User manual for

TomoWarp2

Erika Tudisco, Edward Andò, Rémi Cailletaud and Stephen A. Hall

Contents

1 Introduction **5**

2 Installation **7**

2.1 Linux . . . 7

2.2 MACOSX . . . 7

2.3 Windows . . . 8

2.3.1 Prerequisites . . . 8

2.3.2 Installation . . . 12

3 Getting Started **13**

4 Required Parameters **15**

4.1 Identifying the datasets and the output folder . . . 15

4.2 Essential DVC parameters . . . 16

4.3 Image Specification . . . 16

5 Optional Parameters **19**

5.1 File specifications . . . 19

5.2 Region of interest and prior information . . . 20

5.3 Computation specifications . . . 20

5.4 Subpixel refinement . . . 21

5.4.1 Advanced settings . . . 21

6 Post Process **23**

6.1 Strain calculation . . . 23

6.2 Filters . . . 24

6.3 Save preferences . . . 24

6.4 Pixel size correction . . . 25

6.5 Run Post process and Results Preview . . . 25

6.5.1 Show Displacements . . . 25

1. Introduction

TomoWarp2 belongs to the local variant of image correlation techniques This approach involves first defining the analysis points within a reference volume (the first image "*im1*"), which are then associated by a local subset (called correlation window) about he corresponding point. The displacements of the points are determined by finding the best mathematical match of each subset in the reference volume with equivalent subsets in a subsequent volume (*im2*). The search for the best match is restricted to a "search window" defined by the user-provided expected displacements. The goodness of the match is defined by some form of correlation coefficient. This approach provides displacements with a resolution of 1 pixel, but higher resolution is usually required; subpixel resolution can be obtained by either interpolating around the maximum of the local correlation coefficient field in displacement space or by interpolating the image itself (which allows more degrees of freedom).

2. Installation

2.1 Linux

1. Install necessary packages
   As root run
   `apt-get install python-numpy python-scipy python-matplotlib python-imaging-tk`

2. Install SWIG
   As root run
   `apt-get install swig`

3. Set path to the folder containing TomoWarp2
   example: `cd /opt/TomoWarp2`

4. Set path to the c code folder
   `cd pixel_search/c_code`

5. Run command
   `python setup.py build_ext --inplace`

2.2 MACOSX

1. install prce (can be found here)

2. install swig

   `$ ./configure --prefix=/home/yourname/projects $ make $ make install`

3. Set path to the folder containing TomoWarp2

4. Set path to the c code folder
   `$ cd pixel_search/c_code`

5. Run command
   `$ python setup.py build_ext --inplace`

6. Set path to the tools folder
   `$ cd ../../tools`

7. Run command
   `$ python setup_tifffile_c.py build_ext --inplace`

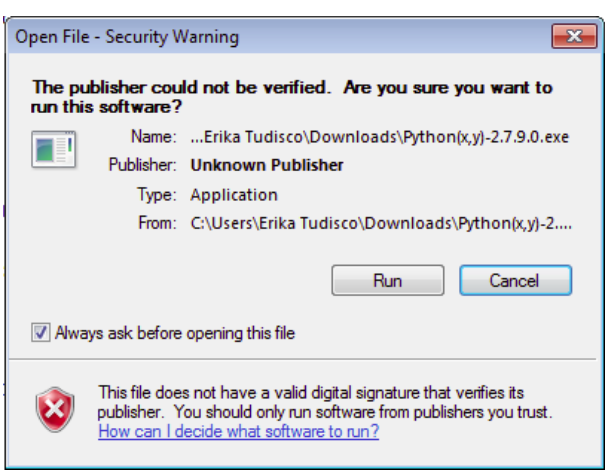
2.3 Windows

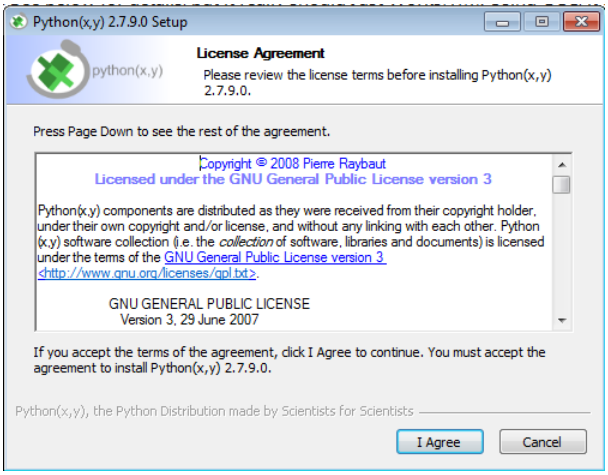
2.3.1 Prerequisites

To install TomoWarp 2 you will need Python, a C compiler, and SWIG (a program which connects C or C++ to higher level languages – Python)

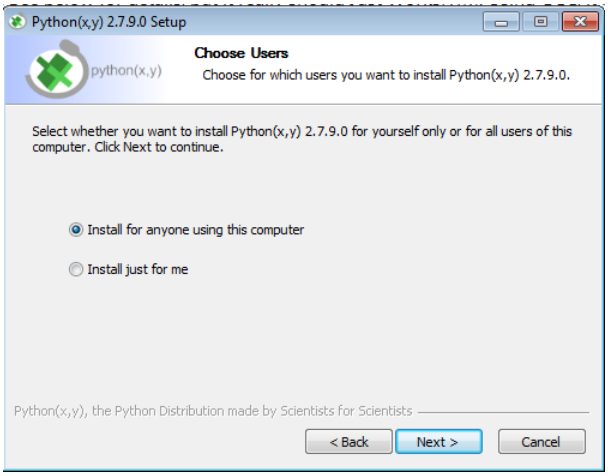
- Python and iPython

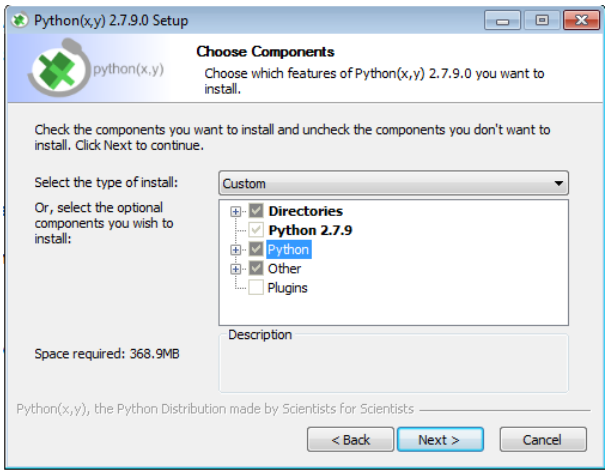
  1. Download python(xy) version 2.7.9 here
  2. Open exe file, eventually authorize to run it, and agree to licence



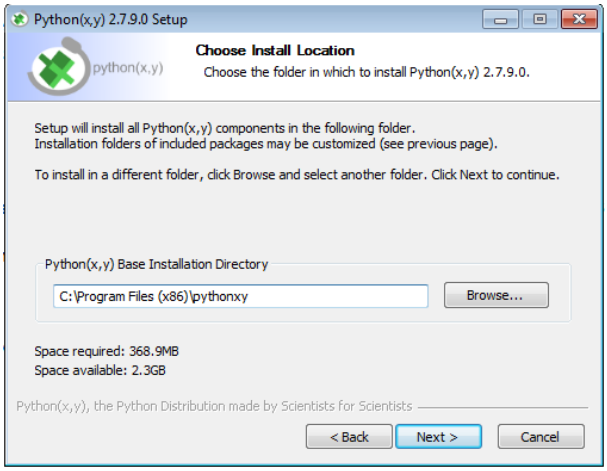


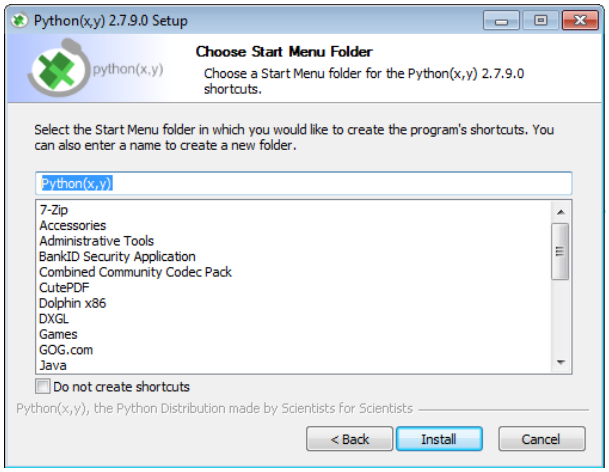
  3. Choose users that can access the program and components to be installed





