

Feature Tracking Velocimetry Applied to Airborne Measurement Data from Murg Creek

Liekai Cao^{1,2,*}, Volker Weitbrecht², Danxun Li¹, Martin Detert²

¹State Key Laboratory of Hydro-Science and Engineering, Tsinghua University, 100084, Beijing, China

²Laboratory of Hydraulics, Hydrology and Glaciology VAW, ETH Zurich, 8093, Zurich, Switzerland

Abstract. A new image feature tracking velocimetry is presented and tested on airborne video data available from a previous study at Murg Creek (Canton Thurgau, Switzerland). Here, the seeded flow scenery had been recorded by an off-the-shelf action camera mounted to a low-cost quadcopter, and video frames were ortho-rectified to sizes of 4482×2240 px² at a scale of 64 px/m. The new velocimetry approach is as follows: An adaptive Gaussian mixture model is used for video background subtraction. Then, scale-invariant keypoints on each remaining binary foreground image frame are determined by a feature detection algorithm, and corresponding feature points in subsequent frame pairs are matched using the iterative random sample consensus method. The related feature shifts in metric space divided by the video frame rate finally give the velocity vectors. The obtained velocimetry fields are compared with findings from both a particle image velocimetry and particle tracking velocimetry analysis in terms of accuracy and needed computational power. Indication is given that the feature tracking algorithm presents slightly less precise results, but clearly outperforms the other two in relation to computational power. Therefore, the new simplified method provides a high potential tool that may enable a future way to real time surface velocity measurements obtained from unmanned airborne vehicles.

1. Introduction

In recent years, airborne image velocimetry (AIV) by rotary-winged drones gets developed to conduct in-situ measurement for river surface velocity fields (for example, [1, 2, 3, 4]), and even to estimate both bathymetry and flow discharge ([5]). AIV can be performed with satisfactory accuracy on low-cost basis ([2, 5]), and it has the advantages of obtaining river flow fields up to reach length scale ([2]).

AIV mainly consists of two procedures: image rectification, including undistorting lens effects, and flow field computation.

For the former, a straight forward approach comprises to use projective transformation based on matching feature points between the frames (for example, [1, 2]) that have been lens-corrected by adequate mathematical camera models. More advanced, techniques of Structure from Motion (SfM) and MultiView Stereo (MVS) algorithms can be used to derive

* Corresponding author: clk15@mails.tsinghua.edu.cn

ortho-images, with the by-product of 3D bathymetry data biased by refraction (e.g. [2, 5]). However, some studies give no clear statement to their approach to ortho-rectify airborne image at all (e.g., [3, 4]).

Then, during the latter procedure, flow fields are computed by Particle Image Velocimetry (PIV, e.g. [2, 3]), Particle Tracking Velocimetry (PTV, e.g. [4]), or, in case the local main flow direction is known and only time averaged velocities are of interest, by space-time image velocimetry (STIV, see [1]). However, these three methods require a high level of user input to setup the computation parameters. For example, users have to choose a proper interrogation area for PIV, an adequate particle centroids' identification method and displacement threshold for PTV, or an appropriate search line arrangement for STIV. In addition, a major shortcoming of PIV and PTV is that they are both demanding in terms of computational power.

In the actual paper, we apply Feature Tracking Velocimetry (FTV, [6]) for AIV. It uses a feature tracking algorithm to automatically determine the displacement field based on matched feature points in successive images. It can be used with only a low level of user input and performs velocity computations at high speed. All computations in this study are performed with *MATLAB 2017b*.

2. Data set of Detert et al. (2017)

Recently, airborne river flow measurement data have been published [5]. The data set is available at https://figshare.com/articles/S1S2S3_Murg_20160406_zip/4680715/1 for free. This data set is used in the actual study to test our feature tracking velocimetry approach. A brief description is as follows: In [5], the seeded flow scenery has been recorded on a length of 80 m by a gimbal-stabilized, off-the-shelf action camera mounted to a low-cost quadcopter. During the analysis, 1'000 video frames had been georeferenced and ortho-rectified by SfM and MVS with a high-resolution of 64 px/m and 4482×2240 px².

