Digital particle image velocimetry (DPIV) is a non-intrusive analysis technique that is very popular for mapping flows quantitatively. To get accurate results, in particular in complex flow fields, a number of challenges have to be faced and solved: The quality of the flow measurements is affected by computational details such as image pre-conditioning, sub-pixel peak estimators, data validation procedures, interpolation algorithms and smoothing methods. The accuracy of several algorithms was determined and the best performing methods were implemented in a user-friendly, GUI based open-source tool for performing DPIV flow analysis in MATLAB.

**Keywords:** particle image velocimetry; PIV; MATLAB; fluid dynamics; flow analysis; direct cross correlation; discrete Fourier transform; sub-pixel peak finding; window deformation; interpolation; histogram equalization
see Figure 2). All of these steps are accessible from the 
GUI of PIVlab. The workflow is menu-based, starting at 
the left with image input and pre-processing options, and 
then continuing to the right of the menu (image evalua-
tion / PIV analysis, post-processing, data exploration). This 
workflow is demonstrated in tutorials and screen capture 
videos that can be found on the project website.

The following section will give an overview of the rele-
vant features and techniques that are accessible in PIVlab:

**Image pre-processing**

One common approach to improve the measurement 
quality is the enhancement of images before the actual 
image correlation takes place [8, 9]. This section presents 
a selected number of pre-processing techniques that are 
implemented in PIVlab (see Figure 3 for examples).

**Histogram equalization**

Contrast limited adaptive histogram equalization (CLAHE) 
was developed to increase the readability of image data in 
medical imaging [10]. CLAHE operates on small regions 
(tiles) of the image: In every tile, the most frequent inten-
sities of the image histogram are spread out to the full 
range of the data (from 0 to 255 in 8-bit images). Regions 
with low exposure and regions with high exposure are 
therefore optimized independently. CLAHE significantly 
improves the probability of detecting valid vectors in 
experimental images by 4.7 ± 3.2% [9].

**Intensity highpass**

Inhomogeneous lighting can cause low frequency back-
ground information which can be removed by applying a 
high-pass filter that mostly conserves the high frequency 
information from the particle illumination [11]. The fil-
ter emphasizes the particle information in the image, and 
suppresses any low frequency information in the images 
(including all low frequency displacement information).

**Intensity capping**

The DPIV method assumes that all particles within an 
interrogation window have the same motion. This will 
not be the case in reality, as perfectly uniform flow does 
hardly exist. Bright particles or bright spots within the 
area will contribute statistically more to the correlation 
signal, which may bias the result in non-uniform flows [9]. 
The intensity capping filter circumvents this problem. An
This statistical technique is implemented with the discrete correlation function in the frequency domain (discrete Fourier transform, DFT). The direct cross correlation computes the correlation matrix in the spatial domain. In DCC, the interrogation areas A and B can have two different sizes [14]. When B is chosen twice as large as A, a particle displacement of up to half the size of A will not result in any loss of information and provide a reliable correlation matrix with low background noise (see Figure 4, top middle and Figure 5, top). DCC has been shown to create more accurate results than a standard DFT approach [12]. The disadvantage of DCC is the increased computational cost with respect to a standard DFT approach, especially with large interrogation areas [8, 12, 15].

**Image evaluation**

The most sensitive part of a DPIV analysis is the cross correlation algorithm: Small sub images (interrogation areas) of an image pair are cross correlated to derive the most probable particle displacement in the interrogation areas. In essence, the cross-correlation is a statistical pattern matching technique that tries to find the particle pattern from interrogation area A back in interrogation area B. This statistical technique is implemented with the discrete cross correlation function [12]:

\[
C(m,n) = \sum_{i} \sum_{j} A(i, j) B(i - m, j - n)
\]

where A and B are corresponding interrogation areas from image A and image B.

The location of the intensity peak in the resulting correlation matrix C gives the most probable displacement of the particles from A to B [12].

There are two common approaches to solve equation 1: The most straightforward approach is to compute the correlation matrix in the spatial domain (see Figure 4 for a graphical representation of this correlation). This approach is either called direct cross correlation [13], particle image pattern matching [12], or convolution filtering [14].

Another approach is to compute the correlation matrix in the frequency domain (discrete Fourier transform, DFT). The DFT is calculated using a fast Fourier transform [15]. Both approaches are implemented in PIVlab as they both have advantages as well as some drawbacks; these will be presented in short in the next sections. More details on the mathematical background of cross correlation can be found elsewhere (e.g. [8, 16]).