  4. Choose install location and start menu folder

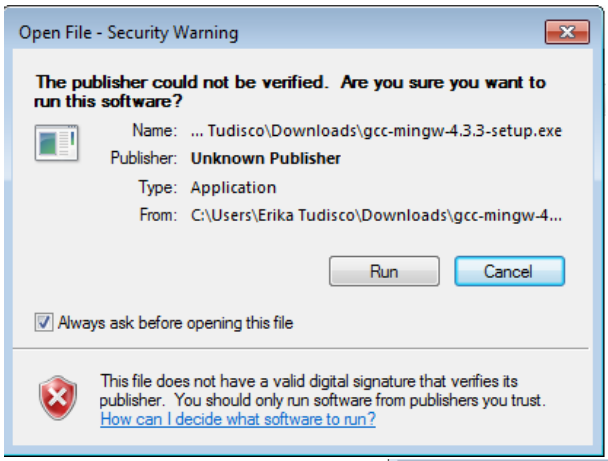


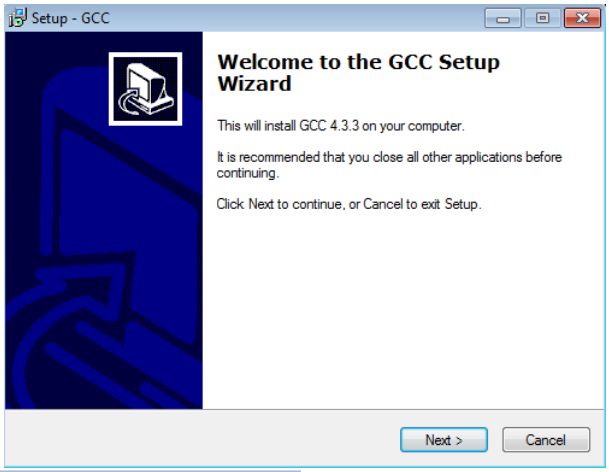


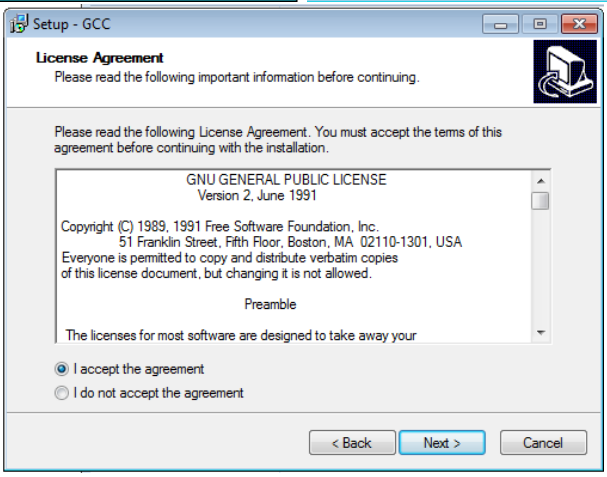
- C compiler

1. Donwload MingGW here. This version fully integrates within official Python installation to allow compilation of extensions.

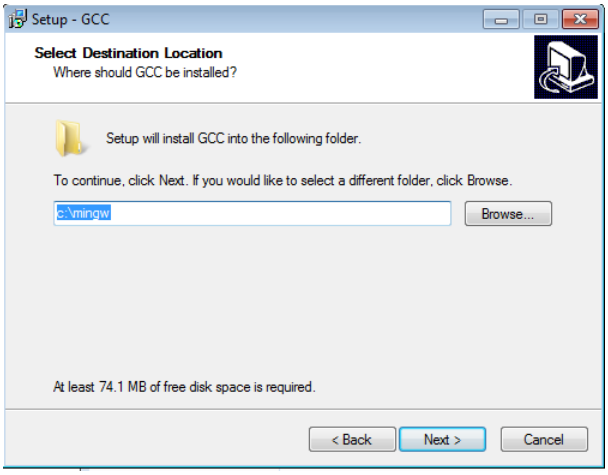
2. Open exe file, eventually authorize to run it, and agree to licence

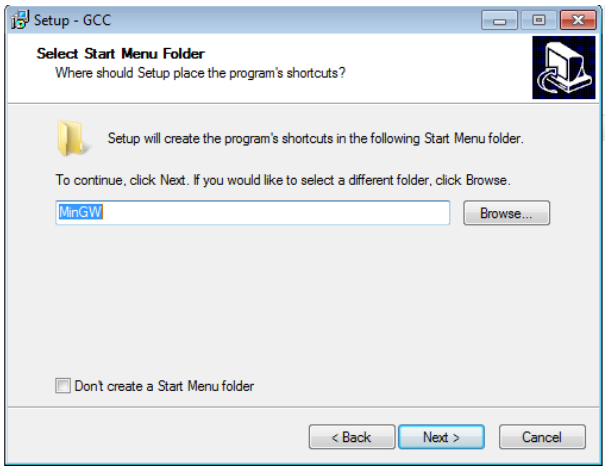




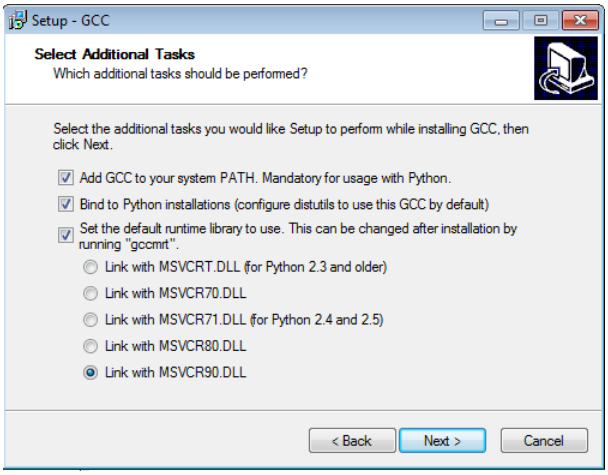


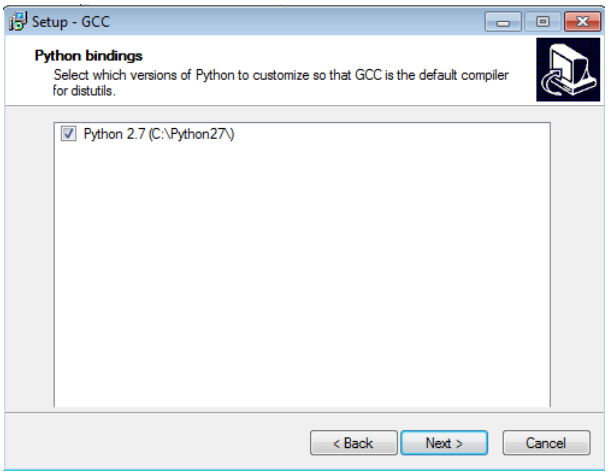
3. Choose install location and start menu folder



