Exp Fluids (2011) 50:135–147
DOI 10.1007/s00348-010-0907-z

RESEARCH ARTICLE

Integrating cross-correlation and relaxation algorithms for particle tracking velocimetry

W. Brevis · Y. Niño · G. H. Jirka

Received: 18 November 2008 / Revised: 12 May 2010 / Accepted: 25 May 2010 / Published online: 12 June 2010

Abstract An integrated cross-correlation/relaxation algorithm for particle tracking velocimetry is presented. The aim of this integration is to provide a flexible methodology able to analyze images with different seeding and flow conditions. The method is based on the improvement of the individual performance of both matching methods by combining their characteristics in a two-stage process. Analogous to the hybrid particle image velocimetry method, the combined algorithm starts with a solution obtained by the cross-correlation algorithm, which is further refined by the application of the relaxation algorithm in the zones where the cross-correlation method shows low reliability. The performance of the three algorithms, cross-correlation, relaxation method and the integrated cross-correlation/relaxation algorithm, is compared and analyzed using synthetic and large-scale experimental images. The results show that in case of high velocity gradients and heterogeneous seeding, the integrated algorithm improves the overall performance of the individual algorithms on which it is based, in terms of number of valid recovered vectors, with a lower sensitivity to the individual control parameters.

W. Brevis (✉) · G. H. Jirka
Institute for Hydromechanics, Karlsruhe Institute of Technology,
Kaiserstr. 12, 76128 Karlsruhe, Germany
e-mail: wernher.brevis@kit.edu

Y. Niño
Department of Civil Engineering and Advanced Mining
Technology Center, Universidad de Chile,
Blanco Encalada 2002, Santiago, Chile

1 Introduction

Lately, velocity measurements based on image analysis have become a standard for experimental research in fluid mechanics. There are several techniques for the determination of instantaneous velocity fields, but in general, most of them use solid particles as the basic tracer element. Depending on the frame of reference of the measurements, the methods can be classified in two main groups, particle image velocimetry (PIV), which determines the velocity field in an Eulerian frame of reference, and particle tracking velocimetry (PTV), which works in a Lagrangian one. In case of PTV, the velocity is determined at each particle position, an important characteristic for research dealing with the flow description from a Lagrangian point of view.

An important aspect in PTV analysis is that both, the identification of the particle centroids and the matching algorithms, have a strong influence on the quality and quantity of recovered spatiotemporal information. The errors associated with the determination of the particle centroids are particularly important when the particle displacements are of the same order of the particle image size. If the displacements are long enough to avoid the influence of the centroid-detection errors, nonlinear deformation of the particles patterns can be recorded, especially in regions of high velocity gradients, making it difficult for the temporal matching algorithms to find valid correspondences.

Historically, PTV has been classified as suitable for images with a low particle density, where density is defined as the number of particles per pixel, but the current algorithms for particle detection, such as the particle mask correlation method (Takehara and Etoh 1998) and the dynamic threshold binarization (Ohmi and Li 2000), combined with current algorithms to solve for the temporal matching problem, have provided good performance for images with a moderately high particle density. Algorithms that solve the temporal matching problem include those based on Binary Cross-Correlation (BCC) (Uemura et al. 1989; Hassan et al. 1992), deterministic annealing (Stellmacher and Obermayer 2000), the velocity gradient tensor (Ishikawa et al. 2000), the variational approach (Ruhnau et al. 2005), and those based on the implementation of the relaxation labeling technique, also called relaxation method (RM) (Wu and Pairman 1995; Baek and Lee 1996; Ohmi and Li 2000; Pereira et al. 2006; Mikheev and Zubtsov 2008). Each algorithm has its own strengths and weaknesses, and their use depends on the flow characteristics and tracer particle density level. In this paper, the discussion is focused on the integration of a version of BCC and RM for PTV. The aim of this integration is to provide a flexible methodology able to analyze images with different seeding and flow conditions. The integration is based on the natural complementation existing between both methods for the analysis of different seeding and flow conditions. The paper begins with a review of the fundamentals of both approaches, followed by the presentation of the proposed integration methodology and a comparison of its new capabilities through its application to the analysis first of synthetic images and second of large-scale measurements behind an array of cylinders in a shallow flow.

2 Algorithms

To understand the algorithms used here and their integration, some definitions are necessary. The term *target* particle will be used to identify the particles located in the first frame, at time $t = t_0$, for which its position in a second frame, at time $t = t_0 + \Delta t$, is required. An individual target particle is identified by a unique sub-index "i", which range from 1 to N, where N is the total number of particles at the first frame. The term *candidate* particle refers to the particles located at the second frame that potentially correspond to a certain particle "i". An individual particle at the second frame is identified by the sub-index "j", which can range from 1 to M, where M is the total number of particles at the second frame. The candidate particles associated with "i" are identified by the sub-index "$j(i)$". The particles "$j(i)$" can be found using some selection criteria, such as the maximum displacement expected for "i" after Δt.

2.1 The cross-correlation–based algorithm for PTV

In this algorithm, the velocity associated with a particle is found by using the highest cross-correlation coefficient obtained after comparing a reference intensity matrix in the first frame and a set of sub-matrices at the second one. The first reference matrix is found after extracting from the first frame the image intensities located inside of a square interrogation window of size l_{w}, which is centered at the target particle position $\vec{x}_i$. The set of second matrices is obtained after extracting the intensities of the second frame located inside of an interrogation windows centered at each of the candidate positions $\vec{y}_{j(i)}$.

Normally, the algorithm makes use of binary intensity matrices and due to this, it is known as BCC method (Uemura et al. 1989; Hassan et al. 1992). The BCC tracks individual particles based on the highest similarity of particle distribution patterns, and it is characterized by a low computational time (Ruhnau et al. 2005; Ishikawa et al. 2000). In order to binarize the original image, a single or multiple intensity threshold level can be used (Ohmi and Li 2000). In our present implementation, the images are not binarized, and the whole intensity field is used during the matching process. To differentiate the BCC algorithm from the algorithm used here, the latter is named simply cross-correlation (CC). A schematic view of the cross-correlation algorithm is presented in Fig. 1. In the figure, the reference matrix for a particle located at $\vec{x}_i$ is extracted at the first frame by using a square interrogation window of size l_{w}. The length l_{w} is an estimation of the maximum displacement of particle "i". In the second frame, the interrogation window is situated at the same location as in frame one. The particles within the window correspond to the candidate particles of "i". The interrogation window is centered at each of the candidate particles locations $\vec{y}_{j(i)}$, where $j(i) = 1, 2, \ldots, ni$ and "ni" denotes total number of candidates associated with particle "i". The cross-correlation coefficients, between the reference matrix at frame 1 and each of the matrices centered at the candidate particles, are calculated as follows:

$$R = \frac{\sum_M \sum_N (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \bar{A})^2\right)\left(\sum_m \sum_n (B_{mn} - \bar{B})^2\right)}} \qquad (1)$$

where R is the cross-correlation coefficient, A and B are matrices of size $m \times n$, and $\bar{A}, \bar{B}$, are the mean values of light intensities of elements of respective matrices.

