Tutorial OpenPIV (ver. 0.02)

ver 01: ISTA - Institut für Strömungsmechanik und Technische Akustik
TU - Berlin, Germany
ver 02: Turbulence Structure Laboratory, Tel Aviv University, Israel

January 31, 2018

Contents

1 Introduction

OpenPIV is an open source Particle Image Velocimetry (PIV) analysis software written in Python, MATLAB and C++. The aim of this document is to deliver a fast overview of the Matlab version of *OpenPIV* only. For Python and C++ see in the respective repositories.

The next sections will refer to the main parameters, the output files and other features of *OpenPIV*. Some basics of the PIV technique will be mentioned as well. All the distributions of *OpenPIV* and other related material can be found at [1, 2].

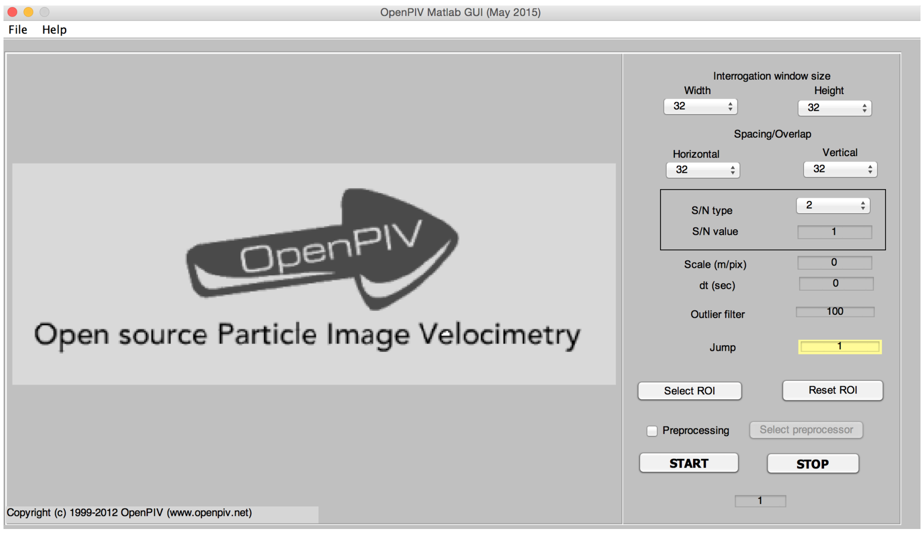
This tutorial runs *OpenPIV* on MATLAB, nevertheless similar set of parameters and output files applies for other distributions of *OpenPIV*. The PIV-Data used as example in this tutorial was extracted from [3].

2 Graphical User Interface GUI

The Matlab version of OpenPIV has a graphical user interface (GUI). The program starts from the command window using:

```
openpivgui
```

The following window is opened:

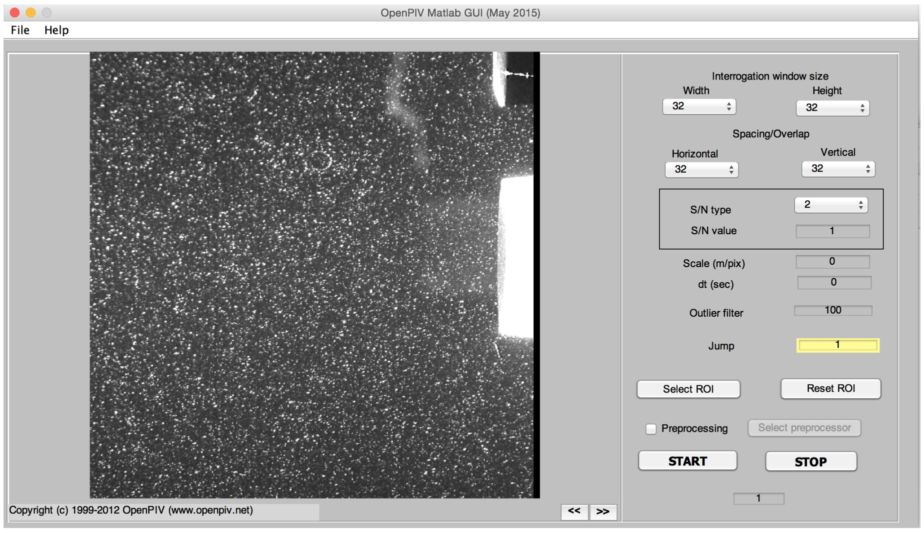

OpenPIV Matlab GUI (May 2015)
File
Help
OpenPIV
Open source Particle Image Velocimetry
Select ROI
Reset ROI
Select preprocessor
START
STOP
Copyright (c) 1999-2012 OpenPIV (www.openpiv.net)

Interrogation window size
Width 32
Height 32
Spacing/Overlap
Horizontal 32
Vertical 32
S/N type 2
S/N value 1
Scale (m/pix) 0
dt (sec) 0
Outlier filter 100
Jump 1
☐ Preprocessing
1

Figure 1: First window of the OpenPIV

Figure 3 depicts the Graphical User Interface (GUI) of *OpenPIV* after loading some example data.

