StageDetect: An image-based tool for automatic water stage detection

This program detects water stage using image sequences. It includes camera orientation, template matching for GCP detection, master retrieval from image sequence, image co-registration, water line detection, and transforming 2D points into 3D coordinates.

Runs in: Python 2.7 in Linux

Additionally needed Python libraries:

- openCV version 2.4.13
- scikit-image
- scipy
- statsmodels
- seaborn
- shapely
- scikit-learn
- Tkinter

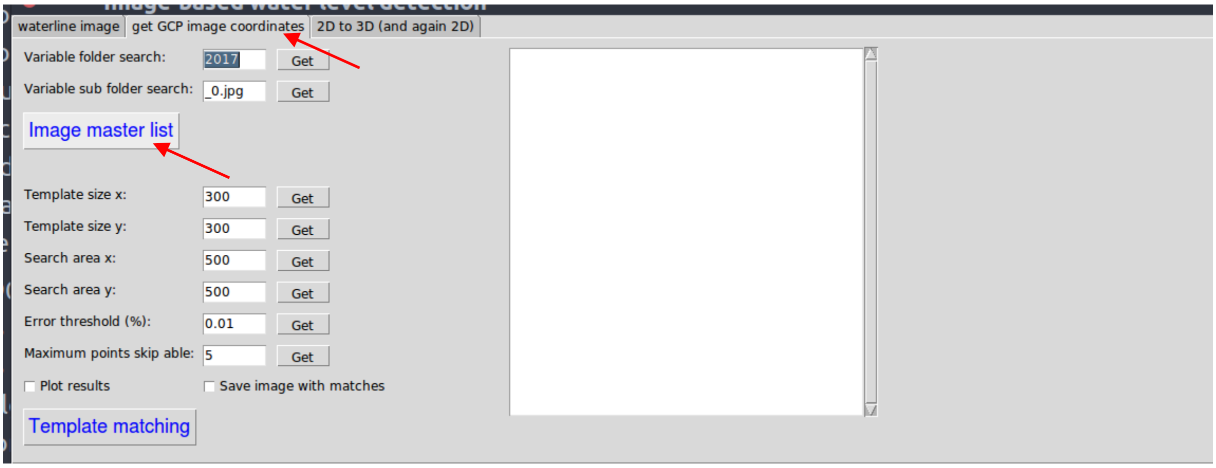
Tutorial data: *StageDetect_tutorial.zip* available at https://cloudstore.zih.tu-dresden.de/index.php/s/1IPZgvnkTz89kHG

Run script: *StageDetect_gui.py*

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

Short Guideline:

1. Retrieve master image of each image sequence for further analysis: rider "get GCP image coordinates" → button "Img Master List"
   - Folder with images is folder "img_raw" in tutorial dataset
   - save all masters in a new folder



