

# Analysis and treatment of errors due to high velocity gradients in particle image velocimetry

P. Meunier, T. Leweke

408

**Abstract** This paper deals with errors occurring in two-dimensional cross-correlation particle image velocimetry (PIV) algorithms (with window shifting), when high velocity gradients are present. A first bias error is due to the difference between the Lagrangian displacement of a particle and the real velocity. This error is calculated theoretically as a function of the velocity gradients, and is shown to reach values up to 1 pixel if only one window is translated. However, it becomes negligible when both windows are shifted in a symmetric way. A second error source is linked to the image pattern deformation, which decreases the height of the correlation peaks. In order to reduce this effect, the windows are deformed according to the velocity gradients in an iterative process. The problem of finding a sufficiently reliable starting point for the iteration is solved by applying a Gaussian filter to the images for the first correlation. Tests of a PIV algorithm based on these techniques are performed, showing their efficiency, and allowing the determination of an optimum time separation between images for a given velocity field. An application of the new algorithm to experimental particle images containing concentrated vortices is shown.

## 1 Introduction

Over the last decade, particle image velocimetry (PIV) has become a powerful and widely used technique for measuring instantaneous velocity fields in a plane. For this, the flow is seeded with reflecting micro-particles, whose displacement over a small period is measured and used to calculate the local velocity.

The particle displacement is normally obtained by a correlation technique, either by auto-correlation of a doubly (or multiply) exposed single particle image or by cross correlation between two successive single-exposure images. In this paper, we focus on the latter technique, although some results are also valid for the former. With the two-image method, the mean displacement in a given

subpart of the images (the correlation window) is given by the location of the maximum of the cross-correlation function between the pixel intensities in the box.

With the strong increase in computer power in recent years, and the availability of high-resolution digital cameras, image acquisition and correlation computation for PIV purposes have become very easy to implement, compared with the recordings on photographic film and complex optical correlations used in the early days of PIV (see, e.g. Adrian 1991). In a majority of applications today, the PIV process is entirely digital. In this situation, where PIV has become a common tool, more efforts are now directed towards the precise analysis of the accuracy of this method in different situations, and towards reducing possible errors. In this respect, the algorithm used to obtain the velocity field from the digital particle images has received special attention. It is also the object of the present paper, i.e. we shall not consider errors linked to optical effects or the image acquisition procedure, but assume that the images correctly represent the true particle positions at the corresponding instants in time.

A number of studies (e.g. Willert and Gharib 1991; Weesterweel 1993; Fincham and Spedding 1997; Raffel et al. 1998) have analysed the errors in the velocities calculated by PIV algorithms, and their variation with different parameters: particle image density, diameter of the particles, size of the correlation window, noise in the images, and displacement amplitude. In general, the correlation methods were found to be very accurate for nearly uniform flows. However, the accuracy is drastically reduced in the presence of high velocity gradients, which are found, for example, in turbulent flows, in boundary layers or in the cores of concentrated vortices.

One error is caused by the corresponding deformation of the particle patterns between successive images, leading to lower and broader correlation peaks. A solution to this problem was proposed by Huang et al. (1993a, 1993b): deformation of the correlation windows according to the velocity gradients, in an iterative process. Jambunathan et al. (1995) generalized this method for flows with length scales smaller than the correlation box size, by deforming the window according to the displacement of each pixel of the window. The problem of instabilities, which were observed in such an iterative processes, was recently solved by Nogueira et al. (1999). These techniques of window deformation are beginning to be commonly used, and error tests have been carried out by Fincham and Delerce (2000) and Scarano and Riethmuller (2000). In all these approaches, the calculation of the first correlation poses a

Received: 8 October 2001 / Accepted: 13 June 2003  
Published online: 19 September 2003  
© Springer-Verlag 2003

P. Meunier, T. Leweke (✉)  
Institut de Recherche sur les Phénomènes Hors Equilibre,  
UMR 6594 CNRS/Universités Aix-Marseille I & II,  
49 rue Frédéric Joliot-Curie, BP 146, 13384  
Marseille Cédex 13, France  
E-mail: Thomas.Leweke@irphe.univ-mrs.fr

major difficulty, since the velocity gradients are still unknown. Lin and Perlin (1998) proposed a method of achieving this first step, through a complex algorithm that reduces the size of the windows.

The deformation of the particle patterns causes similar difficulties in particle tracking velocimetry algorithms. A discussion can be found in Ishikawa et al. (2000), who also propose a solution involving the velocity gradient tensor.

An additional problem arising at high velocity gradients has been pointed out only recently by Wereley and Meinhart (2001). In PIV, a velocity is calculated using the displacement of particles during a short time interval, i.e. one obtains an average velocity of the particle on its trajectory between the two recorded positions. If the velocity is not constant along this trajectory, which is the case when the gradients are high, this average value can be substantially different from the one at the beginning of the trajectory (the particle position in the first image), which is the point the calculated velocity is usually assigned to. It will be shown below that the resulting error in the velocity field can be an order of magnitude higher than the generally admitted uncertainty of PIV measurements in nearly uniform flows.

In the following, we intend to analyse quantitatively these two problems arising at high velocity gradients, and to find ways of reducing the associated errors.

## 2 Lagrangian displacement and window shifting

### 2.1 Background

In cross-correlation PIV, the velocity field of a fluid (projected onto a plane) is deduced from the two-dimensional displacement  $\Delta\mathbf{r}$  of small tracer particles, whose images were recorded at two times, separated by  $\Delta t$ . Assuming that  $\Delta t$  is “sufficiently” small, the particle/fluid velocity  $\mathbf{v}$  is derived from the approximate relation

$$\Delta\mathbf{r} \approx \mathbf{v}\Delta t \quad (1)$$

(This approach does not take into account possible differences between the particle and the fluid velocities, a point which will not be addressed here.) Equation (1) represents the first term of a Taylor series expansion of the velocity field, which, in general, depends on both time and space coordinates. For nearly uniform velocities (on the scale of the correlation window), this approximation is justified. However, when spatial gradients and/or the time-dependence of the velocity field get larger, noticeable differences can appear between the measured velocity  $\Delta\mathbf{r}/\Delta t$  and the real velocity  $\mathbf{v}$ . In this section, we calculate higher-order terms of the expansion in Eq. (1), in order to quantify the measurement errors appearing at high velocity gradients. We also demonstrate analytically and numerically how a simple technique consisting of a symmetric correlation window shift can drastically reduce these errors.

### 2.2 Displacement of one particle

In the following, we treat two-dimensional displacements in a two-dimensional domain, which corresponds to the

most commonly used PIV applications. Consider a particle which, at an initial time  $t_i$ , is located at a position  $\mathbf{r}_i$ , and at a final time  $t_f$  at  $\mathbf{r}_f$  (see Fig. 1). We seek an expression for the displacement  $\Delta\mathbf{r}=\mathbf{r}_f-\mathbf{r}_i$  of the particle in a given velocity field  $\mathbf{v}(\mathbf{r},t)$ , as a function of the different derivatives of  $\mathbf{v}$  and the time interval  $\Delta t=t_f-t_i$ . The derivatives are evaluated at a fixed point  $O$ , which is located close to  $\mathbf{r}_i$  and  $\mathbf{r}_f$  and can be interpreted as the measurement location (e.g. a predefined grid point or centre of a correlation window; see Fig. 1). It is the point at which the velocity needs to be known with precision. Without loss of generality, its coordinates are set to  $\mathbf{r}=\mathbf{0}$ . Similarly, the origin of time ( $t=0$ ), representing the time associated with the velocity measurement, is chosen close to  $t_i$  and  $t_f$  (this point is discussed further in the Appendix). Throughout this paper, all lengths and displacements are measured in pixel units. They can, of course, be converted to physical quantities of the flow field under consideration, using appropriate scaling factors.

We suppose that the non-uniform and time-dependent velocity field

$$\mathbf{v}(\mathbf{r}, t) = \begin{bmatrix} v_x(x, y, t) \\ v_y(x, y, t) \end{bmatrix} \quad (2)$$

can be expanded into a Taylor series around  $t=0$  and the point  $O$  with respect to time  $t$  and the Cartesian coordinates  $x$  (horizontal) and  $y$  (vertical).

Up to second order, this Taylor expansion can be written in compact form as:

$$\mathbf{v}(\mathbf{r}, t) = \mathbf{v}_0 + \mathbf{v}'\mathbf{r} + t\partial_t\mathbf{v} + \frac{1}{2} \begin{pmatrix} \mathbf{r}^\dagger \mathbf{v}''_x \mathbf{r} \\ \mathbf{r}^\dagger \mathbf{v}''_y \mathbf{r} \end{pmatrix} + \frac{1}{2} t^2 \partial_{tt}\mathbf{v} + t\partial_t\mathbf{v}'\mathbf{r} \quad (3)$$

where  $\mathbf{r}^\dagger=(x,y)$  is the conjugate of  $\mathbf{r} = \begin{pmatrix} x \\ y \end{pmatrix}$  and

$$\mathbf{v}_0 = \mathbf{v} \begin{pmatrix} r=0 \\ t=0 \end{pmatrix} \quad \mathbf{v}''_x = \begin{pmatrix} \partial_{xx}v_x & \partial_{xy}v_x \\ \partial_{yx}v_x & \partial_{yy}v_x \end{pmatrix}$$

$$\mathbf{v}' = \begin{pmatrix} \partial_xv_x & \partial_yv_x \\ \partial_xv_y & \partial_yv_y \end{pmatrix} \quad \mathbf{v}''_y = \begin{pmatrix} \partial_{xx}v_y & \partial_{xy}v_y \\ \partial_{yx}v_y & \partial_{yy}v_y \end{pmatrix}$$

with all derivatives being evaluated at  $\mathbf{r}=\mathbf{0}$  and  $t=0$ .

