

A new algorithm for the interrogation of 3D holographic PTV data based on deterministic annealing and expectation minimization optimization

R. Kreпки, Ye Pu, Hui Meng, K. Obermayer

Abstract Recently we have presented a new particle tracking algorithm for the interrogation of 2D-PTV data [Kuzmanowski et al. (1998); Stellmacher and Obermayer (2000) *Exp Fluids* 28: 506–518], which estimates particle correspondences and local flow-field parameters simultaneously. The new method is based on an algorithm recently proposed by Gold et al. [Pattern Recognition (1998) 31:1019–1031], and has two advantages: (1) It allows not only local velocity but also other local components of the flow field such as rotation and shear to be determined; and (2) it allows flow-field parameters also to be reliably determined in regions of high velocity gradients (e.g., vortices or shear flow).

In this contribution we extend this algorithm to the interrogation of 3D holographic particle image velocimetry (PIV) data. Benchmarks with cross-correlation and nearest-neighbor methods show that the algorithm retains the superior performance which we have observed for the 2D case. Because PTV methods scale with the square of the number of particles rather than exponentially with the dimension of the interrogation cell, the new method is much faster than cross-correlation-based methods without sacrificing accuracy, and it is well adapted to the low particle seeding densities of holographic PIV methods.

1 Introduction

Holographic particle image velocimetry (HPIV) is a method which allows the recording of instantaneous 3D information from a volume of fluid. The configuration of tracer particles is recorded by a hologram which is then used to reconstruct the particle images optically.

In brief, the flow field is seeded with particles and is illuminated by a laser pulse of no longer than a few nanoseconds, which is short enough to freeze the instantaneous particle positions. The scattered light from the seeding particles is superimposed by a reference wave and the interferogram is recorded on a holographic plate. Two instances separated by a small time interval are recorded either on a single hologram or on two separate frames. The developed holograms are then illuminated by the same reference wave to generate a virtual image of the 3D particle image field. The virtual images are then recorded slice by slice using a CCD camera, and the displacements of the particles between exposures are determined from the resulting voxel data. Recently, classical Gabor or in-line HPIV (Weinstein et al. 1985; Meng and Hussain 1991; Scherer and Bernal 1997) has been replaced by off-axis techniques which tolerate higher seeding densities and offer a better image signal-to-noise ratio (Barnhart et al. 1994; Zhang et al. 1997; Pu 1998; Pu and Meng 2000).

The relatively low seeding densities in 3D space and huge computational cost in 3D fast Fourier transform (FFT) make applications of standard cross-correlation-based interrogation procedures unsuitable for HPIV. Several methods have been proposed for the interrogation of 3D-PTV data. However, these methods require some kind of a priori knowledge about the flow field, such as the maximal expected velocity in a certain subregion. Though they involve physical properties of fluid motion (Malik et al. 1993) or use more than two frames for the estimation of higher-order transformation moments (Guezennec et al. 1994), they require very accurate preprocessing procedures for particle centroid extraction. However, particle extraction procedures applied to HPIV often yield spurious particles or miss particles which are actually present (later referred to as acquisition noise). Moreover, the uncertainty of the centroid coordinate (later referred to as positional noise) in the z direction (depth) for HPIV is much higher than on the xy plane. Methods such as the 3D particle tracking method, which searches all possible particle trajectories in multiple frames, and determines the “most likely” trajectory for each particle by evaluating a penalty function (Guezennec et al. 1994) search for a “most likely corresponding” particle within a certain radius from the expected particle position, which requires a priori knowledge about the flow field, or use heuristic weights in the particle pairing procedure. The penalty function is often based on the deviation of the position of a particle from its expected position (Nishino and Kasagi 1989); however, particles may get lost during coordinate

R. Kreпки (✉), K. Obermayer
Department for Computer Science
Technical University of Berlin
Sekt. FR 2-1, Franklinstrasse 28/29
10587 Berlin, Germany

Y. Pu, H. Meng
Department of Mechanical and Aerospace Engineering
The State University of New York, University at Buffalo
324 Jarvis Hall, Buffalo, NY 14260-4400, USA

We are grateful to Martin Stellmacher for valuable discussions. This work was funded by the Technical University of Berlin via IFP 11/21.

extraction, which leads to wrong partnering. Finally, Okamoto et al. (1995) suggested the Spring Model, which is not affected by distorted motion, strong velocity gradients, rotation and shear flow, but does not extract these higher-order transformation components, which are of great interest in turbulent flows, flows in the near-wall region or jets. Rotation and shear are taken into account by the affine transformation technique (Okamoto 1998). This method is a practical heuristic, but theoretical formalization is lacking.

Ideally, an algorithm for HPIV velocity extraction should be insensitive to (i) a large variation in particle number density (from less than 15 to over 100 per interrogation cell (IC)), (ii) a large amount of particle pattern deformation, i.e., rotation, shear, or elongation, (iii) a large number of unpaired particles (up to 40%) resulting from particle movement out of the IC and acquisition noise, and (iv) errors in determining particle position. Since the HPIV algorithm must operate in 3D, it must be able to handle the large uncertainty of particle position along the z axis (perpendicular to the hologram). Furthermore, a 3D image from a hologram contains typically over 100 Gb of data, such that the algorithm must either be efficient, or be able to work with compressed data, such as particle centroids only.

In this paper, we describe a principled approach to the interrogation of 3D HPIV data which overcomes several of the above-mentioned problems. The particle matching (PM) method extracts translation and rotation components of the local flow field as well as particle correspondences simultaneously. Parameterization can be adopted to shear flows, and the shear factors can be extracted from the data.

Stellmacher and Obermayer (1999) recently developed a 2D version of the PM PTV-interrogation procedure, which shows good performance with respect to speed (compared with conventional PIV algorithms such as cross-correlation (Keane and Adrian 1992)), accuracy (compared with conventional PTV algorithms such as nearest-neighbor search (Baek and Lee 1996) and image relaxation (Ohmi and Lam 1998)), and robustness against several sources of noise. In the spirit of Keane et al.'s (1995) super-resolution PIV, the new method extracts flow-field parameters as well as particle correspondences, but the estimation of both sets of quantities is performed simultaneously: particle correspondences help to calculate flow-field parameters which, in turn, help to disambiguate particle correspondences. All these properties make PM an excellent candidate to be applied to the interrogation of holographic PTV. In this paper, we therefore extend the PM procedure to interrogate 3D holographic PTV data and provide extensive benchmarks with artificial as well as real-world data sets.

