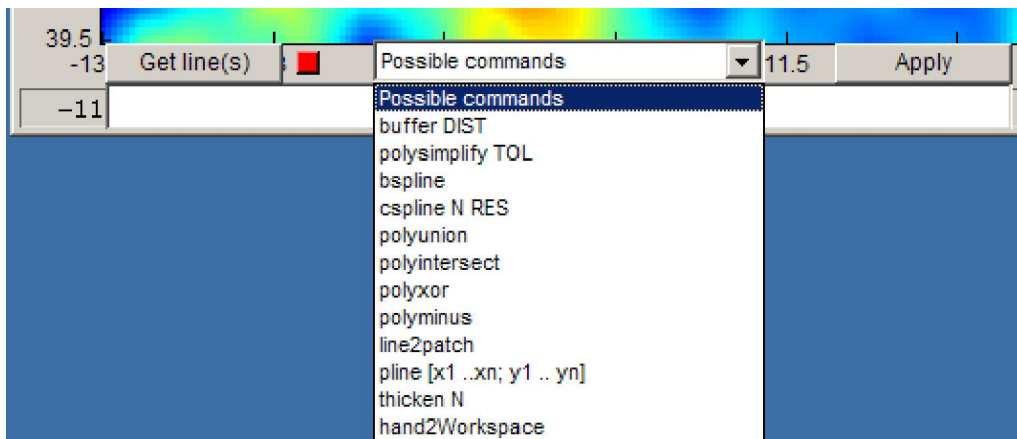


This is a tool where you can enter an arbitrarily complicated MATLAB single line expression that manipulates either the underlying grid (in case you loaded one) or the displayed image. NOTE, in case of the displayed image is indexed, it is converted to RGB prior to any manipulation. The only constrain is that you give a valid ML command.

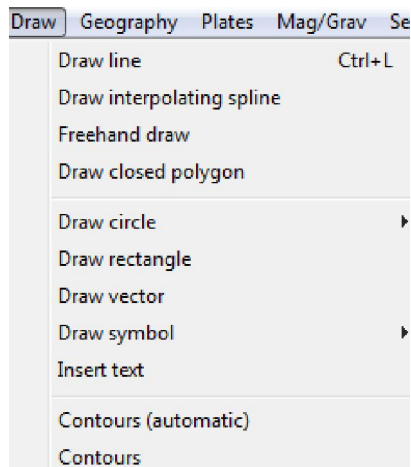
In case of error the error message won't probably be of much help. It doesn't work on the compiled version due to its inability to recognize a generic ML command.

Line Operations

This option allows you to perform an important number of specialized operations involving lines. After select the line(s) with mouse double-clicks one can compute buffer zones, Douglas-Peucker line simplification, polygon Boolean arithmetic's, B-spline and Cardinal-spline smoothing, etc ... NOTE, buffer calculus with geographical data can be done with geodetic ellipsoidal accuracy. To hide this option, just click on the "Line Operations" again.



Draw



This menu is where most mouse driven drawing tools are located.

BIG NOTE: All graphic elements drawn with the options in this menu are given many properties that include changing the line style, color, thickness, editing, deleting and so on. For accessing to these properties just right click on the selected element and choose what you want to do.

Draw line

Use left mouse clicks to draw the line (in fact a polyline) and stop with a double left click. You can use these line to extract a profile from the grid (if it is the case) where it was drawn.

Draw interpolating spline

Draw a smoothed #spline (cardinal). End with a double-click.

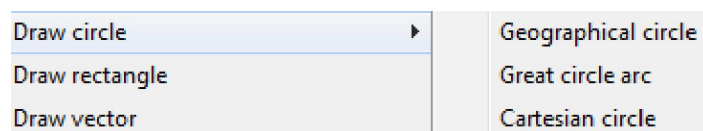
Freehand draw

Draws a line that follows the mouse movement. Start with a left-click and stop with a double-click. While drawing hit the "c" key to close the line.

Draw closed polygon

Do what it says. End with a double-click. Elements drawn with this tool have the extra properties of allowing color filling and assign transparency.

Draw circle



Draw circle lets you draw small circles in geographic coordinates (or plain circles with Cartesian coordinates). The three sub-options available are **Geographical Circle**

where the first left click defines the circle center and the second a point in the perimeter. Meanwhile, moving the mouse draws the circle on the fly. The second option draws a **Great circle arc**. The third option draws a circle but in Cartesian coordinates.

Draw rectangle

Works pretty much in the same way as the circle's case, except that the given properties are different. You can now crop images and/or grids, change/move the rectangle and so on. Relatively hidden in these properties is the possibility of using the rectangle coordinates to register an image. This is more correct and gives more freedom to the user than the [Map Limits](#) option. The rectangle corner coordinates are used to compute an affine transformation that registers the image to the rectangle's coordinate system. To compute an affine transformation we need at least 3 points. With the rectangle we have 4, which is ok but imposes the unnecessary (to the affine transformation) restriction that the points lay on a rectangle. In fact, if you edit and change the rectangle to another shape, the registration option will no longer be available. From the rectangle properties you can also select the **Transplant Image here**. This non-finished tool allows selecting a second image that will be inserted inside the area enclosed by the rectangle. In its current state, the second image is always reinterpolated to fit the space (in terms of rows and columns) inside the rectangle. The **crop tools** options provide useful options that apply only to the grid elements contained in the rectangle. I wrote the *spline smooth, median filter and fill gaps* having in mind post processing of swath bathymetry grids, mainly to disguise the region of overlapping swaths. While this works pretty well (you may even work on shaded illuminated images) it has one drawback. It may leave a too clear suture zone between the processed and non-processed zones. This option has an experimental undo.

Draw vector

Draws a vector by the same clicking sequence as above. However, this cannot be edited (but deformed yes), only deleted.

Draw symbol

Filled circles, squares, triangles and stars are plotted by a left click. Right clicks on these elements lets you change size, fill and border color, select other symbols ...

Insert text

Left click for the point of text insertion and then write whatever you want. Left click (do not hit ENTER unless you want to change line) outside the text box to end. If you have previous line elements you will notice that they vanished when you started to write your text. Don't worry, it's just an innocuous Matlab BUG (it has many). To get back all your stuff click on the refresh button (the one with two curved arrows) of the Mirone button collection. Text may be edited and moved or change the font, size, color by accessing its properties (again, right click on it).

Contours (automatic)

For grids, this automatically calculates and draws a set of contour lines (see [Grid Tools -> contours](#))

Contours

Raise an option window where you can select which contours to draw. An example image is shown in the more in context for this tool entry [Grid Tools -> contours](#).

Geography

Geography	Plates	Mag/Grav	Seismol
Plot coastline			▶
Plot political boundaries			▶
Plot rivers			▶
Global seismicity (1990-2009)			
Hotspot locations			
Magnetic isochrons			
Volcanoes			
Meteorite impacts			
Hydrothermal sites			
Tide Stations			
Tides (download)			
Plate boundaries			
Cities			▶
DSDP/ODP/IODP sites			▶
Atlas			

Here we can plot many different quantities of relevant interest for Earth Sciences. Nearly all options display data that comes with the Mirone installation and load from files located in the \$MIRONEROOT/data sub-directory. The exceptions are the coastlines files, which are provided by GMT. That is, it is necessary to have GMT installed and the detection is done automatically (but see also [Installation](#) for instructions on how to do a manual install of those files)

Plot coastline, political boundaries, rivers

Use the netCDF GMT database. Depending on what you installed in GMT so you will access to the crude, low, intermediate or full resolution files. Drawing from this data will only work with geographic or for grid/images with a known projection definition. If you didn't install GMT and therefore these options appear grayed out, see instructions on [Installation](#) on how to solve this situation.

Global seismicity

Select earthquakes with magnitudes ≥ 5 for the period 1990-2009. The Seismicity was downloaded from the USGS web site.

Hotspot locations

For those of you that are still Hotspotist this option plots the (B. Steinberger, JGR, v105, p11127--11152, 2000) Hotspot locations list. Right clicking lets you ask for the ghost name.

Magnetic isochrons

Plots the Muller et. al. Digital Isochrons of the World's Ocean Floor (on the Açores area some by much better ones). Right clicking on individual isochrons provides a short info about it.

Volcanoes

Plots the Smithsonian volcanic activity list. Again, right clicking on an event gives access to a short description of it.

Meteorite impacts

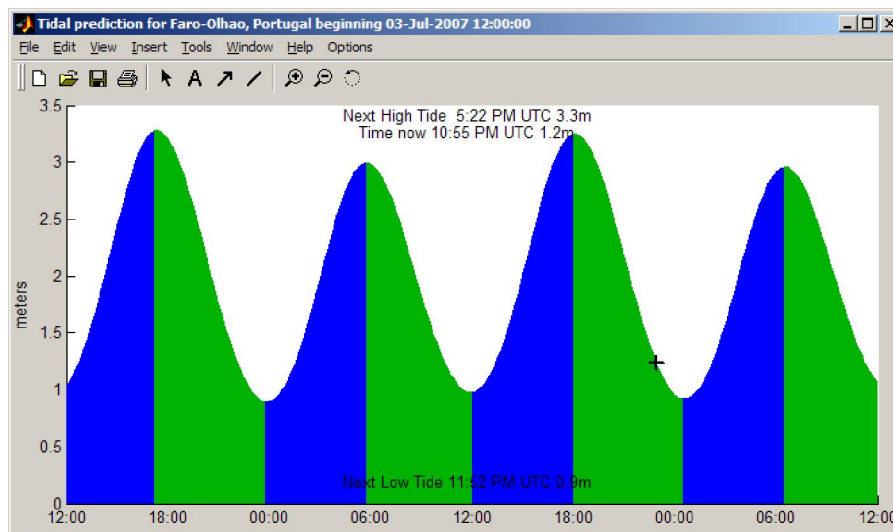
Plot the site locations of acknowledged meteorite impacts.

Hydrothermal sites

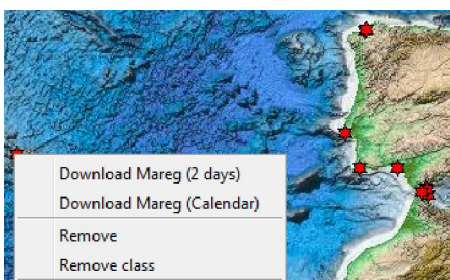
Plot the locations of known hydrothermal activity.

Tide Stations

This will plot the tidal stations of the `t_xtide.mat` file that resides in the "data" directory. Once the tidal stations are plotted in your base figure, right clicking gives access to either the station info (name, location, n° of harmonic coefficients, etc) or to the "Plot Tides" option. This will open a new window showing the tides for a period of two days more or less centered in the current date. This window mimics the original *WXtide* program but does not offer all the possibilities of the original program (no tide calendar, etc). The `t_xtide.mat` file was converted from the harmonics file contained in *WXtide* (version 2.6) using the `t_xtide.m` code of the "tides" toolbox (Pawlowicz et al., *Computers and Geosciences*, 28 (2002), 929-937). In fact most of the computing code of this tool was adapted from `t_xtide`. Users that interested on knowing more about the harmonics used in this program are directed to the *WXtide* (just search for it on the web) help files. You will also find there instructions on how to compute new harmonics files (if you have the data for doing it). *WXtide* comes with more harmonics files. In case that you are interested to use one of them instead of the one provided here, use "Harmonics to mat file" option to generate a new `t_xtide.mat` and replace it by the original one.



Tides (download)



Attempts to download tide gauge data from a list of map locations. The tools starts by plotting symbols for the locations it knows (stored in the `../data/mareg_online.dat` file). Right-click on a symbol to bring up a menu where user can choose to download the data from the last 2 days or do a selection based on a certain date. If there are no download errors the

result will be shown automatically in the X,Y Grapher tool.

Plate Boundaries

Draws the P. Bird (element.ess.ucla.edu) plate boundary list. Different boundary types are plotted with different colors. Right clicking on a boundary gives access to the boundary type, neighboring plates and velocity across that boundary. I slightly changed the original file for the Açores Triple Junction region.

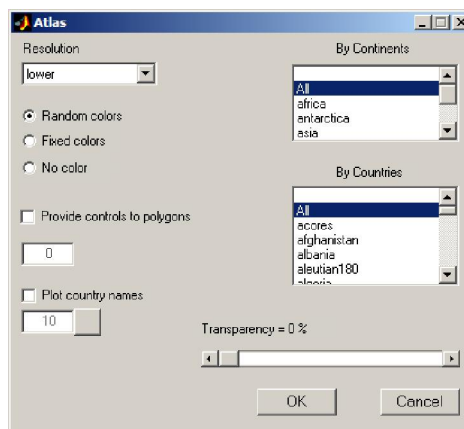
Cities

Selects the *Major cities* or *Other cities* to plot city places and names. I recovered these files from the iGMT distribution.

DSDP/ODP/IODP sites

Plots the DSDP, ODP or the IODP (or all three of them) sites. Right clicking on a site lets you have access to minimalist site information.

