Last edit: November 05, 2024

# User Guide

## PaIRS – version 0.2.3



email: etfd@unina.it

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

# Contents


**1. Introduction** ........ **4**

**2. Related works** ........ **4**

**3. Installation** ........ **5**

**4. Run** ........ **5**

- From command prompt ........ 5
- In Python environment ........ 5

**5. Main Window** ........ **6**

**6. Menu bar** ........ **7**

- ........ 7
- File ........ 7
- Help (?) ........ 7

**7. Explorer** ........ **8**

- Copy process to loop over folders ........ 11
- Launch processing ........ 11
- Useful hints and clarifications ........ 13

**8. Step editor** ........ **14**

- Header ........ 14
- Tab button bar ........ 14
- Tab area ........ 15

**9. Input** ........ **16**

- Input folder tool ........ 16
- Image import tool ........ 17
- Image set tree ........ 18

**10. Output** ........ **20**

- Output file specifications ........ 20
- Resize/reshape ........ 20
- Resolutions and time delay ........ 21
- Physical region ........ 21
- Laser plane equation ........ 22

**11. Process (Image pre-processing)** ........ **23**

**12. Process (PIV and stereoscopic PIV analysis)** ........ **24**

- Simple mode ........ 24
- Advanced mode ........ 27
- Expert mode ........ 30

**13. Process (Disparity correction)** ........ **33**

**14. Log** ........ **34**

**15. Vis ........................................................................................... 36**
Plot tools ........................................................................................... 36
Field box ........................................................................................... 39
Plot area ........................................................................................... 40
Important remark ........................................................................................... 41
**16. Calibration ........................................................................................... 42**
**17. Input/Output (Camera calibration) ........................................................................................... 43**
**18. Process (Camera calibration) ........................................................................................... 45**
Target parameters ........................................................................................... 45
Calibration parameters ........................................................................................... 47
**19. Vis (Camera calibration) ........................................................................................... 51**
**20. CalVi: use of the interactive calibration tool ........................................................................................... 52**
Spot detection ........................................................................................... 52
Calibration ........................................................................................... 53
Point removal ........................................................................................... 53
Selection of the area for the auto-correlation mask ........................................................................................... 53
Origin not found or not in the image ........................................................................................... 53
**21. CalVi: calibration of pinhole + cylinder camera model ........................................................................................... 54**
Procedure 1 (recommended): separate pinhole camera model calibration and cylinder parameter estimation with different sets of target images ........................................................................................... 54
Procedure 2: separate pinhole camera model calibration and cylinder parameter estimation with the same set of target images ........................................................................................... 57
Procedure 3: simultaneous pinhole camera model calibration and cylinder parameter estimation ........................................................................................... 58
**22. Use of the calibrated mapping functions ........................................................................................... 59**
**23. Appearance ........................................................................................... 61**
**References ........................................................................................... 62**
**About the authors ........................................................................................... 63**

# 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 the module for the 2D planar PIV analysis and the stereoscopic PIV analysis and a module for optical calibration of camera systems, namely Calibration Visualizer (**CalVi**).

CalVi is the calibration module of PaIRS-UniNa and allows accurate optical calibration of single and multiple camera bundles with the camera models mostly used in the PIV community: polynomials, rational functions and the pinhole camera model. Among the other features, it supports camera calibration procedures working with unknown positions and orientations of the calibration target and the integration of the pinhole camera model with a refractive correction model for cylindrical geometries (based on ray-tracing and Snell's law).

PaIRS-UniNa is supported by Python 3.9+ 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. On the other side, PaIRS requires, among other packages, SciPy and matplotlib.

For further information, please visit PaIRS website.

# 2. Related works

Please cite the following works if you intend 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.

**[6]** Giordano, R., & Astarita, T. (2009). "Spatial resolution of the Stereo PIV technique". *Experiments in Fluids*, 46(4), 643-658. doi: 10.1007/s00348-008-0589-y.

Please cite the following works if you intend to use CalVi for your purposes:

**[1]** Paolillo, G., & Astarita, T. (2020). "Perspective camera model with refraction correction for optical velocimetry measurements in complex geometries". IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(6), 3185-3196. doi: 10.1109/TPAMI.2020.3046467.

**[2]** Paolillo, G., & Astarita, T. (2021). "On the PIV/PTV uncertainty related to calibration of camera systems with refractive surfaces". Measurement Science and Technology, 32(9), 094006. doi: 10.1088/1361-6501/abf3fc.

