Computationally efficient sparse algorithms for tomographic PIV Reconstruction

P. Cornic¹, F. Champagnat¹, A. Cheminet², B. Leclaire², G. Le Besnerais¹
¹ Department of Modeling and Information Processing, ONERA the French Aerospace Lab
philippe.cornic@onera.fr
² Department of Fundamental and Experimental Aerodynamics, ONERA the French Aerospace Lab

ABSTRACT

“Back to physical particles” is the idea behind this paper. We introduce the physically sound Point Spread Function (PSF) model for tomographic PIV reconstruction, that aims at sparse and spiky reconstructions. Using the PSF, we show that the tracer particles are quasi-systematically in the direct vicinity of local maxima of reconstructions obtained with MLOS. Taking advantage of both the PSF model and this local maxima extraction (which we call “LocM”), we manage to dramatically reduce the dimensionality of the reconstruction that follows. For this reconstruction, we consider the SMART algorithm applied on this local maximum restriction, and also introduce a sparse algorithm, CoSaMP. We investigate the performances of LocM-SMART and LocM-CoSaMP, compared to the traditional, blob-like, MLOS-SMART used in particular by [2]. LocM-XX methods combine dramatic computational gains and a very good reconstruction accuracy. LocM-CoSaMP in particular reaches equivalent or better performances than classical MLOS-SMART especially if a voxel to pixel ratio equal to 1/2 is adopted.

1. Introduction

The reconstruction of the 3D distribution and intensity of the illuminated particles is the critical step of 3D PIV. A major problem with this technique is the size of the linear system which has to be inverted for the 3D reconstruction step. At the very beginning of tomographic PIV [1], huge linear systems had to be inverted. An initial multiplicative back-projection step (MLOS, for Multiplicative Line Of Sight, introduced in [2]) allowed to reduce the number of nonzero voxels that have to be considered. Nevertheless, for large images, even this may lead to hardly tractable data amounts, or very long processing times. Moreover, it is well known that, despite its large size, the true volume is in fact very sparse. This property underlies some recent works such as [3][4]. However, these references do not address the practical issue of reducing the size of the problem, and, although accounting for sparsity, they are actually more computationally intensive than previous approaches.

Unlike the dominant tomographic PIV paradigm of [1], we adopt a physically sound “particle approach”. It leads to reconstruction of truly point-like particles with an image formation model accounting for the point spread functions (PSF) of the imaging system. The detailed principle of this approach is the subject of a companion communication [5], and thus will be only briefly recalled in section 2. From an algorithmic point of view, the main advantage is that only one voxel is needed to explain a particle's image. This paves the way to the dramatic dimensionality reduction we focus on.

Our contribution is twofold. The first contribution of this paper is to provide theoretical evidence and results from Monte-Carlo tests that true particles are found near local maxima of the volume obtained with MLOS. Since a single voxel can render a particle image, we show that only the columns of the weight matrix corresponding to local maxima of MLOS can be retained. This leads to a class of methods that we denote as LocM-XX, where XX refers to the algorithm used to solve the linear system. Our second contribution is then to use a reconstruction algorithm based on sparsity, CoSaMP (Compressed Sampling Matching Pursuit) introduced in [7] for compressed sensing. It is made possible thanks to the underlying particle approach.

We then present a simulation study that comparatively evaluates two algorithms of the LocM-XX class: LocM-CoSaMP and LocM-SMART. We show that these methods can reach a performance equivalent or better than classical methods, while being able to reconstruct large volumes with very good computational efficiency. As shown in [5], we confirm that, for the considered PSF, the commonly agreed voxel to pixel ratio (v/p) equal 1 is not the most convenient for these algorithms: voxel to pixel ratio equal ½ improves significantly the performances, while remaining computationally efficient.

It should be emphasized that our aim here is clearly to localise particles, ie. to determine which voxels are occupied by a particle and to estimate their intensities. Refining the particle's position inside the voxel (in order to reach subvoxel accuracy in the displacement estimation) can be done in a second step, which, in our opinion should be handled together with 3D displacement field computation.

This paper is organised as follows. Section 2 introduces the PSF model. Section 3 provides the rationale of using only MLOS local maxima. Section 3.2.2 introduces performance measures adapted to the particle approach. Section 4
presents the principle of the reconstruction algorithms used. Finally, section 5 is devoted to the presentation and discussion of the results.

2. PSF model for Tomographic PIV: a particle approach

It is well known that in most common optical setups with a voxel to pixel ratio (v/p) equal 1, the true particles’ sizes, at most a few microns in the air, are far smaller than the voxel size. Despite that, conventional tomographic PIV explains the image patterns by particles several voxels in diameter, and leads to a blobs reconstruction. Its worthy to note that for synthetic experiments, the images are usually generated using spherical gaussian particles several voxels in diameters [2][12]. The “particle” approach we adopt attempts to get closer to the physical image formation. The point spread function (PSF) describes the response of an imaging system to a point source. As the particles can be seen as point sources, i.e. their geometric image is far smaller than the pixel, the image patterns they generate are given by a PSF. This is the basis of the particle approach we consider here, as illustrated on the left hand side of figure 1. The approach initially introduced by [1], and widely used since then, differs precisely by its tomographic nature, in that it considers instead a pseudo-volume made of blobs which are several voxels wide, and models the imaging process as a combination of line integrals, as sketched on the right hand side of Figure 1. Note that some authors [11][12] have used the concept of PSF for increasing the accuracy of the estimation of particle's location. However these works remain in the tomographic paradigm of [1] and reconstruct blobs, while we adopt a particle approach and thus aim at a sparse reconstruction.