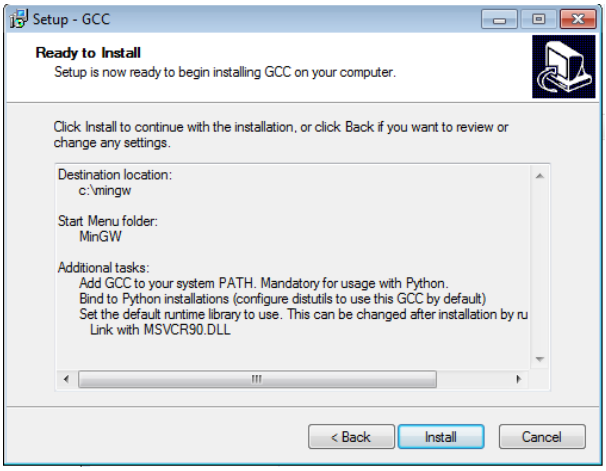


4. Choose runtime libraries and python version



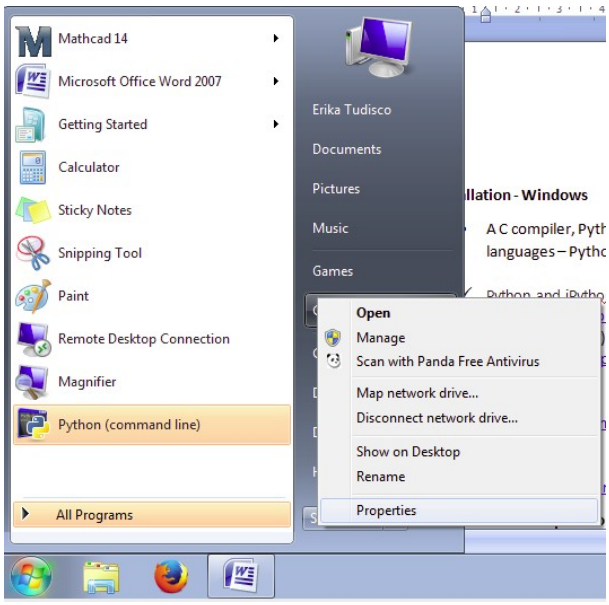


5. Install

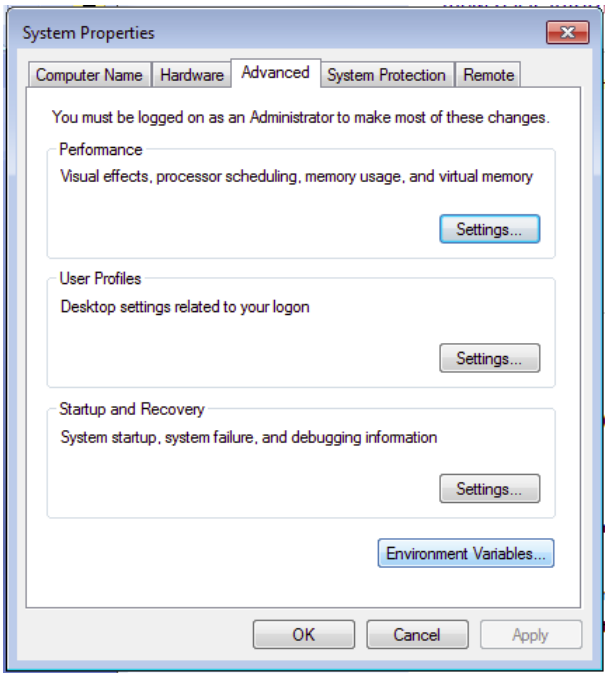


- SWIG

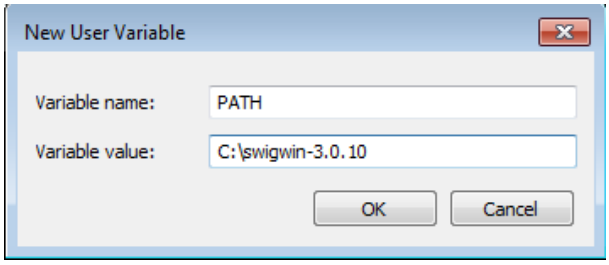
1. Download SWIG from here
2. Unzip the file in C:
3. Open “Control Panel\System and Security\System” or In Windows7 right click on “Computer” from the main menu and then “Properties”



4. Choose “Advanced sysstem settings” to open the System Properties and then “Environmental Variables”



5. Make a new user variable PATH or modify the existing one



N.B. Check the name of the swig folder for current version.

2.3.2 Installation

- Open Command prompt
- Set path to the folder containing TomoWarp2
  example: `cd C:\TomoWarp2`
- Set path to the c code folder
  `cd pixel_search\c_code`
- Run command
  `python setup.py build_ext --inplace`

3. Getting Started

From the command line set the path to the main folder of TomoWarp2 and run

- `python TomoWarp2.py $PathToInputfile` - run the programme using the parameters from `Inputfile`
- `python TomoWarp2.py --desktop` - open the GUI
- `python TomoWarp2.py --desktop $PathToInputfile` - open the GUI and load the parameters from `Inputfile`

`Inputfile` is is an ASCII file in which the parameters are specified in separated rows in the form `variableName = variableValue`. When values have to be entered for the three possible direction, these has to be included in square brackets and always in the in the order Z, Y, and X. Where, if the volume is provided as a set of 2D images, Z is the direction through the slices while Y and X are the vertical and the horizontal directions in the slice.

This file can be manually or automatically edited (by using the GUI) as specified in the following sections where the `variableName` is given in squared brackets followed by its default value (if any) in italic and its format underlined.

Examples of `Inputfile` are provided in the folder *examples*, for 3D and 2D cases, together with the relative sets of images.

The GUI is composed by three tabs: "Required Parameters", "Optional Pa- rameters" and "Post Process". Through the options reported in the footer the User can:

- load an existing configuration file (button "Load input file" on the left);
- select default values for optional parameters (button "Default values" on the left);
- specify if the dataset is a surface or a volume ("Image dimensions 2D/3D" radio box in the centre);
- Quit the GUI, Save the configuration file, run TomoWarp2 with the configurations entered into the GUI masks ("Exit", "Save" and "Run" buttons on the right).

4. Required Parameters

This tab is intended to acquire all the information related to the datasets to be processed (location, file format, size and encoding) and the essentials about the DIC procedure (extension of the subvolume, spatial resolution of the output lattice where displacements are calculated and initial guess for the displacements).