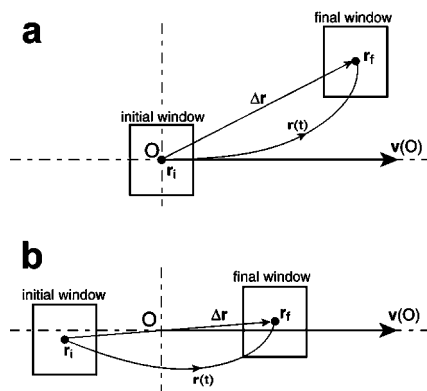


Fig. 1. a Non-symmetric and b symmetric translation of correlation windows with respect to the measurement point  $O$

The full expressions for  $\Delta \mathbf{r}$ , extending Eq. (1) to second and third order in  $\Delta t$ , are derived and given in the Appendix. Here, we focus on two special cases that are of particular interest for PIV algorithms. Both are related to the now frequently used technique of correlation window shifting, in which a first estimate of the velocity field is obtained using identical windows in both images, and where a second computation of correlation is then performed, with the windows shifted by an amount corresponding to the mean local particle displacement. The aim is to reduce the apparent particle displacement between the shifted windows as much as possible to zero, in order to minimize particle loss and the associated noise effects (see e.g. Raffel et al. 1998).

The first case corresponds to the situation depicted in Fig. 1a, where only the window in the second image is shifted according to the measured displacement, the one in the first image remaining centred at the measurement point O. In this “non-symmetric” situation, the particle displacement is given by (see Appendix):

$$\Delta \mathbf{r} = v_0 \Delta t + \frac{\Delta t^2}{2} (v' v_0 + \partial_t v) + O(\Delta t^3) \quad (4)$$

i.e. the displacement error is, as expected, of order  $O(\Delta t^2)$ . We shall see in Sect. 6 that a PIV algorithm using window deformation can easily handle velocity gradients of the order of  $\|\mathbf{v}' \Delta t\| = 0.2$ . If at the same time the displacement is about  $\|\mathbf{v} \Delta t\| = 10$  pixels, i.e. about a third of a typical window size of 32 pixels, which is the upper displacement limit proposed by Adrian (1991), the absolute error between the measured displacement of the particle and the “displacement”  $v_0 \Delta t$  associated with point O can amount to as much as 1 pixel. This is an order of magnitude higher than the generally admitted uncertainty of classical cross-correlation algorithms, which is about 0.1 pixel (Raffel et al. 1998). It is therefore useful to look for a different procedure, for which this error is considerably smaller.

In the second case, the windows used by the algorithm are translated in a symmetric way (see Fig. 1b). This method was proposed recently by Wereley and Meinhart (2001).

The result of the correlation process gives the displacement of a particle whose initial and final positions are symmetric with respect to point O. In this situation, we find that the second-order term vanishes, and the displacement of the particle becomes, at third order (see Appendix):

$$\Delta \mathbf{r} = v_0 \Delta t + \Delta t^3 \left\{ -\frac{v' v_0}{12} + \frac{1}{24} \begin{pmatrix} v_0^\dagger v''_x v_0 \\ v_0^\dagger v''_y v_0 \end{pmatrix} + \frac{1}{12} \partial_t v v_0 - \frac{1}{12} \mathbf{v}' \partial_t v + \frac{1}{24} \partial_{tt} v \right\} + O(\Delta t^4) \quad (5)$$

When applying the general result in Eqs. (4) and (5) to the special case of axisymmetric flow, expressions identical to those given by Wereley and Meinhart (2001) are obtained. For  $\|\mathbf{v}' \Delta t\| = 0.2$  and  $\|\mathbf{v} \Delta t\| = 10$  pixels, the first term of the error is of the order of 0.03 pixel. Assuming that, in most cases, the other terms are of similar magnitude or less, and partly compensate each other, this error between

the velocity in O and the displacement  $\Delta \mathbf{r} / \Delta t$  measured by the algorithm can be neglected, since it is smaller than the noise in the measurements (of the order of 0.1 pixel). If one nevertheless wishes to reduce this error, two solutions are possible: either decreasing the time separation between the two images or removing the error numerically, using Eq. (5) and an approximation of the velocity gradients. However, the second solution amplifies the noise present in the measurements and should be used with caution.

### 2.3 Numerical test of window shifting methods

The previous results were derived for the displacement of a single particle. In the following, we analyse the effect of a non-symmetric and a symmetric translation in an actual PIV algorithm, where average velocities in more or less extended correlation windows are calculated. We tested the two schemes on a velocity field containing a high velocity gradient, using artificial images. We used a horizontal and stationary velocity field given by:

$$\mathbf{v} = \begin{pmatrix} Sx / \Delta t \\ 0 \end{pmatrix} \quad (6)$$

For such a field, the exact displacement, over a time  $\Delta t$ , of a particle which is at a position  $x$  on the first image is:

$$\Delta \mathbf{r} = \begin{bmatrix} x(e^S - 1) \\ 0 \end{bmatrix} \quad (7)$$

For the test, we have chosen a relatively high velocity gradient parameter  $S=0.2$ , which is, however, still small enough for the algorithm to find the correlation peaks efficiently. The first and second images are created using this displacement (see Sect. 5 for details on the creation of the images). We then calculate the error of the horizontal “displacement”  $v_x \Delta t$ , obtained using an algorithm which translates windows of 32 pixels in either a non-symmetric or a symmetric way. The results are shown in Fig. 2. The agreement between the calculated errors and the theoretical predictions in Eqs. (4) and (5) is very good. Both show that the error remains weak in the symmetric case, whereas it is important and cannot be neglected in the non-symmetric case.

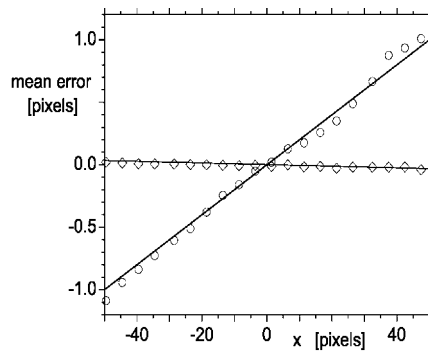


Fig. 2. Error for the velocity field in Eq. (6): open squares, non-symmetric algorithm (as in Fig. 1a); diamonds, symmetric algorithm (as in Fig. 1b). Solid lines correspond to the theoretical predictions of Eqs. (4) and (5)

In the literature, artificial images are sometimes constructed using a displacement  $\mathbf{v}_0\Delta t$  instead of the real displacement of each particle in the velocity field. [This can be shown, for example, for some tests presented by Nogueira et al. (1999).] The error discussed in this section is then hidden, and mostly not considered further, despite its importance.

An additional error arises due to the finite size of the window. Indeed, the algorithm averages the velocity of the particles over the whole window. This introduces a systematic error equal to  $\frac{1}{24}W^2\Delta t\nabla^2v$ , where  $W$  is the size of the windows. This result can be obtained by integrating Eq. (36) (in the Appendix), with the position  $\mathbf{r}_i$  of the particle varying in the initial window (translated by  $-\mathbf{v}\Delta t/2$ ). This error remains negligible as long as  $W$  is smaller than the typical wavelength of the flow. If the wavelength becomes too small, appropriate algorithms, presented e.g. by Nogueira et al. (1999) or Jambunathan et al. (1995), should be used. This error is not present in our tests (including the ones in Sect. 5), since  $\nabla^2v = 0$  here.

In conclusion, the use of symmetric window shifting highly improves the performance of a cross-correlation algorithm by reducing the error between velocities and particle displacements to a lower value than the measurement noise due to other effects, in particular in the presence of high velocity gradients, which deform the correlation peaks. It may be noted that, in principle, the same increase in accuracy is obtained with a non-symmetric algorithm when redefining the measurement position as the mid-point between the two windows, as is done in some PIV algorithms. The inconvenience of this is that the final velocity field is defined on an irregularly spaced grid, requiring complex interpolation procedures (possibly introducing other errors) onto a regular grid for further processing. In the following, we assume that symmetric window shifting is used, and the error discussed in this section is therefore disregarded.

### 3

#### Velocity gradients and window deformation

The main error of PIV algorithms is a noise in the measurements, linked to the width and maximum value of the correlation peak. If the peak is too low, the noise in the correlation function introduces spurious vectors in the velocity field. If the peak is too wide, the determination of the position of its maximum is less accurate and the corresponding displacement is noisy. (Broad peaks can also be caused by a turbulent flow field with scales smaller than the correlation window size, a situation not considered in this paper.) It is thus important to keep the correlation peaks as high and narrow as possible. In the following, we analyse the effect of a velocity gradient on the shape and height of two correlation functions: one using symmetrically shifted square windows, and one using windows which are deformed according to the velocity gradients present in the flow.