Figure 1: (left) particle approach ; (right) classical tomographic model

In the particle approach, the image of a volume containing P particles with intensity $E_p$ located at $X_p \in \mathbb{R}^3$ for $p=\{1...P\}$, is given by:

$$I(x) = \sum_{p=1}^{P} E_p h(x - F(X_p)) \text{ with } x \in \mathbb{R}^2$$  (1)

where $x=(x,y)$ denotes a pixel’s position in the image plane, $h$ is the so-called Point Spread Function, defined on the entire image support, and $F: \mathbb{R}^3 \rightarrow \mathbb{R}^2$ denotes the geometric projection from 3D space to the image plane.

We assume that the PSF is a separable function, i.e. $h(x,y)=h(x).h(y)$. We use a single 1D PSF for both directions, resulting from the convolution of a Gaussian function $g_{\sigma_{psf}}$ accounting for diffraction and defocalisation blur and a gate function $\Pi_\Delta$ for spatial integration over the detector’s surface:

$$h(x)=g_{\sigma_{psf}}(x) = \int_{-\infty}^{\infty} \Pi_\Delta(t) g_{\sigma_{psf}}(x-t) dt = \frac{1}{2} \left( \text{erf} \left( \frac{x + \frac{1}{2} \Delta}{\sqrt{2} \sigma_{psf}} \right) - \text{erf} \left( \frac{x - \frac{1}{2} \Delta}{\sqrt{2} \sigma_{psf}} \right) \right)$$  (2)

In this work, we assume that $\sigma_{psf}$ does not depend on the depth, i.e., defocalisation blur is supposed to be uniform in the reconstructed volume.

Our companion paper [5] studies in detail the benefits of the particle approach for 3D PIV reconstruction performance. Here we focus on the computational gain associated to such an approach, as it opens the way to efficient sparse algorithms in the context of 3D PIV, which will be introduced in section 4.

3. Particles and local maxima of MLOS

Given that the image pattern of a particle can be explained by only one voxel, determining a priori which voxels may be occupied by a particle is of great interest to reduce the dimensionality of the problem. [2] proposed to use an initial multiplicative back-projection step (MLOS) to reduce the number of nonzero voxels that have to be considered at each step. However all nonzero voxels obtained after MLOS are far from being occupied by true physical particles. We provide hereafter theoretical evidence, and results from synthetic tests, supporting what we call the “LocM property”, that is, that true particles are quasi-systematically located at (or close to) Local Maxima (LocM) of the MLOS function.
3.1. Theoretical elements

In this theoretical framework we define the MLOS mathematical function. It is closely related to the MLOS reconstruction.

Let us suppose that we have \(N_c\) camera imaging \(P\) particles with intensity \(E_p\) located at \(X_p \in \mathbb{R}^3\). For the sake of simplicity, we deal with continuous functions, that is we suppose the interpolation can be precise enough to approximate the continuous function. The continuous MLOS function can then be written as:

\[
\forall X \in \mathbb{R}^3 \quad \text{MLOS}(X) = \frac{1}{N_c} \sum_{k=1}^{N_c} \prod_{p=1}^{P} E_p h(F_k(X) - F_k(X_p))
\]

It is worth to note that the MLOS function is defined over \(\mathbb{R}^3\) and doesn't depend on a particular discretization scheme of the voxel space.

The PSF function \(h\) is maximum for \((0,0)\). Thus, beginning with the simple case of a single particle \((P=1)\) located in \(X_i\), one can see that \(\text{MLOS}(X)\) is maximum for \(X=X_i\).

In the following, we suppose that \(h\) is smooth and compactly supported, what is indeed true in practice:

\[
\forall X \in \mathbb{R}^3 \quad h(x) = 0 \quad \text{if} \quad \|x\| > 3 \cdot \sigma_{psf}
\]

In the sequel, we define the cone of a particle \(p\) associated with camera \(k\) as:

\[
C_k(X_p) = \{ X \in \mathbb{R}^3 \mid h(F_k(X) - F_k(X_p)) \neq 0 \}
\]

We then consider the case of several particles, but without interaction between them. In this case, \(N_c\) cones associated with the \(N_c\) cameras intersect if and only if they all belong to the same particle.

