



SmartPIV: flow velocity estimates by smartphones for education and field studies

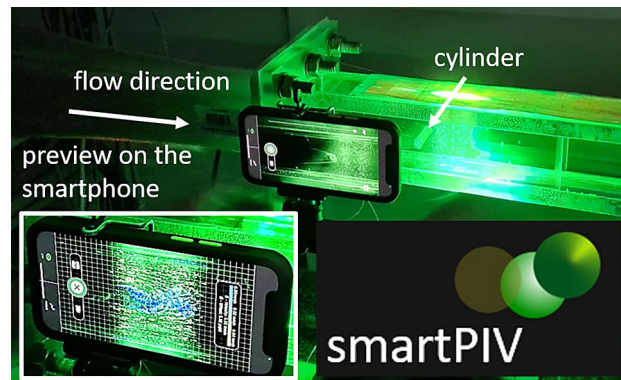
Christian Cierpka¹ · Henning Otto¹ · Constanze Poll² · Jonas Hüther² · Sebastian Jeschke² · Patrick Mäder²

Received: 4 February 2021 / Revised: 1 July 2021 / Accepted: 5 July 2021 / Published online: 25 July 2021
© The Author(s) 2021

Abstract

In this paper, a smartphone application is presented that was developed to lower the barrier to introduce particle image velocimetry (PIV) in lab courses. The first benefit is that a PIV system using smartphones and a continuous wave (cw-) laser is much cheaper than a conventional system and thus much more affordable for universities. The second benefit is that the design of the menus follows that of modern camera apps, which are intuitively used. Thus, the system is much less complex and costly than typical systems, and our experience showed that students have much less reservations to work with the system and to try different parameters. Last but not least the app can be applied in the field. The relative uncertainty was shown to be less than 8%, which is reasonable for quick velocity estimates. An analysis of the computational time necessary for the data evaluation showed that with the current implementation the app is capable of providing smooth live display vector fields of the flow. This might further increase the use of modern measurement techniques in industry and education.

Graphic abstract



1 Introduction

Smartphones considerably changed our behavior and communication in recent years and are used on a daily (or even every minute) basis without any trouble by students. Fluid flows also belong to their daily experiences, however, the education of the basics of fluid mechanics or thermodynamics is sometimes cumbersome. Due to the nonlinearity of the Navier–Stokes equations, it is difficult to establish an intuitive access to flows. This problem is best solved in practical sessions applying flow visualization techniques in wind or water tunnels and directly learn from observations.

✉ Christian Cierpka
christian.cierpka@tu-ilmenau.de
Patrick Mäder
patrick.maeder@tu-ilmenau.de

¹ Institute of Thermodynamics and Fluid Mechanics, Technische Universität Ilmenau, 98684 Ilmenau, Germany

² Data-intensive Systems and Visualization, Technische Universität Ilmenau, 98684 Ilmenau, Germany

Nowadays, often optical methods like particle imaging velocimetry (PIV) (Willert and Gharib 1991; Raffel et al. 2018; Adrian and Westerweel 2011) or particle tracking velocimetry (PTV) are used for these purposes.

By adding tracer particles to the fluid and measuring their movement with digital camera equipment and advanced evaluation algorithms, these methods provide the possibility to measure velocities in a measurement plane or even a volume. Compared to point measurement methods such as laser Doppler velocimetry (LDV) or hot wire anemometry (HWA), the introduction of these measurement techniques has already contributed enormously to a better and more intuitive understanding of flows. A recent overview of advanced methods is provided by Kähler et al. (2016). Unfortunately, a typical PIV/PTV setup consists of a (double-) pulse laser, a scientific camera and a synchronization device. The costs for this equipment can easily add up to 100,000 euro. Furthermore, the installation and set up is complex. For these reasons, universities often only offer practical courses for a small number of students in which students may not even be allowed to use and to set up the systems themselves since the equipment is used for scientific projects in parallel.

However, modern smartphones offer a great selection of different sensors and are easy to use. The camera technology is quickly advancing, and the processors will become more and more powerful. For this reason, smartphones were already used for physical experiments in classes (Staacks et al. 2018; Klein et al. 2016), for plant identification in botanical classes (Mäder et al. 2021; Wäldchen and Mäder 2018), for determining fluid properties (Chen et al. 2017; Goy et al. 2017; Solomon et al. 2016) and even for flow visualization by Schlieren techniques (Settles 2018; Miller and Loebner 2016). The high frame rates of several hundred Hz captured by modern smartphone cameras enables their use for PIV. Cierpka et al. (2016) have shown that it is possible to use a smartphone with a continuous wave (cw-) laser for reliable velocity estimates in a plane. Kashyap et al. (2020) validated the image recoding of different smartphones using an open PIV software and achieved relative differences below 7% in comparison to numerical results. Käufer et al. (2020) extended the planar PIV system to stereoscopic PIV using two consumer action cameras and a modulated cw-laser and Aguirre-Pablo et al. (2017) even used smartphones and colored LEDs for a tomographic reconstruction of the velocity field in a jet. All these attempts were based on previously recorded videos that were later processed on a powerful workstation with conventional PIV software.

Only recently, a survey among students in engineering showed that there is a strong interest in a mobile application (app) to perform PIV measurements (Minichiello et al. 2020). Therefore, the aim of the current study was to provide an app that allows for a direct evaluation of the

data in order to enable students to directly see how the flow behavior changes when certain boundary conditions are varied. Furthermore, this allows them to already estimate if the data processing with the current video setting will be successful if a larger video will be captured. In addition, to the benefits a smartphone PIV application offers in laboratory courses, it can also be used for measurements in the field. For example, some of the app's test users reported to use it for estimating the flow velocity of a river from a bridge by tracking air bubbles on the surface or to visualize entering mist through an opened window in the winter or the rising mist on heated walls on sunny winter mornings. Moreover, any kind of little experiments can easily be performed using, for instance, kitchen accessories to visualize the Marangoni effect in a cooking pot. Therefore, the app offers the possibility to demonstrate this measurement technique with such small experiments to children and people without fluid engineering background and to convey enthusiasm for such topics. In addition, it may also be used in wastewater treatment plants or civil engineering of dams and channels. As this allows applications beyond education and from now on the term users instead of students will be used.

The paper is structured as follows. In Sect. 2, the design of the app is described, Sect. 3 provides details on the software implementation for Android, iOS and Harmony OS. A validation experiment with known displacement is shown in Sect. 4 and a typical lab course example of the flow behind a cylinder is presented in Sect. 5 followed by a summary and outlook in Sect. 6.

2 The SmartPIV app

Since one of the major aims was to lower the technical barrier in applying PIV, the app was designed to be very similar to common camera and video capturing apps. In Fig. 1, SmartPIV's main screen is shown. For validation purposes, a rotating disk was equipped with a printed particle pattern.



Fig. 1 Main menu of the SmartPIV app showing the option data evaluation, recording and live preview (here cross-correlation analysis of a rotating disk)

The corresponding vectors can be seen in the live preview mode. The main advantage is that this live preview directly responds to changes of the motion of the particle pattern. In lab sessions, students can get a direct impression of the change in the velocity field when they vary flow parameters. A color bar and the length and color of the displayed vectors will directly give an estimate for the magnitude of the vectors. In addition, the mean magnitude of the particle displacement (measured in pixels) per time step Δt and the mean velocity magnitude are given, the latter in the case a scaling factor in units of millimeter per pixel is set to calculate it from the displacement (more details on the topic follow in the subsequent discussion of Fig. 3). The mean particle displacement allows the users to estimate whether the settings (see Fig. 2) are appropriate for a larger video capture and a later evaluation of more data. The users can also choose between autofocus (AF) and manual focus (MF) where they set the focus via a slider. The autofocus helps to see if the mobile phone is placed accordingly but it is always recommended to turn off the camera's autofocus during the measurements as in the case of changing light conditions, smartphones try to refocus sometimes, which may lower the frame rate for some time and change the optical magnification. To minimize its influence would require specific post-processing routines that are not included in the app. Therefore, the manual focus can be used with the advantage that it is fixed for the duration of the image acquisition.