This paper is divided into six parts. Section 2 briefly describes the experimental setup. Section 3 contains a description of the two new interrogation methods, PM mentioned above, and the concise cross correlation (CCC) method, which is an example of state-of-the-art PTV approaches to the interrogation of 3D data. This is followed by a brief description of the cross-correlation method based on Fast Fourier Transformation (FFT). Section 4 provides benchmark results for PM, FFT-based cross-

correlation (Keane and Adrian 1992), nearest neighbor search (Baek and Lee 1996), and the CCC procedure. Section 5 shows results for real-world 3D HPIV data and Section 6 concludes with a brief discussion.

2

Data acquisition and preprocessing

Data were acquired using an off-axis HPIV recording setup as described in Pu and Meng (2000). After the hologram is recorded and chemically processed, the 3D information is reconstructed optically and recorded slice-wise by a CCD camera (see Pu (1998) for details). Note that the pitch distance in depth direction, i.e., the distance between adjacent slices, must be small enough to resolve particle images in depth.

In order to extract the 3D coordinates of the tracer particles, particle boundaries are first located on each of the 2D slices. Particle boundaries which are located on adjacent planes and which overlap on their 2D projection are grouped and identified as belonging to the same particle. Finally, coordinates are determined via the “center-of-mass” of the enclosed volume, i.e., via the intensity weighted mean of the pixel coordinates within the particle boundaries.

3

Interrogation procedures

3.1

The PM algorithm

Let us consider two 3D “frames”, X , taken at time τ , and Y , taken at time $\tau + \Delta\tau$, obtained from a double frame single exposure HPIV experiment. After the coordinates of the particle centroids have been extracted, the input data for each IC consists of two lists $\{X_j\}$, $j = 1, \dots, J$, and $\{Y_k\}$, $k = 1, \dots, K$, of 3D particle coordinates ($X_j \in \mathbb{R}^3$ and $Y_k \in \mathbb{R}^3$) with J and K elements. The correspondences between particles j and k in the two ICs are described by the binary assignment matrix $m = \{m_{kj}\}$,

$$m_{kj} = \begin{cases} 1, & \text{if } j \text{ corresponds to } k \\ 0, & \text{else} \end{cases}, \quad (1)$$

$$m_{k(J+1)} + \sum_{j=1}^J m_{kj} = 1, \quad m_{(K+1)j} + \sum_{k=1}^K m_{kj} = 1, \quad (2)$$

where $\{m_{k(J+1)}\}$ and $\{m_{(K+1)j}\}$ are two sets of slack-variables. $J+1$ and $K+1$ are interpreted as the indices of “missing particles”. The movement of particles between frames is described by the transformation

$$T_\pi : X \rightarrow Y, \quad (3)$$

where π denotes a set of flow-field parameters.

The estimation of the correspondence matrix m provides a flow field with velocity vectors for every (non-spurious) particle pair (classic PTV). Transformation parameters π , however, are estimated for the whole IC and provide transformation parameters (e.g., local velocity, rotation or shear components) distributed on an equidistant grid (classic PIV).

Following Stellmacher and Obermayer (1999), we define a quality measure

$$E(m, \pi) = \sum_{k=1}^K \sum_{j=1}^J m_{kj} |\mathbf{Y}_k - \mathbf{T}_\pi(\mathbf{X}_j)|^2 + \alpha \left(\sum_{k=1}^K m_{k(J+1)} + \sum_{j=1}^J m_{(K+1)j} \right) + r(\pi) \quad (4)$$

for every hypothesis about the correspondences m_{kj} between particles and the parameters π of the local flow field. $E(m, \pi)$ is then minimized using an expectation-maximization (EM) algorithm, in combination with a deterministic annealing scheme (Fig. 1). In the M-step the calculation of the transformation parameters may be performed analytically or via gradient descent. If the transformation \mathbf{T}_π consists only of rotation and translation components Eq. (A1), the optimal transformation parameters $\pi = (\omega_x, \omega_y, \omega_z, t_x, t_y, t_z)$ are given by Eqs. (A3) through (A8). For arbitrary affine transformations Eq. (A2), optimal values can be estimated using Eqs. (A9) and (A10).

Table 1 summarizes a set of parameters which lead to good performance of the PM procedure. β_{stop} must be set to high values in order to obtain binary values for the

```

β = β_start
DO Annealing
  DO EM
    // E-Step: Calculation of probabilities ⟨m_kj⟩:
    FOR j = 1, ..., J+1; k = 1, ..., K+1
      IF ((j ≠ J+1) AND (k ≠ K+1))
        ⟨m_kj⟩ = exp(-β · |Y_k - T_π(X_j)|^2)
      ELSE
        ⟨m_kj⟩ = exp(-β · α)
      END IF
    END FOR
  // Normalization of m:
  DO Normalization
    ⟨m_kj⟩ = ⟨m_kj⟩ / ∑_{j=1}^{J+1} ⟨m_kj⟩ ; ⟨m_kj⟩ = ⟨m_kj⟩ / ∑_{k=1}^{K+1} ⟨m_kj⟩
    Norm = max(|rows of m|, |columns of m|)
    WHILE Norm > maxNorm
  // M-Step: Calculation of transformation π:
    ∂E / ∂π = 0 (see text)
    EM = ΔE(m, π)
    WHILE EM > maxEM
      β = β · β_mul
    WHILE β < β_stop
  
```

Fig. 1. The particle matching algorithm (Kuzmanowski et al. 1998; Stellmacher and Obermayer 1999)

Table 1. Parameter set for particle matching

Parameter	Value
β_{start}	0.5
β_{stop}	5000 or 750 (see text)
β_{mul}	1.5
α	$> \mathbf{Y}_k - \mathbf{T}_\pi(\mathbf{X}_j) ^2$
MaxNorm	0.1
MaxEM	0.05

assignment variables m_{kj} from the probabilities $\langle m_{kj} \rangle$, but good results for flow-field parameters are achieved already at lower values of β . α has to be chosen as the maximum tolerable assignment cost for a given particle pair and should be matched to the estimated errors in the particle coordinates, e.g. particle deviation.

