Particle Image Velocimetry with Optical Flow

Georges M. Quénot, Jaroslaw Pakleza
LIMSI-CNRS, BP 133
Université de Paris Sud, Bâtiment 508, 91403 Orsay Cedex, France.
Email: {quenot,pakleza}@limsi.fr

Tomasz A. Kowalewski
IPPT PAN, Centre of Mechanics
Polish Academy of Sciences, L 00-049 Warszawa.
E-mail: tkowale@ippt.gov.pl

Abstract

An Optical Flow technique based on the use of Dynamic Programming has been applied to Particle Image Velocimetry yielding a significant increase in the accuracy and spatial resolution of the velocity field. Results are presented for calibrated synthetic sequences of images and for sequences of real images taken for a thermally driven flow of water with a freezing front. The accuracy remains better than 0.5 pixel/frame for tested two-image sequences and 0.2 pixel/frame for four-image sequences, even with a 10% added noise level and allowing 10% of particles to appear or disappear. A velocity vector is obtained for every pixel of the image.

1 Introduction

The aim of this investigation is to explore the possibility of using an optical flow technique in measuring fluid flow velocity. Classical flow visualization is based on direct observation of tracer particles. Analysis of subsequent images searching for local displacements allows quantitative measurement of two-dimensional flow fields. The optical flow method offers a new approach for analysing flow images. It largely improves spatial accuracy and minimizes the number of spurious vectors. Application of this method may help in quantitative analyses of several challenging problems of fluid mechanics, as well as in full plane validation of their numerical counterparts.

1.1 Particle Image Velocimetry

Recently the experimental fluid mechanics technique of Particle Image Velocimetry (PIV) has proven to be a valuable method for quantitative, two-dimensional flow structure evaluation. It enables the measurement of the instantaneous in-plane velocity vector field within a planar section of the flow field. Due to the large accessible amount of quantitative vectorial velocity data, the PIV method is of great interest to engineers and researchers, allowing them the calculation of spatial gradients, dissipation of turbulent energy, spatial correlations, and the like. The classical PIV technique uses multiple-exposure images and optical autospectrum or autocorrelation analysis (Hesselink, 1988). Conventionally, PIV images are recorded on photographic film, and the flow field is obtained via the computation of the spatial correlation into small search regions. The point-by-point search analysis is repeated until the entire negative is analyzed. Processing large numbers of such images becomes a very laborious task. Therefore, an alternative approach - referred as Digital Particle Image Velocimetry (DPIV) - was introduced (Willert and Gharib 1991, Westerweel 1993). Images are recorded directly with a CCD camera and frame-grabber, and can be studied without the unnecessary delay and overhead associated with the scanning of photographs. The application of DPIV allows for a simple realization of the cross-correlation technique for pairs of two separate images. It removes the ambiguity of the sign of the displacement and improves signal dynamics.

The typical DPIV evaluation procedure is based on the analysis of two successive images of the flow. The digital images are decomposed into search windows (small square regions). Inside pairs of corresponding search windows, may be searched for as a global translation corresponding to the average spatial shift of particles, provided a flow is present in the illuminated plane. Usually a two-dimensional discrete Fourier transform (2D DFT) is used to facilitate the evaluation of the cross-correlation function. The location of the cross-correlation peak provides the mean displacement value, and its relative amplitude indicates the accuracy of the evaluation. The ratio of the displacement vs. the time scale between images gives the average velocity in the search window. The spatial resolution of the DPIV is presently limited by the quality of the available CCD, recording hardware and size of the search window. Decreasing the size of the search window improves the spatial resolution but at the cost of a loss on dynamic range and signal to noise ratio. Hence, an optimum window size must be found. Implementation of special techniques such as the time-series of single or multiexposure images, application of a Gaussian peak-fit estimator, dynamic search windows and final postprocessing may highly increase both the spatial resolution and dynamic range of the DPIV, resulting with approximately the same level of accuracy as a conventional PIV. Despite recent progress in the DPIV development, further improvement of the accuracy and minimization of the computational time still remains a current research goal (Lourenco and Krothapalii 1995, Sun et al. 1996).

One of the main drawbacks of classical DPIV is its inability to accurately resolve flow regions characterized by large velocity gradients. This is due to the strong deformation of the particle image pattern within a DPIV search window. Hence, several alternative evaluation methods have been proposed to remove the above limitation (Huang et al. 1993, Tokumaru and Dimotakis 1995, Gui and Merzkirch 1996). With this in mind, it appeared to us that an *optical flow* method may be an interesting alternative, offering high evaluation accuracy without most of the typical DPIV limitations. Conventionally, this technique was developed for detecting motion of large objects in a real world scene. The idea of this evaluation technique is in some sense similar to the Image Correlation Velocimetry proposed by Tokumaru and Dimotakis (1995).

In this article we describe our efforts to apply the optical flow technique in fluid mechanics, by evaluating displacements of small tracer particles conveyed by the flow. In the first part of the article the accuracy of the velocity measurements using the new implementation is investigated using synthetic image sequences generated with the help of a 2D numerical solution. Next the optical flow method is tested with experimental data collected for natural convection in water with phase change (freezing). Complex flow structures appearing due to the water density anomaly seem specially suited for testing the performance of these methods in regions with strong velocity gradients and colliding flow streams. Finally, results of the evaluation are discussed and compared with their DPIV counterparts.

1.2 Experimental details

The experimental set-up used to acquire flow fields consists of a 38mm cube-shaped convection box filled with water, a halogen tube lamp, a 24-bit frame grabber and a CCD colour camera. The flow is observed at central vertical cross section of the cavity using the light sheet technique. The halogen tube generates a 2 mm thick sheet of white light, which illuminates the selected cross-section of the flow.

The experiments are performed in two configurations (Figure 1). The first one is a differentially heated cavity. The motion is driven by a sudden temperature difference ($T_h = 10^oC$, $T_c = -7^oC$) applied to two opposite metal side walls of the cavity. The other four walls through which the flow is observed are made of plexiglas. The second configuration is a lid-cooled cavity. The top wall of the cube is isothermal at the temperature $T_c = -10^oC$, whereas the other five walls are non-adiabatic, allowing a heat flux from the external fluid surrounding the box. The temperature of the external bath is $T_h = 20^oC$.

Due to the temperature differences natural convection occurs in the box. At the cold wall, phase change (freezing of water) takes place, dynamically changing both thermal and kinematic boundary conditions of the flow. This is an additional non-linear coupling which futher complicates the prediction of the flow patterns. Despite the fact that freezing starts at a planar surface, the surface of the ice does not remain planar. Its distortion in turn affects the convection in the whole cavity. A complex interaction between the flow, the moving boundary and the latent heat released on the surface determines the flow pattern which is established.

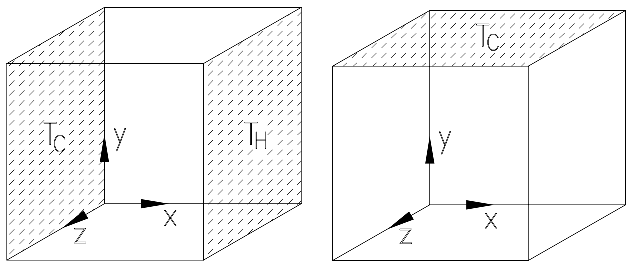

Figure 1: The cubical box. The y-axis is parallel to the gravity vector. Differentially heated cavity (left), Lid-cooled cavity (right).

The velocity fields are measured using two or more separately captured digital images taken at a constant time interval (typically 1 - 5s). Each of the images taken shows a relatively dense cloud of single illuminated particles. By applying thermochromic liquid crystals as seeding particles it is possible to collect at the same instant both velocity and temperature fields (Hiller et al., 1993).

DPIV is performed by cross-correlating the image pairs. The magnitude and direction of the velocity vectors are obtained by applying a series of 2D DFT for small sections (square search windows) of the whole image. Typically, the image of 480×480 pixels is divided into 48×48 (or 32×32) pixel matrices, which are spaced in half window intervals (partly overlapping each other). To improve the accuracy of the DPIV evaluation, special filtering techniques implementing local contrast enhancements were developed. Additionally, an oversampling technique is used, which doubles the dimension of the images through interpolated pixel values. The evaluation of the images is performed mainly on a Pentium PC (133MHz) running Linux OS. A typical DPIV analysis of one pair of images with oversampling takes about 10 minutes (for the 48×48 search window).