# 3. Installation

All PaIRS-UniNa wheels are distributed under LGPLv3+ licences. To install PaIRS, 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. Afterward, 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 /
```

# 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. 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.

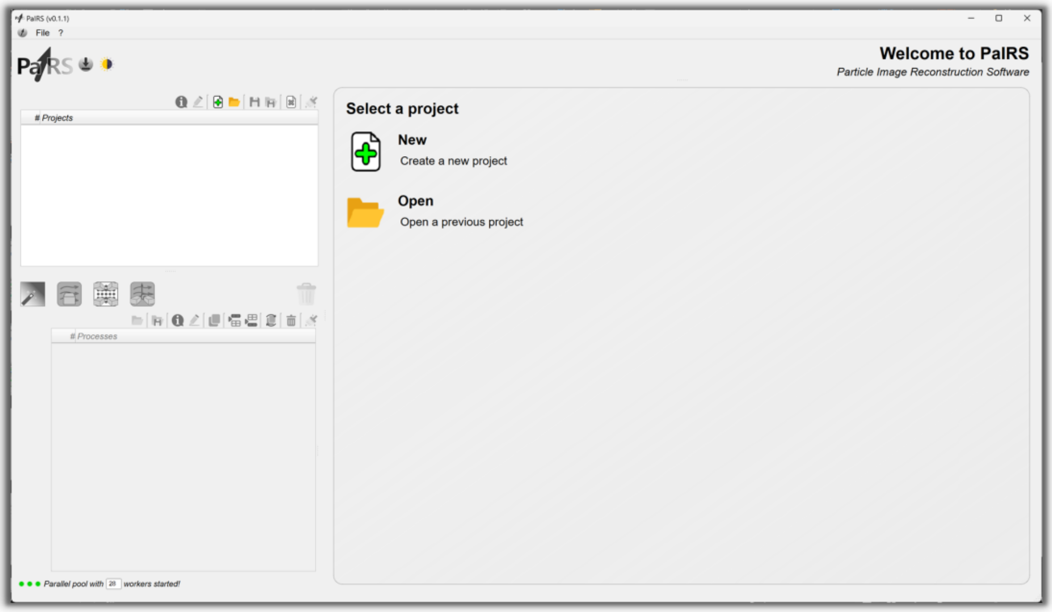



Figure 1 – PaIRS main window.

On the first launch or whenever PaIRS is started in clean mode, the main window opens to a welcome page. To start using PaIRS, you need to create a new project or load one previously saved on disk (file with the .pairs_proj extension). These actions can be performed from the welcome page or by using the buttons above the project tree in the PaIRS Explorer. The project will be added to the project tree in the current workspace. Alternatively, you can directly create a new workspace or load one previously saved on disk (file with the .pairs_wksp extension). These actions can be performed from the File menu in the main window. In the following, the menus of the PaIRS main window and the Explorer are presented in more detail.

# 6. Menu bar

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

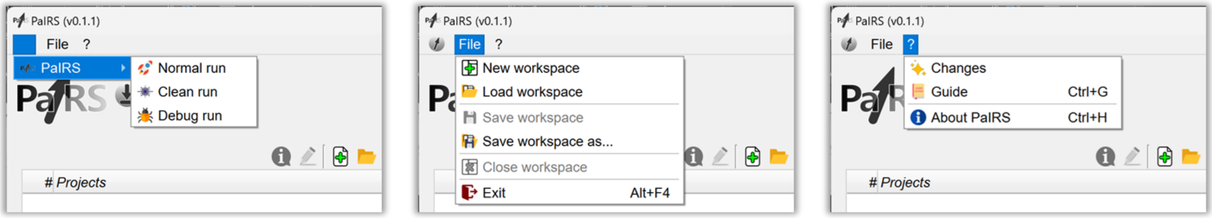


Figure 2 – Menu bar.

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

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

## File

The following functions are provided:

- *New workspace*: save the current worksapce and start a new one;
- *Load workspace*: load a previously saved workspace (.pairs_wksp file);
- *Save workspace*: save the current workspace;
- *Save workspace as...*: save the current workspace to a specified location (.pairs_wksp file);
- *Close workspace:* close the current workspace;
- *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`)[1]: if you are reading this document, you well know this function;
- *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. Explorer

The Explorer (Figure 3) consists of two trees: the project tree (1) and the process tree (2).

**Projects** are groups of **processes** that can be organized for the sake of order or to be executed entirely or partially in sequence. Projects can be modified using the buttons at the top of the project tree. Specifically, the following actions can be performed on projects:

- (`F1`) Check the selected project information regarding creation, last modification, and save date, the computer and account from which it was created and the corresponding version of PaIRS.
- (`F2`) Rename the selected project.
- (`Ctrl+N`) Add a new project to the tree.
- (`Ctrl+B`) Import a previously saved project from disk.
- (`Ctrl+S`) Save the selected project.
- (`Ctrl+Shift+S`) Save the selected project with a new name.
- (`Ctrl+X`) Close the project (removing it from the tree).
- (`Ctrl+Shift+X`) Close all projects (clearing the entire project tree).

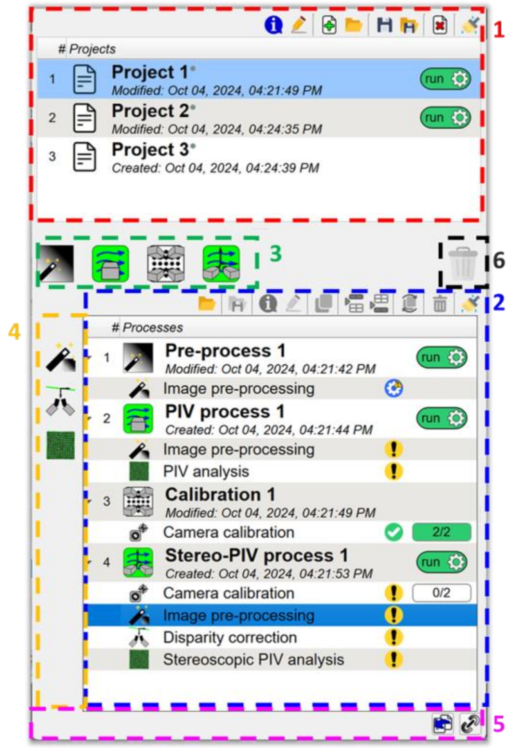


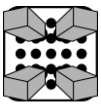

Figure 3 – The Explorer: **1) project tree**; **2) process tree**; **3) process bar**; **4) step bar**; **5) copy/link button bar**; **6) bin**.

The above actions can also be executed from the context menu that opens by right-clicking on the project tree.

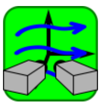
Processes are different types of analyses that can be performed in PaIRS, each consisting of one or more steps. In the current version of PaIRS-UniNa, 4 different types of processes are available:

**Pre-process**: pre-process analysis of a set of images aimed at computing historical minimum background.

**PIV process**: Particle Image Velocimetry analysis for computation of the two-dimensional two-component velocity field.



**Calibration** accurate optical calibration of single and multiple camera bundles.



**Stereoscopic PIV process:** stereoscopic Particle Image Velocimetry analysis for computation of the two-dimensional three-component velocity field**.**

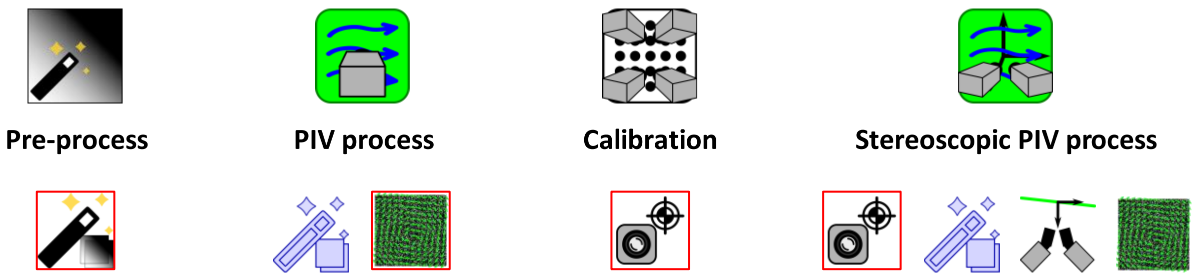
Each process can be added to the process tree (2) by dragging and dropping its corresponding icon from the process bar, i.e., the toolbar above the tree (3), or by double-clicking on the icon. Once added, the process can be edited. The buttons located at the top right above the process tree are used to modify the structure of the process tree, and the corresponding actions can also be executed from the context menu that opens by right-clicking on the tree:

- (`Ctrl+B`) Import a previously saved project from disk.
- (`Ctrl+Shift+S`) Save the selected process with a new name (please note that processes are automatically saved at the end of their execution and are, in any case, saved within their corresponding projects).
- (`F1`) Check the selected process information regarding creation, last modification, and save date, the computer and account from which it was created and the corresponding version of PaIRS.
- (`F2`) Rename the selected process.
- (`Ctrl+C`) Copy the selected processes (you can copy more than one process at a time).
- (`Ctrl+V`) Paste the copied processes below the selected one (you can also copy processes into different projects!).
- (`Ctrl+Shift+V`) Paste the copied processes above the selected one (you can also copy processes into different projects!).
- (`Ctrl+L`) Copy the following process in series, automatically updating the input path of the corresponding images to be analyzed (refer to the following paragraph for more details)
- (`Del` or `Backspace`) Delete the selected processes, moving them to the bin
- (`Shift+Del` or `Shift+Backspace`) Delete all the processes, moving them to the bin

When a process is selected, the steps that can be activated or deactivated for the selected process will appear in the step bar, i.e., left sidebar of the process tree (4). The possible steps for the considered processes are the following:

- **Image pre-processing**: step for computation of the historical minimum background for subsets of particle images corresponding to the same laser light source.
- **PIV analysis** step for analysis of a set of particle images via custom iterative multi-grid method aimed to compute the two-dimensional two-component displacement field.
- **Camera calibration:** step for selection of an appropriate camera model and estimation of the parameters of the mapping functions based on calibration target images.
- **Disparity correction:** step for computation of the laser sheet position and orientation to adjust the camera disparities in a stereoscopic optical setup**.**
- **Stereoscopic PIV analysis**: step for analysis of a set of particle images via custom iterative multi-grid method aimed to compute the two-dimensional three-component displacement field..

The different types of processes by default include the following steps:




The steps highlighted in red are mandatory, and when a process is selected, they are not displayed in the step bar. The other steps are displayed in the step bar and can be activated or deactivated as needed by simply clicking the corresponding icon in the sidebar or using the context menu that opens by right-clicking on the corresponding process item in the process tree.

When a step within a process is selected, the copy/link button bar may appear at the bottom of the process tree. The inherent buttons are used to copy or link the processes together. The copy/link button bar appears if there are further steps of the same type present in the process tree:

- The copy button allows you to copy the settings of the selected step from another active step in the process tree that is not linked to another step.
- The link button allows to create a connection between the selected step and another step of the same type present in the process tree. The selected step (slave step) becomes dependent on the latter step (master step): in this way, any modifications applied to the master step are automatically applied to the slave step. The slave step is identified by the icon reported in the first column of the tree. When selecting the slave step, a message in the copy/link button bar will display the corresponding master step. The slave step is updated based on the changes to the master step even during the PaIRS running phase. Therefore, if the slave step is necessary for executing subsequent steps within the same process, the process including the slave step should be positioned in the tree after the process to which the master step belongs. To remove the link between the slave Step and the master Step, simply click the link button again (it will have the icon ).

Deleted processes are moved to the bin (tree of deleted processes). You can access the bin by clicking the icon located in the top right corner above the process tree (6). It is also possible to display the bin by moving the splitter at the right side of the process tree to the left, which allows to simultaneously view both the process tree and the bin, as shown in Figure 4.

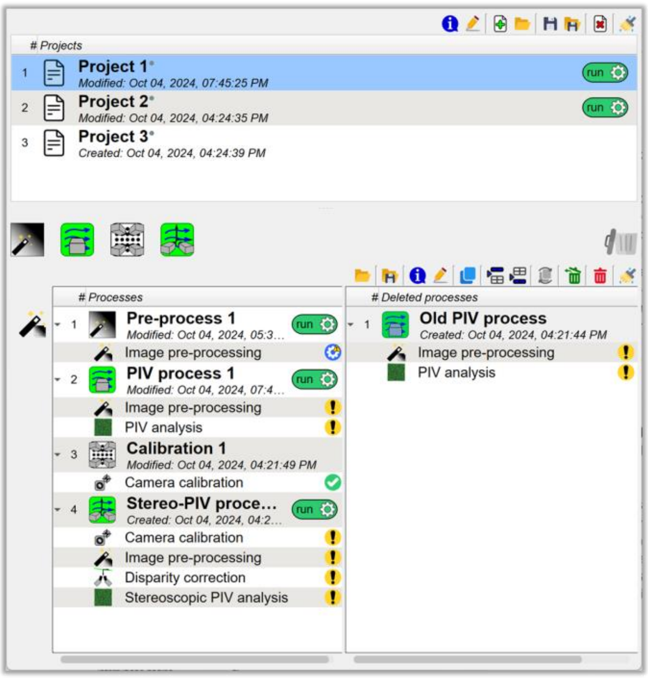



Figure 4 – Simultaneous visualization of the process tree and the bin.

The bin is a tree similar to the process tree, so the same actions as in the process tree are supported. Nevertheless, it is not possible to add new processes to the trash or perform copy/link operations on deleted processes. It is possible to restore deleted processes using the dedicated button in the bar above the process trees. When both trees are displayed simultaneously, you can delete (or restore) processes by dragging them from the process tree to the trash (or from the trash to the process tree). Please note that once processes are deleted from the trash, they cannot be recovered. Likewise, cleaning the trash is an irreversible operation.

Please also note that when processes are deleted and moved to the trash, the links between processes are removed, as links cannot exist between active and deleted processes.

## Copy process to loop over folders

This special feature allows you to create multiple copies of the same process while automatically changing the input path of the images in the specified steps. As a result, it is very useful when the same analysis needs to be performed on a considerable number of experimental tests conducted under similar conditions.

By clicking the button, a first dialog window (Figure 5, left) will open where the user must select the folders for which an updated copy of the selected process will be created. After selecting the folders, a second dialog window (Figure 5, right) appears in which the user must specify, for each step of the selected process, the type of replication they want to perform:

- an independent copy of the step;
- a linked copy of the step (the starting step acts as a master step and the copied one as a slave);
- an updated copy of the step where the input path of the images is modified with the path of the newly selected folder.

After configuring the type of replication, the user can press the "Proceed" button to start the multiple copy operation. With this operation, the new processes are added to the tree and renamed with the same name as the process from which the copy originates, followed by the folder name in parentheses.

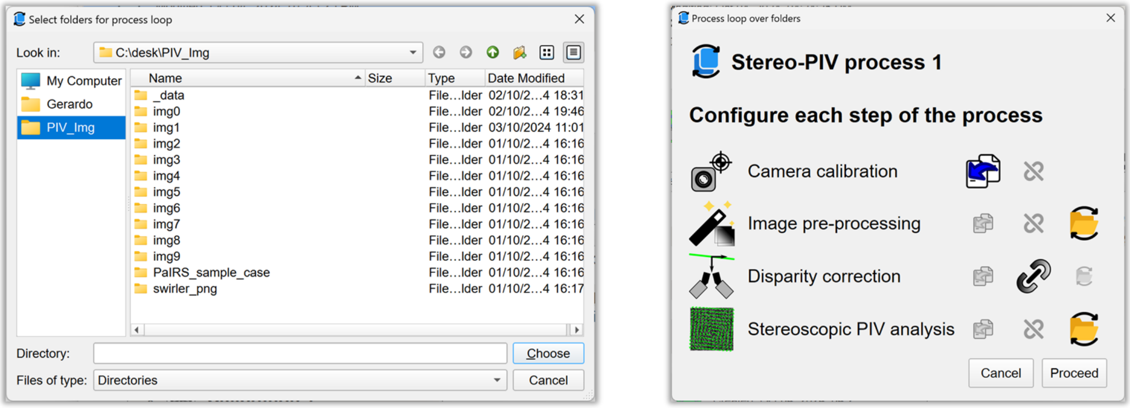



Figure 5 – Dialog windows for copying process to loop over folders. Left: dialog for folder selection; right: dialog for setting up the type of replication for each step of the process.

## Launch processing

Once the projects are fully set up, the user can click the *Run* button (`Ctrl+Return`) to start data processing in PaIRS (Figure 6, left). The projects are executed sequentially and the processing can be stopped in any moment by clicking the pause button (`Ctrl+Return`), as shown in Figure 6, center.. It is possible to resume the processing by clicking on the play button, as shown in Figure 6, right.

The user can select which projects and which processes within each project to execute during data processing by simply toggling the switch that appears next to them in the corresponding trees:

- run — the run state indicates that the project/process will be executed during data processing;
- skip — the skip state indicates that the project/process will be not executed during data processing.

These switches are only visible for projects and processes that have not yet been completed. When a project is toggled between the two above states, all processes within it are set to the same state.

During data processing, the current project and the current process are identified by the spinning of the wheel icon inside the corresponding switch. Relevant information about the execution status of each step within a process can be viewed by selecting the step and inspecting the Log tab (see Section 13 for more details), however information on the progress status is also conveyed in the process tree via the status icon and the progress bar for each step. By clicking on the status icon further information about the status progress is displayed in a separated dialog window.

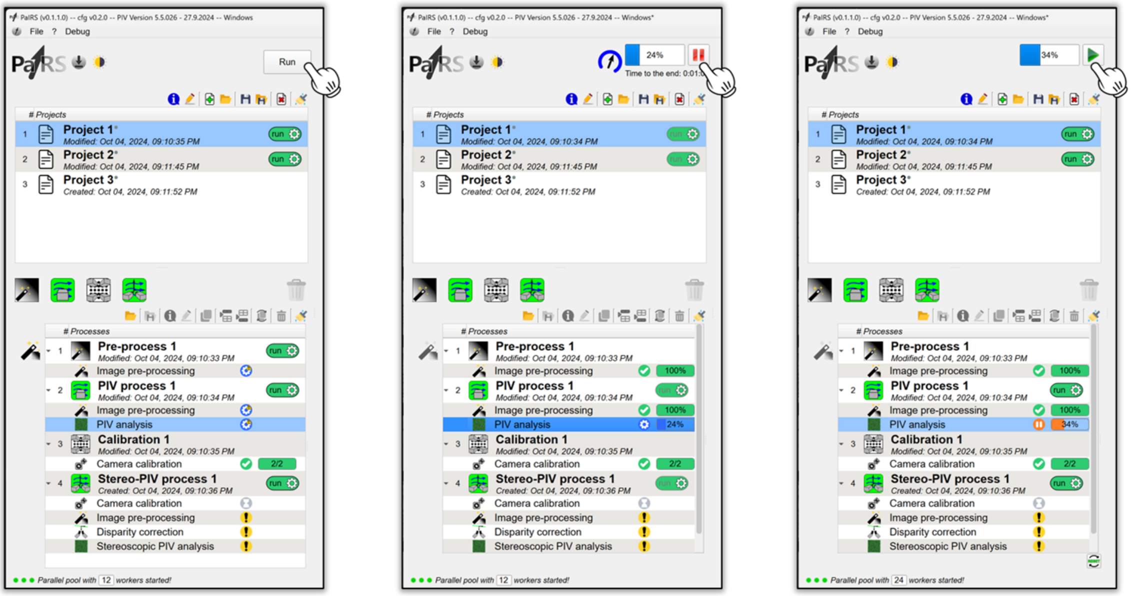


Figure 6 - Running (left), stopping (center) and restarting (right) processing in PaIRS.

The status icon can change depending on the cases listed below:

- the step has not yet been initialized. This state appears when the corresponding process is added to the process tree and before the user accesses and configures the step for the first time. An uninitialized step **will not be executed** during data processing;
- the step has critical issues that will prevent its execution. Examples of critical issues include missing input files or incorrect output paths. Additionally, improper or incomplete configuration of a previous step within the same process may result in a critical warning that prevents the step execution. A step with critical issues **will not be executed** during data processing;
- the step is ready to be executed, however, there are minor issues that the user should check, such as the existence of output files with the same name root in the same path specified in this step that could be overwritten during the data processing;
- the step is ready to be executed, and no critical or minor issues were identified during the parameter setup. During the step execution, this is the displayed icon and the wheel rotates while the progress bar fills as the execution progresses;
- the step was paused by the user. This icon is also displayed for the steps that will be executed later during the data processing phase;
- the step was completed successfully without any issues;
- the step was executed, but some critical issues occurred during execution. The step may not have been 100% completed, with the completion percentage shown in the progress bar, or if the process was fully completed, it may be obsolete (for instance, the input and output paths are no longer valid). In the latter, it is recommended that the user removes the corresponding process from the tree.

The progress bar reflects the advancement of the execution when the step is in progress (in this case the status icon is the spinning ). The bar that appears next to the stop button in the Explorer header refers to the execution of the current step and below it an estimated time remaining for the step completion is displayed.

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 at the bottom of the Explorer.

## Useful hints and clarifications

- Projects and processes can be reordered by dragging and dropping the related items within their respective trees. This is only possible when PaIRS is not actively running analyses (i.e., when data processing is paused).
- Steps that have not yet been modified for the first time are marked with the symbol . When another step within the same process is modified, the changes are automatically transferred to the other steps of the same process, making it faster to configure all the steps of the same process. However, as soon as any modification is applied to a specific step, the symbol disappears, and further changes applied to the other steps will no longer be transferred to this initialized step. It is recommended to make as many common changes as possible to a single step of the process, before adjusting the parameters of the other steps individually.
- It is possible to load processes and projects previously saved in the disk, but this is not possible for individual steps within a process. However, steps are saved inside processes and processes are automatically stored in the disk at the output folder path every time they are completed (with or without critical errors). As a result, you can load a process (either manually saved or automatically saved upon completion) into the current project and then copy each of its steps to another process present in the same project. If you need to copy the parameters from the steps of a process in another project, you can switch to that project, copy the process (use the shortcut `Ctrl+C`), return to the first project, paste the copied process (using the shortcut `Ctrl+V`) and then copy the desired step from it. Once finished, you can delete the loaded or copied process!
- Historical minimum background images are subtracted during the disparity correction, PIV and stereoscopic PIV analysis steps only if the image pre-processing step is activated and executed. If you have previously computer the historical minimum background images and you would like to skip the potentially time-consuming pre-processing step, you follow this approach: in the pre-processing step, under the Input tab, import a list containing only one image per frame and camera, corresponding to the respective background image, then in the Process tab set the *minimum allowed value* and the *minimum allowed standard deviation value* threshold both to zero. The step will complete in just a few seconds without generating critical errors!

# 8. Step editor

The step editor (Figure 7) is the part of the PaIRS main window where the user can modify the parameters related to each step of every process. The editor consists of:

1. a header, displaying information about the current workspace and the currently selected step;
2. the tab button bar, which allows navigation through the tabs to modify each step;
3. the tab area, a scrollable area containing the tabs to set the parameters of the current step.

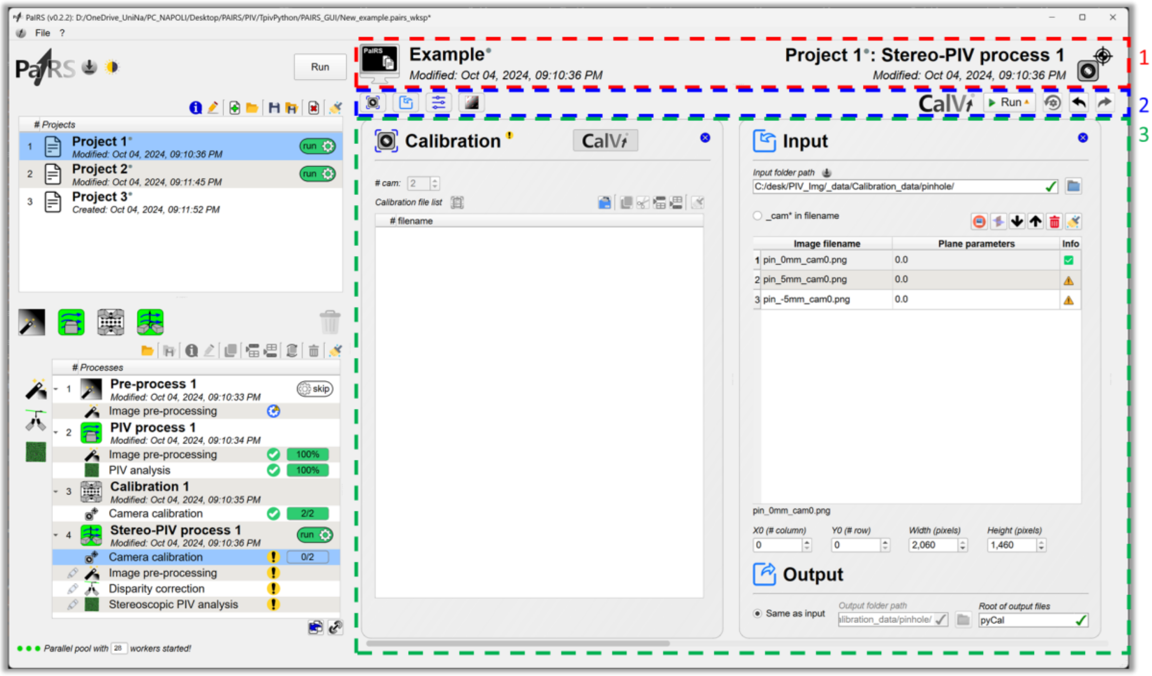



Figure 7 – The editor: **1) editor header**; **2) tab button bar**; **3) tab area**.

## Header

The step editor header displays information regarding the current workspace on the left and details about the current step on the right. The workspace information is displayed only when the current workspace has been previously saved on the disk (this is possible using the File menu). A gray dot next to the workspace name indicates that changes have been made within the workspace itself, but it has not been saved yet. By clicking the icon it is possible to view detailed information, including the creation, last modification and save dates of the workspace. The user can also double-click the workspace name to modify it. Indeed, this is the only way to choose the name of your workspace.

Similarly, for the current step details, the user can click the corresponding icon on the right of the header and view more related details. In this case, the name of the project and the process to which the step belongs is displayed and it is not possible to modify them by double-clicking (in order to change the names of the project and the process the user should use the corresponding trees). A gray dot next to the project name indicates that changes have been made, but it has not been saved yet

## Tab button bar

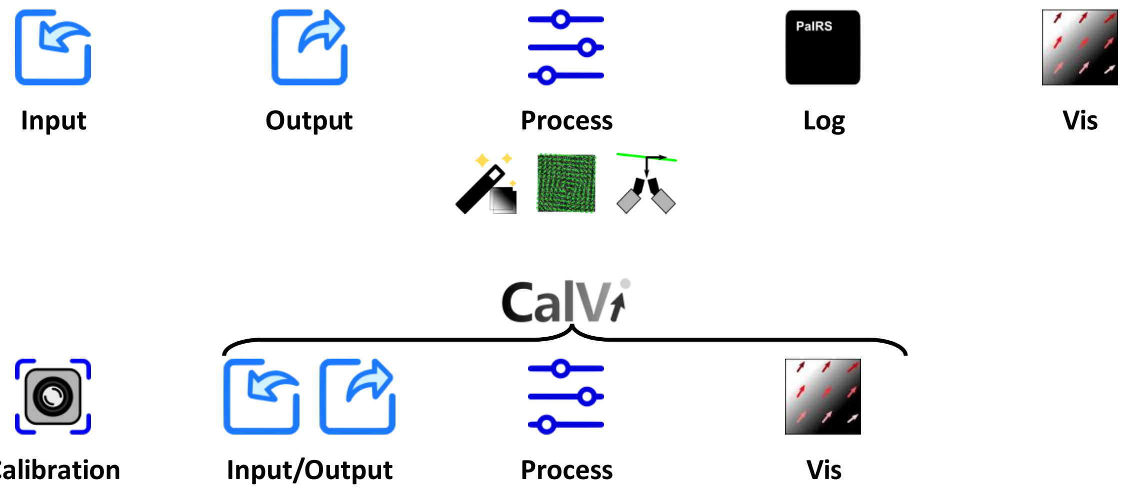
This bar contains the buttons to access the different tabs to set up the current step (on the left part) and additional buttons to run the additional modules (in the present release, they include only CalVi for accurate camera calibration) and the undo/redo buttons (on the right par).

The tab buttons are used to show the tabs. The tabs currently hidden are denoted by a padlock symbol ( ).

The undo/redo buttons allows to display previous/subsequent configurations for the step parameters By right-clicking on them, it is possible to open a list reporting the 10 configurations immediately preceding or following the one currently displayed. Please notice that **to revert to one of these configurations**, **you need either to click the button or to set any of the parameters within the tabs related to the current step** (for instance, you can press enter in an edit line or a spin box or select and activate a combo box). Until one of these two actions is performed, the previous or subsequent configuration is only displayed but not restored.

## Tab area

The tab area is the part of the step editor that contains the tabs for modification of the parameters for each step. Each different type of step contains different tabs. The possible tabs are listed below.




**Click on the tab logos to navigate to dedicates sections**

More details on the parameters that can be configured in each tab are provided in the following paragraphs.

# 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 Input tab consists of 3 sections (Figure 8):

1. the input folder tool, which allows to specify the path of the folder containing the images to be processed and to analyze it to check for any missing files;
2. the image import tool, which allows to generate a list of input image files in a user-friendly way;
3. the image set tree, where the list of the image files included in the present set is displayed and can be further modified.

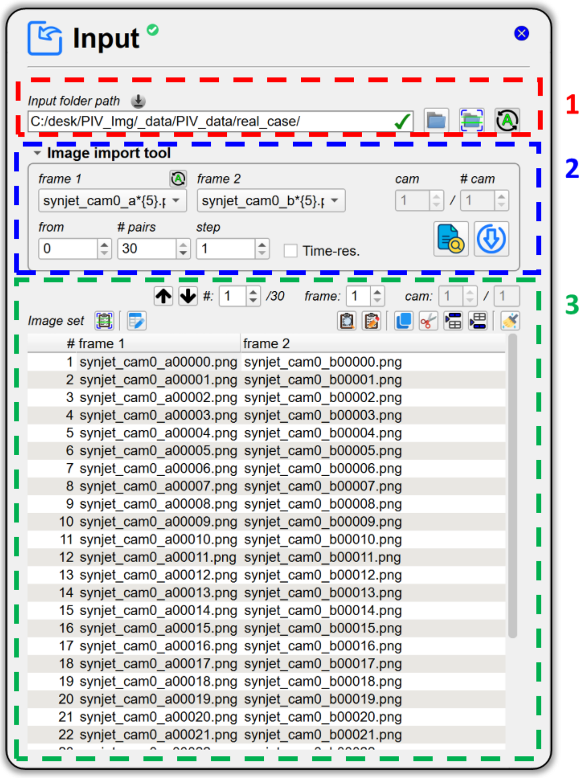



Figure 8 – The Input tab and its sections

## Input folder tool

The user can specify the full path of the input folder containing the image files 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). The results of such an analysis are transferred to the image import tool box in such a way that the user can change the working image set in a fast and intuitive way.

The button .. can be used to repeat the analysis of the specified input path; this may help if any change has been made in the corresponding folder. If the button ..is activated, when changing the input folder path, PaIRS automatically identifies the image set in the new folder with the largest number of files and assign it to the current step, updating not only the image import tool, but also the image set tree.

## Image import tool

The image import tool allows for generation of a list of input image files in a user-friendly way, based on the results of the analysis of the input folder executed automatically by PaIRS upon changing the specified path in the related line box.

**Important remark:** no changes are made to the current image set until the button .. is pressed!

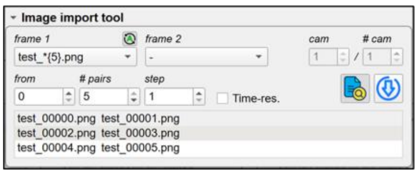
The principle of the image import tool usage is the following. The user first selects the number of cameras (if applicable for the current process), the pattern for each frame and each camera of the set, the intial and final identifier numbers and the spacing and, if this apply to the current case, activates the *Time-res.* Checkbox. Then, by clicking , an example list of image files is generated and the user can check if the result matches the expected outcome; if not, the image import parameters can be adjusted. Finally, once obtained the desidered result, the list can be generated and transfer to the below tree by clicking .

There is only one requirement that image filenames must meet for their pattern to be recognized during the path analysis and added to the *frame 1* and *frame 2* combo menus of the image import tool: image filenames must include a sequential number (a sequence of digits of arbitrary length). The automatic assignment of *frame 1* and *frame 2* is based on the recognition of two identical patterns that differ, apart from the digits of their sequential number, by a single letter, which is assumed to identify the frame. The automatic assignment of cameras is based on the recognition of patterns that differ, apart from the digits of their sequential number, by a single number, which is assumed to identify the camera. When a new folder is analyzed and the button in the folder path tool is active, the image set with the largest number of files is assigned to frame 1 and camera 1 and, using the above criteria, the other frames and cameras are assigned accordingly. If the number of images related to different patterns is the same (as is usually the case for double frame and multicamera image sets), this assignment proceeds alphabetically.

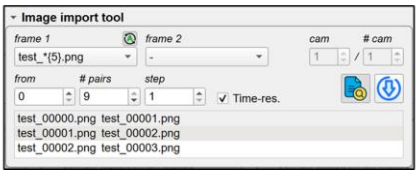
If the image set automatically identified by PaIRS is not the desired one, it is possible to use the image import tool to modify the set based on the patterns automatically recognized by PaIRS itself. To assign the frames of each camera, you need to use the *frame 1* and *frame 2* combo boxes and switch between the cameras using the *cam* spin. If the button above the *frame 1* combo is activated, each time the pattem of frame 1 of a given camera is selected, the patterns of frame 2 of the same camera and the frames of all other cameras are automatically assigned, proceeding in an alphabetical order. By deactivating this button, automatic selection can be avoided, allowing the user to manually choose the frames for all cameras.

For all frames and all cameras, it is possible to select the initial and final values of the sequential number identifying the images, as well as the step for generating the list of images through the *from, to* and *step* spins, respectively. If a step greater than 1 is set, it is possible to skip step images at a time in the image list generation. The *Time-res.* checkbox should be activated in case of time-resolved sequences. Figure 9 reports some examples for generating different types of image lists.

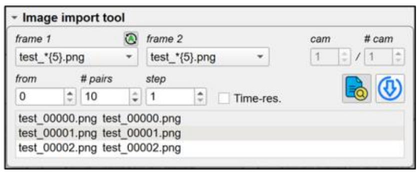
Single-frame sequence




automatic selection

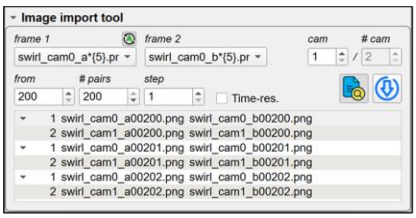



time-resolved sequence

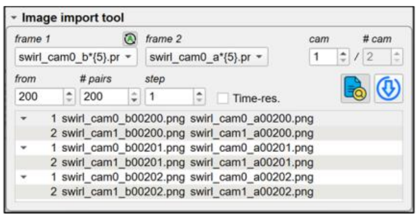



auto-correlation

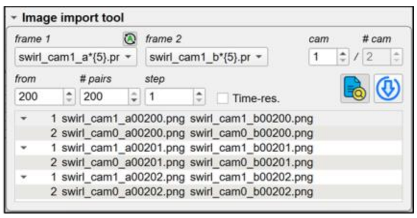
Double-frame sequence




automatic selection




change of frame 1 and frame 2




change of cam 1 and cam 2

Figure 9 – Examples of generation of different types of image lists.

## Image set tree

The image set tree allows navigation through the files of the current image set and enables advanced modifications to the image list generated by the image import tool.

To navigate through the files, the user can employ the buttons and the spins for image number, camera number and frame number. The button (`Ctrl+Up`) moves the selection to the top of the image list, while the button (Ctrl+Down) moves it to the bottom.

For advanced modifications to the image list, the user can use the buttons above the image set tree. These buttons perform specific operations, which can also be accessed through the context menu opening by right-clicking:

- (`F5`) perform an analysis of the current image set to identify any missing files on the disk.
- (`Ctrl+W`) display the next pair of images with missing files;
- (`Alt+X`) remove all the pairs of images with missing files;
- (`F2`) access the mode to modify individual image lists by frame and camera;
- (`Ctrl+T`) read a file containing an image list from the disk. This file must be formatted as follows: Each line should contain the image pairs for each camera. Frame 1 and Frame 2 must be separated by a comma, and the image pairs for different cameras must be separated by a semicolon. Below is an example of such a file:

```
swirl_cam0_a00202.png, swirl_cam0_b00202.png; swirl_cam1_a00202.png, swirl_cam1_b00202.png
swirl_cam0_a00203.png, swirl_cam0_b00203.png; swirl_cam1_a00203.png, swirl_cam1_b00203.png
swirl_cam0_a00204.png, swirl_cam0_b00204.png; swirl_cam1_a00204.png, swirl_cam1_b00204.png
...
```

- (`Ctrl+S`) save the current image set to an image list file;
- (`Ctrl+C`) copy the selected items from the tree;
- (`Ctrl+X`) cut the selected items from the tree;
- (`Ctrl+V`) paste the copied or cut items below the selected row
- (`Ctrl+Shift+S`) paste the copied or cut items below the selected row
- Clean the whole list.

Figure 10 illustrates the edit mode for image list related to specific frame and camera, accessible by clicking the button . In such mode, the spins for camera number and frame number allow to switch between the image lists on which to operate. Moreover, the following additional operations can be executed on the image list:

- (`Ctrl+R`) read single or multiple image files from the disk and import to the current list;
- (`Ctrl+Q`) sort image filenames in alphabetic order;
- (`Ctrl+Alt+Q`) sort image filenames in reverse alphabetic order;;

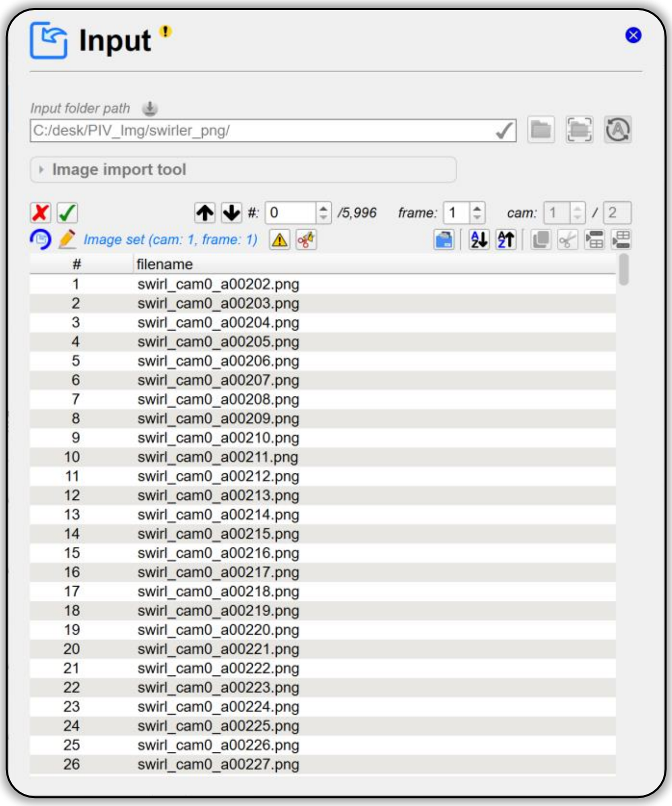



Figure 10 – Edit mode for image list related to specific frame and camera.

During the editing of the image list for individual frames and cameras, the interface is frozen, and it is not possible to access other tabs or sections of the main interface. In this case, the frame and cam spin boxes can be used to switch between lists, and elements can be copied or cut between different lists. Once the modifications to the various lists are complete, the following buttons can be used to confirm or discard the changes:

 discard all changes made since entering single list edit mode and restores the previous list;

 confirm the changes made. If the number of images in any list for a specific frame and camera exceeds the others, missing elements will be filled with empty strings, resulting in a warning in the global list.

In conclusion, it is noted that the order of elements in the trees can be modified both in global list edit mode and in single list edit mode, for frame and camera, by dragging and dropping the elements onto the tree list.

# 10. Output

The Output tab (Figure 11) allows the user to:

1. select the location, the name and the type of the output files

and, for the disparity correction and PIV and stereoscopic PIV analysis steps, also to:

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

and, for the disparity correction and stereoscopic PIV analysis steps, also to:

4. indicate the limits of the physical region in which to perform the disparity correction or the displacement computation;
5. specify the initial guess values for the disparity correction and the final desired values for the stereoscopic PIV analysis of the constants of the laser sheet plane equations.

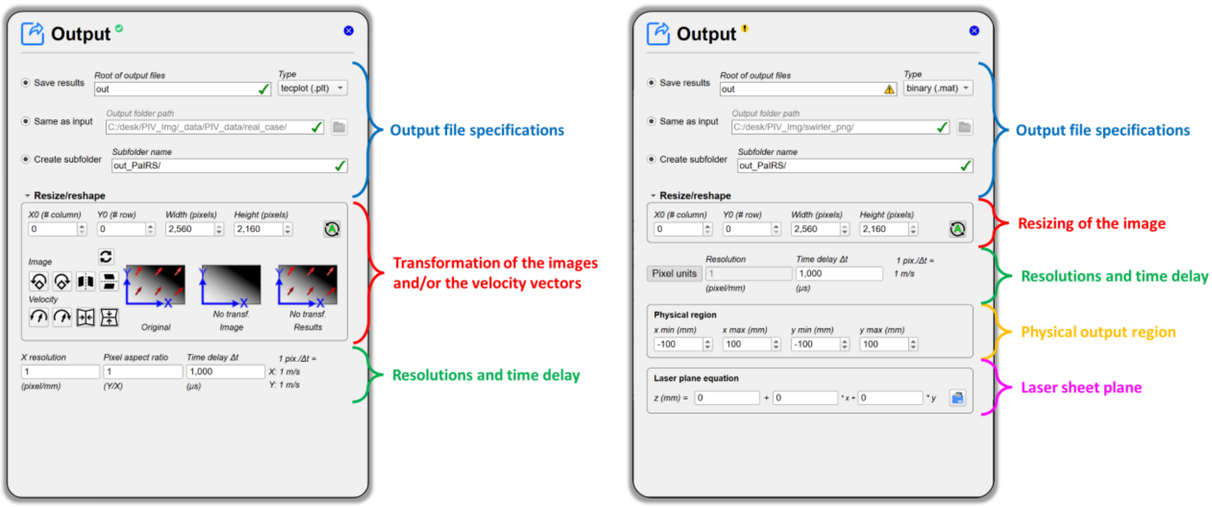



Figure 11 – The Output tab and its sections.

## Output file specifications

In this section (Figure 11, 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.

## Resize/reshape

In this section (Figure 11, red block), 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.

In the case of PIV analysis, transformations of the input images and the output velocity fields can be performed via the following buttons (Figure 11, right, red block):

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

In this section (Figure 11, green block), the user is asked to specify the image resolution and the time delay in μs ($\Delta t$).

In a PIV process (like that in Figure 11, left), since camera calibration is not mandatory, the user must specify the resolution along the two directions of the image. This is done in the Output tab by indicating the resolution along the *X* direction in pixel/mm units ($R_X$), the pixel aspect ratio ($\lambda$)

**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 $\tilde{U}$ and $\tilde{V}$ the corresponding velocity components in pixel/frame units, then the conversion from $\tilde{U}, \tilde{V}$ to $U, V$ is given by:

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

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

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

$R_X = \lambda = 1$ and $\Delta t = 1000$ (for output in pixel/frame units)

In a stereoscopic PIV process (like that in Figure 11, right), since camera calibration is mandatory, the user only needs to specify whether the results must be provided in pixel units or physical units. In the latter case, the resolution is automatically computed by PaIRS employing the calibrated camera models (which is done in the first step of the process, i.e., the camera calibration step).

## Physical region

In this section (Figure 11, right, yellow block), the user must specify the boundaries of the planar physical region in terms of limits of the $x$ and $y$ coordinates where to compute disparity correction or the 2D three-component displacement field for stereoscopic PIV.

This region is chosen in the laser plane (the equation of which is specified in the laser plane equation box of the same Output tab). However, its projection onto on the image planes of both the cameras, must always be entirely contained within the image. PaIRS performs a check of this property, and if the projection of the physical region falls outside one of the two image areas, a warning message is returned in this block. Such an error condition is critical, and in its presence, it is not possible to execute either the disparity correction step or the stereoscopic PIV step.

The limits of the selected physical region can be visualized in the VIS tab by activating the button . Figure 12 shows an example where this critical condition occurs, namely, the projections of the physical region on the image planes of the cameras extend beyond the images themselves.

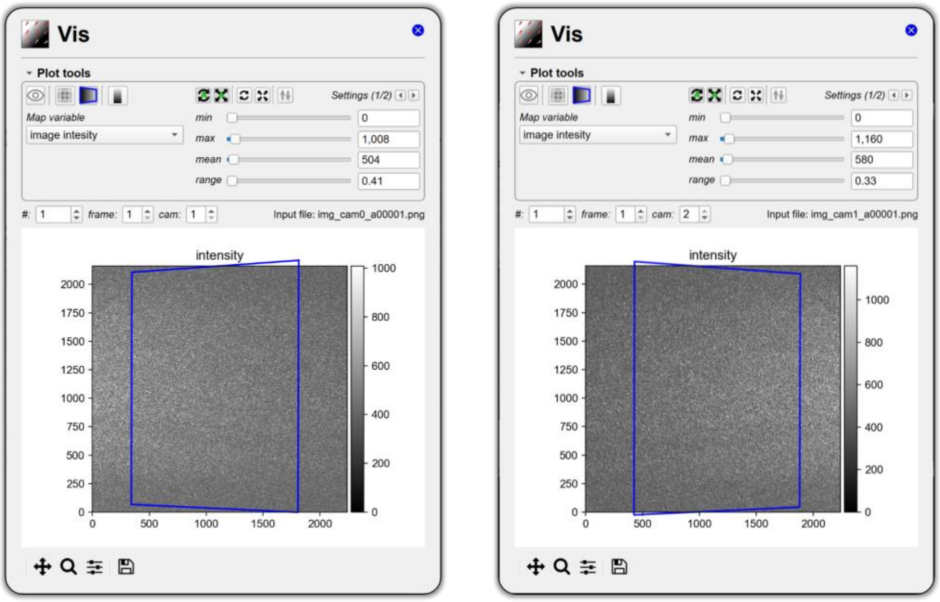



Figure 12 – Example of critical condition in assignment of the $x$ - $y$ limits of the physical region: the projections of the latter fall outside both images of the two cameras.

In the present release, there is no automatic procedure to determine an optimal physical region contained in the images of the two cameras. The user is required to manually adjust the limits of this physical region until it is entirely contained within both images.

## Laser plane equation

In this section (Figure 11, right, magenta block), the user must indicate the values of the constants of the laser sheet plane equation. Such an equation is formulated in the following form:

$$z = z_0 + z_x \cdot x + z_y \cdot y$$

where the coordinates $x$, $y$ and $z$ are expressed in mm, $z_0$ is also in mm, while $z_x$ and $z_y$ are non-dimensional. The user is asked to specify the values of $z_0$, $z_x$ and $z_y$; this can be done by inserting their values in the corresponding spins or by reading a disparity correction result file (.clz with extension) from the disk by clicking the button .

For the disparity correction step, such values represent only an initial guess and the analysis performed in this step provides an estimation of these constants. The values $z_0 = z_x = z_y = 0$ are generally a good starting point, unless significant misalignment is present between the $z = 0$ plane (reference plane for camera calibration) and the laser sheet plane.

In the case of stereoscopic PIV analysis, the laser equation plane together with the limits of the physical region identify the 2D region where the displacement field is computed and thus the correct knowledge of such constants is fundamental. If the disparity correction step in the stereoscopic PIV process is activated, in stereoscopic PIV analysis step the laser equation plane box is disabled and the plane constants are derived automatically from the disparity step. If the disparity correction step is deactivated, the user must specify the plane constants directly in this step.

# 11. Process (Image pre-processing)

Image pre-processing consists in the historical minimum background computation. This is a method used to generate a background image by calculating the minimum pixel values across a sequence of images, typically from a video or a set of frames captured over time. In PaIRS, such a computation is performed for each frame and each camera of the set. Additionally, images related to different frames can be combined if they correspond to the laser source, as occurs in time-resolved sequences obtained using a single high-frequency laser.

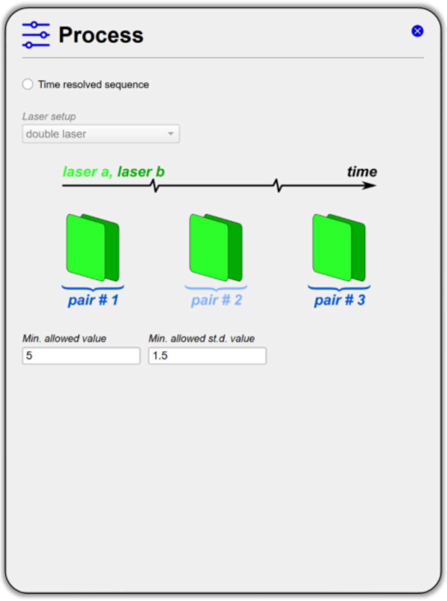
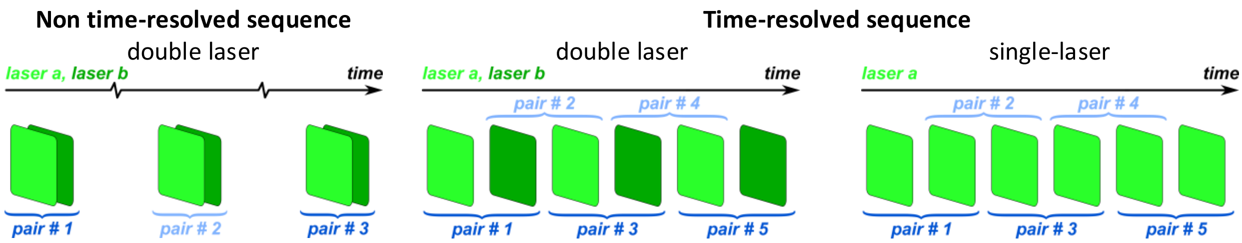



Figure 13 – Process tab for the image pre-processing step.

The process parameters for the image pre-processing can be specified in the Process tab (Figure 13). These parameters include the type of sequence (time-resolved or not) and the type of laser setup (single or double laser). The possible configurations are reported below:




The specification of the configuration type impacts both how the background image for each frame and camera is computed and how it is used in the corresponding analysis step. Additionally, the user must define two thresholds for the historical minimum background computation:

- the *minimum allowed value*: images with an average value below this threshold are excluded from the background computation;
- the *minimum allowed standard deviation value*: images with a standard deviation of their values below this threshold are excluded from the background computation.

# 12. Process (PIV and stereoscopic PIV analysis)

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 14. 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]. In the current release, stereoscopic PIV analysis is based on a mapping approach**,** in which the dewarping of the images is performed directly into the PIV algorithm, avoiding the loss of image quality. Further details are provided in [6].

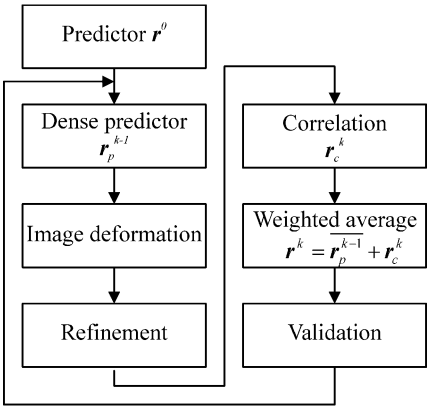



Figure 14 – Schematic flowchart of a possible image deformation method for PIV.

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 15.

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.

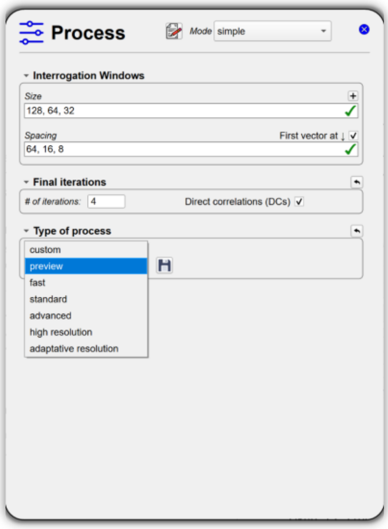



Figure 15 - Process tab for PIV and stereoscopic PIV analysis in the simple mode.

#### *Interrogation Windows*

In this box, the user must choose (refer to Figure 16):

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;
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.

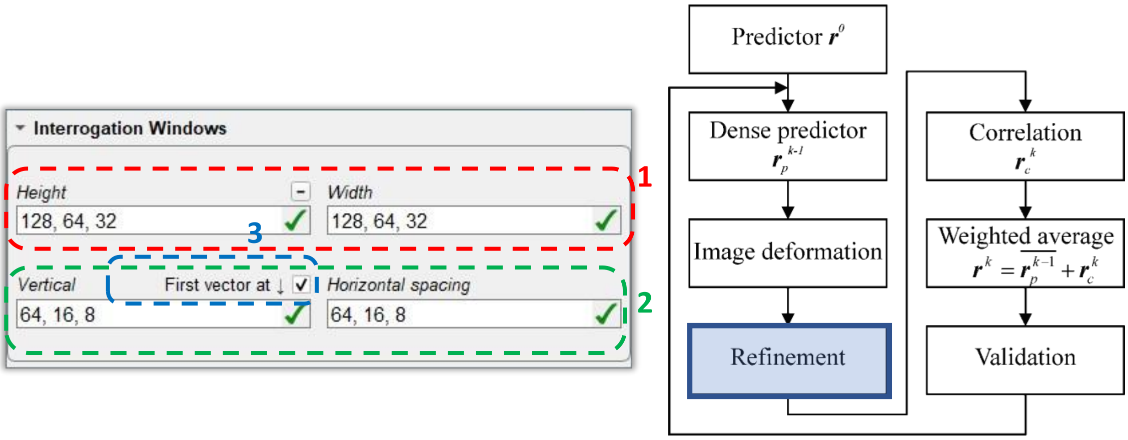



Figure 16 – Interrogation Windows box.

***Final iterations***

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

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.

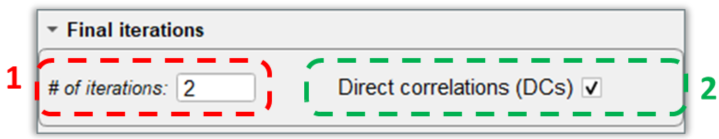



Figure 17 – Final iterations box.

***Type of process***

In this box, the user can (refer to Figure 18):

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  ) or save the current settings as a new custom type (by clicking on the button  ) in such a way that they can readily recovered at a later time.

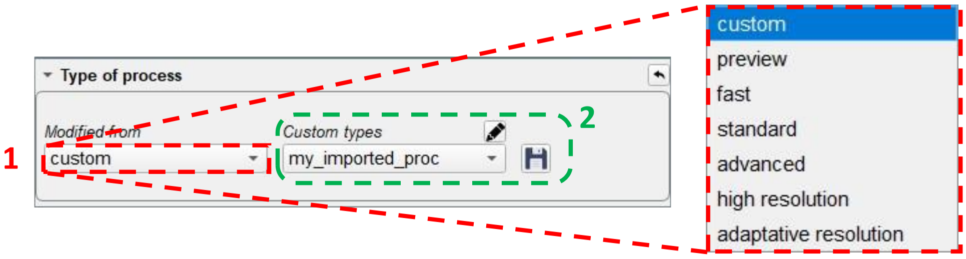



Figure 18 – Type of process box.

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  the user can access *Custom types of process* dialog box, illustrated in Figure 19.

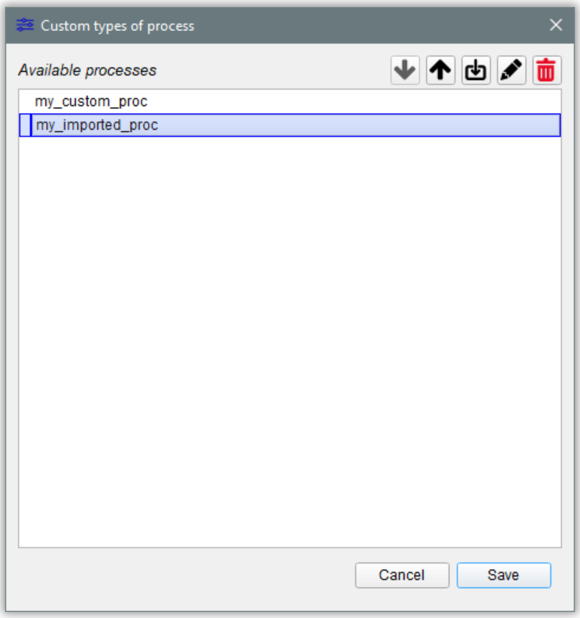



Figure 19 – Custom types of process dialog box.

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. Please notice that the ".cfg" input configuration files can be saved by clicking on the button located in the header of the Process tab (at the left side of the *Mode* combo).

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 20):

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 21.
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 |

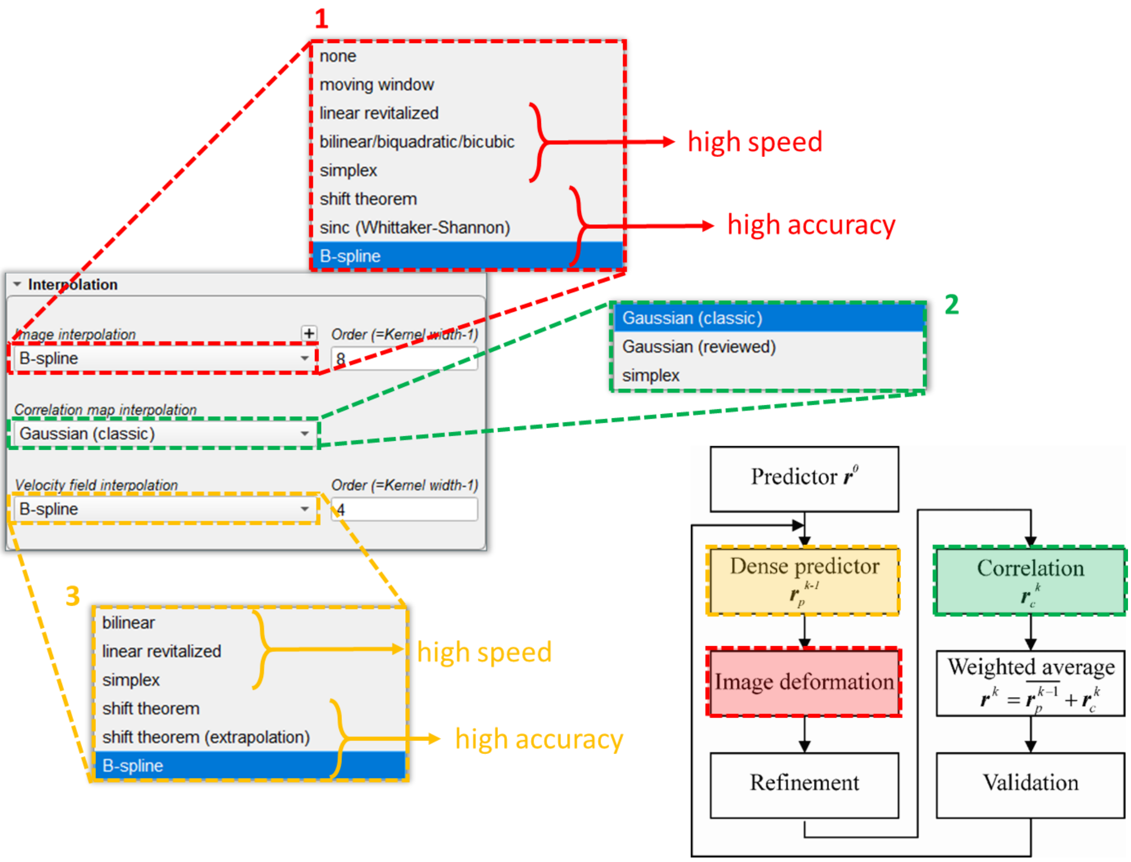



Figure 20 – Interpolation box.

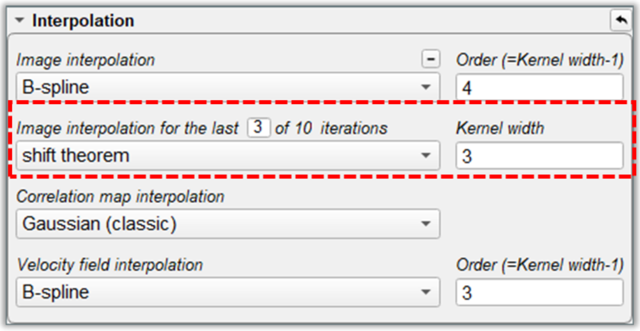



Figure 21 – Section for selecting a different type of image interpolation for a specified number of final iterations.

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 22):

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 $\alpha$ 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 $\varepsilon$. A high value increases the occurrence of outliers detected as correct vectors. The value 0.1 is recommended; as a general rule, values of $\varepsilon$ 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:
     - 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;
     - 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.

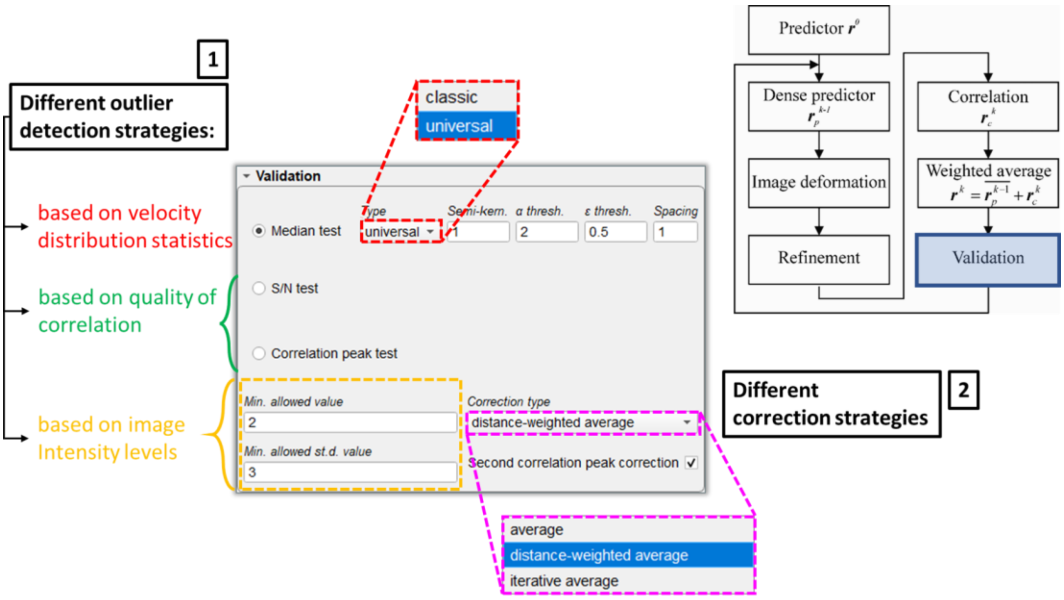



Figure 22 – Validation box.

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 23). 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 correlation map. The possible types are reported in Table 3;
3. 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;
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.

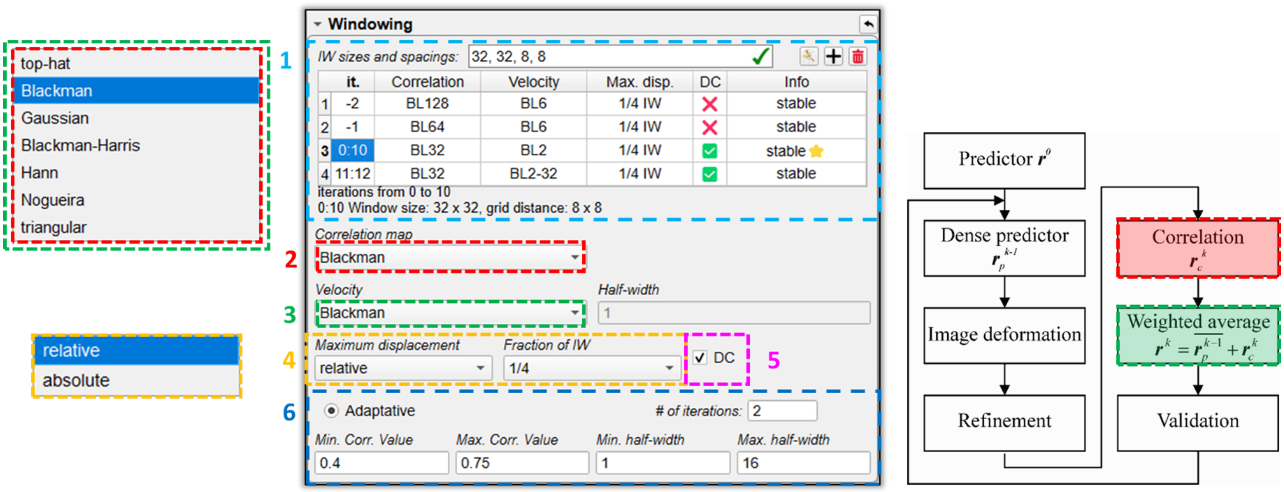



Figure 23 – Windowing box.

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 $\alpha$. | |
| **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.

# 13. Process (Disparity correction)

The disparity correction step is a procedure to estimate accurately the location of the laser sheet plane and correct the errors due to the misalignment between the laser sheet plane and the reference calibration plane and those associated with inaccurate estimation of the positions of the displacement vectors or the local viewing angles in the 3C reconstruction step. This procedure relies on the computation of the disparity vectors, which is performed via window-based cross-correlation of the dewarped images of the two cameras in the stereoscopic setup. The algorithm for the disparity correction in PaIRS follows the approaches explained in [8] and [6], with some improvements based on the more recent work of Wieneke (2018) [9]. A schematic flowchart of the method is illustrated in Figure 24 (left). Information about the laser thickness is also retrieved in this step and provided to the user.

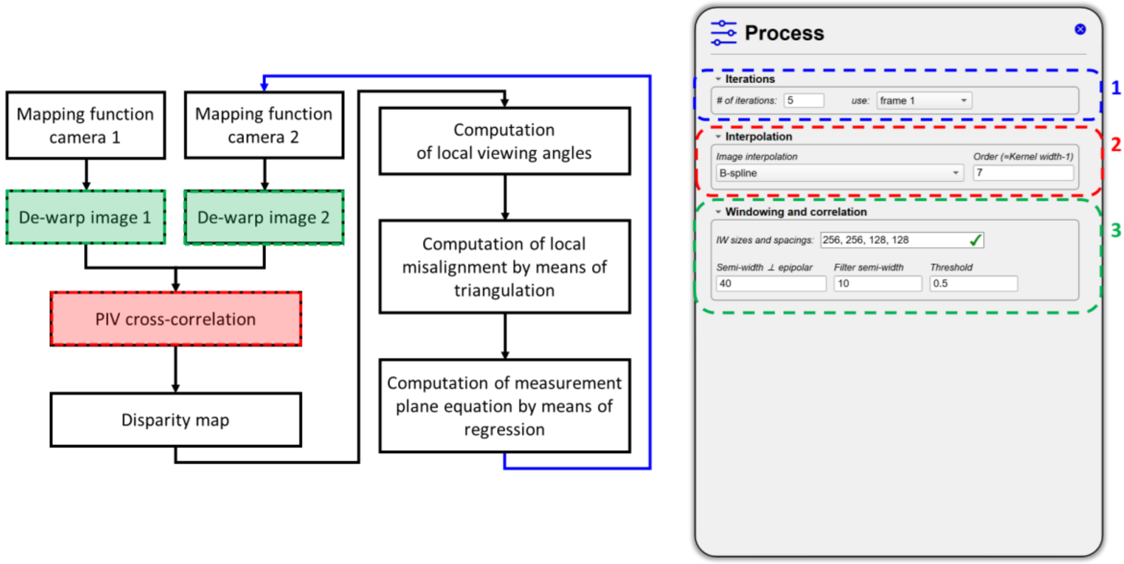



Figure 24 – Schematic flowchart of the disparity correction procedure (left) and related Process tab (right).

In the Process tab for the disparity correction step (Figure 24, right), the user must specify:

1. the number of iterations and if to use for the computation only frame 1, only frame 2 or both;
2. the type of interpolation to be for dewarping the images before the PIV cross-correlation step. The possible types are the same as those selectable in the image deformation step of the PIV and stereoscopic PIV analysis and are reported in Table 2 (page 28), where their variants or the corresponding parameters that must specified are also indicated;
