13th International Symposium on Particle Image Velocimetry – ISPIV 2019
Munich, Germany, July 22-24, 2019

Learning fluid trajectory models for time-resolved PIV

Pierre Godet*, Frédéric Champagnat, Guy Le Besnerais and Aurélien Plyer

ONERA the French Aerospace Lab, Department of Information Processing and Systems, Palaiseau, France

* corresponding author: pierre.godet@onera.fr

Abstract

We present a new multiframe cross-correlation (Lucas-Kanade based) algorithm for time-resolved PIV. This algorithm leverages time coherence in image sequences by decomposing the temporal dependency of motion on an arbitrarily chosen trajectory basis. We propose to learn this basis from the data by performing a Principal Component Analysis on trajectories sampled from the studied sequence. We show on simulated data that such an approach can outperform the polynomial models classically used in multiframe PIV.

1 Introduction

Compared to classical particle image velocimetry, time-resolved PIV (TR-PIV) allows to characterize the time evolution of unsteady phenomena instead of their instantaneous and mean properties. However, as sensors and laser sources work at higher rates, it leads to a reduced signal-to-noise ratio. To deal with this issue, one can exploit the temporal coherence in a multiframe estimation approach. Furthermore, multiframe motion estimation allows computing short trajectories, instead of two-frame displacements. Thus, other quantities than instantaneous velocity can be evaluated such as fluid acceleration.

FTC of Lynch and Scarano (2013), FTEE of Jeon et al. (2014), and the LKFT approach, Yegavian et al. (2016), perform multiframe estimation for PIV assuming a polynomial time dependency. Thanks to this polynomial model, multiframe estimation gains robustness against the noise. It can be seen in the figure 1 which compares the errors on the estimated displacement fields with LKFT or with the two-frame algorithm FOLKI, Champagnat et al. (2011). Nevertheless, LKFT presents localized errors which, as we will see later, appear in regions where the polynomial model is not perfectly adapted to the movement. Here we propose to investigate the use of more precise motion models derived from the data.

For instance, in the context of computer vision, Garg et al. (2010) proposed to use principal component analysis (PCA, known as POD for Proper Orthogonal Decomposition in the fluid mechanics' community) to derive trajectory models adapted to a given video sequence. More generally, the current trend of data-based approaches motivates the development of generic multiframe motion estimation methods able to exploit a learned basis of trajectories. In this purpose, we elaborate on ideas of Champagnat et al. (2011); Yegavian et al. (2016) and derive an efficient algorithm to estimate the coefficients of the decomposition of fluid trajectories on an arbitrary basis by window-based cross-correlation analysis.

We present a first application of this algorithm on a PCA-learned basis of trajectory models. Compared to Garg et al. (2010), our estimation is based on a Lucas-Kanade algorithm instead of a variational approach. The former is more adapted to the noise level encountered in PIV application, and also leads to faster implementation.

2 Previous work

FOLKI is a Lucas-Kanade algorithm first been published for computer vision applications, Le Besnerais and Champagnat (2005), and applied later to the context of PIV, Champagnat et al. (2011). The Lucas-Kanade Fluid Trajectories (LKFT) approach, Yegavian et al. (2016), extended some ideas of FOLKI to the multiframe motion estimation for time-resolved PIV. LKFT assumes a polynomial formulation of the motion and directly estimates the coefficients by cross-correlation analysis on an interrogation window. Yegavian et al. (2016) showed in particular that temporal modelling improves the robustness to noise of LKFT. Such gain had been observed in previous references. FTEE of Jeon et al. (2014) also assumes a polynomial time

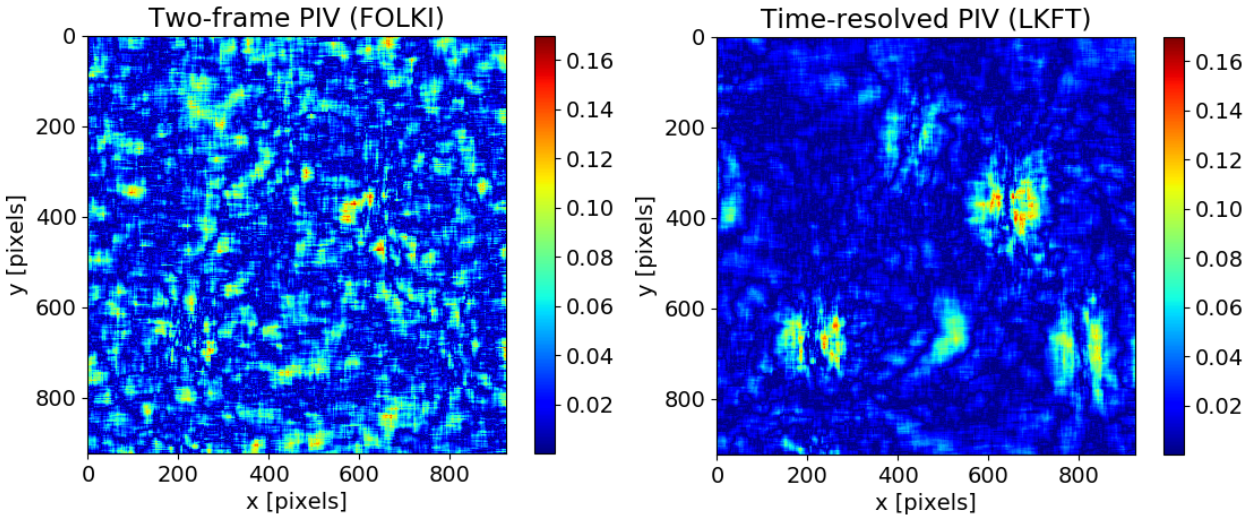

Figure 1: Error on the cross-stream displacement fields estimated by two-frame PIV, Champagnat et al. (2011) and multiframe estimation with LKFT's polynomial approach, Yegavian et al. (2016), both with an interrogation window of 33×33 pixels. Estimation realized on the simulated sequence presented below in section 4.1, degraded by an additive Gaussian noise with a level of 4 %. LKFT was runned with a polynomial degree of 3 on a sequence of 8 images.