To obtain a general view of the flow pattern, several images are recorded periodically within a given time interval and are then superimposed in the computer memory. These images are similar to multiple exposure photographs, showing the flow structure (see Figure 12).

1.3 Optical Flow for DPIV

Optical flow computation consists in extracting a dense velocity field from an image sequence assuming that the intensity (or colour) is conserved during the displacement. Several techniques have been developed for the computation of optical flow. In a survey and a comparative performance study, Barron et al. (1994) classify them in four categories: differential, correlation based, energy based, and phase based. Not all of these are well suited for the DPIV problem. Many of these require long image sequences that are not easily obtainable experimentally and/or do not perform very well on the particle image texture (especially multi-resolution methods). The technique that we choose for the DPIV application was introduced by Quénot (1992) as the Orthogonal Dynamic Programming (ODP) algorithm for optical flow detection from a pair of images. It has been extended to be able to operate on longer sequences of images and to search for subpixel displacements (Quénot, 1996). The ODP based DPIV will be referred to as ODP-PIV. Compared with other optical flow approaches or to the classical correlation based DPIV, the ODP-PIV has the following advantages:

- It can be applied simultaneously to sequences of more than two images.
- It performs a global image match by enforcing continuity and regularity constraints on the flow field. This helps in ambiguous or low particle density regions.
- It provides dense velocity fields (neither holes nor border offsets).
- Local correlation is iteratively searched for in regions whose shape is modified by the flow, instead of being searched ni fixed windows. This greatly improves the accuracy in regions with strong velocity gradients.
- It is able to operate on multiband images.

2 Optical Flow using Dynamic Programming

Dynamic Programming is a very robust technique for searching optimal alignments between various types of patterns because it is able to include order and continuity constraints during the search. However, it is applicable only for the alignment of one-dimensional data sets (or arrays). This is because Dynamic Programming requires a natural topological order between data elements which does not exist in multi-dimensional data sets. Therefore its straightforward application to image matching is not trivial (Otha and Kanade, 1985; Adam et al., 1986; Burg et al., 1986).

2.1 The Orthogonal Algorithm

The algorithm is based on the search of a transformation that relates the second image to the first one and minimizes the $\mathcal{L}_1$ or $\mathcal{L}_2$ distance between them. $\mathcal{L}_n$ is the Minkowski distance: $(\sum_i \sum_j \mid I_0(i,j) - I_1(i,j) \mid^n)^{1/n}$. Minimizing the $\mathcal{L}_n$ distance and minimizing $\sum_i \sum_j \mid I_0(i,j) - I_1(i,j) \mid^n$ is equivalent. Therefore we will no longer consider the $1/n$ exponent in the $\mathcal{L}_n$ distance expression. The matching is global and does not require any previous segmentation or feature extraction. The main idea is to transform the search problem for two-dimensional displacements into a carefully selected sequence of search problems for one-dimensional displacements, thereby decreasing greatly the complexity.

Strip to strip matching

First, the two images are identically sliced into several parallel overlapping strips (Figure 2). Then, for every pair of strips, an optimal match is searched for with displacements allowed only in the slicing direction and identical for all the pixels in the same column in the orthogonal (here horizontal) direction (Figure 3).

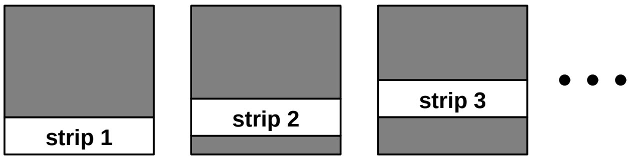

Figure 2: Image slicing.

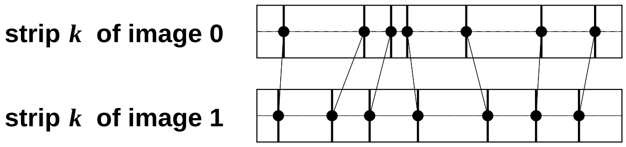

Figure 3: Strip alignment.

A dense field of displacements (between column vectors) is found for every pair of strips minimizing the distance $\mathcal{L}_1$ or $\mathcal{L}_2$ between them with the help of a dynamic programming algorithm. This gives us a displacement value at every point of the central fibre of all strips. Then, displacement values for all other pixels of the image are interpolated (or extrapolated) from the pixel values of the central fibre of the nearest strips. A dense displacement field is obtained for the whole image. This displacement field is then smoothed before the following steps of the algorithm are applied.

The main problem with such strip alignment is that it is possible that there is a shift between the strips in the direction orthogonal to the slicing one (the aligment is searched only in the slicing direction). An initial orthogonal shift between the strips may cause confusion between the patterns to be aligned. Our solution is to select the initial width to be large compared with the maximal expected orthogonal shift (typically a ratio of 2 to 5 is used). The dominating effect of an overlapping section within strips and the robustness of the dynamic programming to local perturbations allows us to obtain a good global alignment of the strip column vectors. The drawback of this approach is that the spatial resolution of the displacement field decreases with the increase of the strip width. This is why an iterative multi-resolution process is introduced.

Orthogonal iterations

The displacement field found in the first step is used to deform the second image relative to the first one. An image $I_1'(i,j)$ is built from the $(v_x(i,j), v_y(i,j))$ displacement field and the image $I_1(i,j)$ as $I_1'(i,j) = I_1(i+v_x(i,j), j+v_y(i,j))$. The image $I_1'(i,j)$ instead of $I_1(i,j)$ is compared and now aligned to $I_0(i,j)$. Generally, the $i + v_x(i,j)$ and $j + v_y(i,j)$ have non integer values. In this case, the $I_1(i+v_x(i,j), j+v_y(i,j))$ value is obtained by bilinear interpolation from the four nearest neighbors. Then, the previously described steps are repeated with the slicing performed in the orthogonal (vertical) direction and the alignment results are used to update and refine the $(v_x(i,j), v_y(i,j))$ displacement field. The combination of a horizontal and a vertical pass results in an alignment in both directions. Even though the horizontal pass provides only a low spatial resolution field, it significantly reduces the initial orthogonal shift between strips for the vertical pass. After both passes are executed, the initial orthogonal shift is reduced in both directions. To refine the accuracy of the matching result, the whole process is reiterated several times in a pyramidal fashion by reducing spacing and width of the strips (Figure 4).

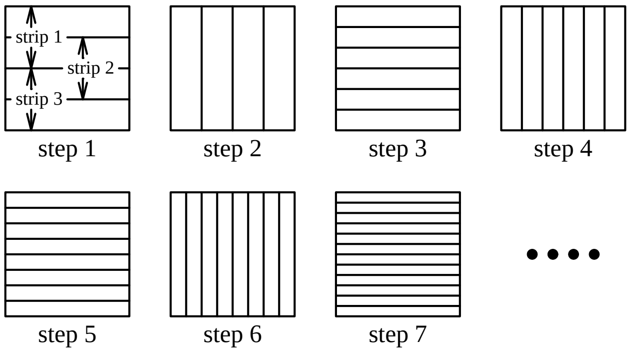

Figure 4: Strip spacing and width reduction.

At each horizontal-vertical iteration this alignment significantly reduces the orthogonal shift between corresponding strips, even if the spatial resolution of the alignment field is limited to the width of the strips used. It allows to reduce iteratively the strip width and spacing, *while* the orthogonal shift remains small relative to the strip width. Best results have been obtained with strips whose spacing is reduced from one fourth of the image size to 1 pixel and by applying a reduction factor of $\sqrt{2}$ at every horizontal-vertical iteration. The strip width is simultaneously reduced from one half of the image size down to 7 pixels.

2.2 Dynamic Programming

Dynamic Programming (DP) is used in the orthogonal algorithm because it appeared to be the most efficient way for performing an optimal strip to strip matching. However, any other efficient strip matching algorithm could be applied instead. The DP algorithm for performing strip matching is actually derived from a speech recognition algorithm. In speechr ecognition, segments of a speech signal are transformed into a 2D (time, frequency) strip representation. Then an optimal time alignment between them is searched using the DP algorithm (Sakoe and Chiba, 1978). Our image strips are matched exactly in the same way. The slicing direction corresponds to the time axis (where the alignment is performed) and the orthogonal direction corresponds to the frequency axis (where columns are fixed and moved globally).

