Python DIC Documentation

Charlie Bourigault
bourigault.charlie@gmail.com

September 2016

Abstract

This document constitutes a detailed guide of the Python digital image correlation software from Fraunhofer IWM. Fonctions and capabilities of the software are presented along with various example to help you familiarize with the application. For any other information or any comment on the software please contact the author or visit the GitHub repository (Python_DIC) on https://github.com/.

Contents

Introduction **3**

1 Installation Instructions **4**

1.1 Windows Systems 4

1.1.1 Download and Install Miniconda3 4

1.1.2 Install Required Libraries 4

1.1.3 Start Python DIC 5

1.2 Unix Systems 5

1.2.1 Download and Install Miniconda3 5

1.2.2 Install Required Libraries 6

1.2.3 Start Python DIC 6

2 Interface **7**

2.1 New Analysis 7

2.1.1 Grid tools and information 8

2.1.2 Shift correction, filters and file importation 10

2.2 Data Analysis - Visualisation 12

3 Cleaning Procedures and Options **14**

3.1 Grid Management 14

3.2 Mask Images 15

3.3 Mask Markers 16

3.4 Displacement versus position 17

3.5 Relative neighbors displacement 18

4 Further information **20**

4.1 Profiles 20

4.2 Analysis Information 21

4.3 Neighbors and New Coordinates 21

4.4 File structure 22

Introduction

Digital Image Correlation (or DIC) is used in several domains and mainly in engineering and sciences. It employs different techniques to track markers placed on images and provides accurate measurement results of changes in those images. Users can acquire displacements, strains fields along with other information in a simplified manner.

The PythonDIC software comes with a variety of functions and algorithms to accompany you through the process of image correlation. This documentation offers an overview of the software capabilities.

Software updates are released regularly and this documentation may not be fully comprehensive with upcoming functions. Please visit the github project (Python_DIC) to get last updates information.

1 Installation Instructions

The Python_DIC application is cross-platform and have been tested on Window 7, MacOS (10.11) and Linux (Ubuntu 16.4). It runs on Python 3.x and requires the installation of some external packages. We recommend you to stick with the following installation procedure to avoid any malfunction or unintended crashes of the software.

Required Packages:

- matplotlib
- scipy
- opencv3
- pandas

Miniconda3 is used to manage and simplify the Python installation process.

1.1 Windows Systems

1.1.1 Download and Install Miniconda3

In order to run Python on your computer, a Python environment needs to be installed. Using a package manager as Miniconda (a light version of Anaconda) simplify the process and provides a good management tool for external libraries.

Visit *http://conda.pydata.org/miniconda.html* and download the last Miniconda 3 (Python 3.x 64 bits) package for Windows. Follow then the installation instructions.

1.1.2 Install Required Libraries

Several libraries are compulsory for the Python DIC software to run properly. The installation process using the Miniconda environment is simplified and convenient. Here is how it works.

- Open the Command Window (Start > Search > cmd)
- Type in : *conda install matplotlib*
- Follow the instruction to install the package
- Do the same with the *scipy* and *pandas* libraries

OpenCV3 is a library which also needs to be installed. The release available on the default conda channel is not compatible with the latest Python3.x version when I'm writing these lines. However, the correct version of OpenCV have been compiled by different people and is available on different channels.

The OpenCV3 compiled version for Windows systems is available on the *menpo* channel. Here is how to install it.

- Terminal Command : *conda install -c menpo opencv3*

In case the package is not available on this channel, use *anaconda search opencv3* to find another place to download it from. (You may need the anaconda client to search for libraries. To install it, use *conda install anaconda-client*).

Libraries have been installed. The software is ready to be started.

1.1.3 Start Python DIC

Once the Python DIC files have been download from the GitHub repository (and extracted if needed). Follow these simple steps.

- Navigate to the folder where the Python DIC files have been extracted.
- Right-click on DIC.py and select *Open With..*
- Select *Browse*
- Navigate to your Miniconda folder (Default: *C:/Users/<SessionName>/Miniconda3*)
- Chose *pythonw.exe* as a default program.

The program can also be started using the windows Command Line. Navigate to the Python DIC folder with the *cd* command and start the Python DIC program using : *python DIC.py.*

1.2 Unix Systems

1.2.1 Download and Install Miniconda3

In order to run Python on your computer, a Python environment needs to be installed. Using a package manager as Miniconda (a light version of Anaconda) simplify the process and provides a good management tool for external libraries.