dependency but uses a different method to estimate the coefficients. Fluid Trajectory Correlation (FTC) of Lynch and Scarano (2013) tracks independent fluid elements over multiple image-pairs and then applies a polynomial fit to each estimated trajectory.

In the computer vision community, it has been shown by Tomasi and Kanade (1992) that the image motion of a rigid scene could be described in a very low-dimensional linear subspace. Irani (2002) used this idea to compute a motion basis to constrain the multiframe optical flow estimation for a rigid motion, and Garg et al. (2010) extended this principle to the non-rigid case using PCA. PCA has also been used to learn a motion basis in Black et al. (1997); Wulff and Black (2015) but their basis vectors represent spatial dependency whereas Garg et al. (2010); Yegavian et al. (2016) model time dependency.

In fluid mechanics, PCA (or POD) is used under to estimate spatial motion models. POD on spatiotemporal patches is studied in Stapf and Garbe (2014). In contrast, we focus here on learning temporal motion models.

3 Approach

3.1 A generic multiframe motion algorithm

In Champagnat et al. (2011), the displacement field between two frames is estimated, at each pixel location k, by minimizing the sum over an interrogation window $\mathcal{W}$ of squared differences between image intensities. In a multiframe context, Yegavian et al. (2016) chose a reference frame $\mathbf{I}_{ref}$ among the N considered consecutive images, and defined a multiframe criterion as the sum of the two-frame criteria for every image pair involving the reference image:

$$\sum_{n=1}^{N} \sum_{k'} \mathcal{W}(k-k') \left(\mathbf{I}_{ref}(k') - \mathbf{I}_n(k' + \mathbf{u}(k,t_n))\right)^2 \tag{1}$$

LKFT reduces the number of indeterminates by approximating the time dependency of motion as a polynomial model. Here we consider a generalization of this approach where the motion is projected on some basis of temporal models. To be more specific: each component (horizontal u and vertical v) of the displacement is written as a linear combination of temporal models, e_i and h_i, which are vectors of $\mathbb{R}^N$, N being the length

of the sequence:

$$\begin{aligned} u(k,t_n) &= \sum_{i=1}^{d} \theta_{k,i}^{(u)} e_i[n] \\ v(k,t_n) &= \sum_{i=1}^{d} \theta_{k,i}^{(v)} h_i[n] \end{aligned} \tag{2}$$

the temporal models are polynomial (ie. $e_i[n] = h_i[n] = t_n^i$) in LKFT (Yegavian et al., 2016), or are learned on motion trajectory sample in our new method. As a result, for each pixel k of the reference image, instead of estimating the displacement at every time-step, a small set of $2d$ coefficients $\theta_{k,i}$ describes the whole trajectory passing over the pixel's location at the reference time instant.

The $2d$ coefficients are estimated by iterative minimization of the multiframe cost function (1). At each iteration, (2) is approximated by a linear least-squares criterion and its optimization is reduced to the resolution of a $2d \times 2d$ linear system for each pixel. In our approach, the first order expansion of (1) follows the inverse form of FOLKI described in Champagnat et al. (2011). As shown in the appendix below (in the case $e_i = h_i$), we take advantage of a space-time separability property to propose a very efficient two-step resolution: a $d \times d$-matrix inversion made once at the beginning (for all pixels, time-steps and iterations) and a 2×2-matrix inversion at each pixel and each iteration. Note that, although this has not been demonstrated in the appendix because of the lengthy mathematical developments, this new algorithm can handle two different bases for the horizontal and vertical components of the motion field.

Our formulation includes the LKFT paradigm. Indeed, using a polynomial model leads to results identical to the initial implementation of LKFT as described in Yegavian et al. (2016), but with a drastic reduction of the computational time and memory usage. However, the genericity of this algorithm, as it can work with any trajectory basis, makes it possible to use temporal models learned from the data.

3.2 Learning a motion basis from the data

The proposed algorithm offers the possibility to regularize motion estimation with a data-driven model: a new specialized basis of temporal models can be used for every new image sequence.

To build a basis adapted to the data, we use PCA on a set of trajectories representative of the studied sequence. Representative trajectories could be estimated from the image sequence in a preprocessing step by particle tracking velocimetry methods, or be extracted from a simulation corresponding to the experimental scenario at hand. We gather such trajectories in a large data matrix and apply a singular-value decomposition. The selected motion basis is made of the vectors related to the highest singular values. The rank of the basis can be set manually or chosen automatically by thresholding the cumulated energy of singular values.

Two different learning processes are proposed here. In the first one, a PCA is applied to temporal samples of the horizontal and vertical displacements taken indifferently together. This approach leads to a single basis for both components – $e_i = h_i$ in Eq. (2) – and is called "isotropic PCA" in the following. In the second one, horizontal and vertical samples are separated, and two PCAs are done independently on each set. Two different bases are obtained, each specialized for one component. This second process is called "asymmetric PCA" in the following.