Local matching residue

Strips of images 0 and 1 are represented by sequences of pixel intensity vectors: $s_0(i,p)$ and $s_1(j,p)$, with $0 \leq i \leq I$ and $0 \leq j \leq I$, $I+1$ being the number of pixels of the image in the slicing direction. Pixel intensity values are the components of the vectors, indexed by p with $-W/2 \leq p \leq W/2$, W being the width of the strip and the length of the vectors. A "local matching residue" between two p-indexed column vectors $(s_0(i,p))$ and $(s_1(j,p))$ is defined as:

$$d(i,j) = \sum_{p=-W/2}^{p=W/2} \alpha(p). \mid s_0(i,p) - s_1(j,p) \mid^n$$

($\mathcal{L}_n$ norm). Usually $n = 1$ or $n = 2$ is chosen. Here we use $n = 1$. The factor $\alpha(p)$ is introduced to limit the window effect (step function) in the direction orthogonal to the slicing. It reduces the effects of the initial orthogonal shift. It also reduces the "effective" width of the strip, however this effect can be diminished by starting with a larger initial width. We use the $\alpha(p) = 1 + cos(2\pi p/W)$ smoothing function.

Global matching residue

A "matching path" between two strips is searched within a neighbourhood of the $i = j$ diagonal corresponding to a maximum absolute displacement m. In order to allow outermost portions of the strips to float one relatively to another, the path should start at the $i+j = m$ line and end at the $i+j = 2I-m$ line. Also, it must specify a continuous and monotonic increasing function (Figure 5). A global matching residue is defined for each matching path as the sum of the local residua along it. An optimal matching path is defined as the one minimizing the global residue. It may be not unique. In that case, one optimal path is selected using additional heuristics like alternative for the shortest or least distorted path.

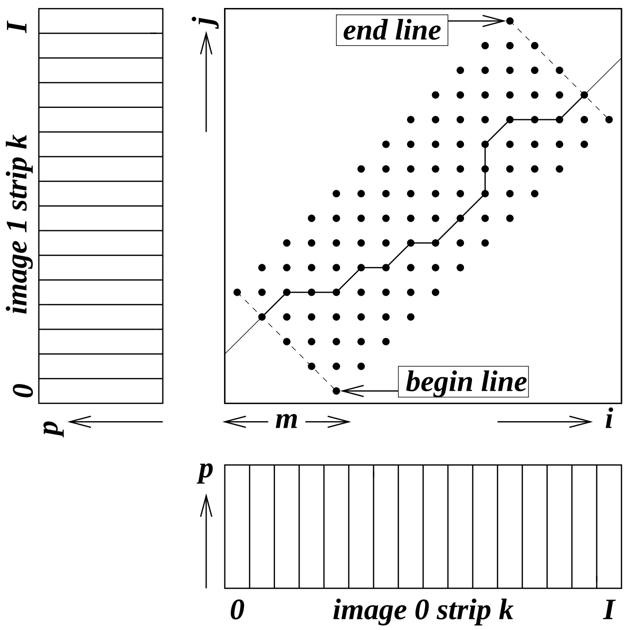

Figure 5: Dynamic Programming. Image strips and the matching path in the disparity matrix.

An (i,j)-optimal path between the begin line and the (i,j) point is defined as a path minimizing the sum of the local residua given for all the paths between these extremities. An "accumulated residue function" $D(i,j)$ is defined as the sum over an (i,j)-optimal path of the local matching function d. Due to the discrete form and the increasing and continuous character of the searched optimal paths, any (i,j)-optimal path must be an extension of one of the $(i-1,j)$-, $(i-1,j-1)$- or $(i,j-1)$-optimal paths. Moreover, the D function follows a recursive equation:

$$D(i,j) =$$

$$\min \begin{cases} D(i,j-1) + d(i,j-1) + d(i,j) \\ D(i-1,j-1) + 2 \times (d(i-1,j-1) + d(i,j)) \\ D(i-1,j) + d(i-1,j) + d(i,j) \end{cases}$$

We chose a sum computed on the matching path, seen as a parametric curve, with a $ds = dx + dy$ weight. Therefore, all paths have the same length. The D function may then be computed by recursion within the search area according to the following initial conditions:

$D(i,j) = 0$ if $i + j = m$
$D(i,j) = \infty$ for $\mid i-j \mid > m$ or $i + j < m$

The minimum of D on the $i + j = 2I - m$ line gives the end of the optimal path. Backtracking from this minimum along the "minimal" path with respect to the recursive equation provides the optimal path (or one of them in the case of non uniqueness). This path is extrapolated continuously outside of the begin and end lines. The alignment found, corresponding to the minimum of the sum, does not depend on the direction chosen for the computations. It is the same for left to right or right to left recursion. With each horizontal-vertical iteration the m parameter as well as the width of strips and spacing are decreased. Several enhancements made to the original ODP algorithm are detailed in the following sections.

2.3 Extension to multi-band images

Multi-band images are images where pixel values have several components (for instance: red, green and blue components for colour pixels). They can be seen as several independent monochrome images (bands). The ODP algorithm can be easily extended to the alignment of multi-band images simply by extending the pixel to pixel difference in the local matching residue definition. This extension is done by summing the differences per component. The local matching residue definition becomes:

$$d(i,j) = \sum_{p=-W/2}^{p=W/2} \alpha(p).\left(\sum_{b=1}^{b=B} \mid s_0(i,p,b) - s_1(j,p,b) \mid^n\right)$$

with b being the band index and B the number of bands. Everything else in the algorithm remains unchanged. The use of the colour information usually improves the quality of the obtained displacement field as compared to the use of intensity information alone.

2.4 Computation of the velocity field at intermediate times

The ODP method results with a displacement vector for each individual pixel of the image. Hence, the velocity field may be computed as v_0 at time t_0 of image 0 by relating image 1 to image 0, or as the velocity v_1 at time t_1 of image 1 for the reverse case (reversing the vector sign, of course). The velocity field can also be computed as v_λ at any intermediate time $t_\lambda = (1-\lambda) \times t_0 + \lambda \times t_1$ by relating both images to a fictitious image λ that would have been taken at the time t_λ. In fact there is a single "physical" velocity field but the actual set of numerical values $v(i,j)$ depends on the corresponding definition of the inter-image displacement. It is possible to define the velocity field relative to the "from" extremity, relative to the "to" extremity or relative to an intermediate point defined by the parameter λ. In practice, due to the small interpixel distance, any representation may be chosen for the computation. In our application we chose the intermediate representation corresponding to $\lambda = 0.5$.

The optimal path inside the disparity matrix (Figure 5) may be written as: $i_0 = i_\lambda - \lambda \times v_\lambda(i_\lambda)$ and: $i_1 = i_\lambda + (1-\lambda) \times v_\lambda(i_\lambda)$. If $\lambda = 0$ or $\lambda = 1$ we fall in one of the original cases (computation at time t_0 ot t_1). For other values of λ, i_λ has a non-integer value. A simple linear interpolation gives v_λ at any i_λ integer position. During the ODP iterations, images to be sliced and aligned are the source images 0 and 1 transformed to the λ image index using the $-\lambda \times v_\lambda$ and $(1-\lambda) \times v_\lambda$ velocity fields respectively.

The high spatial resolution of the resulting displacement field permits us to compute virtual images at intermediate time steps. It allows us to construct a continuous sequence of frames (movie) for the analyzed time period. Replaying such a sequence of images on the computer screen is very helpful for a visual analysis of complex flow structures.

2.5 Extension to image sequences

The original ODP algorithm operates by using only a pair of images. When a longer image sequence is available, a pair of images has to be manually selected to get optimal displacements. A natural extension is to automatically select a pair of images within the sequence. The optimal way to do this is to change the selected pair during the execution of the algorithm. Simultaneously to the reduction of the spacing and width of the strips, the spacing of the images may be increased up to an optimal value that is adaptively determined by the allowed maximum displacement or by the available number of images. In this way, the dynamic range of the method can be increased enormously, limited only by the flow relaxation time.