3.2

The cross-correlation algorithm

Conventional cross-correlation (Keane and Adrian 1992) operates on gray valued images rather than on previously extracted coordinates of particle centroids. Cross-correlation relies on the assumption that the translation of particles caused by the flow field is nearly constant within the whole IC. The correlation function (correlogram)

$$c(t) = \int_{-\infty}^{+\infty} f_1(x)f_2(x+t)dx = f_1 * f_2 \quad (5)$$

is used to estimate the similarity between the two signals f_1 and f_2 which may be shifted against each other by a certain translation $t \in \mathfrak{R}^3$ (Keane et al. 1995). The location of the maximum value of the correlation function $c(t)$ corresponds to the average translation within the IC and can be found by a simple maximum search.

The correlation function can be efficiently calculated in Fourier space via the convolution (*) theorem

$$F(f_1 * f_2) = F(f_1)F(f_2) \quad (6)$$

where $F(\cdot)$ denotes the Fourier transformation.

Cross correlation requires high particle densities and performs well for velocity fields with small velocity gradients within an IC. Furthermore, cross correlation is sensitive to positional noise, because if particle images move in random positions from their expected locations, overlapping particles no longer contribute to the correlation function.

3.3

The CCC algorithm

The CCC algorithm (Pu 1998; Pu and Meng 2000) is a correlation based algorithm, which (i) operates on particle centroid coordinates (allowing for a compression factor of 10^4 compared with the original image), (ii) calculates the correlation function in the spatial domain instead of the frequency domain, (iii) estimates particle displacements between frames separately for the three coordinate axes, and (iv) allows for pairing individual particles after obtaining the correlation results.

The CCC procedure consists of two steps. In the first step, each coordinate axis is processed separately. For any coordinate axis i the coordinates of list X are fixed while the coordinates of list Y are shifted relative to the coordinates of list X along axis i with stepsize ε . The maximum shift is given by a the factor η times the size of the IC in both directions. For every shift λ and for each pair (j, k) of particle coordinates $\mathbf{X}_j \in X$ and $\mathbf{Y}_k \in Y$ a number

$$c_{jk}^{(i)}(\lambda) = \max(\mu^{(i)} - |X_j^{(i)} - Y_k^{(i)} - \lambda|, 0) \quad (7)$$

is calculated, where $\mu^{(i)}$ denotes the correlation radius on each axis, an imaginary radius of a particle. This parameter can be adjusted at will accommodate positional

noise and deformation (rotation, shear, etc.). The locations $\lambda_s^{(i)}, s = 1, \dots, \delta$, of the δ largest maxima of the sum $c^{(i)}(\lambda)$, $c^{(i)}(\lambda) = \sum_{(j,k)} c_{kj}^{(i)}(\lambda)$ (8)

over all particle pairs, are then determined for each axis i by a simple peak-finding method. In the second step, the values of the 3D cross correlations are computed for all combinations $(\lambda_s^{(1)}, \lambda_s^{(2)}, \lambda_s^{(3)})$ of shifts along the individual axes using

$$c(s_1, s_2, s_3) = \sum_{j,k} c_{jk}(\lambda_{s_1}^{(1)}, \lambda_{s_2}^{(2)}, \lambda_{s_3}^{(3)}) \quad (9)$$

where

$$c_{jk}(\lambda_{s_1}^{(1)}, \lambda_{s_2}^{(2)}, \lambda_{s_3}^{(3)}) = \max \left\{ \sum_{i=1}^3 \left(\mu^{(i)} - \left| X_j^{(i)} - Y_k^{(i)} - \lambda_{s_i}^{(i)} \right| \right), 0 \right\} \quad (10)$$

The translation vector \mathbf{t} is then given by the shift vector $(\lambda_s^{(1)}, \lambda_s^{(2)}, \lambda_s^{(3)})$ which corresponds to the maximal value of c .

The main difference between CCC and standard cross correlation is that CCC operates on the coordinates of particle centroids. As a PTV method, CCC is well adopted to the low particle density and requirements for holographic PIV methods.

The parameters of CCC are summarized in Table 2.

Note that in HPIV positional noise in the z direction is much higher than on xy planes. Therefore, $\mu^{(3)}$ (in pixel) is actually much larger compared with $\mu^{(1)}$ and $\mu^{(2)}$ to handle that noise. One $\mu^{(3)}$ means the distance between two scanned planes.

4

Benchmark results for artificial data

512 pairs of frames of artificial double frame single exposure (DFSE) data were generated as follows. Forty particle centroids were placed randomly within a sphere-shaped IC (frame X) with a diameter of 128 pixels. The corresponding particle centroids of the frame Y were then calculated by (i) selecting a random vector and rotating all particle centroids around this vector with a given angle (unless otherwise mentioned), (ii) selecting a random direction and moving all particle centroids along this direction (unless otherwise mentioned), (iii) deleting all particles that leave the interrogation window and adding a similar number of particles at random positions within the volume that enters (later referred to as move-out/in-noise), (iv) randomly changing the coordinates of a certain percentage of particles mimicking acquisition noise (unless otherwise mentioned), and (v) adding Gaussian shaped “positional” noise with standard deviation σ to all particle coordinates (unless otherwise mentioned). In order to apply cross-correlation techniques, Gaussian shaped blobs with $\sigma_p = 1.6$ px are placed at the locations given by the particle coordinate lists.