**Direct cross correlation (DCC)**

**Discrete Fourier transform (DFT) and advanced DFT techniques**

The potential drawback of DCC – the computational cost – can be resolved by calculating the correlation matrix in the frequency domain [8] using FFT (see Figure 6A). This approach uses interrogation areas of identical size; therefore every particle displacement induces some loss of information, which can be noticed by the increasing amount of background noise in the correlation matrix (see Figure 5, bottom). This background noise complicates the detection of the intensity peak and decreases accuracy. It is therefore advisable to reduce the displacement to about one quarter of the interrogation area, in order to keep the background noise in the correlation matrix low [1].

This disadvantage can be offset by running several passes of the DFT on the same dataset [18]: The integer result of the first analysis pass is used to offset the interrogation area in the following passes. The loss of information due to particle displacement is hence minimized. The interrogation grid can be refined with every pass [19], yielding a high spatial resolution in the final vector map, together with a high dynamic velocity range and an optimal signal to noise ratio.

In real flows, the particle patterns will additionally be sheared and rotated; the non uniform particle motion will broaden the intensity peak in the correlation matrix and deteriorate the result. Several methods that account for the transformation of the interrogation areas have been proposed [20–22]. In PIVlab, the following procedure is implemented: The analysis is started with a regular DFT analysis. The first pass yields displacement information at the centre of each interrogation area. When the areas overlap one another by e.g. 50%, there is additional displacement information at the borders and corners of each interrogation area (nine positions in total, see Figure 5, bottom). This background noise complicates the calculation of displacement information at every pixel of the interrogation areas via bilinear interpolation. Next, interrogation area B is deformed according to this displacement information (see Figure 6B, right) using either bilinear interpolation (faster) or spline interpolation (higher precision, but slower). The next interrogation pass correlates the original interrogation area A with the deformed area B. The remaining displacement information of each pass is accumulated. After a few passes, the displacement has been determined with high accuracy. Between the passes, but not after the final pass.
pass, the velocity information is smoothed and validated and missing information is interpolated. Data validation can be relatively strict, as any deteriorating effect of interpolation and smoothing will be corrected in the correlation of the following pass.

**Peak finding**

The choice of the peak finding technique is similar to the choice of the cross correlation technique — another important factor for the accuracy of DPIV. The integer displacement of two interrogation areas can be determined straightforward from the location of the intensity peak of the correlation matrix. The location can be refined with sub-pixel precision using a range of methods [8, 23, 24]. The standard procedure is to fit a Gaussian function to the integer intensity distribution (see Figure 7). It is sufficient to use only the directly adjacent vertical and horizontal pixels (two times a 3-point fit = 2·3-point fit) and to evaluate the x and y axis separately. The peak of the fitted function is used to determine the particle displacement with sub-pixel precision.

If the particle displacement within an interrogation area is exposed to shear or rotation or if the images suffer from excessive motion blur, the displacement peak may have an elliptical shape [8]. In this case, a two-dimensional Gaussian function (9-point fit) has a better performance [25]. The added value of using a two-dimensional Gaussian function is more pronounced in non-deforming methods, such as DCC and single pass DFT. Both peak finding algorithms are implemented in PIVlab.

**Post-processing**

**Data validation**

Post processing of DPIV data is generally required to obtain reliable results [26]. A basic method to filter outliers in PIVlab is to choose limits for acceptable velocities manually. Velocity thresholds can also be determined semi automatically by comparing each velocity component with a lower threshold and an upper threshold ($t_{\text{lower}}$ and $t_{\text{upper}}$):

$$t_{\text{lower}} = \bar{u} - n \cdot \sigma_u$$  \hspace{1cm} (2)

$$t_{\text{upper}} = \bar{u} + n \cdot \sigma_u$$  \hspace{1cm} (3)

where $\bar{u}$ = mean velocity; $\sigma_u$ = standard deviation of $u$.

The user defined value of $n$ determines the strictness of this filter. This filter works very well in practice, as it adapts to some extent to the nature of the flow.