From the main menu, the user can directly go to the video capture mode by selecting the small camera icon above the green button. Alternatively, the current data can be exported by activating the capture icon below. In this case, the two current frames, an image overlaid with vectors, a text file with the main processing parameters, and text files with the underlying displacement data are stored in a specified folder on the device. This allows for a later use for example for flow analyses and lab reports.

In Fig. 2, the parameter settings for the two implemented data evaluation methods are shown. A main parameter in the settings menu is the frame rate of the camera. This frame rate f_r determines the time difference between two successive frames $\Delta t = 1/f_r$. The image recording is performed automatically, and the implementation will be described in Sect. 3. Different smartphone models support different maximum frame rates. For many models (including the hardware used for this study), this maximum frame rate is 240 Hz. This seems to be a typical value for consumer slow motion pictures at a typical resolution of 1280×720 pixel, and many systems support this frame rate already. From the software side, there is no limitation to choose also a higher frame rate if supported by the smartphone model. This will increase the measurable velocity range which is $\sim M \cdot f_r$, where M is the optical system's magnification factor. A generic graph providing the limits for the measurable physical velocity for

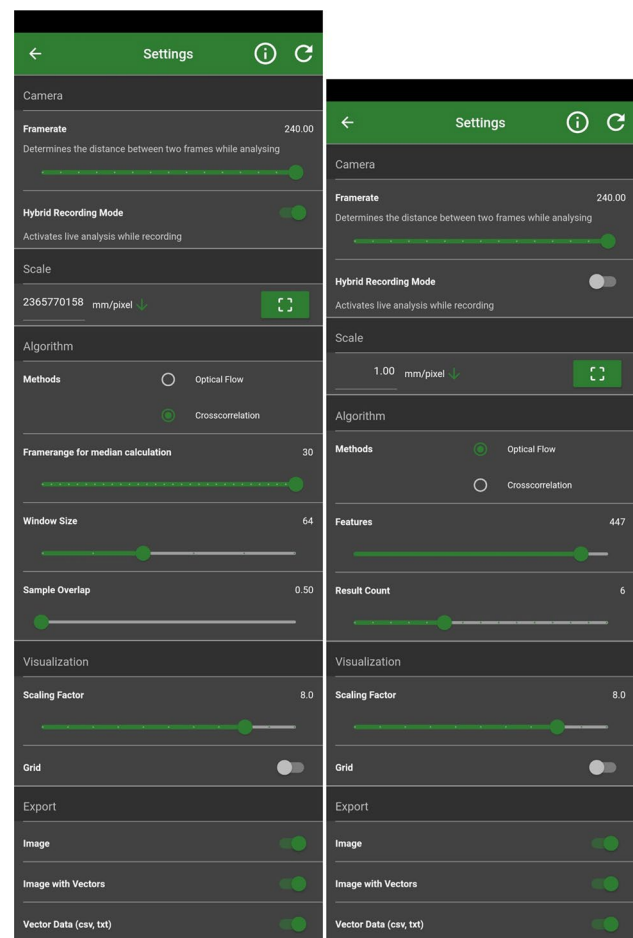


Fig. 2 Parameter options for the two data evaluation modes of cross-correlation (left) and optical flow analysis (right)

different magnifications and frame rates is provided in our previous work and may help to design the desired experiment (Cierpka et al. 2016). However, with increasing frame rate the duration for the illumination will decrease $\sim 1/f_r$, and often the intensity of the particle images gets very low using cw-lasers. One of the main drawbacks of consumer cameras for PIV is that they typically use a rolling shutter. This might result in systematic errors, especially when large pixel displacements shall be measured (Käufer et al. 2020). However, the main scope was not to provide a high precision measurement system but an easily accessible app that features all basic PIV parameters that should be known for education.

The settings menu allows to choose the evaluation methods cross-correlation (CC) or optical flow (OF). The reason to introduce the OF in addition to the CC was to provide users with a less powerful smartphone processor access to the application since the CC algorithm is more demanding compared to the OF. Depending on the chosen method the

parameter menu changes. Common settings for both methods are the hybrid recording (live data evaluation is done while recording videos), the scaling factor (for scaling the vector length in live display) and the export options for the data export.

On the left side of Fig. 2, the typical parameters for cross-correlation are shown. These are in particular the interrogation window size and the sample overlap (here implemented as sample offset, with a sample offset of 0.5 corresponding to 50% interrogation window overlap) that both can be set via software sliders. If the temporal evolution of the flow is slow or the flow is stationary, it is possible to show median vectors to remove spurious vectors from the live display. For this reason, also the range of frames, which are used to calculate the median, can be set. Finally, it is possible to overlay the current grid for a better visual inspection and choice of interrogation window size and sample offset.

The settings associated with optical flow processing are shown on the right side of Fig. 2. The user can decide on the maximum number of features to be tracked. Especially for older hardware models, the computational time can be lowered if less features are chosen. For modern smartphones, the number of features does not affect the computational time significantly in the range between 0 and 500 (see Fig. 5). However, in some cases less vectors may be chosen to have a more clearly representation of the flow field. Similar as in the cross-correlation options, the evaluation from more than one double frame image can be overlaid. This parameter is currently referred to as ‘result count’.

When the settings are chosen, the results are immediately displayed in the live image. Users can directly see if the current settings result in useful velocity data. When the flow field changes, the difference in the velocity fields is directly observable, which may result in a more intuitive access to the flow and an understanding of the main PIV processing parameters.

At this point, it has to be mentioned that no outlier detection (apart from the median calculation for cross-correlation) is implemented so far. On one hand, this would require additional processing resources and slows down the online evaluation. On the other hand, the implementation of an outlier detection is a typical task in lab courses, and students can do this later on the basis of the stored displacement fields.

It was shown in a previous study that the camera optics are reasonable accurate if no additional wide field objective lenses are used (Cierpka et al. 2016). For this reason, no calibration function is implemented in the software. However, to have an estimate on the velocity in physical units, a menu to automatically determine a scaling factor and the relative position of the camera plane was implemented. As can be seen in Fig. 3, the system detects automatically the position of the corners of a square that serves as target and should be placed in the light-sheet plane. To be able to see the detected