#### 3.1

##### Non-deforming correlation function

For two images of intensities  $I_i(\mathbf{r})$  and  $I_f(\mathbf{r})$  at times  $t_i$  and  $t_f$ , respectively, we introduce a new symmetrical correlation function, defined by:

$$R(l) = \frac{\int_w (I_i(\mathbf{r} - l/2) - \bar{I}_i)(I_f(\mathbf{r} + l/2) - \bar{I}_f) d\mathbf{r}}{(\sigma_i\sigma_f)^{1/2}} \quad (8)$$

where

$$\bar{I} = \frac{1}{W^2} \int_w I(\mathbf{r}) d\mathbf{r} \quad \sigma = \int_w (I(\mathbf{r}) - \bar{I})^2 d\mathbf{r}$$

and  $W$  is the side length of a square window centred on the desired location. This function is an adaptation of the standard correlation coefficient (e.g. see Raffel et al. 1998) to the case of a symmetric translation of the windows according to the procedure presented in the previous section. It is normalized in a way to take values between  $-1$  and  $+1$ . This function is defined for continuous values of  $\mathbf{r}$ , even if the real algorithm only uses discrete positions. The effect of the discretization is a slight increase in the effective size of the particles (see Westerweel 1993, ch. 3), which remains small if their diameter is bigger than about 1 pixel.

We now assume that the velocity field  $\mathbf{v}(\mathbf{r})$  is given, and we call  $\mathbf{u}=\mathbf{v}\Delta t$  the ‘‘displacement’’ field. A particle at a position  $\mathbf{r}_i$  on the first image is shifted by an amount  $\Delta\mathbf{r}(\mathbf{r}_i)$  in the second image given by [see Eq. (39) in the Appendix]:

$$\Delta\mathbf{r}(\mathbf{r}_i) = \mathbf{u}_0 + \mathbf{u}' \cdot \mathbf{r}_i + \frac{1}{2}\mathbf{u}' \cdot \mathbf{u}_0 + O(\Delta t^3) \quad (9)$$

In the presence of a velocity gradient, the displacements of the particles are not identical over the whole window (Fig. 3a). The correlation peaks of different particles are located at different positions (Fig. 3b) and cannot interfere constructively as in the case of weak velocity gradients. The correlation peak is thus wider and lower than it is in

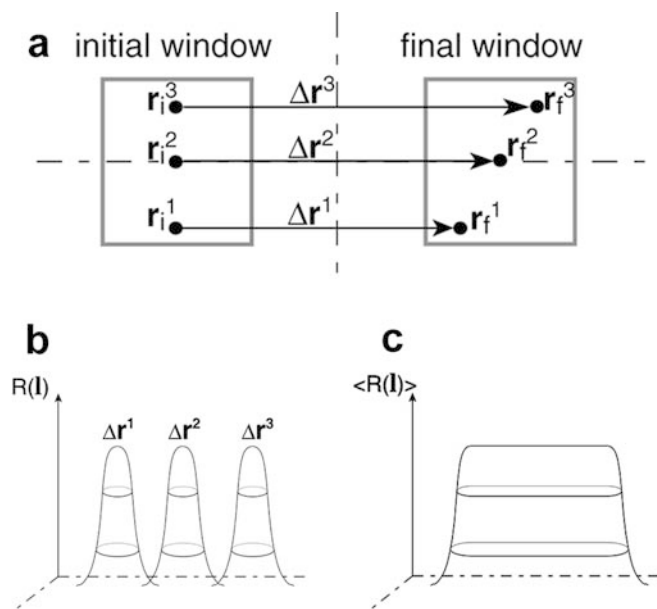


Fig. 3. a Displacement of particles for horizontal shear. b Corresponding correlation function. c Average of the correlation function over different particle distributions

the absence of a velocity gradient. This effect is evaluated quantitatively in the following.

### 3.2 Ensemble average

The shape of the correlation peak highly depends on the distribution of the particles. In order to recover a universal property, we consider the average over all possible particle distributions. The resulting correlation peak then only depends on the velocity field, since the average smooths all the peaks coming from individual particles in a given single distribution. Figure 3c shows this effect schematically. This behaviour was also verified numerically using artificial images; an example is shown in Fig. 4.

For the following calculations, we use a constant particle density  $C$ , as proposed by Adrian (1988). The symbol  $\langle \rangle$  denotes the average over all possible particle distributions.

Westerweel (1993) showed (his Eq. 2.49) that:

$$\begin{aligned} \langle I_i(\mathbf{r}') I_f(\mathbf{r}'') \rangle - \langle I_i(\mathbf{r}') \rangle \langle I_f(\mathbf{r}'') \rangle \\ = C F_{I_0}[\mathbf{r}'' - \mathbf{r}' - \Delta\mathbf{r}(\mathbf{r}')] \end{aligned} \quad (10)$$

where

$$F_{I_0}(\mathbf{l}) = \int I_0(\mathbf{r}) I_0(\mathbf{r} + \mathbf{l}) \, d\mathbf{r} \quad (11)$$

$F_{I_0}$  is the self-correlation function of the intensity of one particle, denoted by  $I_0(\mathbf{r})$ . If the windows are large enough, we can assume that  $\bar{I}$ , which is the mean value of the intensity over the window, is the same for all particle distributions and equal to the ensemble average of the intensity  $\langle I \rangle$ . Similarly, we assume that  $\sigma$ , which is the integral of the variance of the intensity over one image, is given by  $\langle (I - \bar{I})^2 \rangle = W^2$ , which can be simplified using Eq. (10) into  $CW^2 F_{I_0}(0)$ . The expression of the average of the correlation function can then be calculated using Eq. (10):

$$\langle R(\mathbf{l}) \rangle = \frac{1}{W^2 F_{I_0}(0)} \int_W F_{I_0}[\mathbf{l} - \Delta\mathbf{r}(\mathbf{r} - \mathbf{l}/2)] \, d\mathbf{r} \quad (12)$$

This result means that the correlation function is the average of the self-correlation function, centred on the displacement  $\Delta\mathbf{r}(\mathbf{r})$ , for  $\mathbf{r}$  varying in the initial window. For small velocity gradients, the displacement is nearly uniform over the window, and can be approximated by  $\mathbf{u}_0$ . Equation (12) then simplifies into:

$$\langle R(\mathbf{l}) \rangle = \frac{F_{I_0}(\mathbf{l} - \mathbf{u}_0)}{F_{I_0}(0)} \quad (13)$$

This means that, for uniform flow, the correlation peak is exactly equal to the self-correlation function  $F_{I_0}$  centred on the displacement  $\mathbf{u}_0$ .

### 3.3 Height of the peak in the presence of shear

In order to calculate the height of the correlation peak, we define the intensity of one particle as:

$$I_0(\mathbf{r}) = I_{0,\max} e^{-\frac{r^2}{d^2/8}} \quad (14)$$

It is a Gaussian profile of parameter  $d^2/8$ , which is a close approximation of experimental particle image intensity profiles, and for which 95% of the intensity is inside a circle of diameter  $d$ . The self-correlation function  $F_{I_0}$  is also a Gaussian of parameter  $d^2/4$ . We consider a shear “displacement” field defined by:

$$\mathbf{u} = \begin{pmatrix} Sy \\ 0 \end{pmatrix} \quad (15)$$

On the  $x$  axis, the velocity is zero, and the correlation function for windows centred on this axis can be calculated using Eqs. (12) and (39):

$$\begin{aligned} \langle R(\mathbf{l}) \rangle = e^{-\frac{r^2}{d^2/8}} \frac{d}{2SW} \left\{ \operatorname{erf} \left( 2\frac{l_z}{d} + \frac{Sl_y}{d} + \frac{SW}{d} \right) \right. \\ \left. - \operatorname{erf} \left( 2\frac{l_z}{d} + \frac{Sl_y}{d} - \frac{SW}{d} \right) \right\} \end{aligned} \quad (16)$$

“erf” is the error function (integral of the Gaussian function vanishing in 0), and  $\mathbf{l} = (l_x, l_y)$ . This formula was recently given by Hart (2000), who proposed Eq. (12) as a conjecture without derivation. Cuts of this function along the  $x$  and  $y$  axes are plotted in Fig. 5 for different values of

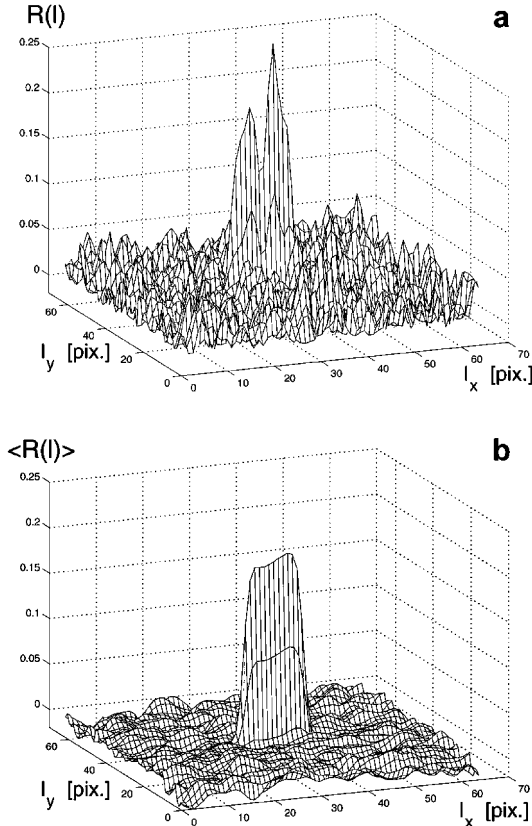


Fig. 4. Correlation functions obtained from test images with a horizontal shear of magnitude  $S=\partial_x u_x=0.4$  (window size  $W=32$  pixels). a Typical result obtained at one position. b Average of 400 correlation functions obtained at different locations with the same mean velocity