A more universal outlier detection method that automatically adapts to local flow situations is the normalized median test [27] (or local median filter). The filter evaluates...
Functional testing has been performed on Windows XP. Quality control gral quantities can be calculated comfortably. translated, data can be extracted from paths or areas and inte-
ber of possibilities to further process and distil the results: maps. The strength of PIVlab is that it offers a large num-
Such a complexity is hard to describe purely with vector
advanced smoothing algorithms are based on a penalized
method to smooth DPIV data is median filtering [8]. More
by applying data smoothing. Raffel et al. [8] propose to
replaced by interpolated data [26]. One common tech-
neighbourhood (3 3 mean) interpolation. The approach provides an interpolation that is generally fairly smooth, and over larger regions with missing data, it will tend towards the average of the boundary velocities, which prevents overshooting.
Data interpolation
After the removal of outliers, missing vectors should be replaced by interpolated data [26]. One common techn-
tility is the 3 3 neighbourhood (3 3 mean) interpolation. Two-dimensional linear or spline interpolation [5] are other alternatives. PIVlab uses a boundary value solver for interpolation. The approach provides an interpolation that is generally fairly smooth, and over larger regions with missing data, it will tend towards the average of the boundary velocities, which prevents overshooting.

Data smoothing
A certain amount of measurement noise will be inevita-
Standard DPIV (see Figure 8, left). An increasing amount of random vectors (0% to 15%) is removed from the resulting vector matrix (see Figure 8, middle). The miss-
ing data is interpolated using one of the interpolators (see Figure 8, right), and finally, the mean absolute differ-
ence between the original data and the interpolated data is determined. This whole procedure is repeated 1000 times for each image pair and each level of missing data. Under challenging conditions and large amounts of missing data, the boundary value solver which is implemented in PIVlab performs best (see Figure 9). More detail is given in [29].

**Accuracy of the DPIV analyses**
The quality of the DPIV measurements in PIVlab was extensively evaluated using synthetic particle images with known properties. The effect of particle image diameter, particle density, sensor noise, particle pair loss, motion blur and shear was determined and is reported in detail elsewhere [29]. These quality tests revealed that DFT using window deformation outperforms the basic DCC and DFT correlation, especially under challenging conditions. The additional computational load is compensated by the increased robustness and accuracy of the algo-
method to smooth DPIV data is median filtering [8]. More
advanced smoothing algorithms are based on a penalized least squares method [28] – the latter technique is implement-
ed in PIVlab.

Data exploration
Many DPIV studies reveal very complex flow patterns. Such a complexity is hard to describe purely with vector maps. The strength of PIVlab is that it offers a large num-
ber of possibilities to further process and distil the results: Derivatives, such as vorticity and divergence can be calcu-
calculated, data can be extracted from paths or areas and inte-
gal quantities can be calculated comfortably.

Quality control
Functional testing has been performed on Windows XP and Windows 7 with MATLAB releases R2010a, R2011a and R2013b. Although the authors do not have access to additional operation systems and MATLAB versions, the reports from many users claim that PIVlab works flaw-
lessly on Mac OS X and UNIX/Linux too. Bugs that have 
been discovered in some of the last ten PIVlab releases have been corrected before the next release. The user acceptance of PIVlab has been monitored and optimized while the software was used by more than 150 supervised students with several different operating systems and MATLAB releases.

Furthermore, extensive tests on the quality of the results obtained with PIVlab were performed using more than 6 10⁴ synthetic particle images. The image properties were modified in a way that allowed to cal-
culate the accuracy of all important PIV image param-
eters according to [8]. Detailed results of these tests are reported in [29]. Furthermore, the PIVlab tool-
box contains a script that performs a fully automatic analysis of the accuracy of the DPIV analysis: The script 'Accuracy.m' will generate random PIV images, evaluate these images, and perform a comparison between the calculated and the real displacement of the inter-
rogation areas. This script helps users to make sure that PIVlab works accurately on their combination of MATLAB and operating system.

**Interpolation performance**
The performance of the most popular interpolation tech-
iques and the boundary value solver are tested: A num-
ber of real and synthetic image pairs are analyzed using standard DPIV (see Figure 8, left). An increasing amount of random vectors (0% to 15%) is removed from the result-
ing vector matrix (see Figure 8, middle). The missing 
data is interpolated using one of the interpolators (see Figure 8, right), and finally, the mean absolute differ-
ence between the original data and the interpolated data is determined. This whole procedure is repeated 1000 times for each image pair and each level of missing data. Under challenging conditions and large amounts of missing data, the boundary value solver which is implemented in PIVlab performs best (see Figure 9). More detail is given in [29].