Visit *http://conda.pydata.org/miniconda.html* and download the last Miniconda 3 (Python 3.x 64 bits) package for your system. Follow then the installation instructions.

1.2.2 Install Required Libraries

Several libraries are compulsory for the Python DIC software to run properly. The installation process using the Miniconda environment is simplified and convenient. Here is how it works.

- Open a Terminal window
- Type in : *conda install matplotlib*
- Follow the instruction to install the package
- Do the same with the *scipy* and *pandas* libraries

OpenCV3 is a library which also needs to be installed. The release available on the default conda channel is not compatible with the latest Python3.x version when I'm writing these lines. However, the correct version of OpenCV have been compiled by different people and is available on different channels.

The OpenCV3 compiled version for Unix systems is available on the *menpo* channel. Here is how to install it.

- Terminal Command : *conda install -c menpo opencv3*

In case the package is not available on this channel, use *anaconda search opencv3* to find another place to download it from. (You may need the anaconda client to search for libraries. To install it, use *conda install anaconda-client*).

Libraries have been installed. The software is ready to be started.

1.2.3 Start Python DIC

Once the Python DIC files have been download from the GitHub repository (and extracted if needed). Follow these simple steps.

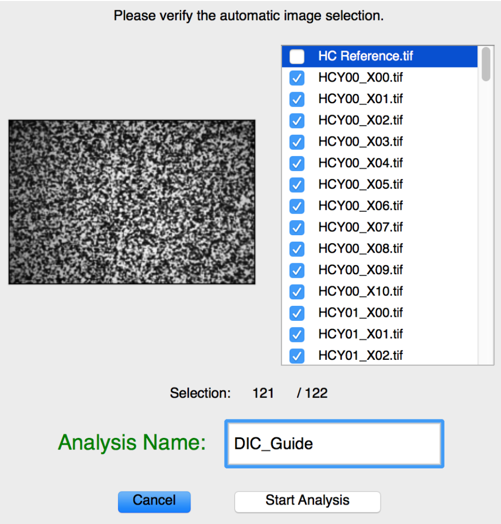
- Open a Terminal window
- Navigate to the folder containing the Python DIC *.py* files (use the *cd* command)
- Type in : *python DIC.py*

2 Interface

On startup, the Python DIC application proposes two options. One can either start a new correlation process or open a previously made analysis to access results and enhance the data if necessary.

2.1 New Analysis

When starting a new analysis, the software asks for an image to start with. Images from the same type present in the folder will be automatically selected and listed. Unselect images which shouldn't be part of the analysis, give your analysis a name and the grid creation tool will start.



Please verify the automatic image selection.

☐ HC Reference.tif
☑ HCY00_X00.tif
☑ HCY00_X01.tif
☑ HCY00_X02.tif
☑ HCY00_X03.tif
☑ HCY00_X04.tif
☑ HCY00_X05.tif
☑ HCY00_X06.tif
☑ HCY00_X07.tif
☑ HCY00_X08.tif
☑ HCY00_X09.tif
☑ HCY00_X10.tif
☑ HCY01_X00.tif
☑ HCY01_X01.tif
☑ HCY01_X02.tif

Selection: 121 / 122

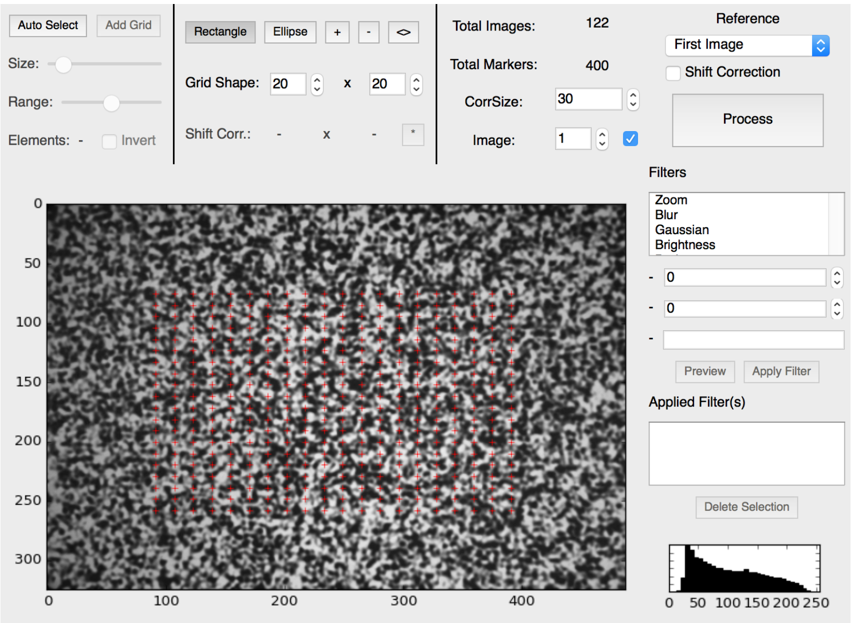
Analysis Name: DIC_Guide

Figure 1: New Analysis: Image Selection

The grid creation tool is divided into three main areas. The top part provides tools to build a grid necessary for the correlation process. Under this tools zone, the application is separated in two entities : the left side displays your images and the grid, the right side takes care of filters for your images.