All interrogation procedures are used to extract translation parameters for each of the 512 DFSE pairs; PM also extracts rotation parameters. An interrogation is called

Table 2. Parameter set for CCC

Parameter	Value
ε	1.0 pixel
η	0.5
$\mu^{(1)}$	16 pixel
$\mu^{(2)}$	16 pixel
$\mu^{(3)}$	6 planes
δ	8

“unsuccessful” if the extracted translation parameters \mathbf{t} (or the rotation parameters ω in the case of PM) differ by more than 5% from the true values. If the true rotation parameter is zero, the threshold is set to be 5% of the maximum possible rotation (180°). Benchmark results are summarized in Figs. 2–7. The figures show the percentage of unsuccessful interrogations averaged over all 512 data sets.

Figure 2 shows the percentage of unsuccessful interrogations as a function of the percentage of acquisition noise. As long as the local flow field can be described by translation only, all interrogation procedures, except the nearest neighbor search, provide nearly equally good results.

When a rotational component is present (Fig. 3), the number of errors made by cross correlation and CCC increases dramatically. PM is also affected, but the number of unsuccessful interrogations remains under 10% for a noise level below 30%.

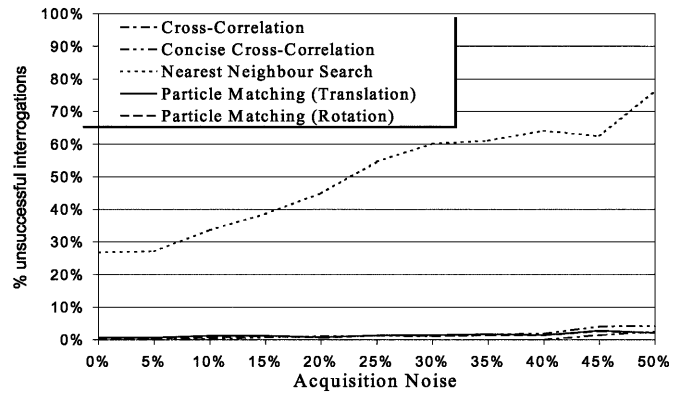


Fig. 2. Impact of acquisition noise; translation: 5% of cell width (or 6.4 px); rotation: 0°

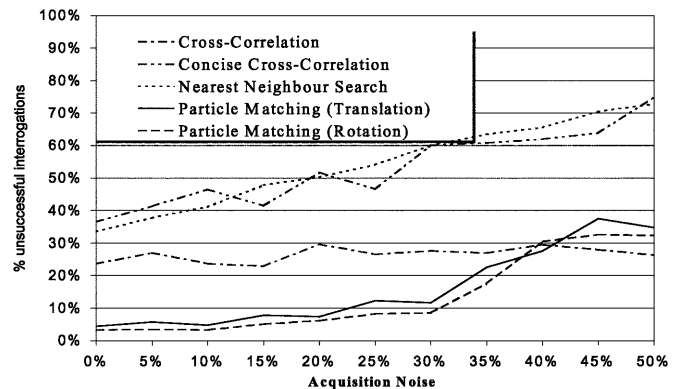


Fig. 3. Impact of acquisition noise; translation: 5% of cell width (or 6.4 px); rotation: 4°

Figure 4 shows the percentage of unsuccessful interrogations as a function of the standard deviation σ of the positional noise as a percentage of the particle diameter σ_p . For σ larger than 60% of σ_p , standard cross correlation breaks down because overlaps of corresponding particles no longer contribute to the correlation function. All PTV methods perform well, because they operate on particle distances rather than images of particle overlaps. The parameter α of PM is responsible for the strength of positional noise being tolerated by the assignment procedure.

Figure 5 shows the number of unsuccessful interrogations as a function of the ratio between the true translation and the size of the IC. All interrogation procedures except nearest-neighbor search perform with almost zero error when the value of the translation vector length is below 8% of the cell width. Above a value of 8% PTV methods show an increase in error, while correlation-based methods perform well up to a value of 14%.

When a rotation component is present (Fig. 6), however, the performance of correlation-based methods drops, while PM almost retains its performance.

Figure 7 shows the percentage of unsuccessful interrogations as a function of the rotational angle applied to the data. Correlation-based methods perform well up to an angle of 0.1 rad – particle images do not overlap any more for higher angles. PM performs well, extracting the correct transformation parameters even for large rotational angles. Figure 7 shows the major advantage of PM compared

with other methods – the capability to extract and handle higher-order flow-field components (e.g., rotation).

PIV and PTV methods have been developed for different particle densities. To ensure that benchmarks are fair and neither the PIV- nor the PTV-based methods receive advantages from the amount of particles per IC, artificial data with variable numbers of particles, but the same transformation parameters ($|t| = 12.8$ px, $\omega = 4^\circ$) have been processed. Figure 8 shows the percentage of unsuccessful interrogations as a function of the number of particles within an IC. Both correlation-based methods

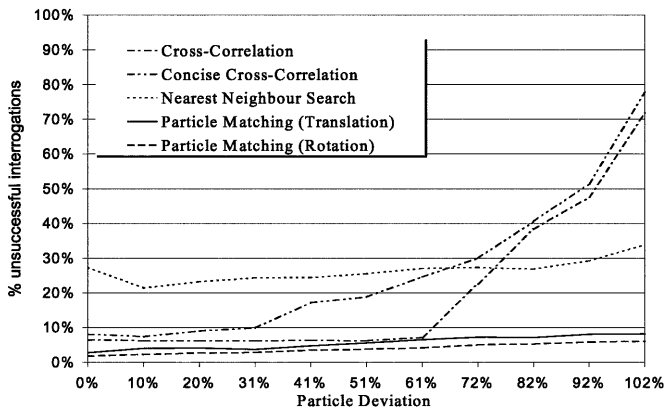


Fig. 4. Impact of positional noise; translation: 5% of cell width (or 6.4 px); rotation: 0°