3. Feature tracking velocimetry (FTV)

3.1. Foreground detection based on GMM

In order to detect features that present the movement of seeded river surface, the adaptive Gaussian mixture model (GMM, [7, 8]) was used to detect the foreground image. It models each pixel as a mixture of Gaussians and uses an on-line approximation to update the model. The Gaussian distributions of GMM are then evaluated to determine which are most likely resulting from a background process. This probabilistic method deals robustly with lighting changes, repetitive motions of scene elements, tracking through cluttered regions and slow-moving objects, which are also the main features on seeded river surface. Another advantage is the low level of user input, only containing two significant parameters: the learning rate and the proportion of data that should be used to determine the background distribution.

In this study, a five-mode GMM was used to process all 1'000 ortho-images. To speed up the computation, the non-water surface area was firstly blanked out by a mask before applying the GMM. On a *Windows 10* PC (64 bit, *Intel(R) Xeon(R) CPU E3-1245v5*) the process speed reached about 4–5 fps. Fig. 1 shows a snapshot of a typical result after applying the GMM. Moving particles, but also textures of people, shrubbery (apparently moving due to the motion of the camera) and the rope of the ADCP boat become pronounced.

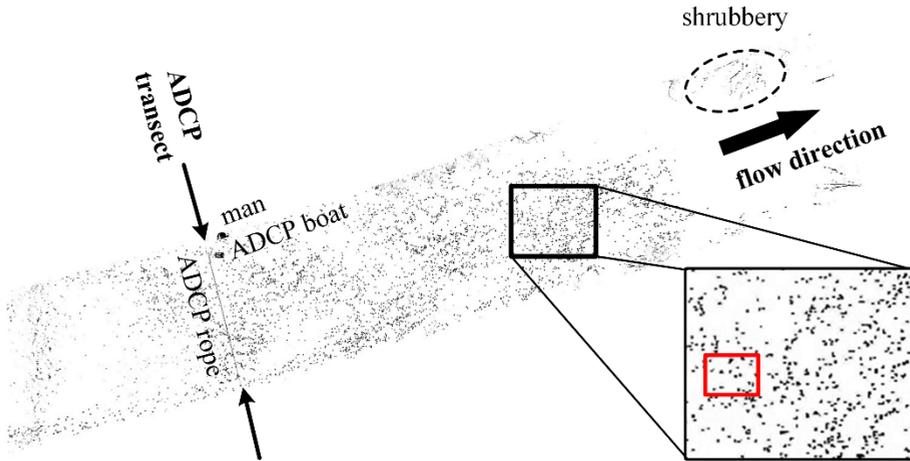


Fig. 1. Snapshot view to binary foreground image frame #620 as filtered by the GMM method. Besides moving particles also other moving textures become obvious. To enable interpretation, black and white are reversed here. The bold rectangle of 419×264 px² gives a closer view to the appearance of the seeding. The red bold rectangle marks the region shown in Fig. 2.

3.2. Feature detection and matching

Feature detection and matching between images is a fundamental step in computer vision. Generally, one feature is defined by one keypoint and its descriptor. Generally, keypoints are high in (color) texture, referred to as corners and edges, which can be mathematically described as regions with high gradient of intensity ([6]). Once the keypoints are detected, a descriptor should be assigned to each keypoint. Commonly used descriptors are, e.g., SIFT ([9]), SURF ([10]), and BRIEF ([11]). Finally, the feature matching process will be done by RANdom SAMple Consensus (RANSAC, [12]). In the actual study, SURF, a scale- and rotation-invariant interest point detector and descriptor, was applied to detect keypoints on the water surface. Fig. 2 illustrates the approach of feature detection and matching on a small region of 86×73 px², with 31 pairs of matched SURF points.