The displacement between a particle located at a position $\vec{x}_i$ on the first frame and its candidate particles located at $\vec{y}_{ij(i)}$ is defined as $\vec{d}_{ij(i)} = \vec{u}_{ij(i)}\Delta t$, where $\vec{u}_{ij(i)}$ is the mean particle velocity. When the flow contains high velocity gradients and $\vec{d}_{ij(i)}$ is several times longer than the particles diameter, the particle patterns can suffer from strong nonlinear deformation. These pattern deformations can produce an important reduction in the correlation level calculated by CC, decreasing the reliability of the PTV analysis. In addition, the correlation level used in CC can

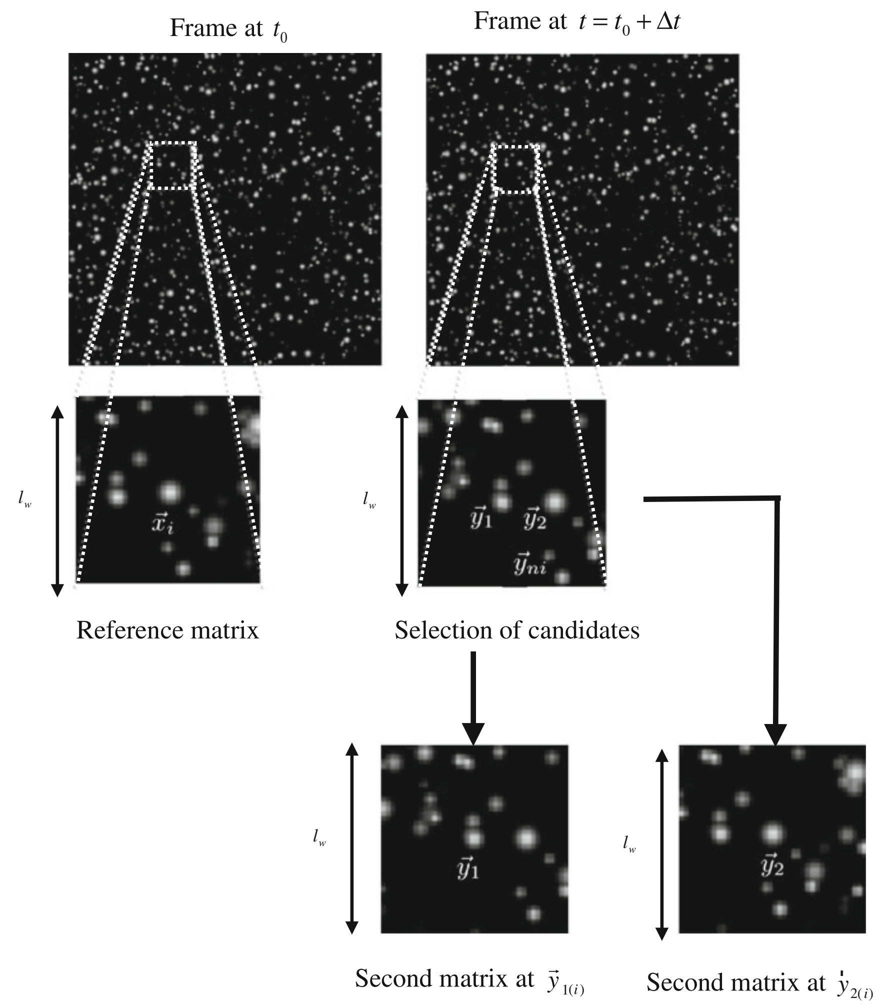


Fig. 1 Fundamentals of cross-correlation PTV. The reference matrix is cross-correlated with the interrogation matrix, centered at each of the candidate particles. The higher cross-correlation coefficient in this case is obtained when the matrix at the second frame is centered at $\vec{y}_{2(i)}$. Image source: PIV-Standard Project of the Visualization Society of Japan (VSJ)

be also reduced due to changes in the image intensity distribution after Δt, a common problem in large-scale measurements illuminated, for example, by means of diffusive light sources. These intensity changes can be also due to the heterogeneity of the particles shapes or due to changes in the cross-plane position of the particles inside of the light sheet.

2.2 The relaxation algorithm

This PTV algorithm is based on the use of the iterative relaxation labeling technique. Even though the technique has been widely used to solve matching problems on computer vision (Kittler et al. 1993), its first implementation for PTV was due to Wu and Pairman (1995) followed by Baek and Lee (1996) in the context of turbulent flows. Additional modifications were done by Ohmi and Li (2000), but according to Pereira et al. (2006), these modifications have proven to yield minor improvements.

In RM, a quasi-rigidity radius t_q is used to select neighbor particles that are used during the update process of the displacement $\vec{d}_{ij(i)}$. The matching probability $P_{ij(i)}$ of a displacement $\vec{d}_{ij(i)}$ and the associated no-match probability P_i^* of a particle at $\vec{x}_i$ are updated using the displacements $\vec{d}_{k(i)l(k)}$ of selected neighbors particles. A neighbor particle is a particle located within a radius T_n with respect to $\vec{x}_i$, where T_n is an estimation of the size of the region containing particles with a similar motion to that located at $\vec{x}_i$, which is of the order of the local spatial length scale of the flow. The selected neighbors are those, located at $\vec{x}_{k(i)}$, that satisfy the quasi-rigidity condition $\left|\vec{d}_{ij(i)} - \vec{d}_{k(i)l(k)}\right| < t_\mathrm{q}$. It is defined here that the selected neighbors displacements during an iteration have a probability of matching $P_{k(i)l(k)}$. As showed earlier, the selected neighbor particles of a particle "i" are denoted here by the sub-index "$k(i)$", where $k(i) = 1, \ldots, n_\mathrm{ns}$, and their candidate particles at the second frame are denoted here by "$l(k)$", where $l(k) = 1, \ldots, ni_k$.

The iterative probability update process adopted here is the one presented by Baek and Lee (1996), which can be summarized as follows. First, for each target particle "i", the following condition is used for the renormalization of probabilities:

$$\sum_{j(i)=1}^{ni} \left(P_{ij(i)}\right) + P_i^* = 1 \tag{2}$$

The initialization of probabilities for the iterative procedure is defined by:

$$P_{ij(i)}^{(0)} = P_i^{*(0)} = \frac{1}{ni+1} \tag{3}$$

For the probability update, it is necessary to calculate $Q_{ij(i)}$, which is the sum of the probability of all neighbors.

$$Q_{ij(i)}^{(n-1)} = \sum_{k(i)=1}^{n_{ns}} \sum_{l(k)=1}^{ni_k} P_{k(i)l(k)}^{(n-1)} \tag{4}$$

where “n” is the iteration step. Finally, the normalized match and no-match probability can be calculated by using Eq. 2 as:

$$P_{ij(i)}^{n} = \frac{AP_{ij(i)}^{(n-1)} + BQ_{ij(i)}^{(n-1)}}{\sum_{j(i)}^{ni}\left(AP_{ij(i)}^{(n-1)} + BQ_{ij(i)}^{(n-1)}\right) + P_i^{*(n-1)}} \tag{5}$$

$$P_i^{*(n)} = \frac{P_i^{*(n-1)}}{\sum_{j(i)}^{ni}\left(AP_{ij(i)}^{(n-1)} + BQ_{ij(i)}^{(n-1)}\right) + P_i^{*(n-1)}} \tag{6}$$

where A and B are constants equal to 0.3 and 4.0, respectively. In Fig. 2, a graphical representation of the relaxation principle for PTV is shown. In this Figure, one of the neighbors displacements $(\vec{d}^*)$ does not provide information for the update of $\vec{d}_{ij(i)}$, because the quasi-rigidity radius condition is not fulfilled $\left(\left|\vec{d}_{ij(i)} - \vec{d}^*\right| > t_q\right)$.