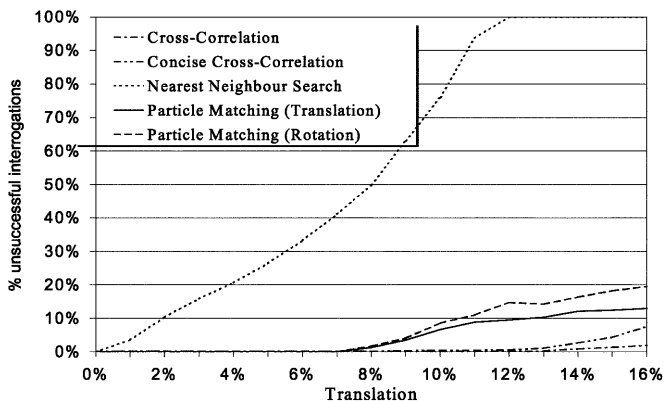


Fig. 5. Impact of “move in/out noise”; rotation: 0°

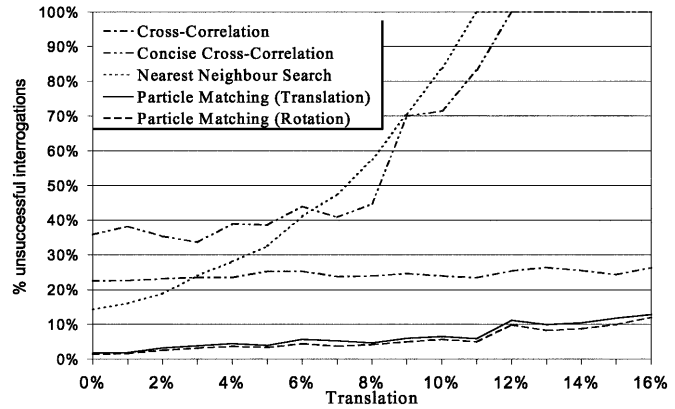


Fig. 6. Impact of “move in/out noise”; rotation: 4°

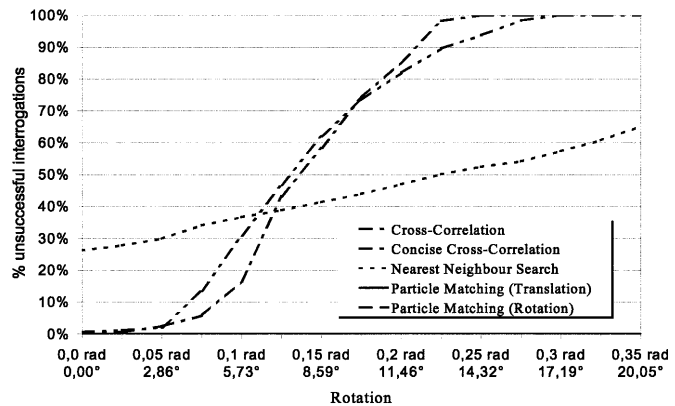


Fig. 7. Impact of rotation angle; translation: 5% of the cell width (or 6.4 px)

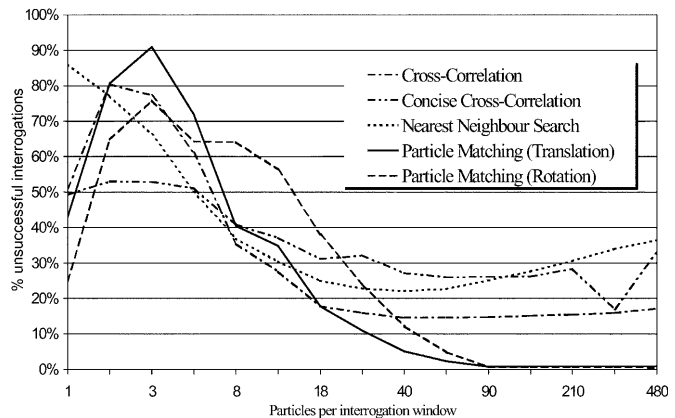


Fig. 8. Impact of particle density (particles per IC); translation: 10% of cell width (or 12.8 px); rotation: 4°

work best with 30–50 particles. The nearest-neighbor search procedure produces best results at a density of about 40 particles per IC. The PM procedure (parameterized with rotation and translation parameters) performs better the more particles are present, but good results can already be achieved with densities of roughly 40 particles per IC. One should note, that PM has to extract more transformation parameters than other interrogation procedures ($\omega_x, \omega_y, \omega_z, t_x, t_y, t_z$) and – therefore – needs more data for its estimates.

In the following, we briefly list conditions for which PM fails to extract the correct flow-field parameters:

- The number of non-spurious particle centroids drops below the number of parameters (six for translation and rotation, three for translation only).
- The annealing process is too fast, i.e., parameters $\beta_{\text{start}}, \beta_{\text{stop}}, \beta_{\text{mul}}$ are set to values for which the algorithm converges to local minima of E Eq. (4).
- The accuracy parameter MaxEM is set to values above 0.2 such that the minimum of the cost function will not be found with sufficient accuracy.
- The assignment parameter α is set to values below the standard deviation of the positional noise present in the data with the effect of corresponding particle pairs not being assigned to each other.
- The assignment parameter α is set to values which are larger than the expected distance between neighboring particles.
- The values of the transformation parameters are too high (e.g., the velocity vector is longer than 13% of the cell width).

Accuracy and stability of PM increases if IC size is enlarged along the direction of the flow. This increases the number of particles per IC, and lowers the ratio between the local translation and the size of the IC. Note that increasing IC size does not necessarily lead to a decrease in spatial resolution, because PM allows the estimation of the whole local flow field (e.g., \mathbf{t} and \mathbf{A} in Eq. A2) rather than providing a velocity estimate only. Because the estimate of the whole field becomes more accurate for increasing IC size (increasing number of particles), it counterbalances the loss in spatial resolution due to increasing IC size (lower number of estimates). Moreover, extracting higher-order transformation parameters simultaneously makes rendering a subsequent determination of the derived flow-field parameters from a discretized velocity field unnecessary.

Table 3 compares the processing times of the cross-correlation, CCC, nearest neighbor search and PM interrogation procedures (translation only).