**Figure 8:** Performance of the most popular interpolation tech-
iques and the boundary value solver are tested: A num-
ber of real and synthetic image pairs are analyzed using standard DPIV (see Figure 8, left). An increasing amount of random vectors (0% to 15%) is removed from the result-
ing vector matrix (see Figure 8, middle). The missing 
data is interpolated using one of the interpolators (see Figure 8, right), and finally, the mean absolute differ-
ence between the original data and the interpolated data is determined. This whole procedure is repeated 1000 times for each image pair and each level of missing data. Under challenging conditions and large amounts of missing data, the boundary value solver which is implemented in PIVlab performs best (see Figure 9). More detail is given in [29].

**Figure 7:** Principle of the Gaussian 2 3-point fit: Sub-
pixel precision is achieved by fitting a one-dimensional Gaussian function (solid line) to the integer intensity distribution of the correlation matrix (dots) for both axes independently (only one axis is shown here).
Smoothing performance
The performance of the smoothing algorithms is tested using DPIV data of synthetic particle images (more detail is given in [29]). Data smoothing – no matter what algorithm is used – always decreases the amount of noise introduced by the DPIV algorithm and hence increases the quality of the velocity estimation (see Figure 10). The application of the ‘Smoothn’ algorithm [28] reduces the difference between analysis and true velocities. It was therefore decided to implement this algorithm in PIVlab in order to further enhance the accuracy of the velocity estimation.

Support
Support is available through the website of PIVlab (http://PIVlab.blogspot.com), where questions can be posted and generally receive quick responses from the authors. The email address for support requests is also noted on this website. It furthermore contains news, tips, updates and tutorials for PIVlab users.

(2) Availability
Operating system
Based on MATLAB 7.10.0 (R2010a): Windows, UNIX/Linux, Macintosh

Programming language
MATLAB 7.10.0 (R2010a), upward compatible

Additional system requirements
MATLAB 7.10.0 (R2010a): 1 GB disk space, 1 GB RAM

Dependencies
MATLABs Image Processing Toolbox is required

List of contributors
WT: Programming of the tool and the graphical user interface (GUI); selection, implementation and evaluation of the algorithms; writing this article; support and maintenance of the tool.
EJS: Support with the layout of both the tool and the GUI; feedback, tips and discussions on the functionality.

Archive
Name Figshare.com
Persistent identifier http://dx.doi.org/10.6084/m9.figshare.1092508
Licence CC-BY
Publisher William Thielicke
Date published 03/07/2014
Language English

(3) Reuse potential
The analysis of flow velocities is an integral part of many research disciplines. Here, DPIV has become a very popular method. The development of user-friendly, accurate and free DPIV tools greatly enhances the practicability and availability of this technique. In many cases, it can even be sufficient to use low cost digital cameras and lasers to study complex flows with PIVlab (e.g. [30]). The simple graphical user interface of PIVlab enables users that are new to the field of DPIV to generate accurate results quickly.

PIVlab has proved its value in a number of research projects. The tool has been used for ‘conventional’ measurements of fluid velocities, but also in a very different context, for example flow visualizations within cells, echocardiographic velocimetry of the human heart and deformation of sand and gravel [29, 31–54].

PIVlab can be extended with custom functionalities and features using MATLAB’s GUI editor and many of MATLAB’s pre-built functions. It is e.g. the basis of ‘PTVlab’, an independent software tool that was designed to track particles. Individual functions of PIVlab (e.g. the image correlation code) can be used for custom projects.

PIVlab is highly integrated with MATLAB and benefits from MATLAB’s extensive plotting and data handling features. But data can also be exported to generic ASCII files, which can be processed e.g. with Excel (Microsoft, Redmond, WA). As PIVlab can also export to binary vtk files, Paraview (Kitware, Inc., Clifton Park, NY) is another very appropriate possibility to visualize and explore the flow data.