4 Experiments

Temporal modelling allows reducing the number of estimated coefficients with respect to the classical PIV processing of each pair of consecutive images. This not only reduces computational time but also increases robustness to noise thanks to temporal redundancy. However, the estimation quality depends on the choice of the model: a trajectory stepping out of the model cannot be well estimated. The aim is to provide new trade-offs between noise and model errors.

Through different simulated experiments, we have compared the estimations given by our new generic algorithm with different basis choices:

- a polynomial basis, leading to the same result than LKFT
- a basis shared by both horizontal and vertical components estimated by the isotropic PCA process.
- two different bases, one for each component, estimated by the asymmetric PCA process.

To focus on the ability of these models to adapt to a complex motion sequence, we compute the PCA on the best data-suited trajectories we can hope for by using a subset of the simulated trajectories.

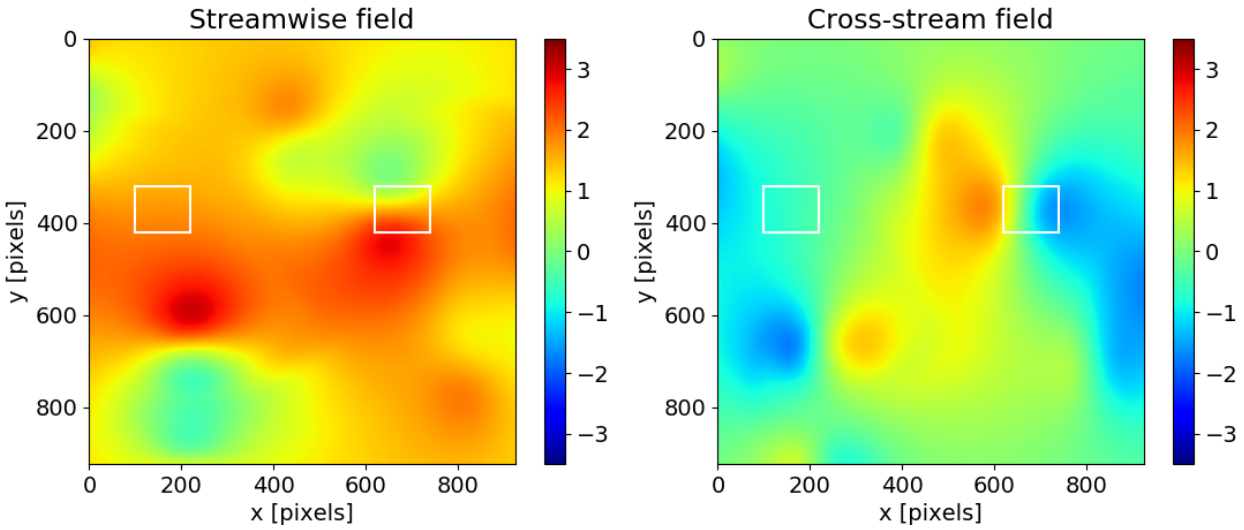

Figure 2: Streamwise and cross-stream components of the ground truth instantaneous displacement at the central time-step.

The estimated multiframe motion fields are 3D-tensors which provide for each spatial location, given by the pixel grid of the reference image, the trajectory on the whole given sequence. In all the following experiments the reference image is at the center of the sequence (for instance, for a sequence of 5 or 6 images, the reference frame is the third one). On the one hand, the result can be evaluated in the whole spatial field at one specific time-step: for the first time-step following the reference image this corresponds to an instantaneous velocity, it can be compared with other PIV results. In Sec. 4.2 we study instantaneous spatial fields of multiframe methods, a comparison with the two-frame FOLKI algorithm of (Champagnat et al., 2011) using the same interrogation window size is displayed in Fig. 1. On the other hand, by selecting a particular spatial location, the whole estimated trajectory can be studied, as proposed in Sec. 4.4.

4.1 Testing sequence

In the following experiments, we use the velocity field in the wake of a flapping foil at Reynolds number 250, obtained by direct numerical simulation, Jallas et al. (2017). The streamwise and cross-stream components of the instantaneous displacement at the central time-step are shown on figure 2. Using these simulated velocity fields, we compute trajectories and generate a particle image sequence on which the PIV methods are tested. The particles are Gaussian-shaped with a standard deviation of 0.4 pixels. To evaluate robustness to noise, an additive Gaussian noise is added to the sequence for some of the following experiments. To represent a time-resolved PIV context, this sequence is temporally well sampled, leading to interframe displacements of the order of one pixel. Working on such simulated data makes it possible to derive quantitative comparison metrics with regard to the ground truth motion. We use the ”endpoint error” defined as the norm of the difference between the estimated vector and the ground truth vector for each pixel location.

4.2 Instantaneous motion field: spatial analysis

In this section, we study the errors on the instantaneous displacement field at the time-step following the reference frame. We first compare the potential of the PCA decomposition to the polynomial model, by computing the projection error of the simulated known motion field of 2 on both bases. The projection error is the endpoint error between the ground truth field and its projection on the considered basis (polynomial or obtained by PCA). By doing so, it can be determined to what extent the different models are adapted to the studied motion. All trajectories are projected on the PCA basis on the one hand, and on the polynomial basis on the other. We consider 8 time-steps, an order 3 polynomial model and a PCA basis made of 3 vectors. According to the projection error displayed in figure 3 (upper line), the PCA leads to a better approximation in regions with high motion gradients.

