PaIRS

Particle Image Reconstruction Software

User Guide

PaIRS – version 0.1.13

©2024 Gerardo Paolillo & Tommaso Astarita.
All rights reserved.

email: etfd@unina.it

Experimental Thermo-Fluid Dynamics (ETFD) group,
Department of Industrial Engineering (DII)
University of Naples "Federico II", Naples, Italy

UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

DIPARTIMENTO DI
INGEGNERIA
INDUSTRIALE

Contents

1. Introduction

PaIRS-UniNa is a project developed by the Experimental Thermo Fluid-Dynamics (ETFD) group of University of Naples "Federico II" since 2000. It is aimed to provide fast and efficient tools for digital particle image velocimetry (PIV) analysis in research and industrial applications.

PaIRS-UniNa is based on a C library (**PaIRS-PIV**) and relies on a graphical user interface (**PaIRS**) that is developed via PySide6 and makes the use of PaIRS-PIV easy and intuitive. PaIRS-PIV includes several modules that allow to process double-frame or single-frame 2D planar PIV images as well as stereoscopic and tomographic PIV or Lagrangian particle tracking velocimetry (4D PTV) measurements. The current release of PaIRS-UniNa features only the module for the 2D planar PIV analysis.

PaIRS-UniNa is supported by Python 3.8+ and is compatible with all the operating systems, however, the PaIRS-PIV library relies on OpemMP library, which must be installed on the macOS platform (see section Installation). On the other side, PaIRS requires, among other packages, SciPy and matplotlib.

2. Related works

Please cite the following works if you are intended to use PaIRS-UniNa for your purposes:

[1] Astarita, T., & Cardone, G. (2005). "Analysis of interpolation schemes for image deformation methods in PIV". *Experiments in Fluids*, 38(2), 233-243. doi: 10.1007/s00348-004-0902-3.

[2] Astarita, T. (2006). "Analysis of interpolation schemes for image deformation methods in PIV: effect of noise on the accuracy and spatial resolution". *Experiments in Fluids*, vol. 40 (6): 977-987. doi: 10.1007/s00348-006-0139-4.

[3] Astarita, T. (2007). "Analysis of weighting windows for image deformation methods in PIV." *Experiments in Fluids*, 43(6), 859-872. doi: 10.1007/s00348-007-0314-2.

[4] Astarita, T. (2008). "Analysis of velocity interpolation schemes for image deformation methods in PIV". *Experiments in Fluids*, 45(2), 257-266. doi: 10.1007/s00348-008-0475-7.

[5] Astarita, T. (2009). "Adaptive space resolution for PIV". *Experiments in Fluids*, 46(6), 1115-1123. doi: 10.1007/s00348-009-0618-5.

3. Installation

All PaIRS-UniNa wheels are distributed under LGPLv3+ licences. For PaIRS installation, you must first download and install Python. You can follow the tutorial at https://realpython.com/installing-python/ to install or update the official Python distributions on Windows, macOS or Linux. After that, the PaIRS installation can be performed from a command-line application with:

```
python -m pip install PaIRS-UniNa
```

On macOS and Linux `python` must be replaced by `python3`.
Normally the OpenMP library is not preinstalled in MacOs. A possible way to install this library is:

```
curl -O https://mac.r-project.org/openmp/openmp-12.0.1-darwin20-Release.tar.gz
sudo tar fvxz openmp-12.0.1-darwin20-Release.tar.gz -C /
```

Please, notice that on Windows 7 you should install Python 3.8 and PySide 6.1.3 version. For this purpose, after the Python and PaIRS installations (as above explained), execute the following commands:

```
python -m pip uninstall PySide6
python -m pip install PySide6==6.1.3
```

4. Run

From command prompt

It is possible to run PaIRS directly from the command prompt with:

```
python -m PaIRS_UniNa
```

PaIRS automatically saves and stores its configuration upon exit and starts from the latter at the next run. If any trouble with loading the last configuration file (saved in the package folder) occurs, the user is suggested to execute a clean run of PaIRS via the following command:

```
python -m PaIRS_UniNa -c
```

A debug mode is also available for developers. It can be accessed via:

```
python -m PaIRS_UniNa -d
```

After the above command, the user will be asked to enter a password. Interested users can ask the password to the authors by sending an email to: etfd@unina.it. The debug mode can be turned on/off at any time via the keyboard sequence: `Alt+Shift+D`.

On macOS and Linux `python` must be replaced by `python3`.

In Python environment

PaIRS can be run also in Python environment. For this purpose, open a command prompt and launch Python:

```
python
```

and then use the following commands for normal run:

```
>>> from PaIRS_UniNa import PaIRS
>>> PaIRS.run()
```

For clean mode:

```
>>> PaIRS.cleanRun()
```

while for debug mode:

```
>>> PaIRS.debugRun()
```

5. Main Window

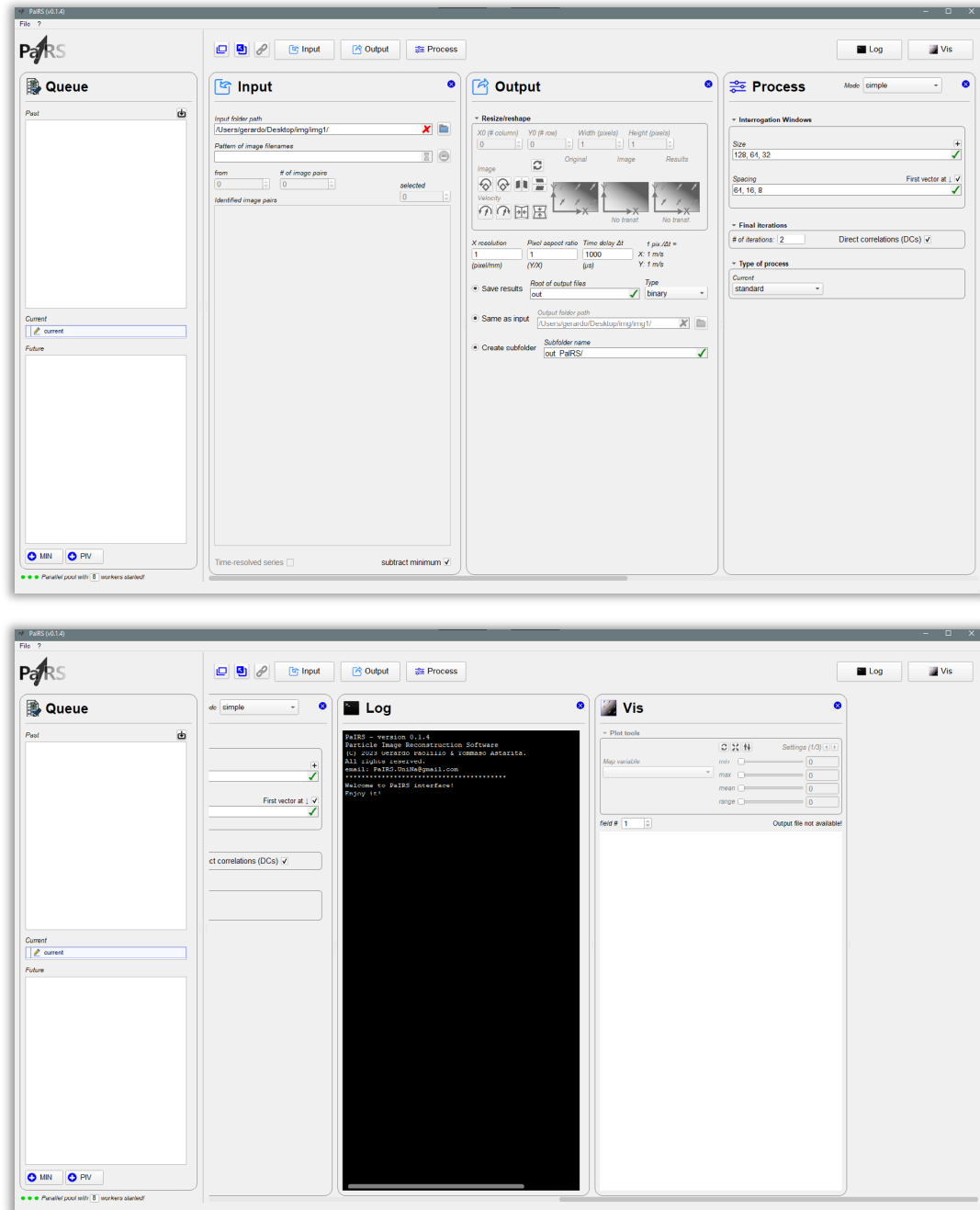PaIRS main window looks like in Figure 1.

Figure 1 – PaIRS main window.

Several tabs are included in the main window and allow the user to specify the image and process settings before launching an instance of image preprocessing or a PIV analysis. The Queue tab allows to prepare a list of process that can be subsequently launched sequentially. The Input, Output and Process tabs are used to specify the parameters of the image files and the output files produced by PaIRS (including their full paths) and the settings of the PIV process to be performed. The Log tab is used to visualize output from the PaIRS-PIV library and further error/warning messages, while the Vis tab can help user in visualizing the input images and the results from image preprocessing and/or PIV analyses. Finally, the Button bar is used to hide/show or dock/undock the other above-mentioned tabs and to manage the interface geometry. PaIRS has also a Menu bar, where further functions are provided.

6. Menu bar

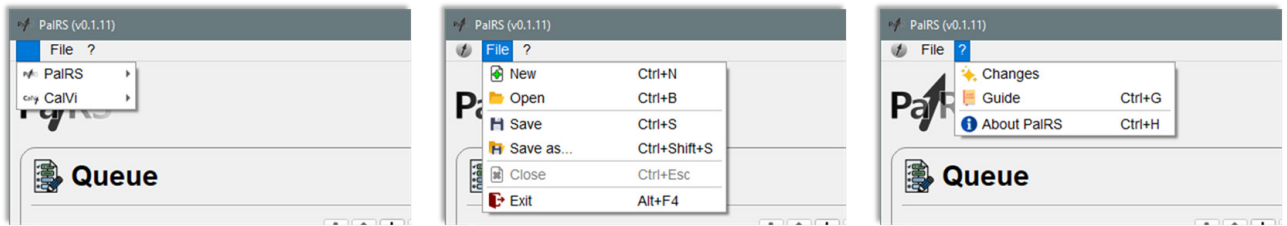The Menu bar consists of the section 🧭, File and Help (?), as show in Figure 2.

Figure 2 – Menu bar.

🧭

From this section, new instances of PaIRS and CalVi can be launched:

- 🚀 *Normal run*: launch a new instance of PaIRS/CalVi starting from the last saved configuration;
- ✳ *Clean run*: launch a new instance of PaIRS/CalVi in clean mode;
- 🐛 *Debug run*: launch a new instance of PaIRS/CalVi in debug mode.

File

The following functions are provided:

- ➕ *New* (`Ctrl+N`)[1]: save the current project and start a new one saved to a specified location;
- 📂 *Open* (`Ctrl+B`): load a previously saved project (extension of the file is '.pairs_cfg');
- 💾 *Save* (`Ctrl+S`): save the current project;
- 💾 *Save as* (`Ctrl+Shift+S`): save the current project to a specified location (extension of the file is '.pairs_cfg');
- ❎ *Close* (`Ctrl+Esc`)*:* close the current project and turn the interface to its initial state;
- 🚪 *Exit* (`Alt+F4`): exit the application.

Help (?)

The following functions are provided:

- ✨ *Changes*: relevant updates with respect to the previous versions of PaIRS-UniNa are shown;
- 📙 *Guide* (`Ctrl+G`): if you are reading this document, you well know this function;
- ℹ️ *About PaIRS* (`Ctrl+H`): display information about the release of PaIRS in use and the authors and their work.

---

[1] Please, notice the Macintosh equivalent of the Windows/Linux `Ctrl` key is the ⌘`(Command)` key

7. Queue

The Queue tab (Figure 3) consists of three trees:

1. *Past*: here it is possible to access old processes to inspect both input parameters and results (if available). Each item in this tree is associated with an icon according to the following:

Figure 3 – The Queue tab.