Acknowledgements
See http://www.mathworks.com/matlabcentral/fileexchange/27659

Notes
1 inpaint_nans by John D’Errico, http://www.mathworks.com/matlabcentral/fileexchange/4551-inpaintnans
2 The accuracy tests are part of the PIVlab documentation and they are made available on the website of our tool (http://PIVlab.blogspot.com)

References
1. Keane, R. D. & Adrian, R. J. 1990 Optimization of particle image velocimeters. i. double pulsed systems Measurement Science and Technology 1, 1202–1215. DOI: http://dx.doi.org/10.1088/0957-0233/1/11/013
2. Adrian, R. J. 1991 Particle-imaging techniques for experimental fluid mechanics Annual Review of Fluid Mechanics 23, 261–304. DOI: http://dx.doi.org/10.1146/annurev.fl.23.010191.001401
3. Willert, C. E. & Gharib, M. 1991 Digital particle image velocimetry Experiments in Fluids 10, 181–193. DOI: http://dx.doi.org/10.1007/BF00190388
4. Buchhave, P. 1992 Particle image velocimetry - status and trends Experimental Thermal and Fluid Science 5, 586–604. DOI: http://dx.doi.org/10.1016/0994-1777(92)90016-X

5. Stamhuis, E. J. & Videler, J. J. 1995 Quantitative flow analysis around aquatic animals using laser sheet particle image velocimetry Journal of Experimental Biology 198, 283–294.

6. Willert, C. 1996 The fully digital evaluation of photographic piv recordings Applied Scientific Research 56, 79–102. DOI: http://dx.doi.org/10.1007/BF02249375

7. Grant, I. 1997 Particle image velocimetry: A review Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 211, 55–76. DOI: http://dx.doi.org/10.1243/0954406971521665

8. Raffel, M., Willert, C., Wereley, S. & Kompenhans, J. 2007 Particle Image Velocimetry-Springer 2 edn.

9. Shavit, U., Lowe, R. & Steinbuck, J. 2007 Intensity capping: a simple method to improve cross-correlation piv results Experiments in Fluids 42, 225–240. DOI: http://dx.doi.org/10.1007/s00348-006-0233-7

10. Pizer, S. M., Amburn, E. P., Austin, J. D., Cromartie, R., Geselowitz, A., Greer, T., ter Haar, Romeny, B., Zimmerman, J. B. & Zuiderveld, K. 1987 Adaptive histogram equalization and its variations Computer Vision, Graphics, and Image Processing 39, 355–368 DOI: http://dx.doi.org/10.1016/S0734-189X(87)80186-X

11. Gonzalez, R. C. & Wintz, P. 1987 Digital image processing Addison-Wesley Publishing Company, Inc.

12. Huang, H., Dabiri, D. & Gharib, M. 1997 On errors of digital particle image velocimetry Measurement Science and Technology 8, 1427–1440. DOI: http://dx.doi.org/10.1088/0957-0233/8/12/007

13. Okamoto, K., Nishio, S., Saga, T. & Kobayashi, T. 2000 Standard images for particle-image velocimetry Measurement Science and Technology 11, 685–691. DOI: http://dx.doi.org/10.1088/0957-0233/11/6/311

14. Stamhuis, E. J. 2006 Basics and principles of particle image velocimetry (piv) for mapping biogenic and biologically relevant flows Aquatic Ecology 40, 463–479. DOI: http://dx.doi.org/10.1007/s10452-005-0657-z

15. Soria, J. 1996 An investigation of the near wake of a circular cylinder using a video-based digital cross-correlation particle image velocimetry technique Experimental Thermal and Fluid Science 12, 221–233. DOI: http://dx.doi.org/10.1016/0894-1777(95)00086-0

16. Keane, R. & Adrian, R. 1992 Theory of cross-correlation analysis of piv images Applied Scientific Research 49, 191–215. DOI: http://dx.doi.org/10.1007/BF00384623

17. Frigo, M. & Johnson, S. G. 2005 The design and implementation of FFTW3 Proceedings of the IEEE 93, 216–231. DOI: http://dx.doi.org/10.1109/JPROC.2004.840301

18. Westerweel, J., Dabiri, D. & Gharib, M. 1997 The effect of a discrete window offset on the accuracy of cross-correlation analysis of digital piv recordings Experiments in Fluids 23, 20–28. DOI: http://dx.doi.org/10.1007/s003480050082

19. Scarano, F. & Riethmuller, M. L. 1999 Iterative multigrid approach in piv image processing with discrete window offset Experiments in Fluids 26, 513–523. DOI: http://dx.doi.org/10.1007/s003480050318

20. Huang, H., Fiedler, H. & Wang, J. 1993 Limitation and improvement of piv, part ii: Particle image distortion, a novel technique Experiments in Fluids 15, 263–273.