The numbers refer to the CPU time needed for processing a single IC ($128 \times 128 \times 128$ pixels) with 40 particles. Benchmark parameters were: $\mathbf{t} = (7.4 \text{ px}; 7.4 \text{ px}, 7.4 \text{ px})^T$, i.e., $|\mathbf{t}| = 12.8 \text{ px}$ equivalent to 10% of the cell width, 10% acquisition noise, and $\sigma = 0.5 \sigma_p$. Note that the choice of the numerical values of \mathbf{t} and ω does not influence CPU times. CPU times were measured on a Sun Sparc Ultra 30 (333 MHz, 128 MB RAM, SunOS 5.6).

The table shows that PM and CCC are more than one order of magnitude faster than cross correlation. Nearest neighbor search is fastest – as expected – but could not compete in accuracy with the other methods (see Sect. 4). Note that the conventional cross-correlation scales with at least $O[(n \log n)^d]$ for implementations using FFT, where n is the size of the interrogation window in pixel and d is the dimension of the particle coordinate vectors (i.e., $d = 3$ for HPIV). PM is a factor of two faster than CCC for the estimation of the local velocity, but becomes computationally more costly if other flow-field parameters are to be determined (because expressions like Eqs. (A3)–(A10) must be evaluated). Note that numerical calculations for solving the M-step may actually speed up the method. CCC and PM only scale with $O(p^2)$, where p is the number of particles per IC, which explains their low CPU-times. Particle extraction also scales with $O(n^d)$, and this CPU time must be added to the computing times of the PTV methods. Particle extraction, however, is fast.

5 Simulation results on real-world data

Two HPIV experiments were analyzed in order to demonstrate the performance of PM and CCC interrogation procedures for real-world data. The first data set shows a simple laminar flow with small velocity gradients, while the second data set includes several vortices, as they are often found in turbulent flows. Holograms were acquired and particle centroids were extracted as described in Sect. 2. The hologram was then divided into $14 \times 11 \times 8$ ICs. Every IC was $3,968 \times 4,032 \times 3,600 \mu\text{m}$ in size and contained between 20 and 350 particle centroids. CCC and the PM (translation and rotation) interrogation procedures were applied to these two PTV data sets. The resulting vector fields are summarized in Fig. 9, where a single XY-plane was selected for demonstration purposes.

Table 3. Computational speed

Interrogation procedure	CPU time [s]			Relative CPU time
	Interrogation	Particle extraction	Complete procedure	
Cross-correlation ($64 \times 64 \times 64$)	9.4267	0.00	9.4267	9.4532
Cross-correlation ($128 \times 128 \times 128$)	96.523	0.00	96.523	96.794
Nearest neighbor	0.0049	0.32	0.3249	0.3258
CCC	1.2366	0.32	1.5566	1.5610
PM (translation only)	0.6772	0.32	0.9972	1.0000
PM (rotation and translation)	3.3724	0.32	3.6924	3.7027

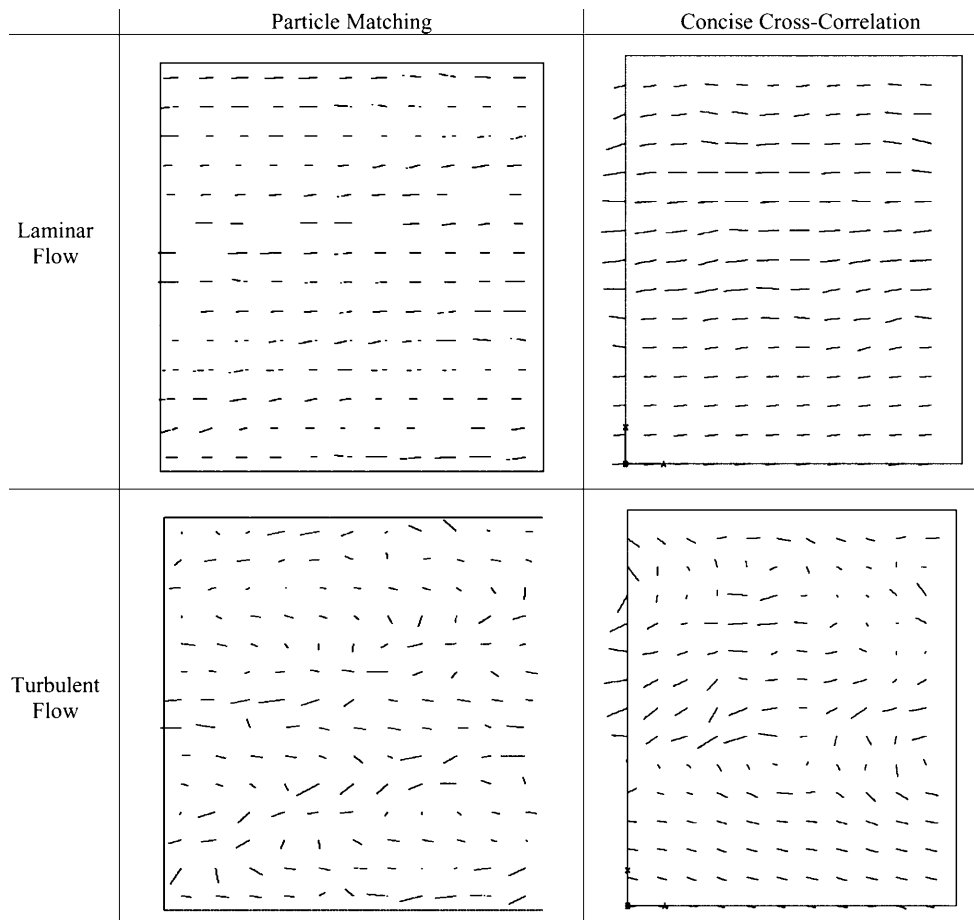


Fig. 9. Application of PM and CCC to real-world data. For explanation see text

The results for the laminar flow data set are similar for both procedures, but interrogation results strongly deviate for the case of turbulent flows. Given the benchmark results of Sect. 4, we expect PM to provide the more accurate estimations because it shows better results on artificial data for a similar scenario (high-velocity gradients).