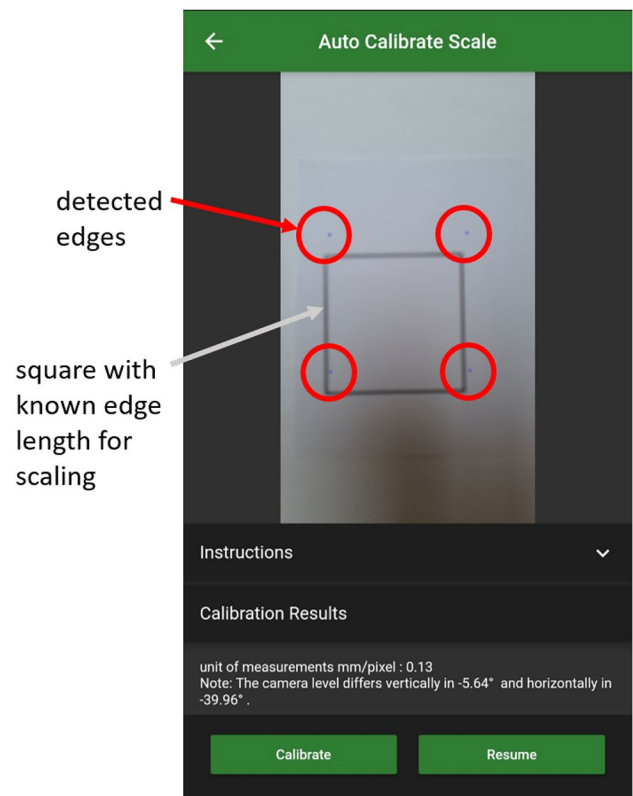


Fig. 3 Automatic calibration menu to determine a scaling between physical units and image coordinates. The target was displaced to make the corner detection visible

corners with the small blue circles, the smartphone was moved for the picture. The focus settings for this image are the same as in the main menu. If the detection was successful the vertical and horizontal rotation angle will be given. This indicates if the target was placed correctly and is planar enough to give a proper scaling factor. If a more sophisticated calibration, including image deformation correction of more complex experimental setups is necessary, videos with a typical calibration target can be captured. Later, users can extract the corresponding frames and develop their own calibration routines to convert image coordinates to physical coordinates in a post-processing step based on the stored displacement fields (Käufer et al. 2020).

Since the main purpose of the app is the use in lab courses, it allows the storage of videos for later post-processing. In contrast to the evaluation of only two frames, the videos are compressed using the smartphone’s preset compression methods. However, so far no significant influence of the image compression on the vector results could be seen (Cierpka et al. 2016). The video capture enables users to test different processing parameters and to calculate mean flow fields based on the average vector fields of a video. The video capturing works as in a typical camera app. In the hybrid mode (see settings menu in Fig. 2), the app shows the

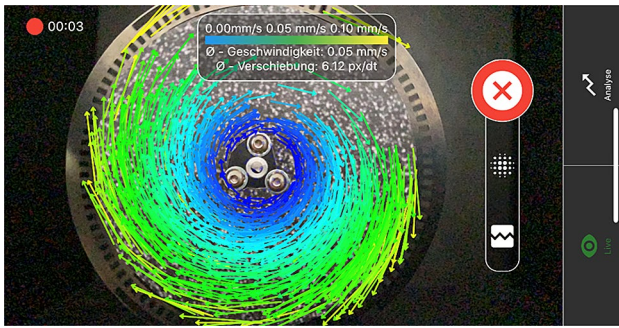


Fig. 4 In the hybrid mode, the app shows the estimate of the velocity vectors while recording a high speed video for later analysis for the lab protocol

estimate of the velocity vectors while recording a high speed video for later analysis as can be seen in Fig. 4. The recorded duration of the video is shown in the upper left of the screen, and the recording can be stopped by the big red button.

3 Implementation

The SmartPIV app was implemented as a multi-platform app being available for Android, iOS and Harmony OS devices. The Google open-source Flutter framework (Google 2021) was used for the development of all user interfaces and high-level application logic, i.e., export, file handling, settings. Flutter forms an abstraction layer between the operating system and the application software and allows to write just one codebase for large parts of the software, i.e., there is no need to program individual versions of SmartPIV per operating system, which simplifies maintenance and guarantees that all systems share the same functionality. Flutter apps are developed in Dart, an object-oriented programming language characterized by a C-style syntax.

However, since the major aim was an efficient implementation, all frame grabbing and analysis parts were developed as device-specific algorithms and programs, i.e., Java and C for Android and Swift for iOS. These device specific solutions allow to adapt closely to the camera hardware and to utilize efficient mechanisms for the intensive analytical computations. On both platforms, frames are acquired from a H.264 (aka MPEG-4 Part 10) compressed video stream for the analysis. This utilizes the highest supported encoding level per device ensuring for best video quality. It was found that this is the only option when aiming for a widely applicable non-hardware-dependent implementation. However, in future versions of the app, it may be possible to adaptively select more up-to-date codecs like H.265 or let the user choose a preferred codec. Aiming for an efficient implementation of algorithms, e.g., Apple's Accelerate was utilized for the iOS implementation, a high-performance,

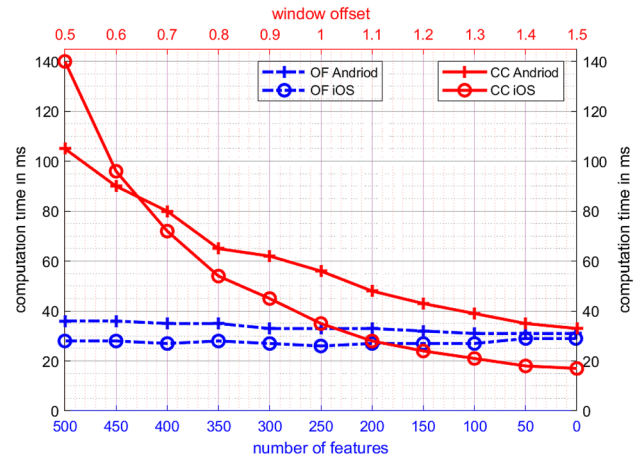


Fig. 5 Computational time for the determination of the displacement field for cross-correlation (CC) for different sample offsets (red curves and top x -axis) and for optical flow (OF) changing the number of tracked features (blue curves and bottom x -axis) for Android (cross symbols) and iOS (circles)

energy-efficient hardware-accelerated compute framework. Accelerate allows to, e.g., off-load the cross-correlation's large matrix multiplications to the phone's vector processing capabilities and thereby enables a massively parallelized and fast computation.

In Fig. 5, the computational time for the determination of the displacement field for cross-correlation (red curves and top x -axis) and for optical flow (blue curves and bottom x -axis) for Android (cross symbols) and iOS (circles) are shown. Note that a reversed x -axis in the bottom is used since decreasing numbers of features in the OF method correspond to an increasing sample offset of the CC method in a way that both lead to a decreasing number of resulting vectors. The analysis was performed on an OnePlus 7T Pro (Android 10)¹ and an Apple iPhone X (iOS 14.3)² and is based on ten repeated measurements to minimize the influence of operating system and hardware features that may influence execution speed. The mean values of these tests are plotted in the figure. Among the ten consecutive measurements, the evaluation time varied on the order of 1 ms. During these tests, no data transfer of other background application were running. However, it can be expected, that the evaluation time will increase for certain processes. An influence of the battery status was not observed.

In the case of cross-correlation, the analysis was performed using an interrogation window size of 64×64 pixel on images with a total size of 1280×720 pixel. For the analysis, the sample offset was changed. A value for the

¹ https://www.gsmarena.com/oneplus_7t_pro-9872.php.

² https://www.gsmarena.com/apple_iphone_x-8858.php.

sample offset 0.5 results in a sample overlap of 50% (920 vectors), 1 no offset using adjacent (220 vectors) interrogation windows and for 1.5 (104 vectors) the interrogation windows are increasingly separated excluding the region in between from the analysis. For the highest number of vectors, the computational time for the iOS implementation was about 140 ms whereas for the Android version it was only 105 ms. The computational time decreases with a decreasing number of vector fields for both systems. At a sample offset of 0.7, the iOS implementation is faster than the Android implementation. This is probably due to the fact that the calculation is conducted by vector processing units and the time needed to split the image into interrogation windows and transfer the data to the GPU becomes less important for a smaller number of interrogation windows. However, for the typical case of no sample offset the computational time is in the order of 35–56 ms for the iOS and Android implementation, respectively.