21. Jambunathan, K., Ju, X. Y., Dobbins, B. N. & Ashforth-Frost, S. 1995 An improved cross-correlation technique for particle image velocimetry Measurement Science and Technology 6, 507–514. DOI: http://dx.doi.org/10.1088/0957-0233/6/5/012

22. Scarano, F. & Riethmuller, M. L. 2000 Advances in iterative multigrid piv image processing Experiments in Fluids 29, S051–S060. DOI: http://dx.doi.org/10.1007/s003480070007

23. Lourenco, L. & Krothapalli, A. 1995 On the accuracy of velocity and vorticity measurements with piv Experiments in Fluids 18, 421–428. DOI: http://dx.doi.org/10.1007/BF00208464

24. Roesgen, T. 2003 Optimal subpixel interpolation in particle image velocimetry Experiments in Fluids 35, 252–256. DOI: http://dx.doi.org/10.1007/s00348-003-0627-8

25. Nobach, H. & Honkanen, M. 2005 Two-dimensional gaussian regression for sub-pixel displacement estimation in particle image velocimetry or particle position estimation in particle tracking velocimetry Experiments in Fluids 38, 511–515. DOI: http://dx.doi.org/10.1007/s00348-005-0942-3

26. Nogueira, J., Lecuona, A. & Rodriguez, P. A. 1997 Data validation, false vectors correction and derived magnitudes calculation on piv data Measurement Science and Technology 8, 1493. DOI: http://dx.doi.org/10.1088/0957-0233/8/12/012

27. Westerweel, J. & Scarano, F. 2005 Universal outlier detection for piv data Experiments in Fluids 39, 1096–1100. DOI: http://dx.doi.org/10.1007/s00348-005-0016-6

28. Garcia, D. 2010 Robust smoothing of gridded data in one and higher dimensions with missing values Computational Statistics and Data Analysis 54, 1167–1178. DOI: http://dx.doi.org/10.1016/j.csda.2009.09.020

29. Thielicke, W. 2014 The Flapping Flight of Birds – Analysis and Application Ph.D. thesis Rijksuniversiteit Groningen.

30. Ryerson, W. G. & Schwenk, K. 2012 A simple, inexpensive system for digital particle image velocimetry (dpiv) in biomechanics Journal of
Experimental Zoology 317, 127–140. DOI: http://dx.doi.org/10.1002/pez.725

31. Leong, T., Collis, J., Manasseh, R., Ooi, A., Novell, A., Bouakaz, A., Ashokkumar, M. & Kentish, S. 2011 The role of surfactant headgroup, chain length, and cavitation microstreaming on the growth of bubbles by rectified diffusion The Journal of Physical Chemistry 115, 24310–24316. DOI: http://dx.doi.org/10.1021/jp208862p

32. Booth-Gauthier, E., Alcoser, T., Yang, G. & Dahl, K. 2012 Force-induced changes in subnuclear movement and rheology Biophysical Journal 103, 2423–2431. DOI: http://dx.doi.org/10.1016/j.bpj.2012.10.039

33. Jiang, Y. J. & Tsuchita, I. 2012 Experimental study of dry granular flow and impact behavior against a rigid retaining wall Rock Mechanics and Rock Engineering 1–17.

34. Piro, V., Piro, N. & Piro, O. 2013 Experimental Zoology 317, 735–740. DOI: http://dx.doi.org/10.1007/j.oplaseng.2013.01.008

41. Datta, S. S., Chiang, H., Ramakrishnan, T. S. & Weitz, D. A. 2013 Spatial fluctuations of fluid velocities in flow through a three-dimensional porous medium Phys. Rev. Lett. 111, 064501. DOI: http://dx.doi.org/10.1103/PhysRevLett.111.064501

42. Eriksson, I., Powell, J. & Kaplan, A. 2013 Melt behavior on the keyhole front during high speed laser welding Optics and Lasers in Engineering
53. Jalalisendi, M., Pancirolli, R., Cha, Y. & Porfiri, M. 2014 A particle image velocimetry study of the flow physics generated by a thin lamina oscillating in a viscous fluid *Journal of Applied Physics* **115**, 054901. http://dx.doi.org/10.1063/1.4863721

54. Schlüßler, R., Czarske, J. & Fischer, A. 2014 Uncertainty of flow velocity measurements due to refractive index fluctuations *Optics and Lasers in Engineering* **54**, 93–104. DOI: http://dx.doi.org/10.1016/j.optlaseng.2013.10.011