Atlas

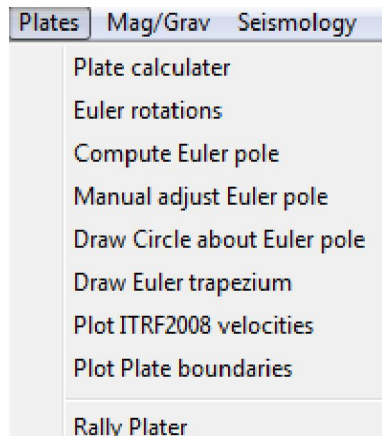


This tool is at a rudimentary stage but it works ok. The GMT coastlines are built in such a way that they don't allow plotting individual countries. I built up a new (simple) database made by countries defined individually and wrote a C program to read it. You may select a country or a whole continent. If a base Mirone window exists than the country as a patch is added to it, otherwise a new area is created to receive that country. Now countries may be attributed properties accessible by a right-click. However, I found out another very unpleasant MATLAB feature. Attributing properties using uicontexts (that's the technical description) is VERY VERY memory consuming. To give you an idea the whole world at the maximum resolution swallowed me more

than 500 Mb of RAM. Meanwhile I reduced the number of uicontexts but it still consumes about 200 Mb for the whole world (can we believe that?). And more, killing the figure may take more than 30 seconds (these ML guys...). So, think twice before checking the "Provide controls to polygons" option if you are loading many countries (but for a small number you do not notice this). Bonus, the "Write GMT script" option knows how to deal with these data a will make a nice GMT ps figure.

If you select this from a Mirone window with a previously loaded grid/image, country plots will be clipped to the region delimited by the image with an automatic map projection if necessary. If, however, you call it from within an empty Mirone figure, data BoundingBox is used to create a background region image to hold the selected plotting countries.

Plates

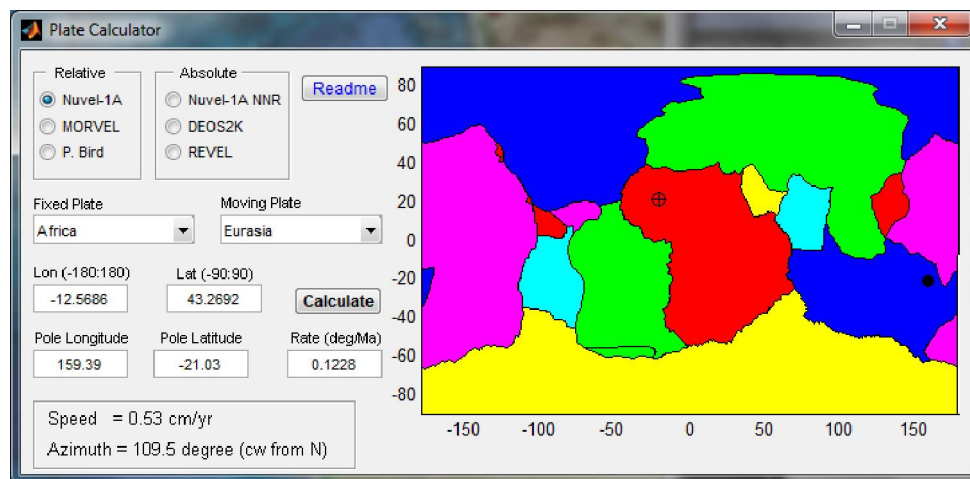


used in Plate kinematics

This menu host the tools

Plate Calculator

Opens a window tool from where you can calculate plate velocities according to the available models (e.g. NUVEL1A, MORVEL, REVEL, DEOS2K). It has a help of its own but its use is as simple as clicking on the image.



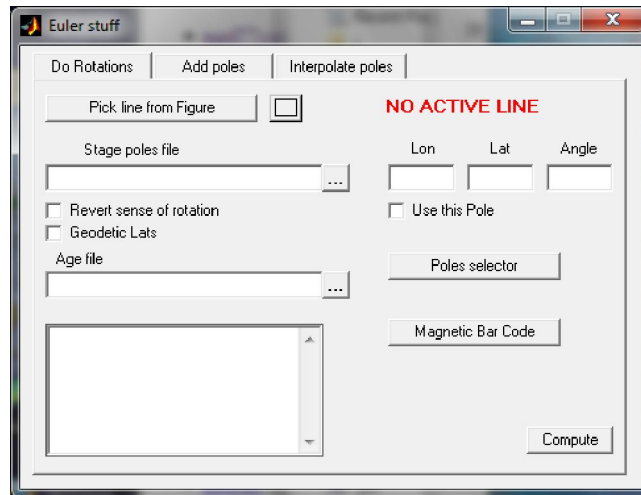
Euler rotations

Rotating magnetic isochrones or creating flow lines is very easily accomplished using this tool. Any line element, either drawn or imported from an ascii file, or point symbol may be rotated using either a stage pole file like the ones described in the [Rally Plater](#) tool or using a finite rotation pole. For selecting a line or point do a single left-click on it. The selected element will than show red stars in its vertices. Next, right-click on any free zone of the image to confirm the selection (the red stars will disappear). When used to do rotations with stage poles it is also necessary to provide an age file which determines the ages at which the rotation is computed. For each entry of that age file (displayed in the window's listbox) a new rotated line is plotted in Mirone main window. The "Magnetic Bar Code" button provides an easy way to select an age sequence using the geomagnetic polarity scale as reference. For using it one has to double-click over the period bars. A marker, as well as

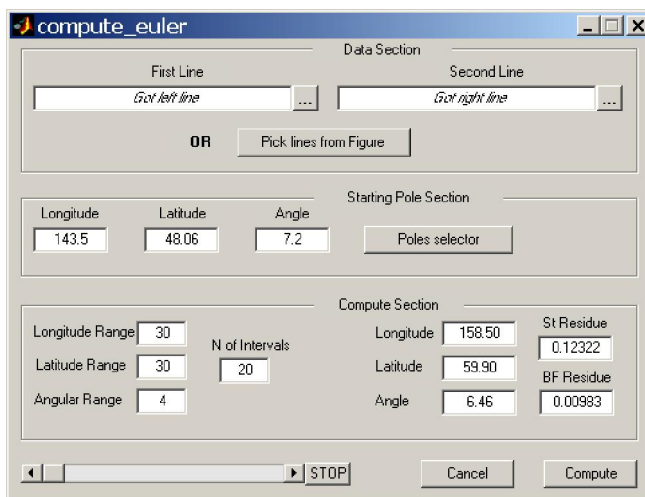
new window with its age will show up. Insert as many age markers as wanted and drag them to fine adjust the wanted time.

If a single point was selected (using the "Pick line from figure" button) instead of a line/polyline, than a single flow line is plotted. In the "Add poles" tab the user may compute the pole that results from a combination of two finite rotations.

With this tool you can also add or interpolate poles. Pay attention to the tooltips.



Compute Euler pole



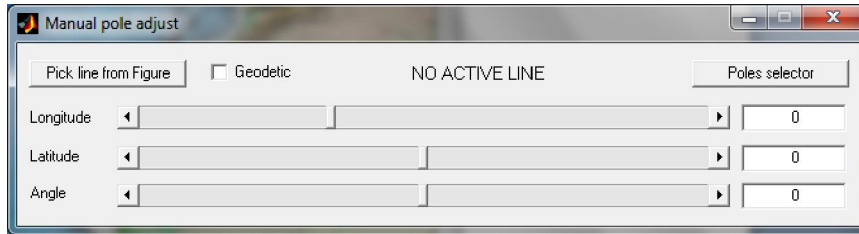
Users that want to compute their own Euler poles can do it with this tool. For using it all is needed are two lines in the main window, which will be mouse selected after hitting the "Pick lines from figure" button, and an initial estimate of the Euler pole. For selecting the lines follow the same procedure as described above for Euler rotations. That is: left-click to select and right-click to accept – once for each line. The results of the searching algorithm are graphically displayed every time a better fit pole is found. If at the end of the process the user is not fully satisfied with the result, that entire he needs is to restart again

using this time a better initial guess of the starting pole.

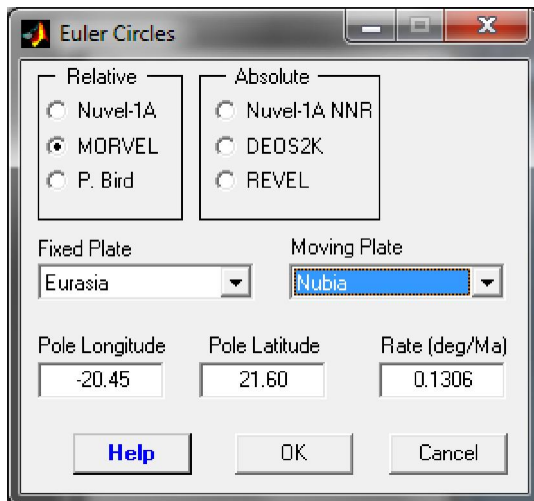
Starting on version 2.0 the computation of the Euler poles is done by converting first the latitudes from geodetic to geocentric. The latitude of the pole location is reported in geodetic latitudes. In previous versions the calculus was performed without doing any latitude conversion. The result from both ways of computing the pole are normally very similar, except for large angles and when the north-south component of the movement is important. But the most important of all is to not mix rotation carried with geocentric latitudes and poles calculated in geodetic lats, or *vice-versa*.

Manual adjust Euler pole

If you want to do either a free rotation of a line or try to fine tuning one pole, you can use this tool. Again, you select a line, enter the pole parameters and move the sliders. The line is rotated accordingly. Try it and find out how is far from trivial to fit two lines.



Draw Circle about Euler pole



Lets you select the circle's center from a relative (or absolute) Euler pole, selectable from the popup menus in the window like the figure below. Once you made your pole choice, the center is already defined and you only have to give another left click to select a point on the perimeter. Now, if you want to change the circle's radius or move it double-click on the circle. A red point shows the center (if visible on the figure). Click and drag to move the circle. The radius can be changed by click and dragging the red circle on the perimeter. These and other settings may be also done in the parameter window that has meanwhile shown up. Double clicking again removes the circle's editing properties. If the circle is of the **Circle about an Euler pole** type,

right clicking lets you choose to calculate the Euler velocity on the point where the red circle is displayed.

Draw Euler trapezium

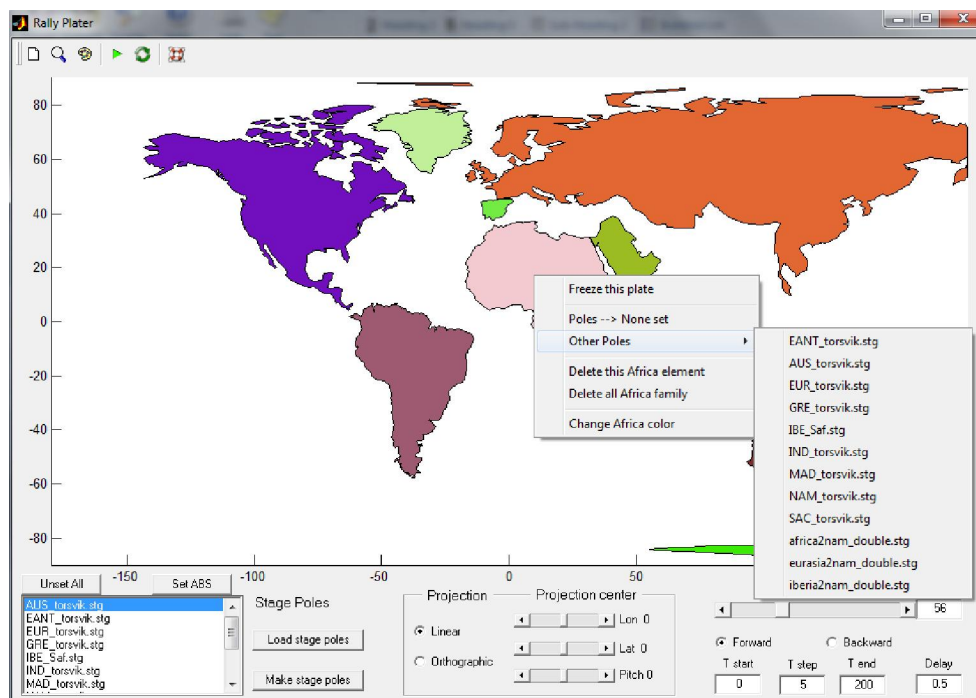
This a special tool to fast compute an Euler pole using the so called Bonin method. It works in the following way. You draw a trapezium (in much the same way as you draw a closed polygon). Then right click on it and select "Compute Euler Pole", the result will appear either on the ML command window or in the DOS window if you are running the stand-alone version. Now what it does. First and second points define side **A**, third and fourth the side **B**. The side **A** is rotated to fit over side **B**. This involves two separate rotations. The first rotation brings points 1 and 4 together. A second rotation is than performed, about the now coinciding points 1 and 4, by the angle defined by the first line segment (the one that originally made up by points 1 and 2) AFTER the first rotation and the second line segment (original points 3 and 4). The result is the combination of the two rotations. Though it is a crude method, it sometimes gives very good results. If you don't know what I'm talking about, just forget this option. If, on the other hand you do, you can thank me.