As can be seen on the blue curves the optical flow method is much faster for both systems and starts with about 30 ms for one tracked feature (please note that the axis is reversed to have a decreasing number of vectors starting from the left). Whereas the computational time does not change with the number of features for the iOS implementation, a slight increase is visible with increasing numbers of tracked features for the Android implementation up to 36 ms. The computational effort for the optical flow analysis is thus over a wide range in the same order of magnitude as the case of no sample offset using 64×64 pixel interrogation windows. Therefore, it can be concluded that the current implementation of both methods for Android and iOS are reasonably fast enough for a live vector display.

The time for frame grabbing (~ 3 ms Android, ~ 20 ms iOS) and the vector display (~ 15 ms) remains constant for all methods and adds up to the computational time so that the total processing times range between 50 and 160 ms. Therefore, the rate of refreshment of the live display changes between 6 and 20 Hz depending on the settings. As the refreshment rate of about 10–15 Hz will be considered by the human eye as a continuous motion the data evaluation by the smartphones is in most cases fast enough for a smooth live display of the vector field and changes in the flow, which was one of the major aims of the app development.

4 Validation

4.1 Synthetic images with ground truth

To test if in the implementation of the code was done in proper way, synthetic images with known ground truth were generated and evaluated by the different methods for the Android and iOS system. For image generation, the freely

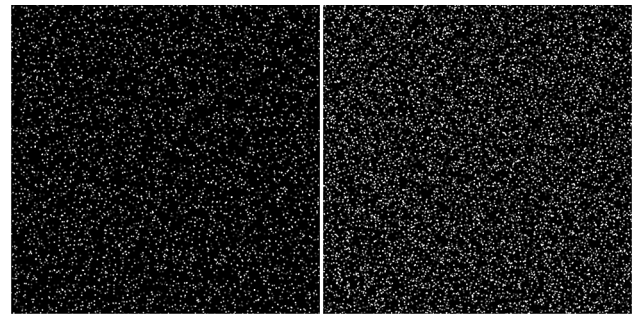


Fig. 6 Synthetic particle images with a uniform displacement of 5.5 pixel in the x -direction and 1.75 pixel in the y -direction for a particle image density of 0.02 ppp (left) and 0.04 ppp (right)

available synthetic image generator provided by PivTec GmbH was used.³ The particle images were randomly distributed. They had a particle image diameter of 2.5 pixel and show a uniform displacement of 5.5 pixel in the x -direction and 1.7 pixel in the y -direction, respectively. Two image pairs with a size of 512×512 pixel and two different particle image densities of about 0.02 and 0.04 particles per pixel (ppp) were simulated. These parameters are very well suited for precise cross-correlation analysis (Raffel et al. 2018). The synthetic images had no noise, and the pixel fill factor was set to 100%. To get a visual impression, the synthetic images for a particle image density of 0.02 ppp and 0.04 ppp are shown in Fig. 6 on the left and right side, respectively.

In addition to the evaluation with the smartphone app, the cross-correlation analysis was also performed using the commercially available software package PIVview2C 3.9.3 by PivTec GmbH. This package performed very well on the latest PIV challenge (Kähler et al. 2016) and could indicate if the scatter in the results of the smartphone-based evaluation is due to a wrong implementation of the algorithms. In order to have comparable results the same evaluation parameters were used for the software package (no image preprocessing, single pass evaluation, 64×64 pixel windows, sample overlap 0.5). At this point it has to be mentioned that PIVview2C has advanced parameter settings and routines that should be used for the evaluation of real experimental images and would give much better results than presented here as a benchmark. In addition, no outlier filters were used for post-processing. Therefore, the results are not representative of the best practice use of advanced PIV algorithms but represent the same level of evaluation complexity as the current smartphone implementation.

As can be seen in Table 1 all different algorithms are able to determine the displacement for this ideal conditions well. The mean displacement never deviates more than 0.04 pixel

³ <https://www.pivtec.com/>.

Table 1 Mean displacement and standard deviation for the different algorithms for the case of synthetic images with known ground truth displacement of 5.5 pixel in the x -direction and 1.7 pixel in the y -direction, respectively

| Particle image density in ppp | Algorithm | Mean displacement in px | |
|-------------------------------|------------|------------------------------|------------------------------|
| | | $\Delta X \pm SD (\Delta X)$ | $\Delta Y \pm SD (\Delta Y)$ |
| 0.04 | PIVview2C | 5.49 ± 0.01 | 1.69 ± 0.03 |
| | Android CC | 5.47 ± 0.02 | 1.69 ± 0.02 |
| | Android OF | 5.51 ± 0.19 | 1.69 ± 0.15 |
| | iOS CC | 5.47 ± 0.02 | 1.69 ± 0.02 |
| | iOS OF | 5.50 ± 0.07 | 1.68 ± 0.09 |
| 0.02 | PIVview2C | 5.49 ± 0.02 | 1.69 ± 0.02 |
| | Android CC | 5.47 ± 0.02 | 1.68 ± 0.02 |
| | Android OF | 5.54 ± 0.56 | 1.68 ± 0.05 |
| | iOS CC | 5.47 ± 0.02 | 1.68 ± 0.02 |
| | iOS OF | 5.50 ± 0.08 | 1.68 ± 0.05 |

from the ground truth, which indicates that the algorithms are implemented correctly on the smartphones. However, the measured standard deviations differ among the algorithms. For the analysis with the commercial software 0.01–0.03 pixel are determined. This is in the same order as for the CC analysis on the smartphones. The standard deviation for the optical flow methods is almost one order of magnitude higher than for the cross correlation. This is caused by only several (~10) outliers, which differ strongly in the displacement amplitude. To highlight this finding, the histograms for the displacement for a particle image density of 0.04 ppp are presented in Fig. 7. As can be seen the displacement distribution is much more narrow for the analysis with OF in comparison to the CC methods.

In summary, it can be stated that the algorithms are well implemented and give an accuracy that is close to the one by commercially available software if most advanced features (e.g., window weighting, multi-grid processing, multi-pass evaluation, etc.) are turned off in that software.

4.2 Experimental images without ground truth

To get an impression of the accuracy of the algorithms and determine whether systematic problems like, e.g., peak locking occur, experimental images will be used here. These were acquired with a smartphone a video sequence as described in Sect. 5. Four frames were extracted that show a signal to noise ratio (here defined as the ratio of the mean background peak in the histogram to the mean particles intensity) of ~10. The images suffer from the rolling shutter and short particle image streaks in regions of larger velocities (Cierpka et al.2016; Kaufer et al.2020). A visual impression can be seen in Fig. 11. The evaluation was done with PIVview2C using a multigrid evaluation starting with