We also present results of multiframe PIV processing of the generated sequence of images introduced before. Error fields on the cross-stream displacement are shown on the lower line of Figure 3. LKFT with an order 3 polynomial is compared to the proposed PCA-based multiframe estimation with 3 vectors; both methods use a 33×33 interrogation windows and on the same sequence of 8 images. These parameter choices are justified in the parametric study in section 4.3. We use a noise level of 4 % of the maximal intensity. The localized model errors seen on the projected field (upper line) match the ones found on the estimated field, and are also slightly higher with the polynomial basis. Thus, the basis learned by PCA appears better suited to the studied motion, and its use in the whole dense motion estimation process leads to better results than the polynomial basis. We did not present the results for the "asymmetric PCA", because they are very similar to those obtained with the "isotropic PCA".

The multiframe paradigm can be compared to classical two-frame PIV on figure 1 and 4. Figure 1 shows that unlike the two-frame estimation which is globally noisy, the multiframe one is well regularized except in some areas where the error is higher because the local motion steps out of the chosen model. Figure 4 enlights that when the noise level increases, the error of the two-frame method increases faster than the error of the multiframe algorithms. Multiframe estimation is more robust to noise. However, for low noise levels, the two-frame estimation error is lower than the multiframe one in the areas of high bias.

To estimate an instantaneous motion field, multiframe approaches appear more robust to noise but also exhibit localized errors in the high-gradient regions, where the model is not fully suited to the motion. In these regions, the learned PCA basis allows a reduction of the bias compared to the polynomial approach.

4.3 Parametric study

For LKFT as for our new algorithm, some parameters can be tuned :

- The radius of the interrogation window (a radius of 16 means an interrogation window of size 33×33).
- The number N of consecutive images considered for the estimation.
- The number d of vectors (temporal models) in the basis. For LKFT it corresponds to the polynomial degree.

For the problem to be well-defined, N has to be greater than d. By plotting the averaged endpoint error in the instantaneous field at the center of the sequence, as a function of these parameters, we determine their influence on the estimation quality.

The influence of the size of the interrogation window is shown on figure 5. As the window gets bigger, the error of every tested method decreases sharply, reaches an optimum and then increases slowly. If the window is too small the problem is not well-conditionned, if it is too big the estimation fails to capture fine details. The optimum of the multiframe methods varies with the dimension d of the basis: a small number of temporal models means a stronger temporal regularisation which can compensate a lack of spatial regularisation due to a smaller window. The two-frame method error is always higher, and the gap with the multiframe approaches is larger for small windows. In our case, a 33×33 window is a good choice for both $d = 2$ and $d = 3$.

Figure 6 shows the influence of the length of the sequence. The two-frame PIV error level is given for comparison but, of course, does not depend on the sequence length. By increasing the number of frames, one can hope a growing robustness to noise and a lower estimation error. This is true if the number d of temporal models in the basis is high enough to capture the complexity of the motion. As this complexity generally grows when raising the sequence length, using many frames implies to increase the rank of the basis to limit the bias error. This can be seen on figure 6: with two temporal models, the multiframe methods are better than the two-frame's only for short sequences (4 or 5 frames), and then the error radically increases; longer sequences can be used with 3 models, until 8 or 10 frames before the error goes up.

Finally, note that in almost all these experiments, PCA models lead to lower or equivalent errors than the polynomial model, with a slight advantage for the asymmetric PCA model.

4.4 Trajectories: temporal analysis

Multiframe paradigm also gives the possibility to consider the trajectories over the whole sequence, some are presented on figure 7. For this evaluation, we consider a longer sequence, with 25 frames, 3 temporal models and no added noise on the image sequence. We focus on two different areas of the field (underlined by a white square in Fig. 3): the first one exhibit high motion gradients, the second one has smoother