Plot ITRF2008 velocities

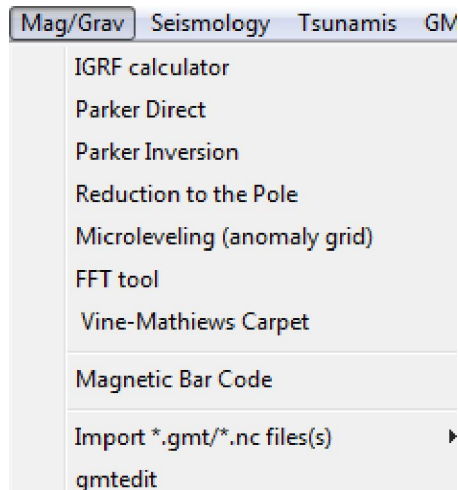
Plot an arrow field with the velocities of the ITR2008. The data used in this option was downloaded from the ITRF home page, reformatted, and is stored at \$IRONEROOT\data\itrf208.dat

Rally Plater

This tool lays somewhere between a Plate reconstruction demo and something that can be used to do real work. There are some dedicated software available on the web that do plate reconstructions but this one has a big advantage. It is ready to work upon start, and is very easy to use. There is no need to explain what the buttons do. They self explain with their tooltips. To access other fundamental options right-click over the "plates". New menus popup and is there that the user controls how the plates move. By default, the program associates a "movement history" to three plates: Iberia, Eurasia and Africa. The poles that show up in the window's lower left corner define the movement. In addition to the default files, the user can load its own data. There are two types of data: the poles files, which control how the plates move; and the plate files. The poles used in this program are stage poles (not finite rotation poles). They have a particular format described in detail in the *rotconverter* GMT program, but the example files can be found at them at the "continents" directory of Mirone's installation. To create new stage poles files from finite rotations poles hit the "Make stage poles" and follow instructions there. The plates files are simple ASCII files with two columns: longitude and latitude (in that exact order). Now, those files may be multi-segments, with the character ">" serving as the separator. The individual segments may be closed (equal first and last point). If they are closed, the contour is filled; otherwise they are drawn as lines. Very important: The first line in file must be of the form "# Name" (without the quotes), where Name is the plate's name. This is the mechanism that the program uses to know how to associate a plate and its companion lines with a stage pole. We can see the default plates name when right-clicking over it, so if we want to add it another element (e.g. the 3000 m bathymetric) we must give in that file the exact same name as the plate we want it to join.

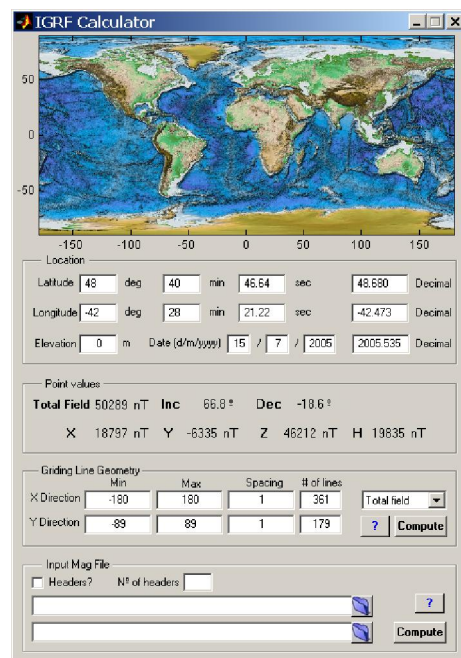


Mag/Grav



This menu contains tools to do potential methods (magnetic and gravimetric) processing.

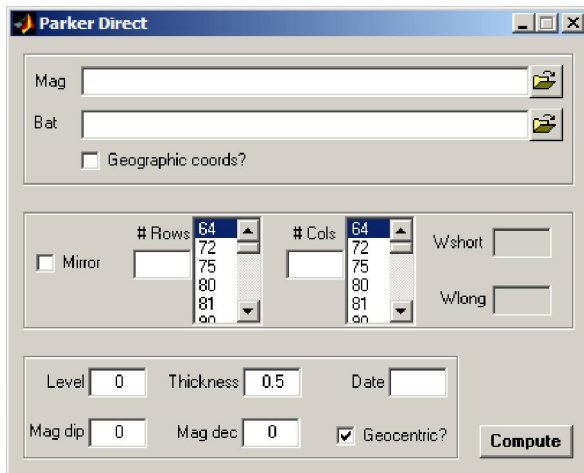
IGRF calculator



Calls the IGRF calculator tool where you can compute the value of the total magnetic field (or its components) on a given point, generate a grid of regularly spaced magnetic quantities or read a file with at least lat, long positions. When used to import a file, a new window pops up where you select which data column means what. You have here the possibility to choose to compute the magnetic anomaly (measured total field - IGRF magnitude). Computing magnetic anomaly requires knowing the date of measuring. The reading file function has the ability to decode many date formats (that's why it may be a bit slow to read large files) including text formats. Hit the "?" button for help in text date formats. The IGRF model used is the 10th generation IGRF as agreed in December 2004 by IAGA Working Group V-MOD. It is valid 1900.0 to 2015.0 inclusive. Values for dates from 1945.0 to 2000.0 inclusive are definitive, otherwise they are non-definitive.

Parker direct, Parker inversion and Reduction to the Pole

Those are standard procedures used in magnetic field modeling. It is out of scope of this document to explain the principles behind those methods. Either you know them and then the use of these modules should be pretty straightforward or you don't. If you do not know



and want to learn, I suggest that you start by reading the Blakely textbook. The credit to these modules goes mainly to Maurice Tivey for I used functions from his magnetic toolbox as backbones to write them.

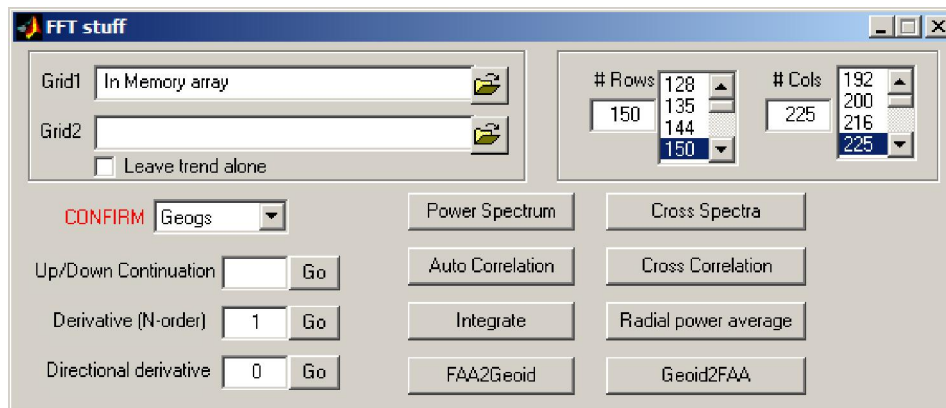
Microleveling (anomaly grid)

This is specialized tool that tries to do a very chirurgical smoothing on marine magnetic anomaly grids where individual ship tracks stand out too clearly. Interested users should consult this pictorial example on how to use this tool:

sites.google.com/site/mironehowtos/geophysics/micro-levelling-magnetic-data-grid

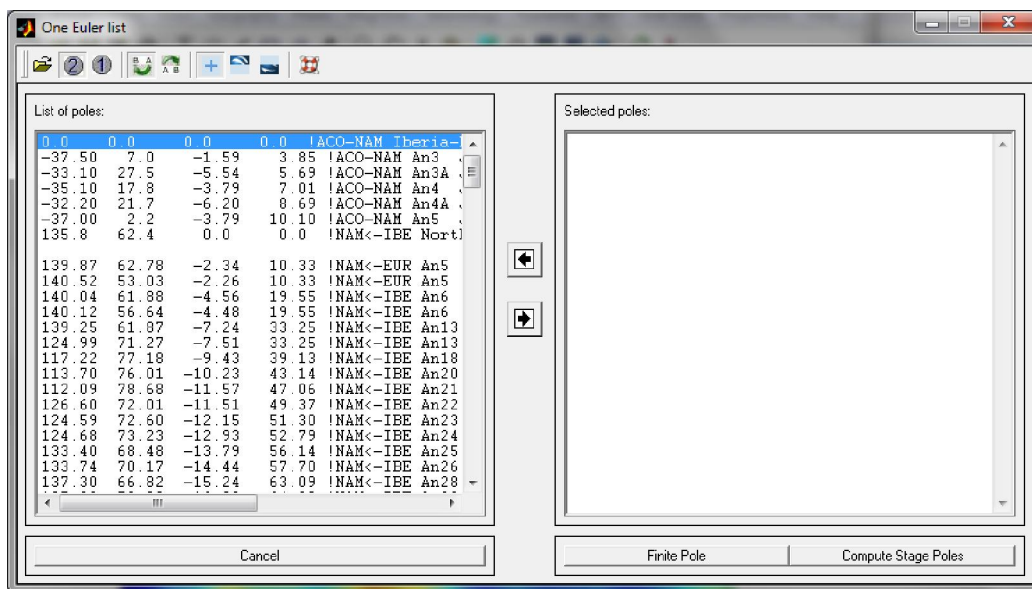
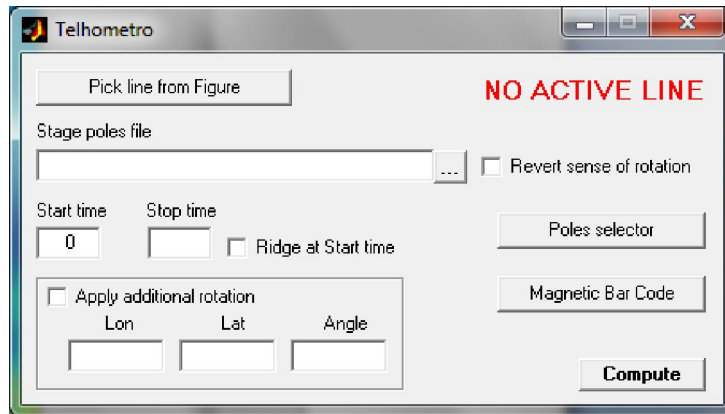
FFT tool

This tool reproduces most of the functionalities of the grdffft GMT program (except filtering and the chaining of operations) and adds it some others. Namely the Cross Spectra and Cross Correlation. It is, however, entirely programmed in Matlab. That means double precision computations and potentially very large memory consumption.

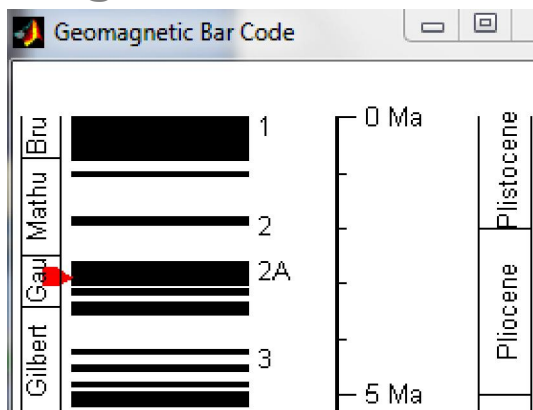


Vine-Mathiews Carpet

This tool plots stripes of red and blue patches that represent periods of direct and reverse polarity of the geomagnetic field. To work with it, it is necessary to calculate a collection of stage poles that describe relative movement between two plates. A large number of finite rotation poles are already shipped with Mirone. Hitting the "Poles selector" button calls a new window to assist the stage poles computation.



Magnetic Bar Code



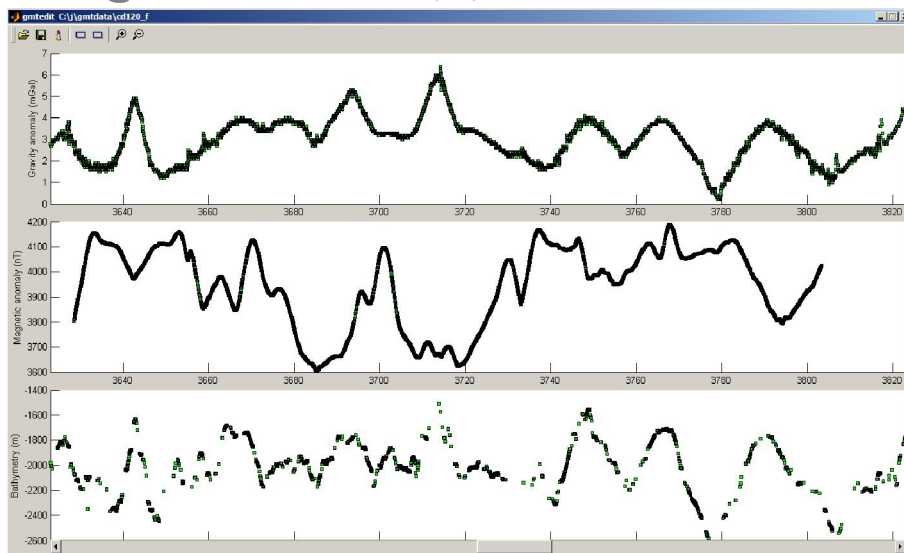
This option opens a window with the Cand & Kent (1995) Magnetic inversion time scale. Double clicking over the black or white bar creates a red marker and a secondary window with the age corresponding to the position of the red marker. This marker can be click-and-dragged and age updates automatically. You can draw as many markers as you want. A second double click on the marker deletes it.

gmtedit

gmtedit was an old GMT program from the time of version 2 that used to run only in Sun stations with the openview OS. Most of you don't even know what I'm talking about. However, it was a very useful program to view and edit the MGD77+ netCDF and the older .gmt files created and manipulated by the GMT supplements *MGD77* and *mgg*. If you still don't know what I'm referring to probably the best is to jump ahead. Well because I needed it I decided to resuscitate *gmtedit*. I don't show a picture of it here because it occupies the whole screen. Its use is very easy. Load a MGD77+ or a .gmt file and click on the points that you decide are outliers. They will turn red (bad points). Another click will make them green (good points) again. The first rectangle icon allows doing this selection by area. The second only applies to the magnetic channel and permits to move the points by dragging the rectangle. At the end save the edited file. Red marked points will be replaced by the no-data-value.