- ✅ : the process was completed without errors and the corresponding results are available;
- 🏳️ : the process was completed with errors and the corresponding results and error details are available;
- ❌ : the process is out-of-date (it was removed from the disk or overwritten). The related Log can be still read and images visualized (if existing). It would be better to remove the corresponding item from the past queue;
- 🛑 : the process was paused and then removed from the *Future* queue before completion. The corresponding partial results are available;
- 🗑 : the process was deleted from the *Future* queue before being launched.
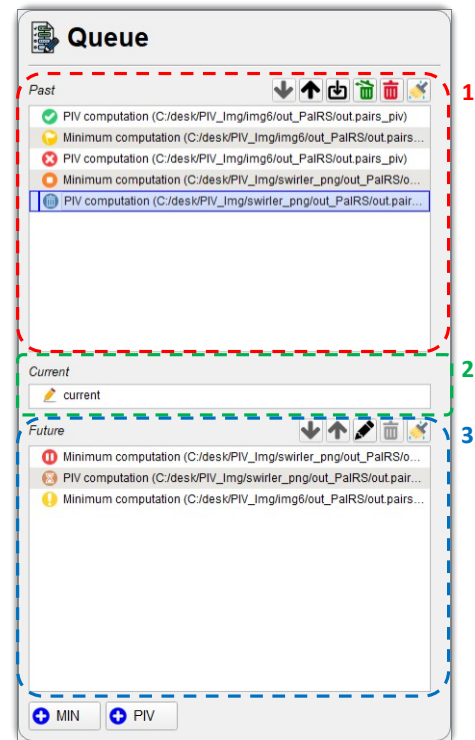
The user can act on items in the *Past* queue via the following buttons:

⬇️ ⬆️ Move items in the list down/up thus changing the list sorting (`Ctrl+Down`, `Ctrl+Up`).

📥 Import a process saved in the disk (`Ctrl+D`).

♻️ Resume the item and move it to the *Future* queue (`Ctrl+R`).

🗑 Delete the item permanently (`Delete` or `Backspace`).

🗑 Delete the item without generating warning messages (`Shift+Delete` or `Shift+Backspace`). This button is replaced to 🗑 if the Shift key is pressed.

🧹 Clean the whole queue permanently (`Ctrl+Shift+T`).

Apart from the reported keyboard shortcuts, the above actions can be accessed and executed also from the context menu of each item generated by right-clicking on it.

2. *Current*: here it is possible to access the current settings. The icon of the *Current* item is generally ✏️ unless an item in the *Future* queue is being modified. ✏️ indicates that any interaction of the user with the interface widgets results in a change of the *Current* item, even though a different process (item) is selected in the Queue tab and presently visualized. This allows the user to set a process in the *Past* or *Future* queue as "current" by simply modifying one of the process settings that are not included in the Vis tab. The changes in the Vis tab always affect only the selected process in the Queue tab (never the *Current* item);

3. *Future:* here it is possible to access processes that have been setup for future running, thus checking their settings. Each item in this tree is associated with an icon according to the following:

- ⏳ : the process has configured but not launched yet;

7

⚠️   : the process involves some issue that the user should revise in the Log tab before running;

⚙️   : the process has been launched and it is running;

⏸️   : the process has been launched and then paused.

The user can act on items in the *Future* queue via the following buttons:

⬇️ ⬆️   Move items in the list down/up thus changing the list sorting (`Ctrl+Down`, `Ctrl+Up`).

✏️   Edit the current process (`F2`). This is possible only for processes that have not been launched yet. When this button is pressed, any interaction of the user with the interface widgets results in a change of the selected item and <u>not</u> of the *Current* item. This is also indicated by the icon which is temporarily replaced by the editing icon 🖊️ . When the user selects another item in the Queue tab or presses this button a second time, editing is associated again with the *Current* item.

🗑️   Delete the item and move it to the *Past* queue (`Delete` or `Backspace`).

🧹   Clean the whole queue permanently (`Ctrl+T`).

Apart from the reported keyboard shortcuts, the above actions can be accessed and executed also from the context menu of each item generated by right-clicking on it.

New processes can be added to the *Future* queue via the underlying buttons:

➕ MIN   Add a historical minimum background image computation (`Ctrl+M`).

➕ PIV   Add a PIV analysis (`Ctrl+,`).

It is worth remarking that when a process is selected in any queue and modified, the current process in the *Current* tree is updated, unless editing mode has been set for an item in the *Future* queue by pressing the button ✏️ . Only the items corresponding to future processes that have not been launched yet are editable. For such editable items, including the *Current* item, it is possible to access previous configurations of settings using the Undo/Redo buttons located in the header of the Queue tab:

↩️   Undo (`Ctrl+Z`): undo last changes and access last configuration of settings before the present one.

↪️   Redo (`Ctrl+Y`): redo last changes and access last configuration of settings after the present one.

As shown in Figure 4, by right-clicking on the above buttons, it is also possible to switch to a previous or next configuration of settings by selecting it from the generated context menu. In the latter, only 10 configurations per time are shown, for brevity.

Please note: the undo/redo queues are lost after closing PaIRS (this helps saving disk space!). Therefore, if you want to store a configuration of settings, the best practice is to add it as minimum background image computation or a PIV analysis process to the *Future* queue using the related buttons and then delete it, thus moving it to the *Past* queue as a refuse item (🗑️).

In conclusion, it is also noticed that while a completed (✅) process cannot be restored from the *Past* queue, it is however possible to select it, modify it (thus setting it a *Current* item) and then add a new minimum computation process or a new PIV analysis using the buttons under the *Future* queue.
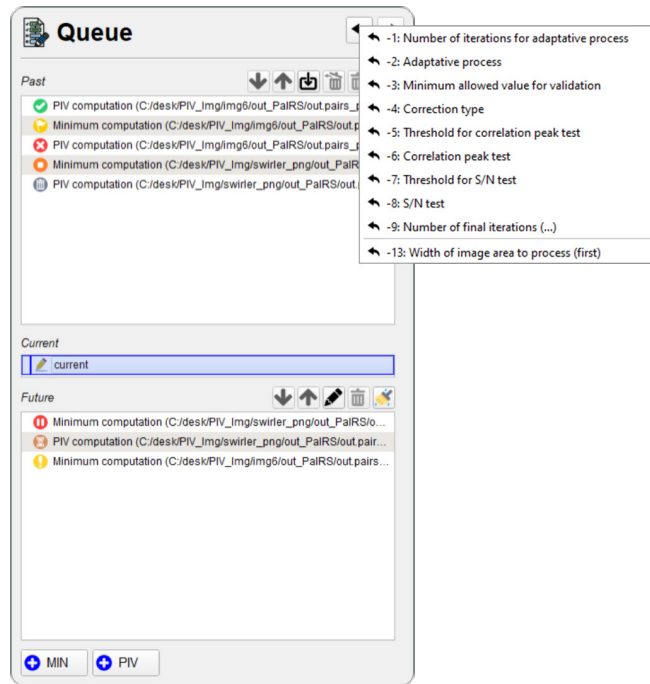
Figure 4 – Undo list.

Useful hint

To move the focus to the Queue tab from the keyboard, use the key sequence: `Ctrl+W`.

8. Button bar

The Button bar (Figure 5) is used to hide/show or dock/undock the Input, Output, Process, Log and Vis tabs. By clicking on the corresponding button, it is possible to show one of these tabs and/or focus on it. The tabs currently hidden are denoted by a padlock symbol (🔒).
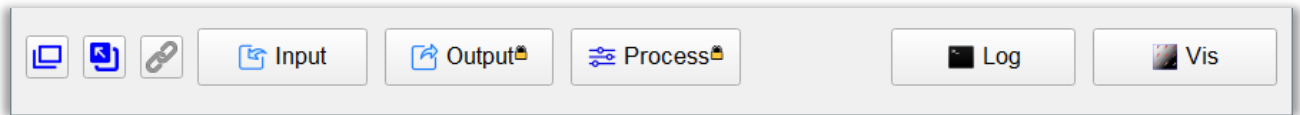
Figure 5 – Button bar.

The following buttons are present in the bar:

Change sizes (`Ctrl+Space`): it opens a popup where different geometry configurations can be selected:

Default geometry configuration (a clean run of PaIRS starts with it) (`0`).

Geometry featuring Queue and only one of the remaining tabs (`1`).

Geometry featuring Queue and two of the remaining tabs (`2`).

Geometry featuring Queue and three of the remaining tabs (full screen) (`3`).

Geometry suggested during processing (`R`).

Last user-defined geometry (`Backspace`).

Undock/Dock button (`Ctrl+U`): by clicking on it, it is possible to unpack the main window in separated windows each containing a different tab.

Join/release button (`Ctrl+J`): enabled only when the tabs are docked to the main window, it allows to visualize one at a time (joined mode) or simultaneously (released mode) the Input, Output and Process tabs.
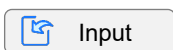
Show/focus on Input (`Ctrl+I`). Input can be closed clicking on ⊗ (`Alt+I`).
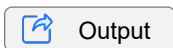
Show/focus on Output (`Ctrl+O`). Output can be closed clicking on ⊗ (`Alt+O`).

Show/focus on Process (`Ctrl+P`). Process can be closed clicking on ⊗ (`Alt+P`).

Show/focus on Log (`Ctrl+L`). Log can be closed clicking on ⊗ (`Alt+L`).

Show/focus on Vis (`Ctrl+K`). Vis can be closed clicking on ⊗ (`Alt+K`).

9. Input

The Input tab allows the user to specify the set of images to process. Images are opened via Pillow, a Python package that supports over than 30 different file formats. For more details, visit the webpage https://pillow.readthedocs.io/en/stable/handbook/image-file-formats.html.

The image files related to the same set must be contained in the same folder. Image filenames must include a sequential number (sequence of digits of arbitrary length) preceded or followed by a letter identifying the frame. Multi-frame image sequences can be handled; however, the frames are ordered in an alphabetical way according to the frame identifier letter. Single-frame sequences are also admitted. In the latter case, the sequential number is not expected to be preceded or followed by a frame identifier. Examples of valid sets of image filenames are reported below:

Set of double-frame images:	Set of triple-frame images:	Set of single-frame images:
`test00000a.png`	`img_A00.jpg`	`Jet_010.tif`
`test00000b.png`	`img_B00.jpg`	`Jet_011.tif`
`test00001a.png`	`img_C00.jpg`	`Jet_012.tif`
`test00001b.png`	`img_A01.jpg`	`Jet_013.tif`
`test00002a.png`	`img_B01.jpg`	`Jet_014.tif`
`test00002b.png`	`img_C01.jpg`	`Jet_015.tif`
`test00003a.png`	`img_A02.jpg`	`Jet_016.tif`
`test00003b.png`	`img_B02.jpg`	`Jet_017.tif`
`test00004a.png`	`img_C02.jpg`	`Jet_018.tif`
…	…	…

(the files are not ordered in an alphabetical way)

The user can specify the full path of the image files in one of the following ways:

1) (*not preferable*) locating the path of the folder containing the image files and choosing the correct pattern among those identified automatically by PaIRS (Figure 6): the path can be located either by inserting its name in the corresponding line text box (*Input folder path*) or clicking the button ⬛ and using the dialog window generated. In both cases, PaIRS analyzes the content of the input folder and automatically detects the potential patterns of the image sets contained in it (this operation can take some seconds depending on the location and content of the folder). By clicking on the below line text box, named *Pattern of image filenames,* the list of the identified patterns pops up and it is possible to select the desired one and then press the Return key to set it as the current one.
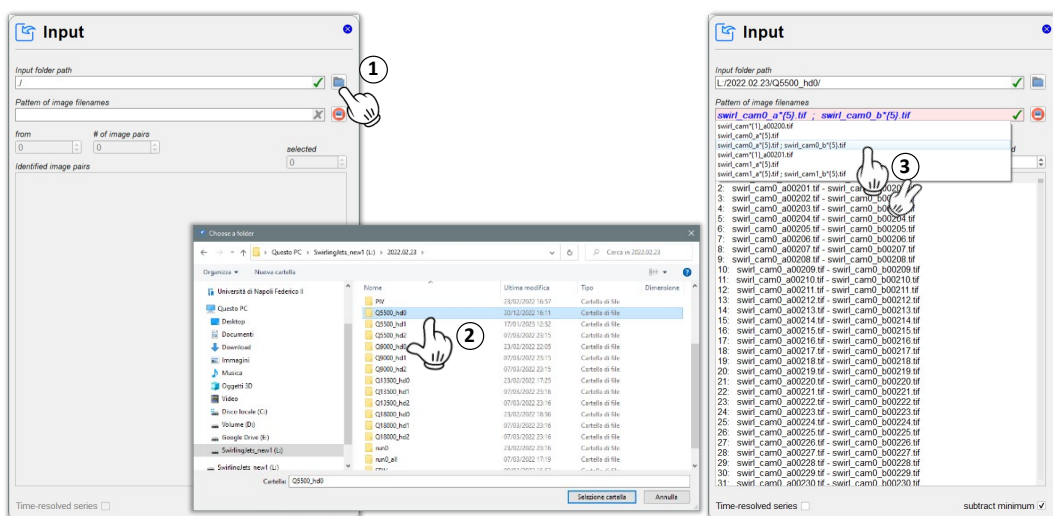
Figure 6 – Steps for selecting the image set by setting the folder path and choosing the correct pattern among those identified by PaIRS automatically.