This condition can be reformulated as:

\[
\forall X \in \mathbb{R}^3 \quad \prod_{k=1}^{N_c} h(F_k(X) - F_k(X_p)) = 0 \quad \forall (a_1, \ldots, a_{N_c}) \in [1..P]^\mathbb{N} \quad \text{not all equal}
\]

Under this hypothesis, it can be shown that the MLOS function can be written as:

\[
\forall X \in \mathbb{R}^3 \quad \text{MLOS}(X) = \frac{1}{N_c} \sum_{k=1}^{N_c} \prod_{p=1}^{P} E_p h(F_k(X) - F_k(X_p))
\]

Let \(V_p\) be the 3D support of the MLOS function for a particular particle \(p\), defined as the intersection of all its cones:

\[
V_p = \cap C_k(X_p) \quad k=1..N_c
\]

Under no interactions hypothesis, it can be shown that \(\forall p \neq q \quad V_p \cap V_q = \emptyset\) and

\[
\forall X \in V_p : \quad \prod_{k=1}^{N_c} h(F_k(X) - F_k(X_p)) = 0 \quad \forall X \notin V_p : \quad \prod_{k=1}^{N_c} h(F_k(X) - F_k(X_p)) = 0
\]

Thus,

- if \(X \in V_i, i=1..P\) : \(\text{MLOS}(X) = \frac{1}{N_c} \prod_{k=1}^{N_c} E_k h(F_k(X) - F_k(X_i))\)
- if \(X \notin \cup V_i\) : \(\text{MLOS}(X) = 0\)

Hence, \(X_{p} \quad p=1..P\) are local maxima of the MLOS function.

The general case, when interactions between particles occur, is hardly tractable with the formula. Indeed, in this case, all local maxima are not true particles since the cones from different particles intersect and generate ghosts. For four cameras, it can be written as:

\[
\forall X \in \mathbb{R}^3 \quad \text{MLOS}(X) = \frac{1}{N_c} \sum_{i,j=1}^{P} E_i E_j E_k E_l h(F_i(X) - F_i(X_j)) h(F_j(X) - F_j(X_k)) h(F_k(X) - F_k(X_l)) h(F_l(X) - F_l(X_i))
\]

which can be expanded in:

\[
\forall X \in \mathbb{R}^3 \quad \text{MLOS}(X) = \frac{1}{N_c} \sum_{i,j=1}^{P} h(F_i(X) - F_i(X_j)) + \frac{1}{N_c} \sum_{i,j,k=1}^{P} E_i E_j E_k h(F_i(X) - F_i(X_j)) h(F_j(X) - F_j(X_k)) h(F_k(X) - F_k(X_i)) h(F_i(X) - F_i(X_k)) h(F_j(X) - F_j(X_i)) h(F_k(X) - F_k(X_j)) h(F_i(X) - F_i(X_l)) h(F_j(X) - F_j(X_l)) h(F_k(X) - F_k(X_i)) h(F_l(X) - F_l(X_i))
\]

\(X=X_p(p=1..P)\) is a local maximum for the first sum, the second sum being either null or positive. Without claiming to give a rigorous mathematical demonstration, we will report empirical evidences that \(X=X_p(p=1..P)\) remains with a high
probability a local maximum of the MLOS function.

In practice, we deal with a discretised voxel space, which makes the demonstration more difficult; however as shown in section 3.3 below (Figure 2) the property remains essentially true. Before that, we need to briefly introduce the synthetic tests and the performance measures that will be used in this paper.

3.2. Synthetic tests

3.2.1. Setup and parameters

We here describe the setup and parameters defining the synthetic test cases that will be used throughout the article. All our simulations involve four cameras, which are positioned on a single side of the laser sheet at the vertices (\(\pm \frac{1}{2}, \pm \frac{1}{2}, \frac{1}{\sqrt{2}}\)) of a square of 1 meter side. They are positioned at 1 meter from the centre of the reconstructed volume located at (0,0,0) and point at it. The pin-hole model is assumed for the cameras (without Scheimpflug adapter) and calibration is supposed to be perfectly known. The focal length is 100 mm (thus the magnification factor M is equal to 0.1), and the CCD size is 10 µm with a fill factor of 1. The size of the back-projected pixel is 0.1 mm. The images' size, and hence the field of view, depend on the test cases and will be specified for each simulation.

The laser sheet is modelled as a parallelepiped of 20 mm thick. The reconstructed volume, also 20 mm thick, is the smallest parallelepiped including the illuminated volume seen by all the cameras. The pin-hole model makes it easy to characterise the illuminated volume seen by one or several cameras. The illuminated volume common to all the cameras is obtained by intersecting 4 square pyramids and the parallelepiped laser sheet. It depends on field of view and is given for each case.

The tracers particles are uniformly distributed in the light sheet volume. The density is controlled by the particle per voxel count (ppv). Horizontal and vertical extension of the sheet is larger than the field of view covered by all the cameras. It is important to notice that all the illuminated particles cannot be seen by all cameras, as in real dataset. Our companion paper [7] and § 5.6 give further details on this important issue. The scattered light is proportional to the square of the particle physical diameter \(d_p\). Consequently, the intensity of a particle depends on its diameter and its depth. It is given by: \(I = I_0 d_p^2 e^{-x^2/\sigma^2}\) where \(I_0\) is a constant, \(\sigma\) is the standard deviation of the laser sheet profile modelled as a gaussian. However in this study the dependence in \(z\) will be very weak as we will consider the laser sheet as an almost perfect top hat with \(\sigma_z = 0.05\). The particles' diameters are supposed to be small enough (a few microns) to neglect the size of their geometric image \(M d_p\). The particles' physical diameters are randomly drawn in \([\text{mind}_p, \text{maxd}_p]\) according to a gaussian law with mean \(\text{md}_p\) and standard deviation \(\text{stdd}_p\). Unless otherwise specified: \(I_0 = 300, \text{mind}_p = 0.05, \text{maxd}_p = 2.5, \text{md}_p = 1.5\). The distribution is controlled with \(\text{stdd}_p, \text{stdd}_p = 0.15\) yielding medium low diameter scattering and \(\text{stdd}_p = 0.5\) high scattering. All the presented simulations use \(\text{stdd}_p = 0.15\).