The first step before to run the PIV-Analysis is to load the PIV-Data. On the top left the menu bar is located. Under the drop-down menu “File" the option “Load" is available, either this option or its shortcut Ctrl+L opens a window where to select the data to be loaded. Once the PIV-Images are loaded the buttons ≪ and ≫ at the bottom jump between the loaded images, the indicator at the right-bottom displays the number of the currently shown image.


OpenPIV Matlab GUI (May 2015)

File Help

Interrogation window size: Width 32, Height 32

Spacing/Overlap: Horizontal 32, Vertical 32

S/N type 2

S/N value 1

Scale (m/pix) 0

dt (sec) 0

Outlier filter 100

Jump 1

Copyright (c) 1999-2012 OpenPIV (www.openpiv.net)

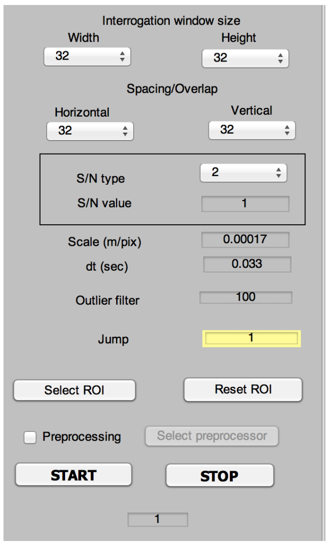
Figure 2: OpenPIV Matlab GUI

3 Load Data

Note that at least two PIV-Images are needed for a PIV-Analysis.

4 Analysis parameters

In this section the parameters on the right side of the GUI are described.


Interrogation window size: Width 32, Height 32

Spacing/Overlap: Horizontal 32, Vertical 32

S/N type 2

S/N value 1

Scale (m/pix) 0.00017

dt (sec) 0.033

Outlier filter 100

Jump 1

Figure 3: OpenPIV Matlab GUI

4.1 Interrogation window size

The PIV-Image will be divided in interrogation windows. The width and the height control the size in pixels of the interrogation windows. Since the analysis is based on statistical a large interrogation windows are more robust against background noise and outlier vectors.

4.2 Spacing/Overlap

The horizontal and vertical parameters control the size of the correlation windows and set the resolution of the two components of the velocity field u_x u_y. In figure 4 is depicted an example.

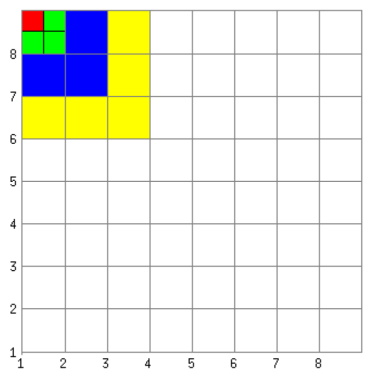


Figure 4: Image of 256x256 pixels divided by 32x32 interrogation window. Space/Overlap: red 16x16, green 32x32, blue 64x64 and yellow 128x128

4.3 Signal-to-Noise ratio and filters

In *OpenPIV* are implemented two types of Signal-to-Noise ratio (S/N): **peak to the mean** and **peak to the second peak**. In the drop-down menu 1 corresponds to peak to the mean and 2 to peak to the second peak, option 1 is faster.

The parameter S/N value is the threshold between the correct/wrong vectors according to the S/N type. There are no default values, a try-out has to be perform in order to find out the best value.

OpenPIV applies global and local filtering. By global filtering the obvious outlier vectors will be removed, i.e. the vectors which length is **larger than the mean of the flow field plus 3 times its standard deviation**. These are global outliers in the statistical sense.

In order to find local vector outliers, i.e. vectors that are dissimilar from the close neighbors, a local filtering is performed on small neighborhoods of vectors, 3×3. Typically there are about 5 % of erroneous vectors, these are removed and later the missing values are interpolated from the neighbor vector values.