If the pattern is not in the list, the user should manually insert it in the line text box *Pattern of image filenames* or try the method 2) described below. The rules for specifying a valid pattern are reported in the following subsection and should be rigorously applied in case PaIRS fails in detecting the target image set;

2) (*preferable*) picking directly one image file of the sequence (Figure 7): it is sufficient to click the button 🔴 and using the dialog window generated. When opting for this method, PaIRS first identifies the folder location of the picked image file and analyzes it to automatically detect the potential patterns of the image sets contained in it (exactly as when selecting the path with method 1)). Then, the "best" pattern matching the picked image file is set as the current one. The only criterion to choose the best candidate is the number of images matching the pattern.
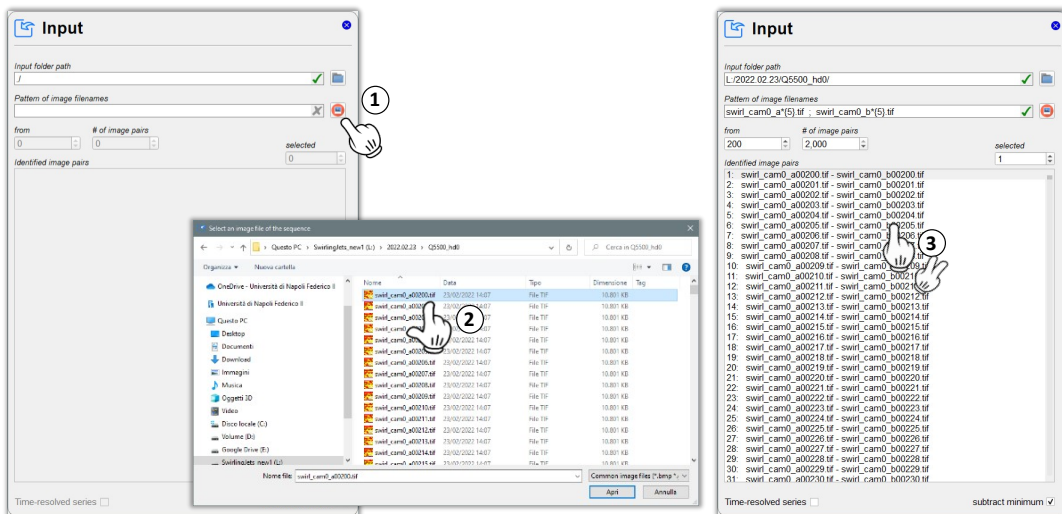
Figure 7 – Steps for selecting the image set by picking one image file.

Now, let us clarify how PaIRS automatically detects the potential patterns of the sets of images contained in a folder. Not all the potential patterns are indeed identified by PaIRS.

Suppose that the chosen folder (selected in method 1) or as the folder of the file picked in method 2)) contains the following files:

Example folder:

```
swirl_cam0_a00200.tif                    swirl_cam1_a00200.tif
swirl_cam0_a00201.tif                    swirl_cam1_a00201.tif
swirl_cam0_a00202.tif                    swirl_cam1_a00202.tif
swirl_cam0_a00203.tif                    swirl_cam1_a00203.tif
swirl_cam0_a00204.tif                    swirl_cam1_a00204.tif
swirl_cam0_b00200.tif                    swirl_cam1_b00200.tif
swirl_cam0_b00201.tif                    swirl_cam1_b00201.tif
swirl_cam0_b00202.tif                    swirl_cam1_b00202.tif
swirl_cam0_b00203.tif                    swirl_cam1_b00203.tif
swirl_cam0_b00204.tif                    swirl_cam1_b00204.tif
…                                        …
```

Then the identified patterns and the corresponding sets of matching files are:

Detected patterns:
```
swirl_cam*{1}_a00200.tif:
    1:   swirl_cam0_a00200.tif - swirl_cam1_a00200.tif

swirl_cam0_a*{5}.tif:
    1:   swirl_cam0_a00200.tif - swirl_cam0_a00201.tif
    2:   swirl_cam0_a00202.tif - swirl_cam0_a00203.tif
    3:   swirl_cam0_a00204.tif - swirl_cam0_a00205.tif
    4:...

swirl_cam0_a*{5}.tif ; swirl_cam0_b*{5}.tif:
    1:   swirl_cam0_a00200.tif - swirl_cam0_b00200.tif
    2:   swirl_cam0_a00201.tif - swirl_cam0_b00201.tif
    3:   swirl_cam0_a00202.tif - swirl_cam0_b00202.tif
    4:...

swirl_cam*{1}_a00201.tif:
    1:   swirl_cam0_a00201.tif - swirl_cam1_a00201.tif

swirl_cam1_a*{5}.tif:
    1:   swirl_cam1_a00200.tif - swirl_cam1_a00201.tif
    2:   swirl_cam1_a00202.tif - swirl_cam1_a00203.tif
    3:   swirl_cam1_a00204.tif - swirl_cam1_a00205.tif
    4:...

swirl_cam1_a*{5}.tif ; swirl_cam1_b*{5}.tif:
    1:   swirl_cam1_a00200.tif - swirl_cam1_b00200.tif
    2:   swirl_cam1_a00201.tif - swirl_cam1_b00201.tif
    3:   swirl_cam1_a00202.tif - swirl_cam1_b00202.tif
    4:...
```

In the above patterns, *$\{n\}$ denotes the number of digits in the sequential number identifying the image file. Indeed, the first and fourth patterns 'swirl_cam*{1}_a00200.tif' and 'swirl_cam*{1}_a00201.tif' are identified as two sets of single-frame images where the sequential number is a $n = 1$ digit number placed between the word "cam" and the "_" character. For the same folder there are at least three more patterns like these two, namely, 'swirl_cam*{1}_a00202.tif', 'swirl_cam*{1}_a00203.tif' and 'swirl_cam*{1}_a00204.tif' which are not identified by PaIRS for brevity (however, if needed the user can add them by using the line text box *Pattern of image filenames* following the rules reported below). Also the patterns 'swirl_cam*{1}_b00200.tif', 'swirl_cam*{1}_b00201.tif', ... 'swirl_cam*{1}_b00204.tif' are not identified by PaIRS.

The patterns 'swirl_cam0_a*{5}.tif' and 'swirl_cam1_a*{5}.tif' are identified as two sets of single-frame images where the sequential number is a $n = 5$ digit number placed between the letter "a" and the file extension. For the same folder there are two further similar patterns 'swirl_cam0_b*{5}.tif' and 'swirl_cam1_b*{5}.tif' which are not identified by PaIRS for brevity.

Finally, the patterns 'swirl_cam0_a*{5}.tif ; swirl_cam0_b*{5}.tif' and 'swirl_cam1_a*{5}.tif ; swirl_cam1_b*{5}.tif' correspond to two sets of double-frame images where the sequential number is a $n = 5$ digit number placed between the letter identifying the frame ("a" and "b") and the file extension. There are still two different possibilities for such sets of images, namely 'swirl_cam0_b*{5}.tif ; swirl_cam0_a*{5}.tif' and 'swirl_cam1_b*{5}.tif ; swirl_cam1_a*{5}.tif' obtained by switching the frames "a" and "b", which are not identified by PaIRS.

In summary, PaIRS does not identify the following patterns (for brevity and assuming that these are not desired by the user):

Omitted patterns:

```
swirl_cam*{1}_a00202.tif:
   1:   swirl_cam0_a00202.tif - swirl_cam1_a00202.tif

swirl_cam*{1}_a00203.tif:
   1:   swirl_cam0_a00203.tif - swirl_cam1_a00203.tif

swirl_cam*{1}_a00204.tif:
   1:   swirl_cam0_a00204.tif - swirl_cam1_a00204.tif

swirl_cam*{1}_b00200.tif:
   1:   swirl_cam0_b00200.tif - swirl_cam1_b00200.tif

swirl_cam*{1}_b00201.tif:
   1:   swirl_cam0_b00201.tif - swirl_cam1_b00201.tif

swirl_cam*{1}_b00202.tif:
   1:   swirl_cam0_b00202.tif - swirl_cam1_b00202.tif

swirl_cam*{1}_b00203.tif:
   1:   swirl_cam0_b00203.tif - swirl_cam1_b00203.tif

swirl_cam*{1}_a00204.tif:
   1:   swirl_cam0_b00204.tif - swirl_cam1_b00204.tif

...

swirl_cam0_b*{5}.tif:
   1:   swirl_cam0_b00200.tif - swirl_cam0_b00201.tif
   2:   swirl_cam0_b00202.tif - swirl_cam0_b00203.tif
   3:   swirl_cam0_b00204.tif - swirl_cam0_b00205.tif
   4:...

swirl_cam1_b*{5}.tif:
   1:   swirl_cam1_b00200.tif - swirl_cam1_b00201.tif
   2:   swirl_cam1_b00202.tif - swirl_cam1_b00203.tif
   3:   swirl_cam1_b00204.tif - swirl_cam1_b00205.tif
   4:...

swirl_cam0_b*{5}.tif ; swirl_cam0_a*{5}.tif:
   1:   swirl_cam0_b00200.tif - swirl_cam0_a00200.tif
   2:   swirl_cam0_b00201.tif - swirl_cam0_a00201.tif
   3:   swirl_cam0_b00202.tif - swirl_cam0_a00202.tif
   4:...

swirl_cam1_b*{5}.tif ; swirl_cam1_a*{5}.tif:
   1:   swirl_cam1_b00200.tif - swirl_cam1_a00200.tif
   2:   swirl_cam1_b00201.tif - swirl_cam1_a00201.tif
   3:   swirl_cam1_b00202.tif - swirl_cam1_a00202.tif
   4:...
```

   If one of the above patterns is indeed the desired one, the user should insert it manually in the line text box *Pattern of image filenames* and press the Return key to set it as current. It is noticed that after setting a new pattern as current, it is also added to the list of suggestions appearing when focusing on the line text box *Pattern of image filenames*.

   The right syntax to specify the pattern is that reported as bold string in the above box displaying the omitted patterns. Indeed, the user should remind only two rules for formatting correctly the pattern:

1. the patterns for the frame "a" and the frame "b" are separated by a semicolon ";". Notice that it is not required that the image files for the different frames are denoted by the letters "a" and "b". The only requirement is that the image files corresponding to the same pair have the same sequential number. For instance, also the pattern string: 'test_1_*{4}.png ;

test_2_*{4}.png' is valid. Notice that the pattern string 'test_1_*{4}.png ; test_1_*{4}.png', or better, more in general, the same pattern repeated twice with a comma between the repetitions allows the user to build a sequence for autocorrelation analysis;

2. the sequential number is identified by the syntax *{n} with n being the number of digits. Alternatively, the user can simply put * in place of the sequential number and PaIRS automatically detects the number of digits.

