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

**Figure 1:** Principle of DPIV: A laser sheet illuminates the particles contained in the fluid. A high-speed camera records the displacement of the particle pattern.

DPIVlab is programmed in MATLAB and additionally requires the image processing toolbox to run. The main file is 'PIVlab_GUI.m'. It contains all GUI-related program parts and most of the functions that are accessible from the GUI. The most important functions for performing a PIV analysis (pre-processing: 'PIVlab_preproc.m'; cross-correlation: 'piv_fftmulti.m' and 'piv_DCC.m') are also accessible from the command line (see example in 'PIVlab_commandline.m') when the user desires to automate the whole process or to include it in another application.

A DPIV analysis typically consists of three main steps: image pre-processing, image evaluation, post-processing,
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 evaluation / 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 relevant 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 intensities 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 $4.7 \pm 3.2\%$ [9].

**Intensity highpass**

Inhomogeneous lighting can cause low frequency background information which can be removed by applying a high-pass filter that mostly conserves the high frequency information from the particle illumination [11]. The filter 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.
Thielicke and Stamhuis: PIVlab – Towards User-friendly, Affordable and Accurate Digital Particle Image Velocimetry in MATLAB

Cross correlation function [12]: This statistical technique is implemented with the discrete from interrogation area A back in interrogation area B. Matching technique that tries to find the particle pattern probable particle displacement in the interrogation areas. In essence, the cross-correlation is a statistical pattern yielding a high spatial resolution in the final vector map, yielding a high spatial resolution in the final vector map, 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 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].

Figure 4: Calculation of the correlation matrix using DCC as it is performed in MATLAB. Interrogation area A (size 4x4 pixels) is correlated with interrogation area B (size 8x8 pixels) and yields the correlation matrix (size 9x9 pixels). Adapted from [8].

Upper limit of the greyscale intensity is selected, and all pixels that exceed the threshold are replaced by this upper limit. Therefore, unlike CLAHE, only a small amount of the pixel intensity information is adjusted, limiting the potential negative impact of image modifications [9]. Intensity capping improves the probability of detecting valid vectors in experimental images by 5.2 ± 2.5% [9].

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]:

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

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)

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].

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 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 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 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 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.
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 $t_{\text{upper}}$):

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

$$t_{\text{upper}} = \bar{u} + n * \sigma_u$$  \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...
Thielicke and Stamhuis: PIVlab – Towards User-friendly, Affordable and Accurate Digital Particle Image Velocimetry in MATLAB

Although the authors do not have access to Windows 7 with MATLAB releases R2010a, R2011a and R2013b. Functional testing has been performed on Windows XP. Quality control gral quantities can be calculated comfortably. Derivatives, such as vorticity and divergence can be calculated, data can be extracted from paths or areas and integrated in PIVlab. The strength of PIVlab is that it offers a large number of possibilities to further process and distill the results: Such a complexity is hard to describe purely with vector maps. The additional computational load is compensated by the increased robustness and accuracy of the algorithm. Under optimal conditions (particle image diameter ≈ 3 pixels, particle density ≈ 5 · 15 particles/window, no noise, no particle pair loss, no motion blur, no shear) the bias error of the window deformation DPIV algorithms is smaller than 0.005 pixels and the random error is below 0.02 pixels.

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 performance of the most popular interpolation techniques and the boundary value solver are tested: A number 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 resulting 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 difference 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].

Data interpolation

After the removal of outliers, missing vectors should be replaced by interpolated data [26]. One common technique 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 [8]. 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 inevitable in DPIV analyses [8]. Noise can be effectively reduced by applying data smoothing. Raffel et al. [8] propose to perform a convolution of the data with a 2-2 or 3-3 kernel with equal weights. Another common and effective 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 implemented 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 number of possibilities to further process and distill the results: Derivatives, such as vorticity and divergence can be calculated, data can be extracted from paths or areas and integral 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 flawlessly 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 · 10^4 synthetic particle images. The image properties were modified in a way that allowed to calculate the accuracy of all important PIV image parameters according to [8]. Detailed results of these tests are reported in [29]. Furthermore, the PIVlab toolbox 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 interrogation areas. This script helps users to make sure that PIVlab works accurately on their combination of MATLAB and operating system.

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).

Figure 8: Interpolation performance

Figure 9: Accuracy of the DPIV analyses
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.

Figure 8: Procedure for testing several interpolation techniques. Left: Original velocity data. Middle: Data is removed at random positions. Right: Gaps are filled with interpolation and compared to the original velocity data.

Figure 9: Performance of popular interpolators. The boundary value solver performs best under the presence of larger amounts (> 10%) of missing data. n = 1000.

(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.
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.

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].

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
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/s10452-005-6567-z
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
30. Ryerson, W. G. & Schwenk, K. 2012 A simple, inexpensive system for digital particle image velocimetry (dpiv) in biomechanics *Journal of


http://dx.doi.org/10.1063/1.4863721