4.4 Scale

The scale value is a factor that will multiply the resulting vectors and the distances between the vectors. It helps with the visualization of the vectors but it has to be taken into account when the units are transform to physical units. if the value is null, the scaling is not used. The results will be in pixels.

4.5 Time interval

The time interval is a factor that will convert the displacement (in pixels or meters, if scale is applied) to the speed units (m/sec). the resulting vectors and the distances between the vectors. It helps with the visualization of the vectors but it has to be taken into account when the units are transform to physical units. If the value is null, then dt = 1 and the speed will remain as displacement, e.g. pix/dt

4.6 Outlier filter

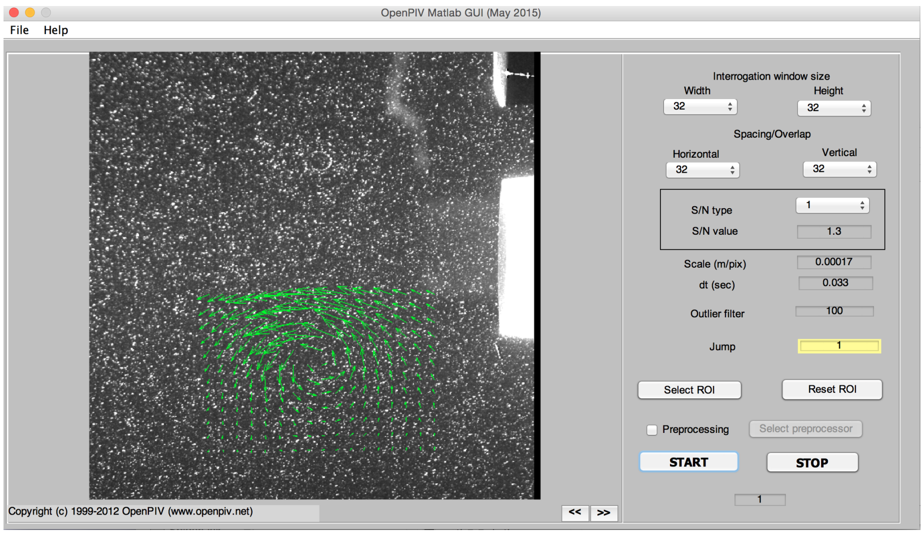
The outlier filter value is the threshold of the global outlier filter and indicates **how many times the standard deviation of the whole vector field** is exceeded before the vector is considered as outlier.

4.7 Jump

The jump value indicates how many PIV-Images are evaluated in the analysis. The count starts at the next image from the current one, i.e. if jump is equal to *n* then *OpenPIV* takes the current image plus the next *n* PIV-Images for one analysis. Note that at least two PIV-Images are needed for the analysis, what makes jump values less than one pointless.

4.8 Select and reset Region Of Interest(ROI)

The option Select ROI provides the possibility of evaluate only part of the whole PIV-Image. The option Reset ROI release the selected region. In figure 5 one example is shown.


OpenPIV Matlab GUI (May 2015)

Interrogation window size
Width: 32
Height: 32

Spacing/Overlap
Horizontal: 32
Vertical: 32

S/N type: 1
S/N value: 1.3

Scale (m/pix): 0.00017
dt (sec): 0.033
Outlier filter: 100
Jump: 1

Copyright (c) 1999-2012 OpenPIV (www.openpiv.net)

Figure 5: Results for a selected region

4.9 Pre-processing

The pre-processing option applies transformation-image function to the PIV-Images. The current MATLAB version uses the function *adapthisteq* which enhances the contrast of images. For more details see MATLAB help. The user can develop any image processing algorithm to run on each of the images before the FFT-based cross-correlation algorithm is applied to get PIV data.

5 Start and stop

The start and stop buttons either start or stop the analyze process.