2.6 Search for subpixel displacements

The original ODP algorithm is able to search only for integer (in pixels) displacements at each iteration. Due to the applied interpolation and smoothing procedure, our algorithm is already able to find subpixel displacements. An additional accuracy improvement is still possible through the direct search of subpixel displacement components during strip alignment. This is achieved by shrinking around the diagonal line the set of points used for the dynamic programming computations (Figure 6). This corresponds to searching for a matching column vector in the strip having a non-integer displacement (still a discrete set but with less than 1 pixel spacing). The main effect of this is that the points at which local residua are computed are moved out of the pixel grid. Hence, the corresponding segments obtain new non-integer coordinates. These segments have to be interpolated from the nearest segments of integer coordinates.

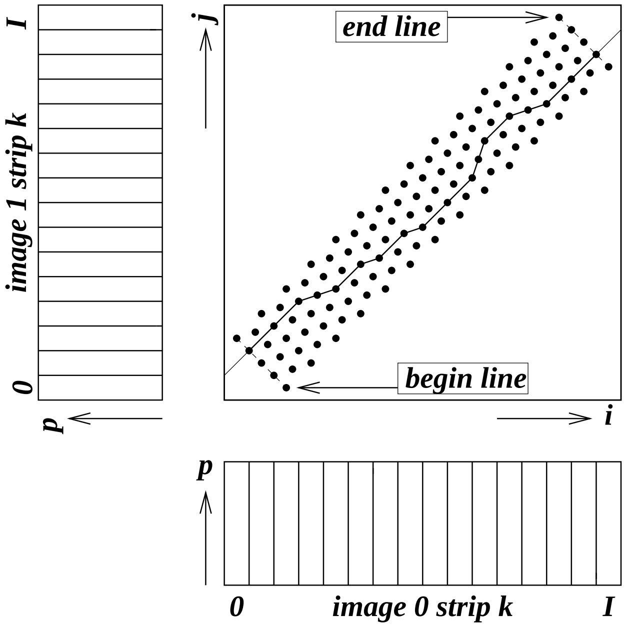

Figure 6: Subpixel resolution search. The point set no longer forms a square grid, it is also no longer aligned with pixel positions in the image strips.

Within the original algorithm, the spacing and width of the strips and the width of the search window around the diagonal are reduced by a factor of $\sqrt{2}$ at each horizontal-vertical iteration. The subpixel iterations are added after the iterations of the original algorithm. At this point the spacing, width of the strips and the width of the search window around the diagonal are kept constant at their minimal value. Here, ten additional subpixel iterations are performed, reducing each time the distance of the points to the diagonal line of the disparity matrix by a factor of $\sqrt{2}$. The subpixel iterations significantly improve the accuracy of the computed velocity field.

2.7 Use of multi-image distances

When more than two images are available, it is possible to select adaptively a pair of images during the ODP iterations optimizing an accuracy / distortion compromise (section 2.5). When the selected images are separated by one or more intermediate images, it is suitable to add the constraint that all the strips of the intermediate images or at least a few of them are aligned with the strips of the outermost images (Figure 7). This was achieved by the following generalization of pixel to pixel distance: $\mid p_1 - p_0 \mid$ to a "multi-pixel distance" defined either by: $(\max_{i=0}^{i=N-1} p_i) - (\min_{i=0}^{i=N-1} p_i)$ or: $\Sigma_{i=0}^{i=N-2} \mid p_{i+1} - p_i \mid$. Like for the subpixel iterations, the segments of the intermediate images generally have non integer coordinates and have to be interpolated from the nearest integer coordinates.

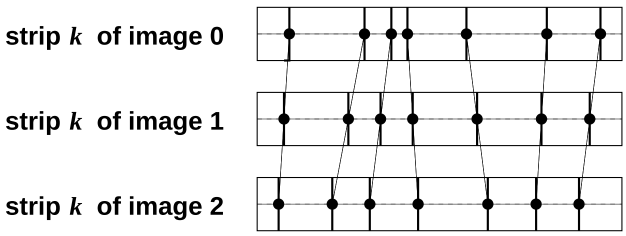

Figure 7: Multi-image strip alignment.

2.8 Added smoothed bands

An additional enhancement specific for the DPIV problem has been implemented to the ODP algorithm. It is based on adding additional smoothed bands to the original (mono-band) image. Each new band is derived from the previous one by smoothing it with a 3×3 Gaussian-like filter. We intuitively guessed that this would help by reducing the local texture ambiguity and by enlarging the spectral structure of the images (rather concentrated around high frequencies). Compared with a simple smoothing, adding bands does not imply any loss of information. This is possible since the ODP algorithm is able to operate on multiband images. As expected, this option yields a significant improvement in the quality of the results. We found that the optimal number of added bands was between 3 and 5.

3 Calibration on synthetic images

The ODP algorithm for optical flow computation has been benchmarked already for standard image sequences available at the public computer domain (`ftp://csd.uwo.ca/pub/vision`). Its performance appeared to be at least as good as or better than the one obtained using classical optical flow methods (Quénot, 1996). The images used for flow velocimetry are different in several aspects. They are characterized by specific non-continuous textures of particles, and recorded displacement fields are rather different from those obtained from typical image scenes. Hence, to perform the accuracy tests an appropriate set of benchmark images is required for DPIV and ODP-PIV algorithms. With this in mind, a complete set of the test sequences (synthetic and real) was developed for the evaluation of the ODP-PIV technique. It is now available for comparative evaluations from the public LIMSI ftp server at: `ftp://ftp.limsi.fr/pub/quenot/opflow/testdata/piv`.

Several test sequences of four images were generated using a synthetic image representative of the random particle texture and a velocity field taken from a numerical solution obtained for two-dimensional flow around pair of cylinders (Lu 1996). These images, numbered from 0 to 3, represent particle textures calculated at four time steps: $-3/2$, $-1/2$, $1/2$ and $3/2$. The velocity field v is defined (and searched for) in the "central" image of the sequence (which has the image index 3/2). Images 0, 1, 2 and 3 are generated by applying the velocity fields $-3v/2$, $-v/2$, $+v/2$, and $+3v/2$ respectively to the central image. The central image is not a part of the sequence.

The sequence labelled "Perfect" is generated ideally from the velocity field and the texture image. The sequences labelled "Noise $N\%$" are identical to the "Perfect" sequence except that the intensity of all four images is modulated by adding to each pixel randomly generated noise from the interval $[-255.N/100, 255.N/100]$ (with saturation in case of overflow or underflow). The sequences labelled "Add/rm $N\%$" are identical to the "Perfect" sequence except that $N\%$ of the particles are randomly removed and $N\%$ of other particles are randomly added between the first and the last image. For the intermediate images, these particles fade gradually between on and off. This simulates the effect of the third velocity component of the physical flow that conveys particles across the light sheet. Though particle appearance/disappearance and noise are obviously not related, sequences mixed with the "Noise $N\%$" and "Add/rm $N\%$" effects added are generated and labelled "Mixed $N\%$" simply to set reference points with both perturbations.

For all sequences the interior of the circles that represent the cylinders around which the particles flow are filled with a fixed texture distinct from the particle texture. The motion at these boundaries and inside the cylinders must be zero. The original velocity field leads to a mean displacement module of 7.58 pixels/frame and a maximum displacement of 13.5 pixels/frame for the default sampling period. The "Perfect" and "Mixed $N\%$" test sequences have also been generated with higher displacements, scaled by a factor of 1.5 and 2. This is equivalent to scaling the velocity field values or the sampling period by the same factor. The mean displacement module becomes, respectively, 11.4 pixels/frame (maximum displacement 20.2 pixels/frame) and 15.2 pixels/frame (maximum displacement 27.0 pixels/frame).

Figure 8 shows one image of a synthetic sequence and the original velocity field used to generate the synthetic sequences. For clarity, in the figure, the number of vectors is reduced 8 times in both directions and their magnitude is doubled.