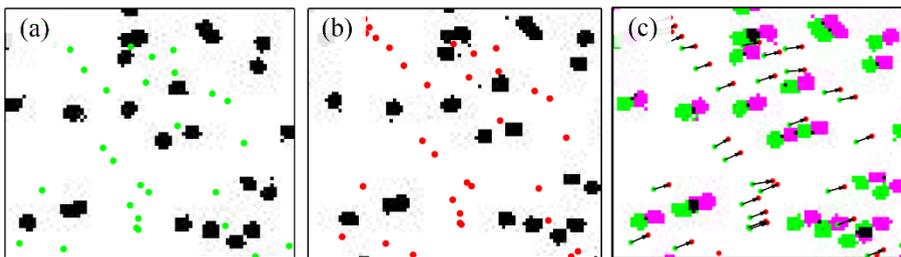


Fig. 2. Foreground image of 86×73 px² (see Fig. 1), SURF points (small dots), and their matching. (a) frame #620, (b) frame #621, and (c) according displacements between frames #620–621.

3.3. Computation of velocity fields

The final step to compute velocity vectors is straight-forward: The related feature shifts in metric space divided by the video frame rate in time space give the velocity vectors. Fig. 3 illustrates an example, where instantaneous velocity fields are calculated for the full frame #620–621. Velocity vectors are densely distributed where also seeding particles are densely populated, while spurious vectors appear at riparian sides due to imperfect ortho-rectification

resulting from apparently moving shrubbery. In general, FTV can produce sufficiently enough vectors in areas well covered with seeding particles.

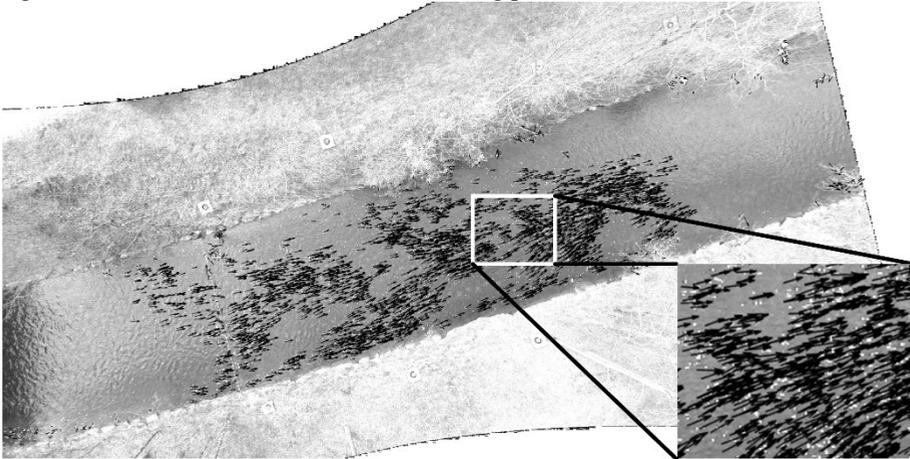


Fig. 3. Computed instantaneous vector field of frames #620–621. The bold rectangle of $419 \times 264 \text{ px}^2$ gives a closer view to appearance of vectors and seeding.

4. FTV versus PIV and PTV

4.1. Approach

To rate the accuracy of the new FTV approach, results are compared to classical PIV and PTV. All three methods are applied to 1'000 successive images in order of [1–2, 2–3, ..., and 999–1000] resulting in 999 velocity fields.

Similar to the FTV pre-processing, the non-water surface was blanked out by a mask first.

For PIV computations the open source software *PIVlab* [13] was used. First, RGB ortho-images were converted to 8 bit grayscale images and then a background image was subtracted which was obtained by applying a disk-shaped filter with diameter of 10 px. Pixels of intensity value below 50 (max. 255) were set to zero and the remaining pixels' intensity were curvedly stretched by nonlinear gamma operation ([14]) with $\gamma = 0.8$ to enhance the particles pixels and weaken the remaining misleading background pixels. PIV analysis was applied on a final window size of $32 \times 32 \text{ px}^2$ with 50% overlap.