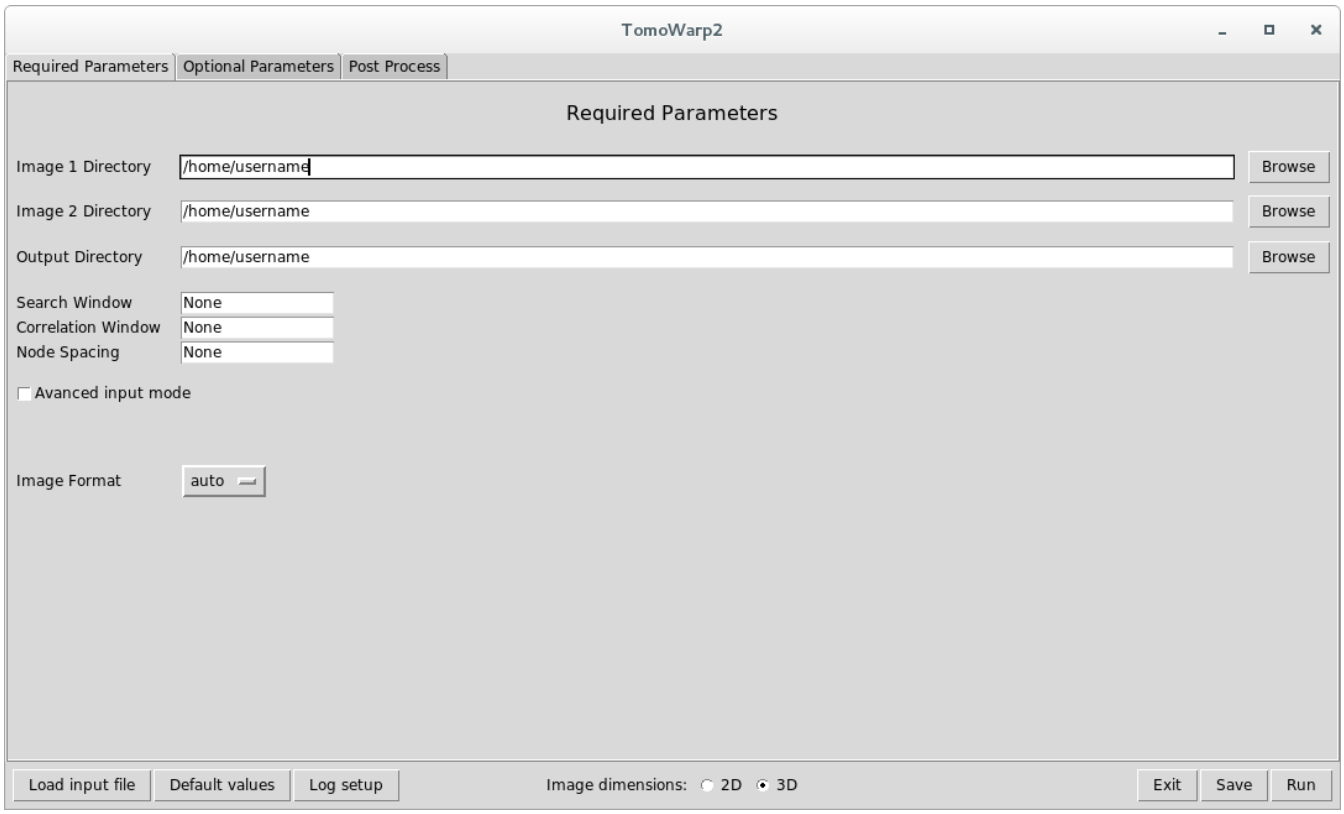


Figure 4.1: GUI - Required Parameter tab

4.1 Identifying the datasets and the output folder

Image 1 Directory [`DIR_image1` string] reports the folder where is stored the first dataset;

Image 2 Directory [`DIR_image2` string] reports the folder where is stored the second dataset; it can be the same directory specified for the first dataset: in this case, each dataset has to be identified by specifying further details on the filename in the tab "Optional Parameters", as reported in section 5;

Output Directory [`DIR_out` string] defines the folder where the output files will be saved.

4.2 Essential DVC parameters

Search Window [`search_window` [[int, int], [int, int], [int, int]]] defines the range of possible displacements values in pixels; if only one value is given the displacement are intended to be in any possible direction.

Correlation Window [`correlation_window` [int, int, int]] defines the subvolume extension from the central node, in pixels (*i.e.,* the size of the dataset portions compared in the reference and deformed images); if only one value is given the extension is intended to be the same in any possible direction.

Node Spacing [`node_spacing` [int, int, int]] the spacing between the points of the output grid where the displacements will be calculated; if only one value is given the spacing is intended to be the same in any possible direction.

If the check box "Advanced input mode" is selected, all the above mentioned filed accept different values for each direction of the reference system, as shown in figure 4.3

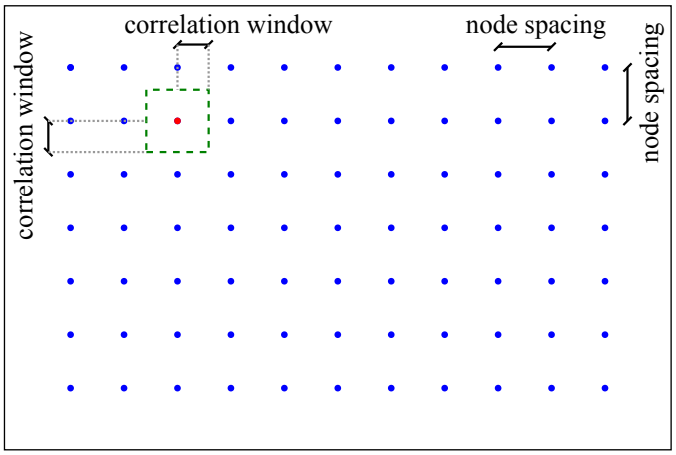


Figure 4.2: DIC essential parameters

4.3 Image Specification

TomoWarp2 reads dataset in a number of format that can be specified in a menu:

Image Format [`image_format`= *auto* string] has to specified as a string between the following options:

- auto
- TIFF_PIL
- EDF
- RAW
- TIFF

If 'auto' is selected the program will try to guess the format from the extension of the files (This is not possible for RAW format).

If 'RAW' is selected as input format, the User must specify further details about the datasets.

N.B. If a 3D RAW volume is used it is necessary to specify, other than the input directory, also a filter for the filename of the image.

Image Size [`image_1_size` [int, int, int] or [int, int]] and [`image_2_size` [int, int, int] or [int, int]] dimensions of each volume in the form [sizeZ, sizeY, sizeX] or [sizeY, sizeX];

Image data format [`image_data_format` string] can be selected among the available options listed into a menu or specified as a string following "Array-protocol type strings" where the first character specifies the kind of data and the remaining characters specify the number of bytes per item (see here for details and examples)

Little-endian byte order the check-box must be selected if little endian byte encoding is adopted. In the input file it is specified by prepending the string in [`image_data_format`] with '<'