2.1.1 Grid tools and information

The top part of the grid creation tool offers different means to create a grid for your analysis. It also displays information and parameters which will be used in the correlation process.



Grid Shape: 20 x 20

Total Images: 122

Total Markers: 400

CorrSize: 30

Image: 1 ☑

Reference: First Image

☐ Shift Correction

Figure 2: New Analysis: Grid Definition

Grid Creation

In order to run an analysis, a grid have to be defined. A grid is a set of markers with different coordinates which constitute the base of the correlation. When the images will be processed, the algorithm will track each marker on each image to determine the local displacement inside of your set of images.

There are three types of grid available:

- *Rectangular* : The grid shape parameters defines the number of marker in X and Y direction
- *Ellipsoidal* : The grid shape parameters defines the number of marker in X and Y direction and an ellipsoidal mask is applied
- *Single Marker (+)* : A 1x1 grid with only one marker

To create a grid, select one of the tool above, click and drag your mouse on the image at the position where you want to draw your grid.

The *(-)* tool can be used to remove one or several markers. Click and drag your mouse on the area where you want to delete markers.

A grid can be moved by using the *(<>)* tool. Click on the grid you want to move and drag it on the new location.

One or several grids can be defined. They will be saved as different entities and can later be evaluated independantly from each other.

Parameters

Different information are available before starting the correlation process. The number of images along with the number of markers created are displayed for your information. You can use the image input box to navigate through your images and individually remove some of these by unchecking the box on the right. A corrsize has to be defined to run the analysis.

The corrsize represent the size in pixel of the squared zone in which each marker will be tracked. If the corrsize is too small, the algorithm may not be able to track markers properly as they may go out of the range. However, a high corrsize value will result in high computation time. We recommend using values between 20 and 30 pixels for small displacements which gives fast and accurate results.

Three modes are available to evaluate the markers displacement:

- *First Image Reference*: The first image will be used has a reference for each image. This is the recommended mode, which gives the most accurate results. However, if displacements on an images are greater than the corrsize value, the algorithm may not be able to track markers.
- *Previous Image Reference*: For each image, the reference markers to track will be the previously tracked markers (ie: the base grid is redefined for each images to the new marker positions). This method reduce the chance of lost markers during the process. However, the displacement error is increased with the time and results can get inaccurate after a certain amount of images.
- *Shifted Image Reference*: The reference image is defined with a step. The first image will be the reference until this step is reached, then, the base grid will be redefined to the current image minus the step. This method can be used if you have big displacements after a certain number of images. You'll get accurate results for the beginning of the analysis and still keep trace of markers for the rest of images which a slighly reduced accuracy.

2.1.2 Shift correction, filters and file importation

Shift Correction

The shift correction have been introduced in the Python DIC application to face problems such as camera lateral moves. In order to reduce the amount of pixel displacements due to the camera moves, a first general image correlation is performed without markers in order to align images. Once the alignment is performed, a normal marker correlation can be performed.

The use of the shift correction may affect the accuracy of your results. It may lead to wrong marker displacement calculation and is therefore non recommended, although it can be used in extrem cases.

The shift correction is performed prior to the correlation process. The procedure to use is the following:

- Check the shift correction option
- Select a relatively large area (if possible outside of the grid location) by clicking and draging your mouse on the image
- Click on the *Track* button and wait until the end of the process
- Click on the *Confirm* button to validate

At the end of the process, the shift correction values are available for each image. Use the input text box displaying the image number to navigate in the image and review the shift correction value. You can modify these values in case of wrong recognition by clicking on the * button next to the shift values.

Uncheck the shift correction tool to cancel the procedure or to change the tracking area. Shift corrections values will be taken in account once the grid is created and the correlation process started.

Images Filters

Different filters can be applied to your set of images before any analysis. These filters allow you to improve the quality of the correlation by performing operations on your images such as contrast enhancement or noise reduction.