Additionally:

3. if the image files corresponding to different frames are denoted by a letter, the pattern can be also indicated with a @. For instance, inserting the pattern 'swirl_cam0_@*.tif' when working with the above example folder produces as result 'swirl_cam0_a*{5}.tif ; swirl_cam0_b*{5}.tif'.

Once selected the desired set of images, the user can choose the range of the images to process using the spin boxes *from* and *# of image pairs*, as illustrated in Figure 8. Notice that in the *from* spin box the images are enumerated according to the sequential number identified in the corresponding filenames.
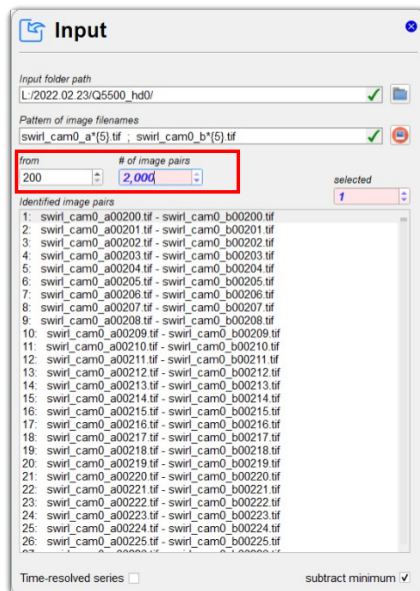
Figure 8 – Spin boxes for selecting the range of the images to process.

Moreover, the user can select one image and visualize it in the Vis tab by clicking on the name of the image file in the list *identified image files* or, equivalently, using the dedicated spin boxes (*selected* and *frame*).

Finally, the Input Tab includes two check boxes:

1. *Time-resolved series*: if checked, the frame 1 of the k-th pair and the frame 0 of the $(k + 1)$-th pair are added to list of the images to process. The effect of activating the *Time-resolved series* check box is shown in Figure 9;

2. *subtract minimum*: if checked, during the PIV analysis the historical-minimum background image is subtracted to each image of the pair before applying the image deformation method.
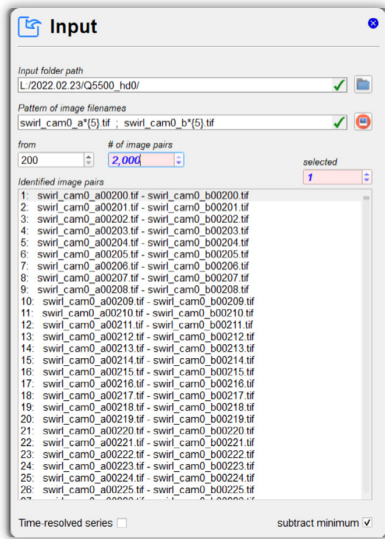
Time-resolved series unchecked   *Time-resolved series* checked
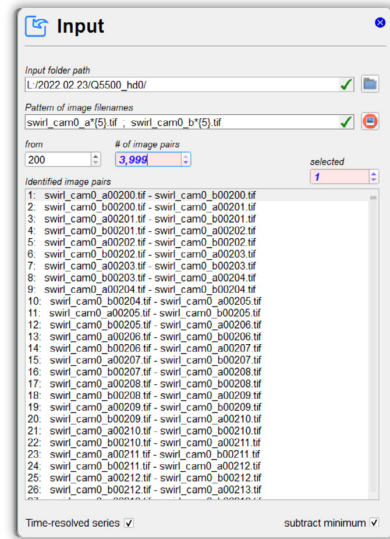
Figure 9 – Effect of activating the *Time-resolved series* check box.

10. Output

The Output tab (Figure 10) allows the user to:

1. specify desired transformations of the images to be performed before the PIV analysis and transformations of the velocity vectors to be performed after the PIV analysis (*Resize/reshape* collapsible box);

2. set the image resolution and the time delay to convert output from "pixel per frame" to "meter per second" units;

3. select the location, the name and the type of the output files.
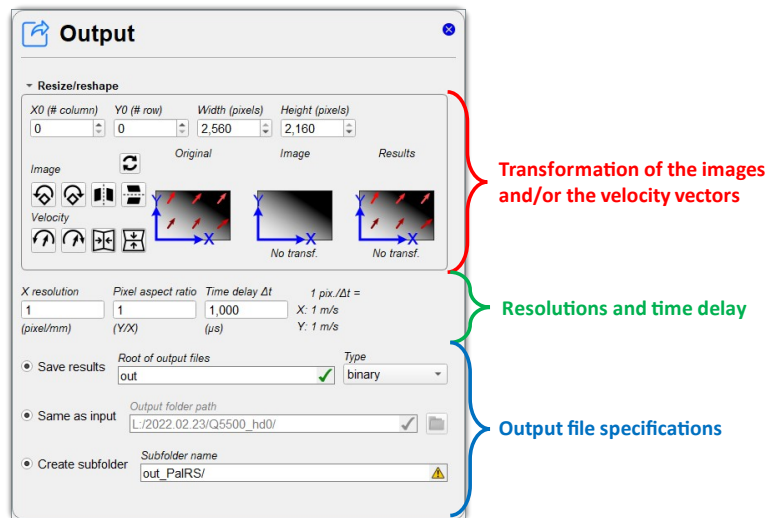
Figure 10 – The Output tab and its sections.

Resize/reshape

The four spin boxes *X0 (# column)*, *Y0 (# row), Width (pixels)* and *Height (pixels)* allow to crop the images. *X0* and *Y0* are the first column and the first row of the region of interest, while *Width* and *Height* are the number of the columns and rows of this region, respectively. The result of the image cropping can be easily visualized in the Vis tab.

The image and velocity transformations are performed via the following buttons:

Rotate the image 90° counter-clockwise or clockwise (before PIV process).

Flip the image horizontally or vertically (before PIV process).

Rotate the image and velocity vectors 90° counter-clockwise or clockwise (after PIV process).

Flip the image and velocity vectors frame horizontally or vertically (after PIV process).

Cancel all previously assigned transformations and restore the original images and reference frame.

The result of the image and velocity vectors' transformations can be easily visualized by comparing the example pictures *Original, Image* and *Transformed* reported in the *Resize/reshape* collapsible box. The *Image* picture shows the result of the image transformations, whereas the *Results* picture shows the result of both the image and velocity vectors' transformations. The latter can be also seen in the Vis tab, where the images are always visualized after all the image and velocity transformations assigned by the user.

Please note: The cropping operations are always performed before the remaining image operations.

Resolutions and time delay

The user is asked to specify the resolution along the *X* direction in pixel/mm units (R_X), the pixel aspect ratio (λ) and the time delay in µs (Δt).

Please note: For resolutions the *X* and *Y*-directions always refer to the rotated image (between "original" and "results")!

If we denote with U and V the velocity *X*- and *Y*- components in m/s units and with $\widetilde{U}$ and $\widetilde{V}$ the corresponding velocity components in pixel/frame units, then the conversion from $\widetilde{U}, \widetilde{V}$ to U, V is given by:

$$U = \frac{1000}{R_X \Delta t} \cdot \widetilde{U} = f_X \cdot \widetilde{U}$$

$$V = \frac{1000}{\lambda R_X \Delta t} \, \widetilde{V} = f_Y \cdot \widetilde{V}$$

The output of PaIRS is always given by U, V. In order to have them expressed in pixel/frame units, the user should set R_X, λ and Δt in such a way that the conversion factors f_X and f_Y are equal to 1. For instance:

$$R_X = \lambda = 1 \quad \text{and} \quad \Delta t = 1000 \qquad \text{(for output in pixel/frame units)}$$

Output file specifications

In this section (see Figure 10, blue block), the user can:

1. specify the root of the output file and the type. Output can be saved as a binary file, namely .mat format, compatible with Matlab or as a Tecplot file, namely .plt format;

2. specify the location of the output folder. This can be chosen to be the same as the input folder or different;

3. specify a subfolder to be created in the output folder where to save the files.

11. Process

In the Process tab the user can setup the settings for the image deformation method on which the PIV analysis relies. A schematic flowchart of the image deformation method used in the PaIRS-PIV library is reported in Figure 11. Further details can be found in the works from the PaIRS authors. In particular, for the process of 2D planar PIV images the reader is referred to the references [1]-[5].
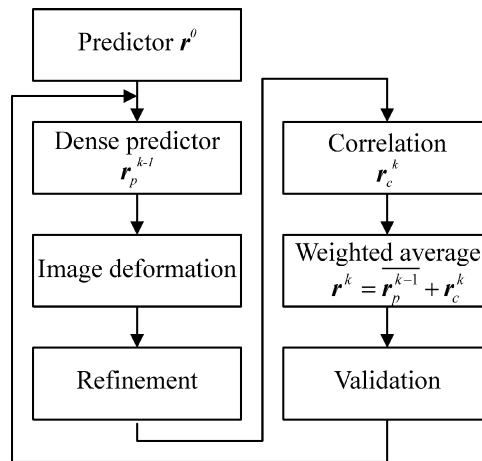
Figure 11 – Schematic flowchart of a possible image deformation method for PIV.

The Process tabs consists of several collapsible boxes in which different parameters of the method can be specified. Some of these boxes are accessible only switching from the *simple* mode to the *advanced* and *expert* modes, using the combo placed close to the tab title.

Simple mode

The appearance of the Process tab in the simple mode is shown in Figure 12.

For most purposes, the user must only:

1. indicate the size and number of the interrogation windows, using the *Interrogation Windows* box;

2. choose the final number of iterations and if performing direct correlations in the final steps of the method, using the *Final iterations* box;

3. select the desired type of process from the *Type of process* box.
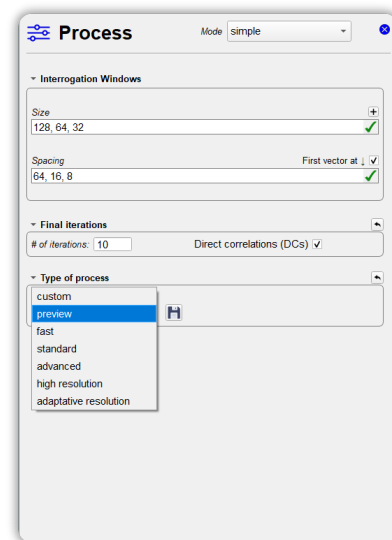
Figure 12 - Process tab in the simple mode.

Interrogation Windows

In this box, the user must choose (refer to Figure 13):

1. the *height* and *width* of the interrogation windows: sequences of integer numbers indicating the height and width of the interrogation spots in the multi-grid interrogation. In the final iterations (see the next subsection), the width and the height of the interrogation windows are equal to the last number of the corresponding field. Each element of the *height* and *width* sequences has to be smaller

than respectively the total height and width of the image to be processed and than the previous element of the sequence;

2. the *vertical* and *horizontal* spacings: sequences of numbers indicating the spacing between the vectors along the vertical and horizontal directions in the image reference frame. Each value should be less than or equal to the final size of the windows; otherwise some parts of the images could be skipped in the analysis. Values smaller than the corresponding sizes of the interrogation windows determine overlapped windows. E.g., a 50% overlap could be obtained setting a grid distance of 32 when the size of the interrogation window is 64. The final element has to be the smallest one of the sequence and is used in the final iterations;

Figure 13 – Interrogation Windows box.

3. the *position* of the first interrogation window: if this checkbox is activated interrogation windows are placed close to the boundary. In particular, the first and the last vectors in each direction are placed at a distance from the boundary equal to the grid distance.

Final iterations

In this box, the user must choose (refer to Figure 14):

1. the number of final iterations: the number of iterations to be added to the refinement process (specified in the *Interrogation Windows box*). The advised values are 1 or 2, the value 3 is enough if the final size of the interrogation windows is equal to the initial size. Higher values can be used if employing both windowing and weighting windows for the correlation map to increase the spatial resolution;

2. if using direct correlations: if the corresponding checkbox is activated, the correlation map is computed via direct correlations only on the 5 points close to the predictor displacement; otherwise, the entire correlation map is calculated via fast Fourier transform. In the former case, it is not possible to use the validation criteria of the Signal to Noise (S/N) ratio, and the classic Gaussian method is used to interpolate the correlation peak.
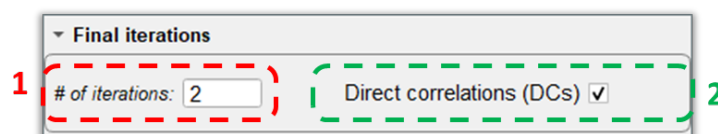
Figure 14 – Final iterations box.

