FlowVelo: An image-based tool to automatically measure flow velocities

This program tracks flow velocities using image or frame sequences. It includes measuring the camera position and orientation (camera pose), automatic extraction of the water area for feature searching, particle detection and filtering, particle tracking and filtering and scaling the tracks to flow velocities. Furthermore, a function is included to automatically co-register frames in case of camera movements.

Requirements:
Python 2.7
Tkinter
scipy 1.2.1
scikit-learn 0.20.3
scikit-image 0.14.2
shapely
imageio 2.5.0
opencv 3.2.0
seaborn 0.9.0
matplotlib 2.2.2

Running in Windows

- Install Anaconda for Python 2.7
- You need to install opencv extra → thus, open cmd and type (if conda not found add C:\Users*YourUser*\Anaconda3\Scripts and C:\Users*YourUser*\Anaconda to environment variable):
    - conda install -c conda-forge opencv=3.2.0
- Unpack FlowVeloTool.zip and put all scripts including the tutorial folder into one folder
- Launch Spyder
- Open *GUI_FeatureDetectionTracking.py*

Input data:

- Interior geometry (minimum information needed is focal length in mm and sensor size in mm and sensor resolution) of the camera (example file *interiorGeometry.txt* in tutorial data and corresponding explanation of parameters in *interiorGeometry_explained.txt*)
- Image coordinates (id,x,y) of GCPs used to estimate camera pose (example file *markersGCPimg.txt* in tutorial data, location of GCPs in image is illustrated in *locationGCPs_frame3.jpg*)
- Object space coordinates (ID,X,Y,Z) of GCPs used to estimate camera pose (example file *markersGCPobj.txt* in tutorial data)
- 3D point cloud (X,Y,Z) of area of interest (example file *3DmodelPointCloud.txt* in tutorial data) to define search area. If no point cloud is given, it is also possible to import a file containing the image coordinates of the search mask)
- Water level
- Folder with the image/frame sequence

Tutorial data:
FlowVeloTooll.zip available at https://cloudstore.zih.tu-dresden.de/index.php/s/2XWTnzqCkoOJvkF

To start program run script *GUI_FeatureDetectionTracking.py*

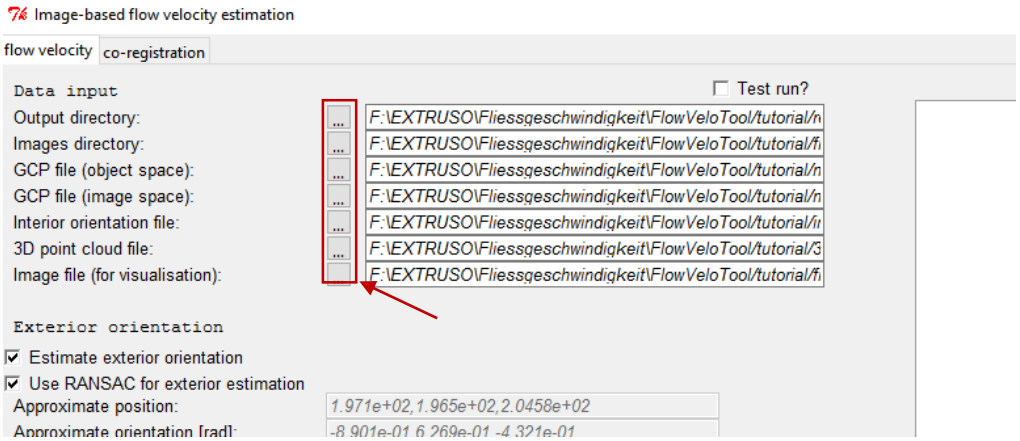
→ open Terminal to type:

```
python /your_path/GUI_FeatureDetectionTracking.py
```

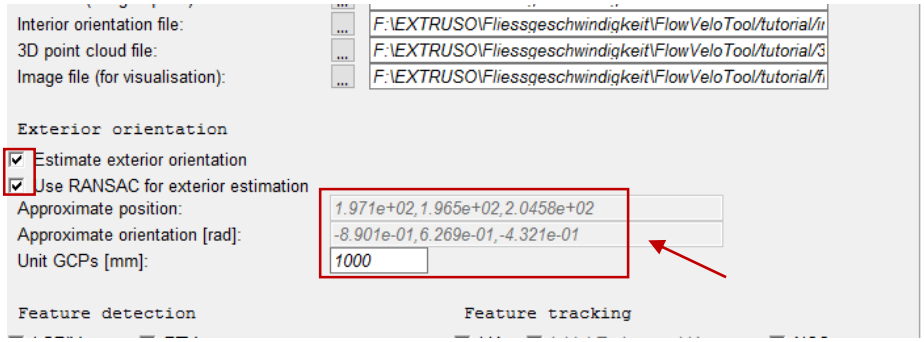
For questions please contact: Anette.Eltner@tu-dresden.de

Short Guideline to estimate flow velocities:

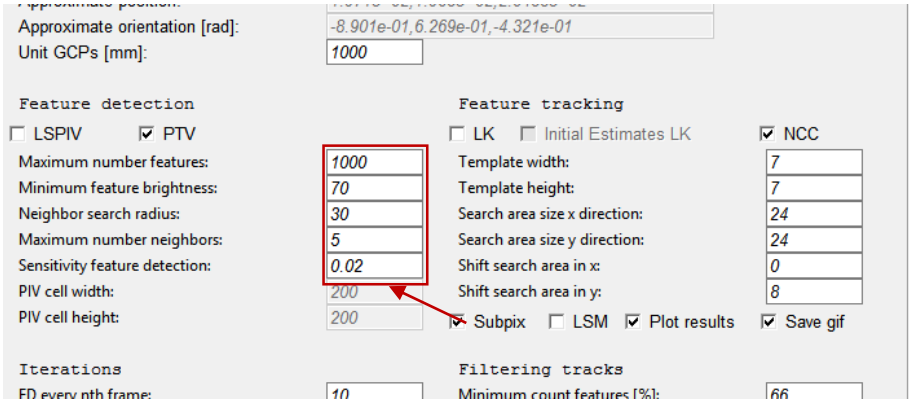
1. Set parameters to load input data



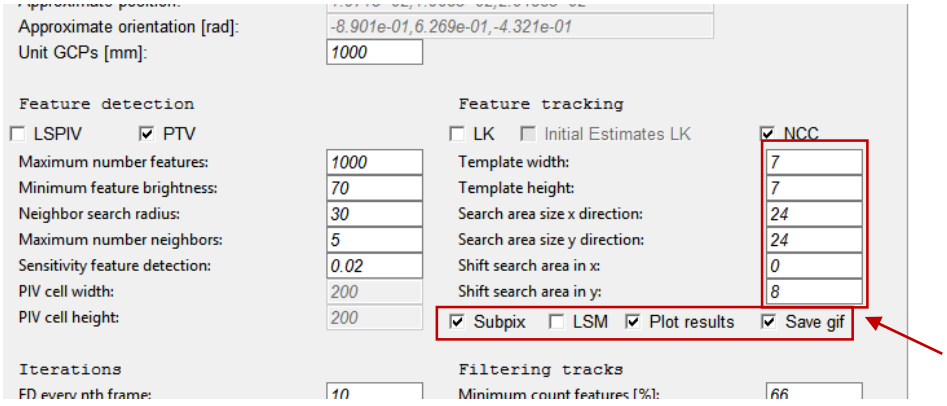
2. Set choices for camera pose estimation
   - "Unit GCPs" define unit of GCP because all measurements are performed in mm because the interior camera geometry has to be provided in mm
   - furthermore, it is possible to provide information of the camera pose (if it is known) instead of using GCPs to estimate the exterior orientation → if this step is chosen, uncheck "Estimate exterior orientation" and "Use RANSAC for exterior estimation" and type in camera pose information ("Approximate position", "Approximate orientation")
   - finally, a combination of approximately provided camera pose information and some GCP measurements is possible → if this step is chosen uncheck "Use RANSAC for exterior estimation" and type in camera pose information



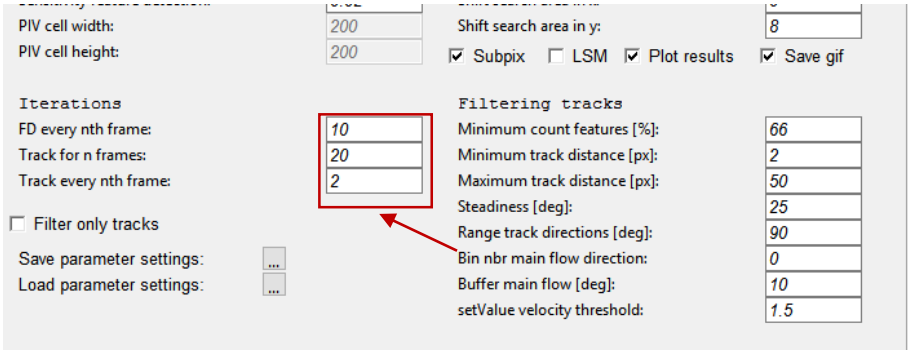
3. Set parameters for feature detection
    - “Minimum feature brightness” threshold for brightness of feature to be considered as particle, the higher the value the brighter the feature has to be (thus increasing this value leads to decrease of found features)
    - “Neighbor search radius” threshold defines radius (in pixels) in which nearest neighbors are searched for to perform subsequent cluster filtering (the higher the value the more features will be filtered)
    - “Maximum number neighbors” defines how many neighbors are allowed within searched radius (the higher the value the more features are kept)
    - “Maximum number features” maximal total number of detected good features to track (afterwards they are filtered)
    - “Sensitivity feature detection” (quality level) the higher the value the less features will be detected