The images are synthesised according to (1) with a PSF given by (2). Note that Mie's scattering is not taken into account in this study. Unless otherwise specified, we take \(\sigma_{\text{ref}} = 0.6\). With this value, a particle has a 4x4 pixel image pattern. In the presented simulations, \(\sigma_{\text{ref}}\) does not vary with the volume depth, ie we consider uniform defocusing blur. Unless otherwise specified, a gaussian noise with mean=5 and standard deviation 2 is added to the images. Its amplitude is thus about 10% relative to the maximum particle intensity.

3.2.2. Performance measures: Precision, Recall vs. Quality factor

The difference between the particle approach considered here and the classical tomographic approach initially introduced by [1] raises the question of which ground truth to consider to build performance criterion for the algorithms. Indeed, for a same set of physical particles, the former will aim at providing the list of voxels containing a particle together with its corresponding intensity, while the latter will reconstruct volumic blobs, of approximately the back-projected particle image size, centered around these physical particles. In this tomographic framework in particular, [1] builds the ground truth by expanding locally the physical particles to a 3D gaussian blob, usually of the order of 2-3 voxels side, in the idea of having a volumic distributions well adapted to the subsequent correlation step yielding the 3D displacement field. A natural quality measure is then the Q criterion, which indicates the degree of correlation between the reconstruction and this ground truth.

In our case, the Q criterion cannot be applied directly, as our reconstructions represent particles by one single voxel, and not by blobs. Therefore, to compute this criterion in the sequel, we will expand the reconstruction yielded by LocM-XX methods (see section 4) using the same method as for building the ground truth, i.e. expand each voxel retained using the same 3D gaussian kernel. Here, it is important to emphasize that even with this, our reconstruction can only yield \(Q = 1\) in the fictitious case where the particles are supposed to be located exactly at voxels’ centres. This is linked to the fact that in this reconstruction step, we do not aim at subvoxel precision (a question which we will consider together with the next step of 3D displacement estimation), and thus have no other choice for now, in order to expand particles to
blobs, than to consider that they are located at the voxel’s centres. This point will be discussed again in section 4.1, regarding the construction of the weight matrix used in the reconstruction.

In order to have a more complete and adapted performance diagnostic of our methods, we will thus also introduce two quantities which are well known measures in pattern recognition and information retrieval, Precision and Recall. These quantities are well suited to detection problems such as ours. A “detection” (ie here, a local maximum voxel of the reconstruction) is a True Positive (TP) if it is in the neighbourhood of a true particle. Unless otherwise specified, the neighbourhood is here a 3x3x3 voxels cube centred on the voxel of the true particle. A detection is a False Positive (FP), ie. a ghost, if it is not in the neighbourhood of a true particle. A particle is recorded as False Negative (FN) if there is no detection in its neighbourhood. Precision then gives the fraction of true particles among all detected particles, and Recall is defined as the number of true positive divided by the total number of true particles, ie:

\[
\text{Precision} = \frac{\#TP}{\#TP+\#FP}, \quad \text{Recall} = \frac{\#TP}{\#TP+\#FN}, \quad \text{where \# stands for “number of”}.
\]

The best achievable performance is given by Recall=1 (#FN=0, every particle is detected) and Precision=1 (#FP=0, all the detected particles are true). In practice both measures depend on a common parameter (eg. ppp count) and are correlated (ie. Precision decreases as Recall increases), so it may be convenient to plot Recall as a function of Precision. An example of Recall/Precision curve is given in the next section to discuss the LocM property.

Note that, whatever the quality criterion considered, and in all the simulations below, the ground truth will consist of the particles that are seen by all the cameras exclusively, which is consistent with the fact that all reconstructions are initialized with MLOS.

3.3. Results: assessment of the LocM property

Figure 2 illustrates the performances of local maxima detection on the MLOS reconstructed volume for varying ppp. To perform the MLOS computation, a threshold equal to 8 is applied to the images in order to remove background noise, (mean noise =5 ; max particle intensity =110 in the images) and the image intensity is sampled using bilinear interpolation. The local maxima in this volume can then be easily detected by performing a mathematical morphology dilatation.

Two Recall/Precision curves are plotted in Figure 2, corresponding to different detection criteria. For the blue curves, a detection is considered true if it lies within a 3x3x3 voxels neighbourhood of a particle, whereas for the green curves, a detection is true if an actual particle is inside that voxel. The blue Recall is very close to 1, even for ppp=0.1. It means that (almost) all the particles are in a voxel corresponding to a local maximum or in an adjacent one. Looking at the green curves, one can see that for ppp=0.1, 60% of true particles are located exactly in the local maxima voxels. The very low Precision, close to zero, shows there are a lot of FP (ie. ghost) and thus the fraction of detected true particle is very low.