The PTV results are obtained by a relaxation algorithm ([15]). For image pre-processing, the images first were high-pass filtered by a disk filter with diameter of 10 px. Then they were transferred to binary images using Otsu's ([16]) threshold method. This generated particle centroids with a combined threshold of a particle area of 28 px and long-to-short axis ratio of 2.5. During the PTV computation, the limits were set to 8 px for maximal displacement and to a maximum number of iterations of five.

Finally, a universal outlier detection method ([17]) was applied to all velocity fields obtained by FTV, PTV, and PIV to eliminate misleading velocity vectors.

To enable comparison of all three velocimetry methods, results from PTV and FTV are converted to the equidistant grid raster of the PIV results. To this end, a nearest neighbour weighting approach was applied, that considers the distance of raster midpoints and the starting point of the velocity vector. Only rasterized vectors were accepted if they were computed out of at least three raw vectors.

4.2. Time-averaged surface flow fields

Fig. 4 gives a geo-referenced survey of FTV, PIV, and PTV to the full surface velocity fields within a total reach length of about 80 m. It becomes obvious that FTV computes velocity fields being quite similar as those from PIV and PTV – except some riparian zones. Furthermore, PTV results in the smoothest mean velocity field due to its high spatial resolution, where even the influence of the ADCP rope diminishes.

Quantitative analysis of the computation error was conducted in a rectangle region as marked in Fig. 4 to exclude the influence of misleading vectors at the river banks. Accuracy is accessed by Root-Mean-Square Error (RMSE) between the matrix of differences, **MoD**, of measured and the ‘true’ time averaged velocity. Here, we assume that PTV results give the most correct velocities, i.e. they are the ‘true’ values. In this way, the error gives RMSE of $\mathbf{MoD}(U_{bulk}(FTV) - U_{bulk}(PTV)) = \pm 11 \text{ mm/s}$ and RMSE of $\mathbf{MoD}(U_{bulk}(PIV) - U_{bulk}(PTV)) = \pm 6 \text{ mm/s}$, with U_{bulk} being the time averaged flow field of $\sqrt{U_x^2 + U_y^2}$ computed in the rectangular area as highlighted by dashed lines in Fig. 4. These values correspond to only 1.4% and 0.7% of the mean value of $U_{bulk}(PTV) = 0.81 \text{ m/s}$, indicating that U_{bulk} is almost the same for all three velocimetry methods.

Thus, the differences in U_{bulk} are acceptably small. In general, these findings indicate that the new FTV method is capable of quantitatively describing the averaged surface velocity field quite well – although its RMSE-value to PTV is doubled in comparison to PIV.