Variable folder search: 2017

Variable sub folder search: _0.jpg

Template size x: 300

Template size y: 300

Search area x: 500

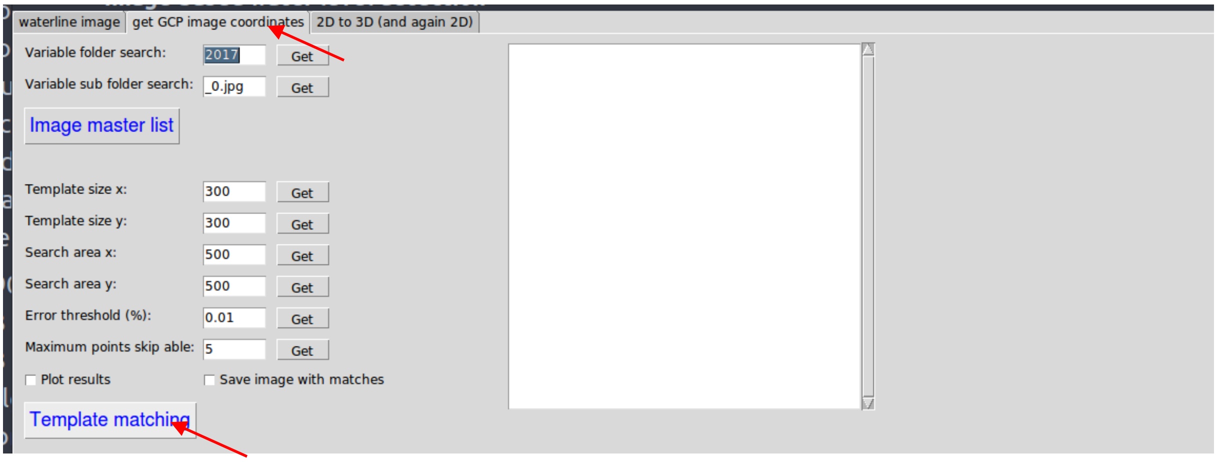
Search area y: 500

Error threshold (%): 0.01

Maximum points skip able: 5

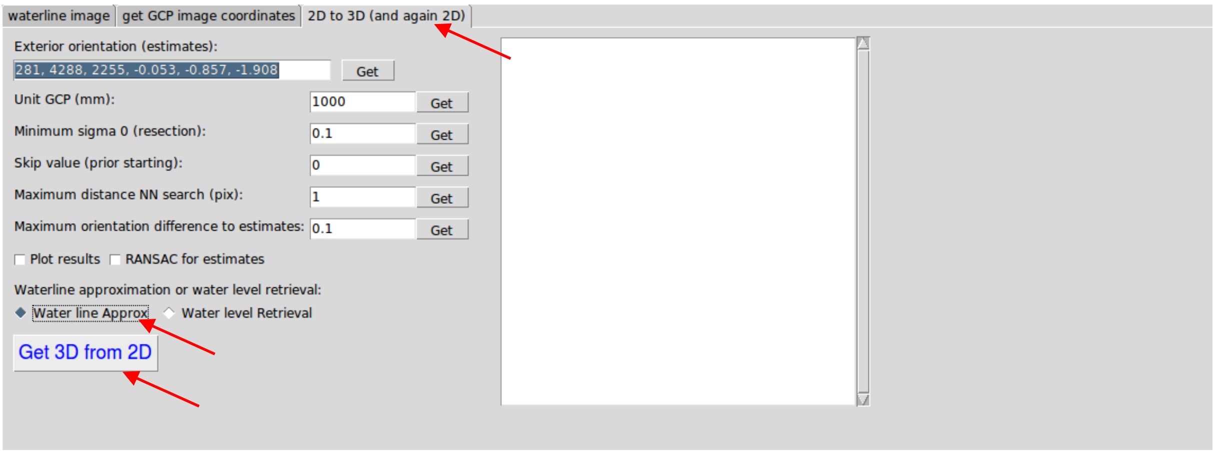
□ Plot results □ Save image with matches

2. Retrieve GCP image points for each master image of all sequences: rider “get GCP image coordinates” → „Template Matching“
    - for first image GCP coordinates have to be measured in image manually, e.g. using GIMP → in tutorial dataset it is file “Template_GCP_img.txt”
    - corresponding image to initial GCP image coordinates has to be given/defined → in tutorial dataset it is file “firstImgTemplate.jpg”
    - master image folder (step 1) should be used for template matching



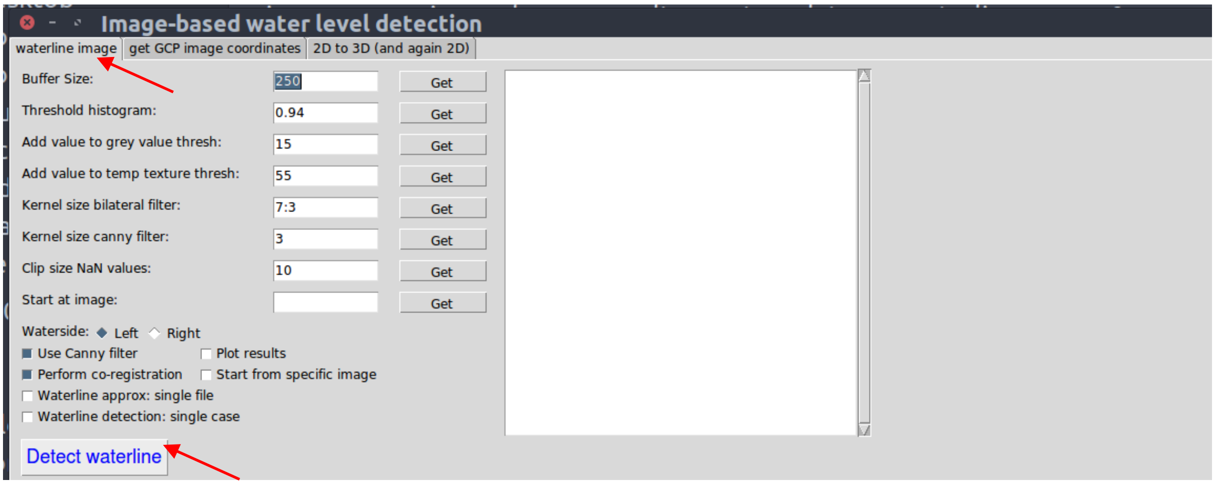
Variable folder search: 2017
Variable sub folder search: _0.jpg
Template size x: 300
Template size y: 300
Search area x: 500
Search area y: 500
Error threshold (%): 0.01
Maximum points skip able: 5
□ Plot results
□ Save image with matches