The success of the RM algorithm depends on the existence of neighboring particles with similar motion. RM can fail in case of solitary particles, but using the modifications introduced by Pereira et al. (2006), this restriction can be avoided. However, this modification requires a previous estimation of the local flow velocity, which can be obtained, for instance, by a previous PIV step as proposed by Kim and Lee (2002), which is a similar approach to the one presented by Keane et al. (1995). According to Pereira et al. (2006), the method shows a low sensitivity to the tracking parameters and a good performance in flows with high velocity gradients and relatively high particle density. The reader may refer to the works of Baek and Lee (1996) and Pereira et al. (2006) for more information on the relaxation implementation and its recent modifications (Fig. 2).

On previous implementations of the relaxation algorithm, such as the ones by Baek and Lee (1996) and Ohmi and Li (2000), a fixed number of iterations were used for the probability update. Ohmi and Li (2000) point out that the optimal number of iterations depends on many factors, and no formalization could be realistic. In the present work, a simple and automatic approach is adopted. The iterative update process is stopped for a particle “i”, when the difference between its probability in the previous and current iteration is less than a certain threshold λ, and its probability is higher than the no-match probability P_i^* and a probability threshold, P_t. If the particle passes the constraints, the analysis for the particle is assumed to have converged. After this step, the particle is switched to an inactive mode where its information is only used to update the probabilities of the remaining no-match particles. In the present implementation, a value of $\lambda = 0.001$ was used. Figure 3 shows a flow diagram for the convergence criteria proposed for RM.

In order to speed up the calculation, an adaptive neighborhood definition is used. Based on a radius of coherence T_n and a maximum number of neighbors n_n, the algorithm imposes that, if the numbers of neighbors contained within T_n is larger than n_n, only the closer n_n particles are used as neighbors. A similar approach is followed for the definition of the candidate particles at the second frame. An interrogation window of radius l_w is used together with a maximum number of candidates n_c. If the displacements are short and $n_c = 1$, the relaxation method acts similar to the nearest neighbor method with a post-validation through relaxation. The proposed schemes for the determination of neighbors and candidates not only can

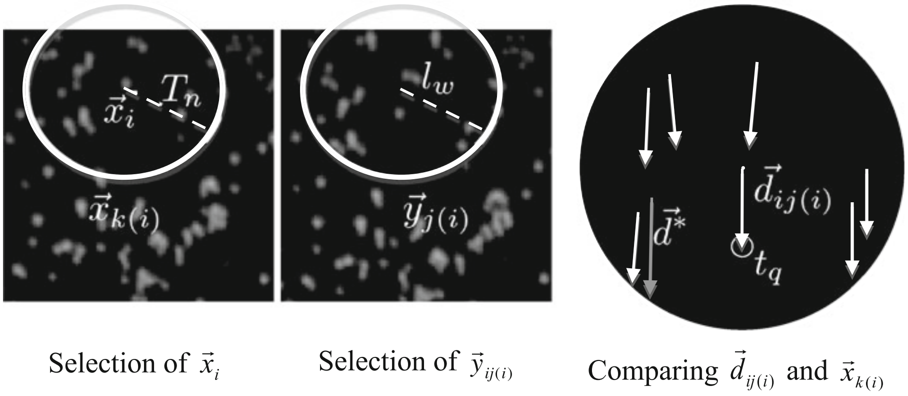


Fig. 2 Fundamentals of the relaxation method. First, the neighbors of particle $\vec{x}_i$ are determined by using the radius T_n. After $\vec{y}_{ij(i)}$ are determined, the neighbors selected for the update process are obtained by means of a quasi-rigidity radius t_q

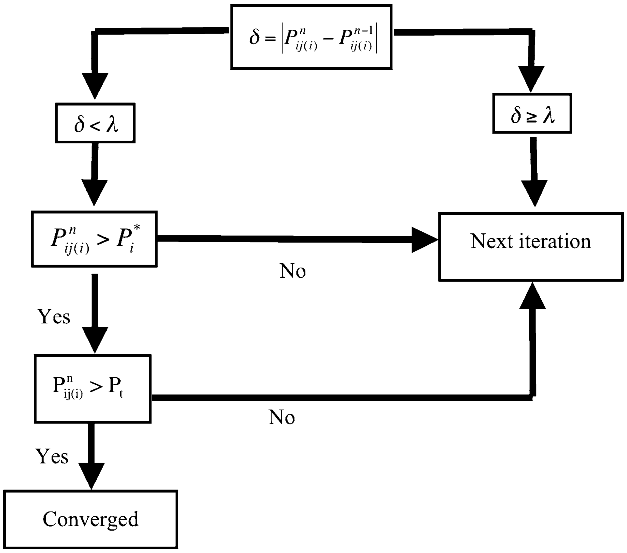


Fig. 3 Convergence criteria for RM. The probability difference between the previous and current iterations is calculated. If the difference is lower than λ, the probability is compared with the no-match probability and a threshold probability. If all the constraints are passed, the analysis of the particle is assumed to have converged

help to improve the time performance but also they help to deal with the analysis of nonhomogeneous particle densities in a better way.

2.3 Integrating the cross-correlation and relaxation algorithms: The ICCRM algorithm

Based on the characteristics of the RM approach, it is clear that (1) the algorithm can fail when a flow with a low density of particles and without previous estimation of the local velocity is analyzed; (2) the RM computational time depends on the total number of particles in the first frame and the number of the corresponding neighbors and candidate particles in the second frame; and (3) the number of iterations for the convergence of the algorithm to a global solution depends on the initial solution and the quality of the neighboring information.

In order to understand how CC and RM can be integrated, it is necessary to analyze the previous observations from the CC point of view. The low-density case mentioned in observation (1) corresponds to one of the most favorable conditions for the CC algorithm. In case of observation, (2) if the velocity is determined in the zones where the CC can give a valid matching, the computational time needed for RM is reduced due to the better time performance of CC and to the lower number of particles that need to be analyzed with RM at the first and second frame. Regarding observation, (3) the RM iteration process can be initialized using preliminary results obtained during the CC analysis. As a result, the number of iterations needed for the matching process by RM can be reduced.

Using the previous analysis, a two-step algorithm applying both the CC and RM methods is proposed. A CC step is used to determine a preliminary solution of the velocity field. The results of this step are filtered by using a correlation threshold level, C_t, together with mean and median filters (Westerweel and Scarano 2005; Nogueira et al. 1997). The particle matches accepted at this step are assumed valid, and a probability equal to 1 is assigned for the next relaxation step. The idea of assigning a probability 1 to the matched particles after the CC step is to ensure a strong weight of the CC solution during the RM analysis. The particles matched by CC do not participate actively in the relaxation process, i.e., their match does not change during the process, but they are used to update the probability of the particles analyzed by RM.

The matching during the RM step is dynamic in the sense that, after a particle reaches the convergence criteria showed in Fig. 3, it is also switched into an inactive mode, and its probability is used for the update of the remaining ones. This approach ensures a continuous decrease in the number of particles analyzed by RM with the number of iterations, speeding up the analysis. A flow diagram of the proposed integrated algorithm is presented in Fig. 4.