Figure 2: performances of MLOS local maxima detection for varying ppp. Four 512x512 images. voxel to pixel ratio (v/p) equal 1

The results figure 2 are presented for a voxel to pixel ratio equal 1 (v/p=1), but it is noticeable that the results are almost the same for a voxel to pixel ratio equal ½ with the same neighbourhood defined in voxel (thus smaller in metric size).

These results justify that the LocM strategy is an efficient way to restrict the number of candidate voxels to consider into the iterative algorithms which will refine the reconstruction. Their task will be in particular to reduce the number of ghost particles, which still remain in a high proportion in the LocM detection.

4. Reconstruction algorithms with PSF model: LocM-XX methods

Tomographic reconstruction solves iteratively large linear system \( Y=W.E \), which relates the observation \( Y \) (pixels) with the intensity of the discretised reconstruction space \( E \) (voxels), through the weight matrix \( W \).

In this section, we first describe how we build the matrix \( W \) corresponding to the present particle approach, taking into account the PSF model and the pointwise nature of the physical particles. We then present two alternatives to solve the
linear system, which both take as an input only the local maxima of the MLOS volume, and use the PSF matrix W. Due to this specificity, we term the global processes obtained (ie starting from the preliminary MLOS step) as “LocM-XX”, with XX the system inversion method.

4.1. Weight matrix construction

In the present PSF model, matrix W has to account for the image formation of pointwise particles. Stated as such, this goal would require volume discretizations with a vanishing voxel size, in order to account at best for this pointwise character. Such a scenario is of course unrealistic, and considering voxels of finite size (e.g., as typically done in tomographic PIV, a voxel to pixel ratio of 1 or ½) thus requires an approximation. This approximation comes quite naturally by considering, for the construction of W, that the position $X_p$ of each particle to be detected coincides with a voxel's centre. Upon introducing the voxel grid spacing $\Delta$, this writes:

$X_p = k \Delta \quad \text{with} \quad k \in \mathbb{Z}^3$,

and the images obtained by projecting these particles have the intensity field

$I(x) = \sum_k E(k) h(x - F(k \Delta)) = \sum_k W(x, k) E(k) \quad \text{with} \quad W(x, k) = h(x - F(k \Delta))$

As pointed out in [11], given the present approximation, h should be named here «voxel spread function». In practice we deal with images that are both limited in extent and sampled at discrete points (pixels), ie:

$I(n) = \sum_k W(n, k) E(k), \quad n \in \mathbb{Z}^2 \quad (3)$

With suitable ordering of image space and voxel space, relation (3) can then be written as a linear system: $Y = WE$ where:

$Y$ is the N-dimensional vector of pixels intensity

$E$ is the M-dimensional vector of voxels intensity

$W$ is the NxM weight matrix whose entries $W_{ij}$ gives the weight of the contribution of voxel “j” to pixel “i”. It is directly given by the PSF: $W_{ij} = h(n_i - F(k_j \Delta))$ where $n_i$ denotes the coordinates of pixel “i” and $k_j$ the coordinates of voxel “j”. As in traditional tomographic PIV:

- $\{ i / W_{ij} \neq 0 \}$ is the set of pixels to which voxel j contributes
- $\{ j / W_{ij} \neq 0 \}$ is the set of voxels that contribute to pixel i

It is important to notice that with this model, contradictory to matrices used in conventional tomographic PIV, the image of a particle within the jth voxel only involves the jth column of W, and hence, each voxel is associated with a unique column of W. Thus, as we only retain the local maxima voxels from the MLOS reconstruction (LocM approach, see section 3), the dimension of matrix W is dramatically reduced compared to the full matrix, and also to the matrix retaining only the nonzero voxels found by MLOS (ie, without performing LocM). Quantitative data for this will be presented in section 5.1

The number of elements of the set $\{ i / W_{ij} \neq 0 \}$, ie. the support of h, is adapted according to the value of $\sigma_{psf}$. We use:

- 4x4 window if $\sigma_{psf} \leq 0.65$
- 6x6 window if $0.65 < \sigma_{psf} \leq 1$
- 8x8 window if $1 < \sigma_{psf} \leq 1.5$
- 10x10 window if $1.5 < \sigma_{psf}$

Figure 3 illustrates the principle of this PSF model.

As mentioned above, W is built assuming an arbitrary position for the particles inside the voxels, namely their centre. This is a natural choice, but of course is not verified in practice, with finite size voxels. Hence, a physical particle will not project exactly as the voxel centre does, except if it is indeed located at a voxel centre. As shown by our simulations, this approximation is a factor of performance drop compared to a fictitious “ideal” case where the particles
would be located at the voxels' centres. The latter situation constitutes in fact the optimal achievable performance case. In practice, despite this slight performance drop, the model will appear rather robust to this discrepancy, and we will see that, logically, refining the voxel grid is a solution to further improve its robustness (see § 5.3).

4.2. System inversion

To invert system $Y = W E$ with the specific PSF matrix introduced above, generic methods as well as more specific methods adapted to the sparse nature of the problem can be used. We describe the two choices retained in this article below.

4.2.1. SMART

The Simultaneous Multiplicative Algebraic Reconstruction Technique (SMART) solves linear systems under non-negative constraints. It is a popular choice for tomographic PIV reconstruction (see, e.g. [2]), and thus will also be considered here. Convergence properties of this algorithm can be found in [13]. The update equation is:

$$\log E^{k+1} = \log E^k + \mu \tilde{W}'(\log Y - \log W.E^k)$$

where $\tilde{W}$ is normalised over the columns and $\mu$ is a relaxation parameter. In our simulations, as traditionally done, $\mu$ is set to 1, and we use 20 iterations.