The sequences of synthetic images were evaluated using the classical 2D DFT based DPIV method and the new ODP-PIV algorithm. For the cross-correlation DPIV two search window sizes were applied: 32×32 (DPIV32) and 48×48 (DPIV48). For the ODP-PIV evaluation three variants were investigated. The first one (ODP2) uses only 2 images (indices 1 and 2) like in the classical DPIV. The second (ODP4S) and third (ODP4M) variant use 4 images (indices 0, 1, 2 and 3). They differ in the pixel to pixel distance definition. The ODP4S variant uses the $d(p_0, p_1, p_2, p_3) = \Sigma_{i=0}^{i=2} \mid p_{i+1} - p_i \mid$, whereas the ODP4M variant uses the $d(p_0, p_1, p_2, p_3) = (\max_{i=0}^{i=3} p_i) - (\min_{i=0}^{i=3} p_i)$ definition of the multi-pixel distance. For all ODP-PIV variants subpixel iterations and the added smoothed bands option (5 added bands for the ODP2 and 3 added bands for the ODP4S and ODP4M) were used.

For each evaluated case the velocity error is computed for the whole image (no border offset), except at the inner surface of the circles the particles are moving around. The mean velocity error and its standard deviation (±) are collected in Tables 2, 3 and 4. Table 1 displays the angle error as defined by Barron et al. (1996). This is the angle (in degrees) between the correct and computed 2-component (v_x, v_y) vectors representing the flow field. Table 2 displays the absolute displacement error which is given as the $\mathcal{L}_1$ norm of the difference between the correct and computed displacement in pixels/frame. Tables 1 and 2 display the error for the original velocity field which leads to a mean displacement module of 7.58 pixels/frame and a maximum of 13.5 pixels/frame. Table 3 and 4 display error results for the scaled velocity fields (mean displacement modules of 11.4 pixels/frame and 15.2 pixels/frame respectively). Hence, the absolute displacement error is also scaled by the factor of 1.5 and by the factor of 2 respectively.

Figure 9 shows the velocity field obtained using ODP-PIV (for the mixed 20% test, ODP4M) and the error as the module of the difference between the correct and evaluated velocity field displayed in form of the grayscale image (non linear scale). Table 5 gives the error histogram for the three ODP variants.

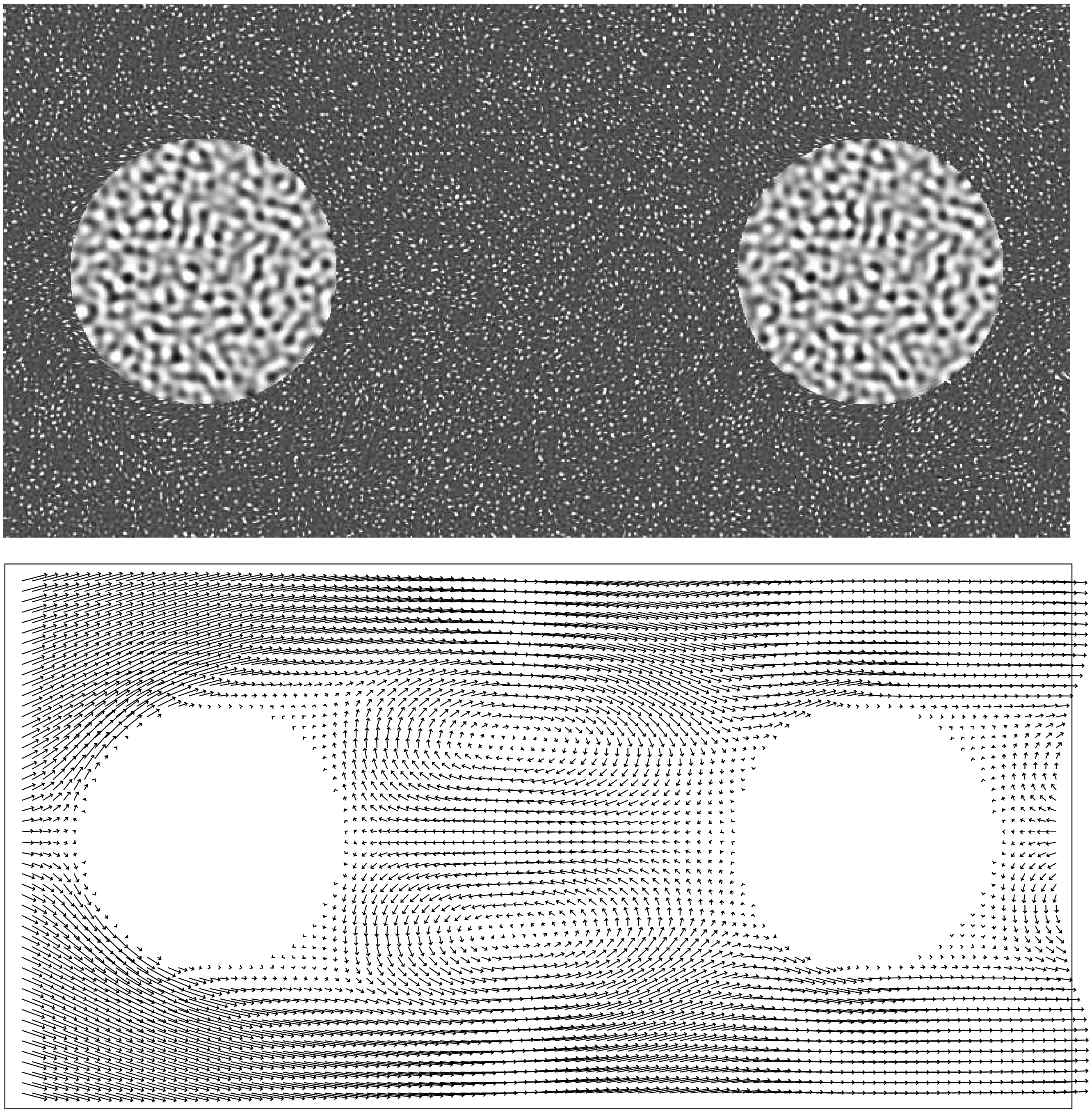

Figure 8: Synthetic particle image (top) and the "Correct" velocity field, magnified twice, (bottom). Average velocity module is 7.58 pixel/frame.