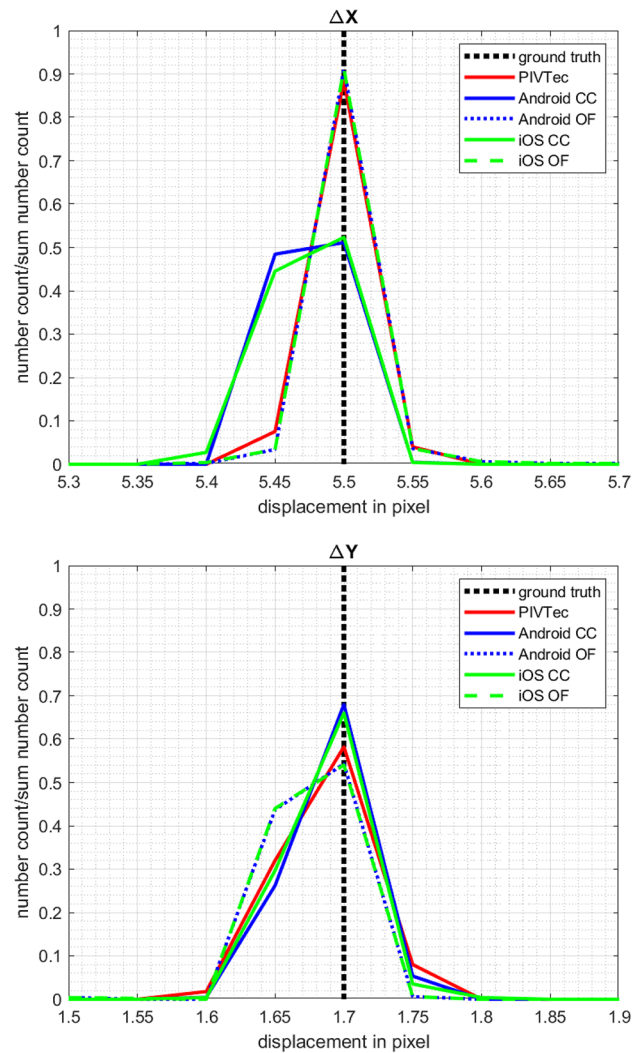


Fig. 7 Histograms of the measured displacement in the x -direction (top) and in the y -direction (bottom) for a particle image density of 0.04 ppp

128 × 128 pixel interrogations windows. The final window size was 64 × 64 pixel with 50% overlap. Outlier detection with the normalized median test with (standard deviation < 3) and interpolation was used to smooth the vector fields. For the smartPIV only the results for iOS are shown, as the Android results are very similar as demonstrated above. The vector field for the first image pair can be seen in Fig. 8 in the top. Due to different coordinate systems, the vector positions for the correlation methods do not coincide. However, the main features of the flow field can be seen. As inherent to the method the vector position for the optical flow evaluation are randomly distributed, but show also good agreement by visual inspection. The outlier test and the advanced processing parameter for the commercial software give especially in the region of the cylinder wake and close to the walls different results in comparison to the simple processing by

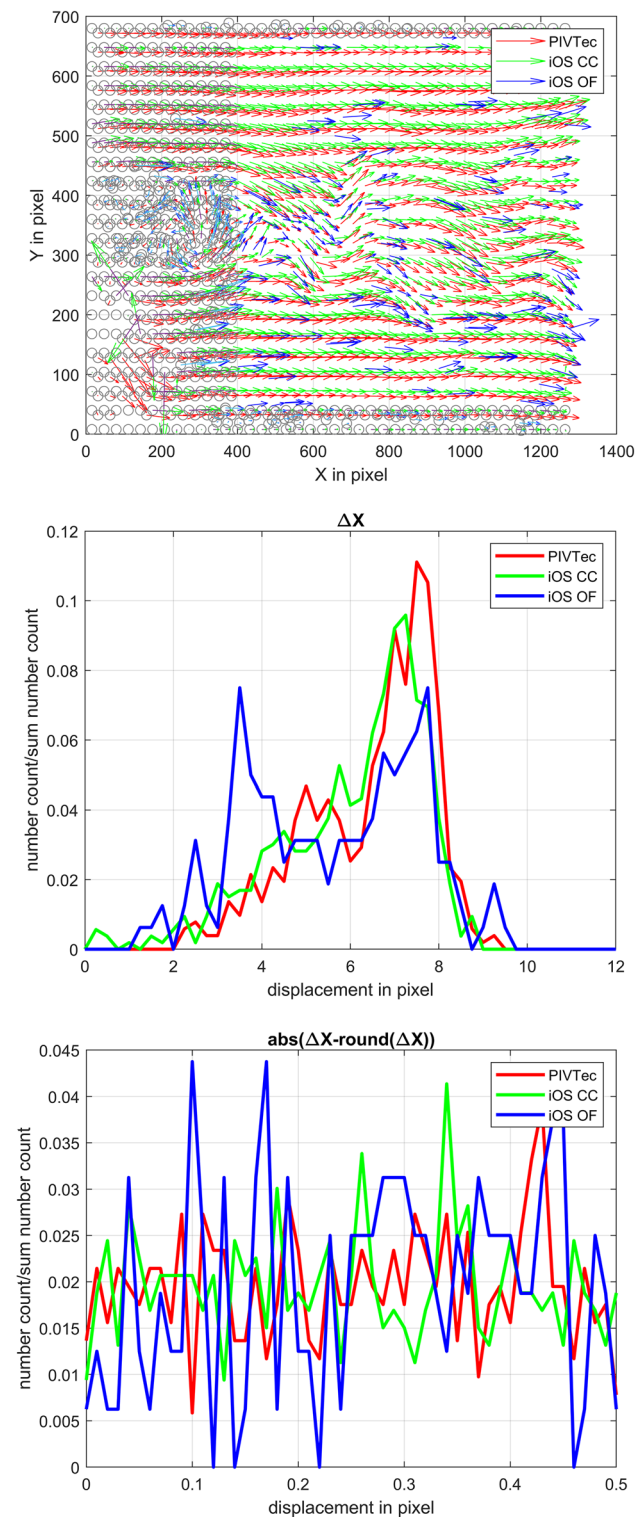


Fig. 8 Vector field of the first image pair (top). Grey dots indicate the filtered vectors for the histograms. Histograms of the measured displacement in the x -direction (middle) and histogram of the sub-pixel displacements in x -direction (bottom) in a region downstream the cylinder

the app. For this reason, the grey marked vectors were filtered out for the analysis of the histograms in Fig. 8 in the middle and bottom part. In the middle, the histogram for the displacement in x -direction for all four double frames can be seen. The agreement between the advanced cross-correlation analysis and the smartPIV app is well. The mean displacements for the commercial software are 6.45 pixel and 6.05 pixel for the smartphone app, respectively. This difference is due to the different processing and would be minimal if the same simple processing would be used for both methods. However, the good agreement shows that in the case of high signal to noise ratios (here ~ 10) the results by the smartphone app are reasonably good.

The histogram for the optical flow method differs much stronger. This is expected, since the algorithm determines only results for certain features and does not guarantee a uniform spatial vector distribution. In the current case, especially features with smaller displacements seem to be favoured. However, if the randomly distributed vectors were interpolated onto a similar grid as for the cross-correlation analysis the resulting vector fields agree very well, as shown in Sect. 5.

In the bottom part of Fig. 8, the histogram of the sub-pixel displacements $\text{abs}(\Delta X - \text{round}(\Delta X))$ are shown. The bins are chosen to match the stored accuracy of 0.01 pixel for the smartphone app. As can be seen, the sub-pixel displacement is uniformly distributed, and no indication of systematic errors or peak locking can be seen.

4.3 Experimental images with ground truth

For a validation using experimental images with a known ground truth, a printed particle pattern, with a radius of roughly 40 mm, was attached to an optical chopper blade that is closed-loop controlled and provides a uniform clockwise rotation. The rotation rate n can be preset precisely and the value of the circumferential velocity ω as function of the radius r can be determined by $\omega(r) = 2\pi \times r \times n$. For the validation experiment, the scaling factor was determined with the procedure described above, and the frame rate was set to $f_r = 240$ Hz which results in displacements in the order of 10 pixel for the current optical magnification. For the optical flow method (OF), a maximum number of 500 features were chosen, and for the cross-correlation analysis, the window size was set to 64×64 pixel with a sample offset of 0.85. This results in an overlap of 15% or 10 pixel. The vector field in the region of interest for both methods as stored by the app is shown in Fig. 9 in the upper part with OF on the left and CC on the right side of the figure.