Since the new MGD77+ format allows storing of a larger (and variable, due to its netCDF nature) number of variables, the *gmtedit* program has a mechanism to allow controlling which variable is plotted in each axes (there can be more than one). Interested users should consult the "OPTcontrol.txt" file located in \$MIRONEROOT\data directory and follow instructions explained there.

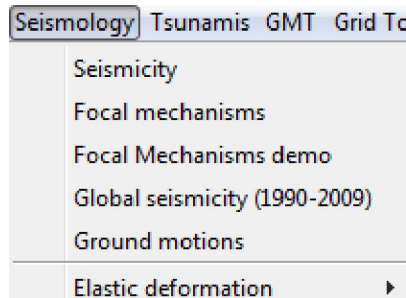
Import *.gmt/.nc file(s)



A .gmt file is handy format to store the Gravity, Magnetics, Topography data contained in the mgd77 exchange format file provided by NGDC. Read the GMT mgg supplement manuals to learn more about the format and how to create .gmt files. This format, however, has several limitations and is supersede by the new MGD77+ netCDF format. Both formats are readable by the gmtedit tool.

These options let you import either one MGD77+ (or gmt) file or a list of files. The list of files is a simple ASCII file with the full path of each MGD77+ file, one per line. Once read, the files appear as tracks in the figure. Right-clicking gives the line info as provided by the GMT program *gmtinfo*. And more, one may select to view the file with gmtedit. And even more, the chunk of data initially displayed in gmtedit corresponds closely to the geographical position where the mouse clicking was executed. Isn't that fantastic?

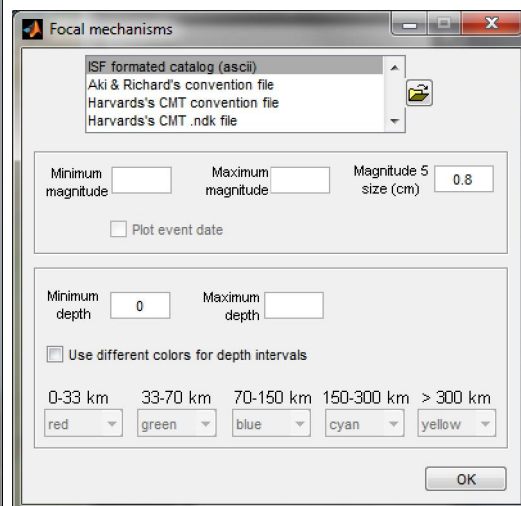
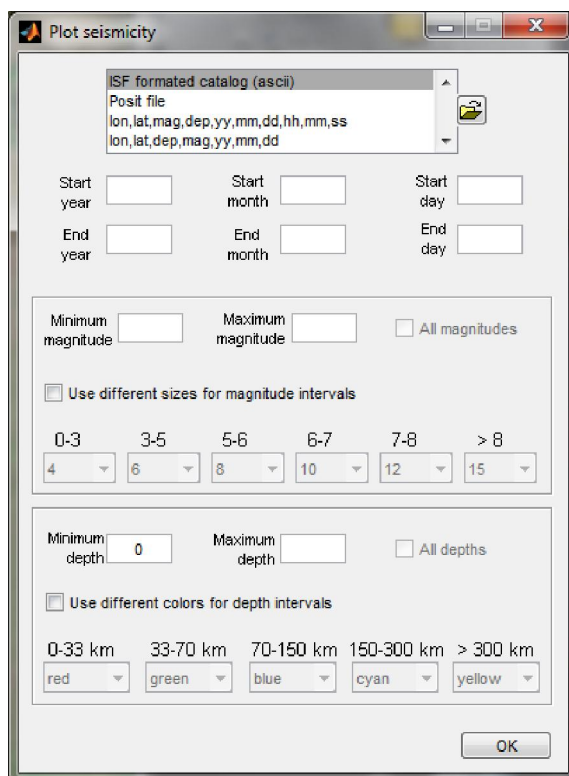
Seismology



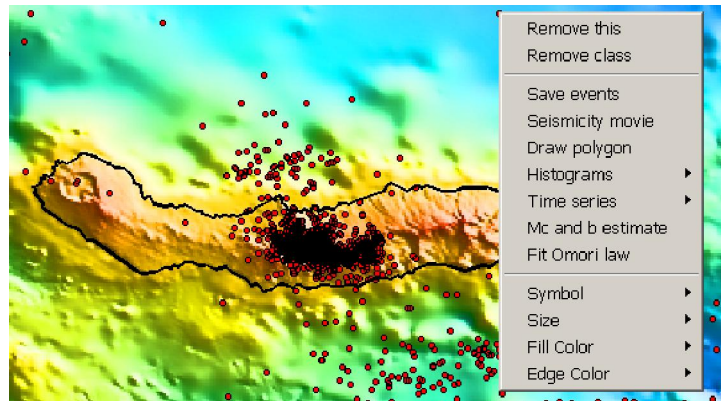
This menu contains tools to display earthquake related data and to perform elastic deformation calculations (use for example in the Tsunami modeling).

Seismicity and focal mechanisms

The figures below show the two interfaces to plot either epicenters locations or draw focal mechanisms (the seismologist's beach balls). To plot seismicity use either an ASCII file with data in lines organized as: lon, lat, depth, magnitude, year, month, day, [hour, minute, second]. As an alternative one may use a ISF formatted catalog (like the ones we can get from www.isc.ac.uk). Reading the ISF format is done with a MEX file. A "Posit file" reads the format used by PMEL to divulgate the hydrophone detected seismicity. Once you have loaded a seismicity file, there are many operations available through a right-click. Namely, play a seismic movie (a separate window controls the "movie" parameters), display several types of histograms and fit a Modified Omori Law. This later as well as the Mc and b value estimate was taken from parts of code of the Zmap toolbox. A finer refinement is achieved when we select to "Draw a polygon" around an interest area. Then the same options are available but will apply only to the data contained inside the polygon. It contains also a tool to detect swarms. That is a very basic tool that uses a homemade criterion of space and time distances to detect the events connectivity.

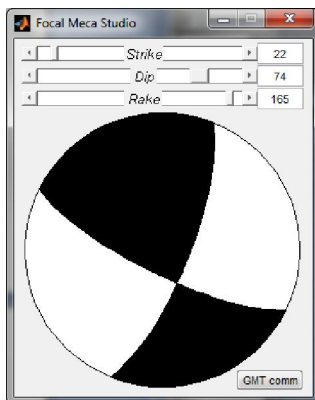


To plot focal mechanisms one can use the formats shown in the list. In the Aki & Richard's convention the data is ordered as: lon lat depth strike dip rake (in degrees) magnitude. Optionally add two more columns with the lon and lat at which to place the beach ball. The Harvard's CMT convention uses: lon lat depth str1 dip1 rake1 str2 dip2 rake2 mantissa and exponent (of moment in N-m). Again, optionally add two more columns with the lon and lat at which to place the beach ball. WARNING, contrary to *psmecca*, the moment magnitude uses N-m (and not Dyn-cm). Also, it is not yet possible here to have a last string column containing the event label. However, the ISF formatted files uses the event date to build up an event label.



A big advantage of using the focal mechanism tool over using *psmecca* is that the user may easily displace the beach balls using the mouse. This is particularly useful when we have many events on top of each other. Also, the [Save GMT script](#) tool is able to detect beach balls and to make the appropriate calls to *psmecca* that reproduce the graphical display.

Focal Mechanisms demo



With this tool the user can see instantaneously how the parameters of the Aki & Richards convention translate into a focal mechanism. The use is extremely simple, change the values with the help of the sliders and see the result right away. The button at the lower right corner will display the equivalent *psmecca* command that can be used to reproduce the figure's content in GMT.

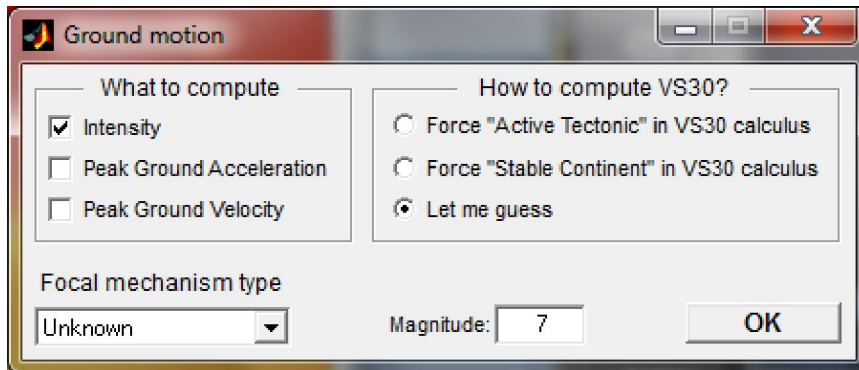
Global seismicity

Calls the same tool as in [Seismicity](#) above but with a default seismicity file with events in the 1990-2009 period and magnitude ≥ 5 .

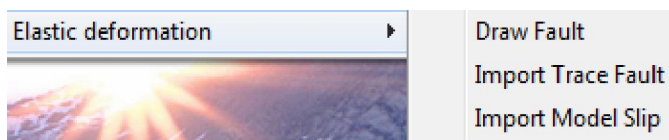
Ground motions

This tool will estimate the Intensity, Peak Ground Acceleration or Peak Ground Velocity from an earthquake source. In order to use this tool it is necessary to have previously loaded one grid with topography (can have or not bathymetry in it). A second step before use is to

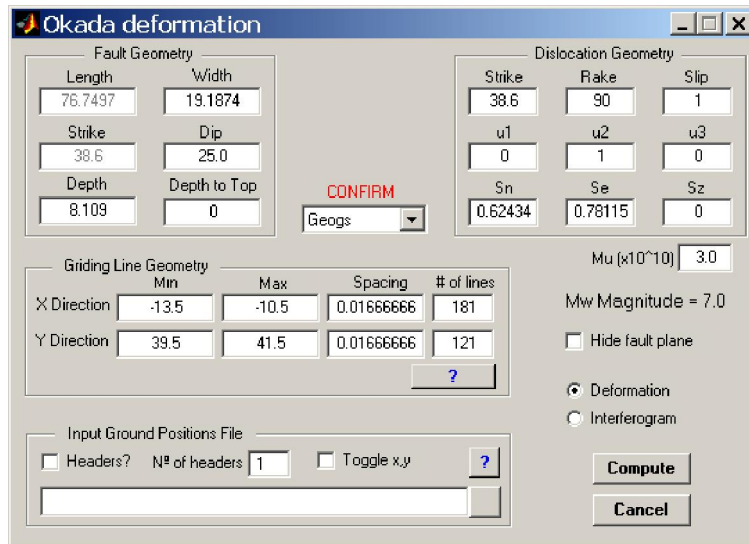
draw a trace fault in that map. That is accomplished with "Elastic deformation -> Draw Fault" option as explained also bellow. This tool uses the algorithm from: David J. Wald and Trevor I. Allen. Topographic Slope as a Proxy for Seismic Site Conditions and Amplification. Bulletin of the Seismological Society of America; October 2007; v. 97; no. 5; p. 1379-1395; DOI: 10.1785/0120060267



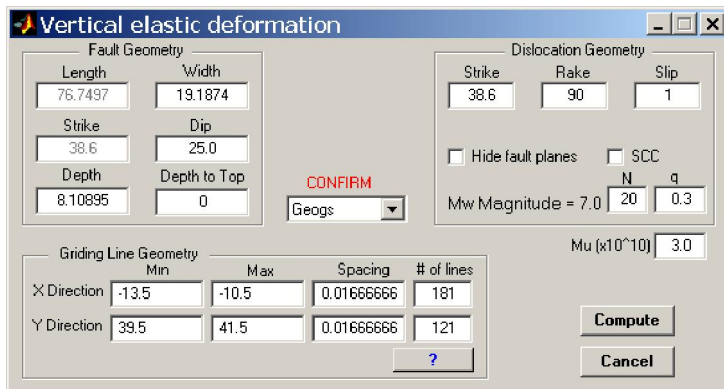
Elastic deformation



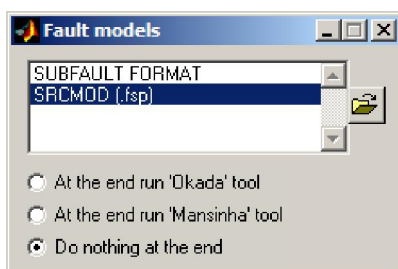
Here you can either import a line (or polyline) or draw it over your image grid. That line is then interpreted as a surface trace of a fault. Right clicking on the line will show (among others) two options to call new windows where you will specify the fault parameters (width, depth, dip) as well as the movement you will simulate. The two available options are called Okada and Mansinha. The Okada window is more general and oddly nearly twice fast and here you can: model co-seismic displacement - that can be compared with GPS measuring - for an initial condition of the far field tsunami modeling (see below) or to compute synthetic interferograms. The Okada window provides short help in the form of tooltips when the mouse rests a bit over individual entry boxes. For a complete reference you must read the RNGCHN paper (Feigl and Dupre 1999; Computers and Geosciences, 25 (6) pp. 695-704). The paper and the source code used to be available on authors ftp site but it seems to have vanished.