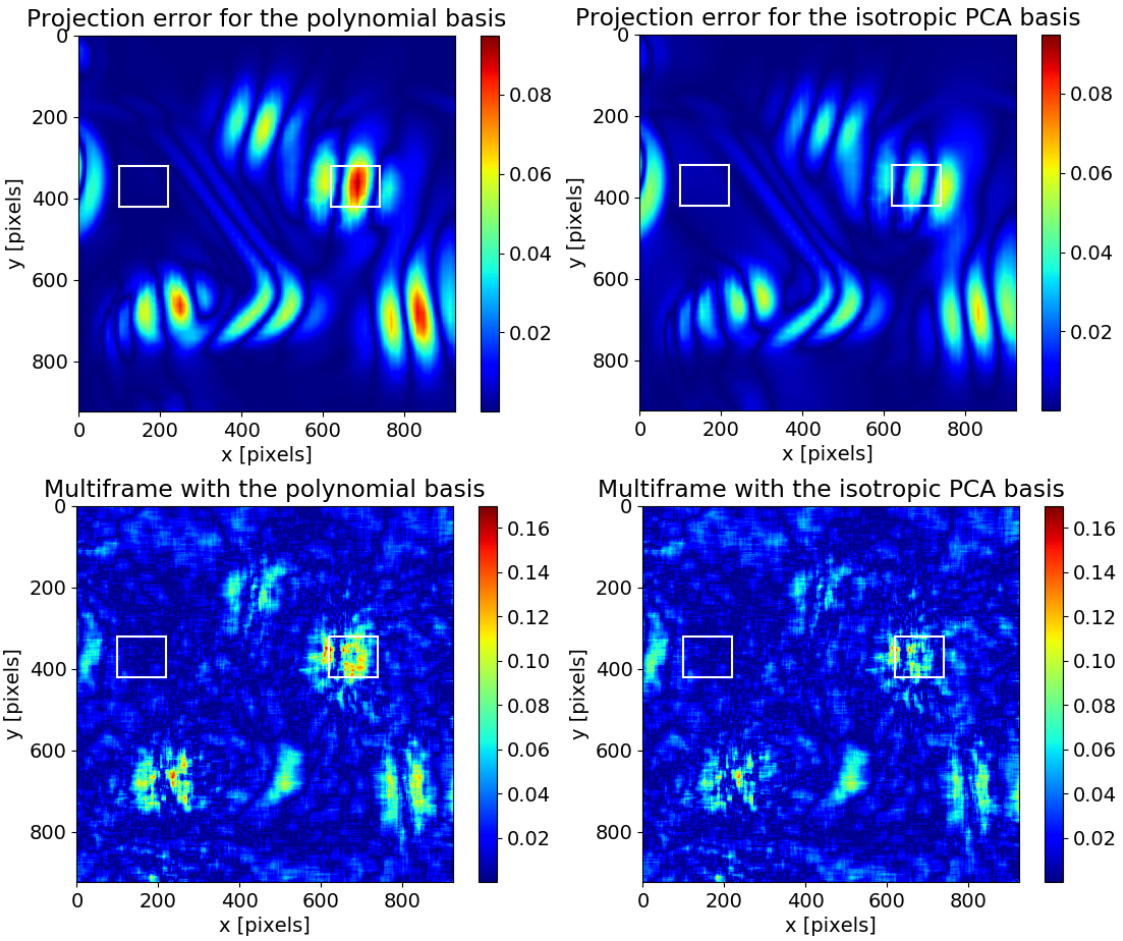

Figure 3: Error (in pixels) on the cross-stream component of the instantaneous displacement at the central time-step. First row: projection error on the polynomial (left) and PCA (right) bases. Second row: error on the displacement fields estimated by multiframe estimations with LKFT's polynomial approach (left) and the proposed algorithm using an "isotropic PCA" basis (right), both with an interrogation window of 33×33 pixels. A sequence of 8 images was used, LKFT was tuned with a polynomial degree of 3, and the PCA basis was composed of 3 vectors. A Gaussian noise with a level of 4 % was added to the images.

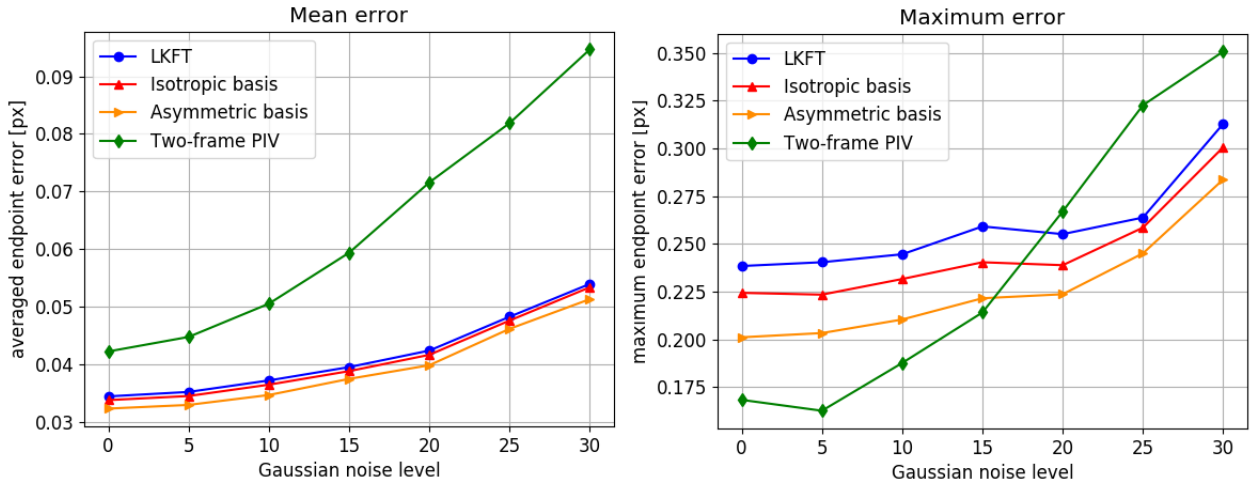

Figure 4: Endpoint error [px] on the instantaneous displacement, spatial average and maximum (for the maximum we ignore the 50 pixels nearest to each boundaries), as a function of the additive Gaussian noise level. Sequence of 5 images, basis of 2 vectors for multiframe methods, interrogation window of 33×33 pixels. A noise level of ten corresponds to 4% of the maximum intensity.