The difference between our LOCM-SMART reconstruction and that used for instance by [2] is thus simply the nature of matrix $W$, which accounts for the PSF and retain only columns associated with local maxima voxels obtain with MLOS (LocM approach), and the initial value of $E$, which consists of the value of these local maxima only.

4.2.2. Compressive Sampling Matching Pursuit (CoSaMP)

It is well known that in the majority of 3D PIV experiments, the number of particles compared to the number of voxels is very low. In our simulations, with a laser sheet 20 mm thick, an image density of 0.05 ppp corresponds to a density per voxel (ppv) equal to 2e-4. Consequently, it seems natural to seek for algorithms that account for sparsity. Pursuit algorithms are specifically suited to this, as they are algorithms designed to solve the constraint least-squares problem:

$$\min_{E} \|W.E - Y\|_2 \text{ subject to } \|E\|_0 \leq S$$

where $||E||_0$ is the number of nonzero entries of $E$, and $S$ is an integer. Transposed to tomographic PIV, this amounts to find the location and intensities of $S$ nonzero voxels in the volume, solving the system in the least-squares sense. $S$ is a parameter set by the user.

CoSaMP [7] is a recent improvement of basic Matching Pursuit (MP) [8] and Orthogonal Matching Pursuit (OMP) [9]. To the best of our knowledge, these algorithms root in CLEAN, initially introduced in astrophysics [6]. As MP and OMP, CoSaMP is an iterative algorithm. It selects a fixed number $S$ of voxels and computes their intensities. But unlike MP and OMP that add voxels one by one with no possibility of removing, CoSAMP can put into question the already selected locations, adding or removing many locations from the current considered set at each iteration.

Roughly, at each step $k$, the $S$ non zeros entries of the current solution $E^{(k-1)}$, and the $\alpha S$ locations of highest back-projected residual ($W'(Y - W.E^{(k-1)})$) are used to compute a least squares solution with $(\alpha + 1)S$ locations (default value for $\alpha$ is 2). The solution for step $k$ is then the restriction to the $S$ highest entries of the least-squares solution. Proof of convergence can be found in [7]. Hereafter we give the basic CoSaMP algorithm, but some variants are possible, see [7].

Notation: Let $T$ be a subset of $\{1...N\}$ and let $X \in \mathbb{R}^N$. We define the restriction of $X$ to the set $T$ as:

$$X|_T = [X(i) \text{ if } i \text{ in } T, 0 \text{ otherwise}.$$  

Let $X|_S \in \mathbb{R}^S$ be the restriction of $X$ to its $S$ highest components (the $N-S$ others are set to 0).

Let $\text{Supp}(X) = \{ i \in \{1...N\} / X(i) \neq 0 \}$

Finally, using a Matlab-like notation we note $W(:,T)$ the submatrix of $W$ whose columns are listed in $T$.

Pseudo Code
Goal: solve $Y = W \cdot E$ subject to $\|E\|_0 \leq S$

$E^{(0)} = 0$ ; $r = Y$ ; $k = 0$  % intialisation of the solution and residual

Repeat {
  $k = k + 1$
  $X = [W^T \cdot r]$
  $\Omega = \text{Supp} |x|_{[2S]}$
  $T = \Omega \cup \text{Supp} |E|^{(k-1)}$  % support of the 2S highest components
  $E = (W(:, T)^T \cdot W(:, T))^{-1} \cdot W(:, T)^T \cdot Y$
  $E^{[k]} = E|_S$  % least square
  $r = Y - W \cdot E^{[k]}$  % restriction to the S highest
} Until halting criterion

Notice that, by nature, LocM-CoSaMP only produces $S$ voxels with non zeros values. Varying $S$ thus yields different operating points, that may influence the quality of the results. The role of this parameter will be investigated in section 5.2. In our simulations, we use 15 LocM-CoSaMP iterations for the reconstruction.

4.3. Synthesis

Figure 4 summarizes the flow charts of the new methods proposed here. The LocM prefix indicates that matrix $W$ only retains the columns corresponding to the local maxima voxels from MLOS. Their common point is to first build a guess value using MLOS followed by a detection of the local intensity maxima in this volume (LocM). Only these maxima are retained for the construction of the PSF based weight matrix $W$, leading to a dramatic dimension reduction (see also section 5.1 for quantitative results). The inversion of the system that follows, either with SMART or CoSaMP, is then significantly accelerated.

![Figure 4: Flow chart of LocM-XX methods. Inversion of system $Y = W \cdot E$ is here done with either SMART or CoSaMP, but could also be performed with other algorithms.](image)

After reconstruction is performed, we compute the performance measures defined in section 3.2.2. The quality factor $Q$ is computed using the entire results for both methods, except for section 5.2. Concerning Precision and Recall, It should be noticed that unlike LocM-CoSaMP, LocM-Smart doesn't produce, strictly speaking, a sparse solution (in fact its a mild sparsity). This algorithm doesn't select a fixed number of nonzero entries: the output dimension is the same as the input one, with a lot of entries whose values are close to zero. To compute consistent Precision and Recall measures with $S$ locations, $S$ being the value tuned by the user for LocM-CoSaMP, we keep the $S$ highest entries.