As can be seen, the optical flow method shows randomly distributed vectors attached to features with strong contrast that indicate the clockwise rotation. In that case also the outer rim of the disk shows velocity vectors. The underlying grid for the

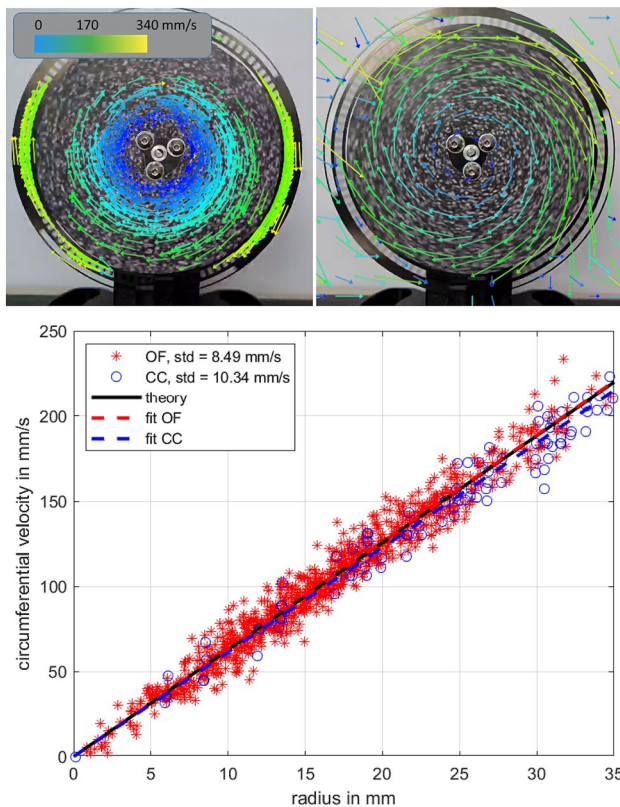


Fig. 9 Validation experiment using a rotating disk with a printed particle pattern. Evaluated data with the optical flow algorithm for the Android app (top, left) and cross-correlation analysis (top, right). Circumferential velocity over the radius for both methods and the theoretical value. The indicated standard deviation is the standard deviation of the difference between measured and theoretical velocity

cross-correlation analysis was not shown in order to improve the visibility of the vectors. The clockwise rotation is also clearly visible for the cross-correlation analysis on the right. However, since no outlier filter is applied also the erroneous vectors as a result of correlation of image noise in regions without movement can be seen. In order to assess the errors of the respective methods, the center of the rotating disk was determined using image processing to a accuracy of a pixel, and the circumferential velocity was plotted over the radius as can be seen in the lower part of Fig. 9. The scatter plot of the estimated vectors follows the theoretical value well. For both methods, the velocity data were fitted with a line through the origin. Both fits show good agreement with the theoretical values, the fit for the optical flow analysis almost perfectly matches the theoretical profile.

To quantify the scattering, the standard deviation of the difference between measured velocity and theoretical value was calculated to be 10.34 mm/s for the cross-correlation analysis and 8.49 mm/s for the optical flow method. The relative mean absolute deviation from the theoretical value is 8.0% for optical flow and 7.4% for the cross-correlation in the current case.

It has to be mentioned that no special precautions were taken to ensure that the image plane and the plane of the rotating particle pattern were perfectly parallel. The smartphone was adjusted with the naked eye as might be typical for a lab session or an experiment in the field. In addition, it is not known for smartphones if the image sensor is placed in parallel to any accessible edge of its frame that can be used for adjustment. For this reason, the experiment was repeated multiple times with completely removing and rearranging the smartphone. The uncertainty estimates were always in the same order.

4.4 Summary validation

To sum up, the implemented algorithms show a maximum deviation of 0.04 pixel for the displacement determination using ideal noise-free images. Furthermore, no systematic errors can be seen using images acquired by the smartphone camera in agreement to the previous study (Cierpka et al. 2016). The relative mean absolute deviation from the theoretical known velocity was 8.0 and 7.4% for optical flow and the cross-correlation, respectively, which includes all uncertainties from the whole measurement chain (calibration, adjustment, changes in illumination, printed pattern, black dots in comparison to bright particle images, etc.).

5 Typical lab course setup

A typical setup for educational purposes may consist of a cw-laser for illumination. For the current example, the flow past a cylinder was chosen. A cylinder with $d = 0.8$ cm in diameter was installed in a 5×5 cm² cross-sectional water channel that introduces unsteady oscillating vortex motion in the wake flow. The blockage is 16% which results in small deviations of the flow in comparison to a cylinder in free flow. However, the experiment is mainly used to introduce basic concepts in fluid dynamics, and thus, the influence can be considered as negligible. The mean flow velocity was set to about $\bar{u} = 0.18$ m/s which results in a Reynolds number of roughly $Re \approx 1500$. For illumination, a cw-laser diode (Z-Laser GmbH, 40 mW) with a wavelength of 532 nm was chosen. Polyamid particles with a diameter of $d_p = 20$ μ m and a density of $\rho_p = 1150$ kg/m³ were used as tracer particles. Since the resulting Stokes number of $St \approx 6 \times 10^{-4}$ was less than 10^{-1} , these particles can follow the flow with high fidelity (Raffel et al. 2018). A photograph of the setup is shown in Fig. 10. The smartphone in front of the field of view can clearly be seen. The vectors in the insert already indicate the flow direction from left to right. The light sheet was adjusted to have the highest intensity close to the cylinder and is shining through the transparent test section from the top. It has to be mentioned, that the current setup should be more secured for the use in practical sessions with many

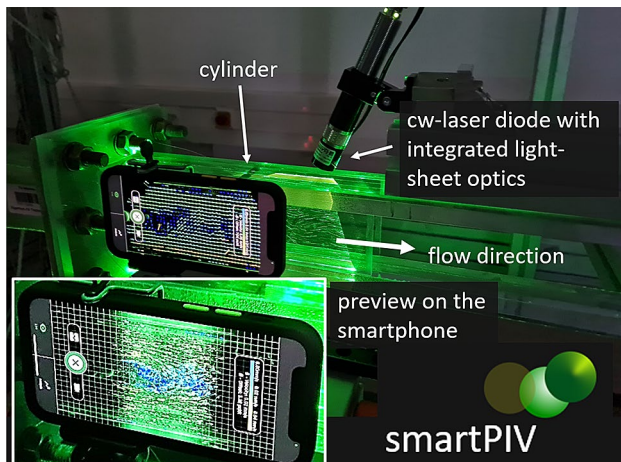


Fig. 10 Typical educational setup using a cw-laser for the light sheet generation for the illumination from the top in a small water channel. The insert shows the live preview of the flow past a cylinder

users/students using non-combustible curtain to cover the laser light. Moreover, LED-based illumination can replace the laser light since it has only little disadvantages, for instance in light sheet thickness, compared to laser light sheets which is not of great importance for laboratory classes or rough field estimates.

Figure 11 shows an example for the stored images for the measurement of the flow around the cylinder for $Re \approx 1500$ with $f_r = 240$ Hz for optical flow (top) and cross-correlation analysis (bottom). It can clearly be seen that the flow is coming from left to right and is much faster above and below the cylinder wake. The wake flow with vortices is also visible. In this case, no calibration target was introduced in the water tunnel, and therefore, no specific scaling was set. The calculation of the averaged velocity in the snapshots is therefore not correct in physical units. However, the exported data were later scaled by the known size of the cross section. The scaling factor was determined to be 13.5 pixel per mm, and the velocity vectors in physical scale are shown in Fig. 12. The mean displacement per Δt indicates a maximum shift of around 10 pixel between two successive frames and is in the same order of magnitude for both methods. This means that also for the optical flow a representative distribution of features are detected in the field of view.