The correlation levels calculated during CC, between the no-match particles and their remaining candidates, are converted to initial probabilities, $Po_{ij(i)}$, using Eq. 7, where $R_{ij(i)}$ is the correlation coefficient associated with the displacement $\vec{d}_{ij(i)}$, and the function "max" is introduced to

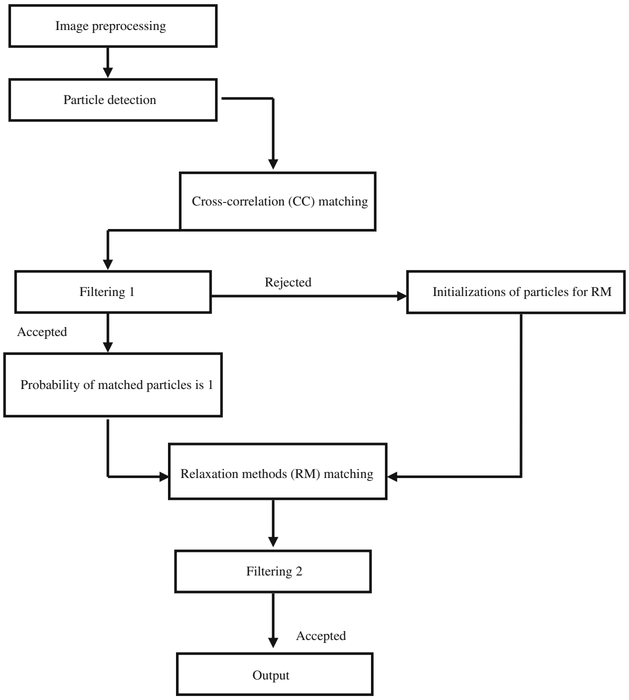


Image preprocessing
Particle detection
Cross-correlation (CC) matching
Filtering 1
Rejected
Initializations of particles for RM
Accepted
Probability of matched particles is 1
Relaxation methods (RM) matching
Filtering 2
Accepted
Output

Fig. 4 Flow diagram of the integrated ICCRM algorithm

determine the maximum value of $R_{ij(i)}$. As shown in Eq. 8, an initial no-match probability is also determined using the difference between 1.0 and the maximum correlation $R_{ij(i)}$. These probabilities are used as the initial condition for the iteration of the RM algorithm. In the work of Wu and Pairman (1995), the matching process is basically calculated twice, one by CC and the other by applying RM, without using the reliable correspondences obtained by CC to improve the overall performance.

$$Po_{ij(i)} = \frac{R_{ij(i)}}{\sum_{j(i)=1}^{ni}\left(R_{ij(i)}\right) + \left(1 - \max\limits_{j(i)=1\cdots ni}\left(R_{ij(i)}\right)\right)} \tag{7}$$

$$P_i^* = \frac{\left(1 - \max\limits_{j(i)=1\cdots ni}\left(R_{ij(i)}\right)\right)}{\sum_{j(i)=1}^{ni}\left(R_{ij(i)}\right) + \left(1 - \max\limits_{j(i)=1\cdots ni}\left(R_{ij(i)}\right)\right)} \tag{8}$$

To determine individual particle trajectories, a sequential labeling process was used for the identified particles. In the first frame of an image sequence, a unique identification number (ID) is assigned to the detected particles. After the PTV analysis is completed for the first image pair, each matched candidate particle receives the same ID of its associated particle at first frame. If a candidate particle does not have a valid match, a new ID is assigned and used for the PTV analysis of the next frame pair. The process is repeated for the whole image-pair sequence. After the analysis is finished, the trajectory of a single particle can be recovered by tracking the ID number over the output results.

2.4 Filters

As previously mentioned and in order to increase the reliability of the CC and of the final results, the ICCRM algorithm uses the following filter steps:

2.4.1 *Cross-correlation threshold (C_t)*

Only used after CC. The filter compares the CC coefficient obtained during this step with a threshold value. All values larger than the threshold are accepted.

2.4.2 *Double match filter*

This filter checks that a particle “*j*” in the second frame has a unique match at the first one. If particle “*i*” has more than two matches in the second frame, the displacements are compared with those of the neighbor particles, in terms of direction and magnitude, and the most similar displacement is kept as a valid match.

2.4.3 *Mean filter*

This filter compares the direction and magnitude of each velocity vector with its local mean neighborhood information. The neighborhood is defined by using a Tf_{n} coherence radius for filtering and a maximum number of neighbors nf_{n}. The filter also requires a maximum allowable angular difference $M\angle$. For a review on the use of the mean filter, the reader may refer to Westerweel (1994). The filter is mainly used in case of low particle densities; in the case of the ICCRM algorithm, it is used after CC and to validate the final results of the ICCRM analysis.

2.4.4 *Median filter*

The implementation of the median filter follows the ideas proposed by Westerweel and Scarano (2005). The same definition of neighborhood used for the mean filter is used here. The method is based on a normalized residual (r_i^*) that incorporates a minimum normalization level ε, representing the acceptable fluctuation level of the neighborhood median, in PTV terms, due to detection problems on the particles centroids. Equation 9 shows the calculation of the residual for a particle “*i*”. In this equation, $\vec{u}_i$ is the estimated velocity of particle “*i*”, $\vec{u}_{\mathrm{m}}$ is the local median velocity within the neighborhood, and r_{m} is the median of the residuals. The particle “*i*” is rejected when r_i^* is larger than a normalized residual threshold, t_{r}.

$$r_i^* = \frac{|\vec{u}_i - \vec{u}_{\mathrm{m}}|}{r_{\mathrm{m}} + \varepsilon} \tag{9}$$

For a given velocity field, the mean and median filters are applied iteratively until the conditions defined by the filter parameters are fully fulfilled. In Fig. 4, the positions of the filtering stages in the flow diagram of the ICCRM are shown; the filter sequence applied to the results of CC is called “filtering 1” and to the final results “filtering 2”. The order of application of the filters for each of these filtering steps is presented in Table 1.

Table 1 Sequence for the application of filters within the two filtering stages of ICCRM

Order of application	Filtering 1	Filtering 2
1	Cross-correlation threshold	Double match filter
2	Double match filter	Median filter
3	Mean filter[a]	
4	Median filter	

[a] The filter is used only for certain seeding conditions

2.5 Quality measurement

Following Hassan et al. (1992), the quality of a PTV analysis can be described using the yield and reliability parameters. Yield (Yi) is the ratio between the known number of particles displacements available between two images and the number of valid displacements recovered from the images. Reliability (Re l) is the ratio between the total number of recovered displacements and the number of valid displacements. In case of the ICCRM algorithm, it is difficult to make a difference between Yi and Re l, because filtering is considered to be part of the whole process. As pointed out by Ruhnau et al. (2005), the filtering parameters can influence both measurements; and in case of the ICCRM algorithm, a high filtering will produce a Re l of 100%. As a consequence, in this work only the Yi value is considered representative of the quality of the analysis, because it is the only one measuring the real capabilities of the algorithm. It is worth pointing out that future works on PTV working with the images of the Visualization Society of Japan (VSJ) should include the Yi value in order to measure, independently of the particle detection, the performance of the algorithms.

2.6 Computational implementation

The ICCRM code was implemented in Matlab, a high-level language that is slow compared with languages such as Fortran or C++, but extensively used in research. Even though the algorithm is highly vectorized, the time-performance comparison with other researches is difficult because there are many factors that can modify the computation time, for instance the computer model, the programming technique and the algorithm implementation. The most important factor is the limitation of the language when iterations are needed. The CC algorithm makes use of optimized Matlab functions, which are faster compared with the iterative cycles needed for the RM algorithm. As a consequence, in this work, the computation time is not presented. A further implementation of the algorithm in a compiled language is needed to fairly compare its time performance with other algorithms.