5. Results

All the tests presented in this article have been performed using a Matlab implementation of the algorithms. A GPU implementation is under way and should reduce drastically the processing times.
5.1. Dimensionality reduction obtained thanks to LocM

We consider the setup described in section 3.2.1, in the case of 1024x1024 images, leading to a 1253x1181x201 voxels reconstructed volume (voxel to pixel ratio equal to 1). As in the tests of section 3.3, MLOS is implemented with a bilinear interpolation, and the images are thresholded with a threshold equal to 8 to remove background noise (mean noise = 5; the brightest particles have a gray level of 110). Table 1 gives, for varying ppp, the percentage of non zero voxels in the MLOS volume, the percentage of voxels whose intensity is higher than 8, and the percentage of local maxima with intensity higher than 8 (we use a 3x3x3 neighbourhood for this search for local maxima). It can be seen that the dimensionality reduction with local maxima is indeed dramatic and their growth is much slower with respect to ppp.

<table>
<thead>
<tr>
<th>ppp</th>
<th>0.0280</th>
<th>0.0429</th>
<th>0.0575</th>
<th>0.0709</th>
<th>0.0864</th>
<th>0.1001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nonzero voxels</td>
<td>10 %</td>
<td>17 %</td>
<td>24.4 %</td>
<td>31.9 %</td>
<td>38.8 %</td>
<td>45 %</td>
</tr>
<tr>
<td>Voxel ≥8</td>
<td>2.7 %</td>
<td>7 %</td>
<td>12.8 %</td>
<td>19.7 %</td>
<td>27.2 %</td>
<td>34.4 %</td>
</tr>
<tr>
<td>Local maxima ≥8</td>
<td>0.39 %</td>
<td>0.6 %</td>
<td>0.73 %</td>
<td>0.78 %</td>
<td>0.81 %</td>
<td>0.81 %</td>
</tr>
</tbody>
</table>

Table 1: percentage of nonzero voxels, voxels over 8 and local maxima of the MLOS volume for varying ppp.

5.2. Operating point

Unlike SMART, CoSaMP requires that the user provides a maximum number S of nonzeros elements, called “sparsity”. Varying this parameter gives different operating points of the algorithm. We will call “normalised sparsity” the sparsity S divided by the true number of particles. It can easily be shown that when normalised sparsity equals 1, Precision equals Recall. As shown in figure 5, this operating point yields the optimal quality factor Q for LocM-CoSaMP, whereas LocM-SMART reaches its optimal Q when normalised sparsity is about 2. In the following, as a choice has to be done for this parameter, we will consider a value of 1, which is a natural choice in the context of synthetic tests. Indeed, one then obtains the fraction of detected true particles when the algorithms are tuned to retrieve the exact number of particles. For LocM-SMART, this will be done by retaining the S brightest voxels among the whole reconstruction. In the following, as for this tuning Precision=Recall, only the Recall will be plotted. Figure 5 shows that for the present setup, and for normalised sparsity equal to 1, performances of LocM-CoSaMP are 12% higher than that of LocM-SMART. Notice that in the sequel, as explained section 4.3 The LocM-SMART quality factor is computed using the whole result.

Figure 5: Recall, Precision and Q as a function of normalised sparsity for LocM-CoSaMP (blue) and LocM-SMART (red), 1024x1024 images, ppp=0.058. The voxel to pixel ratio (v/p) is equal to 1, and the reconstructed volume is 1253x1181x201 voxels. Reconstruction is performed with the same psf as for the image generation.

5.3. Subvoxel location error and voxel to pixel ratio

In air flows, the particles’ physical diameters are typically a few microns at most. Common experimental setups with voxel to pixel ratio equal to 1 (considering for instance a CCD size of 1024 pixels and 100 mm as the typical side of the reconstructed volume) lead to a voxel size of the order of 100 µm, which leaves much room inside each voxel for the particles. These dimensions illustrate the extent of the approximation which is done in the present PSF approach, where matrix W is built assuming that the particles are located at the voxels’ centres (see section 6), leading to a discrepancy in the image projections. In this section, we investigate the practical consequences of this approximation.

The dotted curves in figure 6 give the performances, in terms of Recall and Q criterion, for the fictitious ideal case when the particles are located at the voxels’ centres. In this case, the model is exact and the dotted curves correspond to an upper bound (best possible performance) for the algorithms in the considered configuration.
The other two curves correspond to the realistic situation where the (same) particles are randomly distributed within the voxels. This naturally results in a performance drop compared to the bound for LocM-CoSaMP (left) and LocM-SMART (right), for varying ppp. The solid curves corresponds to a voxel to pixel ratio (v/p) equal to 1, and the dashed curves to ½. As the same physical volume is reconstructed, the latter configuration has 8 times more voxels than the former, but still remains practically tractable with Matlab thanks to the LocM dimensionality reduction (see section 11 for typical processing times). One may see that choosing v/p=1/2 increases significantly the performances, both in terms of detection and of result quality. Interestingly, further refinement with smaller v/p does not significantly improve the results; while being very computationally intensive.

