PART2TRACK
double frame & time resolved
2D PTV

PART2TRACK

Quickstart Guide

October 15, 2019

1 General

This document accompanies the 2D-Particle Tracking Velocimetry code Part2Track developed at the Institute of Mechanics and Fluid Mechanics at the TU Bergakademie Freiberg. It shall give the reader a short introduction in how to run the program and it also provides a brief description of all important processing parameters.

2 Download & Dependencies

The program is stored at github. To directly access the project follow: `https://LINK`
From there, the source code can either be downloaded manually or copied via git. The source files consist of the main files `main_proc.m` and `main_post.m` and the three folders `doc` (documentation), `src` (source files of the program) and `test_cases` (images and parameters of provided test cases). As the program is written within the MATLAB framwork, a working license of MATLAB is needed. The earliest version tested with the software is 2017a. Earlier versions might work but could not be tested. In addition, the following toolboxes are utilized and therefore necessary to run the program:

- Image Processing Toolbox
- Statistics and Machine Learning Toolbox
- Curve Fitting Toolbox
- Signal Processing Toolbox
- Parallel Computing Toolbox

3 Test Cases

Two different test cases are provided alongside the source code. Experimental images of the flow through a model of the human lung are given for the evaluation with the double frame mode. In order to process time resolved data, images of a flow within a bend pipe can be used. The test cases can be run by changing the path `dir_eval` within `main_proc.m` to one of the following lines:

- `'..\test_cases\double_frame\Lung\'`
- `'..\test_cases\time_resolved\Pipe\'`

Working parameters are defined in the parameter file for each test case but the user is free to experiment with different values of the input parameters and their impact on the results. Recommendations on setting suitable values for these parameters are given in table 1.

4 Set up new Case

4.1 Image Folder & Files

Within the script `main_proc` the user is supposed to change the folder path `dir_eval` in order to point to the images to be evaluated. The results from the PTV processing are saved per

default within the newly created subfolder `results`. But this path can be changed by defining another path within the variable `dir_save`.
A few minor restrictions on the image names are still hard coded within the scripts but can be modified for your needs. Right now, the image order is determined by an increasing four digit number. The prefix is custom and no other suffix then the file format is allowed. For the double frame method, the image pair is distinguished by image `A` and `B`.

4.2 Parameter File

Recording Parameters

Important processing parameters are saved inside a `parameter.m` MATLAB script file. The first parameters are recording information covering the image acquisition mode `acq_mode` (either `double_frame` or `time_resolved`), the image resolution `im_res` H x W in pixel, the number of captured images or image pairs `n_frames`, the time separation `dt`, the image mapping scale `m` as well as a pre-defined image mask `im_roi`. The parameter file has to be created for each new case and should be located in the image folder.

Particle Detection Parameters

Two variables have to be adjusted for an optimal particle detection. These are the particle size `p_size` measured in pixel and the particle intensity `p_int` measured in image intensities/ counts.

Tracking Parameters

In the following the user can decide, whether the nearest neighbour or the histogram matching tracking algorithm should be used for the double-frame processing. For this the parameter `track_method` should contain the string value `nearest` or `hist_match`, respectively. For both methods an important parameter is the field of search `f_o_s` for the particle linking. When using the histogram matching method, the number of neighbouring particles `n_neighbours` and the decision, whether a sub-pixel gauss interpolation `gauss_interp` shall be conducted, have to be defined additionally.
When time resolved image sequences are evaluated the variable `min_dist` has to be defined, too. It describes the maximum allowed error in pixel between the predicted particle position and the actual coordinates within the images. At last, it is possible to perform an iterative first trajectory initialization by increasing the number of steps `n_mp_ti` above 1.

Outlier Detection Parameters

After each tracking step, an outlier detection is performed. This function needs three input values. These are the number of considered neighbouring particles `n_outlier`, a threshold level `thr`, which decides whether a vector is rejected or not and an estimated noise level `noise` of the experiment.

Multi-pass Parameters

The double-frame particle tracking also incorporates a multi-pass approach. This can either be consist of an iterative particle matching, where the matching is perform again after rejected

Table 1: Recommendations for input parameters.

Parameter	Recommendation
`p_size`	odd pixel value (1, 3, 5, ...)
`p_int`	min. value of particle intensity found in image
`track_method`	histogram matching algorithm `hist_match`
`f_o_s`	twice the maximum expected particle displacement
`n_neighbours`	25
`gauss_interp`	yes
`n_outlier`	same as `n_neighbours`
`thr`	2 - 5 (lower number - more rejected vectors)
`noise`	0.1
`n_outl_iter`	1 - 3
`n_mp`	1 - 3 (depends on strength of displacement gradients)
`min_dist`	1
`n_mp_ti`	1-3

vectors have been removed or where the field of search is step wise increased or where both utilities are combined. The number of iteration steps is defined by `n_outl_iter` and the number of steps to increase the field of search is set by `n_mp`.