6 Conclusion

In this contribution, we have presented a new method for the evaluation of 3D holographic PIV data, which is derived from graph and point matching algorithms commonly used in statistical pattern recognition. The match quality is evaluated by a cost function which depends on the estimated flow-field parameters and on the correspondences between particles. Optimization is based on maximum entropy inference and deterministic annealing. It is due to this optimization framework that, even in the presence of large amounts of noise, the global minimum of the cost function is found in most cases.

The new algorithm performs tolerably compared with correlation-based procedures such as CCC and allows the evaluation of regions with large velocity gradients across the interrogation window. The spatial resolution of particle matching is similar to conventional PTV or hybrid methods because the particle-matching algorithm also extracts particle correspondences. Furthermore, PM automatically yield a quality measure (the value of the cost

function) for the extracted transformation parameters and the correspondence matrix.

Finally, quantities of interest such as local vorticity may be calculated directly from the parameter list – if the transformations \mathbf{T}_π are parameterized properly. This should make rendering calculations from the measured and usually coarse-grained velocity field unnecessary, and should increase the spatial resolution for those derived quantities down to the size of an individual interrogation window. We also find that the new algorithm is more robust against acquisition (particle detection) noise and positional (particle deviation) noise than previously proposed methods.

Appendix

Analytical expressions for PM, M-step

If we are interested in the local velocity, as well as in the local rotation components, the transformation \mathbf{T}_π can be written as

$$\begin{aligned} \mathbf{T}_\pi : \mathbf{X}_j &\mapsto \mathbf{R}(\omega_x, \omega_y, \omega_z) \times \mathbf{X}_j + \mathbf{t}, \\ \pi &= (\omega_x, \omega_y, \omega_z, \mathbf{t}) . \end{aligned} \quad (\text{A1})$$

$\mathbf{R}(\omega_x, \omega_y, \omega_z) = \mathbf{R}(\omega_x) \times \mathbf{R}(\omega_y) \times \mathbf{R}(\omega_z)$ denotes a rotation matrix parameterized by the angles ω_x , ω_y and ω_z , which describe turns around the x -, y - and z axes, and $\mathbf{t} \in \mathbb{R}^3$ denotes the translation vector in space. Equations (A3) through (A8) provide the analytical expressions for the

PM, M-step. In order to estimate a general linear transformation, T_π can be modeled as:

$$T_\pi: \mathbf{X}_j \mapsto \mathbf{A} \times \mathbf{X}_j + \mathbf{t} \quad (\text{A2})$$

Equations (A9) and (A10) provide the corresponding analytical expressions for the M-step of the PM algorithm.

$$\mathbf{t}_x = \frac{\sum_{j,k=1}^{J,K} m_{kj} (Y_{jx} - X_{kx} \cos \omega_y \cos \omega_z + X_{ky} \cos \omega_y \sin \omega_z + X_{kz} \sin \omega_y)}{\sum_{j,k=1}^{J,K} m_{kj}} \quad (\text{A3})$$

$$\mathbf{t}_y = \frac{\sum_{j,k=1}^{J,K} m_{kj} (Y_{jy} - X_{kx} \{ \cos \omega_x \sin \omega_z - \sin \omega_x \sin \omega_y \cos \omega_z \} - X_{ky} \{ \sin \omega_x \sin \omega_y \sin \omega_z + \cos \omega_x \cos \omega_y \} + X_{kz} \sin \omega_x \cos \omega_y)}{\sum_{j,k=1}^{J,K} m_{kj}} \quad (\text{A4})$$

$$\mathbf{t}_z = \frac{\sum_{j,k=1}^{J,K} m_{kj} \cdot (Y_{jz} - X_{kx} \{ \cos \omega_x \sin \omega_y \cos \omega_z + \sin \omega_x \sin \omega_y \} + X_{ky} \{ \cos \omega_x \sin \omega_y \sin \omega_z - \sin \omega_x \cos \omega_y \} - X_{kz} \cos \omega_x \cos \omega_y)}{\sum_{j,k=1}^{J,K} m_{kj}} \quad (\text{A5})$$

$$\omega_x = -\tan^{-1} \left[\frac{\sum_{j,k=1}^{J,K} m_{kj} (X_{kx} \{ (Y_{jy} - t_y) \sin \omega_y \cos \omega_z - (Y_{jz} - t_z) \sin \omega_z \} - X_{ky} \{ (Y_{jy} - t_y) \sin \omega_y \sin \omega_z + (Y_{jz} - t_z) \cos \omega_z \} + X_{kz} \{ (Y_{jy} - t_y) \cos \omega_y \})}{\sum_{j,k=1}^{J,K} m_{kj} \cdot (X_{kx} \{ (Y_{jy} - t_y) \sin \omega_y + (Y_{jz} - t_z) \sin \omega_y \cos \omega_z \} + X_{ky} \{ (Y_{jy} - t_y) \cos \omega_z - (Y_{jz} - t_z) \sin \omega_y \sin \omega_z \} + X_{kz} \{ (Y_{jz} - t_z) \cos \omega_y \})} \right] \quad (\text{A6})$$

$$\omega_y = -\tan^{-1} \left[\frac{\sum_{j,k=1}^{J,K} m_{kj} (X_{kx} \{ (Y_{jy} - t_y) \sin \omega_x \cos \omega_z - (Y_{jz} - t_z) \cos \omega_x \cos \omega_z \} - X_{ky} \{ (Y_{jy} - t_y) \sin \omega_x \sin \omega_z - (Y_{jz} - t_z) \cos \omega_x \sin \omega_z \} + X_{kz} \{ Y_{jx} - t_x \})}{\sum_{j,k=1}^{J,K} m_{kj} (X_{kx} \{ (Y_{jx} - t_x) \cos \omega_z \} - X_{ky} \{ (Y_{jx} - t_x) \sin \omega_z \} - X_{kz} \{ (Y_{jy} - t_y) \sin \omega_x - (Y_{jz} - t_z) \cos \omega_x \})} \right] \quad (\text{A7})$$