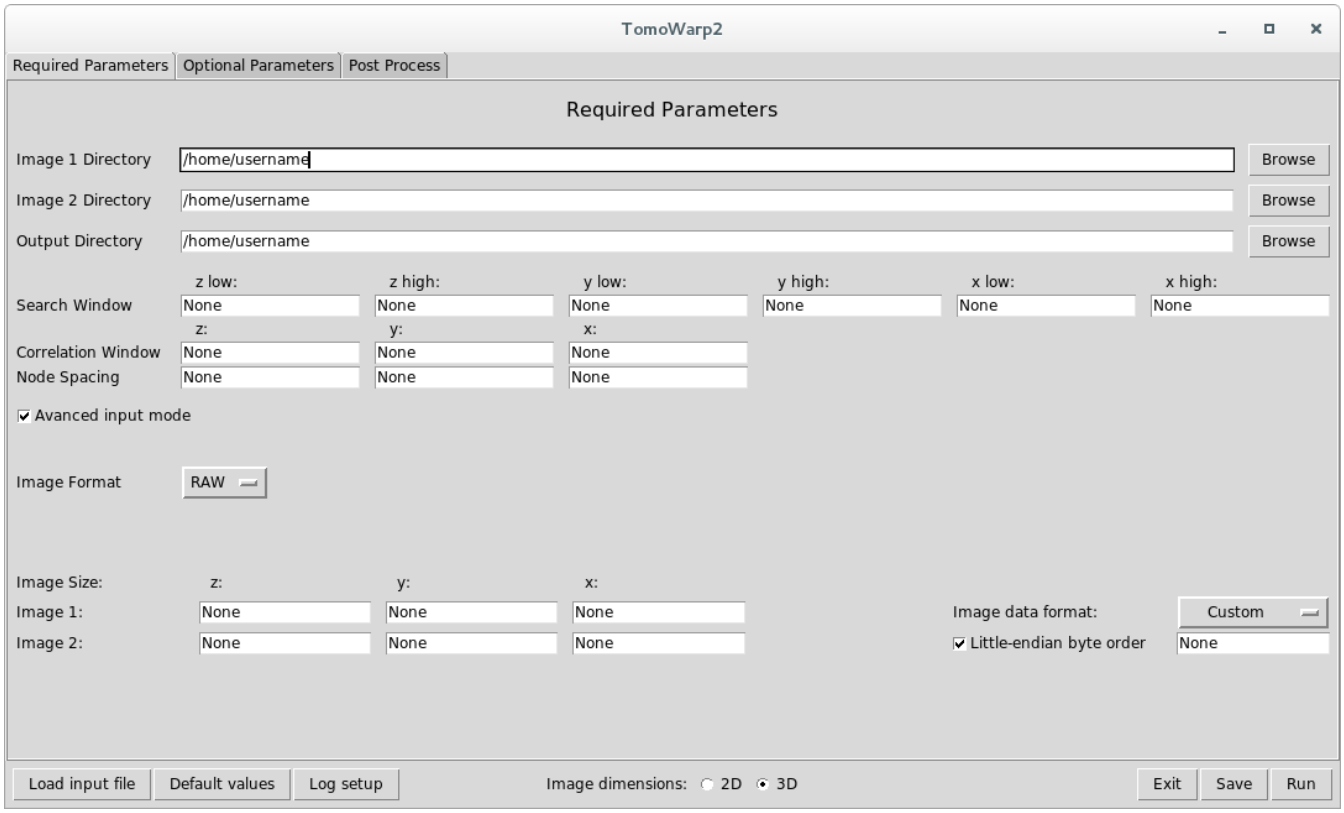


Figure 4.3: GUI - Required Parameter tab advanced mode

5. Optional Parameters

This tab gives the possibility to better specify the images to analyse, the region of interest and other advanced options.

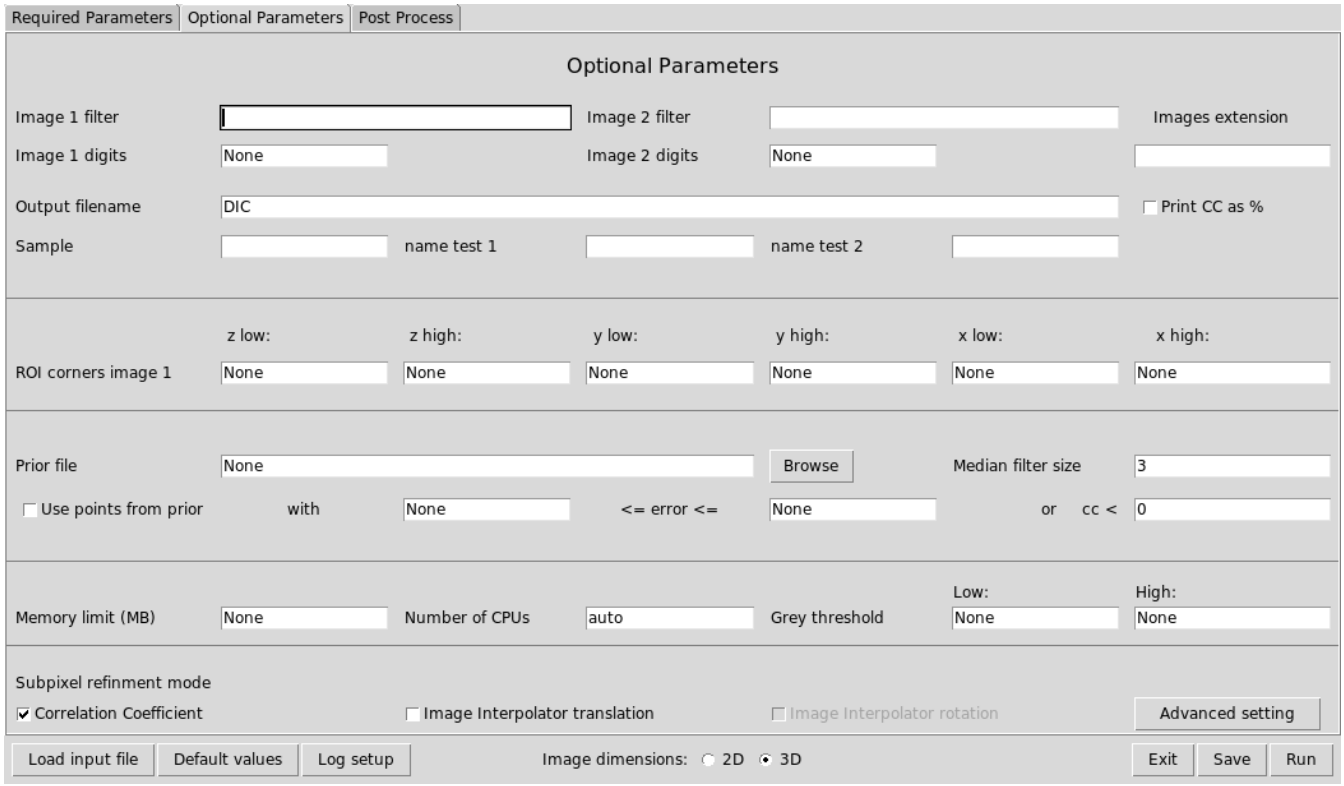


Figure 5.1: GUI - Optional Parameter tab

5.1 File specifications

Image 1 filter [`image_1_filter` string] filter on the filename of the first images that allows only specific files to be selected in the input directory;

Image 2 filter [`image_2_filter` string] filter on the filename of the second images that allows only specific files to be selected in the input directory;

Image 1 digits [`image_1_digits` int] number of digits identifying each slice of the first image;

Image 2 digits [`image_2_digits` int] number of digits identifying each slice of the second image;

Images extension [`image_ext` string] file extension. It can be specified as a regular expression

Output filename [`output_name`=*'DIC'* string] Prefix for all the output files produced by the code. It can be automatically generated by specifying the following parameters:

Sample [name_prefix string]

name test 1 [name_1 string]

name test 2 [name_2 string]

The Output filename will be constructed as [name_prefix]-[name_1]-[name_2]-ns=[node_spacing]

Print CC as % [cc_percent=*False* boolean] if checked the correlation coefficient is saved with a scale between 0 and 100 (100 = perfect correlation). By default the scale is set to be between 0 and 1.

5.2 Region of interest and prior information

ROI corners image 1 [ROI_corners [[int, int, int], [int, int, int]]] define the extension of the region in which the analysis will be performed. It is specified in the form [[Zlow, Ylow, Xlow], [Zhigh, Yhigh, Xhigh]]. The region refers to the first image, the extended region of the second image needed to perform the analysis is calculated automatically based on Correlation Window and Search Window.

TomoWarp2 has the facility to run the correlation for a coarse grid of nodes to provide a prior estimate of the displacements for a finer grid by interpolation of the coarse grid results. In this way a multi-scale calculation can be performed that can reduce significantly the computational time. Priors are accepted and loaded as TSV files in the same format as the output from the program, *i.e.*, they can be generated by a previous run of the code or generated by some external process. The prior field will be interpolated linearly to assign a guess value for the displacement to each node of hte new grid defined with the Node spacing specified in the first tab.

Prior file [prior_file string] TSV file containing a row for each node with node number, Z, Y, and X coordinates, and a guess displacements in the three directions.

Median filter size [priorSmoothing=*3* int] defines the size of the cube used to perform a median filter on prior field. It has to be a an odd number. To disable the filter set it to zero.

The TomoWarp2 provides also the possibility of reprocessing a subset of nodes for which a previous correlation did not give satisfactory results. The analysis can be improved in those nodes changing input parameters such as the search window, the correlation window sizes or the subpixel refinement. The nodes to be reprocessed can be selected based on the output error or a threshold on the computed correlation coefficient. All other nodes retain their original correlation result.

Use points from prior [usePriorCoordinates=*False* boolean] if checked the prior field will not be interpolated, the Node spacing ignored and the same nodes as in the prior file will be used;

¡= error [errorLowLimit int] the new analysis will be carried out only on nodes having a higher error code;

error ¡= [errorHighLimit int] the new analysis will be carried out only on nodes having a higher error code;

cc ¡ [prior_cc_threshold=*0* float] the new analysis will be carried out only on nodes having a lower correlation coefficient.