Type of process

In this box, the user can (refer to Figure 15):

1. select one among the preset types of process: the description of these type is provided in Table 1. The user can visualize the values of the process parameters corresponding to such types by switching to the *advanced* or *expert* modes and checking across the boxes of the Process tab;

2. select one of the previously saved or imported custom processes (by using the combo box *Custom types*), edit the list of custom processes (by clicking on the button ![pencil]) or save the current settings as a new custom type (by clicking on the button ![H]) in such a way that they can readily recovered at a later time.
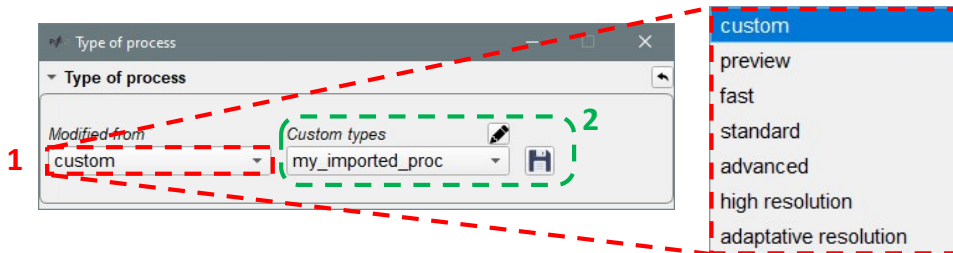
Figure 15 – Type of process box.

Table 1 – Types of process.

Type	Description
custom	User defined process.
preview	Fast analysis, but not really reliable.
fast	A little slower of the preview type, but more reliable.
standard	Standard analysis method based on image deformation with accurate interpolation schemes.
advanced	Standard analysis method based on image deformation with accurate interpolation schemes for both interpolation of the velocity field and of the value of intensity of the images.
high resolution	It employs windowing to enhance the frequency response for high frequencies. It is less reliable at the low frequencies.
adaptive	Adaptive resolution process [5]. It applies an adaptive windowing based on the value of the correlation peak and on the standard deviation of the dense predictor.

By clicking on the button ![pencil] the user can access *Custom types of process* dialog box, illustrated in Figure 16.
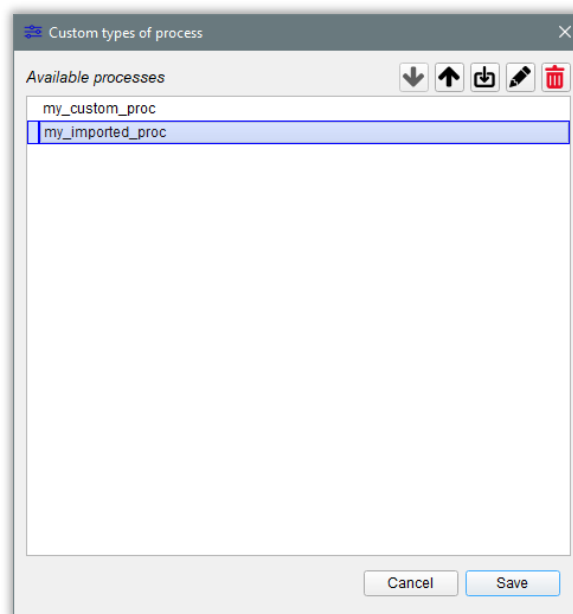
Figure 16 – Custom types of process dialog box.

In this dialog box, the user can easily edit the list of custom types by using the inherent tool buttons:

Move items in the list down/up thus changing the list sorting (`Ctrl+Down`, `Ctrl+Up`).

Import a process saved in the disk (`Ctrl+D`). The user can import the process settings from both output files generated by PaIRS and ".cfg" input configuration files compatible with the PaIRS-PIV library by browsing and selecting them.

Edit the item name (`F2`)/Undo name changes (`Ctrl+Z`).

Delete item (`Backspace`)/Restore item (`Ctrl+R`).

All the changes made by the user to the list of custom types of process will be effective only after clicking the button Save (`Ctrl+S`); the button Cancel (`Esc`) allows to exit this dialog box without any change. Please, notice that removal of items from the list is irreversible. For this reason, the user is asked to confirm the saving of changes when items have been removed.

Advanced mode

This mode gives the user access to the *Interpolation* box.

Interpolation

In this box, the user can choose (refer to Figure 17):

1. the type of interpolation to be used in the deformation of the images after the evaluation of a dense predictor displacement field. The possible types are reported in Table 2, where their variants or the corresponding parameters that must specified are also indicated. By clicking on the button ⊞ it is also possible to open a section where the user can select a different type of image interpolation for a specified number m of final iterations, as shown in Figure 18.

2. the type of interpolation of the correlation map, namely the type of interpolation to evaluate the displacement with sub-pixel accuracy on the correlation map. Three different types are provided: classic Gaussian, reviewed Gaussian and simplex method;

3. the type of interpolation for the velocity map at each iteration. The possible types are reported in Table 3, where their variants or the corresponding parameters that must specified are also indicated.

Table 2 – Types of image interpolation.

Type	Description	Variants/Parameters	# points
none	No interpolation (advised for rapid analysis).	-	-
moving window	No interpolation, integer shift of the interrogation windows.	symmetric, asymmetric	-
linear revitalized	Linear Interpolation Revitalized.	-	4
bilinear/ biquadratic/ bicubic	Based on repeated polynomial interpolation along the two directions in the image plane.	Bilinear, biquadratic, bicubic, bicubic (spline)	4, 9, 16, 16
simplex	Simplex method (advised for a quick and pretty accurate process, setting the interpolation of the velocity field to bilinear)	-	12
shift theorem	Interpolation by using the shift theorem. This type of process is quite slow.	Width in pixel of the window used for the Fourier transform (range: 3-20; advised values: 8-10)	9-400
sinc (Whittaker-Shannon)	Interpolation via the cardinal interpolation formula. This type of process is quite slow, but faster than the shift interpolation.	Half-width in pixel of the interpolation kernel (range: 3-10; advised values: 4-5)	9-400
B-spline	Interpolation by using B-splines. The memory storage is more intensive.	Order (=width in pixel of the interpolation kernel -1) (range: 3-20; advised values: 7-10; fast implementations for 3 and 7)	9-441

Figure 17 – Interpolation box.
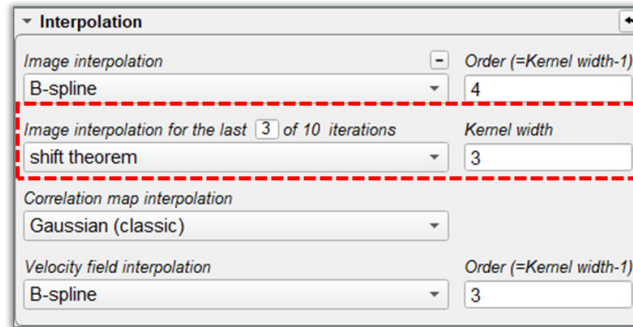
Figure 18 – Section for selecting a different type of image interpolation for a specified number of final iterations.

Table 3– Types of velocity field interpolation.

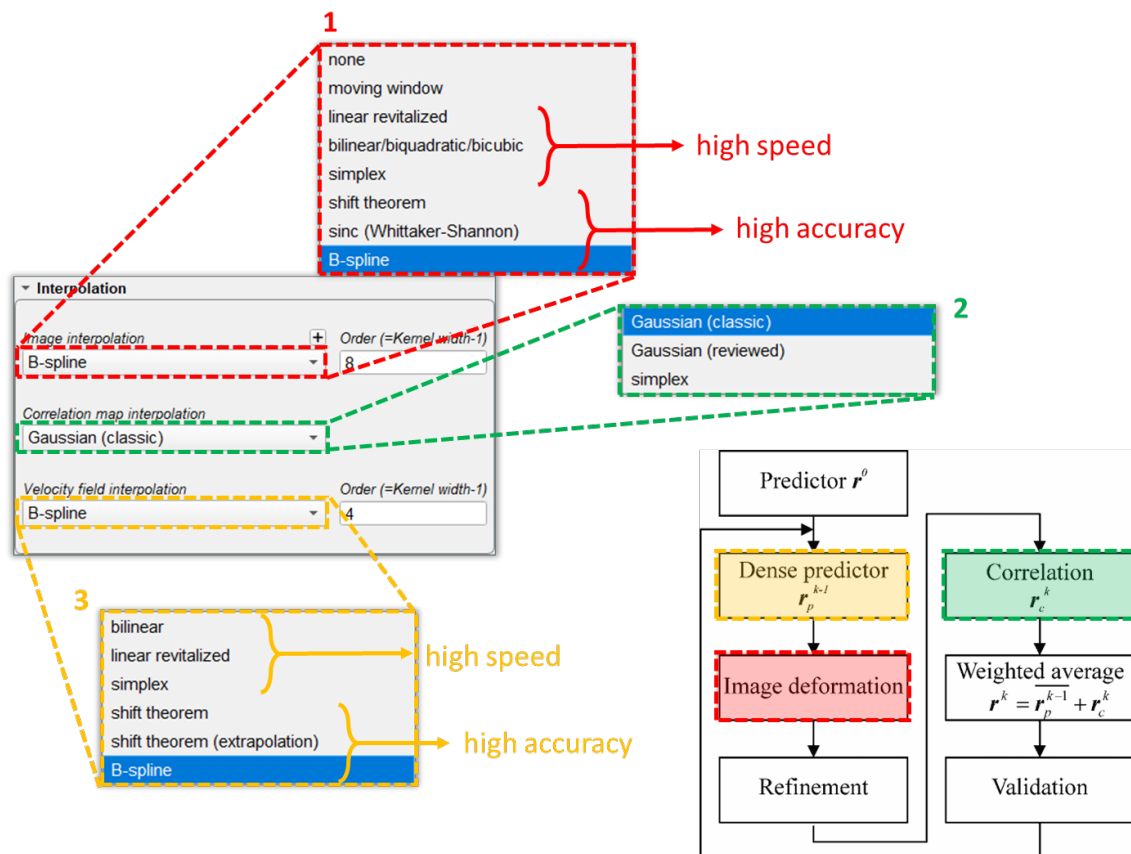| Type	Description	Variants/Parameters	# points
bilinear	Based on repeated linear interpolation along the two directions in the image plane.	-	4
linear revitalized	Linear Interpolation Revitalized.	-	4
simplex	Simplex method (reliable in case of extrapolation; the checkbox in the interrogation Windows box for placing vectors close to the boundary should be deactivated)	-	9
shift theorem	It works using the shift theorem on all the values of the displacement (the rows are processed before the columns). A line is subtracted to the initial data in order to avoid boundary effects. Generally not efficient since the FFT has to work with big prime numbers.	-	-
shift theorem (extrapolation)	It works using the shift theorem on all the values of the displacement (the rows are processed before the columns). In order to avoid boundary effects, the signal is extended with a cubic, increasing the number of points to reach a power of 2.	-	-
B-spline	Interpolation by using B-splines. The memory storage is more intensive.	Order (= width in pixel of the interpolation kernel -1) (range: 3-20; advised values: 7-10; fast implementations for 3 and 7)	9-441

Expert mode

This mode gives the user access to the Validation and the Windowing boxes.