3 Synthetic images analysis

In this section, the performance of the ICCRM is measured using synthetic images provided by the PIV-standard project of the VSJ. The images were generated based on three-dimensional Large Eddy Simulation of a jet impingement on a wall. More information about the simulated flow field and image characteristics can be found in Okamoto et al. (2000). In this work, the images 0 and 1 of the series 301 were used as benchmark. The VSJ provides several text files containing the position of each particle in the images and their identification number. In order to compare the performance of the new algorithm without including external sources of error, the particle identification provided by VSJ was used. The images, with a particle density of about 0.06 $\frac{\text{particles}}{\text{pixel}}$, provide a way to verify the implementation and performance of PIV–PTV algorithms. The image pair 0 and 1 contains 4042 valid matches, this value was determined after analyzing the particles coordinates contained in the files provided by the VSJ. The parameters used for the analysis of these images are presented in Table 2.

A comparison of the results given by the proposed ICCRM algorithm with those given by other algorithms, including the present implementation of RM and CC, is summarized in Table 3. Therein, NAV is the number of available vectors, NFV the number of found vectors, and NVV the number of valid vectors. In addition, the values of Yi and Re l of the analysis are presented depending on the case. The number of valid vectors recovered by the ICCRM and the modified RM algorithm was 3980, which gives a Yi of 98.46%, slightly higher than the 91.23% obtained by the present modified version of CC. The result of the analysis of the VSJ images is presented in Fig. 5. The same level of performance obtained for ICCRM and RM was due to the homogeneity and particle density condition of the image, which facilitated a high performance of RM. However, and in case of the present implementation in an interpreted language, the computing time needed for the analysis of the images using ICCRM was several times lower than when

Table 2 Cross-correlation (CC) and relaxation method (RM) parameters for the analysis of VSJ standard image 301

Parameter	Value	Units	Description
l_{w}	10	Pixel	Size of interrogation window
n_{c}	5	–	Maximum number of candidate particles
C_{t}	0.78	–	Correlation coefficient threshold
t_{q}	0.35	Pixel	Quasi-rigidity radius for relaxation method
λ	0.001	–	Threshold for convergence criteria
T_{n}	20	Pixel	Coherence radius for relaxation method
n_{n}	25	–	Maximum number of neighboring particles
Tf_{n}	25	Pixel	Coherence radius for filtering
$M\angle$	90	Degrees	Maximum angle for filtering
nf_{n}	5	Pixel	Maximum number of neighbors for filtering
ε	0.25	Pixel	Acceptable fluctuation of residuals median
t_{r}	2	–	Threshold of normalized residual

Table 3 Comparison of present and previous analysis of VSJ standard image 301

Algorithm	Particle detection	NAV[a] or NFV[b]	NVV	Yi[a] or Re l^b (%)
ICCRM (present work)	VSJ data	4,042[a]	3,980	98.46[a]
RM (present work)	VSJ data	4,042[a]	3,980	97.46[a]
CC (present work)	VSJ data	4,042[a]	3,687	91.23[a]
EPTV (Mikheev and Zubtsov 2008)	VSJ data using known intensity	4,042[a]	3,823	94.58[a]
NRX (Ohmi and Li 2000)	Dynamic threshold binarization	808[b]	788	98[b]
BCC (Ohmi and Li 2000)	Dynamic threshold binarization	860[b]	788	91.6[b]
FIT (Ohmi and Li 2000)	Dynamic threshold binarization	630[b]	559	95[b]
ORX (Ohmi and Li 2000)	Dynamic threshold binarization	786[b]	762	96[b]
VAR (Ruhnau et al. 2005)	Random values from VSJ data	872[b]	865	99.2[b]
EPTV (Mikheev and Zubtsow 2008)	Dynamic threshold binarization	1,759[b]	1,733	98.6[b]

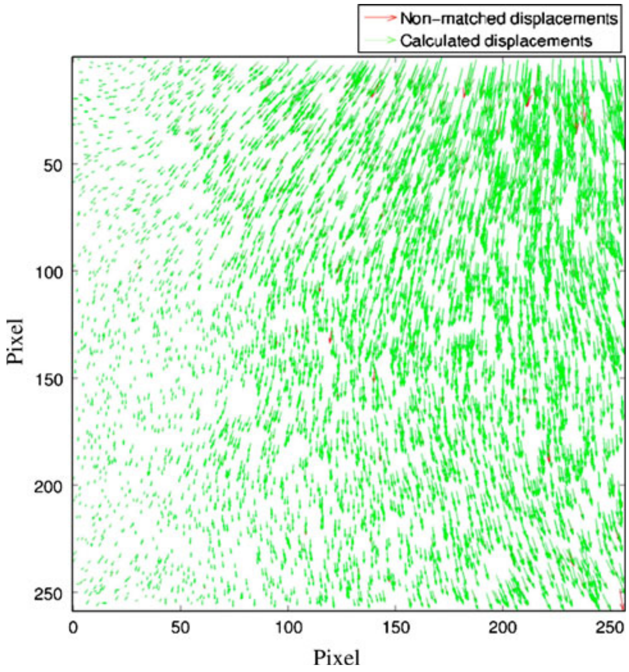


Fig. 5 Result of the analysis of VSJ standard image pair 0 and 1 of series 301 using the proposed ICCRM

using RM. This computing efficiency was due to the high time performance of CC, which reduces the number of particles passed to RM thus improving the performance of the global analysis. A previous work by Mikheev and Zubtsov (2008), also using the known particle data provided by the VSJ, reports 3823 valid vectors for the same image pair, which corresponds to a Yi level of 94.58%, similar to, although slightly lower than, the present ICCRM and RM results. The work of Mikheev and Zubtsov (2008) is the only one, to the best of our knowledge, reporting results that can be compared in quality with those of the ICCRM algorithm. As a reference, the Re l value obtained by other algorithms is also presented in Table 3.

4 Experimental images analysis

To test the performance of the ICCRM algorithm on experimental images, a series of frames have been analyzed. They were obtained from experiments conducted to determine the influence of a cylinder row on coherent structures in shallow flows (Uijttewaal and Jirka 2003). The shallow water basin at the Institute of Hydromechanics of the University of Karlsruhe was used for these experiments. The basin has a flat bottom and is 13.5 m long and 5.48 m wide. A PCO-Sensicam camera with a 14 mm Nikon lens was used to record the movements of surface particles just downstream of the cylinders array. The camera covered an area of 1.2 × 1.5 m with an image resolution of 1,024 × 1,280 pixel at an acquisition frequency of 7 Hz.

The cylinders had a diameter of 0.063 m, the incoming flow velocity was 0.05 m/s, and the water depth was 0.045 m. Polypropylene granulated particles with a diameter between 2 and 3 mm and a density slightly less than that of the water were used in these experiments. More details on the particles and the seeding system used during the experiments can be found in Weitbrecht et al. (2002).