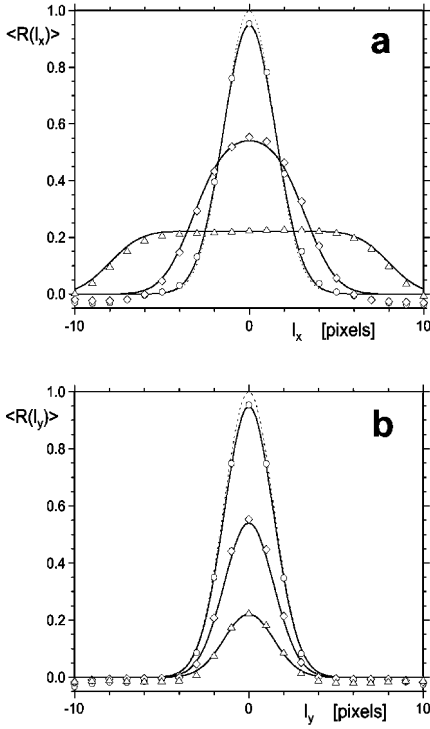


Fig. 5. **a** Horizontal and **b** vertical cut through the correlation functions for horizontal shear flows (Eq. 15), obtained from test images (particle diameter  $d=4$ , window size  $W=32$  pixels): squares,  $S=0.05$ ; diamonds,  $S=0.2$ ; triangles,  $S=0.5$ . The solid lines correspond to the theoretical predictions of Eq. (16), and the dotted line to the result for uniform flow ( $S=0$ )

the shear  $S$ . The agreement with the correlation functions obtained numerically, using artificial images (see Sect. 5 for details) and a non-deforming algorithm, and also shown in these plots, is very good. [The slightly negative values are due to the use of fast Fourier transforms (FFTs) in the correlation process.] It is obvious that increasing velocity gradients tend to widen the correlation peak. At high values of the shear, the peak even becomes flat, making the determination of the location of its maximum impossible.

The height of the peak, found for  $l=0$ , is equal to:

$$\langle R_{\max} \rangle = \frac{d}{SW} \operatorname{erf} \left( \frac{SW}{d} \right) \quad (17)$$

It is interesting to note that the height only depends on the non-dimensional shear parameter  $SW/d$ . Figure 6 shows how it decreases with increasing  $SW/d$ . The numerical results found with artificial images are again very close to the theoretical predictions. A similar analysis was made by Huang et al. (1993a) for particles with a binary intensity (black or white). The same parameter was derived, although their analytical expression for the peak intensity was different.

The analytical and numerical results in this section demonstrate how the presence of velocity gradients widens and lowers the cross-correlation peaks when (symmetrically shifted) rigid correlation windows are used. For high gradients, it is thus necessary to introduce a new correlation function.

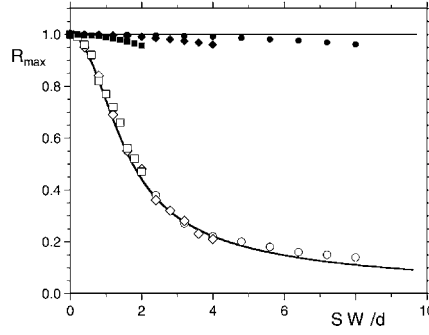


Fig. 6. Height of the correlation peak as a function of the normalized horizontal shear stress for different window sizes: squares,  $W=16$ ; diamonds,  $W=32$ ; circles,  $W=64$ . Particle diameter in the test images is  $d=4$ . Open symbols: without window deformation [the solid line represents the prediction in Eq. (17)]. Filled symbols: with window deformation according to Sect. 3.4

### 3.4 Deforming correlation function

Huang et al. (1993b) proposed to deform the correlation windows, according to the velocity gradients present in the flow, in order to increase the height of the correlation peak. This leads to a new correlation function defined by:

$$R(\mathbf{I}) = \frac{\int_W [I(\mathbf{r} - \frac{1}{2} - \frac{\mathbf{u}'\mathbf{r}}{2}) - \bar{I}_i][I(\mathbf{r} + \frac{1}{2} + \frac{\mathbf{u}'\mathbf{r}}{2}) - \bar{I}_f] d\mathbf{r}}{(\sigma_i \sigma_f)^{1/2}} \quad (18)$$

where

$$\mathbf{u}' = \begin{pmatrix} \partial_x u_x & \partial_y u_x \\ \partial_x u_y & \partial_y u_y \end{pmatrix}$$

The same calculation as in Sect. 3.2, using Eqs. (1) and (39), yields an average correlation function independent of the window size at second order in  $\Delta t$ :

$$\frac{\langle R(\mathbf{I}) \rangle}{F_{I_0}(0)} = F_{I_0} \left[ \frac{(1 + \mathbf{u}'/2)(\mathbf{I} - \mathbf{u})}{F_{I_0}(0)} \right] \quad (19)$$

For this correlation function, the height of the peak is always equal to 1. Its width increases only by 25% for a velocity gradient of 0.5, whereas it was multiplied by a factor of 4 with the non-deforming algorithm. For the shear flow given by Eq. (15), the average correlation function is:

$$\langle R(\mathbf{I}) \rangle = \exp \left( -\frac{4(l_z + Sl_y)^2}{d^2} - \frac{4l_y^2}{d^2} \right) \quad (20)$$

A cut along the  $x$  axis gives the same curve as without shear (i.e. the dotted line in Fig. 5a). On the vertical axis, it is very close to the curve without shear.

Equation (19) predicts a constant peak height of 1. The numerical values in Fig. 6, although close to 1 (and always much larger than those resulting from the non-deforming algorithm), slightly decrease with increasing shear for a given window size. This may be due to the fact that not only the correlation windows, but also the individual particle images, are deformed by the algorithm; this is not taken into account in the theory.

It may be noted here that the use of deformed (as well as translated) correlation windows also greatly reduces the unwanted effects of particles entering or leaving the windows between successive images (Huang et al. 1993b), since these deformations are, by definition, designed to match the particle motion.

The results of this section clearly demonstrate the usefulness of window deformation in cross-correlation PIV, when high velocity gradients are present. However, Eq. (19) was obtained using Eq. (39), which is only a second-order approximation of the displacement of a particle. When taking into account the third-order terms, the errors discussed in Sect. 2 reappear. Moreover, the height of the peak decreases when  $\mathbf{v}''$  is not zero. One way to prevent this would be to define yet another correlation function using the third-order approximation of the displacement given in Eq. (36), but the resulting expressions and calculations become exceedingly complex. An alternative was proposed by Jambunathan et al. (1995) and Nogueira et al. (1999). Unfortunately, their techniques require quite high computing power, since the algorithm must perform at least 30 iterations, whereas the present algorithm converges in only two or three iterations (see Sect. 6). Their method becomes necessary for flows with very small wavelengths, of the order of the window size or less.

#### 4 Gaussian filter for fixed windows

For the calculation of the correlation function in Eq. (18), the algorithm deforms the windows according to the velocity gradients of the flow. Since the velocity field is completely unknown in the beginning, several iterations must be performed in order to converge towards a solution where velocities and velocity gradients are known together. The main problem is to obtain a sufficiently accurate result in the first run. Without the velocity gradients, the algorithm cannot calculate the deforming correlation function given by Eq. (18) in this first run. It must find the displacement using the non-deforming correlation function in Eq. (8). If the gradients are high, the associated effects discussed in Sect. 3 lead to increased noise in the results, resulting in turn in a large number of spurious vectors, which can be very different from the real velocity vectors. If there are too many of them in the first iteration, the velocity gradients cannot be determined correctly for the second iteration, and the iterative process cannot continue successfully. We thus need to increase the height of the peaks, even if some accuracy is lost. The aim is to obtain at least a rough approximation of the velocity gradients, so that the next iterations are carried out correctly.

The height of the correlation peak for non-deformed windows is linked to the parameter  $SW/d$  through Eq. (17) (see also Fig. 6). A recipe for increasing its height proposed in the literature (Fincham and Spedding 1997; Lin and Perlin 1998) is to decrease the size  $W$  of the correlation window. But by doing this, the number of particles in the window also decreases and the correlation peak is more sensitive to noise. Another idea would be to increase the diameter  $d$  of the particles. However, in an experiment,

there is a maximum allowable size of the particles if one wants them to be accurate tracers that move very closely with the fluid velocity. One way around this problem, which has been used before, consists in optical defocusing of the flow images at acquisition, leading to bigger apparent particle images. The drawback is that, for higher iterations of the algorithm, the accuracy will always be limited by the width of the auto-correlation function of the particle intensities (Eq. 18), i.e. it will be noisier than it could have been without defocusing.

In the present algorithm, the idea is to increase the size of the particles numerically for the first run only, by applying a Gaussian filter to the images, i.e. an operation similar to numerical defocusing. We convolute the image intensity matrix with a Gaussian function of parameter  $\delta^2/8$  defined by:

$$G(\mathbf{r}) = \frac{8}{\pi\delta^2} e^{-\frac{r^2}{\delta^2/8}} \quad (21)$$

This technique has the following advantages:

- Large sizes of the correlation window can be used, keeping the number of particles images in them high.
- The actual particles in the fluid do not need to be large. They can be chosen small enough to be considered as accurate tracers.
- The noise is smoothed during the filtering. The signal to noise ratio does not decrease in the presence of the filter.
- The unfiltered images are not lost, they are available for further iterations of the algorithm. The error, which scales on the diameter of the particles, is thus independent of the velocity gradient.