5.3 Computation specifications

Memory limit [memLimitMB int] in the volumes are given in slices this sets the limit to the RAM that will be used. Slices will be loaded in blocks and the analysis performed by layer. the output file will not be affected by the procedure;

Number of CPUs [`nWorkers`=*'auto'* int] set the maximum processor. If set to auto all of the available CPUs will be used;

Gray threshold Low [`grey_low_threshold` float] if the mean value inside the correlation window is lower than this value the analysis on the corresponding node will not be carried out. This can be used, for instance, to reduce the computational time by skipping nodes falling in voids;

Gray threshold High [`grey_high_threshold` float] if the mean value inside the correlation window is higher than this value the analysis on the corresponding node will not be carried out. This can be used, for instance, to reduce the computational time by skipping heavy objects in a x-ray tomography.

5.4 Subpixel refinement

The subpixel refinement of the displacements has two different implementations. The fastest one, Correlation Coefficient Interpolation (CC-interpolation), consists in finding the maximum of a function interpolating the correlation coefficient at the 27 points surrounding the best integer-displacement match to provide a sub-pixel interpolated displacement vector. Another possibility, Image Interpolation, numerically displaces the subset of the deformed image until it matches the reference one. This implementation is much more computationally expensive, but allows rotational degrees of freedom to be taking into account easily.

Subpixel refinement mode [`subpixel_mode`=*True, False, False* [boolean, boolean, boolean]] it defines which implementation is used. In order the three values set

Correlation Coefficient

Image Interpolator translation

Image Interpolator rotation only enabled when Image Interpolation translation is checked

If both modalities are selected, the sub-pixel displacement obtained from the first technique is used as an initial guess for the optimiser.

5.4.1 Advanced settings

Clicking on the "Advanced settings" button will open a new window (see figure 5.2)

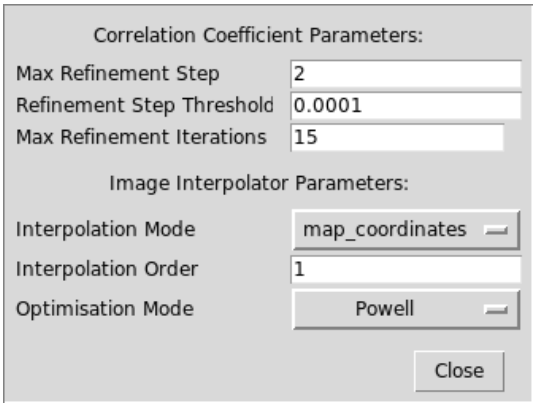


Figure 5.2: GUI - Subpixel refinement advanced settings

Correlation Coefficient Parameters

Max Refinement Step [`subpixel_CC_max_refinement_step`=*2* int] maximum distance from the best integer-displacement;

Refinement Step Threshold [`subpixel_CC_refinement_step_threshold`=*0.0001* float] threshold for stopping criterion;

Max Refinement Iterations [`subpixel_CC_max_refinement_iterations`=*15* int] number of maximum iterations before stopping the process.

Image Interpolator Parameters

Interpolation Mode [`subpixel_II_interpolationMode`=*'map_coordinates* string] can be set from a menu between the following values:

- pytricubic
- map_coordinates

Interpolation Order [`subpixel_II_interpolation_order`*1* int] order of the image interpolation (odd-modes are strongly encouraged);

Optimisation Mode [`subpixel_II_optimisation_mode`=*'Powell'* string] minimisation framework that takes advantage of all the different schemes implemented in *scipy.optimize.minimize*. can be set from a menu between the following values:

- Nelder-Mead;
- Powell;
- CG;
- BFGS;
- Newton-CG;
- L-BFGS-B;
- TNC
- COBYLA;
- SLSQP;
- dogleg;
- trust-ncg;
- subPixelSearch.

6. Post Process

This tab give the possibility to control the post process parameters and output preferences as well as to preview the found displacement field and filter it before strain calculation.

The post process is run automatically after the correlation analysis but it can also be run again from this tab on the last or another TSV file.

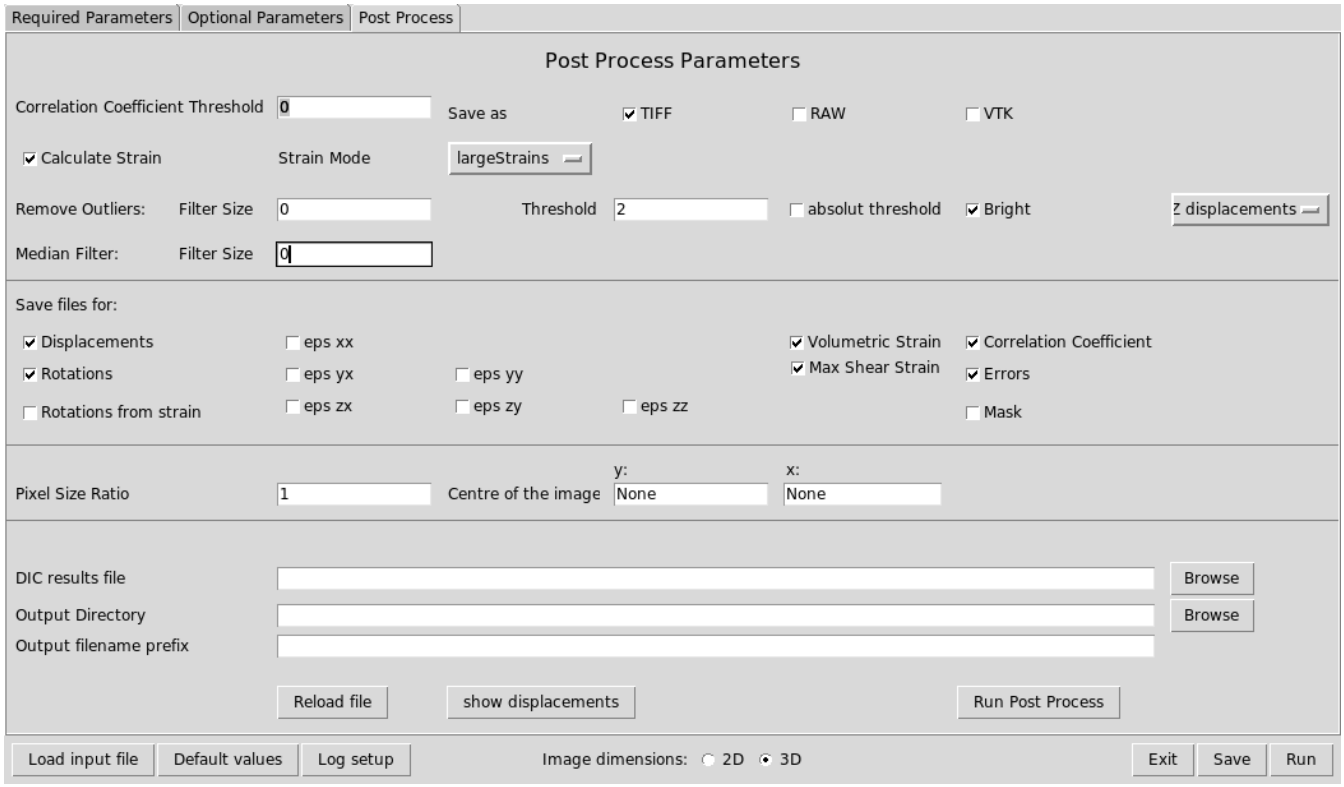


Figure 6.1: GUI - Post Process tab

6.1 Strain calculation

Correlation Coefficient Threshold [`cc_threshold`*0* float] points with a correlation coefficient lower than this threshold will be given a Nan and not taken into account in the strain calculation.