Validation

In this box, the user can choose (refer to Figure 19):

1. **different outlier detection strategies**:

- Median test: the classic and the universal variant [6] are available. For both methods, the user has to specify the half-size of the validation kernel and the α threshold for the median test (recommended values between 1.0 and 3.0) and the spacing of velocity vectors. For the universal median test, in addition, the user has to specify the allowed error in pixel ε. A high value increases the occurrence of outliers detected as correct vectors. The value 0.1 is recommended; as a general rule, values of ε between 0.0 and 0.4 should be selected;

- S/N (Signal-to-Noise) test: signal-to-noise is defined as the ratio of the primary to the secondary peak value in the correlation map. In this test, outliers are identified as displacement values for which S/N is lower than the prescribed threshold. Threshold values must be greater than 1 (recommended values between 1.3 and 1.7). The S/N validation is active only in the predictor estimation in case of process with direct correlations;

- Correlation peak test: in this test, outliers are identified as displacement values for which the correlation peak value is lower than the prescribed threshold. Threshold values must be lower than 1 (the recommended value is about 0.25);

- Test based on the image intensity distribution: two thresholds has to be specified:

  o minimum allowed average value in the interrogation window: if the average intensity level is below this threshold, the vector is identified as an outlier since the signal is too low. The recommended value depends on the image. The value 0 disables the control;

  o minimum allowed standard deviation value in the interrogation window: if the standard deviation of the intensity levels is below this threshold, the vector is identified as an outlier since the signal is too low. The recommended value depends on the image. The value 0 disables the control;

Please note: the above threshold are also used in the minimum background image computation to check if the average and standard deviation of the intensity level in the whole image is critically low.
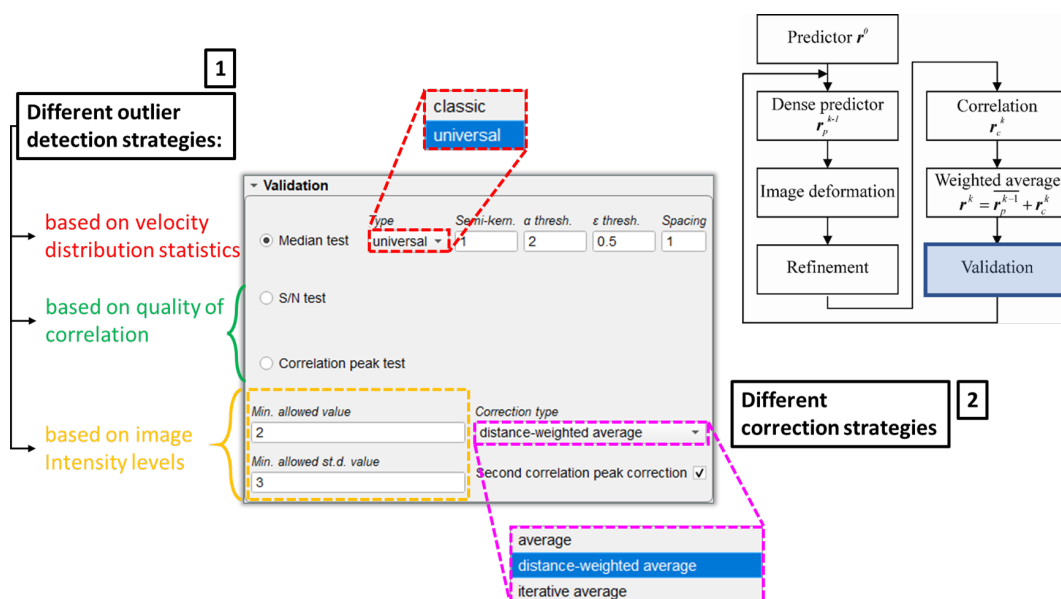

Figure 19 – Validation box.

2. **different correction strategies:**

   a. replacement methods based on averaging neighboring correct velocity vectors (*Correction type* combo box). Three different options are available: the outliers are corrected by average on the correct vectors in the neighborhood (average); the average is weighted with the distance of the correct vectors in the neighborhood (distance-weighted average); the average is iterative on the kernel of eight vectors close to the outlier (iterative). The first method is the more robust, while the last one should be more suited when higher spatial resolution is required;

   b. correction with the secondary peak in the correlation map: if activated, the second maximum (if available) is checked in case of detection of an outlier. It is automatically inactive in case of direct correlations.

Windowing

   In this box, the user can choose the windowing parameters for each iteration in the initial phase of the PIV process (Figure 20). In order to modify the parameters related to a specific iteration, the user must select the corresponding row in the table widget 1 and then use the widgets 2-5 to choose:

2. the type of weighting window to be used in the calculation of the absolute velocity. The possible types are reported in Table 3. In the *Half-width* spin box the user must specify the semi-width w of the weighting window. If this value is equal to 0, then the weighting window has the same sizes as the interrogation window. If the size of the interrogation window is even, $2w$ points are used; if the size of the interrogation window is odd, $2w - 1$ points are used;

3. the type of weighting window to be used in the calculation of the correlation map. The possible types are reported in Table 3;

4. the maximum allowed displacement (MAD) in the interrogation window: this parameter limits the search of the peak in the correlation map. Two kind of limitations are possible: relative, in which the MAD is assigned as a fraction of the interrogation window size (the only possible options in such a case are the values 1/2, 1/3, 1/4 and 1/5; 1/4 corresponds to the widely used Adrian's rule); absolute, in which the MAD is assigned in pixel units (an integer value ranging from 1 to one half of the minimum size of the interrogation window can be specified in such a case). Using a large interrogation window with a large MAD at the first iteration enables to have a quite robust estimate of the predictor displacement field; in this way, using smaller values of the MAD in the successive iterations it is possible to avoid a drift from the predictor displacement.
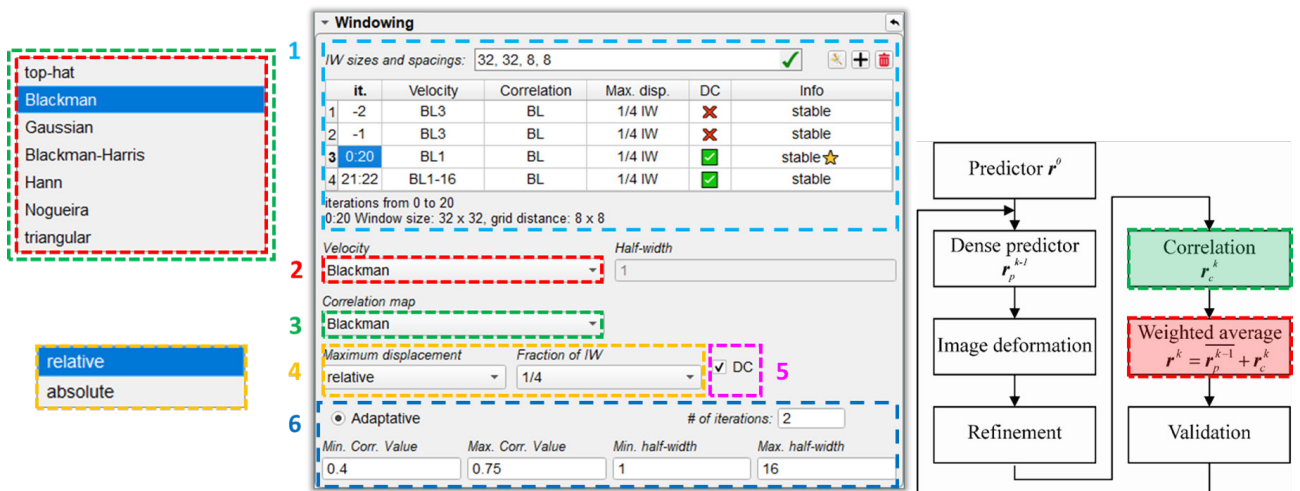

Figure 20 – Windowing box.

Table 4 – Type of weighting windows for computation of the absolute velocity and the correlation map.

Type	Variants/Parameters	Description
top-hat	full (no windowing)	No windowing Most stable method both for the velocity and for the correlation map: in this case the frequency response is a sinc.
	50%	Equivalent to a window with zero padding.
	contoured	A frame of zeros is added to the uniform weighting window.
Blackmann	-	Blackmann window (without zeros at the borders).
Gaussian	Standard deviation of the Gaussian shape α.	
Blackmann-Harris	3 term -67	
	3 term -61	
	4 term -92	
	4 term -74	
Hann	-	Hann reduced.
Nogueira	et al. 1999	Nogueira et al. (1999).
	et al. 2005	Nogueira et al. (2005).
triangular	-	Bartlett reduced.

5. if enabling the use of direct correlations: if the corresponding checkbox is activated, the correlation map is computed via direct correlations only on the 5 points close to the predictor displacement; otherwise, the entire correlation map is calculated via fast Fourier transform. In the former case, it is not possible to use the validation criteria of the Signal to Noise (S/N) ratio, and the classic Gaussian method is used to interpolate the correlation peak.

6. if using the adaptative method: in this case the user has to specify four different parameters, namely the minimum and maximum values of the correlation peak allowed and the minimum and maximum value of semi-size of weighting window for the displacement field in the adaptative process (see [5] for more details). In addition, the number of extra iterations to be performed in the adaptative process must be specified.

It is also noticed that the table widget 1 not only reports a schematic of the crafted windowing scheme, but also allows the user to modify the sizes and the spacings of the interrogation windows (in alternative to the widgets in the *Interrogation windows box*) by using the *IW sizes and spacing* edit box and to add or remove initial iterations in the process via the buttons ➕ and 🗑. Among the useful details reported in the table, even an indication about the stability of the windowing scheme for each iteration is reported (last column). Moreover, for the final iterations of the PIV process it is possible also to display the Modulation Transfer Function (MTF) corresponding to both the chosen number of iterations and an infinite number of iterations (i.e., at convergence) by clicking on the button 🎇 in the upper part of the table.

12. Log

The Log tab is used to display useful information about the processes defined by the user, as show in Figure 21. Such information is organized in several sections:

- **Header:** this section reports information about the version of PaIRS used for the currently selected process, the identifying name of the process and the day and time of its last modification;

- **Output**: for a historical minimum background image computation, this section reports a list of the names of the correctly opened images; for a PIV analysis the following pieces of information are reported:

  - interrogation window sizes (IW);
  - grid size (number of vectors along X and Y) (#IW);
  - number of correct and total vectors (#Vect/#Tot);
  - percentage of correct vectors (%);
  - average coefficient of correlation between the total images (CC);
  - average correlation coefficient relative to the peak of correlation in each IW (CC(avg));
  - percentage of failed peak detection with fast direct correlations on the 5 points-stencil (DC%).

- **Progress status**: this section reports information about the number of images processed (with and without errors) and not processed yet (in case the process has been paused by the user) and about the process time (total time of process, time per image pair and estimated time to the end). The percentage of image pairs processed without errors is also displayed in the progress bar placed in the Log tab header;

- **Error report**: this section reports (if any) errors occurred during the process. Critical errors should be taken into consideration and kindly communicated to the authors;

- **Further information**: it regards potential issues that were present when the process was launched;

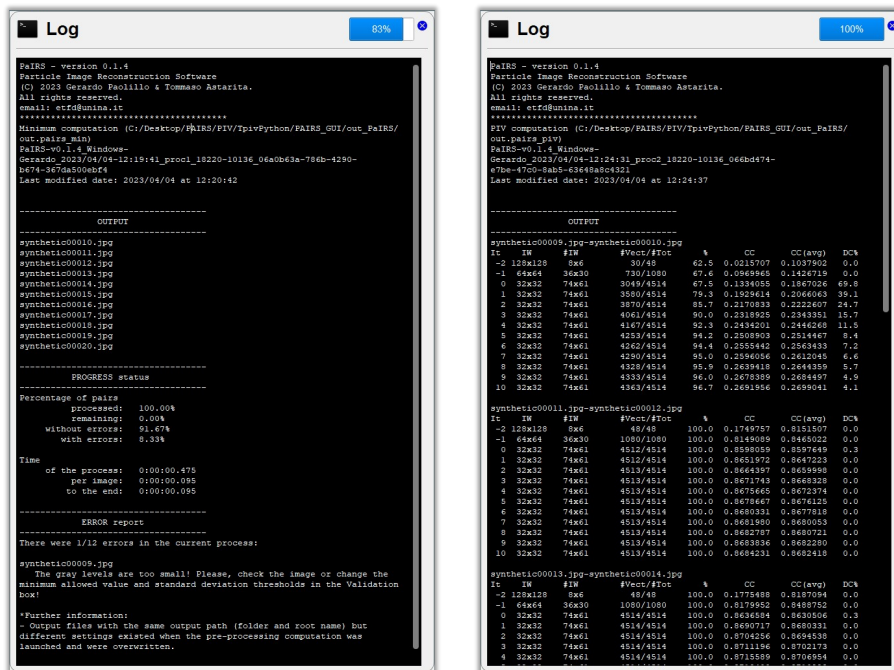- **Warnings**: this section reports potential issues related to the process in its current state.