The Python DIC software will **not** modify any of your images. Filters are applied after the reading process and do **not** affect the original files.

Filters are simple and convenient to use:

- Select one of the available filters

- Modify parameters specific to the selected filter
- Click on the *Preview* button to have a instant view of the filter effect
- Click the *Apply Filter* button when you're satisfied to add the filter to applied filters list
- Select any filter in the applied filter list and click the *Delete Selection* button to remove the filter from the applied filters list

Note: An histogram is available and provides an instant view of the gray scale dispersion on the current image.

Grid and Filters Importation

An option have been included to import grids and filters from previously performed analysis. Under the *File* menu, you will find the *Open Grid* and *Open Filter* options.

Select the project folder containing the grid(s) and/or the filter(s) that you want to import in your analysis. The software will then look for the grid files or the filter file in this folder and use the same parameters for the current analysis.

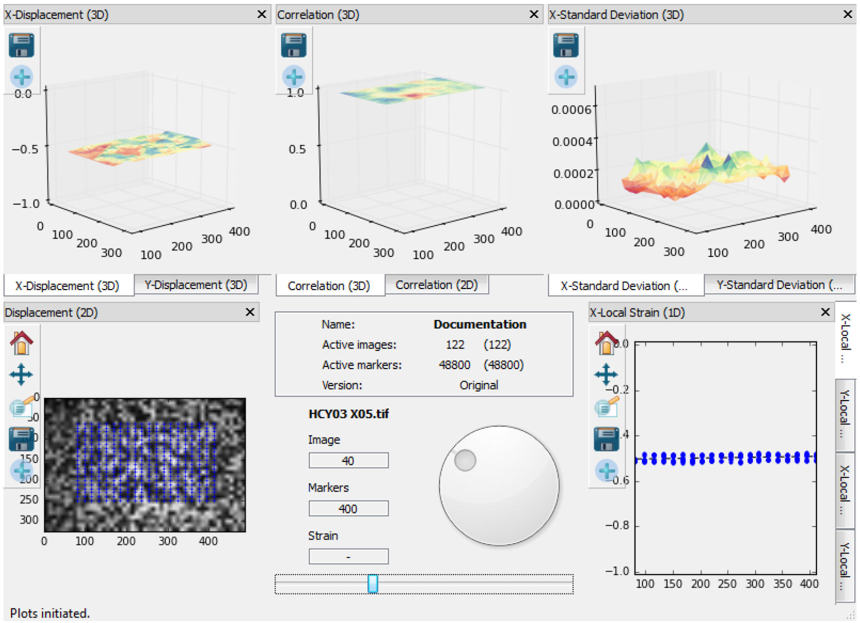
2.2 Data Analysis - Visualisation

The visualisation tool is automatically opened after a correlation process. It is also accessible on the start-up widget by clicking on the *Open Analysis* button.

There is two type of element in the visualisation window:

- A certain amount of plots
- A control area with information

The control area is fixed and displays information on the analysis and the current image selected.
You can navigate through the data using the two cursor available in the control area.



Name: Documentation
Active images: 122 (122)
Active markers: 48800 (48800)
Version: Original

HCY03 X05.tif
Image: 40
Markers: 400
Strain: -

Plots initiated.

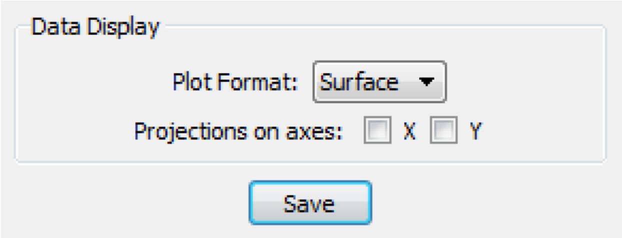
Figure 3: Visualisation : Navigation through results

Plots are located around the control area and can be re-organized by the user.

- Click and drag a plot to move it
- Close unwanted plots using the individual cross
- Open plots by right-clicking on an empty area or next to the menu

When the data is refreshed, only the visible opened plots are taken in account. For large analysis, closing unnecessary plots will sensibly decrease the refreshing time.
You can also stack plots on top of each other for a quick access. This will not affect the refreshing time as only visible plots are reloaded.

For each single plot, you can use the toolbar to save the plot as an image and zoom/pan on your data. Different display options are also available, click on the + button in the toolbar to get access to parameters.