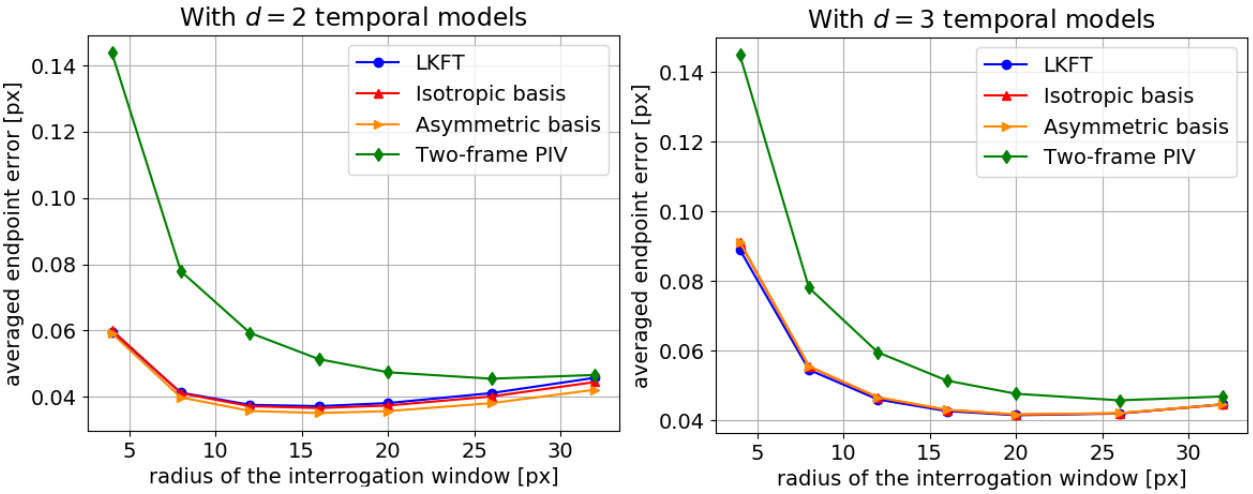

Figure 5: Averaged endpoint error [px] on the instantaneous displacement, as a function of the size of the interrogation window. Sequence of 5 images, basis of d vectors for multiframe methods. A Gaussian noise with a level of 4 % was added to the images.

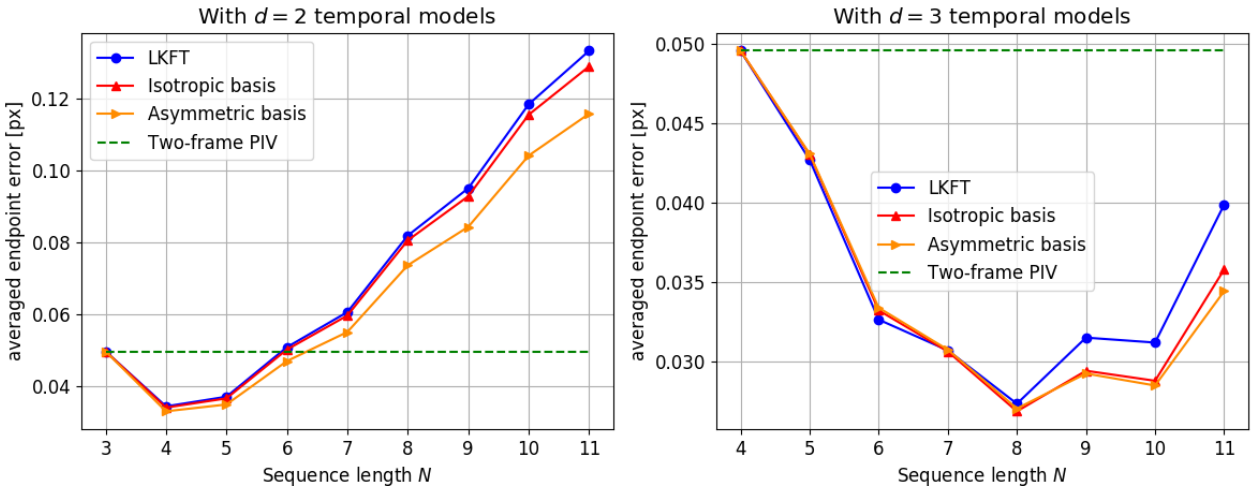

Figure 6: Averaged endpoint error [px] on the instantaneous displacement, as a function of the number of images in the sequence, with an interrogation window of 33×33 pixels. A Gaussian noise with a level of 4 % was added to the images. The two-frame PIV error level does not depend on the sequence length but is given for comparaison.

motions. We compare the estimated trajectories by our algorithm for each basis choice to the ground truth trajectories. In the high motion gradient case (figure 7, first row), compared to the polynomial model, the "isotropic PCA" significantly improves the trajectory estimation. This is particularly noticeable on the left trajectories of the high-gradient area. The "asymmetric PCA" provides some further minor improvements. In the smoother case (figure 7, second row), the polynomial model is globally better. The PCA approaches make some errors, which are a bit higher with the "isotropic PCA".

The PCA bases represent well the non-polynomial motions; however, by construction, such bases focus on the statistically dominant trajectories. As a consequence, this method is unlikely to give a reasonable estimate of a motion which only appears in a small part of the field, and which is too different from the rest of the field.

5 Conclusion

We have proposed a novel and efficient algorithm for multiframe PIV using an arbitrary temporal basis. The genericity of the proposed algorithm opens the way to better noise/model error trade-off compared to polynomial approaches. As a first application, we proposed to use a basis learned on the studied sequence by PCA. We have evaluated this approach on a simulated unsteady flow, and obtained better results than with the polynomial model.

This work can be considered as a proof of concept on learned temporal models for multiframe motion estimation. The next step will be to apply it on real time-resolved PIV data. This may be challenging because of the following issues. First of all, image degradation encountered in real datasets are much more complicated than the Gaussian noise used in our simulation study. Moreover, a strategy for learning basis in a real experiment has to be designed. It could rely on a preprocessing of the data, for instance exploiting trajectories obtained by particle tracking, or use trajectories sampled from a representative numerical simulation.