3. Get approximation of water line (to define RoI): rider “2D to 3D (and again 2D)” → check “Water line Approx” → button “Get 3D form 2D”
    - initial water line file has to be defined manually, e.g. getting xy image coordinates using GIMP → in tutorial dataset it is file “waterline_approx.txt”
    - interior camera geometry (e.g. from calibration file has to be given to correct for image distortion and to orient image) → in tutorial dataset it is file “CameraCalibration.txt” (CameraCalibration_explained.txt includes definition of parameters)
    - GCP coordinates have to be given → in tutorial dataset it is file “GCP_coordinates.txt”
    - Corresponding image measurements to GCP coordinates are in result folder of step 2



Exterior orientation (estimates): 281, 4288, 2255, -0.053, -0.857, -1.908
Unit GCP (mm): 1000
Minimum sigma 0 (resection): 0.1
Skip value (prior starting): 0
Maximum distance NN search (pix): 1
Maximum orientation difference to estimates: 0.1
□ Plot results □ RANSAC for estimates
Waterline approximation or water level retrieval:
● Water line Approx ○ Water level Retrieval

4. Estimate waterline: rider “waterline image” → button “Detect waterline”
    - Water line approximations from step 3 are needed or if only one RoI defined for all sequences (i.e. if no camera movement is expected) “Waterline approx.: single file” has to be checked



Buffer Size: 250

Threshold histogram: 0.94

Add value to grey value thresh: 15

Add value to temp texture thresh: 55

Kernel size bilateral filter: 7:3

Kernel size canny filter: 3

Clip size NaN values: 10

Start at image:

Waterside: ◆ Left ◇ Right

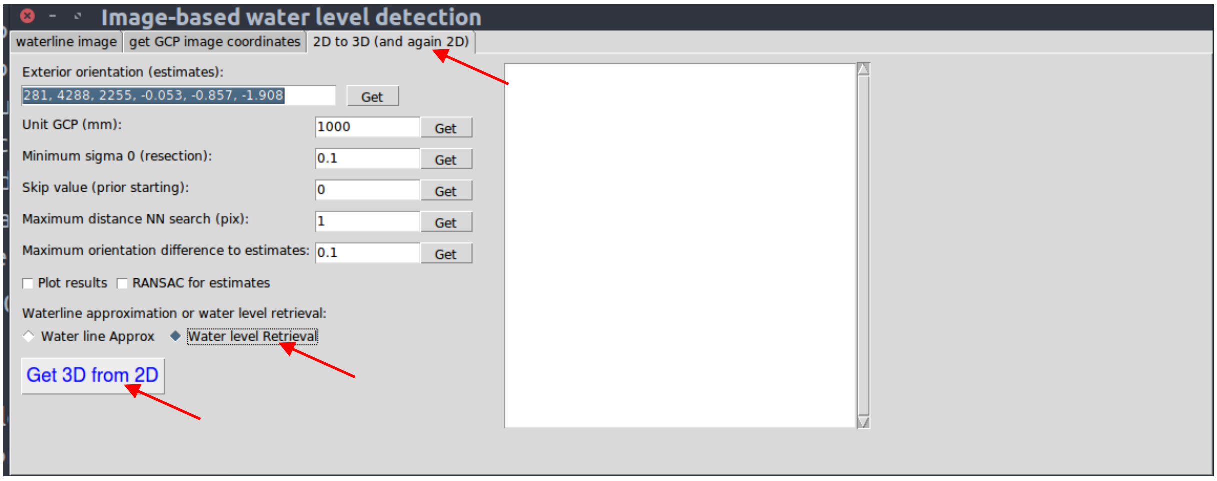
■ Use Canny filter □ Plot results

■ Perform co-registration □ Start from specific image

□ Waterline approx: single file

□ Waterline detection: single case

5. Get 3D water stage from water line in images considering 3D reference data (e.g. scaled SfM point cloud): rider “2D to 3D (and again 2D)” → check “Water level Retrieval” → button “Get 3D form 2D”
    - interior camera geometry (e.g. from calibration file has to be given to correct for image distortion and to orient image) → in tutorial dataset it is file “CameraCalibration.txt” (CameraCalibration_explained.txt includes definition of parameters)
    - GCP coordinates have to be given → in tutorial dataset it is file “GCP_coordinates.txt”
    - Corresponding image measurements to GCP coordinates are in result folder of step 2
    - 3D point cloud of observed river area has to be given → in tutorial dataset it is file “PointCloud_3D.txt”
    - Detected water lines from step 5 are needed



Exterior orientation (estimates): 281, 4288, 2255, -0.053, -0.857, -1.908

Unit GCP (mm): 1000

Minimum sigma 0 (resection): 0.1

Skip value (prior starting): 0

Maximum distance NN search (pix): 1

Maximum orientation difference to estimates: 0.1

□ Plot results □ RANSAC for estimates

Waterline approximation or water level retrieval:

◇ Water line Approx ◆ Water level Retrieval