Data Display

Plot Format: Surface

Projections on axes: □ X □ Y

Figure 4: Visualisation : 3D plots options

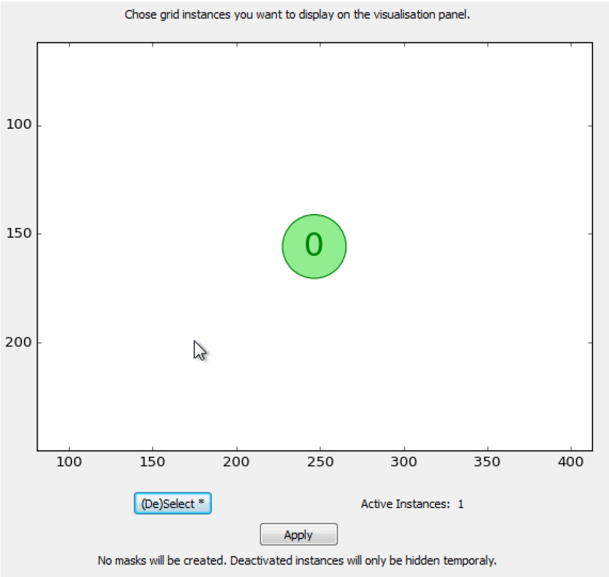
3 Cleaning Procedures and Options

3.1 Grid Management

You can perform separate grid calculation on a single analysis. To do so, draw several grids before starting a correlation in the grid creation tool. Once the calculation done, you'll be able to analyze these grid separately.

If several grids have been created, they'll be dissociated in the visualisation tool. You are then able to differentiate the strain in one grid compared to another. Using the instance management tool, you can temporaly hide one or several grids to focus on the one (or those) that you're interested in.

- Go into *Masks >Manage Grids*
- Chose the grids you want to hide/display
- Validate to apply changes



Chose grid instances you want to display on the visualisation panel.

Active Instances: 1

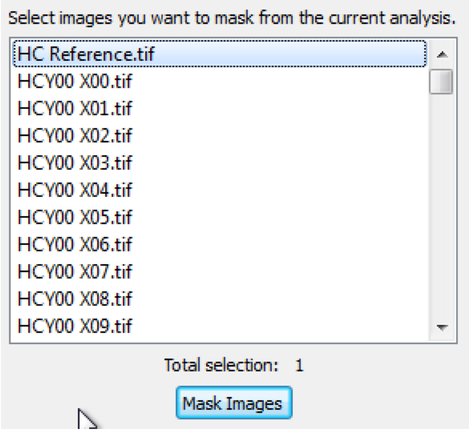
No masks will be created. Deactivated instances will only be hidden temporaly.

Figure 5: Grid Management - Active instances displayed in green

When you hide or display grid instances, the changes made are not saved and all the grid will be displayed by default on the next start-up.

3.2 Mask Images

In the *Masks* menu, you'll find the *Mask Images* feature. This option provides an easy way to remove images from your analysis. Select the images that you want to mask in your analysis to hide them from the analysis.



Select images you want to mask from the current analysis.

HC Reference.tif
HCY00 X00.tif
HCY00 X01.tif
HCY00 X02.tif
HCY00 X03.tif
HCY00 X04.tif
HCY00 X05.tif
HCY00 X06.tif
HCY00 X07.tif
HCY00 X08.tif
HCY00 X09.tif

Total selection: 1

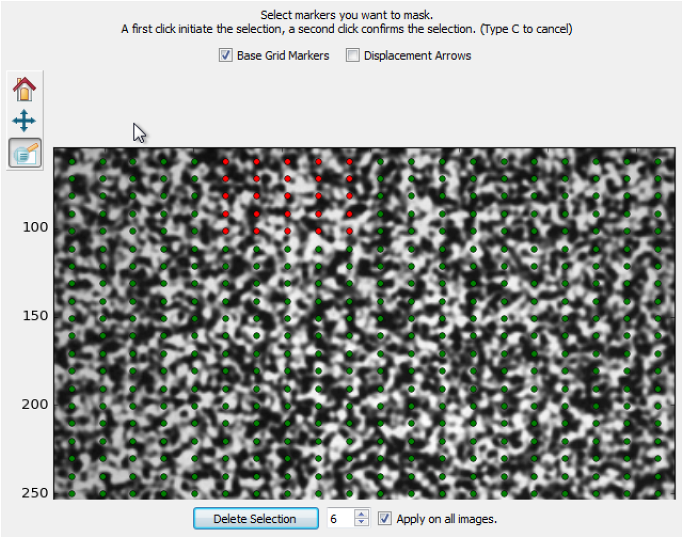
Figure 6: Mask Images - Select images to hide

When a mask is applied, the previous state of the analysis is automatically saved in the analysis folder.
On start-up, the software will load the last mask version by default. If you made a mistake or want to bring back your data, you can open a previous version by using the Open Mask/Version option in the *File* menu.