$$\omega_z = -\tan^{-1} \left[\frac{\sum_{j,k=1}^{J,K} m_{kj} (X_{kx} \{ (Y_{jy} - t_y) \cos \omega_x + (Y_{jz} - t_z) \sin \omega_x \} - X_{ky} \{ (Y_{jx} - t_x) \cos \omega_y - (Y_{jy} - t_y) \sin \omega_x \sin \omega_y + (Y_{jz} - t_z) \cos \omega_x \sin \omega_y \})}{\sum_{j,k=1}^{J,K} m_{kj} (X_{kx} \{ (Y_{jy} - t_y) \sin \omega_x \sin \omega_y - (Y_{jx} - t_x) \cos \omega_y - (Y_{jz} - t_z) \cos \omega_x \sin \omega_y \} - X_{ky} \{ (Y_{jy} - t_y) \cos \omega_x + (Y_{jz} - t_z) \sin \omega_x \})} \right] \quad (\text{A8})$$

$$\mathbf{t} = \frac{\sum_{j,k=1}^{J,K} m_{kj} (Y_j - \mathbf{A} X_k)}{\sum_{j,k=1}^{J,K} m_{kj}} \quad (\text{A9})$$

$$\mathbf{A} = \begin{pmatrix} \frac{\sum_{j,k=1}^{J,K} m_{kj} (Y_{jx} - t_x) - (a_{12} X_{ky} + a_{13} X_{kz})}{\sum_{j,k=1}^{J,K} m_{kj} X_{kx}} & \frac{\sum_{j,k=1}^{J,K} m_{kj} (Y_{jx} - t_x) - (a_{11} X_{kx} + a_{13} X_{kz})}{\sum_{j,k=1}^{J,K} m_{kj} X_{ky}} & \frac{\sum_{j,k=1}^{J,K} m_{kj} (Y_{jx} - t_x) - (a_{11} X_{kx} + a_{12} X_{ky})}{\sum_{j,k=1}^{J,K} m_{kj} X_{kz}} \\ \frac{\sum_{j,k=1}^{J,K} m_{kj} (Y_{jy} - t_y) - (a_{22} X_{ky} + a_{23} X_{kz})}{\sum_{j,k=1}^{J,K} m_{kj} X_{kx}} & \frac{\sum_{j,k=1}^{J,K} m_{kj} (Y_{jy} - t_y) - (a_{21} X_{kx} + a_{23} X_{kz})}{\sum_{j,k=1}^{J,K} m_{kj} X_{ky}} & \frac{\sum_{j,k=1}^{J,K} m_{kj} (Y_{jy} - t_y) - (a_{21} X_{kx} + a_{22} X_{ky})}{\sum_{j,k=1}^{J,K} m_{kj} X_{kz}} \\ \frac{\sum_{j,k=1}^{J,K} m_{kj} (Y_{jz} - t_z) - (a_{32} X_{ky} + a_{33} X_{kz})}{\sum_{j,k=1}^{J,K} m_{kj} X_{kx}} & \frac{\sum_{j,k=1}^{J,K} m_{kj} (Y_{jz} - t_z) - (a_{31} X_{kx} + a_{33} X_{kz})}{\sum_{j,k=1}^{J,K} m_{kj} X_{ky}} & \frac{\sum_{j,k=1}^{J,K} m_{kj} (Y_{jz} - t_z) - (a_{31} X_{kx} + a_{32} X_{ky})}{\sum_{j,k=1}^{J,K} m_{kj} X_{kz}} \end{pmatrix} \quad (\text{A10})$$

References

Barnhart DH; Adrian RJ; Meinhardt CD; Papen GC (1994) Phase-conjugate holographic system for high-resolution particle image velocimetry. *Appl Opt* 33: 7159–7169

Beak S; Lee SJ (1996) A new two frame particle tracking algorithm using match probability. *Exp Fluids* 22: 23–32

Gold S; Lu CP; Rangarajan A; Pappu S; Mjolsness E (1998) New algorithms for 2D and 3D point matching. Pose estimation and correspondence. *Pattern Recognition* 31: 1019–1031

Guezennec YG; Brodkey RS; Trigui N; Kent JC (1994) Algorithms for fully automated three-dimensional particle tracking velocimetry. *Exp Fluids* 17: 209–219

Keane RD; Adrian RJ (1992) Theory of cross correlation analysis of PIV images. *Appl Sci Res* 49: 191–215

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

Malik NA; Dracos Th; Papantoniou DA (1993) Particle tracking velocimetry in three-dimensional flows. *Exp Fluids* 15: 279–294

Meng H; Hussain F (1991) Holographic particle velocimetry: a 3D measurement technique for vortex. *Fluid Dyn Res* 8: 33–52

Nishino K; Kasagi N (1989) Turbulence statistics measurement in a two-dimensional channel flow using a three-dimensional particle tracking velocimeter. In: *Proceedings of Symposium on Turbulent Shear Flows, Stanford*, pp 21–23, 22.1.1–22.1.6

Ohmi K; Lam DH (1998) Particle image velocimetry using an image relaxation method. *Ninth International Symposium on Applications on Laser Techniques and Fluid Mechanics*. pp. 10.2.1–10.2.6, Lisbon

Okamoto K (1998) Particle cluster tracking algorithm in particle image velocimetry. *JSME Int J B* 41: 151–154

Okamoto K; Hassan YA; Schnidl WD (1995) New tracking algorithm for particle image velocimetry. *Exp Fluids* 19: 342–347

- Pu Y** (1998) Advanced holographic particle imaging velocimetry (HPIV) for instantaneous full-field 3D flow measurement. Master Thesis, Kansas State University
- Pu Y; Meng H** (2000) An advanced off-axis holographic particle image velocimetry (HPIV) system. *Exp Fluids* 29: 184–197
- Scherer JO; Bernal LP** (1997) In-line holographic particle image velocimetry for turbulent flows. *Appl Opt* 36: 9309–9318
- Stellmacher M; Obermayer K** (2000) A new particle tracking algorithm based on deterministic annealing and alternative distance measures. *Exp Fluids* 28: 506–518
- Zhang J; Tao B; Katz J** (1997) Turbulent flow measurement in a square duct with hybrid holographic PIV. *Exp Fluids* 23: 373–381