Save as output can be saved as:

TIFF [`saveTIFF`=*True* boolean]

RAW [`saveRAW`=*False* boolean]

VTK [`saveVTK`=*False* boolean]

Calculate Strain [`calculate_strain`*true* boolean] if checked strains are calculated

Strain Mode [`strain_mode`=*'largeStrains'* string] strains can be calculated in different frames. The mode can be set from a menu between the following values:

- largeStrain - strain is calculate for every cube of the grid
- largeStrainsCentred - strain is calculate using the 8 nodes surrounding the every node
- tetrahedralStrains - strain is calculate for every tetrahedral formed between the nodes of the grid
- smallStrains - strain is calculate for every cube of the grid in the hypotesys of small strain.

6.2 Filters

Displacement field can be automatically filter before strain calculation by setting the following parameters:

Remove Outliers outliers are spotted by comparing the point value to a fixed value or to the median calculated on a cube.

Filter Size [`remove_outliers_filter_size`=*0* int] the size is intended as half of the cube side. If set to 0 the filter is not applied;

Threshold [`remove_outliers_threshold`=*2* float] threshold that determines if a point is an outlier;

absolut threshold [`remove_outliers_absolut_threshold`=*False* boolean] if checked the point value is compared directly to the threshold; if not the value is compared to the mean value;

Bright [`remove_outliers_filter_high`=*True* boolean] if checked the point is considered an outlier if the value is higher than the threshold or than the median value plus the threshold. If not checked the point is considered an outlier if the value is lower than the threshold or than the median value minus the threshold.

[`filter_base_field`=*0* int] the outlier are spotted on one of the component of the displacement field. The possible values are:

- 0 for Z displacements
- 1 for Y displacements
- 2 for X displacements

Median Filter

Filter Size [`kinematics_median_filter`=*0* int] the size is intended as the cube side and it has to be odd. If set to 0 the filter is not applied;

6.3 Save preferences

Save files for in this section it can be chosen which output are saved

Displacements [`saveDispl`=*True* boolean]

Rotations [`saveRot`=*True* boolean]

Rotations from strain [`saveRotFromStrain`=*True* boolean]

Strains [`saveStrain`= $[False, False, False, False, False, False, True, True]$ 8x[boolean]] values are defined in the order: ε_{zz}, ε_{zy}, ε_{yy}, ε_{yx}, ε_{xx}, ε_{zz}, volumetric, maximum shear

Correlation Coefficient [`saveCC`=*True* boolean]

Errors [`saveError`=*True* boolean] error code for points where the analysis did not work

Mask [`saveMask`=*False* boolean] reports a Nan value for points where the analysis did not work or the correlation coefficient is lower than the threshold.

6.4 Pixel size correction

If the two volume images have different pixel size this can be corrected by giving:

Pixel Size Ratio [`pixel_size_ratio`*1* float]

Centre of the image [`image_centre`] point, in a slice, in which the volumetric strain is not influenced by the pixel size change, typically the centre of rotation.

6.5 Run Post process and Results Preview

DIC results file TSV file containing the results of the correlation analysis (it is automatically filled in at the end of a calculation);

Output Directory folder in which the new results will be saved in;

Output filename prefix prefix for the new results files.

Reload file if an other TSV needs to be reprocessed this button has to be pressed after browsing the new file.

Run Post Process applies filters and calculates strain.

show displacements this will open a new window previewing the displacements fields (see figure 6.2).

6.5.1 Show Displacements

The three components of the displacements are shown. Points that do not match the condition on grey threshold are shown in white while points in which an error was encountered are shown in cyan. Text entries below each colorbar allow the contrast to be adjusted.

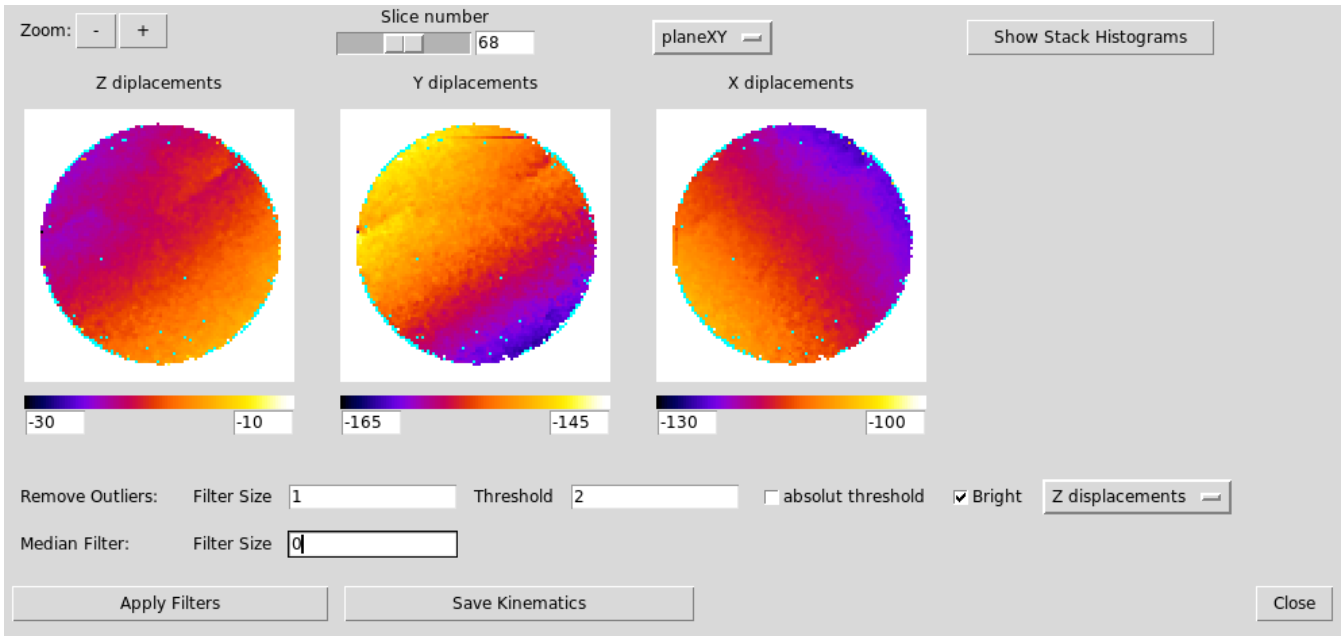


Figure 6.2: GUI - Post Process, show displacements window

From this window filters can be applied repeatedly by pressing the "Apply Filters" bottom. Filter parameters detail can be find in section 6.2. There is no "Undo" option for the filter application. Figure 6.3 shows the results of a