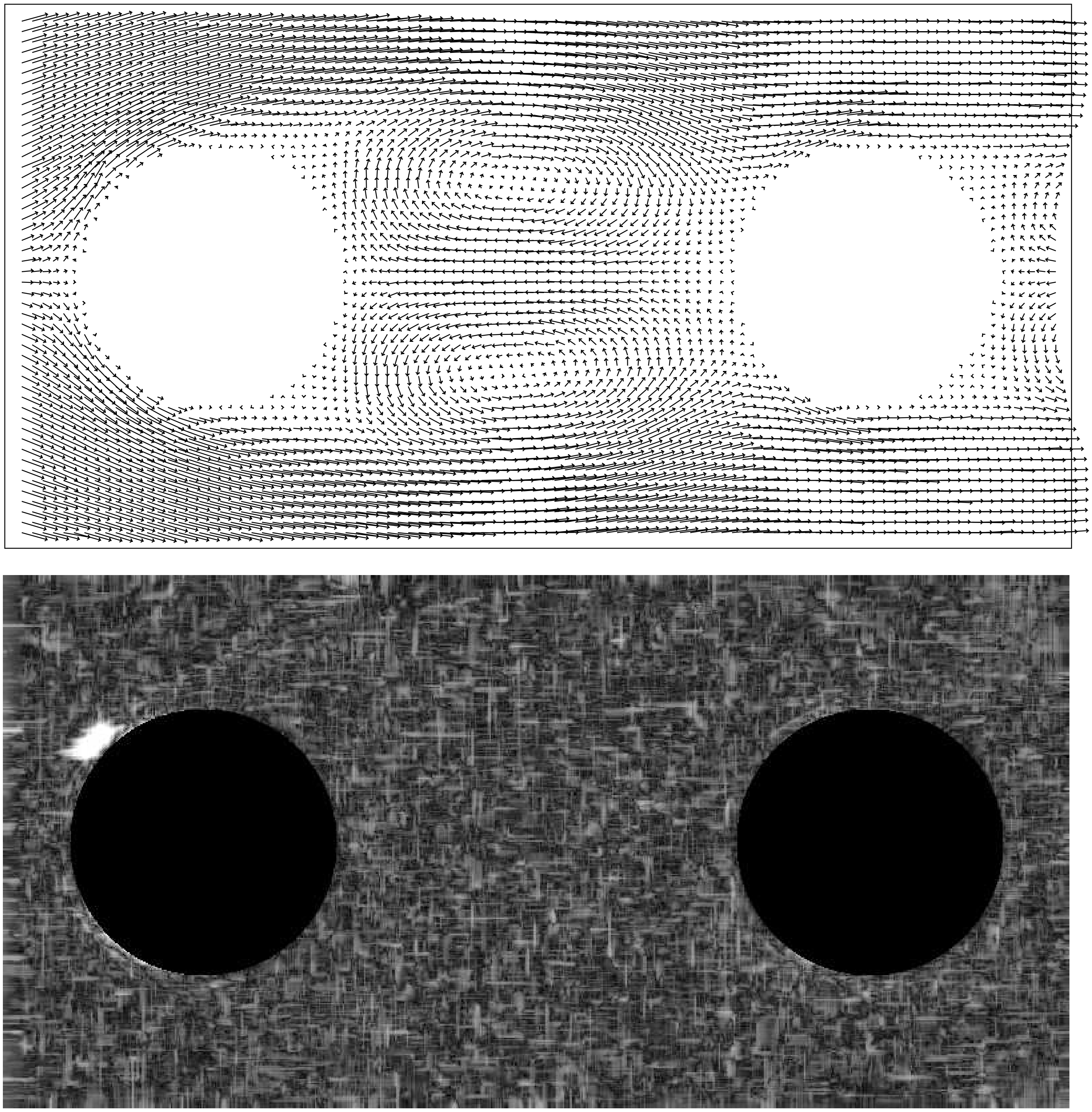

Figure 9: Velocity field, magnified twice, obtained with the OPD 4M variant from the 4-image “mixed 20%” syhthetic sequence (top) and distribution of the corresponding velocity error (bottom), bright spots indicate regions of larger error (non-linear scale).

	DPIV32	DPIV48	ODP2	ODP4S	ODP4M
Perfect	5.95±13.9	9.35±18.3	1.23±2.24	1.07±4.05	0.62±1.44
Noise 5%	6.49±14.6	9.69±19.0	1.83±3.84	0.91±1.96	0.78±1.59
Noise 10%	8.75±17.9	10.8±20.0	4.01±10.8	1.40±4.01	1.01±1.83
Noise 20%	35.0±35.5	31.0±30.4	6.70±11.8	2.36±5.39	1.77±2.88
Add/rm 5%	6.04±13.8	9.35±18.3	1.27±2.35	0.77±1.81	0.67±1.50
Add/rm 10%	5.94±13.5	9.52±18.5	2.61±9.94	1.13±4.29	0.73±1.60
Add/rm 20%	6.11±14.2	9.77±19.2	1.42±2.54	1.45±5.23	0.86±1.81
Mixed 5%	6.40±14.4	9.59±19.0	1.77±2.87	1.24±4.10	0.84±1.67
Mixed 10%	10.2±19.6	11.3±20.8	4.30±11.7	1.58±4.49	1.20±2.14
Mixed 20%	40.8±34.5	38.3±29.7	6.15±9.01	3.46±8.89	2.33±3.69

Table 1: Angle error with a mean displacement module of 7.58 pixels/frame

	DPIV32	DPIV48	ODP2	ODP4S	ODP4M
Perfect	0.55±0.94	0.87±1.46	0.13±0.10	0.13±0.54	0.07±0.07
Noise 5%	0.61±1.18	0.86±1.49	0.21±0.46	0.10±0.13	0.08±0.08
Noise 10%	0.77±1.57	0.91±1.59	0.53±1.44	0.17±0.53	0.11±0.09
Noise 20%	3.11±4.14	2.06±2.88	0.88±1.58	0.30±0.68	0.20±0.14
Add/rm 5%	0.55±0.90	0.86±1.45	0.14±0.11	0.08±0.11	0.07±0.08
Add/rm 10%	0.55±0.93	0.87±1.47	0.34±1.28	0.14±0.56	0.08±0.09
Add/rm 20%	0.56±0.99	0.88±1.52	0.16±0.12	0.18±0.69	0.10±0.10
Mixed 5%	0.60±1.12	0.86±1.51	0.20±0.13	0.15±0.53	0.09±0.08
Mixed 10%	0.91±1.89	0.93±1.66	0.57±1.71	0.20±0.59	0.13±0.11
Mixed 20%	3.73±4.39	2.49±3.19	0.74±0.52	0.43±1.08	0.27±0.22

Table 2: Absolute displacement error with a mean displacement module of 7.58 pixels/frame

	ODP2	ODP4S	ODP4M
Perfect	0.29±1.08	0.51±1.87	0.37±1.37
Mixed 5%	0.38±1.24	0.57±1.95	0.41±1.40
Mixed 10%	0.64±1.59	0.61±1.95	0.44±1.42
Mixed 20%	1.20±2.13	0.96±2.47	0.85±2.16

Table 3: Absolute displacement error with a mean displacement module of 11.4 pixels/frame

	ODP2	ODP4S	ODP4M
Perfect	0.34±1.27	1.92±3.80	9.17±10.3
Mixed 5%	0.67±2.23	2.01±3.84	8.91±9.85
Mixed 10%	0.82±2.21	2.57±4.21	8.68±9.35
Mixed 20%	1.53±2.84	3.38±4.62	23.3±17.8

Table 4: Absolute displacement error with a mean displacement module of 15.2 pixels/frame

Error	ODP2	ODP4S	ODP4M
$0.0 \leq e < 0.1$	21.6%	32.7%	32.6%
$0.1 \leq e < 0.2$	5.3%	23.2%	22.3%
$0.2 \leq e < 0.3$	7.5%	18.6%	17.8%
$0.3 \leq e < 0.4$	8.2%	11.2%	11.8%
$0.4 \leq e < 0.5$	8.2%	6.0%	7.2%
$0.4 \leq e < 0.6$	7.7%	3.0%	4.1%
$0.6 \leq e < 0.7$	7.0%	1.3%	2.1%
$0.7 \leq e < 0.8$	6.1%	0.6%	1.1%
$0.8 \leq e < 0.9$	5.1%	0.3%	0.5%
$0.9 \leq e < 1.0$	4.3%	0.1%	0.2%
$e \geq 1.0$	19.0%	3.0%	0.4%

Table 5: Absolute displacement error histogram with a mean displacement of 7.58 pixels/frame, mixed 20% test sequence

There are some interesting differences between the collected DPIV and ODP-PIV results. For the classical DPIV the vector set is sparse (about 0.3% density), whereas it is dense (100% density) for the ODP-PIV. The DPIV method appears to be rather insensitive to particle appearance/disappearance, but on the other hand, it is very sensitive to the introduced noise. As one would expect for the classical DPIV method, expanding the search window size increases its robustness to noise but, simultaneously, decreases the accuracy. The ODP-PIV appears equally sensitive to both introduced disturbances, but its performance is far better than that of the DPIV in all investigated cases.

We may notice that there is not a strict monotonic increase of the error relative to perturbations for the ODP-PIV. For instance, in Table 2, "Perfect−ODP4S" test is worse than expected and "add/rm 20%−ODP2" is better than expected. If some erroneous or correct matches randomly occur, they are amplified by the global constraints. This implies that, for any ambiguous region, the matching is correct or wrong globally. Spurious vectors are concentrated in spurious regions (comp. Figure 9). They can be easily detected by inspecting the appearance of the velocity field and animations constructed using the extracted velocity field and the original images.

4 Results on real sequences

Numerical velocity fields obtained using a finite differences code (Yeoh, 1993), and their DPIV counterparts are displayed for a differentially heated cavity in Figure 10. A freezing front advances from the cold left wall. Two main recirculating flow regions can be observed: clockwise in the lower - right part of the cavity and counter-clockwise in the upper region. A sudden change of the flow direction in the right lower corner, as well as a colliding region of both recirculations on the left side are difficult to accurate evaluate by the traditional DPIV. This is mainly due to the relatively large search windows used (48 pixel). The direction and magnitude of flow velocity change immensely within the window, hence locally the error of the evaluation becomes unacceptably high. On the other hand, further decrease of the window size deteriorates signal statistics necessary for effectiveness of the DFT method.