Figure 7 shows a typical correlation function obtained without and with a Gaussian filter. Although it might seem counter-intuitive, the height of the peak increases when the images are filtered. However, the width of the peak increases simultaneously. This is why this technique should only be used for the first iteration, where only a rough approximation of the velocity field is needed. The resulting velocities are not highly accurate, but there are very few spurious vectors. This is well illustrated in Fig. 8, where the number of spurious vectors is plotted as a function of the parameter  $\delta$  characterizing the width of the filtering function, for the example of a simple shear given by Eq. (15). For appropriately chosen values of  $\delta$ , the number of spurious vectors can be decreased by an order of magnitude with respect to the unfiltered case. This technique remains efficient in the presence of noise; adding a random white noise to the image, whose amplitude is 20% of the maximum intensity of a particles, does not significantly change the number of spurious vectors.  $\delta$  should be chosen so that this number is a minimum. If one admits that the shear flow used to obtain the result in Fig. 8 is representative for more general types of velocity gradients, this would occur for  $\delta/SW$  near unity. This then leads to an empirical determination of the optimal value for  $\delta$ :

$$\delta \approx W\partial_t v \Delta t \quad (22)$$

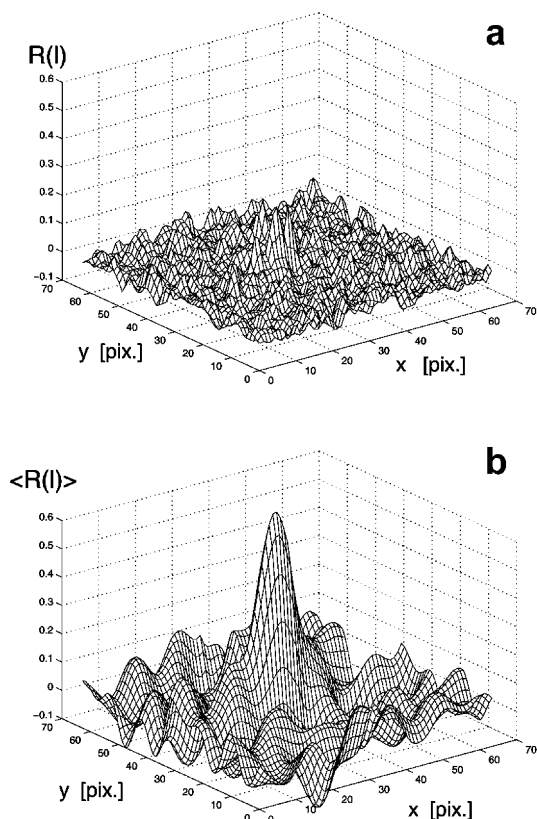


Fig. 7. Typical correlation functions obtained for a velocity field given by Eq. (15) with  $S=0.2$ , with a window size  $W=64$  pixels: **a** without, and **b** with convolution of the image with a Gaussian (Eq. 21) of parameter  $\delta=9$  pixels

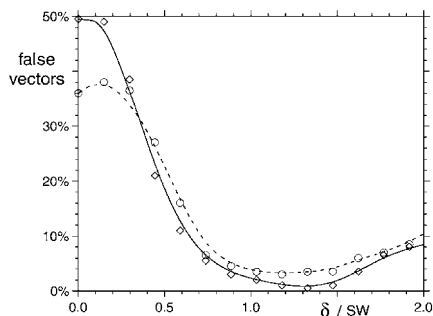


Fig. 8. Fraction of false vectors (error larger than 2 pixels) obtained for images filtered with a Gaussian function defined by Eq. (21). The velocity field is a horizontal shear (Eq. 15) with  $S=0.15$ . Window size is  $W=64$  pixels. *Circles*, without noise; *diamonds*, with 20% noise

where  $\partial_r v$  represents the maximum velocity gradient present in the given flow, which, in many cases, can be estimated roughly beforehand.

The technique of filtering of the images with a Gaussian function thus seems very effective for the determination of a rough approximation of the velocity field, without knowledge of the velocity gradients. In the PIV algorithm used in the present study and described below, it is used for the first correlation in an iterative process.

## 5

### Description of a new PIV algorithm

A cross-correlation PIV algorithm was developed, using the above techniques: symmetric translation of the windows, Gaussian filtering, and window deformation according to the velocity gradients. It is briefly outlined in the following.

#### 5.1

##### First correlation

Before the first cross-correlation run, the two images are filtered with the Gaussian function defined in Eq. (21), with the parameter  $\delta$  chosen according to Eq. (22). The images are divided into correlation windows in the standard way, centred on the points of a grid, where the velocity vectors are to be calculated. The correlation function of a given window pair is obtained by a FFT routine, as recommended by Raffel et al. (1998, Sect. 5.4.4), but initially without the use of the weighting factor compensating the in-plane loss of pairs. This factor strongly amplifies the correlation values for large displacements, which may lead to situations where noise-generated peaks become larger than those associated with the real particle displacement, resulting in spurious calculated velocity vectors. Once the maximum of the unweighted function is detected, the precise peak location is determined to subpixel accuracy, using a three-point Gaussian-fit estimator (Westerweel 1993, ch. 3.8). For this, the in-plane loss correction is now applied for better accuracy. The location of the correlation peak corresponds to the velocity at the centre of the window. At the end, spurious vectors are detected and replaced by a median-filter procedure (see Westerweel 1993 for details). If the values of window size  $W$  and Gaussian filter parameter  $\delta$  were chosen in a way to roughly comply with the condition in Eq. (22), only 2–5% of all velocity vectors would be erroneous (see Fig. 8), which is a sufficiently low fraction to guarantee the successful start of the iteration process. If significantly more vectors are bad, the first correlation must be repeated with more suitable values for  $W$  and  $\delta$ .

#### 5.2

##### Further iterations

In the subsequent iterations, the unfiltered original images are used. For an iteration number  $j$ , the windows are translated and deformed according to the displacement field  $\mathbf{u}_{j-1}$  calculated in the preceding iteration  $j-1$ . The displacement gradients  $\mathbf{u}'$  are obtained through a centred finite-difference scheme. For each correlation window, the algorithm rebuilds two new intensity functions  $\bar{I}_i(\mathbf{r})$  and  $\bar{I}_f(\mathbf{r})$ , where  $\mathbf{r}=(x,y)$ , and  $x,y$  range from  $-W/2$  to  $+W/2$ :

$$\bar{I}_i(\mathbf{r}) = I_i\left(\mathbf{r} - \frac{\mathbf{u}_{j-1}}{2} - \frac{\mathbf{u}'_{j-1}\mathbf{r}}{2}\right) \quad \bar{I}_f(\mathbf{r}) = I_f\left(\mathbf{r} + \frac{\mathbf{u}_{j-1}}{2} + \frac{\mathbf{u}'_{j-1}\mathbf{r}}{2}\right) \quad (23)$$

The value of the intensity  $I_i$  and  $I_f$  between pixels is found by bi-linear interpolation between the four neighbouring values (e.g. see Nogueira et al. 1999). A correlation function of these new intensities is calculated [equal to the



deforming correlation function in Eq. (18)] in the same way as in the first iteration, using FFT, and the location  $\mathbf{l}_{\max}$  of its peak is determined to subpixel accuracy. The new displacement  $\mathbf{u}_j$  is then given by:

$$\mathbf{u}_j = \mathbf{u}_{j-1} + \mathbf{l}_{\max} \quad (24)$$

False vectors are again treated using a median filter procedure.

These iterations are carried out two or three times, depending on the strength of the velocity gradients.

### 5.3

#### Last iteration

In the previous iterations, a relatively coarse grid is used for rapidity of the algorithm. In the final iteration, a refinement of the spatial resolution is achieved by increasing the number of vectors, and possibly by reducing the size  $W$  of the correlation windows. Otherwise, the procedure is the same as for the other iterations. The final displacement field has a high spatial resolution and high accuracy.

## 6

### Error estimates and optimization

#### 6.1

##### Procedure and results

We carried out tests with the above algorithm to determine the error caused by velocity gradients. For this purpose, pairs of artificial grey-scale images (256 intensity levels, 8 bits/pixel) were created numerically. Particles with a Gaussian intensity profile given by Eq. (14), with a diameter of  $d=2$  pixels, were introduced on the first image with an average density of  $C=0.02$  particles/pixel. This corresponds to about 20 particles in a window of  $32 \times 32$  pixels. The new positions are then calculated using the exact Lagrangian displacement of the flow under consideration, and the particles are introduced on the second image. In order to achieve more typical experimental conditions, a random white noise with values between zero and 10% of the maximum intensity of the particles was added to each pixel of both images. The average normalized cross-correlation coefficient between “clean” and “noisy” images obtained in this way is close to 0.955.

We used shear flows defined by Eq. (15), with a shear parameter  $S$  varying from 0 to 0.5. For such flows, the velocity gradients are uniform. The following results are therefore mainly representative for flows with slowly varying velocity gradients (on the scale of the correlation window), e.g. flows with large vortices or large expansion or shear areas. The average “root-mean-square” error between the true displacement  $\mathbf{u}$  and the displacement  $\mathbf{u}_{\text{meas}}$ , found by the algorithm, defined as

$$\varepsilon_{\text{rms}} = \sqrt{\langle \|\mathbf{u} - \mathbf{u}_{\text{meas}}\|^2 \rangle} \quad (25)$$

is calculated. The average is performed over all vectors corresponding to particle displacements of up to a third of the window size  $W$ , which is the generally admitted maximum allowed displacement for the correlation to work

properly. By doing this, the peak locking error is also effectively averaged out.