3. the size and number of the interrogation windows and further parameters for computation of the cross-correlation peak related to the disparity vector. In the *IW sizes and spacings* the sizes and number of the interrogation windows are specified as a vector of four values corresponding to the height, width, vertical spacing and horizontal spacing of the same windows in pixel units. As for the correlation parameters, it is noticed that the map of the cross-correlation between the dewarped images of two cameras in a stereoscopic setup has a peak elongated in the direction of the local epipolar line[2]. For better accuracy, in the dewarping step of the procedure the interrogation windows are rotated to align with the epipolar line and the correlation map is computed only on a limited width perpendicular to this line. The latter is assigned into the *Semi-width ⊥ epipolar* spin. The correlation map is then filtered and the disparity vector is computed as the barycenter of the region of the correlation map above a certain threshold. The user can select the width of the filter for the detection of the disparity vector in the *Filter semi-width* spin and the threshold in the *Threshold* spin.

[2] In this context, the term "epipolar line" refers to the line of intersection between the laser plane and the epipolar plane (a plane passing through the optical centers of the cameras and the object point, which is the centroid of the interrogation window), which would more accurately be referred to as the epipolar line in the object plane.

# 14. Log

The Log tab is used to display useful information about the processes defined by the user, as shown 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**: in this section, different information is provided depending on the type of step considered.
    - historical minimum background image computation (: a list of the names of the correctly opened images is reported;
    - disparity correction: the following pieces of information are reported for each iteration:
        - laser plane equation;
        - residual calibration error in pixel units;
        - estimated thickness of the laser sheet in physical units (millimeters);
    - PIV and stereoscopic PIV analysis: the following pieces of information are reported for each image pair:
        - 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 (DC%).
- **Progress status**: this section reports information about the number of images or iterations processed (with and without errors), the number of the remaining ones not yet processed (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 or iterations processed without errors is also displayed in the progress bar placed in the Log tab header;

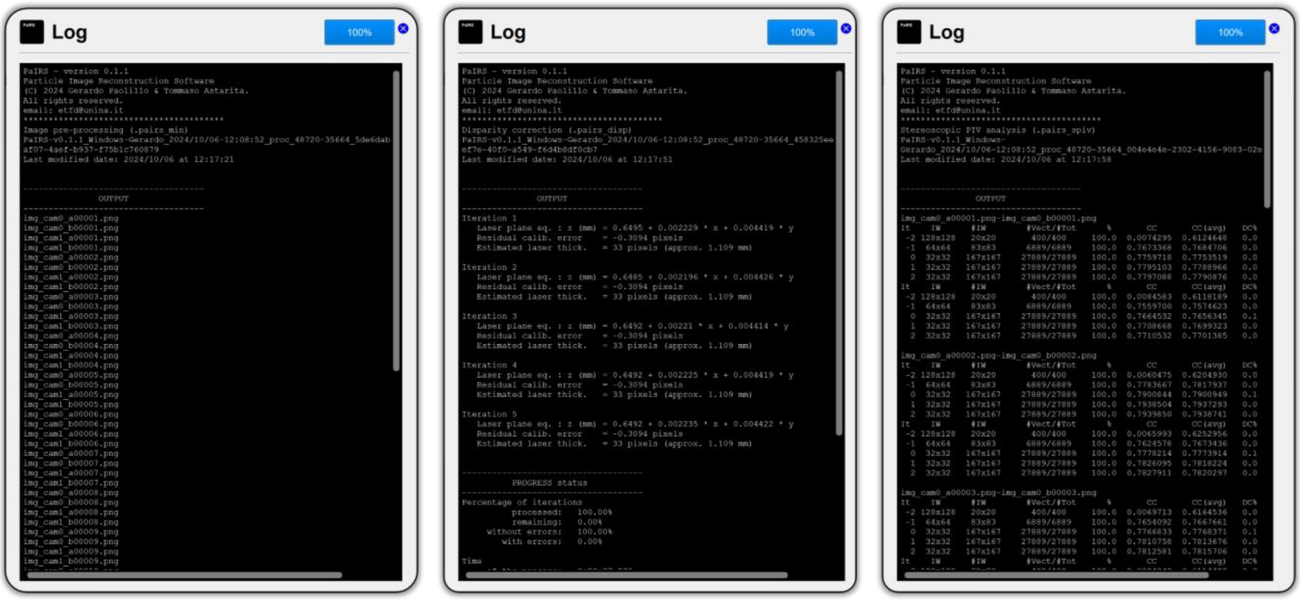



Figure 25 – Information displayed in the Log tab during a historical minimum background image computation (left), the disparity correction (center) and a stereoscopic PIV analysis (right).

- **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.

The information printed in the log is also saved in a text file located in the same output folder indicated in the Output tab. This log file, with a `.log` extension, shares the same filename root as the output files, as specified in the Output tab. The log file is written incrementally, thus it is not overwritten each time a process is executed in the same output folder by PaIRS. This allows the user to retain a complete history of all processes executed in that output folder within the log file.

In addition to the incremental log file, each time a step is executed, a separate log file with the extension `.pairs_`*`step`*`.log` (where *step* is a string corresponding to the type of step performed) is saved in the same output folder. This log file pertains to the most recent step executed in the current output folder. It is recommended not to delete the latter log file, as PaIRS checks its header to identify the last process executed in that folder and correctly associate the output files to the step items present in the process tree.

# 15. Vis

The Vis tab (Figure 26) allows to visualize the input images and available results (if any) of the current set

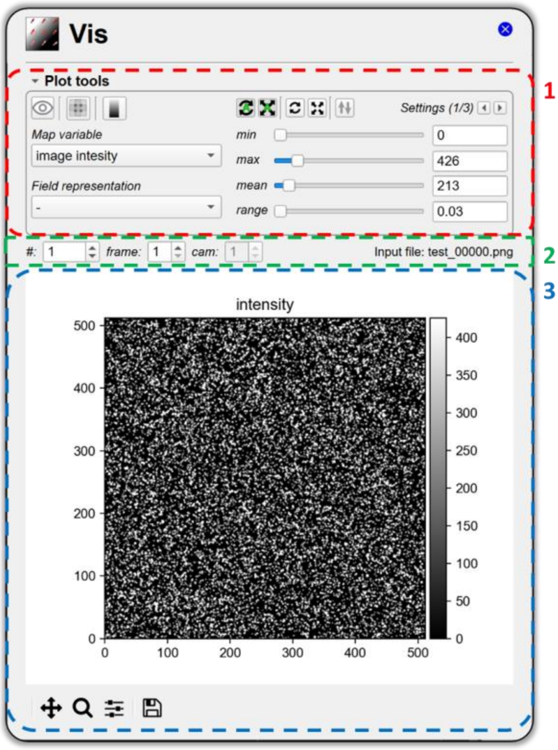



Figure 26 – The Vis tab.

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), the camera and the image frame or the iteration of the process 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 data available for the step selected in the Explorer.

### *Plot tools for input images*

Figure 27 shows the layout of the Plot tools box when only input images and potentially their historical minimum backgrounds are available. In such cases, the user can:

1. only select the item image intensity in the *Map variable* combo;

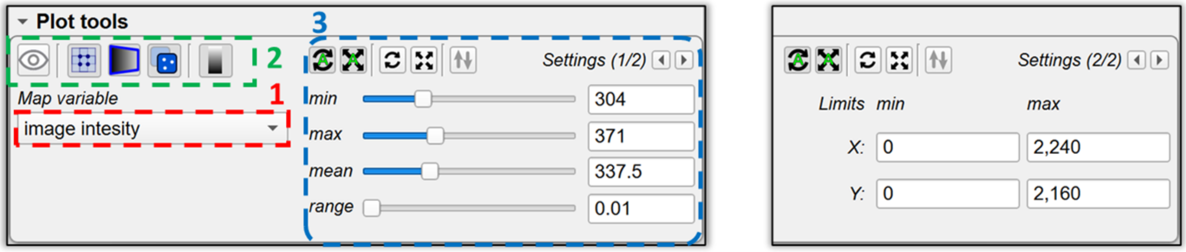
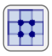


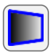

Figure 27 – Plot tools box layout for visualization of input images and historical minimum backgrounds.

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

If activated, this button allows the user to load the result files from the output folder specified in the Output tab even if they do not correspond to the current step (i.e., they were produced after the execution of a different process step). This functionality can help the user to inspect results from a past process also when the corresponding item is no longer present in the process tree.



Visualize the current grid scheme as shown in Figure 28. 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.



Visualize the projection of the physical region specified in the Output tab onto the camera image plane is represented and overlapped to the current image. This button is visible only for the disparity correction and stereoscopic PIV analysis steps.

Visualize the pre-processed images, i.e., the intensity distribution obtained by subtracting the minimum historical background from the images. This button is visible only if the image pre-processing step is activated and complete, in such a way that historical minimum background images have been computed. Alternatively, the user can activate the button .

Change the colormap for the current visualization. The user can choose from a wide range of colormaps available in matplotlib.

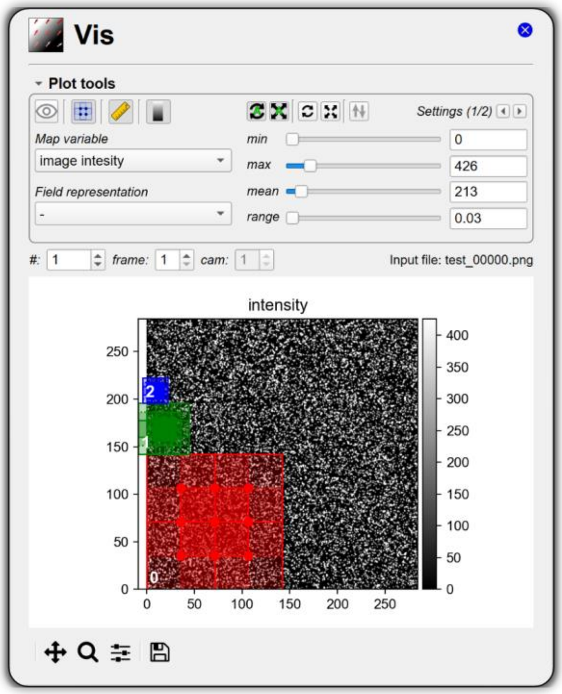



Figure 28 – Visualization of the grid scheme in the Vis tab.

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. Further buttons are available in this block:

 If activated, the levels are automatically adjusted when changing the image.

 If activated, the axis limits are automatically adjusted when changing the image.

 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.

***Plot tools for the output results***

Figure 29 shows the layout of the Plot tools box when results from a process step are available. In such cases, the user can:

1. choose the distribution of the variable to be visualized. In the case of PIV or stereoscopic PIV analysis steps, the result variables are:
   - the image intensity;
   - the velocity magnitude;
   - the velocity vector componets ($U$, $V$, $W$ are respectively the components along $x$, $y$, $z$);
   - the Reynolds stresses (‹$u'u'$›, ‹$v'v'$›, ‹$w'w'$›, ‹$u'v'$›, ‹$u'w'$›,‹$v'w'$›). The latter turbulent statistics are available only for the average velocity field;
   - the signal-to-noise (S/N) ratio;
   - 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 30, *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);
   - the correlation coefficient (*CC*).

   For the disparity correction step, the result variables are:
   - the image intensity;
   - the local $z$ displacement of the laser sheet plane from the reference plane $z = 0$;
   - the component of the disparity vector along the direction parallel to the epipolar line;
   - the residual component of the disparity vector along the direction perpendicular to the epipolar line.

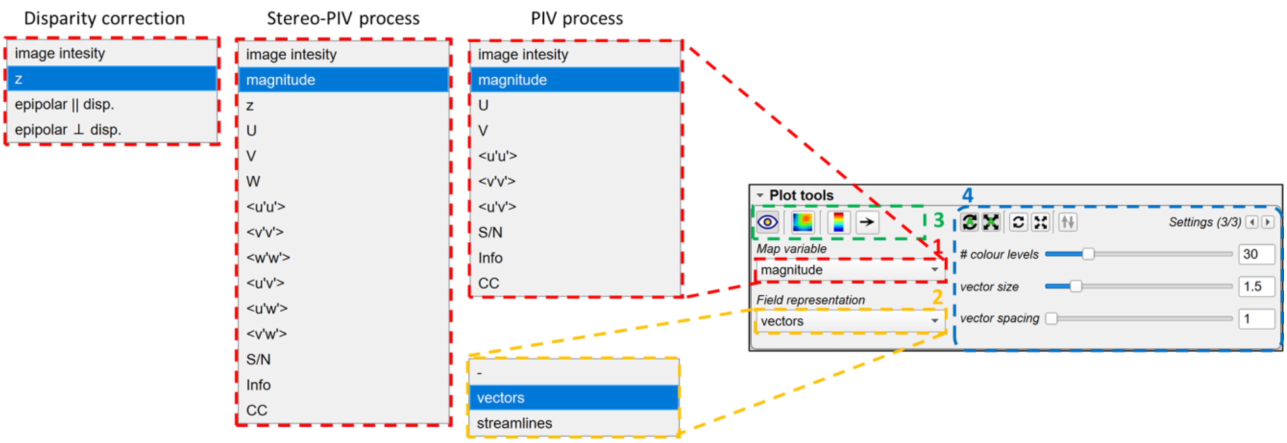



Figure 29 – Plot tools box layout when a PIV computation process is selected

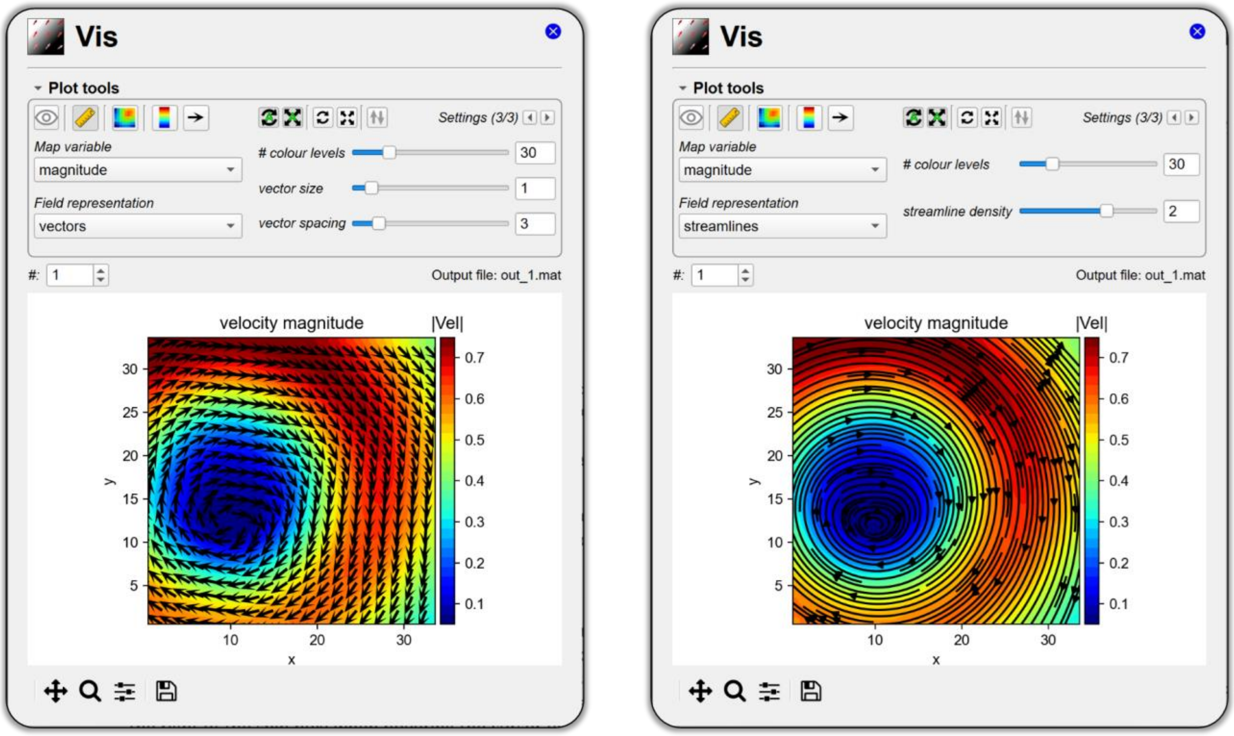



Figure 30 – Visualization of the velocity magnitude distribution with vectors (left) or streamlines (right) superimposed.

2. change the type of field representation by choosing among superimposition of vectors or streamlines (Figure 30);
3. change the type of visualization by clicking one of the following buttons:

 If activated, this button allows the user to load the result files from the output folder specified in the Output tab even if they do not correspond to the current step (i.e., they were produced after the execution of a different process step). This functionality can help the user to inspect results from a past process also when the corresponding item is no longer present in the process tree.

 Convert the $X$ and $Y$ coordinates from physical to pixel units. This button is visible only for the PIV analysis step when the image resolution specified in the Output tab is different from 1.0.

 Change the visualization of the variable map from an image show (pixelwise) at high pixel density to a filled-contour map. The filled-contour map generation is slow in matplotlib and thus not recommended for fast visualization of results.

 Change the colormap for the current visualization. The user can choose from a wide range of colormaps available in matplotlib.

 Change the color of the vectors or the streamlines. The user can choose among standard colors typically for such a kind of field representation.

4. 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.

## 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 and the cameras (if the number of the cameras is larger than one) via the corresponding spins. The name of the file currently visualized is reported in the text label on the right of the field box; when hovering the cursor over the text label, the tooltip shows the full path of the file currently visualized.

By setting the value to 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 process is running, the value 0 results in the visualization of the last result produced by PaIRS. In parallel processing for speed only the results related to one of the workers of the parallel pool are visualized when setting # to 0. For the disparity correction step, setting # to 0 it is possible to visualize the results for each iteration of the process, as shown in Figure 31. An apposite spin allows to user to switch between the iterations.

By setting the value to 1 in the field box, the user can view the imported images or results in Vis. In order to import images or results, the context menu of the plot area can be used. For further details, please refer to the following section.

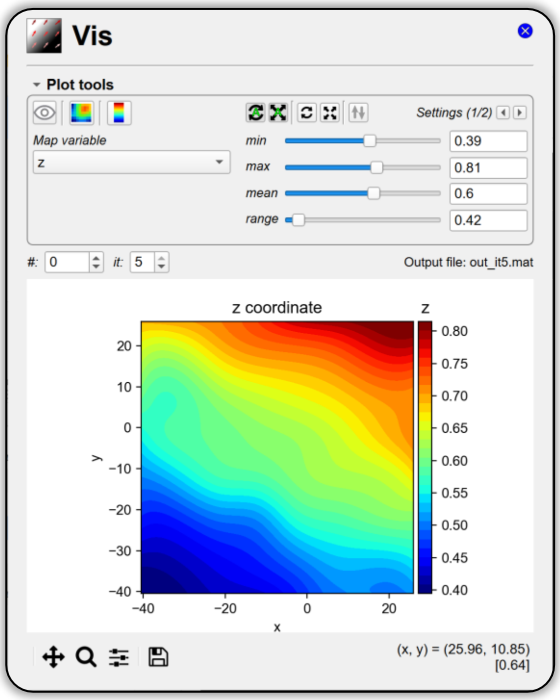



Figure 31 – Visualization of results for a disparity correction step.

## Plot area

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

By right clicking anywhere in the plot area, a context menu is opened, allowing the user to perform the following actions:

- (`Ctrl+C`) Copy the plot to the clipboard, allowing the user to quickly paste it elsewhere (e.g., into an external document).
- (`Ctrl+D`) Copy the plot into a new window, external to the main PaIRS window. This feature can be used to compare different images or velocity fields in a quick and intuitive manner.
- Load an image from disk: this function opens a dialog window that allows the user to browse the disk and select an image file to import.
- Load a results file from disk: this function opens a dialog window that allows the user to browse the disk and select a results file.

Multiple windows opened using the action in Vis can be easily managed through their own context menus. Specifically, the following actions can be performed from the figure window context menu:

- (`Ctrl+C`) Copy the plot to the clipboard, allowing the user to quickly paste it elsewhere (e.g., into an external document).
- (`Ctrl+Down`) Decrease the size of the selected figure window.
- (`Ctrl+Up`) Increase the size of the selected figure window.
- (`Ctrl+Return`) Apply the same scaling to all open figure windows.
- (`Ctrl+S`) Bring all figure windows to the foreground..
- (`Ctrl+A`) Align all figure windows in a Cartesian grid.
- (`Ctrl+X`) Close all figure windows.

## 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.

## Useful hint

Clicking on the Vis tab icon allows you to copy graphic settings from another step of the same type within the processes included in the current project. This functionality can facilitate the comparison of results between different processes.

# 16. Calibration

The Calibration tab (Figure 32) allows to specify the path of the camera calibration files.

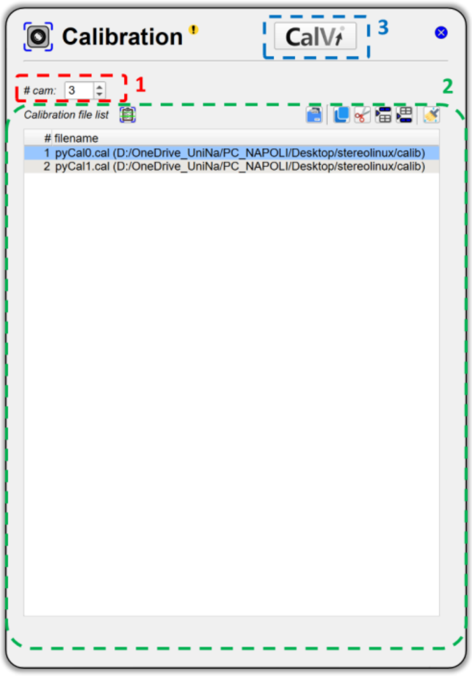



Figure 32 – The Calibration tab.

In the Calibration tab, the user can:

1. select the number of cameras (for a stereoscopic PIV process this is not possible and the number is fixed to 2);
2. import as many camera calibration files (with .cal extension) in the calibration file list tree. Basic modifications of the list can be performed by using the buttons above the tree. The corresponding actions can also be accessed through the context menu opening by right-clicking:
   -  (`F5`) perform an analysis of the calibration file list to identify any missing files on the disk.
   -  (`Ctrl+R`) read single or multiple calibration files from the disk and import to the list;
   -  (`Ctrl+C`) copy the selected items from the tree;
   -  (`Ctrl+X`) cut the selected items from the tree;
   -  (`Ctrl+V`) paste the copied or cut items below the selected row
   -  (`Ctrl+Shift+S`) paste the copied or cut items below the selected row
   -  (`Ctrl+O`) Clean the whole list.
3. access CalVi to perform camera calibration and produce new files to import to the list. When the  button is activated, the CalVi tabs appear in the tab area. Contextually, the button  appears in the tab button bar. This allows to start the camera calibration procedure once all the input/output and process parameters have been properly specified in the corresponding tabs.

# 17. Input/Output (Camera calibration)

The Input/Output tab of CalVi allows the user to specify the set of calibration 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](https://pillow.readthedocs.io/en/stable/handbook/image-file-formats.html).

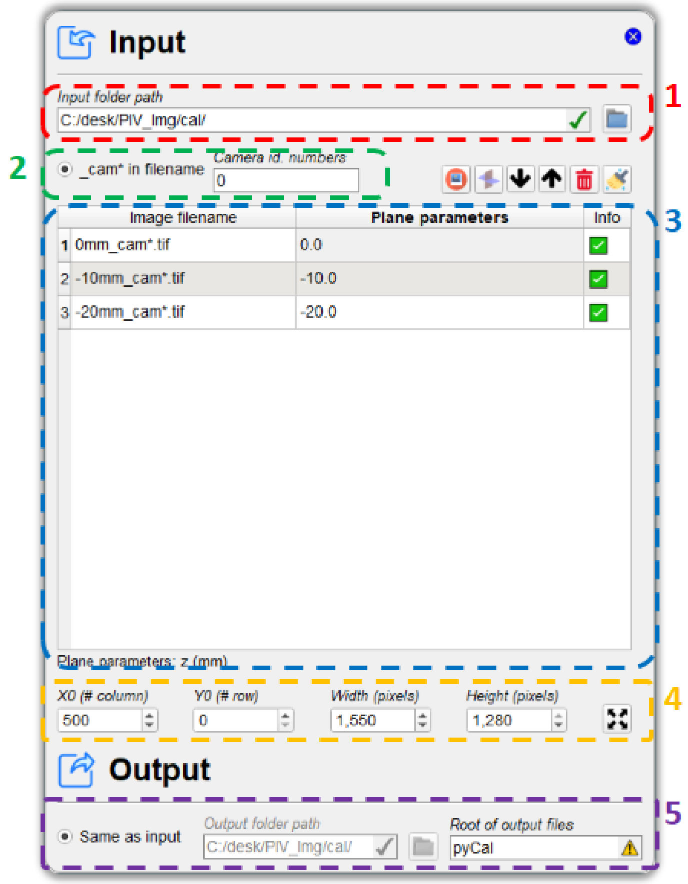



Figure 33 – The Input/Output tab.

In the Input/Output tab (Figure 33) the following parameters can be specified:

1. the path of the input folder containing the target image files: the image files related to the same calibration process must be contained in the same folder. The path of the input folder 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;
2. the identification numbers of the camera in the image filenames: in the current release sets of target images related to multiple camera bundles can be specified only by including the pattern `_cam*` in the filename and activating the radio button *_cam* in the filename*. In particular, the camera images corresponding to the same target position and orientation must have the same filename containing the pattern `_cam*` with the asterisk being replaced by the camera identification number. The camera identification number must be indicated in the dedicated line edit box. An example of valid set of target images for the calibration of a 4-cameras bundle is the following:

```
0mm_cam0.tif
-10mm_cam0.tif
-20mm_cam0.tif
0mm_cam1.tif
-10mm_cam1.tif
-20mm_cam1.tif
0mm_cam2.tif
-10mm_cam2.tif
-20mm_cam2.tif
…
```

(the files are not ordered in an alphabetical way)

3. the order and the geometric parameters of the target plane configurations: for each target position and orientation the plane parameters must be specified in the second column of the image file table. The number and type of parameters depend on the type of calibration to be performed; depending on the case, the must specify only the $z$ coordinate of the target plane position (standard calibration, in which the planar target is shifted along the normal direction) or the Euler angles and the origin translation related to the target plane position (calibration with unknown plane positions or calibration to estimate the equation of the plane, in which the planar target can be placed in an arbitrary way). The Euler angles and the origin translation related to the target plane position are simply an initial guess for the optimization of the mapping function parameters. The Euler angles $\beta, \alpha, \gamma$ are the rotations respectively around the $x$, $y$ and $z$-axes; the rotation matrix used is:

$$\begin{bmatrix} \cos\alpha\cos\gamma & \cos\alpha\sin\gamma & -\sin\alpha \\ \sin\beta\sin\alpha\cos\gamma - \cos\beta\sin\gamma & \sin\beta\sin\alpha\sin\gamma + \cos\beta\cos\gamma & \cos\alpha\sin\beta \\ \cos\beta\sin\alpha\cos\gamma + \sin\beta\sin\gamma & \cos\beta\sin\alpha\sin\gamma - \sin\beta\cos\gamma & \cos\alpha\cos\beta \end{bmatrix}$$

The image file table is provided with the following buttons:

(`Ctrl+D`) Add image files to the current calibration process.

(`Ctrl+F`) Import plane parameters from file and assign to the current target file.

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

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

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

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

4. the image area to be processed: 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;

5. the output file specifications: in the Output tab the user can specify the location of the output folder (this can be chosen to be the same as the input folder or different) and the root of the output filenames.

# 18. Process (Camera calibration)

In the Process tab (Figure 34) the user can setup the settings for the calibration process.

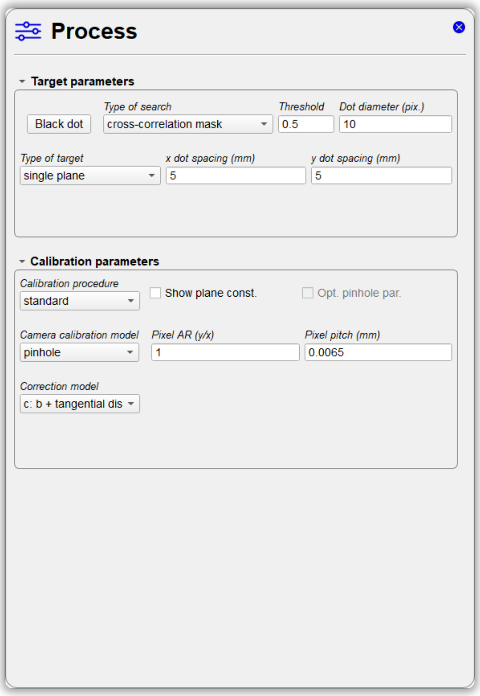



Figure 34 – The Process tab.

The Process tabs consists of two collapsible boxes in which the target and the calibration process parameters can be specified.

## Target parameters

It is worth noticing that the calibration performed in CalVi has been designed to work with calibration targets consisting of a plate with a regular Cartesian grid of circular dots. In this box, the parameters of the control dots and their grid can be specified and, more specifically, the user can choose (refer to Figure 35):

4. if the control points are black dots on a light background or white dots on a dark background. ;

5. the type of dot search: CalVi uses both a template matching algorithm and interpolation or centroid based methods for the detection of the markers in the images of the calibration target. The template (mask) can be generated in different ways:

   a. autocorrelation mask: the template is generated via autocorrelation of a portion of the image that can be interactively selected by the user (using the mouse central button). Before starting the calibration process, the autocorrelation mask is determined based on the whole target image. Such a method yields a subpixel accuracy in most situations and is the method suggested by the authors;

   b. function-based masks (top-hat with tight Gaussian-like tails, top-hat with broad Gaussian-like tails and Gaussian): these masks have been observed to offer better results in the case of large dot images with a moderately variable distribution of the intensity levels;

   The identification of the control points via the template-matching method is based on the computation of the local correlation map. Only the control points characterized by a peak in the correlation map greater than the value specified in the *Threshold* edit box are stored. The mask is square and its size is equal to 2.5 times the value indicated in *Dot diameter* edit box. This is

also the size of the computed local correlation map. The user can easily visualize the mask in the Vis tab with the appropriate image tools.
The dot search can be carried out also in two further ways:

c. interpolation: sub-pixel parabolic interpolation around the local maximum (white dot) or minimum (black dot). The local peak is identified considering only the intensity levels beyond the value specified in the *Threshold* edit box and in a square area of edge equal to 2.5 times the value indicated in *Dot diameter* edit box;

d. centroid: determination of the centroid of the distribution of the absolute image levels above the threshold specified in the *Threshold* edit box in the square area of edge equal to 2.5 times the value indicated in *Dot diameter* edit box.

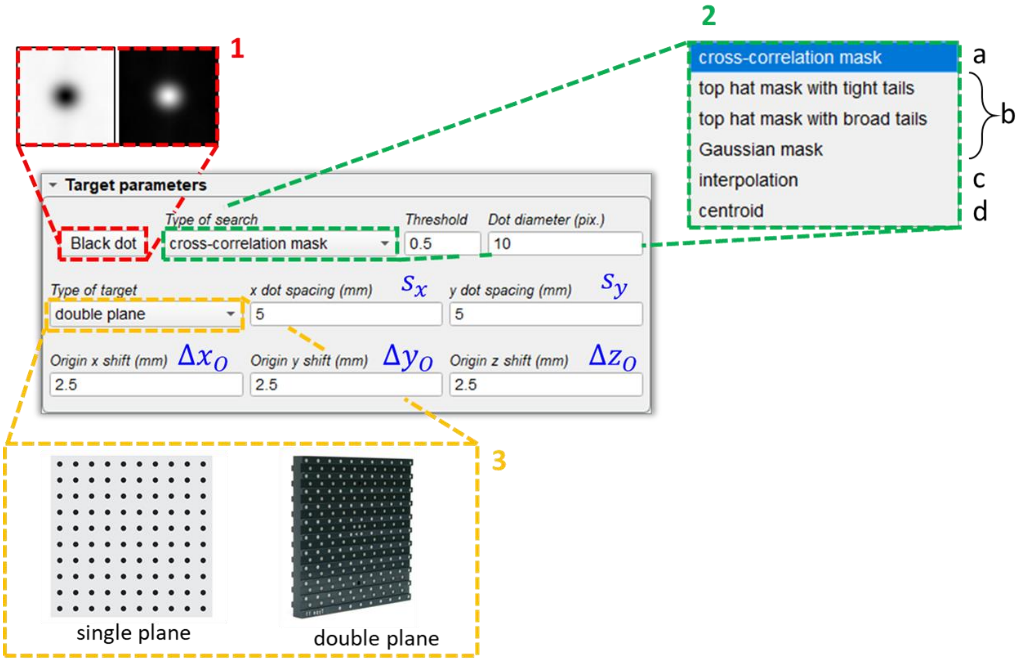



Figure 35 – Target parameters box.

6. the type of target: single plane or double plane. The dot spacings $s_x$ and $s_y$ along the $x$ and $y$-directions and the origin shifts $\Delta x_O$, $\Delta y_O$ and $\Delta z_O$ in the $x$, $y$ and $z$ directions must be specified in the corresponding edit boxes. Their definition is intuitive and clarified in Figure 36 for a double-plane target.

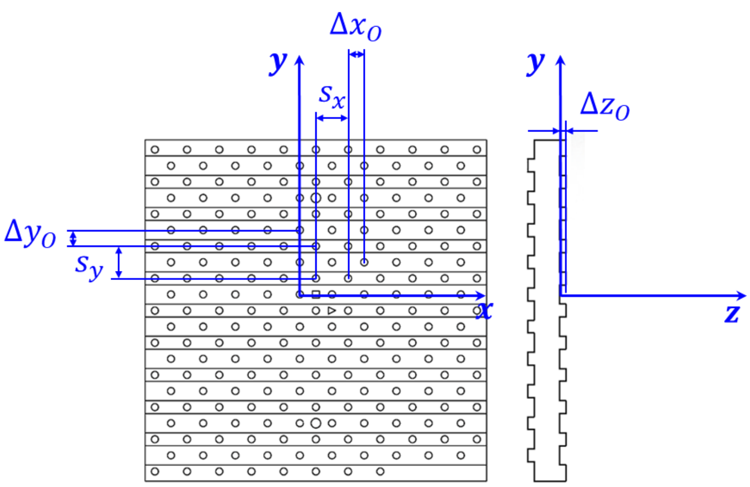



Figure 36 – Definition of dot spacings and origin shifts for a double-plane target.

## Calibration parameters

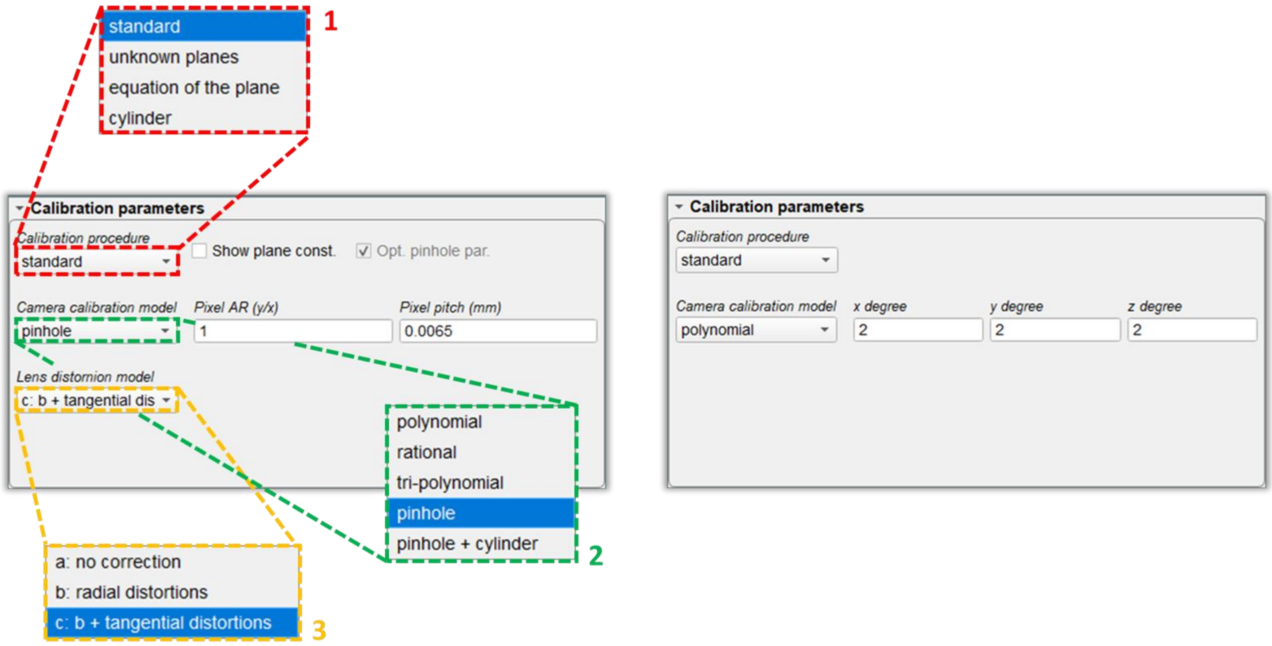



Figure 37 – Calibration parameters box.

In this box the user can choose (refer to Figure 37):

1. the type of calibration procedure: see Table 5 for more information in this regard;
2. the camera calibration model: CalVi supports accurate calibration with the camera models mostly used in the PIV community: polynomials, rational functions and the pinhole camera model. The reader is referred to reference [10] for more information about the definition of the polynomial and rational camera models (also denoted as analytical camera models). CalVi also supports the integration of the pinhole camera model with a refractive correction model for cylindrical geometries. The latter relies on a ray tracing procedure and correctly represents the refraction of the lines-of-sight (LOSs) across the cylindrical interfaces via Snell's law, as explained in [11]. The camera models are formulated in Table 6. In the case of the analytical camera models, the user is asked to specify the degree of the polynomial functions $N_x$, $N_y$; $N_z$, whereas in the case of the pinhole camera model the pixel size $d$ and the pixel aspect ratio $\chi$ must be specified;
3. the lens distortion corrections for the pinhole camera model:

   a: no correction: the pinhole reduces to a linear model (DLT, direct linear transformation);

   b: radial distortions: the correction terms for the radial distortions are included;

   c: b + tangential distortions: all the correction terms for both the radial and the tangential distortions are included.

   When the pinhole + cylinder camera model is selected, further correction models are available:

   d: c + cylinder origin: also the position of the cylinder is optimized in the calibration procedure;

   e: d + cylinder rotation: also the angular orientation of the cylinder is optimized in the calibration procedure;

   f: e + cylinder radius and thickness: also the internal and external radius of the cylinder are optimized in the calibration procedure;

   g: f + refractive index ratio: also the ratio of the refractive indexes of the cylinder wall and the surrounding fluid is optimized in the calibration procedure.

Table 5 - Type of calibration procedure.

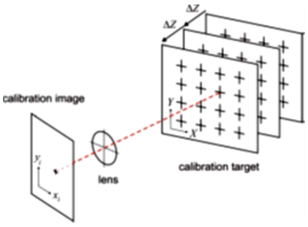
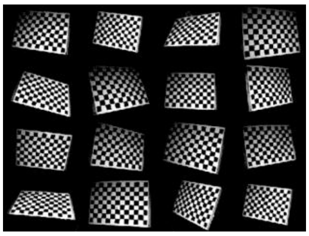
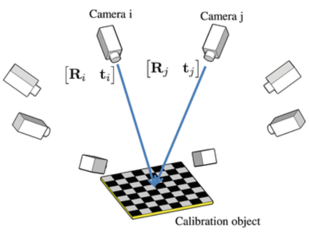
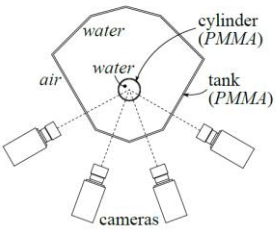
| Type | Description |
|---|---|
| **standard**<br><br> | The calibration target is translated along the $z$ direction (normal to the target plane) in several known positions, accurately measured by the user. The $z$ coordinate of each plane position must be specified in the *Plane parameters* column of the image file table included in the Input tab. CalVi can also determine a-posteriori all the six plane parameters (Euler angles and origin shifts) for each target position, both in case of a single camera and in case of a multiple camera bundle. To visualize such parameters the user must activate the *Show plane const.* check box. |
| **unknown planes (calibration per planes)**<br> | This type of calibration, also defined as "calibration per planes", does not need a perfect alignment of the target during the image acquisition. The user should perform the following steps during the image acquisition:<br>1. place the target in a reference position and acquire a first set of images with all the cameras. The reference position of the target, together with the choice of the origin dot in the image (during the calibration process performed in Vis), fixes the world reference frame;<br>2. rotate the target around the $x$ or $y$-axis of about 4-8 degrees and acquire a second set of images with all the cameras. It should not be necessary to be very precise, a translation shift of the target should not be important at this stage;<br>3. rotate in the opposite direction the target and acquire a third set of images with all the cameras;<br>4. assign another rotation or simply translate the target along the $z$-direction and acquire a fourth set of images with all the cameras;<br>5. you may acquire also other images of the target in different positions.<br>When selecting this type of calibration the generated calibration files contain also the constants of the planes. |
| **equation of the plane**<br><br> | In this case the goal of the calibration is to determine the plane constants of the target position starting from a previous camera calibration and from a set of images of the target in the considered unknown position. It is advised to perform this kind of calibration with at least 3 cameras. |
| **cylinder**<br><br> | This type of calibration procedure is used to optimize the cylinder parameters of the "pinhole + cylinder" camera model by using multiple cameras. It requires a pinhole camera calibration for each camera of the bundle as a starting point. The user can choose whether to optimize the pinhole camera parameters for each camera and/or the plane constants of the target positions recorded in the images. The plane constants and the cylinder parameters are common to all the cameras. Please see section 21 for more information about the procedure to correctly calibrate the "pinhole + cylinder" camera model. |

Table 6 – Type of camera model; see Paolillo and Astarita (IEEE 2020) [11] and (MST 2021) [9] for more details.

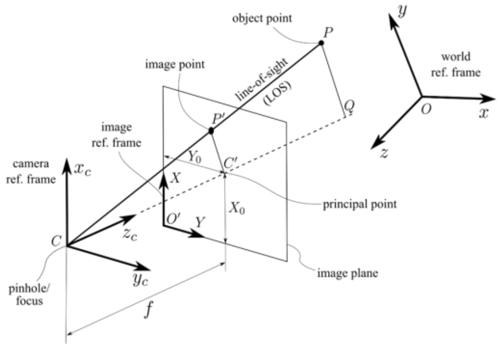
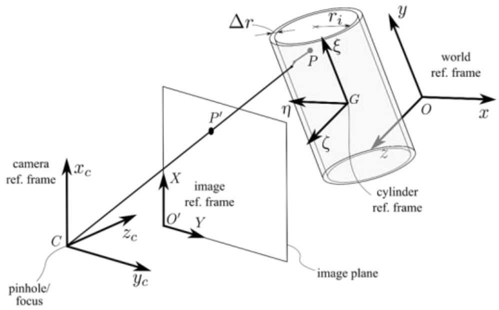
| Type | Formulation |
|---|---|
| **polynomial** | $X=\sum_{k=0}^{N_z}\sum_{j=0}^{N_y^*}\sum_{i=0}^{N_x^*} a_{ijk}x^i y^j z^k, \quad Y=\sum_{k=0}^{N_z}\sum_{j=0}^{N_y^*}\sum_{i=0}^{N_x^*} b_{ijk}x^i y^j z^k$<br>$N_y^*=\min(N_y,N_{max}-k);\ N_x^*=\min(N_x,N_{max}-k-j);\ N_{max}=\max(N_x,N_y,N_z).$ |
| **rational** | $X=\frac{\sum_{k=0}^{N_z}\sum_{j=0}^{N_y^*}\sum_{i=0}^{N_x^*} a_{ijk}x^i y^j z^k}{\sum_{k=0}^{N_z}\sum_{j=0}^{N_y^*}\sum_{i=0}^{N_x^*} c_{ijk}x^i y^j z^k}, \quad Y=\frac{\sum_{k=0}^{N_z}\sum_{j=0}^{N_y^*}\sum_{i=0}^{N_x^*} b_{ijk}x^i y^j z^k}{\sum_{k=0}^{N_z}\sum_{j=0}^{N_y^*}\sum_{i=0}^{N_x^*} c_{ijk}x^i y^j z^k}$<br>with $c_{000}=1$<br>$N_y^*=\min(N_y,N_{max}-k);\ N_x^*=\min(N_x,N_{max}-k-j);\ N_{max}=\max(N_x,N_y,N_z).$ |
| **tri-polynomial** | $X=\sum_{k=0}^{N_z}\sum_{j=0}^{N_y}\sum_{i=0}^{N_x} a_{ijk}x^i y^j z^k, \quad Y=\sum_{k=0}^{N_z}\sum_{j=0}^{N_y}\sum_{i=0}^{N_x} b_{ijk}x^i y^j z^k$ |
| **pinhole** | $c\underline{u}_i=FT_c\underline{x}$<br>$\underline{X}=U\left(B\underline{u}_i+\delta\underline{u}(\underline{u}_i)\right)$<br>$\underline{u}_i=[u_i,v_i,1]^T$ ideal coordinates in the camera reference frame<br>$\underline{x}=[x,y,z,1]^T$ coordinates in the world reference frame<br><br>$T_c=\begin{bmatrix}R_c & \underline{x}_{c0}\\ \underline{0}^T & 1\end{bmatrix};\ F=\begin{bmatrix}f&0&0&0\\0&f&0&0\\0&0&f&1\end{bmatrix};\ B=\begin{bmatrix}1+b_1&b_2&0\\b_2&1-b_1&0\\0&0&1\end{bmatrix};\ U=\begin{bmatrix}\frac{\chi}{d}&0&X_0\\0&\frac{1}{d}&Y_0\\0&0&1\end{bmatrix}$<br>$R_c=$ Euler rotation matrix, $\underline{x}_{c0}=$ coordinates of the origin in the camera reference frame, $f=$ effective pinhole focal length, $b_1,b_2=$ linear distortion coefficient, $d$ pixel size in the $Y$ direction, $\chi=$ pixel aspect ratio, $X_0,Y_0=$ image coordinates of the principal point $C'$<br>$\delta u(u,v)=u(k_1r^2+k_2r^4)+p_1(r^2+2u^2)+2p_2uv$<br>$\delta v(u,v)=v(k_1r^2+k_2r^4)+2p_1uv+p_2(r^2+2u^2)$<br>$r^2=u^2+v^2$, $k_1,k_2=$ coefficients for the radial distortion, $p_1,p_2=$ coefficients for the tangential distortion. |

Table 6 – Type of camera model; see Paolillo and Astarita (IEEE 2020) [11] and (MST 2021) [9] for more details.

| | |
|---|---|
| **pinhole + cylinder** | $\phi\left(\underline{\xi}_A;\ \underline{\xi}_C, r_i, \Delta r, \rho\right) = T_b\underline{x}$<br>$c\underline{u}_i = FT_cT_b'\underline{\xi}_A$<br>$\underline{X} = U\left(B\underline{u}_i + \delta\underline{u}(\underline{u}_i)\right)$<br>$\phi\left(\underline{\xi}_A;\ \underline{\xi}_C, \underline{g}, \rho\right)$ refractive distortion function<br>$\underline{\xi}_A = [\xi_A, \eta_A, \zeta_A, 1]^T$ coordinates of the dewarped point in the cylinder (body) reference frame<br><br><br>$\underline{u}_i = [u_i, v_i, 1]^T$ ideal coordinates in the camera reference frame<br>$\underline{x} = [x, y, z, 1]^T$ coordinates in the world reference frame<br>$\underline{\xi}_C = [\xi_C, \eta_C, \zeta_C, 1]^T$ coordinates of the focus $C$ in the cylinder reference frame<br>$\underline{\xi} = T_b\underline{x} = \begin{bmatrix} R_b & \underline{x}_{b0} \\ \underline{0}^T & 1 \end{bmatrix}\underline{x};\quad \underline{x} = T_b'\underline{\xi} = \begin{bmatrix} R_b^T & -R_b^T\underline{x}_{b0} \\ \underline{0}^T & 1 \end{bmatrix}\underline{\xi}$<br>$R_b$ = Euler rotation matrix from world to cylinder frame; $\underline{x}_{b0}$ = coordinates of the origin in the cylinder (body) reference frame; $r_i$ = cylinder internal radius; $\Delta r$ = cylinder wall thickness; $\rho = n_f/n_c$ = ratio of the refractive index of the fluid to that of the cylinder material (the fluid inside and outside is supposed to be the same).<br>$T_c = \begin{bmatrix} R_c & \underline{x}_{c0} \\ \underline{0}^T & 1 \end{bmatrix};\ F = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & f & 1 \end{bmatrix};\ B = \begin{bmatrix} 1+b_1 & b_2 & 0 \\ b_2 & 1-b_1 & 0 \\ 0 & 0 & 1 \end{bmatrix};\ U = \begin{bmatrix} \frac{\chi}{d} & 0 & X_0 \\ 0 & \frac{1}{d} & Y_0 \\ 0 & 0 & 1 \end{bmatrix}$<br>$R_c$ = Euler rotation matrix, $\underline{x}_{c0}$ = coordinates of the origin in the camera reference frame, $f$ = effective pinhole focal length, $b_1, b_2$ = linear distortion coefficient, $d$ pixel size in the $Y$ direction, $\chi$ = pixel aspect ratio, $X_0, Y_0$ = image coordinates of the principal point $C'$<br>$\delta u(u,v) = u(k_1r^2 + k_2r^4) + p_1(r^2 + 2u^2) + 2p_2uv$<br>$\delta v(u,v) = v(k_1r^2 + k_2r^4) + 2p_1uv + p_2(r^2 + 2u^2)$<br>$r^2 = u^2 + v^2$, $k_1, k_2$ = coefficients for the radial distortion, $p_1, p_2$ = coefficients for the tangential distortion. |

# 19. Vis (Camera calibration)

The Vis tab (Figure 38 – The Vis tab.Figure 38) is the core of the calibration process and allows the user to perform the interactive calibration once the *Run* button in the button bar has been clicked.

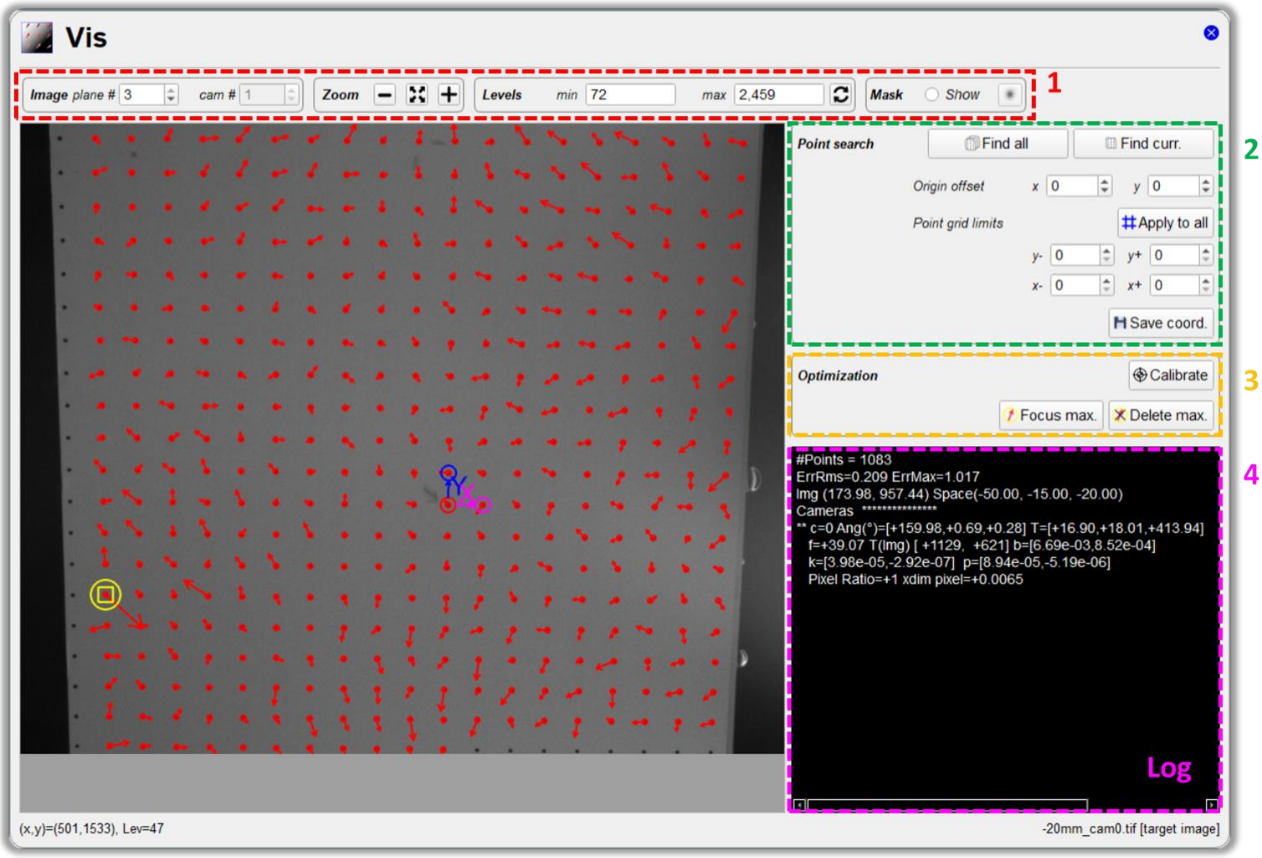



Figure 38 – The Vis tab.

It consists of four blocks:

4. the image plot tool, where the user can select the plane number and the camera number for the target image to be displayed, set the zoom and the level limits for the visualization and show/hide the mask;
5. the control point search box, which allows to find point coordinates in all the target images or only in the currently visualized image, assign an origin offset, limit the grid of dots and save the image coordinates to the disk;
6. the optimization box, where the user can start the optimization of the mapping function parameters and detect and remove the dots with the largest reprojection errors to refine the calibration parameters;
7. the log window, where instructions for the use of the interactive calibration tool, the output from the calibration process and potential errors are provided.

The following section explains how to use the interactive calibration tool.

# 20. CalVi: use of the interactive calibration tool

Once clicked the *Run* button in the button bar the user is driven through the interactive calibration process via the instructions displayed in the log window in the Vis tab. The calibration process consists of the following steps.

## Spot detection

This phase starts upon clicking the *Run* button in the button bar or the *Find all* button in the control point search box in the Vis tab. In this phase, the user must choose the spot that defines the origin of the world reference frame (central point), by simply using the left button of the mouse to select a spot. If a red circle is drawn around the spot, the image coordinates of the origin have been correctly found. Subsequently, with the same procedure, the user must select the point along the $x$ (purple square, defined as: *right point*) and $y$ (blue circle defined as: *upper point*) axis (see Figure 39).

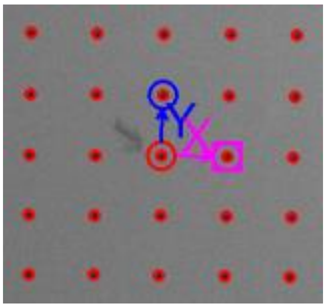