Finally, in the proposed method, the basis used is the same for every spatial location. Further improvement could be made by using different bases for different regions of the field. Methods combining motion estimation and segmentation could be explored here.

Acknowledgements

The authors are grateful to Olivier Marquet and Benjamin Leclaire (ONERA/DAAA) who provided the test data used in this work. Benjamin Leclaire also brought a precious help through several discussions. This

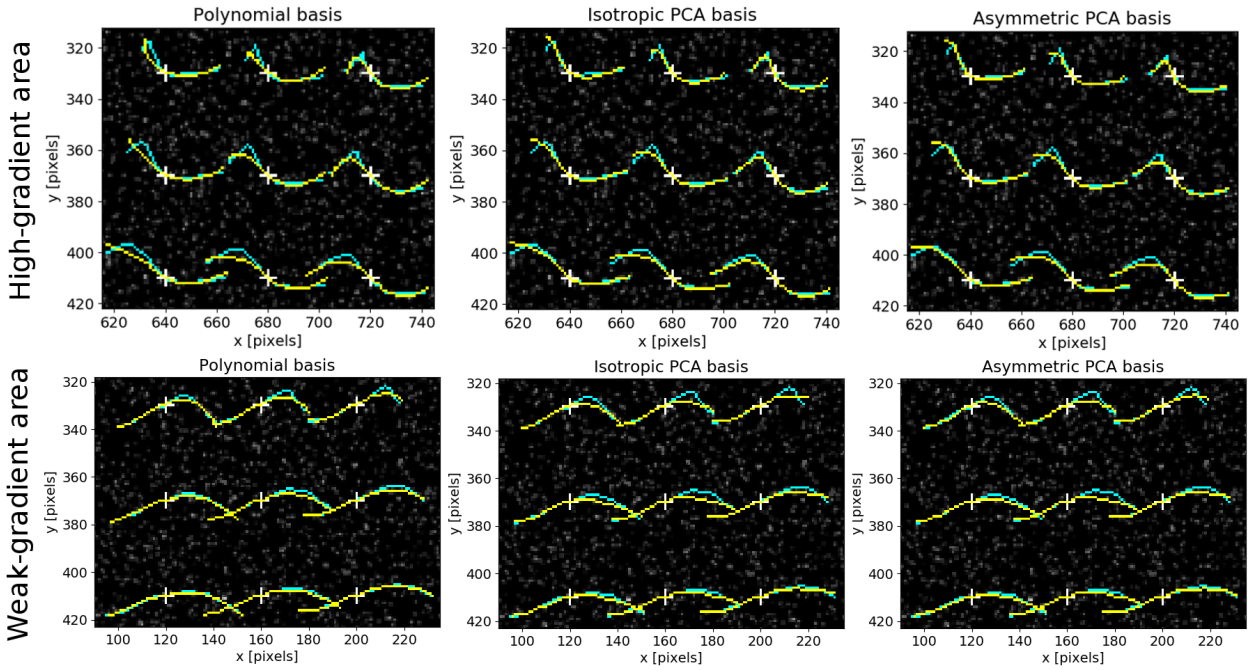

Figure 7: Estimated (yellow) and ground truth (blue) trajectories for the different methods, with 3 temporal models and a 25-image sequence. Two areas are represented : where motion gradients are strong (first row) or weak (second row). The corresponding areas are squared in white in other preceding figures.

work is supported by DGA, the French agency of defense.

References

Black MJ, Yacoob Y, Jepson AD, and Fleet DJ (1997) Learning parameterized models of image motion. in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. pages 561–567. IEEE

Champagnat F, Plyer A, Le Besnerais G, Leclaire B, Davoust S, and Le Sant Y (2011) Fast and accurate PIV computation using highly parallel iterative correlation maximization. *Experiments in fluids* 50:1169

Garg R, Pizarro L, Rueckert D, and Agapito L (2010) Dense multi-frame optic flow for non-rigid objects using subspace constraints. in *Asian Conference on Computer Vision*. pages 460–473. Springer

Irani M (2002) Multi-frame correspondence estimation using subspace constraints. *International Journal of Computer Vision* 48:173–194

Jallas D, Marquet O, and Fabre D (2017) Linear and nonlinear perturbation analysis of the symmetry breaking in time-periodic propulsive wakes. *Physical Review E* 95:063111

Jeon YJ, Chatellier L, and David L (2014) Fluid trajectory evaluation based on an ensemble-averaged cross-correlation in time-resolved piv. *Experiments in fluids* 55:1766

Le Besnerais G and Champagnat F (2005) Dense optical flow by iterative local window registration. in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*. volume 1. pages I–137. IEEE

Lynch K and Scarano F (2013) A high-order time-accurate interrogation method for time-resolved PIV. *Measurement Science and Technology* 24:035305

Stapf J and Garbe CS (2014) A learning-based approach for highly accurate measurements of turbulent fluid flows. *Experiments in Fluids* 55:1799

Tomasi C and Kanade T (1992) Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision* 9:137–154

Wulff J and Black MJ (2015) Efficient sparse-to-dense optical flow estimation using a learned basis and layers. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 120–130

Yegavian R, Leclaire B, Champagnat F, Illoul C, and Losfeld G (2016) Lucas–Kanade fluid trajectories for time-resolved PIV. *Measurement science and Technology* 27:084004