The Mansinha option computes only the vertical component of the deformation produced by an earthquake. The two codes should produce equal results (and they very closely do). On both Okada and Mansinha, a clever guess is made about the type of the grid coordinates. Accepted types are geographical, meters and kilometers. While the guessing works well most of the times it may also fail, so please pay attention to the guessed type and corrected if it is not correct. When working with geographical coordinates all distances are computed by previously projecting the geographical coordinates to Transverse Mercator with origin in fault's first point.

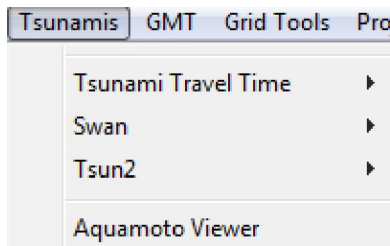


The above figure shows an example of Okada use and bellows a figure of the "Mansinha". Not shown in that figure is the one more of the Mirone's fine details. The horizontal projection of the fault(s) plane(s) are displayed on the main window. Also, if you have the Fledermaus viewer installed you can have a 3D visualization of the fault plane. For time being the fault's top is placed at "Depth to Top" below the grid's minimum and not at the fault length's average depth + "Depth to Top" as it would be correct.



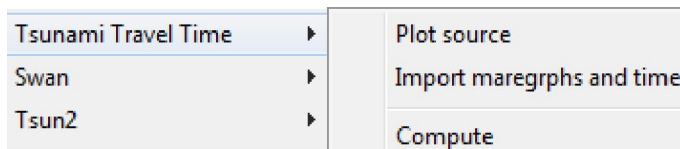
It is possible to import variable slip models using the SUBFAULT format (well, more like what I could guess that format is since I didn't find any formal format description) or the "Finite-Source Rupture Model Database" SRCMOD (.fsp) format maintained by Martin Mai at www.seismo.ethz.ch/srcmod For that, select "Import Model Slip" and select a file from the window below. As options show, you can compute the cumulated deformation of all individual patches. Off course that it may take some time if you are computing over a large grid.

Tsunamis



This menu holds the tools to do Tsunami propagation modeling.

Tsunami Travel Time

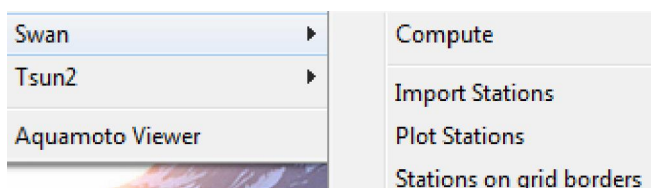


Computes the tsunami travel time according to equation $v = \sqrt{gH}$, where g is the gravity acceleration and H is the water column depth. To use it, first plot the source location (note:

you can give it an exact location), then right click on the source symbol and select Compute. That's all, depending on your grid size the result will appear immediately or wait a bit.

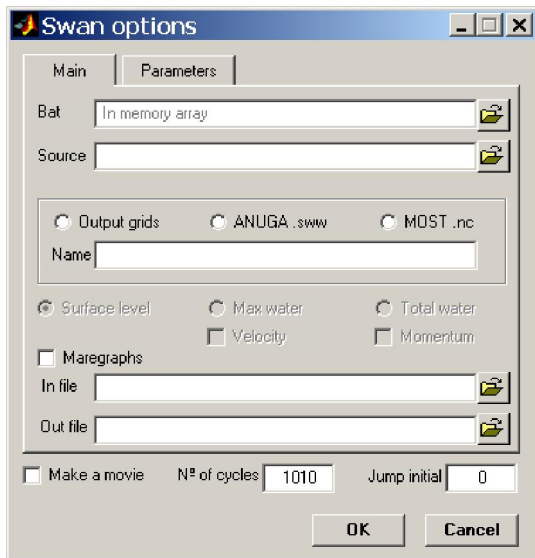
Alternatively you can load a file with x y time records (one per line) using the Import maregrphs and time option. Here the x y time represent, respectively, the coordinates and time of arrival at the tide gauge stations. The Compute will compute the tsunami source location using a least squares estimation of the backward ray-tracing time. Pay particular attention that none of your tide gauges is on land (that is, its depth must always be negative), otherwise the result is NaNs (NaN = Not a Number) and the shown image is all white.

Swan



Use the Mader's SWAN code to model tsunami propagation. To use this program you need to have two grids that are EXACTLY compatible. This means they must cover the same region and have the same number of rows/columns.

One of the grids represents the bathymetry (z positive up) in meters and other contains the initial deformation as produced by the Okada deformation module. The next thing you need is a parameter file. In the "data" directory of the root Mirone installation you will find a file named swan.par. Use this as an example for your case. I don't know the purpose of most of the parameters, only of those who seems to be important. The swan.par has short comments in front of each parameter.



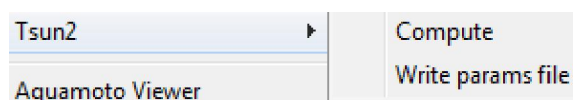
Now that you are ready to do the simulation you have to choose between computing the simulation on specific points - called maregraphs here - or on the whole grid region (but see the NOTE below). If you want the first option, than you either plot those stations graphically (using the Plot Stations option) or import a x,y file with the maregraphs location. This later option is also available in the swan_options window. If you want to compute the simulation in the whole region, then proceed with the **Compute** option. In the swan_options window you will finally select what to do. The Bat and Source edit boxes must contain either the name of your grids or a gray line with the *In memory array* string, which appears when the Swan module was called from a Mirone window in which a valid grid was previously loaded. In the *Params* edit box you load the swan.par parameter file mentioned

above. If you already plotted stations in the bathymetry window grid the Maregraphs section will be automatically set, otherwise check the check box and give the necessary file names to compute the maregraphs (in case you want to do it - not mandatory). Maregraphs are nothing but a record of the water height vs time at a certain location(s). The Output grids check box tells the program that you want to write grids on disk containing the water height at particular time intervals of the simulation cycle. For example if in the swan.par $dt = 1$ and $cumint = 30$ that means the simulation is computed at increments of one second and grids (as well as maregraphs if that option was selected) are written at every 30 seconds of the simulation. Grids are automatically named after the name template provided in the edit box appended with the cycle number. As an alternative to writing grids on disk, you can compute an AVI movie. In N° of cycles you select for how many cycles will the simulation is run. Again, using the same example of $dt = 1$ the default value of 1010 means the simulation will extend for 1010 seconds.

NOTE: if a rectangle exists on the window that contains the bathymetry file and you have selected to save grids, than that rectangle works as a sub-region selection. That is, the grids which are written cover only the region delimited by the rectangle. This option is original meant for creating input files to run the swan2sww command line utility which creates a .sww netCDF file to use with the ANUGA tsunami modeling code. But this is still a very fresh story.

I left behind on purpose the Plot stations on grid borders Swan option. This option's purpose is meant only for using in the Tsun2 module.

Tsun2



Use the Imamura's tsun2 code to model a tsunami inundation. Although the SWAN code has a parameter called flooding I don't know how to use it. Instead I use the tsun2 code to compute the coastal flooding simulations. In order to that these simulations are realistic, they have to be run over small areas described by very detailed DTMs that include both topography as well as bathymetry data. The so called far field codes are not appropriate to model coastal inundations and the shallow water models (like tsun2) use such a small grid spacing that for the sake of running time is just not practicable to make them run since the earthquake epicenter until the coast. The strategy is

than to use a far field model to compute the tsunami propagation using a coarse grid spacing (typically about 1 km) until near shore and then use this simulation to feed a shallow water code.

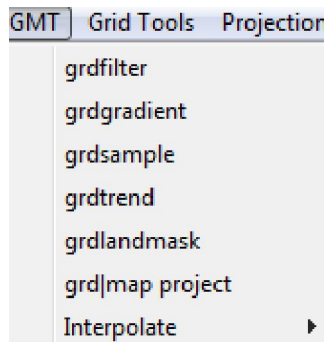
The tsun2 code thus runs on a very fine grid mesh - with grid spacing typically not superior to 50 meters - and must be fed at every grid node of one or two of its grid borders. The feeding procedure consists in providing a maregraph record for all grid nodes of the border(s) at which the external wave arrives at the finer grid. However, as mentioned above, this is impracticable because it would imply running an external feeding code (Swan in this case) at the same (very fine) grid spacing as the fine grid. The solution adopted was then to use finer grids that have grid node spacing that are sub-multiples of the coarse grid. For example, $dx = 50 \text{ m}$ equals $1 \text{ km} / 20$. That way we insure that every 20-th +1 finer grid node will coincide with one node of the coarser grid. To complicate a bit more (but safer against misalignments) it is desirable that the origin coordinates of the finer grid coincide with those of one node of the coarser grid. With these conditions met, feeding all grid nodes of the finer grid is simply accomplished by linear interpolation of the water height using data at nodes actually fed by the far field code (one every 21 in the example above). It is a user task to construct such a pair of coarse/fine grids using a cartesian coordinate system (e.g. UTM).

It is now that the Plot stations on grid borders option in Swan comes handy. By selecting this option the finer grid is loaded and all its grid nodes around the four borders will be plotted on the figure derived from the coarse grid (if you doubt it, zoom in until you are convinced). Here a previous knowledge of the direction from which the tsunami arrives is required. Remove the borders at which the tsunami cannot arrive (that is, without crossing the smaller area). That will leave you with at most two frontiers. Now run Swan and save the maregraph file. That file is the one you need to feed in the Tsun2 module, but before you can run it there is still one last step. Select Write params file in Tsun2 entry and save a new parameter file (this time for Tsun2). This file contains indexes information used to translate the maregraph file created by Swan into coordinates of the finer grid used in Tsun2. Selecting *Compute* pops up an option window very similar to the Swan's options window. In fact it contains only one extra option, which is *Jump initial*. This is meant for when the wave will arrive significantly later than the first records in the maregraph file. The value entered means the number of seconds to jump before the start of the simulation. As an example, with a time increment of 0.25 sec a jump of 100 means jumping the first 400 records of the maregraphs file.

Aquamoto Viewer

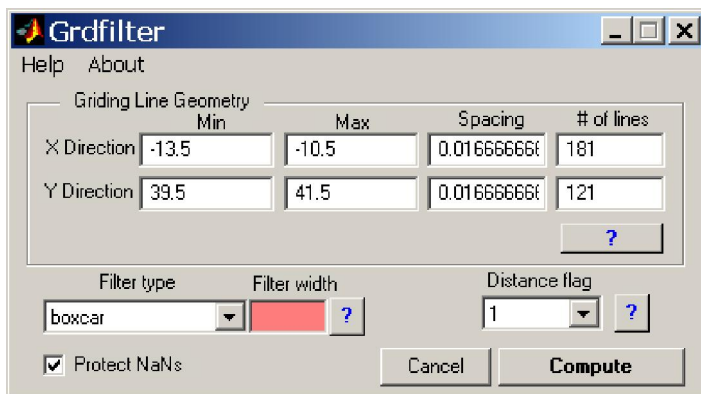
This tool is the same as described in the "Tools -> [Aquamoto Viewer](#)" menu.

GMT



This menu is the entry point to call helper windows to run some of (grid based) the GMT programs.

grdfilter

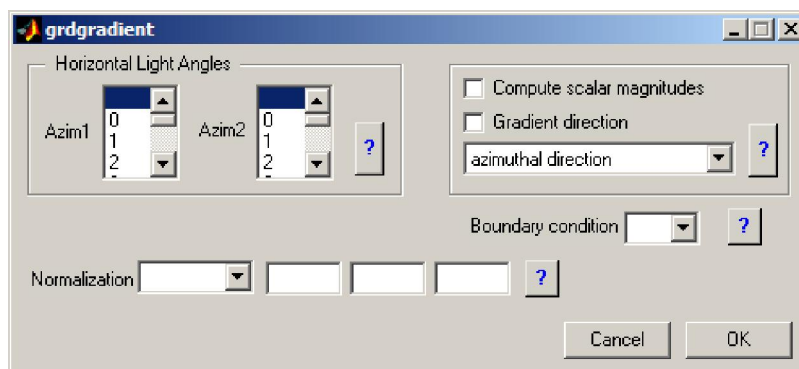


Will filter a grid in the time domain using a boxcar, cosine arch, gaussian, median, or mode filter and computing distances using Cartesian or Spherical geometries. The output can optionally be generated as a sub-region of the input and/or with a new increment. In this way, one may have "extra space" in the input data so that the edges will not be used and the output can be within one-half-width of the input edges. If the filter is low-pass, then

the output may be less frequently sampled than the input. Note that, contrary to the GMT version where the NaNs eaten by the filtering operation on the borders of the data/no-data, here NaNs are preserved by default. By that it is meant that input and output grids have the NaNs (when they exist of course) in the exact same positions.