Figure 21 – Information displayed in the Log tab during a historical minimum background image computation (left) and a PIV analysis (right).

13. Vis

The Vis tab (Figure 22) allows to visualize the input images and available results (if any) of the current set
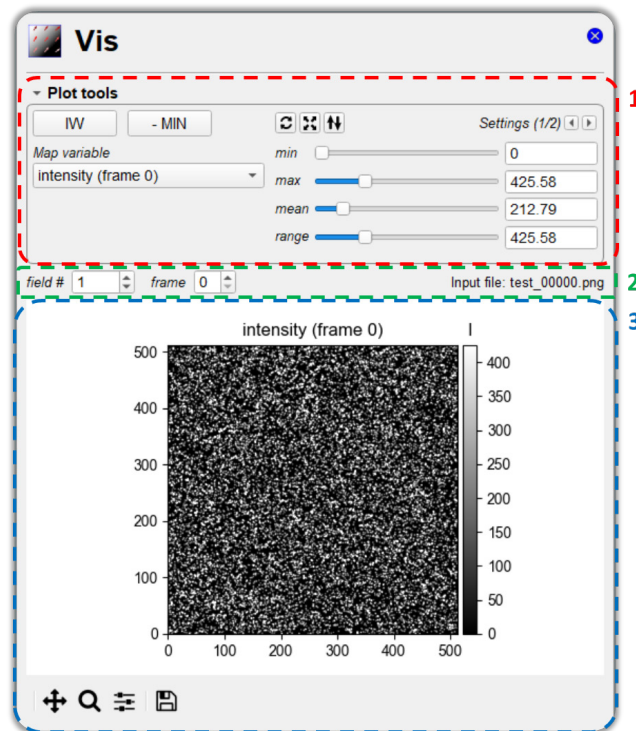

Figure 22 – The Vis tab.

It consists of three blocks:
1. the Plot tools collapsible box, where the user can select the element (input image or output variable of the process) to be visualized and change the visualization settings;

2. the field box, where the user can rapidly change the number of field (i.e., input image pair plus the corresponding output velocity field if any) and the image frame to be visualized without using the Input tab;

3. the plot area, where the selected element is displayed and the user can interact with it via the navigation toolbar provided by the matplotlib package.
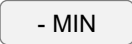
Plot tools

The tools available in the present box depend on the type of process selected in one of the trees of the Queue tab.

Plot tools for the input images and the output from minimum computation

Figure 23 shows the layout of the Plot tools box for the input images and the output from a minimum computation process. In such cases, the user can:
1. choose if visualizing frame 0 (a) or 1 (b) of the selected image pairs. This choice can be also made by selecting the corresponding frame in the Input tab (see section 9);

2. change the type of visualization by clicking one of the two buttons:

IW — Visualize the current grid scheme as shown in Figure 24 (`Ctlr+Alt+W`). The numbers refer to the elements of the sequences in the line text boxes of the *Interrogation Windows box* with 0 corresponding to first element;

- MIN — Visualize the pre-processed images, i.e., the intensity distribution obtained by subtracting the minimum historical background from the images (`Ctlr+Alt+M`). It is visible only for processes in the *Past* queue for which a minimum is available;
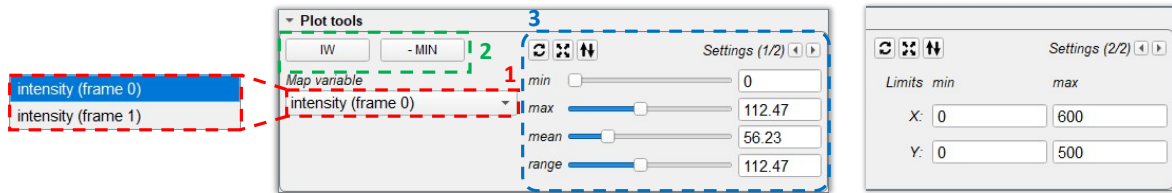
29

Figure 23 – Plot tools box layout when a minimum computation process is selected.

3. set the levels and the domain limits for the image visualization. The range of levels can be specified via four sliders and/or related spin boxes: *min* (to set the minimum level), *max* (maximum), *mean* (average value of the range) and *range* (to set the span of the range). As concerns the domain to be visualized, both *X* and Y minimum and maximum limits can be set in the second page of the *Settings* panel. Three further buttons are available in this block:

Reset the range of levels to the default values.

Restore the axis limits in such a way to visualize the whole field.
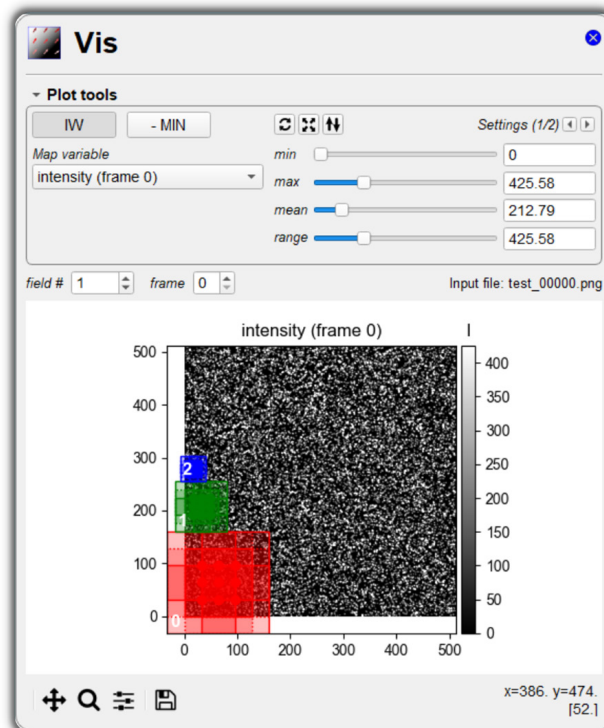
Invert the direction of the vertical axis.

Figure 24 – Visualization of the grid scheme in the Vis tab.

Plot tools for the output from PIV computation

Figure 25 shows the layout of the Plot tools box for the output from a PIV computation process. In such cases, the user can:

1. choose the distribution of the variable to be visualized among:

   o the intensity levels of frame 0 (a) or 1 (b) of the selected image pair;

   o the velocity magnitude;

   o the X- or Y-components of the velocity vector (U and V, respectively);

   o the X-normal, Y-normal and XY-tangential Reynolds stresses (‹$u'u'$›, ‹$v'v'$› and ‹$u'v'$›, respectively). The latter turbulent statistics are available only for the average velocity field;

30

- o   the correlation coefficient (*CC*);

- o   the *Info* variable. The *Info* variable provides information about the type of outlier correction performed on the considered velocity vector. In particular, by referring to list 2 of the paragraph Validation at page 25, *Info* is equal to 0 in the case a(replacement methods based on averaging), 1 if the vector was not an outlier and 3/2 in the case b (correction with secondary peak in the correlation map);

- o   the signal-to-noise (S/N) ratio.

2. change the type of field representation by choosing among superimposition of vectors or streamlines (see Figure 26);

3. set the levels for the image visualization, the domain limits and the number of contour levels and the vector size and spacing or the streamline density.
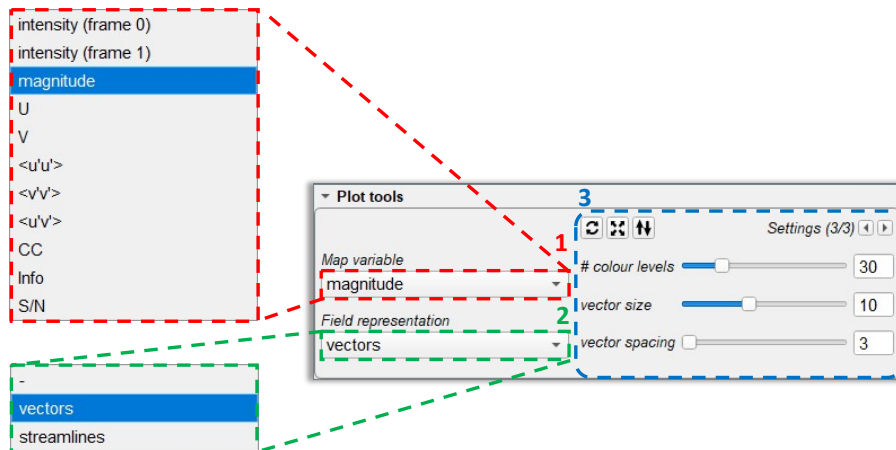

Figure 25 – Plot tools box layout when a PIV computation process is selected
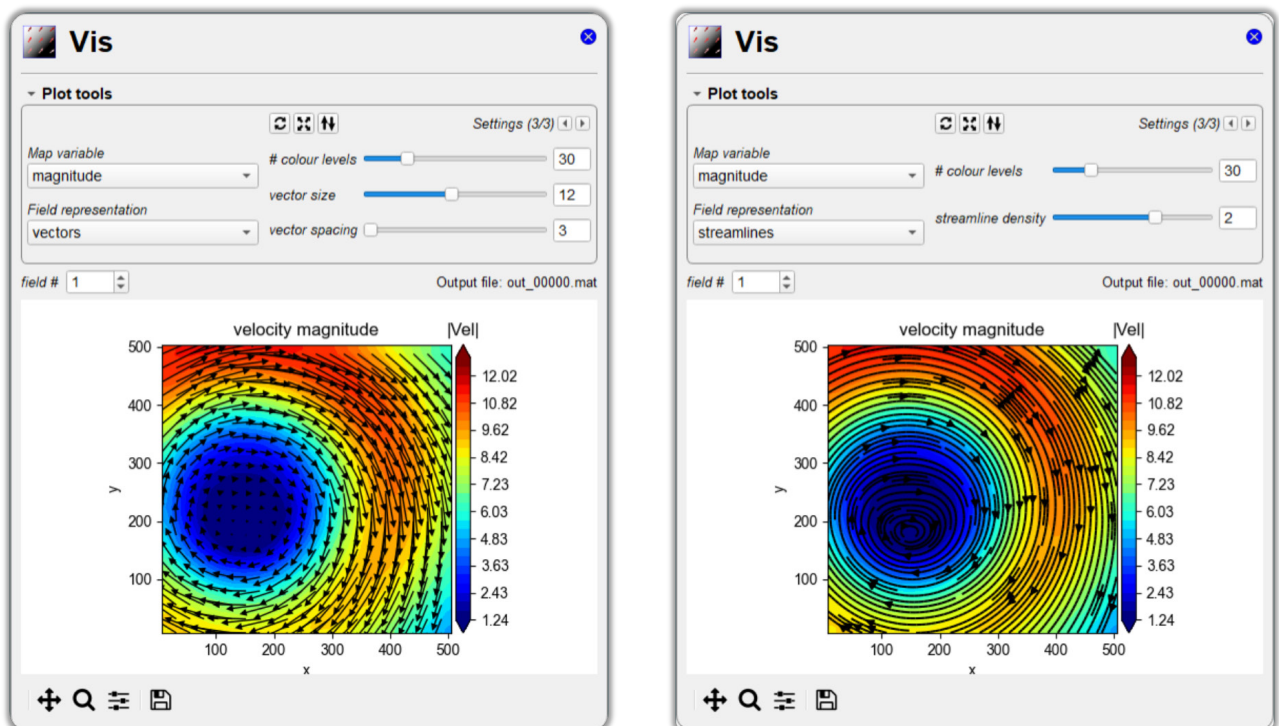

Figure 26 – Visualization of the velocity magnitude distribution with vectors (left) or streamlines (right) superimposed.