Please keep in mind that when a mask is applied, the data is only hidden and not modified. An older version can always be loaded.

3.3 Mask Markers

Using the *Mask Markers* feature will allow you to manually select badly correlated data in order to hide it from you analysis. Select the markers that do not behave as expected and apply the mask to enhance the analysis results.



Select markers you want to mask.
A first click initiate the selection, a second click confirms the selection. (Type C to cancel)

☑ Base Grid Markers ☐ Displacement Arrows

6 ☑ Apply on all images.

Figure 7: Mask Markers - Hide manually selected markers

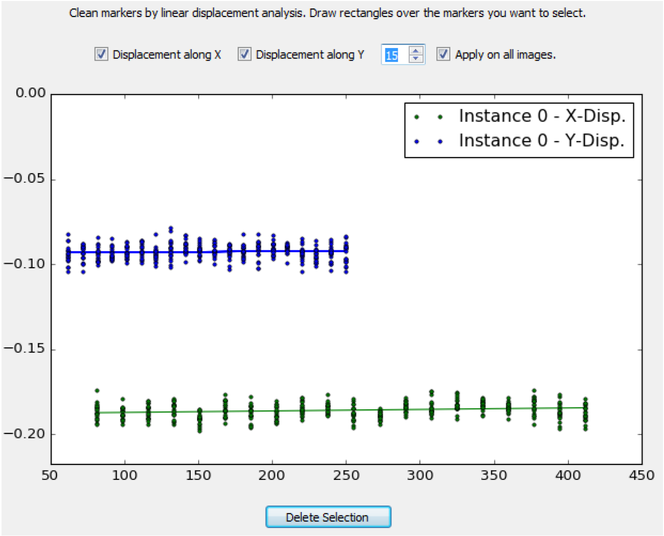
When a mask is applied, the previous state of the analysis is automatically saved in the analysis folder.
On start-up, the software will load the last mask version by default. If you made a mistake or want to bring back your data, you can open a previous version by using the Open Mask/Version option in the *File* menu.

Please keep in mind that when a mask is applied, the data is only hidden and not modified. An older version can always be loaded.

3.4 Displacement versus position

The *Disp. vs Pos.* feature from the *Masks* menu helps you to compare the displacement of each marker depending on its position in the X or Y direction. Select the markers that you would like to hide and confirm to apply the mask on your data.

This tool gives you a fast and easy way to detect wrong behaviour in your markers. All markers will in most of the case follow a certain trend in one direction and you can then locate markers with correlation issues.



Clean markers by linear displacement analysis. Draw rectangles over the markers you want to select.

☑ Displacement along X ☑ Displacement along Y 15 ☑ Apply on all images.

Figure 8: Disp. vs Pos. - Hide markers with wrong behaviour

When a mask is applied, the previous state of the analysis is automatically saved in the analysis folder.
On start-up, the software will load the last mask version by default. If you made a mistake or want to bring back your data, you can open a previous version by using the Open Mask/Version option in the *File* menu.

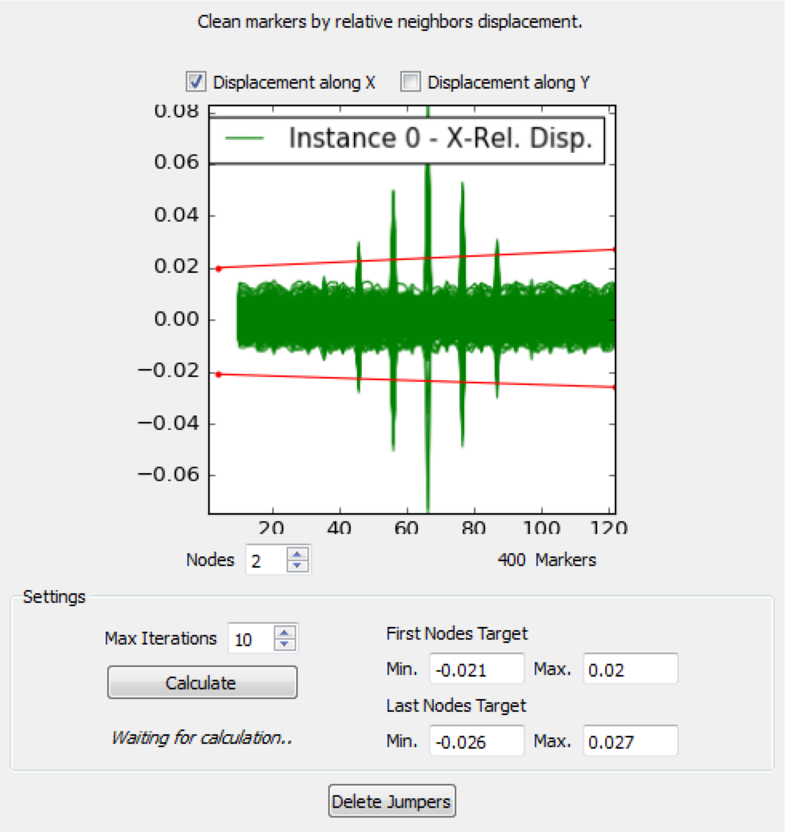
Please keep in mind that when a mask is applied, the data is only hidden and not modified. An older version can always be loaded.