Figure 39 – Detection of the central point and the $x$ and $y$-axis.

CalVi immediately identifies the spots on the image and writes the number of points in the log window. If CalVi fails in finding one spot, a pop-up dialog appears where the user is asked whether to retry or to stop and exit (Figure 40). In this case the user may point the spot better. Normally the pop-up is shown in the left bottom part of the main window and, to speed up the procedure, in Windows and Linux the mouse pointer is moved automatically onto the most probable answer and back to the initial position once the button is clicked. This behavior initially is a little unsettling but in the long run reduces the time spent in clicking.

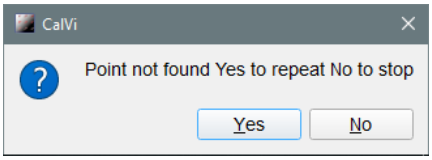



Figure 40 – Pop-up dialog upon CalVi failure in detecting one spot.

After selecting the three control points on the first image defining the origin and the $x$ and $y$-axis of the world reference frame, a pop-up dialog appears on the bottom right part of the screen (Figure 41). CalVi shows all the found spots with red circles and if the spots have been correctly identified the user should just press the *Yes* button (or the return button on the keyboard) to go on with the second image target. If the spots are not found correctly, the user can press the *No* button in order to restart the procedure.

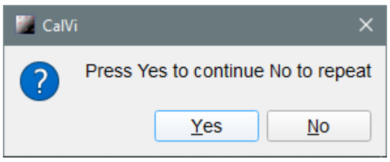



Figure 41 - Pop-up dialog after the detection of the first three control points.

After having correctly found the spots on the first image target, CalVi moves to the second image and asks to point the spot that defines the origin on the second plane, necessary to specify the $z$-axis of the world reference frame (in the case of a double plane target this step is performed automatically, in some cases this is not possible and the user need to repeat the procedure for all the planes). Again, if the origin has been correctly identified, the user just needs to press the *Yes* button in the pop-up window to go on. CalVi shows all the found spots with red circles and again the user can simply confirm and go on.

From the third image target on CalVi automatically identifies the origin and the user needs only to press the *Yes* button in the pop-up window. At the end of the whole spot detection stage, CalVi shows the total number of points found.

The user can start again the spot detection procedure if they desire to change the position of the origin by using the *Find all* button or repeat the detection for a single target image by using the *Find curr*. button in the control point search box.

## Calibration

Normally at this point, the user performs the calibration by simply pressing the *Calibration* button in the optimization box. If the user is only interested in the point position, they may just press the *Save coord.* button in the control point search box.

If the calibration takes too much time, the user can repress the same button (now called *Stop*) to terminate the process. At the end, arrows are shown to identify the reprojection error associated to each control point. It is possible to visualize and/or remove the point with largest reprojection error by pressing the corresponding buttons, i.e., the *Focus max* and *Delete max* buttons in the optimization box, or by using the context menu. The control point with maximum error is identified with a yellow square plus a yellow circle. If needed, the user can perform further iterations of the optimization by pressing again the *Calibration* button.

## Point removal

The control points can be removed in bulk by using the spin boxes within the *Point grid limits* section in the control point search box. By pressing the *Apply to all* button, CalVi uses the same limits for all the target images.

## Selection of the area for the auto-correlation mask

When using the cross-correlation mask type, CalVi initially chooses a relatively large central area of the image to build the correlation mask. The mask can be visualized via the mask box in the image plot tool. In some cases, the mask area can be biased and could result in wrong positioning of the spots or even in further difficulties in finding the spots. In these cases, the user can choose a new mask area by clicking on the image with the central mouse button and dragging till the final position. When a user-defined area for building the auto-correlation mask is used, a blue rectangle indicating such an area will be drawn on the target image. The mask area is normally different for each plane, but if it is chosen during the origin search of the first, the same mask area is applied to all the target images of the same camera. The user can choose a different mask area whenever needed.

## Origin not found or not in the image

If the origin is not found or is not in the image the automatic procedure fails and a pop-up dialog warns the user that the origin has not been found. In these cases, the user, after completing the initial procedure, should first select the plane using the *plane #* spin box in the image plot tool, then press the button *Find curr.* button in the control point search box and finally define a new reference frame in the image. If needed the selected virtual origin can be moved, spot by spot, to the correct position by using the spin boxes within the *Origin shift* section in the control point search box; a double green circle identifies the shifted origin while assigning these parameters. The origin can also be placed outside of the image. Moreover, the origin can be changed *a posteriori* in all the target images.

# 21. CalVi: calibration of pinhole + cylinder camera model

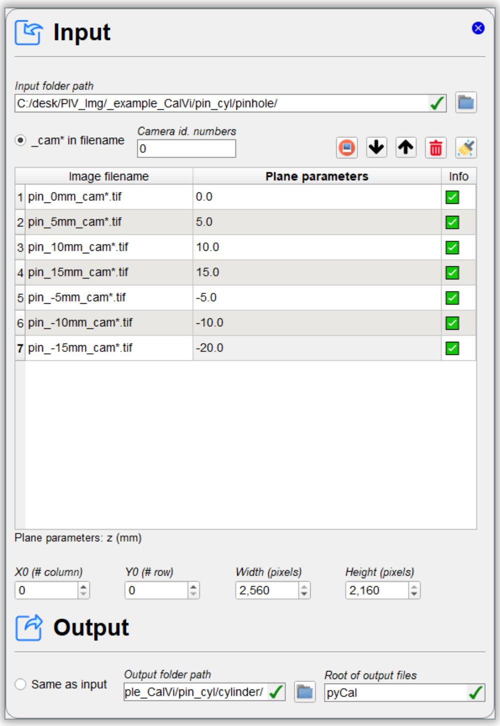
In the present section, three different strategies for the calibration of the pinhole + cylinder camera model are presented. This type of calibration is less robust than the classical ones and the nonlinear optimization of the calibration parameters may not converge easily, so the following are guidelines but, if needed, a more specific approach can be advised by the UniNa team. The recommended procedure (procedure 1) consists in two separate steps: a calibration of the pinhole camera model for each camera of the bundle based on a set of images of the target placed in the field of view in absence of the transparent cylindrical body; an estimation of the cylinder parameters based on a set of images of the target placed in the field of view in presence of the cylinder. Procedure 2 should be followed if it is not possible to record a set of images of the target in absence of the cylinder; in this case both the calibration of the pinhole camera model and the estimation of the cylinder parameters can be performed using the set of images of the target visualized through the cylinder. Procedure 3 is only recommended in the case of a single camera setup and it consists in simultaneous calibration of both the pinhole and the cylinder parameters based on a set of images of the target in presence of the cylinder.

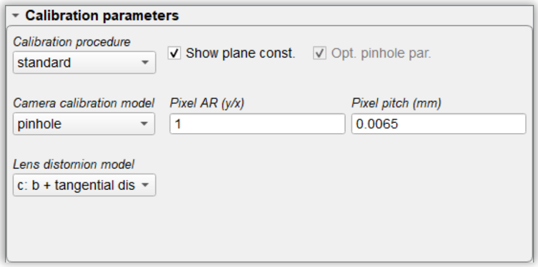
## Procedure 1 (recommended): separate pinhole camera model calibration and cylinder parameter estimation with different sets of target images

***Step 1: pinhole camera model calibration***

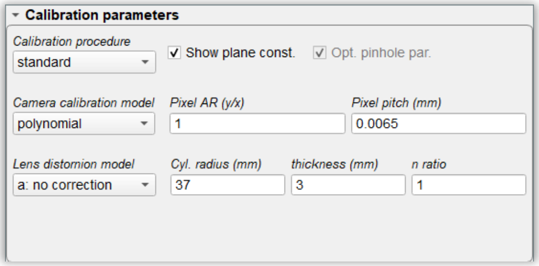
This step consists of a calibration of the pinhole camera model for each camera of the bundle. This can be performed either in a standard way (target moved in known positions) or with images of the target in unknown positions (calibration per planes). The images of the target must be recorded i**<u>n absence of the cylinder</u>** in the field of view and with the cameras arranged in their final configuration.

In case of standard calibration, the user must record a sufficient number of images of the target translated along the $z$ direction (normal to the target plane). It is suggested to put the image files for each camera in the same folder, say the *pinhole* folder, and to select a different output folder for the calibration results, say the *cylinder* folder. The calibration of the pinhole camera model must be performed separately for each camera; for fastness and simplicity, it is recommended to save the image files for each camera with names containing the pattern `_cam*` in such a way that it is possible to switch to the calibration of another camera by simply changing the *Camera id. Number* in the Input tab. In the Calibration parameters box of the Process tab, the standard calibration procedure and the pinhole camera model (with an appropriate lens distortion model) must be selected. Alternatively, the pinhole + cylinder camera model can be selected, but the *n ratio* value must be set to 1. The following figures show an example of how to set the parameters in the Input tab and the Calibration parameters box of the process tab for the present case. In the below example, the user will run the calibration for the camera 0 and then by simply changing the Camer id. Number they will move to camera 1, 2, and so on.

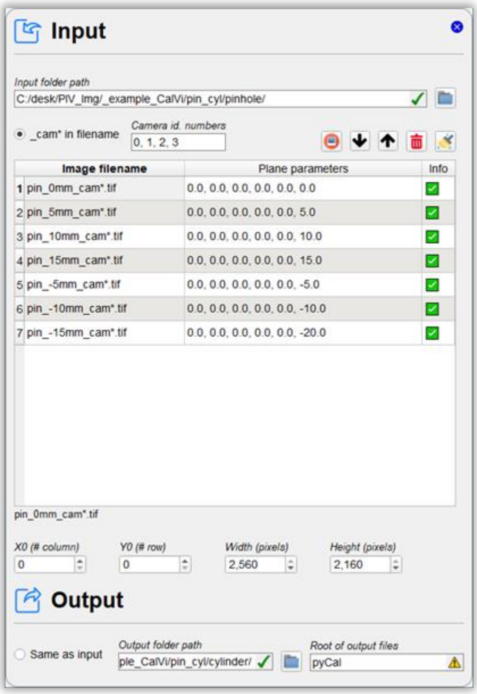


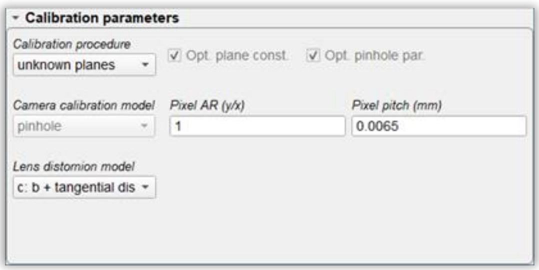




Alternatively…




In the case of calibration per planes, all the cameras can be calibrated in a single step. Thus, the differences with respect to the previous case are: in the Camera id. numbers in the Input tab all the camera numbers must be specified (please notice that in this case it is mandatory to save images of the target with the pattern _cam* in the filename); the initial guess values for the plane parameters must be appropriately assigned in the image file table contained in the Input tab; the calibration procedure in the Calibration parameters box of the Process tab must be set to "unknown planes". The following figures show an example of how to set the parameters in the Input tab and the Calibration parameters box of the process tab for the present case:




It is noticed that the calibration per plane is not recommended in the case of target positions consisting of planes all parallel to each other (as usually happens in the standard calibration).

***Step 2: estimation of the cylinder parameters***

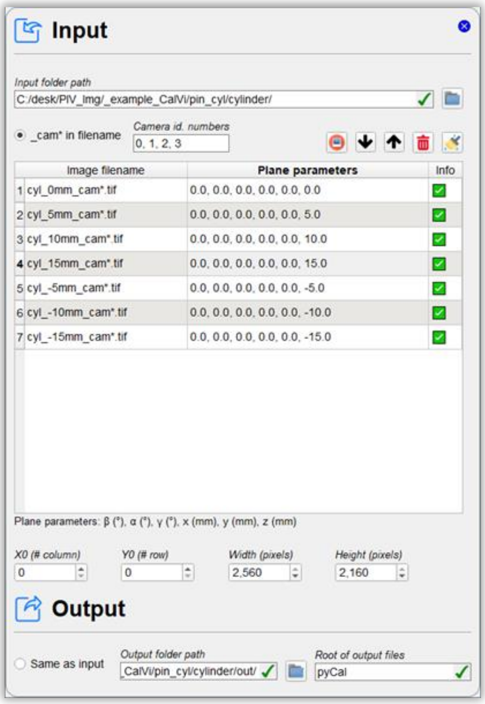
In this step the cylinder parameters are estimated using all the cameras available. The images of the target must be recorded **in presence of the cylinder** in the field of view and with the cameras arranged in their final configuration. In particular, the target can be placed either inside the cylinder or outside of it in the rear part. In fact, the pinhole + cylinder camera model is capable of mapping the whole domain surrounding the cylindrical body.

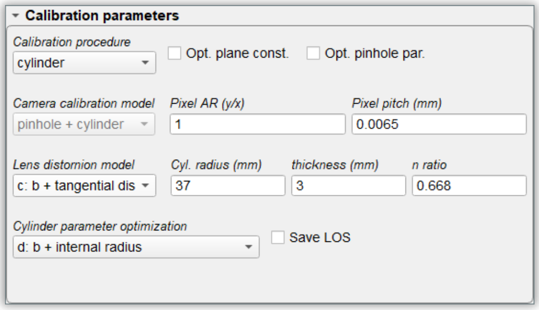
For this step, the image files for each camera must be saved in the folder containing the output calibration files from the previous step (the *cylinder* folder). It is suggested to save the output files from the present step to a different output folder, in such a way that the calibration output files from the previous step are not overwritten. In this case it is mandatory to save images of the target with the pattern `_cam*` in the filename and the known and initial guess values for the plane parameters must be appropriately assigned in the image file table contained in the Input tab. If the plane parameters are not optimized in the calibration procedure, then their values must be assigned accurately in the latter table. When using a single target image in this step, like for instance the image of the target placed behind the cylinder, if the position of the target is not changed while placing the cylinder in the field of view, the user is advised to perform an intermediate step between step 1 and step 2, consisting in the determination of the equation of the plane for the target, based on the previously calibrated pinhole camera model. The thus determined plane constants should be then imported in the Input tab using the related button .

In the Calibration parameters box of the Process tab, the cylinder calibration procedure must be selected. If the pinhole parameters are optimized, the user must select a lens distortion model for the pinhole part. As concerns the cylinder parameter optimization, the known or initial guess values for the internal cylinder radius, the cylinder wall thickness and the ratio of the refractive indexes of the fluid and the wall material must be specified. Finally, the user must choose a strategy for the optimization of the cylinder parameters among the following options (in case of difficulties in the nonlinear optimization the user may start with option a and only after try to optimize also the other parameters):

- a: cylinder origin and rotation: only the position and orientation of the cylinder are optimized, the geometrical and optical properties are not (their values are kept constat and equal to those assigned in the Process tab);
- b: a + cylinder thickness: the position and orientation and the thickness of the cylinder are optimized;
- c: b + refractive index (n) ratio: all the cylinder parameters expect for the internal radius are optimized;
- d: b + internal radius: all the cylinder parameters expect for the refractive index (n) ratio are optimized;
- e: a + internal radius and n ratio: all the cylinder parameters expect for the thickness are optimized;
- f: all cylinder parameters.

The following figures show an example of how to set the parameters in the Input tab and the Calibration parameters box of the process tab for the present case:




When using images of the target outside of the cylinder and in its rear part, a relatively small number of images (also a single image) may be sufficient by virtue of the large distortions caused by the twofold refraction of the lines-of-sight across the front and the rear part of the cylindrical wall. In case the target is placed inside the cylinder it is recommended to record a greater number of images of the target in different positions and try to put the control points as close as possible to the cylindrical wall.

## Procedure 2: separate pinhole camera model calibration and cylinder parameter estimation with the same set of target images

This procedure should be followed when it is not possible to record images of the target in absence of the cylinder, thus only images of the target visualized across the cylinder wall are available. It is again suggested to perform the calibration in two steps as in procedure 1. Some precautions should be adopted in the first step of the current procedure.

### *Step 1: pinhole camera model calibration*

In this step, the pinhole camera model for each camera of the bundle must be calibrated. This task is now performed using the images of the target in presence of the cylinder, neglecting the optical distortion due to the latter. For this reason, it is not advisable to perform a calibration per planes (poorly accurate results would be obtained by neglecting the cylindrical wall). So, the user must perform a standard calibration of the pinhole camera model for each camera. The output folder for the calibration results of this step should coincide with the input folder (activating *Same as input* checkbox in the Input tab), since the images to be used in step 2 are the same as those employed in this step.

In the calibration process, the user should try to limit the control points employed for the optimization to those falling in the region around the cylinder axis, that is to say farther from the cylindrical wall (in all directions). In fact, the optical distortions from the cylindrical wall of the lines-of-sight reaching these points are reduced if the angle between the optical axis of the camera and the cylinder axis is small.

To obtain a more accurate result, the user can calibrate the pinhole + cylinder camera model for each camera of the bundle following the below procedure 3 (applied for single camera) and then pass to step 2.

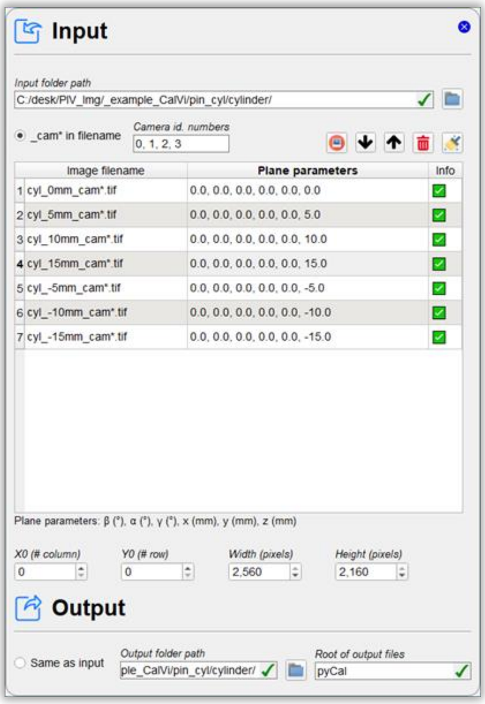
### *Step 2: estimation of the cylinder parameters*

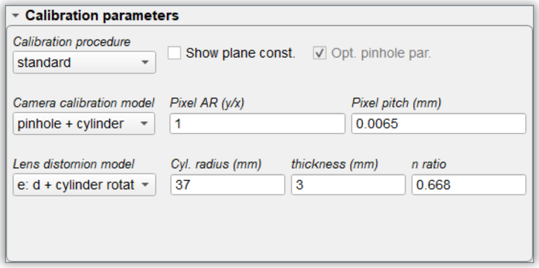
This step is identical to the step 2 of procedure 1, to which the reader is referred (see page 56).

## Procedure 3: simultaneous pinhole camera model calibration and cylinder parameter estimation

This procedure is based on consists the simultaneous calibration of both the pinhole and the cylinder parameters based on a set of images of the target in presence of the cylinder. This can be performed with a standard calibration procedure and selecting the pinhole + cylinder camera calibration model. The known or initial guess values for the internal cylinder radius, the cylinder wall thickness and the ratio of the refractive indexes of the fluid and the wall material must be specified. In this case, the user cannot choose a strategy for the optimization of the cylinder parameters among the options available for the cylinder calibration procedure as in step 2 of the above-described procedure 1 and 2. The cylinder parameters to be optimized can be selected from the *Lens distortion model* combo choosing among the options reported in bullet 3 at page 47 of the present guide. However, also in the case of a single camera bundle, this calibration procedure can be used as step 1 of procedure 2 and refined with a following ad-hoc estimation of the cylinder parameters (step 2 of procedure 2).

The following figures show an example of how to set the parameters in the Input tab and the Calibration parameters box of the process tab for the present case:



# 22. Use of the calibrated mapping functions

After calibrating the camera system with the aid of the CalVi interface, the PaIRS-PIV library can be imported in a Python environment to read the output calibration files and compute the image coordinates of the projections of physical points with assigned world coordinates.

The following code reports an example of such an use of the PaIRS-PIV library:

```
import numpy as np
from PaIRS_UniNa.PaIRS_PIV import MappingFunction
mapFun=MappingFunction()
calFileName=['calib\pyCal1.cal','calib\pyCal2.cal']
mapFun.readCal(calFileName) # readCal reads the calibration files
numPoints=4 # let's use 4 points
# selecting double (np.float64) or single (np.float32 faster) precision
type=np.float64
# selecting the point in the world reference system
points=np.array([[1,2,3]]*numPoints,dtype=type,order='C')+\
       np.array(np.arange(numPoints)*0.001,dtype=type)[:,None]
cam=-1 # selecting the camera (starting from 0);
# if cam is equal to -1 then the mapping function is evaluated for all the cameras
X=None  # Output vector should be already allocated or None
X1=mapFun.worldToImg(points,cam,None)
#in output X1 is equal to X if correctly allocated
print (f'id(X)={id(X)} id(X1)={id(X1)} same Id={id(X)==id(X1)}')
#print (X1)

type=np.float32
points=np.array([[1,2,3]]*numPoints,dtype=type,order='C')+\
       np.array(np.arange(numPoints)*0.001,dtype=type)[:,None]
cam=0
# a variable to be passed should be allocated as shown in the following
if cam == -1: X=np.zeros([mapFun.nCam, numPoints,2],dtype=type,order='C')
else: X=np.zeros([ numPoints,2],dtype=type,order='C')