The two pairs of images from the experiment (Figure 10) are analyzed using the ODP-PIV method. Figure 11 shows the velocity fields obtained from the ODP2 variant (2 images are used for these sequences). The original velocity fields are dense (1 vector for each individual pixel). For clarity, in the figure, the number of vectors is reduced 12 times in both directions, and their magnitude is doubled. At time step 660s, the evaluated velocity module averaged over the whole image equals about 4.2 pixels/frame (it turns into about 5.5 pixels/frame for the area where the flow is present). The maximum velocity module equals 15.0 pixels/frame. At time step 100min the flow apparently slows down. The average velocity equals about 2.5 pixels/frame (i.e. about 5.0 pixels/frame for the flow area) and the maximum evaluated velocity equals 12.2 pixels/frame. According to our calibration, we estimate that the average accuracy of the evaluated vectors is better than 0.5 pixel/frame. The vector field looks quite reasonable except for some small spots at the 100min time step (it appears that the particle density is very low in the corresponding area of the images). The improvement in comparison with the DPIV evaluation is obvious.

The ODP-PIV method is used for a four image sequence taken from the second experiment with the lid cooled cavity. A conical ice crystal develops at the cavity top wall. The four images are taken every 200ms, at 308s after the temperature difference $T_h - T_c$ is set (Figure 12).

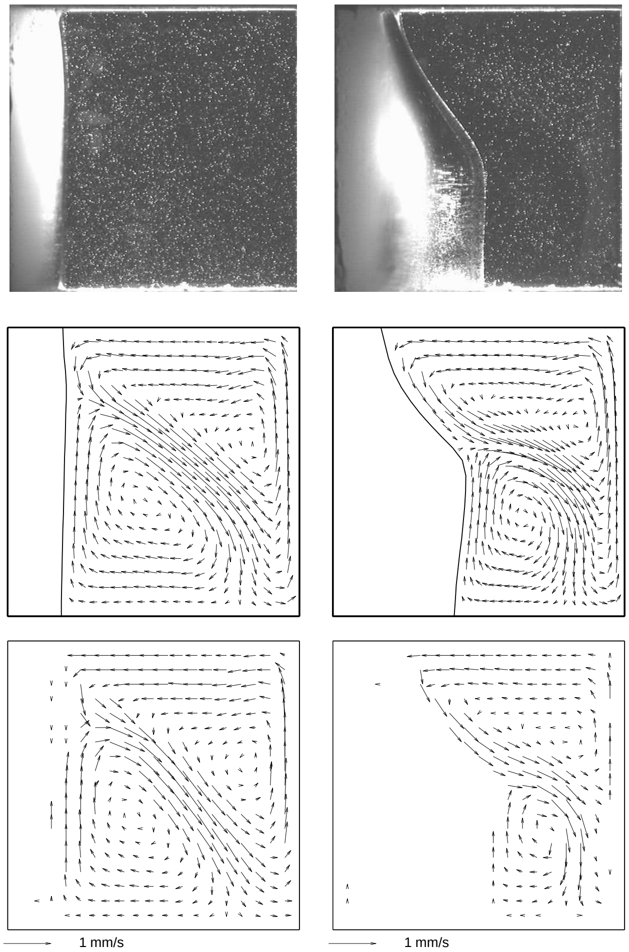

Figure 10: Freezing in the differentially heated cavity. Velocity fields at two time steps: 660s (left) and 100min (right) after initiating the flow; 496 × 496 pixel particle images (top row), numerical calculations (middle row), DPIV evaluated vectors (bottom row), for pairs of images taken at 1.5s intervals

Figure 12 shows one of the image of the sequence, fifteen similar images superimposed and the velocity fields obtained from the ODP-PIV using the ODP2, and the ODP4S variants. The average module of the extracted velocity field is about 4.4 pixels/frame with a maximum of 27.4 pixels/frame. According to our calibration, we estimate that the average accuracy is below 0.5 pixel/frame for the ODP2 variant and below 0.2 pixel/frame for the ODP4S and the ODP4M variants.

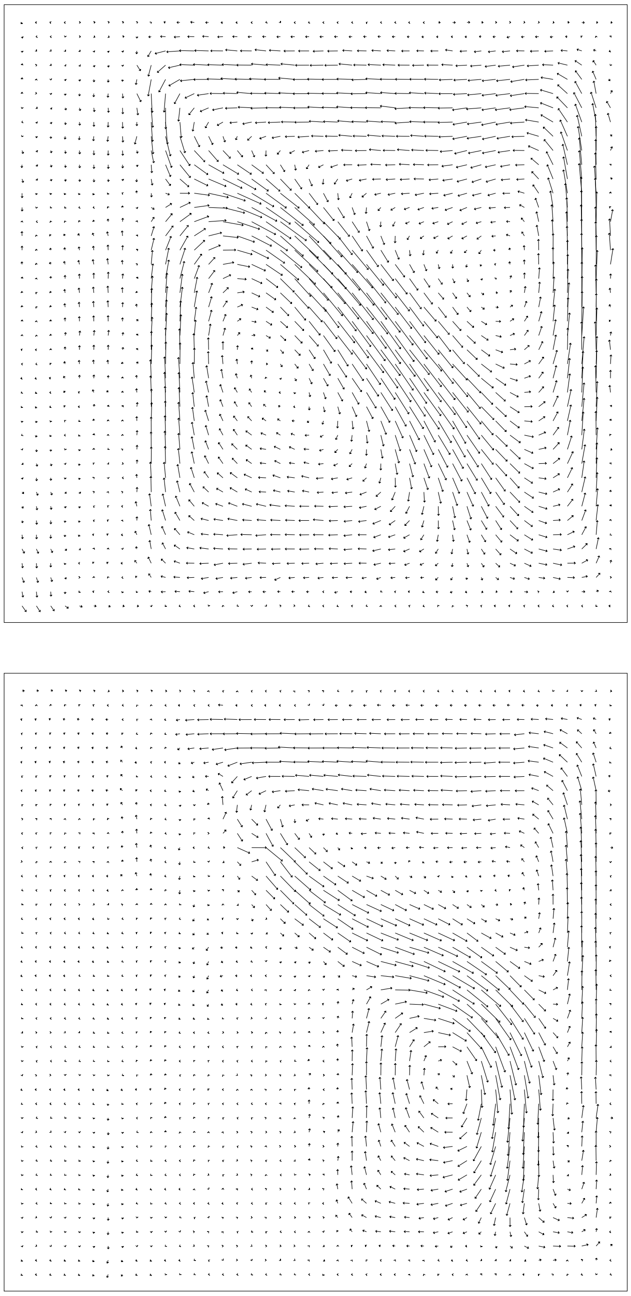

Figure 11: ODP-PIV results for the 2-image real sequences. Velocity field at 660s (left) and 100min (right).