The results of these tests are presented in Fig. 9 for three different window sizes and a varying number of iterations. As a general trend, one observes a faster increase in the error for larger window sizes, which is a consequence of the associated decrease in the height of the peak (see Fig. 6). For a conventional algorithm without window deformation, the error increases rapidly with the velocity gradient. For example, for a window size  $W=32$  pixels, it already reaches 0.3 pixels for a relatively moderate displacement gradient of  $S(=du/dy)=0.1$ . These results are in agreement with previous calculations made by Raffel et al. (1998). With window deformation, the error increases much more slowly, even after very few iterations. For the same conditions as above ( $W=32$  pixels,  $S=0.1$ ), the error is divided by a factor of 10 after only the second iteration. It is important to notice that it is not necessary to carry out more than four iterations, since no further increase in accuracy is obtained. Two or three iterations are even sufficient in the case of moderate velocity

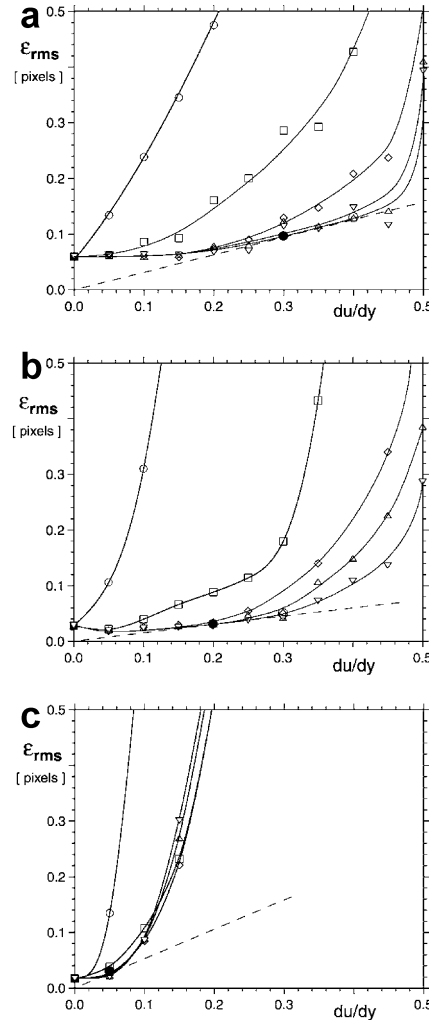


Fig. 9a-c. Rms errors obtained with artificial images of simple shear flow (Eq. 15): *circles*, algorithm without window deformation; *squares, diamonds, triangles, inverted triangles*, with window deformation after 1, 2, 3 and 4 iterations, respectively. Window sizes: a  $W=16$  pixels; b  $W=32$  pixels; c  $W=64$  pixels

gradients. Supposing that enough iterations are made so that the calculated displacements converged, the error is found to remain almost constant up to a critical value of the displacement gradient, above which it then increases rapidly.

The behaviour of the rms error shown in Fig. 9 for simple shear flows is representative of other more general flows with approximately constant gradients. As an illustration, Fig. 10 compares the results for four different linear flows: simple shear (Eq. 15), one-dimensional stretching (Eq. 6), plane stagnation point flow, and solid-body rotation. The latter two have displacement fields given by:

$$\mathbf{u} = \begin{pmatrix} Sx \\ -Sy \end{pmatrix} \text{ and } \mathbf{u} = \begin{pmatrix} -Sy \\ Sx \end{pmatrix} \quad (26)$$

respectively. For these tests, the correlation window size is  $W=32$  pixels, and two iterations were carried out. The overall evolution of the error with the velocity gradient parameter  $S$  is quite similar in all these cases.

## 6.2

### Optimum time separation and window size

The preceding results may be used to determine the optimum time separation  $\Delta t$ , which should be chosen in a given experiment, i.e. the separation which minimizes the relative error in the velocity measurements, provided the algorithm in the preceding section is used, with a sufficient number of iterations. Let  $\mathbf{v}(\mathbf{r})$  be the experimental velocity field, which, up to a scaling factor determined by the optical arrangement, is given in pixels per unit time. We further let  $\partial_r v$  represent the maximum velocity gradient in this flow. The corresponding displacement field and displacement gradient are  $\mathbf{u}=\mathbf{v}\Delta t$  and  $\partial_r u=\partial_r v\Delta t$ , respectively. The relative measurement error is given by

$$\varepsilon_{\text{rel}} = \frac{\varepsilon_{\text{rms}}}{\|\mathbf{u}\|} = \frac{\varepsilon_{\text{rms}}}{\|\mathbf{v}\Delta t\|} \quad (27)$$

At first sight, Eq. (27) would suggest that the relative error decreases with increasing  $\Delta t$ , so that the latter should be chosen as high as possible. However, Fig. 9 shows that the absolute error  $\varepsilon_{\text{rms}}$  tends to increase with the

displacement gradient, and therefore also with  $\Delta t$ , which may work against the positive effect of increasing displacement  $\mathbf{u}$ . In order to assess the net result, it is useful to rewrite the relative error as

$$\varepsilon_{\text{rel}} = \frac{\varepsilon_{\text{rms}}}{\partial_r u} \frac{\partial_r u}{\|\mathbf{v}\|} \quad (28)$$

The second term is entirely determined by the flow field, and independent of  $\Delta t$ ; it is the inverse of a characteristic length scale  $L$  of the velocity field. The first term depends on the time separation via  $\partial_r u$ . Supposing  $\varepsilon_{\text{rms}}$  varies approximately as in Fig. 9, this term is minimum when the displacement gradient  $\partial_r u$  equals some optimum value  $S_{\text{opt}}$ , which is a function of the correlation window size. We find  $S_{\text{opt}} \approx 0.3, 0.2, 0.05$  for  $W=16, 32, 64$  pixels, respectively. These values correspond to the black dots in Fig. 9. They are found by minimizing the slope of the line joining the origin and a given data point in Fig. 9. The first term in Eq. (28) corresponds to this slope. For a given velocity field with a gradient  $\partial_r v$ , this condition on the displacement gradient leads to a first estimate of the optimal time separation:

$$\Delta t_1 \approx \frac{S_{\text{opt}}}{\partial_r v} \quad (29)$$

An additional well-known limitation for  $\Delta t$  is given by the fact that the particle displacement between images should not exceed about a third of the correlation window size (Adrian 1991), in order to prevent excessive in-plane loss of pairs in the first correlation run with fixed windows. The corresponding upper bound for the time separation is

$$\Delta t_2 \approx \frac{W}{3\|\mathbf{v}\|} \quad (30)$$

In summary, since the time separation should never exceed this limit, the condition given in Eq. (29) should be modified into:

$$\Delta t = \min(\Delta t_1, \Delta t_2) \quad (31)$$

As a final result, an estimation of the minimum relative error as a function of the flow length scale  $L$ , achievable

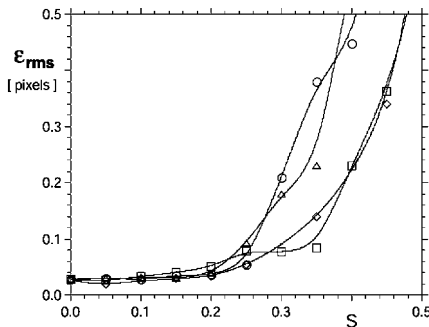


Fig. 10. Rms errors obtained with artificial images and different types of velocity gradients. Algorithm with  $W=32$  pixels, and two iterations: *diamonds*, simple shear (Eq. 15); *squares*, 1-dimensional stretching (Eq. 6); *circles*, converging/diverging flow (plane stagnation point); *triangles*, solid-body rotation (Eq. 26)

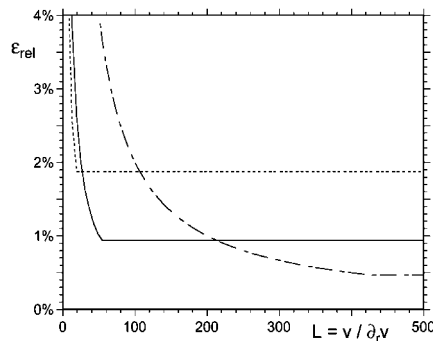


Fig. 11. Minimum relative error that can be obtained with the present algorithm as a function of the characteristic length of the flow, for different window sizes:  $W=16$  pixels (*dotted line*);  $W=32$  pixels (*solid line*);  $W=64$  pixels (*dash-dotted line*)

with the present algorithm, is given in Fig. 11. For this, a somewhat more conservative absolute error of  $\varepsilon_{\text{rms}}=0.1$  was assumed, which is thought to be more representative of measurements on realistic flows than the values in Fig. 9. The relative error was calculated in the following way: for  $\Delta t_1 < \Delta t_2$  (i.e. for small length scales  $L$ ), it is given by Eq. (28), with  $\partial_r u = S_{\text{opt}}$ . For  $\Delta t_1 > \Delta t_2$  (large  $L$ ), we use Eqs. (27) and (30).

The 32-pixel window shows the best overall performance. For most length scales, the error is less than 1%. High deviations are only observed for lengths considerably smaller than the window size. For characteristic lengths larger than about 200 pixels, i.e. for nearly uniform flows, the use of 64-pixel windows leads to more accurate results. The error for  $W=16$  is twice as high as for  $W=32$  for high  $L$ , due to the reduced maximum allowable displacement; and even for low  $L$ , the gain in accuracy is not very significant.

## 7 Application to experimental images

The techniques and algorithm described in the preceding sections were used for the experimental study of the interaction of two co-rotating vortices. The vortices were generated in water by the impulsive motion, from rest, of two flat plates, rotated in a symmetric way. They are initially parallel, laminar, uniform along their axes, and they have the same circulation and no axial flow in their cores. Their core diameter is of the order of 1 cm, the distance between their centres 2–3 cm, and their axial length about 100 cm. This large aspect ratio of the vortex pairs was necessary to minimize the influence of end effects and maintain the central part of the flow essentially two-dimensional for the time of observation.