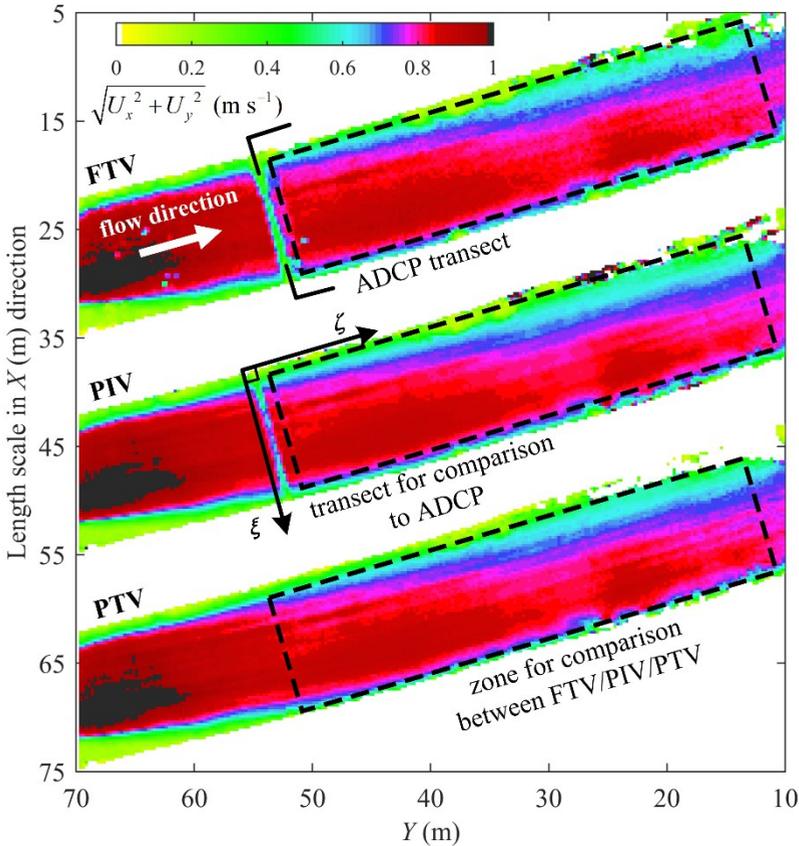


Fig. 4. Mean surface velocity fields of FTV, PIV and PTV, where X denotes easting and Y northing. $[U_x, U_y]$ are [east, north] component of velocity. The coordinates ζ and ξ correspond to the streamwise and spanwise direction. Dash-lined black rectangles highlight the area for quantitative analysis.

4.3. Time averaged image velocimetry versus ADCP results

FTV and PIV produce wrong velocity vectors at the ADCP measurement transect due to the influence of rope, human and ADCP boat in the quasi-simultaneous ADCP measurements. Therefore, the ADCP data are compared to the image velocimetry data obtained at an undisturbed transect 0.5 m upstream of the ADCP transect (see the solid line in Fig. 4). The ADCP velocities were recorded 0.14 m below the water surface, making a direct comparison to the image velocimetry results less conclusive. To this end, a complex log-fit was applied through each vertical ADCP profile to estimate the surface velocity, similar to the method used in [5]. Fig. 5 compares time averaged surface velocity results of image velocimetry and ADCP. Obviously, all measurements give quite similar shapes in transversal direction ξ – besides some small differences close to the banks. In general, the FTV (and PIV and PTV) results fit the ADCP results very well.

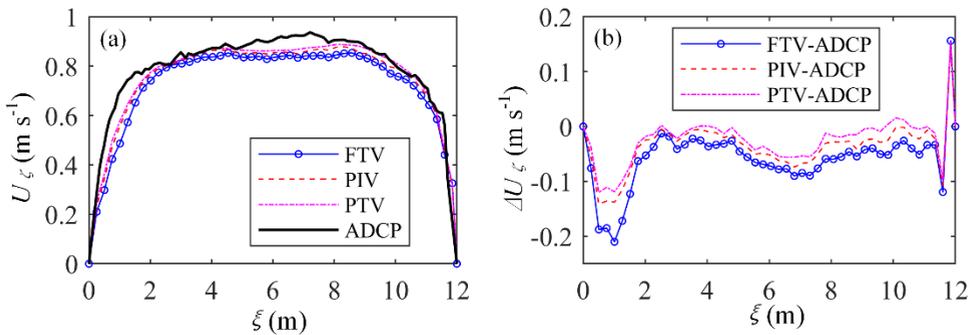


Fig. 5. Cross-sectional surface velocities, where U_{ξ} is the streamwise velocity and ΔU_{ξ} is the difference of streamwise velocity between [FTV, PIV, PTV] and ADCP.

4.4. Instantaneous surface velocity fields

Fig. 6 shows instantaneous surface velocity fields computed for frames #620–621 (i.e. at time $t = 58.3$ s) by FTV, PIV, and PTV. Obviously the flow field of FTV covers the smallest area of river surface compared to PIV and PTV, i.e. less vectors have been derived here.

To get a detailed view to the chronological sequence of RMSE-values of the **MoDs** of all instantaneous frames, they are computed for all resulting 999 flow fields. Results are plotted in Fig. 7. In this way, the error gives RMSE of $\mathbf{MoD}(U_{ij}(\text{FTV}) - U_{ij}(\text{PTV})) = \sim 0.08$ m/s and RMSE of $\mathbf{MoD}(U_{ij}(\text{PIV}) - U_{ij}(\text{PTV})) = \sim 0.05$ mm/s, with U_{ij} being the instantaneous flow field of $\sqrt{U_{x,ij}^2 + U_{y,ij}^2}$ computed between frames $\#i-j$ in the rectangular area as highlighted by dashed lines in Fig. 4. Consequently, for instantaneous flow fields, FTV presents smaller numbers of vectors and less precise values with an error of about $\pm 10\%$ of $U_{bulk}(\text{PTV}) = 0.81$ m/s.