In the work of Uijttewaal and Jirka (2003), the flow field was measured using both PIV and Laser Doppler Anemometry (LDA). To ensure the reliability of the PIV results, they were compared with the LDA measurements at some positions (see Uijttewaal and Jirka 2003). To validate the results presented here, the instantaneous 199 velocity fields calculated using PIV and ICCRM-PTV were compared. The PIV analysis was performed by following the same method used by Uijttewaal and Jirka (2003). A PIV algorithm (DaVis 5.4.4) was applied to the images, with 50% overlapping interrogation areas of 32 × 32 pixel that contained on average 3–5 particles and resulted in a field of 64 × 80 vectors. The interpolation of the irregularly spaced PTV results to the grid positions used in PIV was performed by means of the Inverse distance Weighting

Method with a Gaussian Kernel, also called the Adaptive Gaussian Windows (AGW) (Spedding and Rignot 1993).

The difference between the velocity vectors measured by PIV and PTV was calculated, and the spatial mean of the difference for each instantaneous velocity field determined. Finally, the spatial average of the root mean square (RMS) was calculated as a measure of the difference between the PIV and PTV measurements. The analysis showed that this value for the streamwise and spanwise velocity components was about 7% of the mean depth-averaged velocity, i.e., the spatial average—in the horizontal plane—of the depth-averaged velocity, respectively. Although this value corresponds to a convolution of the errors associated with both the PIV and PTV methods, it was taken as a measure of the closeness of the results yielded by both methods and thus a way to validate the present results with the proposed ICCRM. Based on this analysis, it was concluded that the ICCRM provides a valid representation of the measured flow field.

An additional qualitative and simple method to check the reliability of the PTV results was based on a comparison of long-time exposure pictures with time-series PTV trajectories. One of the 20 frames long-time exposure image used for this comparison is presented in Fig. 6.

As shown in Fig. 6, the flow is characterized by the fluid acceleration between the cylinders, followed by a high velocity gradient zone and a wake zone at the downstream region. The existence of these three zones was the main reason for the selection of this experiment to test the capabilities of the ICCRM. The images contain, in PTV terms, a zone of large displacements, nonlinear deformation and linear deformation. In addition, as in most shallow flow experiments, the seeding is heterogeneous. In the downstream direction, there exists a change in particle concentration due to particle clustering, a typical problem in large-scale measurements as the one presented here. The characteristic densities $\left(\text{in}\,\frac{\text{particles}}{\text{pixel}}\right)$ are 0.0037, 0.0045 and 0.0015 for the acceleration, high velocity gradient and wake zones, respectively. The mean number of particles in each frame was approximately 3,500, corresponding to a spatial-mean particle density of $2.65 \times 10^{-3}\,\frac{\text{particles}}{\text{pixel}}$, a value about 20 times lower than that of the synthetic case analyzed in the previous section. In order to detect the particles, the Gaussian Mask technique was used (Takehara and Etoh 1998). To reduce the computation time during the particle detection, a small intensity threshold level was used to determine the regions of the image that potentially can contain particles. After those regions were determined, their correlation with the Gaussian mask was calculated, and the particles were identified. In order to exemplify how both algorithms, CC and RM, are combined, the analysis for frames 0–1 and 0–2 is presented in Fig. 7. The parameters used for the analysis of the results shown in Fig. 7 are presented in Table 4.

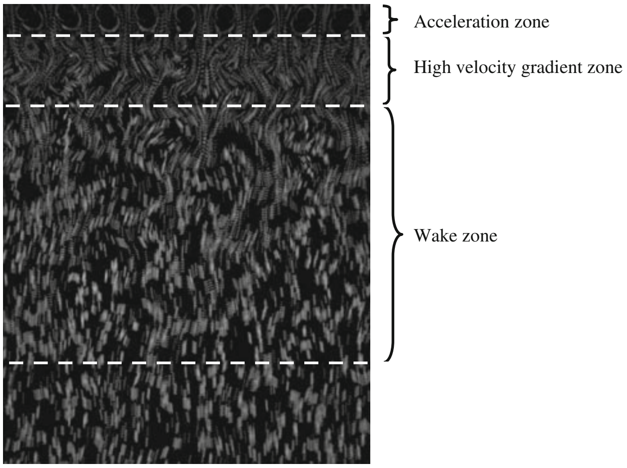


Fig. 6 20-frames long exposure image of one of Uijttewaal and Jirka's (2003) experiments showing particles trajectories in the flow direction from top to the bottom of the figure. The flow field is mainly divided into acceleration, high velocity gradient and wake zones

As seen in Fig. 7, the relaxation method is mainly used in the region located immediately downstream from the cylinder array, which corresponds to the zone where nonlinear deformations of the particles patterns are produced. This deformation is even more pronounced when one frame is skipped (Fig. 7-right); and as a consequence, the number of matches solved by the relaxation method is increased.

Figure 8 shows the trajectories of 1,154 particles, tracked for times longer than 100 frames after 199 analyzed frames. The parameters used for this long-term PTV analysis are those shown in Table 4. This figure illustrates the ability of the proposed algorithm to determine trajectories of tracer particles for long times, providing data to compute Lagrangian statistics.

The trajectories shown in Fig. 8 are raw results, because they do not include any post-processing for their reconstruction, such as that provided by the algorithm proposed by Xu (2008). We assert that the assessment of a PTV algorithm should be more oriented to establish its ability to recover a large number of trajectories; i.e., the algorithm should be flexible enough to deal with unsteady flow patterns and seeding conditions, and not only to prove its performance in a particular frame, which, in many cases, is an iterative optimization problem of the control parameters to get a large number of recovered vectors. It is also necessary to show that the chosen parameters are flexible enough to consistently solve the velocity field, tracking tracer particles along a large number of frames.

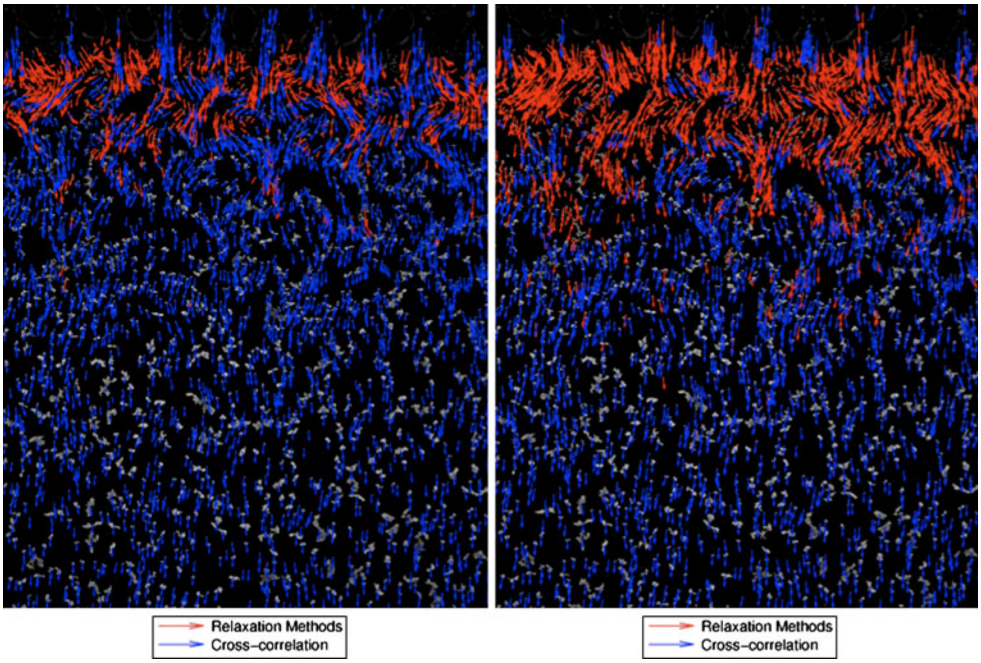


Fig. 7 *Left* Result of the analysis of experimental images 0 and 1. Most of the displacements located in the high velocity gradient zone are solved by RM. *Right* Result for frames 0 and 2. Due to the nonlinear deformations produced by the larger time step, the size of the region solved by RM has increased (The vector scale used for the right figure is half the scale of the left one)