For the PIV velocity measurements, the flow was seeded with spherical plastic particles (Optimage Ltd., UK) of density  $1.03 \text{ g/cm}^3$  and mean diameter  $50 \text{ }\mu\text{m}$ . The seeding density was about  $150\text{--}200 \text{ particles/cm}^3$ . Planes perpendicular to the vortex axes were illuminated by a sheet of light (thickness 3–5 mm) from a 5 W continuous argon ion laser. Pairs of digital images ( $1,008 \times 1,018 \text{ pixels}^2$ , 256-level grey scale) of the particles in this plane were recorded using a Kodak Megaplug ES1.0 camera. With the field of view being about  $(15 \text{ cm})^2$ , the average particle density in the images was  $C=0.02\text{--}0.03 \text{ particles/pixel}$ .

In the example shown here, the approximate values for the circulation  $\Gamma$ , core radius  $a$  and maximum vorticity  $\omega$  of the vortices are:  $\Gamma=15 \text{ cm}^2/\text{s}$ ,  $a=5 \text{ mm}$ ,  $\omega=16/\text{s}$ . The characteristic length scale of the flow is given by the core radius  $a$ ; the velocity gradient by half the vorticity; and the maximum velocity by  $\omega a/2$ . With the particular calibration associated with the present set-up, this results in  $L \approx 30 \text{ pixels}$ ,  $\partial_r v \approx 8/\text{s}$  and  $v \approx 240 \text{ pixels/s}$ .

These flow characteristics lead to the following set of optimum parameters for the PIV algorithm. According to Fig. 11, a window size  $W=32 \text{ pixels}$  would give the lowest relative error. From this we get  $\Delta t_1=0.025 \text{ s}$  and  $\Delta t_2=0.044 \text{ s}$  (Eqs. 29 and 30); the optimum time delay between images is therefore found to be  $\Delta t=25 \text{ ms}$  according to Eq. (31). With the velocity gradient parameter  $S=\partial_r v \Delta t=0.2$ , Eq. (22) gives the optimum value for the

parameter of the Gaussian filter in the first correlation run:  $\delta \approx 6 \text{ pixels}$ .

Figure 12 shows the velocity field obtained from the particle images of the above flow, using a conventional algorithm without Gaussian filter or window deformation on the one hand, and using the present algorithm on the other. Both use symmetric window shifting and calculate a field of  $60 \times 60$  vectors ( $W=32 \text{ pixels}$ , 50% window overlap); only a fraction of which is shown in the close-up views in Fig. 12. The conventional algorithm is clearly unable to give a reliable velocity field in the core of the vortices, in which the displacement gradients are highest. On the contrary, the present algorithm still allows a very good determination of the velocity field in these regions. This difference in performance is even more visible in the vorticity distribution, obtained from this data by a standard finite-difference scheme and shown in Fig. 13. Whereas the new algorithm gives nearly circular vorticity contours, the conventional scheme splits each vortex into a number of smaller patches, creating the false impression of turbulent motion, although the flow was clearly laminar at this stage, as shown by previous dye visualizations.

The vortices in Figs. 12 and 13 later undergo merging into a single final vortex. At higher Reynolds numbers, the flow undergoes a three-dimensional instability and develops smaller structures and intermittent turbulent motion. These different phenomena have been analysed

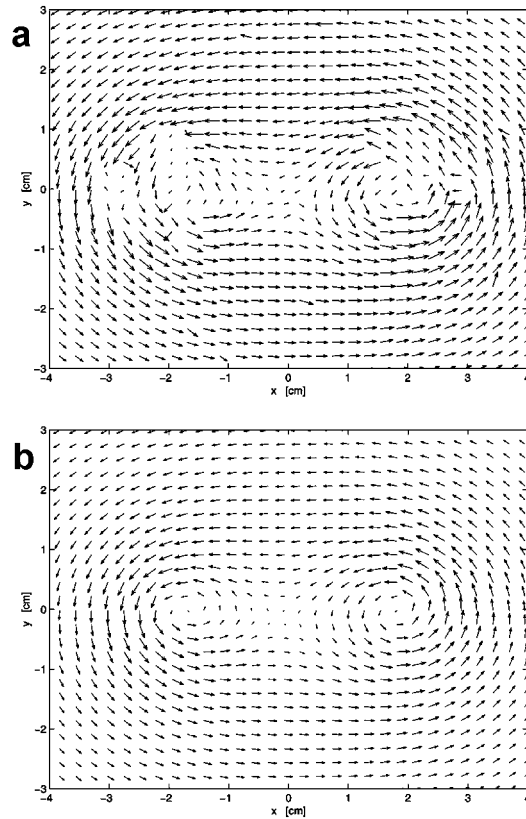


Fig. 12. Example of the velocity field obtained using real experimental particle images of a co-rotating vortex pair flow. a conventional algorithm; b present algorithm

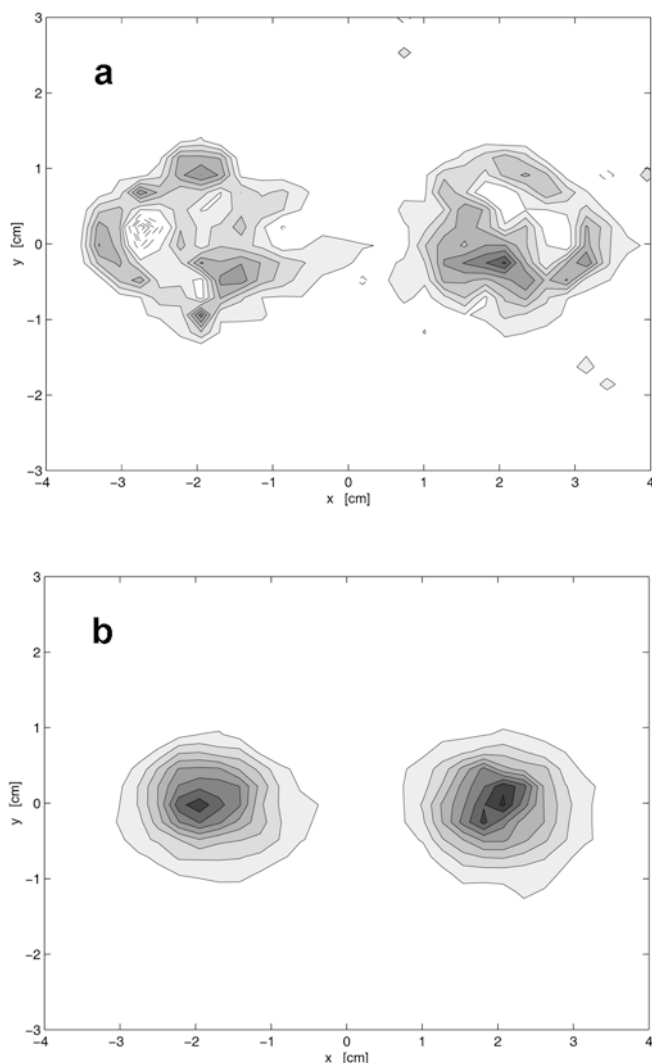


Fig. 13. Vorticity distribution calculated from the data in Fig. 12. Contours are separated by  $2/s$ ; dashed contours correspond to negative vorticity

quantitatively in great detail, using the PIV procedure and algorithm described in this paper. The results show excellent agreement with theoretical and numerical analyses of the same flow. Details of this work can be found in Leweke et al. (2001), Meunier and Leweke (2001, 2002) and Meunier et al. (2002).

## 8 Summary

In this paper, we performed an analytical and numerical study of the effects of velocity (displacement) gradients in cross-correlation PIV algorithms with window shifting and deformation.

Expressions for the error between measured displacement (representing the average velocity of a particle on its trajectory between the two images) and the displacement corresponding to the true velocity at the measurement location were obtained as functions of the velocity field, up to third order in space and time. These results show that

an important bias error exists, even for moderate displacement gradients, when correlation windows are displaced in a non-symmetric way. However, this error is reduced to a level below the standard noise-related error when using symmetric translations of the correlation windows in the two images.

The effect of gradients on the shape and height of the correlation peaks was also analysed in detail. Analytic expressions for peak profiles were calculated for both non-deforming and deforming symmetric algorithms. They show that the strong broadening of the peak and decrease in its amplitude, observed in the presence of gradients for the case without deformation, are strongly reduced when deforming the correlation windows according to the gradients of the flow.

A method of obtaining a reliable first approximation of the velocity field, without previous knowledge of velocities or gradients, is proposed and tested successfully. It is based on a numerical filtering of the images, which increases the apparent size of the individual particle images in the first run only.

All theoretical predictions were tested for representative flow configurations, using artificial images. The agreement between analytical and numerical results, as well as between the present general results and special cases treated in the literature, is found to be very good.

An iterative PIV algorithm was developed, using the techniques described in this paper, and adapting the window deformation technique, initially proposed by Huang et al. (1993b) to the use of FFTs for increased processing speed. Error tests performed with artificial images demonstrate that, even in the presence of relatively large gradients, only a few iterations with window deformation are necessary to reduce the error to a level obtained for almost uniform flows with a non-deforming algorithm. Based on these error estimates, a practical guideline for the choice of the optimum time separation between images is given as a function of the velocities and gradients present in the flow under consideration.