4.5. Computational time

Processing time is measured for the complete process of computation which includes image pre-processing and velocity computing. Tab. 1 lists the results. It becomes obvious that FTV performs best, both at velocity computational time and total time, resulting in a $\sim 50\%$ improvement related to PIV and PTV.

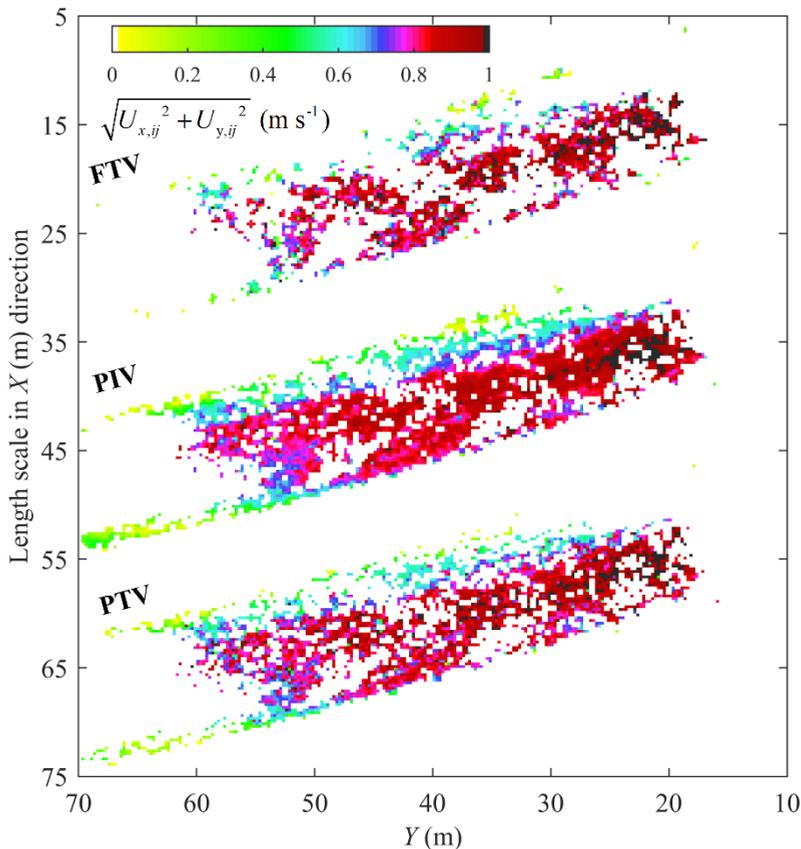


Fig. 6. Instantaneous surface velocity fields computed for FTV, PIV and PTV for frames #620–621.

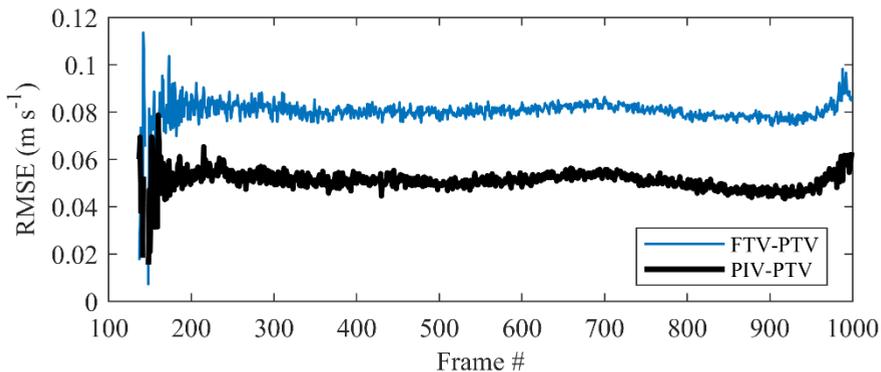


Fig. 7. RMSEs of instantaneous flow fields of FTV and PIV as compared to PTV.