3.5 Relative neighbors displacement

The *Relative neighbors disp.* tool offers you a powerful way to automatically clean your data.

The algorithm will provide you a graph displaying the relative displacement of each marker compared to his closest neighbors during the whole correlation process. A marker with high displacements compared to its neighbors may be badly correlated and should be masked. You can also detect when a marker starts to behave differently compared to its neighbors.

In order to masked the unwanted markers, move the two upper and lower red lines on the graph representing the maximum relative displacement for each marker. If a marker cross the upper or lower limit, it'll automatically be masked.



Clean markers by relative neighbors displacement.

☑ Displacement along X ☐ Displacement along Y

Nodes 2

400 Markers

Settings

Max Iterations 10

Waiting for calculation..

First Nodes Target

Min. -0.021 Max. 0.02

Last Nodes Target

Min. -0.026 Max. 0.027

Figure 9: Relative Neighbors Disp. - Define upper and lower limits

A masked marker may generate new high relative displacement in the new neighborhood of some markers. The algorithm will automatically recalculate the relative displacement after a deletion and continue the cleaning if some markers are then crossing the predefined upper and lower limits. Once all the unwanted have been masked (or the iteration number is reached), the new graph is displayed and you can then confirm the calculation to apply the mask to your data or set new limits.

When a mask is applied, the previous state of the analysis is automatically saved in the analysis folder.
On start-up, the software will load the last mask version by default. If you made a mistake or want to bring back your data, you can open a previous version by using the Open Mask/Version option in the *File* menu.

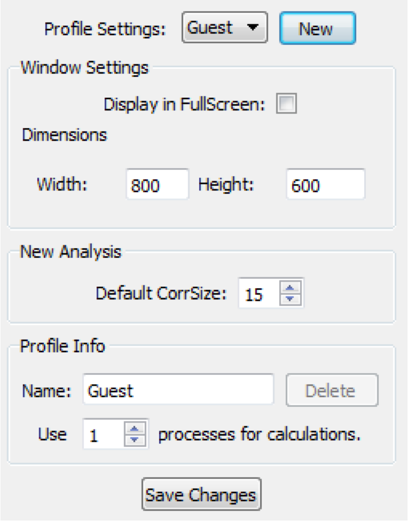
Please keep in mind that when a mask is applied, the data is only hidden and not modified. An older version can always be loaded.

4 Further information

4.1 Profiles

Under the *File* menu, you'll find a profile management tool to register and configure different options relative to your analysis. Different parameters such as the default corrsize or the amount of processor cores to use can be stored and loaded easily when the application is started.

When selecting an existing profile, it'll be set as default and will automatically be loaded on the next start of the application.



Profile Settings: Guest

Window Settings

Display in FullScreen: □

Dimensions

Width: 800 Height: 600

New Analysis

Default CorrSize: 15

Profile Info

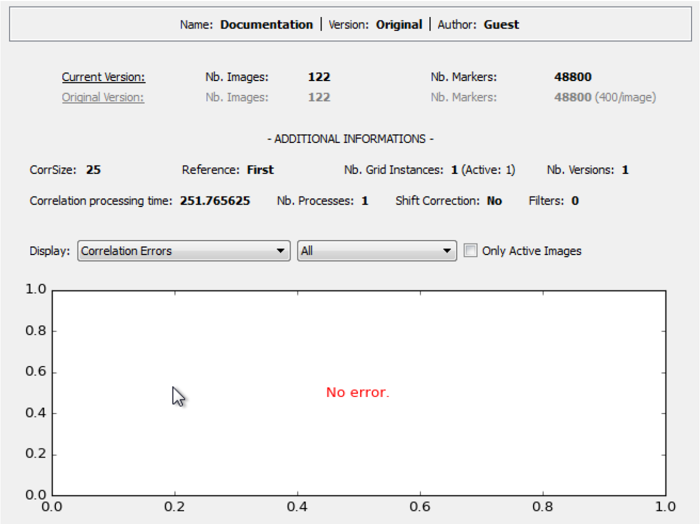
Name: Guest

Use 1 processes for calculations.