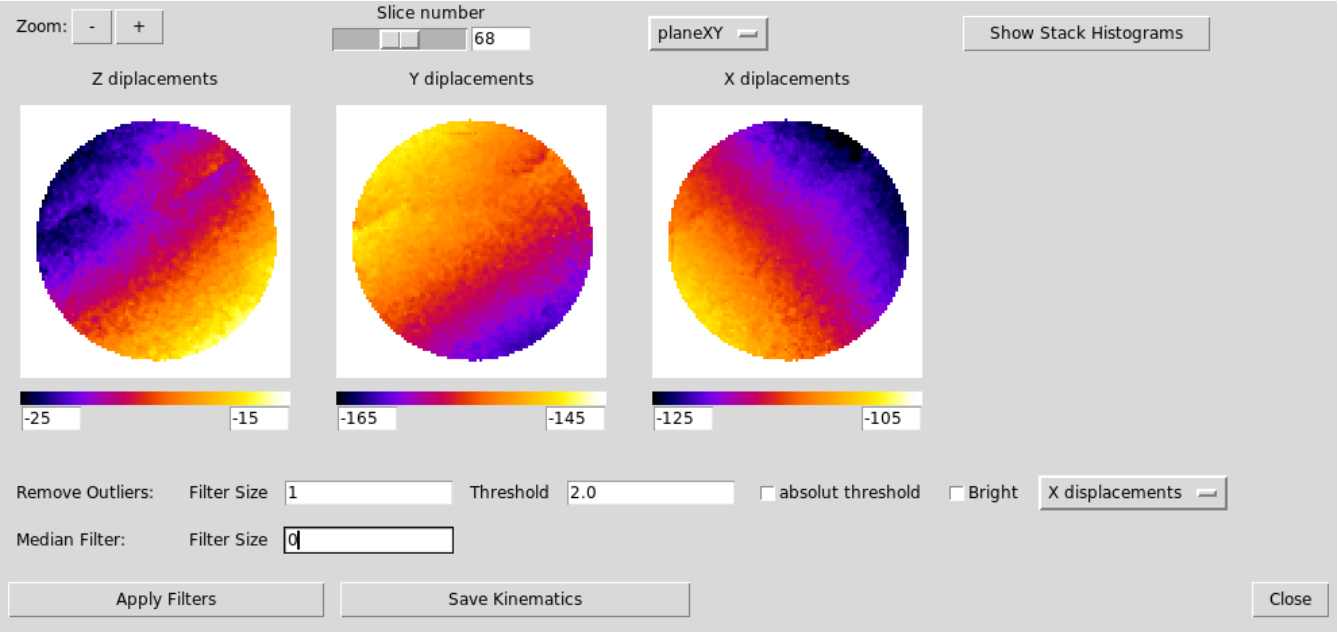


Figure 6.3: GUI - Post Process, show displacements window - Remove Outliers

Remove Outliers filter applied for each displacement component and for both bright and dark points (error points are treated as outliers)

Once the filters are applied the resultant displacement fields can be saved on a new TSV file pressing the **Save Kinematics** (the appended string "_filtered" is automatically suggested)

Commands on the first row allows the control of:

Zoom regulate the size of the image

Slice number allow to slide through the volume

Slice orientation menu volume can be sliced in three direction (figure 6.4 shows vertical slices on YZ plane)

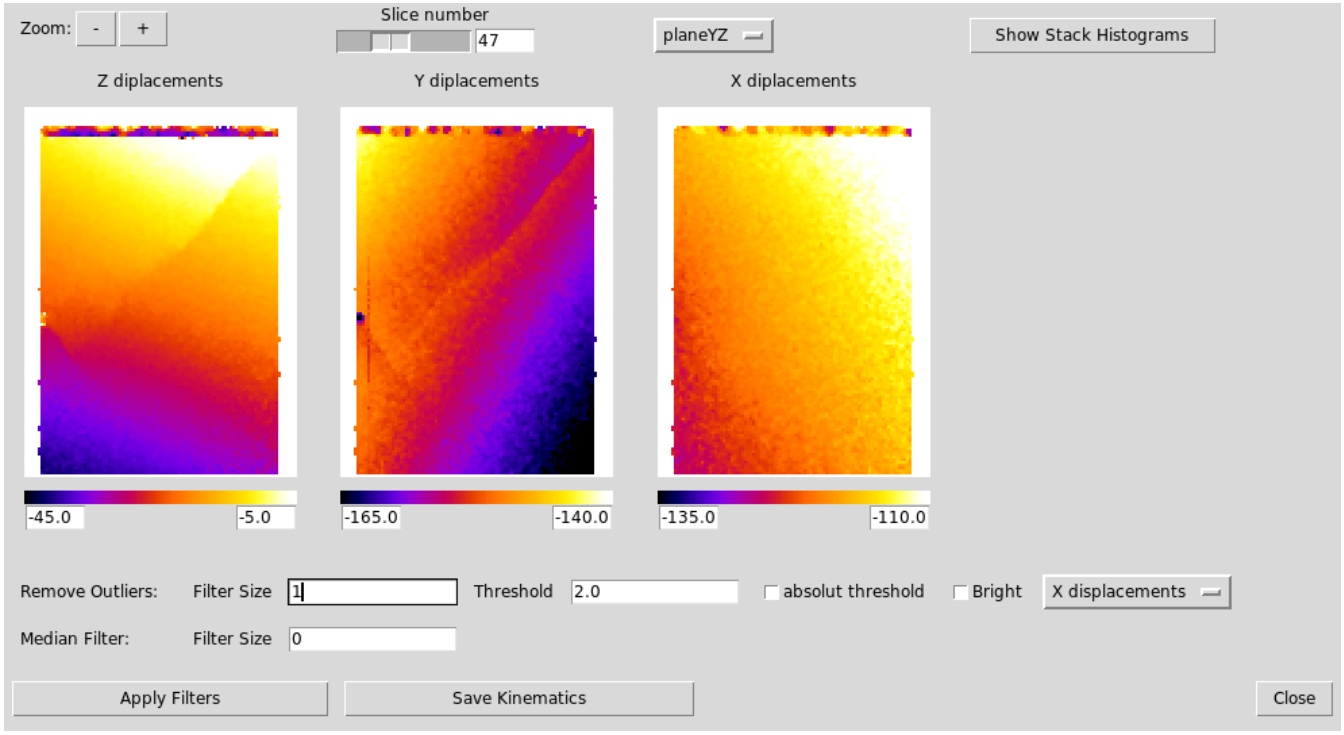


Figure 6.4: GUI - Post Process, show displacements window - vertical slices

Show Stack Histograms bottom will open a new window (see Figure 6.5). X axis limits are controlled by the same text entry that set the contrast of the images.

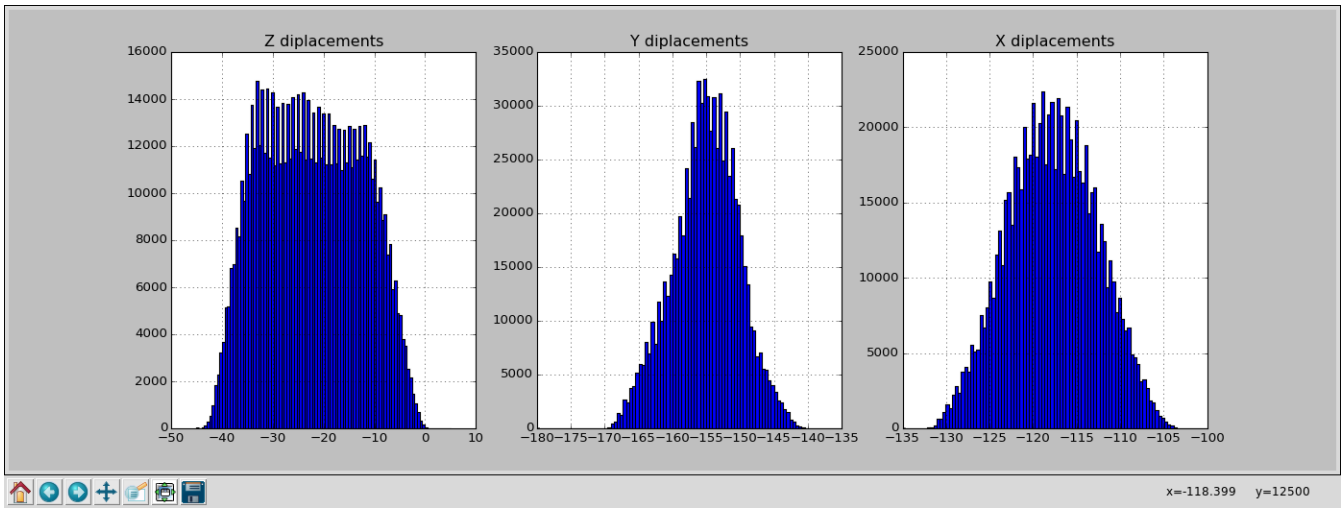


Figure 6.5: GUI - Post Process, histograms window

Close bottom will close the window. The changes applied will remain in memory. To run the post process analysis on the original displacement field the TSV file needs to be reloaded.