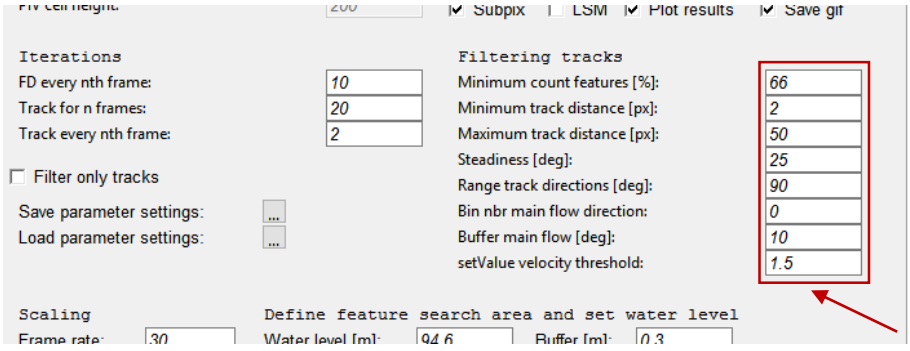
4. Set parameters for feature tracking
    - “Template width” and “Template height” defines template size in pixels (the higher the value the longer the processing time), size has to be odd
    - “Search area size x direction” and “Search area size y direction” defines size of area in pixels within template (i.e. detected particle) is searched for (the higher the value the longer the processing time and the higher the chance for ambiguities)
    - “Shift search area in x” and “Shift search area in y” allows to define approximate flow direction and thus to improve matching chances, value is given in pixels (the higher the flow velocities the larger the shifts)
    - If subpixel accurate results are aimed for check “Subpixel”
    - It is also possible to retrieve subpixel accurate results using least square matching. If this is needed check “LSM”
    - Choose whether results are to be plotted and if gif with animated tracking is wished for. If so, check “Plot results” and “Save gif”



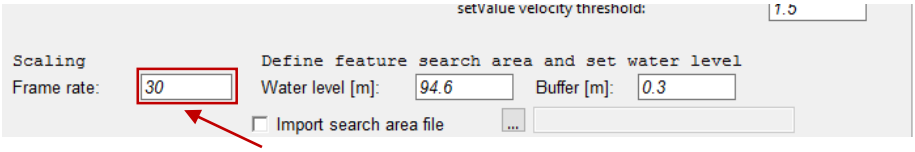
5. Set parameters for repetition of feature detections and duration of feature tracking
    - “FD every nth frame” detect features after defined number of frames/images
    - “Track for n frames” track features for defined number of frames/images
    - “Track every nth frame” track feature in every frame (value 1), every second frame (value 2), every third frame (value 3), …



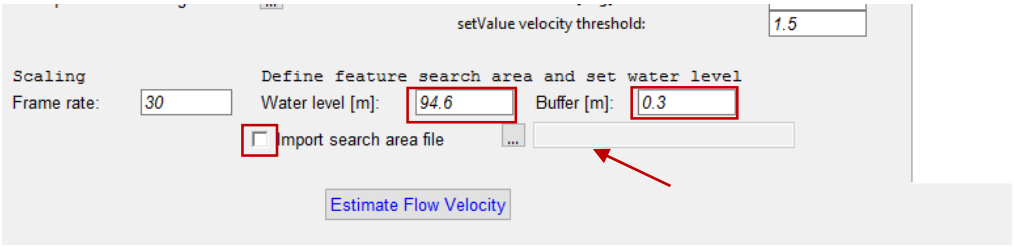
6. Set parameters for track filtering
    - “Minimum count features” define across how many images/frames (in percentage of pre-set track number – track for n frames) has to be tracked to be considered as reliable track
    - “Steadiness” define how strong feature can change direction from one frame to next (value corresponds to maximum allowed average direction change)
    - “Range track directions” define range within features can change directions
    - “Bin nbr main flow direction” leave at 0
    - “Buffer main flow” define how strong entire track can deviate from main flow direction
    - “Minimum track distance” define minimum distance which feature has to move between consecutive frames to be considered as reliable track
    - “Maximum track distance” define maximum distance which feature has to move between consecutive frames to be considered as reliable track
    - “setValue velocity threshold” (threshold = mean + setValue * standard deviation) set value to calculate threshold above and below which features are removed (the higher the value the more tracks are kept)



7. Set parameter to scale velocities
    - “Frame rate” set frame rate with which videos or images had been captured



8. Set parameter to define water area (feature search area) and set water level to correctly retrieve 3D coordinates of tracked features
    - “Water level” set water level
    - “Buffer” set buffer for water level, which needed in very complex terrain or when camera pose estimation not very accurate to buffer water borders
    - “Import search area file” if feature search area is selected manually provide extent with file containing image coordinates of mask (file has to be txt that contains two columns [x,y], which are separated by comma)



<u>Short Guideline co-register frames/images:</u>

1. Set parameters
    - “Maximum number of keypoints” set maximum number of detected Harris corner keypoints
    - “Number of good matches” set minimum number of matched keypoints to allow for estimation of homography
    - “Matching with SIFT” if checked uses SIFT feature descriptor, if unchecked uses ORB feature descriptor
    - “Feature matching 2sided” check to increase robustness, but also increases computation time
    - “Register to first frame” if checked registers all frames of sequence to first frame, if unchecked register frame to previous frame