6 Output files

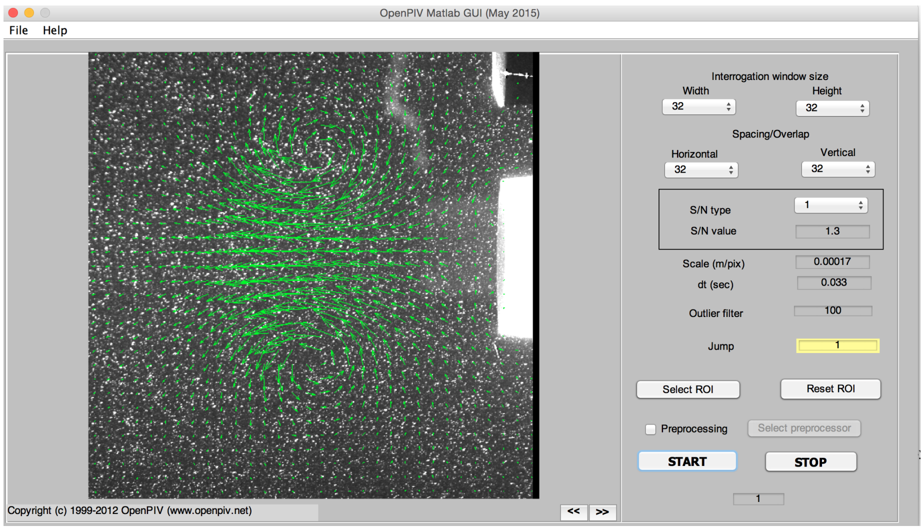
The first vectors results are depicted on the PIV-Images in the GUI, as figure 5 and 6 illustrate. In the folder containing the PIV-Data *OpenPIV* saves three lists of files: no filtered results *dataName_noflt.vec*, filtered results after global and local filters *dataName_flt.vec* and final results after filtering and interpolation *dataName.vec*.

The output file has a **header** that defines the variables and the units of the **5 columns** of data, i.e. x y u_x u_y S/N, and the size of the PIV field.

```
VARIABLES= "X m", "Y m", "U m/s", "V m/s", "CHC", ZONE I=32, J=32
```

Possible units are ‘pix’ or ‘m’ and the velocity can be ‘pix/dt’ or ‘m/s’ (or ‘pix/s’). The first two columns correspond to the coordinates, the third and fourth to the two velocity vector components and the fifth column is the S/N ration. Note that the value is different if the user chose peak to second peak or peak to mean ratio as S/N type. The value of peak to second peak or peak to mean ratio is stored for further processing.

The number of output files will be the number of loaded PIV-Images minus the selected jump value.


OpenPIV Matlab GUI (May 2015)

File Help

Interrogation window size

Width 32

Height 32

Spacing/Overlap

Horizontal 32

Vertical 32

S/N type 1

S/N value 1.3

Scale (m/pix) 0.00017

dt (sec) 0.033

Outlier filter 100

Jump 1

Copyright (c) 1999-2012 OpenPIV (www.openpiv.net)

Figure 6: Output results

Appendix: Units convert

Sometimes it makes more sense to leave the scale and time interval as default and transform the units of the results manually. In this case the velocity vector units of u_x and u_y are stored as displacements. *OpenPIV* results are delivered in $[scale \times (pixels/\Delta t)]$, with *scale* as the selected scale value and Δt as the time interval between two images. In order to convert the units to $[m/s]$ is needed:

- the vector components u_x, u_y in $[scale \times (pixels/\Delta t)]$
- the Δt value in s
- the dimensionless scale value
- the relation between *meters* and *pixels* in that image $[m/pixels]$, i.e. the image size and the number of pixels.

The following example demonstrates a particular case of unit conversion.

- $u_x = 5$, $u_y = 0$ in $[scale \times (pixels/\Delta t)]$
- the $\Delta t = 0.5$ in s
- scale value $= 2$
- image size 0.25 x 0.25 m and 256 x 256 *pixels*, i.e. the relation between *meters* and *pixels* is 0.25/256 $[m/pixels]$.
- the conversion of u_x from $[scale \times (pixels/\Delta t)]$ to $[m/s]$:

$$\frac{u_x}{scale \times \Delta t} \times rel. \frac{m}{pixels} = \frac{5}{2 \times 0.5} \times \frac{0.25}{256} = 0.0049 m/s$$

References

[1] R. Gurka A. Liberzon and Z. Taylor. OpenPIV Home Page. `http://www.openpiv.net`, 2009.

[2] G.A. Kopp A. Liberzon Z.J. Taylor, R. Gurka. Long-Duration Time-Resolved PIV to Study Unsteady Aerodynamics. *Instrumentation and Measurement, IEEE Transactions on*, 59(12):3262–3269, Dec. 2010. `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5464317&isnumber=5609237`.

[3] J. Carlier. Second set of fluid mechanics image sequences. European Project 'Fluid image analisys and description' (FLUID) – http://www.fluid.irisa.fr/, 2005.