Table 1. Processing time of FTV, PIV and PTV on 1000 images.

Items	FTV	PIV	PTV
Image pre-processing time (s)	278	214	414
Computation time (s)	2,783	6,154	6,000
Total time (s)	3,061	6,368	6,414

5. Concluding remarks

This study presents a new FTV approach to compute velocity fields of surface flows in natural rivers. At its current version it is thought to be applied to rectified airborne images with a seeded water surface.

The velocimetry analysis is as follows: An adaptive GMM is used for image background subtraction. Then, scale-invariant keypoints of the binary feature objects in the remaining foreground image frames are detected by the SURF algorithm. Next, corresponding points in subsequent pairs of video frames are matched using the iterative RANSAC method. Related feature shifts in metric space divided by the video frame rate finally give the velocity vectors.

Its application is verified by ortho-rectified airborne video data available from a previous study at Murg Creek (Switzerland, [5]). Here, the scenery had been recorded by an off-the-shelf action camera mounted to a low-cost quadcopter. Benchmark tests against standard algorithms of PIV and PTV give indication that the accuracy reached for time averaged flow fields has an error of about $\pm 1\text{--}2\%$. For instantaneous flow fields, FTV leads to a smaller number of vectors and less precise values with an error of about $\pm 10\%$. However, FTV is about two times faster than PIV and PTV in terms of computing time. Especially compared to PTV, the level of needed user input is much lower and is therefore, less prone to errors by the user. FTV provides a high potential tool in the framework of airborne river flow measurements that may enable a future way to drone-based velocimetry in real time.

Further development steps are required to optimize the new FTV approach to be applied by a broader user group. These steps comprise: (1) implement and study further feature detectors (e.g., SIFT or ORB), (2) smooth the tracks to increase the precision (e.g., [4]), (3) improve the code towards more stable performance, and (4) test FTV's performance under unseeded, i.e. natural flow conditions.

This study was partly founded by the Chinese Scholarship Council (No. 201706210226).

References

1. IC. Fujita, YU. Notoya, MI. Shimono, *36th IAHR World Congress* (2015)
2. M. Detert, V. Weitbrecht, J. Hydraul. Res **53**, 4 (2015)
3. F. Tauro, A. Petroselli, E. Arcangeletti, Hydrol. Process **30**, 7 (2015)
4. P. Thumser, C. Haas, J. A. Tuhtan, J. F. Fuentes-Pérez, G. Toming, ESPL (2017)
5. M. Detert, E.D. Johnson, V. Weitbrecht, J. Remote Sens **38**, 8 (2017)
6. C.C. Zuniga Zamalloa, B.J. Landry, *HMEM Conference* (2017)
7. C. Stauffer, W.E.L. Grimson, *IEEE Computer Society Conference* (1999)
8. P. KaewTraKulPong, R. Bowden, Video-based surveillance systems **1** (2002)
9. D.G. Lowe, *7th IEEE international conference* (1999)
10. H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Comput. Vis. Image Underst **110**, 3(2008)
11. E. Rublee, V. Rabaud, K. Konolige, G. Bradski, *IEEE international conference* (2011)
12. M.A. Fischler, R.C. Bolles, *Communications of the ACM* **24**, 6 (1981)
13. W. Thielicke, E. Stamhuis, J. Open. Res. Softw. **2**, 1 (2014)
14. C. Poynton, *Digital video and HD: Algorithms and Interfaces* (Elsevier, 2012)
15. S.J. Baek, S.J. Lee, Exp. Fluids **22**, 1 (1996)
16. N. Otsu, IEEE Transactions on Systems, Man, and Cybernetics **9**, 1 (1979)
17. J. Westerweel, F. Scarano, Exp. Fluids **39**, 6 (2005)