Field box

In this box, the user can rapidly change the number of field (i.e., the input image pair and the corresponding output velocity field if any) to be visualized without using the Input tab. In case the distribution of the intensity levels of the images is visualized, it is also possible to switch between the two frames of the pair via the corresponding spin box. The name of the file currently visualized is reported in the text label on the right of the spin box; when hovering the cursor over the text label, the tooltip shows the full path of the file currently visualized.

Setting the value 0 in the field box the user can visualize (if existing) the historical minimum background images and/or the average velocity field generated from a PIV analysis. While a PIV process is running, the value 0 results in the visualization of the last computed velocity field. In parallel processing for speed only the velocity fields related to one of the workers of the parallel pool is visualized.

Plot area

For further information on how interacting with the plot via the navigation toolbar of matplotlib, the user is referred to the documentation of this package (https://matplotlib.org/stable/index.html).

Important remark

The plots in the Vis tab are generated with the aid of matplotlib package. This can be slow depending on the machine. For fast use of PaIRS, try to close the Vis tab. In such a way, plotting is avoided and the use of the interface could be way more fluent.

14. Launch processing

Once the *Future* queue is ready, the user can click the *Run* button (`Ctrl+Return`) to start up processing (Figure 27, left). The processes indicated in the *Future* are executed sequentially and the processing can be stopped in any moment by clicking the pause button (`Ctrl+Return`), as shown in Figure 27, center. When processing is paused, it is possible to add new processes to the *Future* queue or change the order of the processes in the queue. After modifying the *Future* queue, it is possible to resume the processing by clicking on the play button, as shown in Figure 27, right.

During processing, the plot in the Vis tab will be automatically updated if the running process (first in the future queue and identified by the icon ⚙) is selected in the Queue tab and the field number in the Vis tab is set to the value 0. The user can however visualize the remaining processes, thus inspecting the results from processes already completed and moved to the *Past* queue. The progress state of the running process is shown by the progress bar in the header of the main window. Below this bar, an estimate of the time to the end of the present running process is also reported.

By default, PaIRS performs a parallel computation using the maximum number of logical cores (workers) available on the local machine. The user can set a lower number of workers using the spin box placed below the Queue tab.
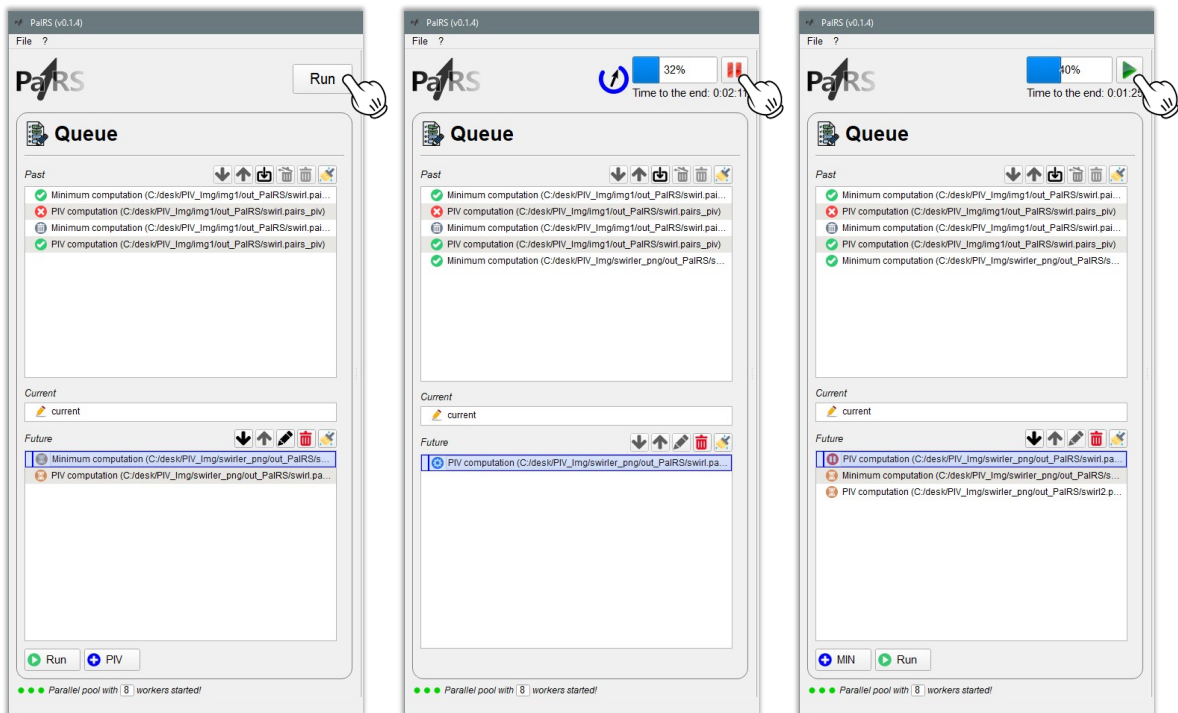

Figure 27 – Running (left) and stopping (right) processing in PaIRS.

33

15. Appearance

It is possible to set the light or dark mode of the PaIRS interface by clicking on the logo PaIRS and selecting the favorite option, as shown in Figure 28.
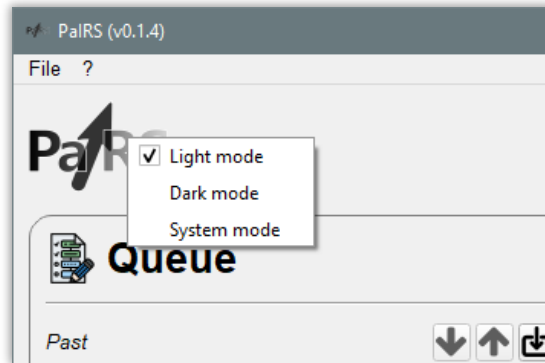

Figure 28 – Color mode context menu.

If you cannot wait for seeing the dark skin of PaIRS, give a look at Figure 29.
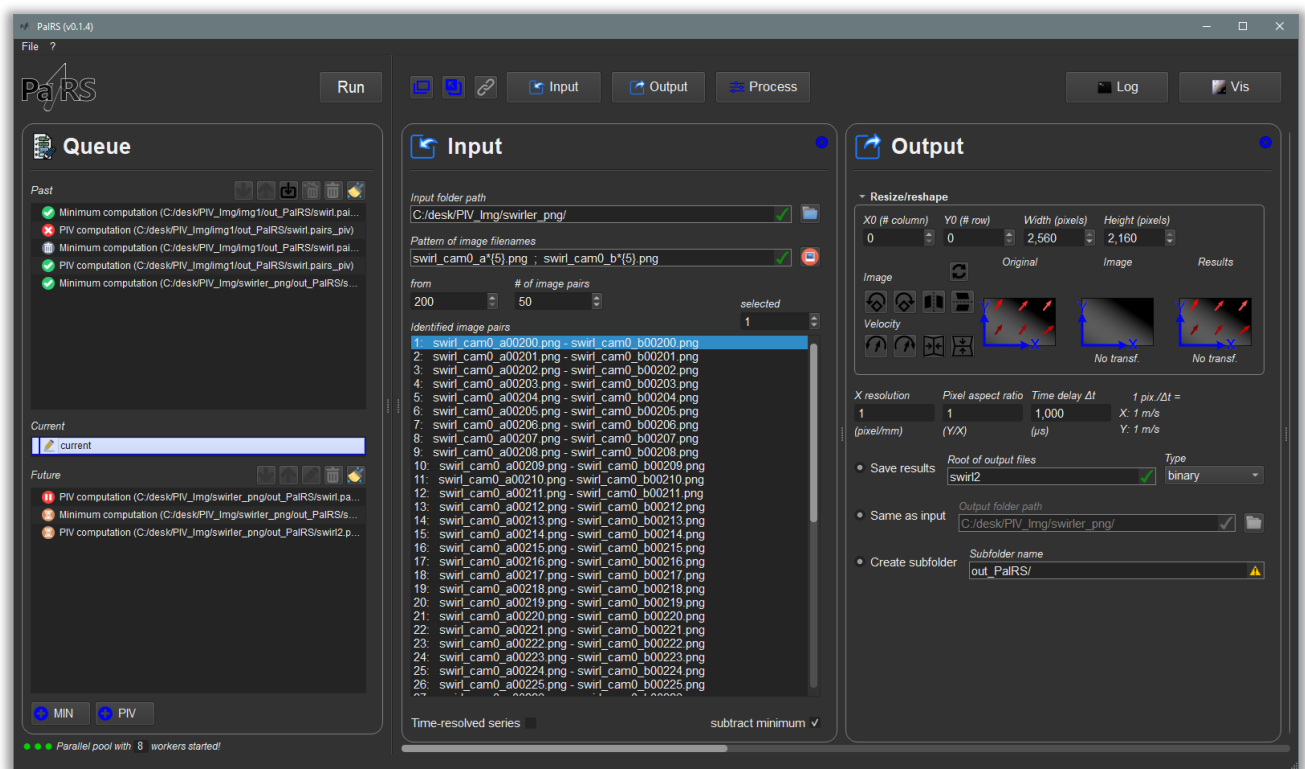

Figure 29 – PaIRS in the dark mode.

Moreover, it is possible to increase or decrease the font size of the graphical user interface via the following keyboard sequences:

    Ctrl+0:   Set the default font size (14 pixels).
    Ctrl+1:   Decrease the current font size by 1 pixel.
    Ctrl+9:   Increase the current font size by 1 pixel.
Ctrl+Shift+1:   Set the font size to the minimum value allowed (8 pixels).
Ctrl+Shift+9:   Set the font size to the maximum value allowed (20 pixels).

References

[1] Astarita, T., & Cardone, G. (2005). "Analysis of interpolation schemes for image deformation methods in PIV". *Experiments in Fluids*, 38(2), 233-243. doi: 10.1007/s00348-004-0902-3.

[2] Astarita, T. (2006). "Analysis of interpolation schemes for image deformation methods in PIV: effect of noise on the accuracy and spatial resolution". *Experiments in Fluids*, vol. 40 (6): 977-987. doi: 10.1007/s00348-006-0139-4.

[3] Astarita, T. (2007). "Analysis of weighting windows for image deformation methods in PIV." *Experiments in Fluids*, 43(6), 859-872. doi: 10.1007/s00348-007-0314-2.

[4] Astarita, T. (2008). "Analysis of velocity interpolation schemes for image deformation methods in PIV". *Experiments in Fluids*, 45(2), 257-266. doi: 10.1007/s00348-008-0475-7.

[5] Astarita, T. (2009). "Adaptive space resolution for PIV". *Experiments in Fluids*, 46(6), 1115-1123. doi: 10.1007/s00348-009-0618-5.

[6] Westerweel, J., & Scarano, F. (2005). "Universal outlier detection for PIV data". *Experiments in fluids*, *39*(6), 1096-1100. doi: 10.1007/s00348-005-0016-6.

About the authors

Gerardo Paolillo received a Master's degree in Aerospace Engineering and a PhD degree in Industrial Engineering from Università di Napoli "Federico II" in 2015 and 2018, respectively.

He is currently a Research Associate in the Department of Industrial Engineering at Università di Napoli "Federico II".

His research interests lie in the area of experimental fluid mechanics, with focus on applications of unsteady jets to flow control and electronics cooling, investigation into dynamics of turbulent Rayleigh-Bènard convection and development of 3D optical velocimetry techniques.

Tommaso Astarita received a Master's degree in Aeronautical Engineering in 1993 and a PhD degree in Aerospace Engineering in 1997, both from Università di Napoli "Federico II".

He was Post-doc at the von Kàrmàn Institute for Fluid Dynamics and he is currently full Professor of Fluid Mechanics at Università di Napoli "Federico II".

His main research interests are dedicated to the experimental study of problems in the fields of fluid mechanics and convective heat transfer, in particular, the application and development of IR thermography and stereoscopic and tomographic PIV techniques for fluid mechanics problems.