A Algorithm details: the case of a single basis for both coordinates

To minimize the criterion of equation 1 we adopt an inverse Lucas-Kanade iteration, Champagnat et al. (2011), for better computational performances. The aim is to find, at every pixel location k in the reference frame, the trajectory $(\mathbf{u}(k,t_n))_{n\in[1..N]}$ minimizing the linearized criterion:

$$\sum_{n=1..N}\sum_{k'\in\mathcal{W}(k)}\left(\varepsilon(k',n)-(\nabla I_{ref})^{\mathrm{T}}(k')\cdot\mathbf{u}(k,t_n)\right)^2$$

with

$$\varepsilon(k',n)=I_{ref}(k')-\tilde{I}_n(k'+\mathbf{u}^{(0)}(k',t_n))+(\nabla I_{ref})^{\mathrm{T}}(k')\cdot\mathbf{u}^{(0)}(k',t_n)$$

The number of indeterminates is reduced thanks to the decomposition:

$$\mathbf{u}(k,t_n)=\begin{pmatrix}u(k,t_n)\\v(k,t_n)\end{pmatrix}=\mathrm{T}_n\theta_k \tag{3}$$

where θ_k stands for the $2d$ coefficients:

$$\theta_k=[\theta^{(u)}_{k,1},\dots\theta^{(u)}_{k,d},\theta^{(v)}_{k,1},\dots,\theta^{(v)}_{k,d}]^{\mathrm{T}}$$

and, T_n is a $2\times 2d$-matrix representing the time dependency:

$$\mathrm{T}_n\triangleq\begin{bmatrix}E_n^{\mathrm{T}} & 0\\ 0 & E_n^{\mathrm{T}}\end{bmatrix}\text{ with }E_n\triangleq[e_1[t_n]\cdots e_d[t_n]]^{\mathrm{T}}$$

with $e_1,\cdots,e_d$ any basis of temporal models. To be more specific, the two components of

$$\mathbf{u}(k,t_n)=\begin{pmatrix}u(k,t_n)\\v(k,t_n)\end{pmatrix}$$

can be written as:

$$u(k,t_n)=\sum_{i=1}^{d}\theta^{(u)}_{k,i}e_i[t_n]$$

$$v(k,t_n)=\sum_{i=1}^{d}\theta^{(v)}_{k,i}e_i[t_n]$$

The criterion to minimize becomes:

$$J_k(\theta) = \sum_{n=1..N}\sum_{k'\in\mathcal{W}(k)}\left(\varepsilon(k',n)-(\nabla I_{ref})^{\mathrm{T}}(k')\mathrm{T}_n\theta\right)^2 \tag{4}$$

$$= \theta^{\mathrm{T}}\mathrm{A}_k\theta-2b_k^{\mathrm{T}}\theta+\text{constant} \tag{5}$$

its solution is:

$$\theta_k = \mathrm{A}_k^{-1} b_k, \tag{6}$$

with A_k a $2d \times 2d$-matrix given by:

$$\begin{aligned} \mathrm{A}_k &\triangleq \sum_{n=1..N} \sum_{k' \in \mathcal{W}(k)} \mathrm{T}_n^{\mathrm{T}} \nabla I_{ref}(k') (\nabla I_{ref})^{\mathrm{T}}(k') \mathrm{T}_n \\ &= \sum_{n=1..N} \mathrm{T}_n^{\mathrm{T}} \left(\sum_{k' \in \mathcal{W}(k)} \nabla I_{ref}(k') (\nabla I_{ref})^{\mathrm{T}}(k') \right) \mathrm{T}_n \end{aligned} \tag{7}$$

This comes from the separability in space and time, equation 3.
The $2d$-vector b_k is given by

$$\begin{aligned} b_k &\triangleq \sum_{n=1..N} \sum_{k' \in \mathcal{W}(k)} \mathrm{T}_n^{\mathrm{T}} \nabla I_{ref}(k') \varepsilon(k', n) \\ &= \sum_{k' \in \mathcal{W}(k)} \left(\sum_{n=1..N} \mathrm{T}_n \varepsilon(k', n) \right)^{\mathrm{T}} \nabla I_{ref}(k') \end{aligned} \tag{8}$$

We then rewrite the matrix A_k of Eq. (7) by defining:

$$\begin{bmatrix} \alpha_k & \beta_k \\ \beta_k & \gamma_k \end{bmatrix} \triangleq \sum_{k'} \nabla I_{ref}(k') (\nabla I_{ref})^{\mathrm{T}}(k')$$

and the $d \times d$-matrix T:

$$\mathrm{T} \triangleq \sum_n E_n^{\mathrm{T}} E_n$$

so that we get:

$$\mathrm{A}_k = \begin{bmatrix} \alpha_k \mathrm{T} & \beta_k \mathrm{T} \\ \beta_k \mathrm{T} & \gamma_k \mathrm{T} \end{bmatrix},$$

From this, we have the following expression of the inverse:

$$\mathrm{A}_k^{-1} = \frac{1}{\alpha_k \gamma_k - \beta_k^2} \begin{bmatrix} \gamma_k \mathrm{T}^{-1} & -\beta_k \mathrm{T}^{-1} \\ -\beta_k \mathrm{T}^{-1} & \alpha_k \mathrm{T}^{-1} \end{bmatrix}, \tag{9}$$

The main operation to compute the inverse of A_k is then the computation of the inverse of T, which can be done only once, for all frames and iterations. At each iteration, it will only remain the computations of α_k, β_k and γ_k.

The time and space separability and the use of a shared temporal basis (making T_n diagonal) reduce computational costs. In the case of two different bases, the separability still exists but is less straightforward and needs additional computations such as Cholesky factorizations and eigenvalues decompositions.