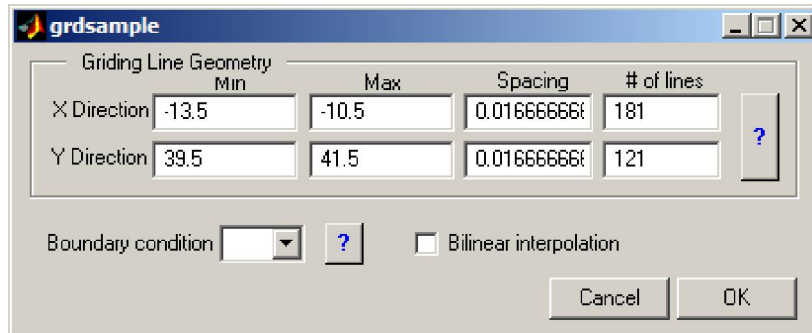
grdgradient

May be used to compute the directional derivative in a given direction, or the direction [and the magnitude] of the vector gradient of the data. Estimated values in the first/last row/column of output depend on boundary conditions. [Illuminate](#) algorithms 1 to 4 come from this module.

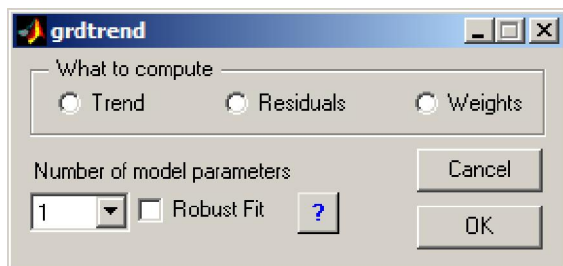


grdsample

Interpolates a grid to create a new grid with either: a new grid-spacing, or number of nodes, and perhaps also a new sub-region. Interpolation is bicubic [Default] or bilinear and uses boundary conditions. [grdsample](#) safely creates a fine mesh from a coarse one; the converse may suffer aliasing unless the data are filtered using [grdfilter](#).



grdtrend

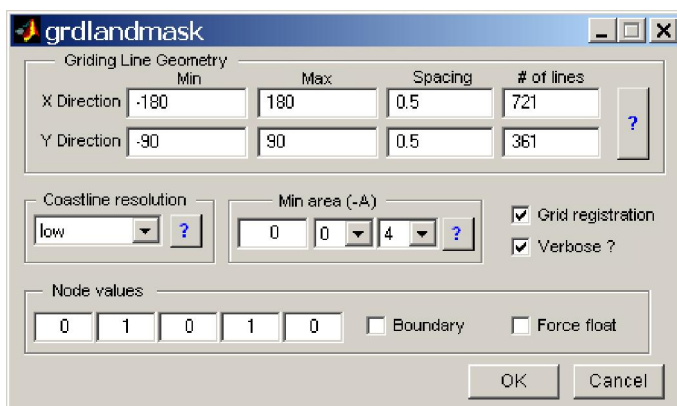


Reads a 2-D gridded file and fits a low-order polynomial trend to these data by [optionally weighted] least-squares. The trend surface is defined by:

$$m_1 + m_2*x + m_3*y + m_4*x*y + m_5*x*x + m_6*y*y + m_7*x*x*x + m_8*x*x*y + m_9*x*y*y + m_{10}*y*y*y$$

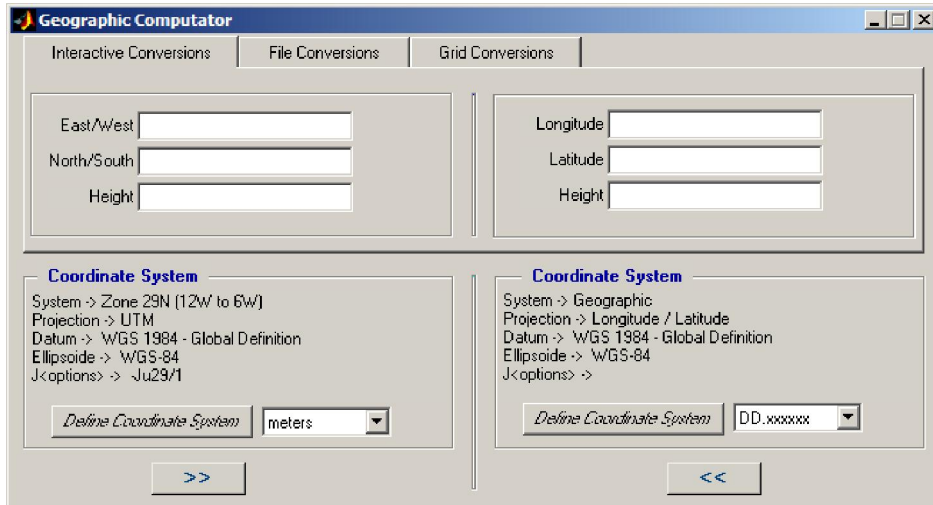
The user must specify `n_model`, the number of model parameters to use; thus, 3 fits a bilinear trend, 6 a quadratic surface, and so on. Optionally, select to perform a robust fit. In this case, the program will iteratively reweight the data based on a robust scale estimate, in order to converge to a solution insensitive to outliers. This may be handy when separating a "regional" field from a "residual" which should have non-zero mean, such as a local mountain on a regional surface. If data file has values set to NaN, these will be ignored during fitting; if output files are written, these will also have NaN in the same locations.

Grdlandmask



Create "wet-dry" mask image from shoreline data base. *grdlandmask* reads the selected shoreline database and uses that information to decide which nodes in the specified grid are over land or over water. The nodes defined by the selected region and lattice spacing will be set according to one of two criteria: (1) land *vs* water, or (2) the more detailed (hierarchical) ocean *vs* land *vs* lake *vs* island *vs* pond. The resulting mask grdf file may be created in a byte (black & white) or float format.

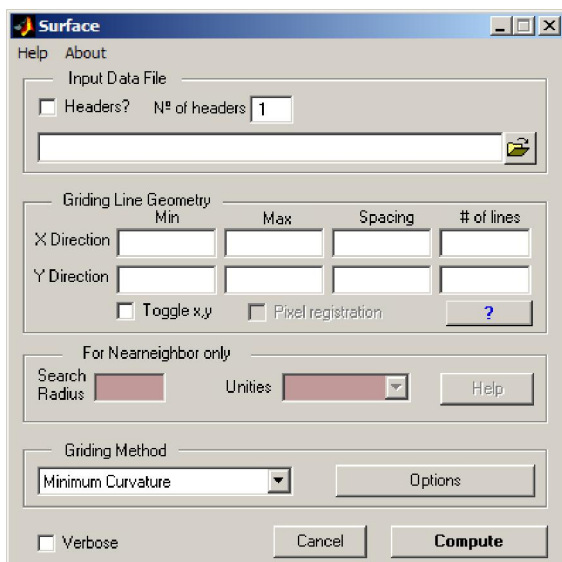
Grd | map project



This tool is an interface to both *mapproject* and *grdproject*. It uses a mex version of these programs and its use should be so simple that no help is needed (well, for users that know what they are doing). GMT4 introduced datum conversions in *mapproject* (but not in *grdproject*) and shifts that allow conversion between the plethora of projection systems used all over the world. In Mirone I coded only the Portuguese systems. I hope in this way to contribute to end the "state top secrete" issue that seems to exist around this matter (if you are Portuguese and need to do coordinate conversions from/to national systems I'm sure you understand what I mean). For other countries, I still need to invent a way to allow them to define their projection systems. Overall, this tool frees GMT users from the need to provide a correct -R (potentially particular hard to know in advance in inverse transformations), and prevented many users (me included) to be able to use *grdproject*.

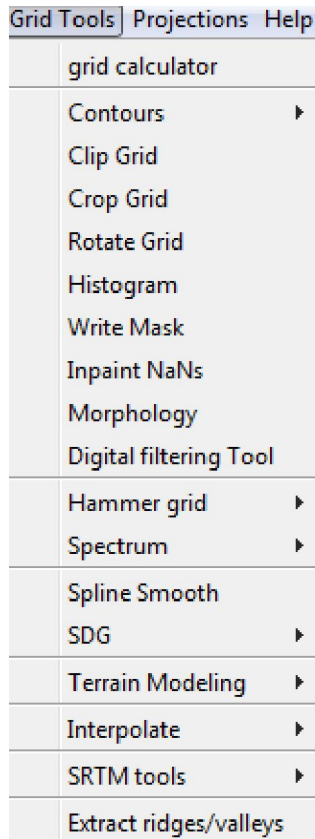
Note: within the grid projection available when the user hit the *Grid Conversions* tab, or used the GMT->*grdproject* path, changing the grid's x & y grid step is only possible after having done one first conversion. The reason is that before a first conversion we don't know yet the parameters of the projected grid.

Interpolate



Interpolate scattered x,y,z triplets into a regular grid array. Available options are GMT's surface and nearneighbor methods. Consult the corresponding GMT man pages for any droughts regarding these methods. Use should be obvious for anyone who knows what he/she is trying to do.

Grid Tools

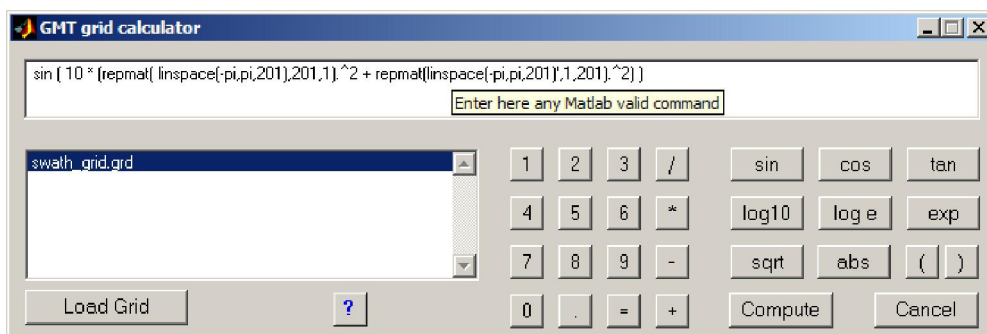


This menu gives access to a series of tools to do generic operations on grids.

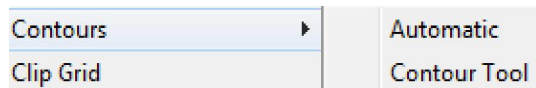
grid calculator

This is what it says. It operates on loaded grids but you can also load more grids (as long as they are GMT grids). When grids are loaded in memory they will show up in the lower box area. Double-click on a name to move it to the command area. The syntax used is that of Matlab. This means that in principle any Matlab single command that produces a 2D array can be used to produce a new image. eg: $\sin (10 * (\text{repmat}(\text{linspace}(-\pi,\pi,201),201,1).^2 + \text{repmat}(\text{linspace}(-\pi,\pi,201)',1,201).^2))$ Notice that we had to use *repmat* instead of *meshgrid* as we usually would do because we need to produce the output array in one single command like I said above.

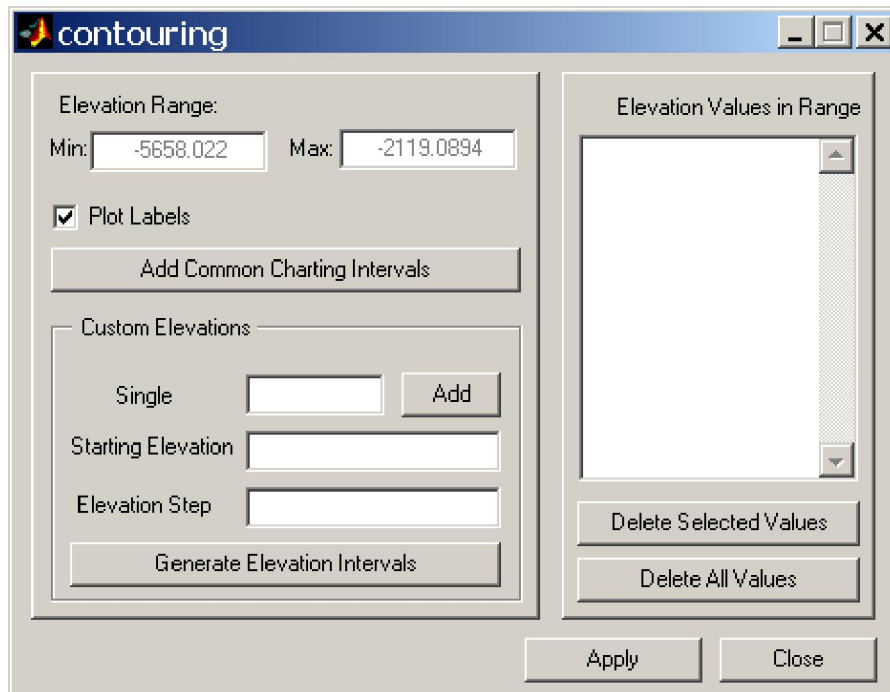
This option is not available in the compiled version.