In the top of Fig. 12, the velocity vector field that was exported by the app and evaluated with optical flow is shown. For the contour plot, the data were interpolated on the same grid as used for the cross-correlation analysis. The instantaneous vectors are shown (5 times enlargement for display) in blue. At the upper and bottom wall, the boundary layers can clearly be distinguished. Also, the higher velocities close to the cylinder can be clearly seen. Moreover, the vortices in the wake are visible, and one may even indicate vortices of alternating vorticity for larger downstream

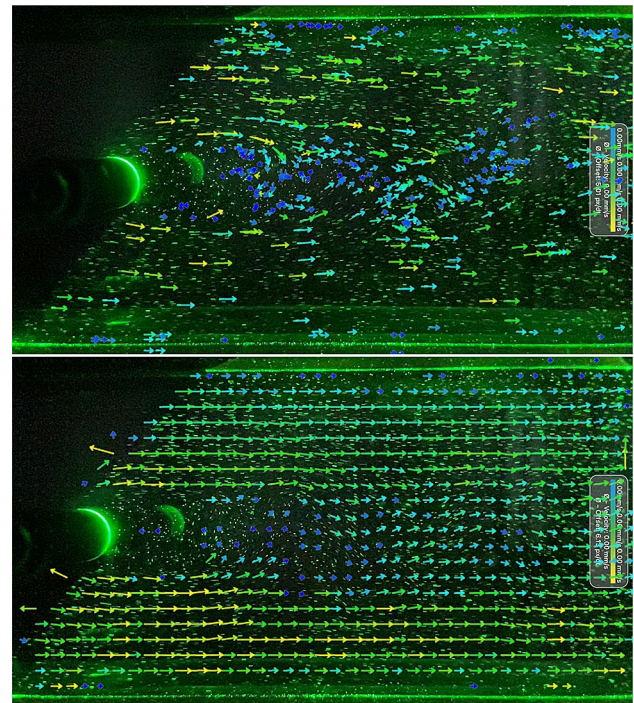


Fig. 11 Saved images with vector overlay for the flow past a cylinder for $Re \approx 1500$. Optical flow (top) and cross-correlation analysis (bottom)

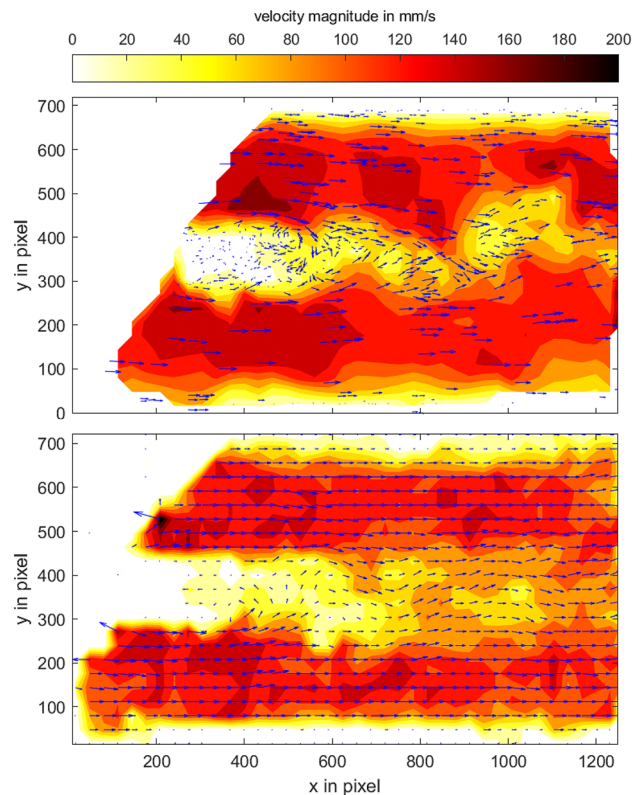


Fig. 12 Extracted and scaled velocity vectors for the flow past a cylinder for $Re \approx 1500$. Optical flow (top) and cross-correlation analysis (bottom)

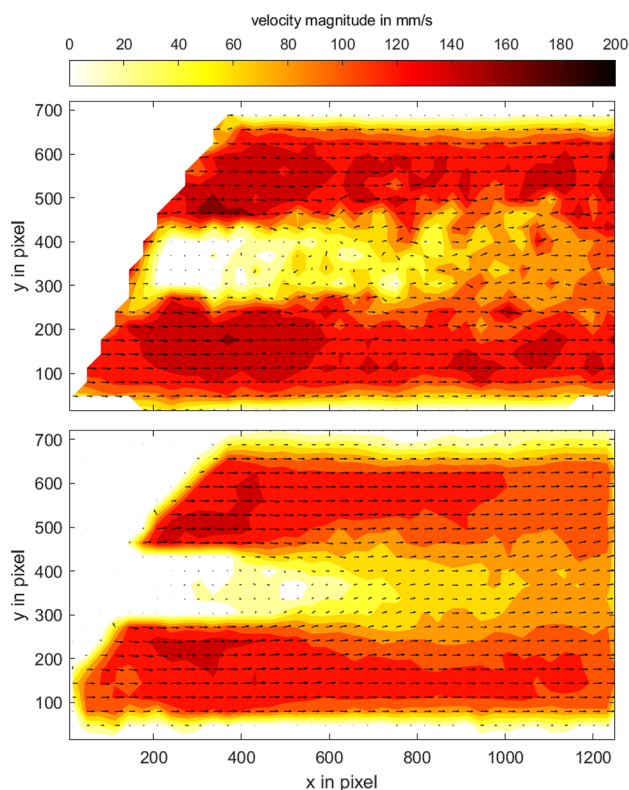


Fig. 13 Averaged extracted and scaled velocity vectors for the flow past a cylinder for $Re \approx 1500$. Optical flow (top) and cross-correlation analysis (bottom)

distances. It has to be mentioned, that even if the results look similar, the optical flow method is not the same like particle tracking velocimetry. Whereas in particle tracking velocimetry, the center positions of identified particles are detected, and later, algorithms are used to find the corresponding particle positions in the next frame (see for example Cierpka et al. 2013), optical flow works on the images and tracks 'features' (high intensity gradients) between successive frames. However, the resulting vectors are also randomly distributed in the measurement plane as also known from PTV. For comparisons among the different algorithms (PIV, PTV, OF) for experimental velocity measurements, the interested reader is referred to the latest PIV challenge (Kähler et al. 2016).

In the lower part of Fig. 12, the velocity vector field evaluated with cross-correlation with a window size of 64×64 pixel and a sample offset of 0.5 is presented. Due to the larger window sizes, the velocity is somewhat underestimated in regions of high velocity gradients in comparison to the optical flow method. In general, this could be improved by using smaller interrogation windows, but in the current case the amount of spurious vectors became to high using a window size of 32×32 pixel. However, all the features described above can also be seen in the snapshot.

In Fig. 13, the averages from ten successive evaluations that were exported by the app for the respective processing scheme are shown. For the cross-correlation analysis, the mean of all ten fields is plotted, whereas, for the evaluation with optical flow, the data from all ten vector fields were interpolated onto the same grid as used for the cross-correlation analysis for better comparison. For both fields, the averaged vectors on the same grid are shown (3 times enlargement for display) in black. The velocity distribution matches quite well as was expected from the validation experiment. Obviously, a representative number of features were tracked by the optical flow analysis and the underestimation of the velocity in high gradient regions is not severe for the cross-correlation analysis as one interrogation window corresponds to $4.7 \times 4.7 \text{ mm}^2$ with a vector spacing of 2.4 mm in each direction.