Table 4 Cross-correlation and relaxation parameters for ICCRM-PTV analysis

Parameter	Value	Units
l_w	30	Pixel
n_c	5	–
C_t	0.4	–
t_q	6	Pixel
λ	0.001	–
T_n	100	Pixel
n_n	15	–
Tf_n	50	Pixel
$M\angle$	90	Degrees
nf_n	5	Pixel
ε	0.3	Pixel
t_r	3	–

The parameters were used for the analysis of frames 0–1, 0–2 and for the whole image series

5 Parameters sensitivity

In this section, the performance of the ICCRM algorithm over a range of values of its control parameters is analyzed. The experimental image pair 0 and 1 presented in Sect. 4 is considered for the analysis, using the solution obtained by ICCRM reported in that section as a reference. It is assumed that the control parameters used in Sect. 4 are the optimal ones for the analysis of the image pair (see Table 4). In addition and as a way to understand the role of CC and RM on the global ICCRM analysis, the results obtained by RM are also presented. The parameters used for the RM analysis are those presented in Table 4. These parameters allowed the recovery of the highest number of valid vectors using RM alone. Our analysis showed that the best ICCRM performance is obtained when the best parameters for CC and RM methods alone are used. Here, the term "number of valid vectors" refers to the amount of vectors that coincide with those of the reference solution obtained by ICCRM. For the parameter sensitivity analysis, only one of the parameters of Table 4 was varied at a time,

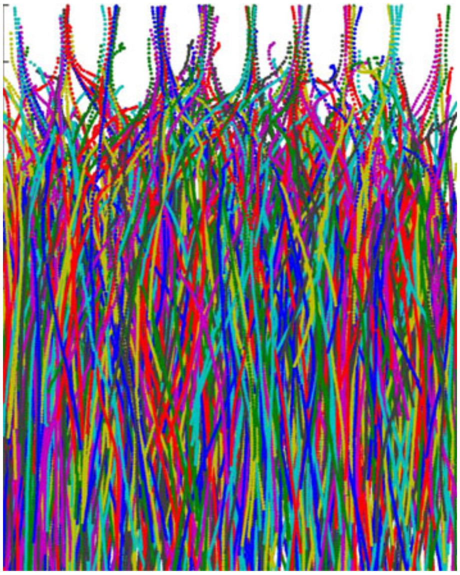


Fig. 8 Recovered particles trajectories after the analysis of 199 frames. The color is used to better visualize individual particle trajectories

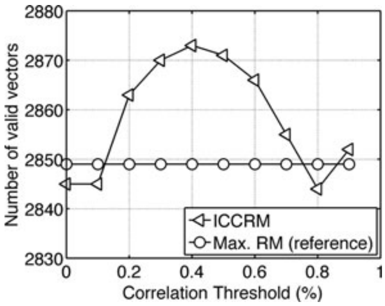


Fig. 9 Number of valid vectors as a function of the cross-correlation threshold (C_t)

while the rest remained constant with the value shown in that table.

Figure 9 shows the comparison of the solution obtained by RM and by ICCRM using a range of C_t values between 0 and 0.9. The performance of the ICCRM is better in most of the tests showing the contribution of CC toward improving the RM results. The exceptions were the tests with values of C_t lower than 0.1 and about 0.8, respectively, for which the reference RM result is better. This is due to the better $\mathrm{Re}\,l$ of RM compared with CC. In case of C_t values close to 0, and even though a filter is applied at the end of the CC step, many outliers pass to the next RM step. The RM step relies on the existence of a neighborhood with valid information, but in case outliers are passed to the second RM step in ICCRM, valid vectors will be compared with a full nonvalid neighborhood and will thus be finally removed. As it is shown in Fig. 9, the maximum number of valid vectors obtained by ICCRM is reached for a value of $C_t = 0.4$. At higher values of C_t, valid matches are filtered thus reducing the performance of the method. In the ICCRM algorithm, as in CC, the results are very sensitive to the C_t value and because of this it must be carefully chosen to ensure the quality of the information passed to the next RM step.

Figure 10, left and right panels, show the effect of different values of n_n and t_q, respectively, on the solution of the ICCRM and RM algorithms. In all cases, the performance of the ICCRM was better, due to the existence of the previous CC solution. The best solution for ICCRM was obtained for $n_n = 15$ and $t_q = 6$. The highest difference between ICCRM and RM was obtained when a small t_q value was used for the calculations. In such a case, the information provided by the candidate particles in RM is highly restricted, decreasing the number of vectors recovered. In case of the ICCRM, on the contrary, the number of particles matched is kept high because most of the flow field is solved by CC.

Figure 11, left and right panels, show the effect of the median filter parameters ε and t_r on the number of valid vectors recovered by the ICCRM and RM algorithms. The solutions in both the synthetic and experimental cases analyzed in this paper showed a high sensitivity to the filtering parameters. As shown in the left panel of Fig. 11, there is a large difference in the number of vectors obtained, of almost 500 vectors, when the values of $\varepsilon = 0$

Fig. 10 Number of valid vectors as a function of number of neighbors (n_n, *left panel*) and quasi-rigidity radius (t_q, *right panel*)

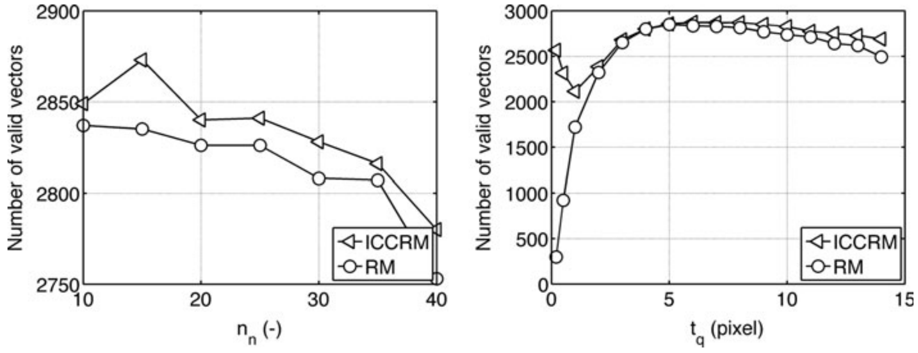
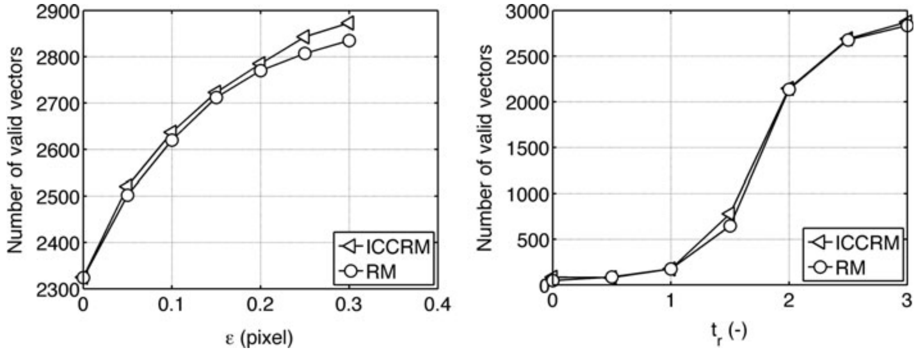


Fig. 11 Number of valid vectors as a function of ε (*left panel*) and residual threshold t_r (*right panel*)



pixel and $\varepsilon = 0.3$ pixel are used. A similar situation can be observed in the right panel of Fig. 11 when t_r is varied. After comparing the PIV results and the long-time exposed images obtained from the experiments of Uijttewaal and Jirka (2003) with the solution of the PTV analysis, it is possible to conclude, using the number of valid vectors as a criterion, that the upper limit for the ε and t_r values was 0.3 and 3, respectively. It is necessary to mention that Westerweel and Scarano (2005) propose values of 0.1 and 2, respectively, for these parameters, but in their case the work was oriented to the PIV technique characterized by a high particle density.