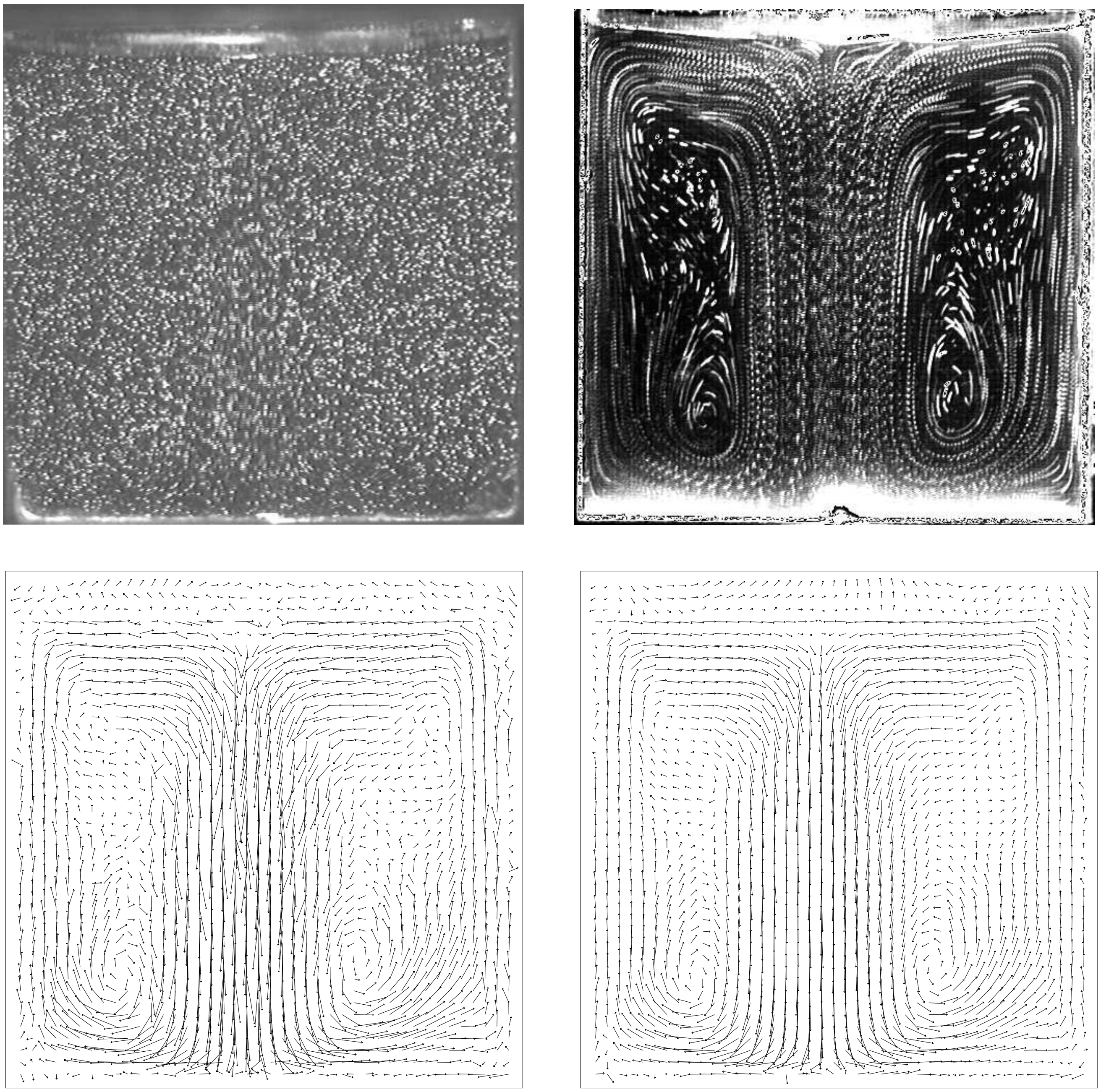

Figure 12: Freezing in the lid cooled cavity. Results for a 4-image real sequence. Particle image (top left), 15 superimposed images (top right), Velocity field obtained with the ODP2 variant using 2 images (bottom left) and with the ODP4S variant using 4 images (bottom right), non linear scale.

The vector field appears more accurate with the ODP4S variant than with the ODP4M variant (not shown). This has been observed with several other 4-image sequences and it is not consistent with the calibration on synthetic sequences which indicates better results for the ODP4M variant. This may come from the fact that in the real experiments there is a weak constant background that does not follow the particle flow against which the ODP4S variant is more robust. The calibration experiments also showed that the ODP4S variant is more robust for large displacements.

For $N \times N$ images, the asymptotic computation time is $O(N^3 \log N)$ for all the ODP variants but there is a portion of the algorithm with a $O(N^2 \log N)$ computation time that still has a significant impact for N values within a few hundreds. Practically, doubling the image size N increases the computation time by roughly six times. It is possible to turn off some quality options and significantly reduce (by up to three times) the computation time at the price of loss in accuracy. Using a 250 Mhz R4400 Indigo2 workstation for 496×496 images the ODP2, ODP4S and ODP4M variants take 20, 210 and 200 minutes CPU time, respectively. Similar computation times were obtained for Pentium 200MHz under Linux OS.

5 Summary and Conclusions

An Optical Flow technique based on the use of Dynamic Programming has been successfully applied to Particle Image Velocimetry yielding a significant increase in the accuracy and spatial resolution of the velocity field. Results have been presented for calibrated synthetic sequences and for real sequences from the experiment on natural convection in freezing water. Using the ODP-PIV a dense velocity vector field for every pixel of the image is obtained. The accuracy is better than 0.5 pixel/frame for two-image sequences and below 0.2 pixel/frame for four-image sequences even with a 10% noise level and a 10% rate of appearance and disappearance of particles. Though computation time of the ODP-PIV method is rather long (compared to classical DPIV), its high accuracy and high spatial resolution allows us to use it for code validations – which nowadays has become a very important task in fluid mechanics.

Future work will be conducted in three directions:

1. Better characterization of the result quality: statistical estimation of the accuracy of the velocity field from the particle density (or texture), the smoothness of the extracted field and the reconstruction error.
2. Improvement of the measuring process: optimal choice of the average velocity value (through time interval between images) and the particle density to minimize the relative error.
3. Direct search of the three-dimensional velocity field for which Dynamic Programming based Optical Flow search is very well suited.

Acknowledgements

We acknowledge support from the Conjoint CNRS-PAN Project No. 2930. The third author acknowledges support of the State Scientific Committee (KBN Grant No. 3P40400107).

References

Adam P; Burg B; Zavidovique B (1986) Dynamic Programming for Region Based Pattern Recognition, Proc. IEEE ICASSP, pp. 2075-2078 (April 1986, Tokyo, Japan).

Barron JL; Fleet DJ; Beauchemin SS (1994) Performance of Optical Flow Techniques, International Journal of Computer Vision, 12:43-77.

Burg B; Missakian P: Zavidovique B (1985) Pattern Recognition through Dynamic Programming, SPIE's 29th Annual Internal Technical Symposium (July 1985, San-Diego,CA).

Hesselink L (1988) Digital Image Processing in Flow Visualization, Ann. Rev. Fluid Mech. 20:421-485.

Hiller WJ; Koch S; Kowalewski TA; Stella F (1993) Onset of natural convection in a cube, Int. J. Heat Mass Transfer, 36:3251-3263.

Lu HZ (1996) Simulation des écoulements externes visqueux incompressibles par une méthode de couplage: Différences finies Particulaires, Notes et documents LIMSI No 96-10.

Otha Y; Kanade T (1985) Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming, IEEE Trans. Pattern Analysis and Machine Intelligence 7:139-154.

Quénot GM (1992) The “Orthogonal Algorithm” for Optical Flow Detection using Dynamic Programming, Proc. IEEE ICASSP, Vol III, pp. 249-252 (March 23-26, San-Francisco, CA).

Quénot GM (1996) Computation of Optical Flow using Dynamic Programming, IAPR Workshop on Machine Vision Applications (Nov 12-14, Tokyo, Japan).

Sakoe H; Chiba S (1978) Dynamic Programming Optimization or Spoken Word Recognition, IEEE Trans. Acoust., Speech, and Signal Proc 26:43-56.

Yeoh GH (1993) Natural convection in a solidifying liquid, Ph.D. thesis, University of New South Wales, Kensington, Australia.

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

Westerweel J (1993) Digital Particle Image Velocimetry - Theory and Application, Delft: Delft University Press.

Lourenco L; Krothapalii A (1995) On the accuracy of velocity and vorticity measurements with PIV. Exp Fluids 18: 421-428

Huang HT; Fiedler HE; Wang JJ (1993) Limitation and improvement of PIV. Part I: Limitation of conventional techniques due to deformation of particle image patterns. Exp Fluids 15: 168-174

Sun JH; Yates DA; Winterbone DE (1996) Measurement of the flow field in a diesel engine combustion chamber after combustion by cross-correlation of high-speed photographs. Exp Fluids 20: 335-345

Gui LC; Merzkirch W (1996) A method of tracking ensembles of particle images. Exp Fluids 21: 465-468

Tokumaru PT; Dimotakis PE (1995) Image correlation velocimetry. Exp Fluids 19: 1-15