Figure 10: Profile - Define your own preferences

4.2 Analysis Information

Under the *Info* menu, you'll find the analysis information window. This provides you a summary of parameters you've used along with others information regarding your current analysis. One can display data graphs representing different information such as errors encountered during correlation or the current poisson ratio calculated on each image.



Name: **Documentation** | Version: **Original** | Author: **Guest**

Current Version: Nb. Images: **122** Nb. Markers: **48800**

Original Version: Nb. Images: **122** Nb. Markers: **48800** (400/image)

- ADDITIONAL INFORMATIONS -

CorrSize: **25** Reference: **First** Nb. Grid Instances: **1** (Active: 1) Nb. Versions: **1**

Correlation processing time: **251.765625** Nb. Processes: **1** Shift Correction: **No** Filters: **0**

Display: Correlation Errors ▾ All ▾ □ Only Active Images

No error.

Figure 11: Analysis Information : Parameters and more

4.3 Neighbors and New Coordinates

In the *More* menu, you'll find the *Recalculate Neighbors* and *New Coordinates* options.

The *Recalculate Neighbors* option is useful when you're using the *Relative Neighbors Cleaning* tool. It allows you to re-calculate each marker neighbors when masks have been applied. You can chose how many neighbors should be calculated for each marker. However, keep in mind that an important number of neighbors may significantly increase the processing time when calculating new coordinates or cleaning markers depending on relative neighbors. Between 10 and 40 neighbors seems to be a reasonnable value for analysis with 100 to 1000 markers per image.

The *New Coordinates* option have been developped to re-calculate coordinates when new masks have been applied. It is only useful if you're interested in the 2D mapped plots (2D Strain, 2D correlation coefficient, ...). You can select the coordinates that you want to update. Depending on your number of images and markers, this can take some time, that's why we recommend to re-calculate only the data you're interested in (for example 2D strain).

4.4 File structure

The main information on each correlation process is stored in CSV format. For each analysis, the following files will be created:

- *filenamelist.csv* : This file contains names of your images files
- *gridx.csv* - *gridy.csv* : These files store respectively the x and y coordinates of each marker in your grids along with their instance number.
- *validx.csv* - *validy.csv* : These files store respectively the x and y coordinates of each marker of your analysis. Each line corresponds to a marker, each column corresponds to an image.
- *dispx.csv* - *dispy.csv* : These files store respectively the x and y displacements in pixel of each marker of your analysis. Each line corresponds to a marker, each column corresponds to an image.
- *stdx.csv* - *stdy.csv* : These files store respectively the x and y standard deviation in pixel of each marker of your analysis. Each line corresponds to a marker, each column corresponds to an image.
- *corrcoef.csv* : This file stores the correlation coefficient or each marker of your analysis. Each line correspond to a marker, each column corresponds to an image.
- *neighbors.csv* : This file stores the closest neighbors of each marker. Each line correspond to a marker. This file is updated when neighbors are re-calculated.
- *infoMarkers.csv* and *infoAnalysis.csv* : These files store informations on your analysis such as correlation errors and parameters used.
- *strainx.csv* and *strainy.csv* : These files store the global strain values of each grid along images. Each line corresponds to the image, each column correspond to a grid. These two files are automatically updated when coordinates are re-calculated.
- *coordinates.csv* (After first start) : This file contains calculated coordinates for the 2D mapped plots. They're stored to save time during start-up and may not be up-to-date if the user decided not to re-calculate them after the cleaning process. An option is available in the *More* menu to re-calculate these coordinates at any time.

- *filter.csv* (Optional) : This file stores information relative to filters applied.
- *largeDisp.csv* (Optional) : This file stores the calculated large displacement on your analysis in case the shift correction have been used.

Masks

Masks (or versions) are saved in the *log* folder of each analysis. This folder is automatically created when the first mask is applied. A mask is then saved whenever markers or images are hide on the analysis. You can always load an old version of your analysis by importing one of these masks.