X1=mapFun.worldToImg(points,1,X)
print (f'id(X)={id(X)} id(X1)={id(X1)} same Id={id(X)==id(X1)}')
#print (X1)

# of course you may use the output variable as an input for a successive call
X2=mapFun.worldToImg(points,1,X1)
print (f'id(X2)={id(X2)} id(X1)={id(X1)} same Id={id(X2)==id(X1)}')
#print (X1)

# if you need just one point simply pass to worldToImgPoint
# a list with the three coordinates
# in output you get a point in the Img reference system
# in this case cam cannot be -1
point=mapFun.worldToImgPoint([0,0,0],cam)
print(f'{point.x} {point.y}')

# example of mapping function inversion (from image to world coordinates)
# selecting type
```

```
for Type in [np.float64, np.float32]:
  print(f'Type={Type} ****************************************')
  for cam in [-1,0]:
    print(f'------------  cam={cam}  z ')

points=np.array([[1,2,3]]*numPoints,dtype=Type,order='C')+np.array(np.arange(n
umPoints)*0.001,dtype=Type)[:,None]
    X1=mapFun.worldToImg(points,cam,None)# In output X1 is equal to X if
correctly allocated
    z=np.array(points[:,2].T,order='C')
    x=mapFun.imgToWorld(X1,cam,z)

    print (x-points)
    xx=np.copy(points)
    #adding some noise
    maxNoise=1
    xx[:,0:2]+=np.random.default_rng().random([xx.shape[0],2])*maxNoise
    if cam ==-1:
      xx=np.tile(xx,[2,1,1])
    x=mapFun.imgToWorld(X1,cam,xx)
    print(f'cam={cam} Type={Type} x,y,z')
    print (x-points)
```

# 23. Appearance

It is possible to set the light or dark mode of the PaIRS interface by clicking on the button and selecting the favorite option, as shown in Figure 42.

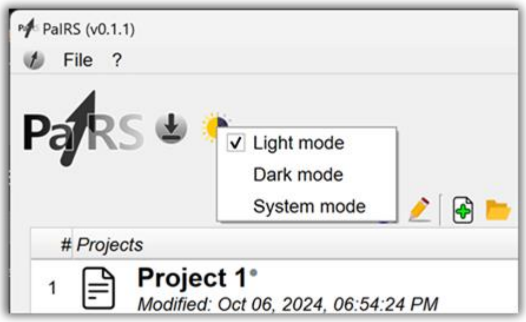



Figure 42 – Color mode context menu.

If you cannot wait for seeing the dark skin of PaIRS, give a look at Figure 43.

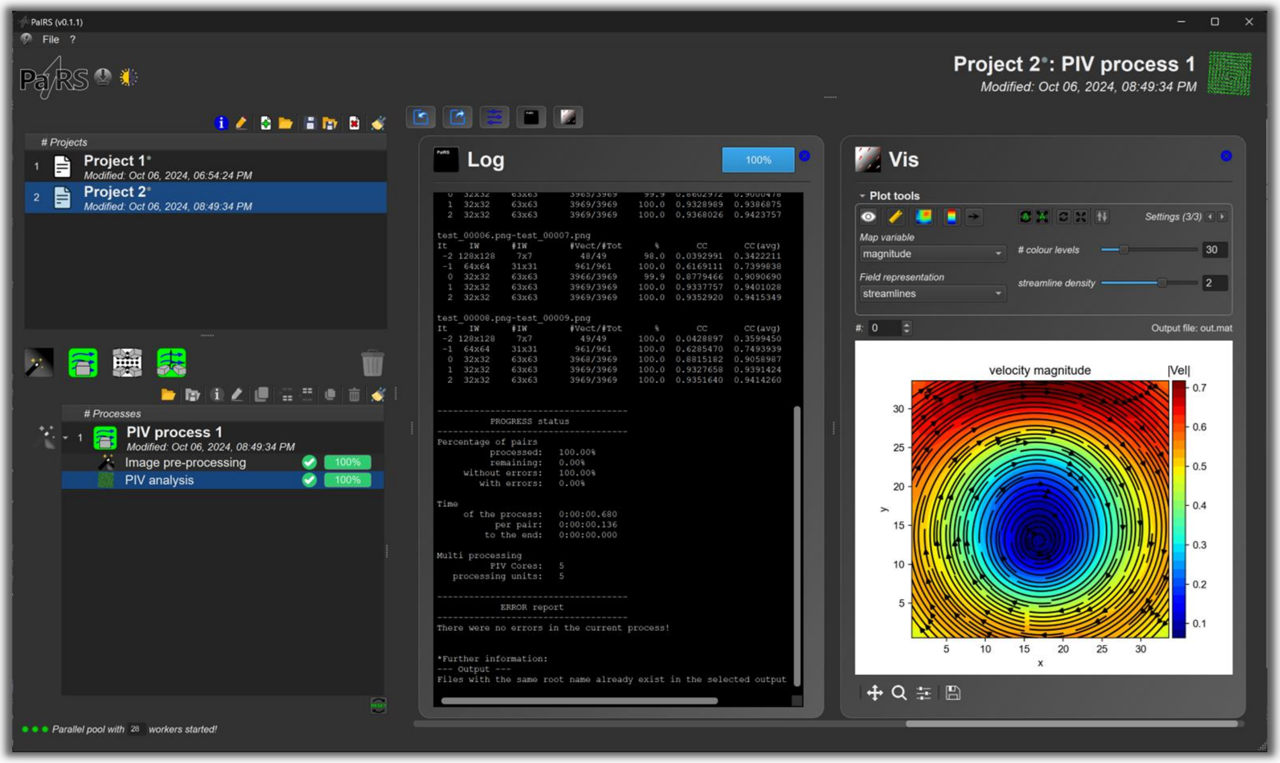



Figure 43 – PaIRS in the dark mode.

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] Giordano, R., & Astarita, T. (2009). "Spatial resolution of the Stereo PIV technique". *Experiments in Fluids*, 46(4), 643-658. doi: 10.1007/s00348-008-0589-y.

[7] 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.

[8] Wieneke, B. (2005). "Stereo-PIV using self-calibration on particle images". *Experiments in Fluids*, 39, 267-280. doi: 10.1007/s00348-005-0962-z.

[9] Wieneke, B. (2018). "Improvements for volume self-calibration". *Measurement Science and Technology*, 29(8), 084002. doi: 10.1088/1361-6501/aacd45.

[10] Paolillo, G., & Astarita, T. (2021). "On the PIV/PTV uncertainty related to calibration of camera systems with refractive surfaces". *Measurement Science and Technology*, 32(9), 094006. doi: 10.1088/1361-6501/abf3fc.

[11] Paolillo, G., & Astarita, T. (2020). "Perspective camera model with refraction correction for optical velocimetry measurements in complex geometries". *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 44(6), 3185-3196**.** doi: 10.1109/TPAMI.2020.3046467.

## 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.