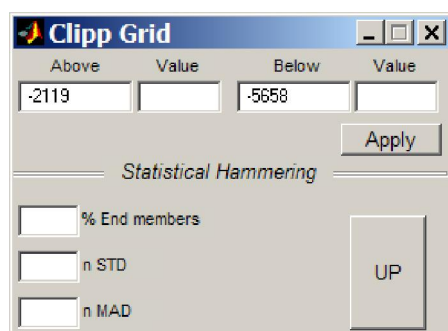
Contours



Draw contours on images derived from grids. The *Automatic* option plots the contours decided by a guessing algorithm. The *Contour Tool* window offers many different possibilities to select which contours to plot.



Clip grid



This corresponds to an interactive version of the *grdclip* GMT program. Shortly, values above "Above value" or below "Below value" are set to a desired value (including NaNs). The lower panel lets the user select the clipping interval based on statistical parameters. Use the "UP" button to transform the selected parameters into a clipping range.

Crop Grid

On images derived from grids, left click, drag and left click again. The rectangle thus defined is used to extract a subset of the original grid. This corresponds to an interactive version of the *grdcut* GMT program with the advantage that you will not fall onto the grips of the "Ivan the Terrible" (I mean the despairing message: GMT ERROR: (x_max-x_min) must equal (NX + eps) * x_inc), where NX is an integer and $|\text{eps}| \leq 0.0001$). Note, this is a crude and poorly controlled way of doing a cropping operation. The best way is to draw a rectangle on the image. Calmly decide on its size and position (or conversely, on its limits) and then inquire its properties (by a right-click on the rectangle borders) for "Crop Tools -> Crop Grid".

Rotate grid

This the same interface as described in Image -> [Image rotation](#) except that what is rotated here is a grid. Note: I'm not certain to be doing correctly the computation of the rotated coordinates.

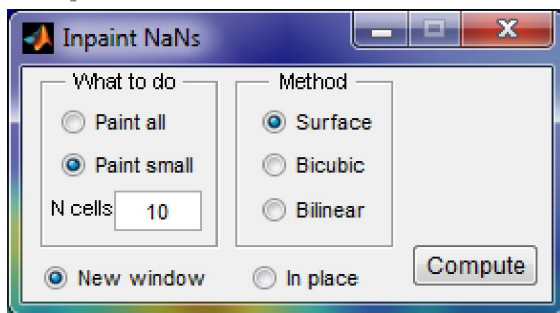
Histogram

Compute a histogram of the z (grid) values. User is prompted to enter the histogram bin width. Note that this option is also available for rectangular ROIs.

Write mask

With grids that contain NaNs it possible to write a mask grid in which the non NaNs will be replaced by the user entered value (e.g. 1).

Inpaint NaNs



This tool is meant to fill holes in grids that are represented with NaNs. The filling process is obtained by interpolation using the good values around the holes. Options are offered to fill only holes of up to a certain size (the "Paint small") or all the holes. Use can also choose between three interpolation algorithms.

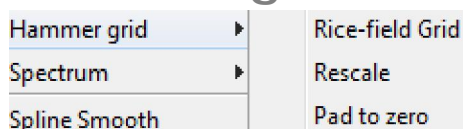
Morphology

This is the same tool already introduced in "Image -> [Image morphology](#)" but when used here it will operate on the grid (the floating point array) and not on the derived image as it is in the other case.

Digital Filtering Tool

This the same interface as described in Image -> [Digital Filtering Tool](#) except that here what is filtered is a grid and not its derived image.

Hammer grid



Rice-field Grid

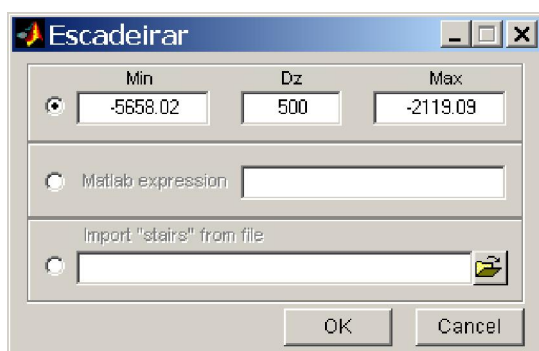


Image a rice field in one mountain in Thailand. Well that will be the aspect of the transformed grid if seen in 3D. To control how the stair-case is constructed, user can tell it via min/max/dz boxes, give it via a Matlab expression or import the stairs from an external file. Use this transform to create a classification grid, for example, for slopes.

Rescale

Create a new grid by rescaling the parent grid. In other words, apply a scale factor to parent grid. Useful for example for cases when one wants to convert between meters and kilometers or to change the grid's sign.

Pad to zero

Cosine taper the grid to zero. The user is prompted for the extension number of rows/columns. This a useful option to pre-process bathymetry and elastic deformation grids before doing tsunami propagations because it highly reduces the annoying (and false) grid edge reflections.

Spectrum

Spectrum	Amplitude spectrum	Compute either the log10 power or the amplitude spectrum. Note1, I am normalizing the FFT output by dividing by n_rows x n_columns. Note2, before FFTing the grid is cosine tapered to the extended dimension that better suits the two conditions: 1, the extended grid is at least 20% larger in both dimensions; 2, only "good FFT numbers" (with prime factors <= 5) are retained. Note3, this option is also available for rectangular ROIs.
Spline Smooth	Power spectrum	
SDG	Autocorrelation	

Compute the grid autocorrelation (it uses FFT). Note that this option is also available for rectangular ROIs.

Spline Smooth

Uses cubic splines to smooth a grid. The important step in smoothing is the value of smoothing parameter. The smoothing parameter (p) determines the relative weight you would like to place on the contradictory demands of having f be smooth vs having f be close to the data. For p = 0, f is the least-squares straight line fit to the data, while, at the other extreme, i.e., for p = 1, f is the variational, or 'natural' cubic spline interpolant. As p moves from 0 to 1, the smoothing spline changes from one extreme to the other. The interesting range for p is often near $1/(1+h^{3/6})$, with h the average spacing of the data sites, and it is in this range that the default value for p is chosen. For uniformly spaced data, one would expect a close following of the data for p = $1/(1+h^{3/60})$ and some satisfactory smoothing for p = $1/(1+h^{3/0.6})$. You can input a p > 1, but this leads to a smoothing spline even rougher than the variational cubic spline interpolant.

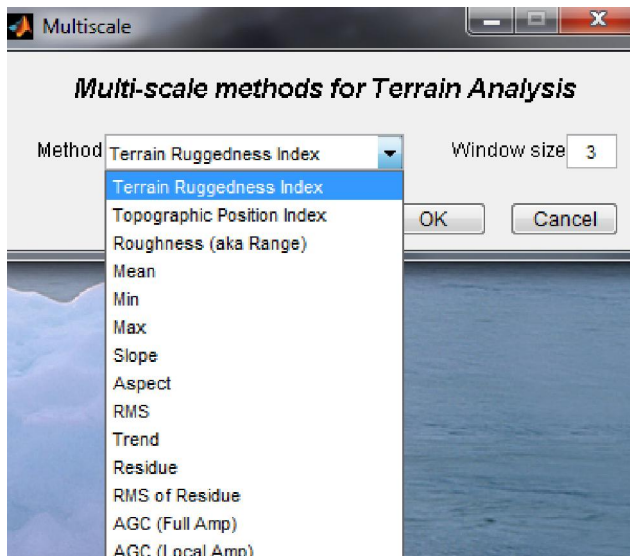
SDG

SDG stands for Second Derivative in direction of Gradient. It is a very interesting operator, particularly to people working in the UNCLOS margins extension program. Shortly, it is the UN recommended algorithm to determine the Foot Of continental Slope (FOS). Even if you do not know what I am talking about, this operator worth trial because it has the ability to reveal very concealed trends in your data. It also depends on the smoothing parameter p as the previous option.

Terrain modeling

Terrain Modeling	Slope	In this section, the slope (either in degrees or percentage), aspect (that is the azimuth of the slope) and first and second directional
Interpolate	Aspect	
SRTM tools	Directional derivative	
Extract ridges/valleys	Multi-scale Analysis	

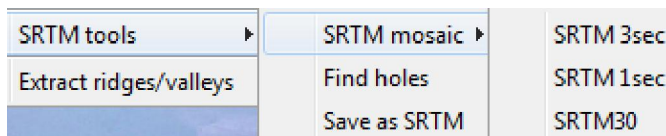
derivatives of geographical referenced grids can be computed.



The “Multi-scale Analysis” is a tool designed to perform block operations. The moving window size, in terms of number of cells, is controlled by the value entered in the “Window size” box. A collection of different algorithms is available. The first three of the list are common in geomorphology (algorithms programmed after: Wilson et al 2007, Marine Geodesy 30: 3-35).

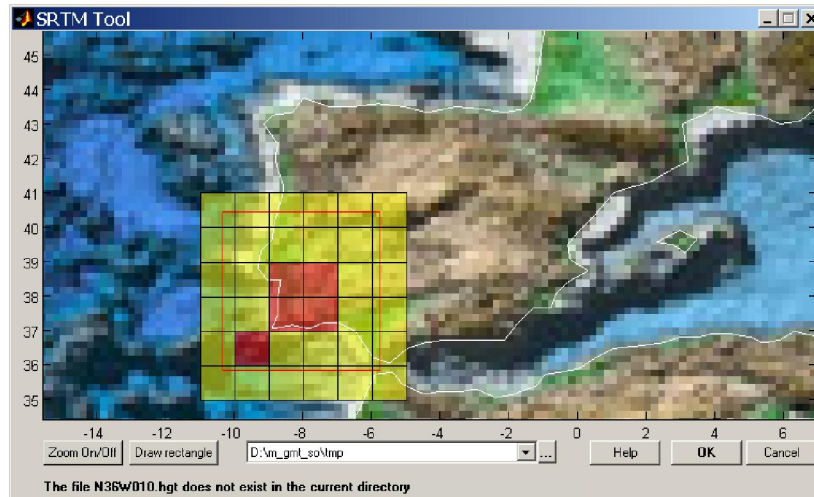
Because the methods here involve a moving window, results obtained for slope or aspect (for example) are smoother than those obtained with the tools above.

SRTM tools



With the SRTM mosaic tool you can draw a rectangle and select an interest region. Right click on the rectangle and choose mesh. This will draw yellow quadrangles corresponding to possible

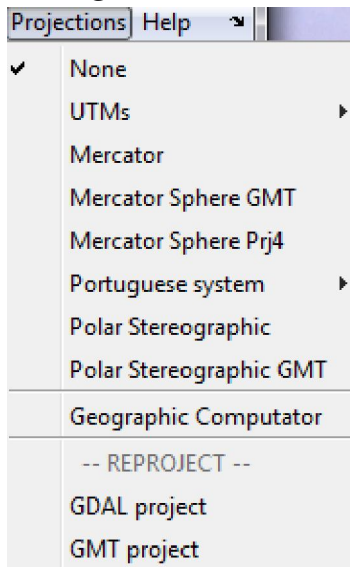
SRTM tiles. Now a mouse click on each individual square will search on the previously selected working directory for existing SRTM files (that may be .zip or .gz). If the matching file does exist the square will turn red, if it does not a warning message is displayed during a second and the square will return to yellow. Note, “mosaicing” is only possible with SRTM1 and 3 arc seconds files. However, this tool is also handy for loading SRTM30+ files where it provides a nice graphical way of selecting the tile of interest. The find holes option will automatically detect regions of no data values (holes) and draw rectangles around them. Right click on those rectangles and select the options under crop tools to fill the holes. At the end the processed file may be saved back into the SRTM format. Note, saving back to SRTM is possible only for the 3 arc seconds files.



Extract ridge/valleys

Again this is more an intention than a true useful tool. While it works reasonably well (but be careful and do not abuse with grid size) in extracting ridges (only working mode) the problem is that the ppa (ref) algorithm used outputs the crest lines as a sequence of non-contiguous two point lines segments. This makes it impossible to treat them as polylines.

Projections



Starting on 1.1.0 version it is possible to instruct Mirone about projections using the GMT `-J` syntax or the [Proj4](#) style projection string. This is accomplished by the instructions inside the `SRSproj_def.txt` file that resides on the 'data' subdirectory. As an example let's analyze the default contents of that file.

```

1. #Menu SubMenu      GMT -J ProjectionString plus
   eventual -C<fe/fn> -T<datum>
2. None                0          0
3. UTM's              UTM26     -Ju26
4. UTM's              UTM29     -Ju29
5. Mercator           0          -Jm
6. # Next we have a Proj4 ProjectionString
7. "Portuguese system" "Gauss-Militar (DLx)" +proj=tmerc
   +lat_0=39.66666666666666 +lon_0=-8.131906111111111 +k=1.0 +x_0=200000
   +y_0=300000 +ellps=intl
8. "Portuguese system" "Gauss (WGS84)" +proj=tmerc
   +lat_0=39.668258333333333 +lon_0=-8.133108611111111 +k=1.0 +x_0=0
   +y_0=0 +ellps=wgs84

```

Lines starting with the '#' character are interpreted as comment lines. The other lines have three fields. For example, line 2 has: `None 0 0` The meaning is the following: `None` is the name of the submenu that appears under the Projections main menu. The two zeros are used in this case just to indicate non used values (`None` means any projection at all). On third line we have `UTM's UTM26 -Ju26` Now `UTM's` is the name of the first submenu and `UTM26` is the name of the sub-submenu. This translates to say UTM projection Zone 26. The last field, `-Ju26`, is the GMT switch used to inform the code that a Universal Transverse Mercator projection, on Zone 26, is selected. If you want to add other projections, use this logic and consult the GMT *psbasemap* manual to learn all details of the `-J` syntax.