The increased efficiency and reliability of the present algorithm with respect to conventional PIV methods was demonstrated by application to real experimental images of a flow with concentrated vortices.

## Appendix

We seek an expression for the displacement  $\Delta\mathbf{r}=\mathbf{r}_f-\mathbf{r}_i$  of a particle in the velocity field given by Eq. (3). At  $t=t_i$ , the particle is located at  $\mathbf{r}_i$ , and at  $t=t_f$  at  $\mathbf{r}_f$ . The origin of time ( $t=0$ ) is given by the time at which one wishes to determine the velocity at the reference point 0 (of coordinates  $\mathbf{r}=0$ ). All derivatives are taken at this point and time. The particle trajectory  $\mathbf{r}(t)$  is calculated by an iterative process as successive solutions of the differential equation

$$\frac{d}{dt} = v[\mathbf{r}(t)] \quad (32)$$

at increasing orders of  $t$  and  $\mathbf{r}$ .

At first order, the solution  $\mathbf{r}_1(t)$  of Eq. (32) [using Eq. (3) taken at order 0] is given by:

$$\mathbf{r}_1(t) = \mathbf{r}_i + \int_{t_i}^t \mathbf{v}_0 dt' = \mathbf{r}_i + \mathbf{v}_0(t - t_i) \quad (33)$$

Introducing this result into (3) leads, at first order, to:

$$\frac{d}{dt} = v[\mathbf{r}_1(t)] = v_0 + v'(\mathbf{r}_i + \mathbf{v}_0 t) + t \partial_t v \quad (34)$$

The solution of Eq. (34) is the approximation  $\mathbf{r}_2(t)$  of the trajectory to the second order:

$$\mathbf{r}_2(t) = \mathbf{r}_i + \mathbf{v}_0(t - t_i) + \frac{t^2 - t_i^2}{2} \partial_t v + v' \mathbf{r}_i(t - t_i) + v' \mathbf{v}_0 \frac{(t - t_i)^2}{2} \quad (35)$$

The third-order approximation  $\mathbf{r}_3(t)$  of the particle trajectory is found in the same way, the final result being

$$\begin{aligned} \mathbf{r}_3(t) &= \mathbf{r}_i + \int_{t_i}^t \mathbf{v}[\mathbf{r}_2(t')] dt' \\ &= \mathbf{r}_i + \mathbf{v}_0(t - t_i) + \mathbf{v}' \mathbf{r}_2(t - t_i) \\ &\quad + \mathbf{v}' \mathbf{v}_0 \frac{(t - t_i)^2}{2} v' \partial_t \mathbf{v} \left[ \frac{t^3 - t_i^3}{6} - \frac{t_i^2}{2} (t - t_i) \right] \\ &\quad + \mathbf{v}'^2 \mathbf{r}_i \frac{(t - t_i)^2}{2} v'^2 \mathbf{v}_0 \frac{(t - t_i)^3}{6} \\ &\quad + \partial_t \mathbf{v} \frac{t^2 - t_i^2}{2} + \begin{pmatrix} \mathbf{r}_0^\dagger \mathbf{v}'_x \mathbf{r}_0 \\ \mathbf{r}_0^\dagger \mathbf{v}''_y \mathbf{r}_0 \end{pmatrix} \frac{t - t_i}{2} \\ &\quad + \begin{pmatrix} \mathbf{v}_0^\dagger & \mathbf{v}''_x & \mathbf{r}_i + \mathbf{r}_i^\dagger & \mathbf{v}''_x & \mathbf{v}_0 \\ \mathbf{v}_0^\dagger & \mathbf{v}''_y & \mathbf{r}_i + \mathbf{r}_i^\dagger & \mathbf{v}''_y & \mathbf{v}_0 \end{pmatrix} \frac{(t - t_i)^2}{4} \\ &\quad + \begin{pmatrix} \mathbf{v}_0^\dagger & \mathbf{v}''_x & \mathbf{v}_0 \\ \mathbf{v}_0^\dagger & \mathbf{v}''_y & \mathbf{v}_0 \end{pmatrix} \frac{(t - t_i)}{6} \partial_{tt} \mathbf{v} \frac{t^3 - t_i^3}{6} \\ &\quad + \partial_t \mathbf{v}' \mathbf{r}_i \frac{t^2 - t_i^2}{2} + \left( \frac{t^3 - t_i^3}{3} - \frac{t^2 - t_i^2}{2} t_i \right) \partial_{tt} \mathbf{v}' \mathbf{v}_0 \end{aligned} \quad (36)$$

- *Non-symmetric displacement*: For the displacement corresponding to Fig. 1a, we have  $\mathbf{r}_i=0$ , and the choice  $t_i=0$  and  $t_f=\Delta t$  seems appropriate. Using Eq. (35), this leads to Eq. (4), showing that, in this case, the error between the measured velocity  $\Delta \mathbf{r}/\Delta t$ , and the true velocity  $\mathbf{v}_0$  at the measurement location and at the time of the first image is of second order in  $\Delta t$ . One could also choose the origin of time halfway between  $t_i$  and  $t_f$  [see Eq. (38) below], which means that the measured velocity field is associated with the instant between the exposures of the two images. In this case, the term proportional to  $\partial_t \mathbf{v}$  in Eq. (4) would vanish, but the error, now given by Eq. (39), would still remain  $O(\Delta t^2)$ .
- *Symmetric displacement*: For the displacement corresponding to the symmetric window shifting in Fig. 1b, the following relations hold:

$$\mathbf{r}_f = -\mathbf{r}_i = \frac{\Delta \mathbf{r}}{2} \quad (37)$$

$$t_f = -t_i = \frac{\Delta t}{2} \quad (38)$$

Introducing Eq. (38) into Eq. (35), we obtain for  $t=t_f$ :

$$\Delta \mathbf{r} = v_0 \Delta t + v' \mathbf{r}_i \Delta t + v' \mathbf{v}_0 \frac{\Delta t^2}{2} + O(\Delta t^3) \quad (39)$$

and, with Eq. (37) and  $\mathbf{I}$  being the unit matrix

$$\left( \mathbf{I} + \frac{\Delta t}{v'} v' \right) (\Delta \mathbf{r} - v_0 \Delta t) = O(\Delta t^3) \quad (40)$$

This results in

$$\Delta \mathbf{r} = v_0 \Delta t + O(\Delta t^3) \quad (41)$$

showing that, for a symmetric displacement, the error is only of order  $\Delta t^3$ . The expression in Eq. (5) for this higher-order term is found by introducing Eqs. (37), (38) and (41) into Eq. (36).

## References

- Adrian RJ (1988) Statistical properties of particle image velocimetry measurements in turbulent flow. In: Adrian RJ et al. (eds) Laser anemometry in fluid mechanics—III. LADOAN Instituto Superior Tecnico, Lisbon, pp 115–129
- Adrian RJ (1991) Particle-imaging techniques for experimental fluid mechanics. Annu Rev Fluid Mech 23:261–604
- Fincham A, Delerce G (2000) Advanced optimization of correlation imaging velocimetry algorithms. Exp Fluids [Suppl] 29:S13–S22
- Fincham AM, Spedding GR (1997) Low cost, high resolution DPIV for measurement of turbulent fluid. Exp Fluids 23:449–462
- Hart DP (2000) PIV error correction. Exp Fluids 29:13–22
- Huang HT, Fiedler HE, Wang JJ (1993a) Limitation and improvement of PIV Part I: Limitation of conventional techniques due to deformation of image patterns. Exp Fluids 15:168–174
- Huang HT, Fiedler HE, Wang JJ (1993b) Limitation and improvement of PIV Part II: Particle image distortion, a novel technique. Exp Fluids 15:263–273
- Ishikawa M, Murai Y, Wada A, Iguchi M (2000) A novel algorithm for particle tracking velocimetry using the velocity gradient tensor. Exp Fluids 29:519–531
- Jambunathan K, Ju XY, Dobbins BN, Ashforth-Frost S (1995) An improved cross correlation technique for particle image velocimetry. Meas Sci Technol 6:507–514
- Leweke T, Meunier P, Laporte F, Darracq D (2001) Controlled interactions of co-rotating vortices. In: Bütetfisch K et al (eds) 3rd ONERA-DLR Aerospace Symposium (ODAS 2001). ONERA, Paris, paper S2-3
- Lin HJ, Perlin M (1998) Improved methods for thin surface boundary layer investigations. Exp Fluids 25:431–444
- Meunier P, Leweke T (2001) Three-dimensional instability during vortex merging. Phys Fluids 13:2747–2750
- Meunier P, Leweke T (2003) Elliptic instability of a co-rotating vortex pair. J Fluid Mech in press
- Meunier P, Ehrenstein U, Leweke T, Rossi M (2002) A merging criterion for two-dimensional co-rotating vortices. Phys Fluids 14:2757–2766
- Nogueira J, Lecuona A, Rodriguez PA (1999) Local field correction PIV: on the increase of accuracy of digital PIV systems. Exp Fluids 27:107–116
- Raffel M, Willert CE, Kompenhans J (1998) Particle image velocimetry: a practical guide. Springer, Berlin Heidelberg New York
- Scarano F, Riethmuller ML (1998) Advances in iterative multigrid PIV image processing. Exp Fluids [Suppl] 29:S51–S60

Wereley ST, Meinhart CD (2001) Second-order accurate particle image velocimetry. *Exp Fluids* 31:258–268  
Westerweel J (1993) Digital particle image velocimetry. PhD Thesis, Technical University Delft

Willert CE, Gharib M (1991) Digital particle image velocimetry. *Exp Fluids* 10:181–193