6 Conclusions

A two-step PTV algorithm, denoted ICCRM, that combines the well-known cross-correlation (CC) and RM has been presented. In general, the ICCRM has a better performance, in terms of number of matched particles than CC and RM alone, because both methods are complementary. The CC method is able to match particles with low neighboring information, a difficult case for RM, due to the requirement of neighbors with similar motion for the success of the iterative procedure. In addition, RM can solve the matching problem in cases with relatively strong velocity gradients, a problematic case for CC due to the loss of similarity in the particles patterns, which decreases the value of the CC coefficient. Relevant improvements of both CC and RM have also been implemented.

In case of low particle density or isolated particles, the matching in the proposed ICCRM can be addressed by means of the CC step, without the need of a previous calculation of the local mean velocity, as is the case in recent modifications on RM or in the hybrid PIV–PTV methods.

The ICCRM was tested with help of synthetic images, with the conclusion that the proposed algorithm and the modified RM and CC showed similar performance, though slightly better in terms of the number of valid recovered vectors, than recently proposed PTV methods. On the other hand, the ICCRM was successfully applied to experimental images of grid turbulence in a large-scale shallow flow, characterized by three zones of different dynamic range and particle concentrations, typical of large-scale measurements. This application showed the reliability of the ICCRM for the analysis of a single 2-frame image, and for the recovery of a large number of particles trajectories, thus providing time series of data that are long enough to compute Lagrangian statistics. This is one of the main contributions of the proposed PTV method.

It was found that in case of low particle density or isolated particles, the matching in the proposed ICCRM can be addressed primarily by means of the CC step. On the other hand, it has been shown that the ICCRM approach is less sensitive to the change of parameters when compared with a reference RM; however, this result must be considered with care because the sensitivity of the processes depends also on the flow characteristics of the analyzed images.

Finally, it is worth commenting on the performance of the CC algorithm observed in the present analysis. Even though this method can fail in case of high velocity gradients, as in the acceleration zone of the presented experimental images, a comparison with the results reported by Ohmi and Li (2000) for the BCC algorithm shows that including the image intensity characteristics and improving the particle detection technique can importantly boost the number of valid vectors produced by this method.

Acknowledgments The first two authors are indebted to their coauthor, the late Prof. Gerhard H. Jirka, for his enthusiastic support and wise guidance during the time we worked together. His memory will always be with us.

The University of Chile and the Karlsruhe Institute of Technology supported this work. The authors gratefully acknowledge the support provided by the German Science Foundation (DFG Grant JI 18/18-1), the scholarship program of the National (Chilean) Commission of Science and Technology research, CONICYT, and the support from Fondecyt Project 1080617. The authors are grateful to Prof. W.S.J. Uijttewaal, who allowed the use of his experimental database for the analysis of the ICCRM algorithm. Finally, we would like to thank the contribution of three anonymous reviewers who contributed importantly to improve the quality of the present work.

References

Baek SJ, Lee SJ (1996) A new two-frame particle tracking algorithm using match probability. Exp Fluids 22:23–32

Hassan Y, Blanchat T, Ch S (1992) PIV flow visualization using particle tracking techniques. Meas Sci Technol 3:633–642

Ishikawa M, Murai Y, Wada A, Iguchi M, Okamoto K, Yamamoto F (2000) A novel algorithm for Particle Tracking Velocimetry using the velocity gradient tensor. Exp Fluids 29(6):519–531

Keane RD, Adrian RJ, Zhang Y (1995) Super-resolution particle imaging velocimetry. Meas Sci Technol 6:754–768

Kim H, Lee S (2002) Performance improvement of two-frame Particle Tracking Velocimetry using a hybrid adaptive scheme. Meas Sci Technol 13(4):573–582

Kittler, J., Christmas, W., and Petrou, M. (1993). Probabilistic relaxation for matching problems in computer vision. Computer Vision, 1993. Proceedings of the fourth international conference on computer vision, pp 666–673

Mikheev A, Zubtsov V (2008) Enhanced Particle Tracking Velocimetry (EPTV) with a combined two-component pair-matching algorithm. Meas Sci Technol 19(085401):085401

Nogueira J, Lecuona A, Rodriguez P (1997) Data validation, false vectors correction and derived magnitudes calculation on PIV data. Meas Sci Technol 8(12):1493–1501

Ohmi K, Li H (2000) Particle tracking velocimetry with new algorithms. Meas Sci Technol 11(6):603–616

Okamoto K, Nishio S, Kobayashi T, Saga T, Takehara K (2000) Evaluation of the 3d-PIV standard images (PIV-STD project). J Vis 3(2):115–123

Pereira F, Stuer H, Graff E, Gharib M (2006) Two-frame 3D particle tracking. Meas Sci Technol 17:1680–1692

Ruhnau P, Guetter C, Putze T, Schnoerr C (2005) A variational approach for Particle Tracking Velocimetry. Meas Sci Technol 16:1449–1458

Spedding GR, Rignot EJM (1993) Performance analysis and application of grid interpolation techniques for fluid flows. Exp Fluids 15(6):417–430

Stellmacher M, Obermayer K (2000) A new Particle Tracking algorithm based on deterministic annealing and alternative distance measures. Exp Fluids 28(6):506–518

Takehara K, Etoh T (1998) A study on particle identification in PTV-particle mask correlation method. J Vis 1(3):313–323

Uemura T, Yamamoto F, Ohmi K (1989) A high-speed algorithm of image analysis for real time measurement of a two-dimensional velocity distribution. Flow visualization, ASME FED-85, pp 129–134

Uijttewaal WSJ, Jirka GH (2003) Grid turbulence in shallow flows. J Fluid Mech 489:325–344

Weitbrecht V, Kuhn G, Jirka G (2002) Large scale PIV-measurements at the surface of shallow water flows. Flow Meas Instrum 13:237–245

Westerweel J (1994) Efficient detection of spurious vectors in Particle Image Velocimetry data. Exp Fluids 16(3):236–247

Westerweel J, Scarano F (2005) Universal outlier detection for PIV data. Exp Fluids 39(6):1096–1100

Wu Q, Pairman D (1995) A relaxation labeling technique for computing sea surface velocities from sea surface temperature. IEEE Trans Geosci Remote Sens 33:216–220

Xu H (2008) Tracking Lagrangian trajectories in position-velocity space. Meas Sci Technol 19(7):75105