Lines 7 and 8 define the projection using the Proj4 style projection string and show another aspect of the menu labeling. If one wants to use names with spaces, we need to wrap it under a pair of `" "`.

Well, the above applies when we have loaded a file that we know the projection but that has no intrinsic mechanism (metadata) to inform the code what is its projection type. This is normally the case of the GMT grids. However, through the GDAL library Mirone can read many different formats that are able to store projection info, for example GeoTiff files.

When that is the case, it takes precedence over any eventually selected projection on the Projections menu.

And what is the use of knowing what the projection is? Well, when you import external data or instruct to plot coastlines the data is automatically projected into the currently active projection.

Another situation that must be accounted for is when the user wants to load already projected data into a projected image. Since by default external data is assumed to be in geographic coordinates we must be able to instruct the program to not project this imported data. This is accomplished by right-clicking on the image coordinates annotations (the frame) and hit the "Load files in geogs". This will toggle the message to "Load in projected coords".

Note: this menu is disabled when the loaded file was guessed to have geographical coordinates.

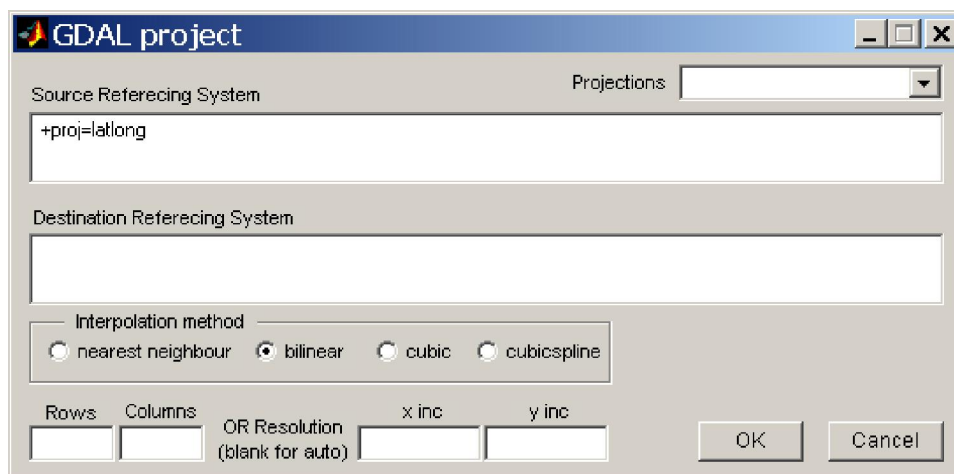
Geographic computator

This entry is a repetition of the [grd|map project](#) but it remains visible and active all the time. The other one, on the GMT menu, will disappear when a non-grid file type is loaded. It is also similar to *GMT project* shown below, but that one will only operate on grids, whilst this one can also be used to transform point data.

-- REPROJECT --

-- REPROJECT --
GDAL project
GMT project

Under this menu the user can select to try to do a projection change using GMT's `gdproject` already presented above (see [Geographic computator](#)), or to try it with GDAL machinery. Note there are two similar entries to the GMT projection tool because one of them is hidden when the file loaded is not a grid. I said "try" because both of them are not entirely stable and may either crash or result in idiot solutions. This latter case is typical of GDAL warping of global grids. Unfortunately, GDAL is quite bugged in this case but developers don't seem to care much about it. The syntax used but by this tool follow the proj4 rules. Templates for the most known projections are provided via the *Projections* popup menu, but it's up to the user to fill in with the correct projection parameters. If the input grid had metadata describing its projection, this will be pasted into the *Source Referencing System* edit field.



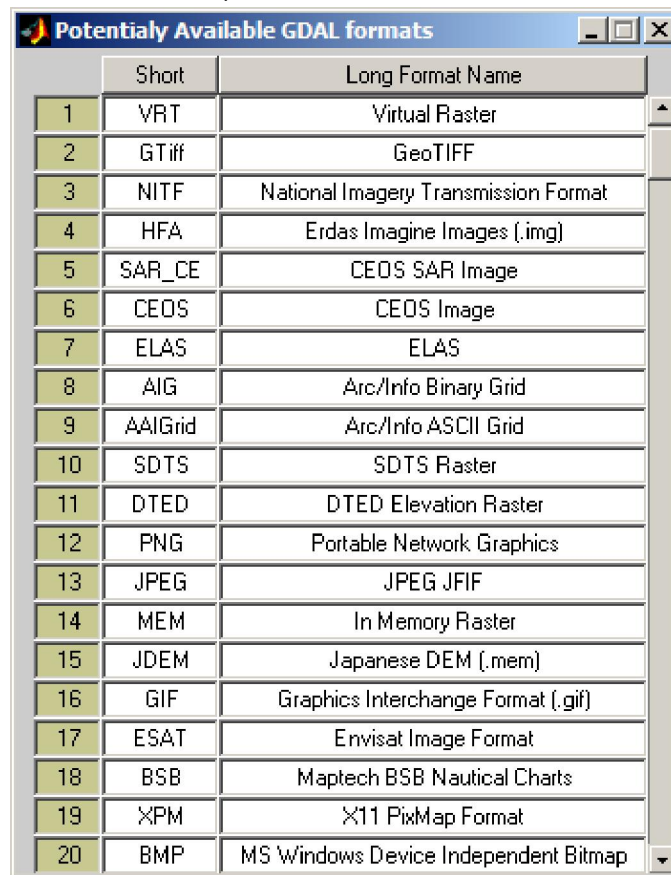
Help

Mirone Help

Calls this help file

List GDAL formats

This pops up a table with the names of the formats currently recognized by the GDAL import wrapper and that are potentially loadable by Mirone. In case of doubt use the File -> Open Grid/Image -> Try Luck with GDAL option.



	Short	Long Format Name
1	VRT	Virtual Raster
2	GTiff	GeoTIFF
3	NITF	National Imagery Transmission Format
4	HFA	Erdas Imagine Images (.img)
5	SAR_CE	CEOS SAR Image
6	CEOS	CEOS Image
7	ELAS	ELAS
8	AIG	Arc/Info Binary Grid
9	AAIGrid	Arc/Info ASCII Grid
10	SDTS	SDTS Raster
11	DTED	DTED Elevation Raster
12	PNG	Portable Network Graphics
13	JPEG	JPEG JFIF
14	MEM	In Memory Raster
15	JDEM	Japanese DEM (.mem)
16	GIF	Graphics Interchange Format (.gif)
17	ESAT	Envisat Image Format
18	BSB	Maptech BSB Nautical Charts
19	XPM	X11 PixMap Format
20	BMP	MS Windows Device Independent Bitmap

“Hidden” options

Mirone still has a LOT more of options that are not ranged in menus, for the simple reason that there is no way to do it without complicating enormously the graphical interface. Instead, they only show up when the situations make it appropriate and then they are accessible with a mouse right-click on a graphical element. For example, polygons drawn on a grid derived grid let you do a median filter that is applied only to the grid nodes inside the polygon. I wrote this option at sea when I wanted to get rid of those awful effects that result on the overlap of contiguous swaths on multi-beam surveys. On the other hand, if the polygon is an exact rectangle you can do FFT like operations. Another example is given by right clicking on the axes coordinates. If the grid is in geographical coordinates, you can then select among several different ways of displaying degrees (eg, dd.xxxx or dd:mm:ss).

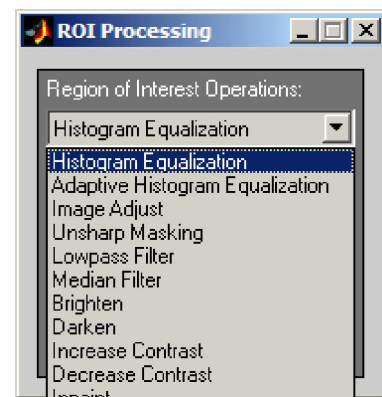
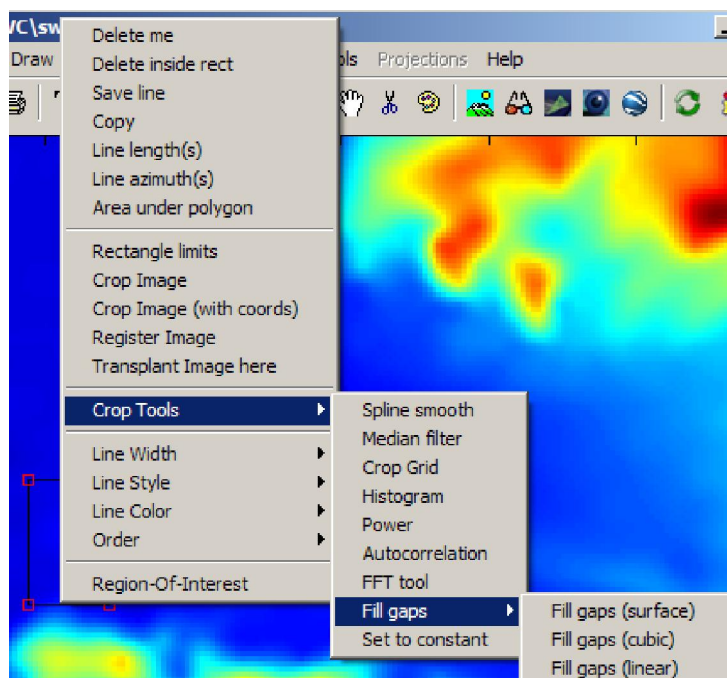
Furthermore, if you decide to save a line into an ascii file, the x,y coordinates will be formatted in the same way as the selected coordinate display. I hope that this will be appreciated by those of you who know what is getting a way points, picked on a computer screen in deg.xx, go up to the ship's bridge and listen the Captain (or whoever is driving the ship that dawn) tell you "Can't you give me those way points in degrees and minutes?"

A very cool new thing in 1.4 is the capacity to copy-and-paste lines between Mirone windows. One only have to set the target line in edit mode (by double-clicking on it) and that hit Ctrl-C followed by a Ctrl-V on the destination window. Furthermore, since the line is stored in the (sorry, Windows clipboard) one can past it anywhere else that Windows accept (text editors, EXCEL, etc ...).

Another probable unique feature is the capacity of immediately display the contents of shape files (well, maybe not all of them) or ASCII .dat files. Instead of working as an text editor, the contents of x,y .dat files are magically displayed. Just try to drop the \$MIRONEROOT\data\gp_girl.dat into the icon of the compiled version or open it via "Datasets -> External db" on an empty window. Hum, nice girl ...

The example figure below shows the options available when one right click on a previously plotted rectangle. On the lower left corner we can see a part of the rectangle, which has little red squares over the vertices. This indicates that the square is in edition mode and may be rescaled/moved in the normal way of click-and-drag. Notice that the options under "Crop Tools", already described in this manual, will operate only inside the area delimited by the rectangle. The same would happen if we had drawn an irregular polygon.

Still another (simple) method to georeference images is provided by the "Register Image" option. If selected, the user is prompted to enter the rectangle's four corners coordinates in the new reference system. After filling the necessary information the image is than referenced to that new system.



Direct data loading

Within the MATLAB version one can use Mirone as an advantageous replacement of the ML *image* or *imshow* functions. This can be done using one of the three following modes:

1. `mirone file_name` (this works with many different formats, as long as it possible to find out what is in "*file_name*")
2. `mirone(Z)` Where Z is an array of any data type. Uint8 or logical arrays are treated as images. 'Upper' types (e.g. Int16, single, doubles) are scaled to uint8 to derive an image and the underlying data is stored in memory to be accessed when needed.
3. `mirone(Z, struc)` Z as above and "struc" is a structure with coordinates information. The "struc" structure has optional and mandatory fields. Bellow is a description of those fields and what they should contain depending the cases of use.

OPTIONAL fields

- `struc.geog` -> = 1 (geog) or = 0 (other coords) (To inform if coordinates are geographical or other)
- `struc.cmap` -> contains a Matlab [Mx3] colormap (Color map)
- `struc.name` -> 'Whatever you like' (Figure name)
- `struc.srsWKT` -> a projection WKT string as the ones used by GDAL (If you know the projection in terms of a WKT (Well Known (by who?) Text).
([Related Note](#), the ogrproj MEX file lets you convert a Proj4 projection string into a WKT string example:
`strWKT = ogrproj('+proj=merc')`)

MANDATORY fields

- `struc.head` -> An header [1 x 9] array which contents are:
`[x_min x_max y_min y_max z_min z_max 0 x_inc y_inc];`
The zero (7th element) indicates that grid registration is used.
- `struc.X` and `struc.Y` - Where we have two cases:
 1. The input array is of type uint8 and contains an image, than:
`struc.X = [x_min x_max]`
`struc.Y = [y_min y_max]`
 2. The input array is of type int16 or float (single or double), than:
`struc.X = linspace(x_min, x_max, n_columns)`
`struc.Y = linspace(y_min, y_max, n_rows)`

Where `[n_rows, n_columns] = size(Z)`;