6 Summary and outlook

Particle image velocimetry was successfully implemented as an intuitively useable smartphone software application (app) offering live cross-correlation analysis and optical flow. This lowers significantly the costs for universities and allows to perform practical sessions using PIV. Furthermore, preconceptions that PIV is a complex and difficult to apply technique are reduced. This may help to increase the number of applications in industries once the students that are familiar with PIV leave the university. In times where the pandemic demands a minimum of shared laboratory equipment the use of the own smartphone still allows experiments and lowers the risk of spreading the virus. The app can also be used in industrial situations or in the field, where a rough estimate of the flow velocity is of interest.

The system allows for individual measurements of flow velocities by users. An estimate for the uncertainty was determined by an experiment of a rotating disk, which shows that the relative mean absolute deviation was in the order of 8% from the theoretical value. An experiment of the flow around a cylinder using 240 Hz frame rate showed that the main features of the flow could be resolved properly. The vector fields can be exported in different formats, and later, post-processing strategies can be tested as homework for students to deepen the understanding of the various methods. Furthermore, videos can be captured and later be processed in order to determine the effects of interrogation window sizes and compare optical flow and cross-correlation methods for further education.

In the future, also different post-processing filters will be implemented to be able to exclude outliers for a better overview. In addition the multi-grid analysis will be implemented to increase the dynamic velocity range. It is also planned to add more intuitive forms of distance calibration that have

previously been studied by our groups (Hofmann et al. 2019) or even use automatic position detection of the smartphones GPS sensors. In the future, it might be also possible to use the very powerful LEDs of smartphones to set up a light sheet illumination. Alternatively, so-called smart lightning devices that can be controlled via bluetooth with a smartphone might be used. These devices could be already arranged in a stripe of LEDs, providing a sheet-like illumination as desired for PIV. Two smartphones might then be enough for a rough velocity estimate in the field. Ideas for further development might also be the connection of more than one smartphone to allow stereoscopic or tomographic measurements. Furthermore, the app will be continuously updated based on experiences of the users and lab instructors.

7 Download

The app is free and was tested on many modern smartphones. It can be downloaded at in the corresponding app stores for Android,⁴ for iOS,⁵ and for Harmony OS.⁶ The authors would be happy for feedback that helps to improve the app.

Acknowledgements The authors acknowledge financial support by the Thüringer Ministerium für Wirtschaft, Wissenschaft und Digitale Gesellschaft and the Carl Zeiss Foundation's Grant: DeepTurb. The authors are grateful to the student co-workers and testers J. Bruischütz, T. Bravo Roger, J. Stephan, J. Hiese, M. Orban, M. John, C. Engelhardt, T. Käufer and to M. Sharifi Gazjahani, A. Thieme and J. König for technical support.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

⁴ https://play.google.com/store/apps/details?id=de.tu_ilmenu.secsy.smartpiv.

⁵ <https://apps.apple.com/app/smartpiv/id1471308387>.

⁶ <https://appgallery.cloud.huawei.com/#/app/C103407577>.

References

- Adrian R, Westerweel J (2011) Particle image velocimetry. Cambridge University Press, New York
- Aguirre-Pablo AA, Alarfaj MK, Li EQ, Hernández-Sánchez JF, Thoroddsen ST (2017) Tomographic particle image velocimetry using smartphones and colored shadows. *Sci Rep* 7:3714. <https://doi.org/10.1038/s41598-017-03722-9>
- Chen H, Muros-Cobos JL, Holgado-Terriza JA, Amirfazli A (2017) Surface tension measurement with a smartphone using a pendant drop. *Colloids Surf A* 533:213–217. <https://doi.org/10.1016/j.colsurfa.2017.08.019>
- Cierpka C, Lütke B, Kähler CJ (2013) Higher order multi-frame particle tracking velocimetry. *Exp Fluids* 54:1533. <https://doi.org/10.1007/s00348-013-1533-3>
- Cierpka C, Hain R, Buchmann NA (2016) Flow visualization by mobile phone cameras. *Exp Fluids* 57:108. <https://doi.org/10.1007/s00348-016-2192-y>
- Google (2021) Google Flutter. www.flutter.dev
- Goy NA, Denis Z, Lavaud M, Grolleau A, Dufour N, Deblais A, Delabre U (2017) Surface tension measurements with a smartphone. *Phys Teacher* 55:498–499. <https://doi.org/10.1119/1.5008349>
- Hofmann M, Seeland M, Mäder P (2019) Efficiently annotating object images with absolute size information using mobile devices. *Int J Comput Vis* 127:207–224. <https://doi.org/10.1007/s11263-018-1093-3>
- Kähler CJ, Astarita T, Vlachos PP, Sakakibara J, Hain R, Discetti S, La Foy R, Cierpka C (2016) Main results of the 4th international PIV challenge. *Exp Fluids* 57(6):97. <https://doi.org/10.1007/s00348-016-2173-1>
- Kashyap V, Kumar S, Jajal NA, Mathur M, Singh RK (2020) Parametric analysis of smartphone camera for a low cost particle image velocimetry system. *ArXiv preprint* 2002:01061
- Käufer T, König J, Cierpka C (2020) Stereoscopic PIV measurements using low-cost action cameras. *Exp Fluids* 62:57. <https://doi.org/10.1007/s00348-020-03110-6>
- Klein P, Kuhn J, Müller A (2016) Experimente mit Smartphone und Tablet-PC: Analyse leistungsbezogener Antwortsicherheiten im Physikstudium. Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-13809-7_14
- Mäder P, Boho D, Rzanny M, Seeland M, Wittich HC, Deggelmann A, Wäldchen J (2021) The flora incognita app—interactive plant species identification. *Methods Ecol Evol*. <https://doi.org/10.1111/2041-210X.13611>
- Miller VA, Loebner KT (2016) Smartphone Schlieren. *ArXiv:1609.04298*
- Minichiello A, Armijo D, Mukherjee S, Caldwell L, Kulyukin V, Truscott T, Elliott J, Bhouraskar A (2020) Developing a mobile application-based particle image velocimetry tool for enhanced teaching and learning in fluid mechanics: A design-based research approach. *Comput Appl Eng Educ*. <https://doi.org/10.1002/cae.22290>
- Raffel M, Willert CE, Scarano F, Kähler CJ, Wereley ST, Kompenhans J (2018) Particle image velocimetry: a practical guide, 3rd edn. Springer, Berlin. <https://doi.org/10.1007/978-3-319-68852-7>
- Settles GS (2018) Smartphone schlieren and shadowgraph imaging. *Opt Lasers Eng* 104:9–21. <https://doi.org/10.1016/j.optlaseng.2017.07.002>
- Solomon DE, Abdel-Raziq A, Vanapalli SA (2016) A stress-controlled microfluidic shear viscometer based on smartphone imaging. *Rheol Acta* 55:727–738. <https://doi.org/10.1007/s00397-016-0940-9>
- Staacks S, Hütz S, Heinke H, Stampfer C (2018) Advanced tools for smartphone-based experiments: phyphox. *Phys Educ* 53(045):009. <https://doi.org/10.1088/1361-6552/aac05e>

- Wäldchen J, Mäder P (2018) Machine learning for image based species identification. *Methods Ecol Evol* 9:2216–2225. <https://doi.org/10.1111/2041-210X.13075>
- Willert CE, Gharib M (1991) Digital particle image velocimetry. *Exp Fluids* 10:181–193. <https://doi.org/10.1007/BF00190388>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.