**Figure 6:** (left) LocM-CoSaMP ; (right) LocM-SMART: Recall and Q for the ideal centered case (see the text for details), and the realistic case with voxel to pixel ratios equal to 1 and ½. 512x512 images. The reconstructed volume is 627x591x201 voxels for v/p=1 and 1254x1182x401 for v/p=1/2. Reconstruction with the same $\sigma_{psf}$ as for image synthesis.

5.4. Sensitivity of the reconstruction to the value used for $\sigma_{psf}$

The PSF model requires knowledge of the camera sensor PSF, $\sigma_{psf}$ to build the weight matrix $W$. In real experiments, this can be done using a calibration pattern, but with limited accuracy. An interesting question is thus how LocM-CoSaMP and LocM-SMART behave with respect to an uncertainty on this parameter. For the present simulation, four 512x512 images are generated with $\sigma_{psf}=0.6$. The reconstruction is performed with various values of $\sigma_{psf}$ ranging from 0.5 to 0.9. Figure 7 presents the performances in terms of Recall and Q for LocM-CoSaMP and LocM-SMART. One may see that LocM-CoSaMP is not very sensitive to this parameter and reaches its best when $\sigma_{psf}$ is slightly over-estimated. On the contrary, LocM-SMART tolerates an under-estimation of $\sigma_{psf}$ but its performances collapse when it is over-estimated, falling well below the values obtained with the classical TomoPIV SMART of [2] (see the next section for its implementation).

**Figure 7:** Reconstruction performances using varying $\sigma_{psf}$ for images generated with $\sigma_{psf}=0.6$ (v/p=1 and v/p=0.5). Left hand side: Recall. Right hand side: Quality factor Q. Performances of classical TomoPIV SMART (see section 5.5) are also given.
This low sensitivity of LocM-CoSaMP to $\sigma_{psf}$ is thus very interesting for processing real images. Preliminary results on experimental data seems to confirm this low sensitivity.

5.5. **Comparison with Classical TomoPIV SMART**

The classical TomoPIV SMART as described in [2] (hereafter simply referred to as SMART, without prefix), with $v/p=1$, computes the entries of W using the intersection of a sphere and a cylinder, and uses all the nonzero voxels of MLOS (or, more precisely, all those greater than a low threshold). It leads to a “blob-like” reconstruction because of its very limited “equivalent PSF” extent. In the present Matlab implementation, fairly soon as the images' size and/or ppp increase, SMART matrix cannot be built for memory limitation reasons, contrary to LocM-XX methods. The operator version with on-the-fly computation has then to be used at the expense of a huge increase in processing time.

To compute Recall and Precision for this classical SMART, we perform a local maxima detection on the result (LocM) and retain the desired number brightest voxels.

The following results are for 512x512 images. With a voxel to pixel ratio equal to one, the reconstructed volume is 627x591x201 voxels, and 1254x1182x401 voxels for $v/p=1/2$. The reconstruction is performed with the same $\sigma_{psf}$ as for the image generation.

![Figure 8](image-url): Comparison of the performances of LocM-CoSaMP and LocM-SMART with classical SMART. (left) Recall, (right) quality factor Q.

One can see in figure 8 that the LocM-XX methods with a voxel to pixel ratio equal to $\frac{1}{2}$ clearly outperform classical SMART. In spite of this requirement, the processing time does not rise much compared to $v/p = 1$, as can be seen in Table 2 below. This table gives the Matlab processing time in seconds on a modern 8 cores workstation. The dramatic surge for classical TomoPiv SMART at $ppp=0.071$ corresponds to the transition towards the on-the-fly operator version, when storing the whole matrix is no longer possible (more than 10 millions voxels here). One may see that in comparison, LocM-CoSaMP takes only 5 minutes to process 1254x1182x401 voxels volume ($v/p=1/2$) with a W matrix occupying 2 gigabyte compared to the 10 gigabytes of the original MLOS-W matrix. Overall, this algorithm is the fastest in all configurations.

<table>
<thead>
<tr>
<th>Ppp</th>
<th>0.028</th>
<th>0.042</th>
<th>0.057</th>
<th>0.071</th>
<th>0.085</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>v/p=1 MLOS+SMART</td>
<td>75 s</td>
<td>128 s</td>
<td>206 s</td>
<td>10257 s</td>
<td>13858 s</td>
<td>17584 s</td>
</tr>
<tr>
<td>v/p=1 MLOS+LocM-CoSaMP</td>
<td>45 s</td>
<td>45 s</td>
<td>52 s</td>
<td>51 s</td>
<td>47 s</td>
<td>48 s</td>
</tr>
<tr>
<td>v/p=0.5 MLOS+LocM-CoSaMP</td>
<td>297 s</td>
<td>315 s</td>
<td>335 s</td>
<td>316 s</td>
<td>313 s</td>
<td>300 s</td>
</tr>
<tr>
<td>v/p=1 MLOS+LocM-SMART</td>
<td>52 s</td>
<td>58 s</td>
<td>68 s</td>
<td>68 s</td>
<td>64 s</td>
<td>64 s</td>
</tr>
<tr>
<td>v/p=0.5 MLOS+LocM-SMART</td>
<td>331 s</td>
<td>368 s</td>
<td>414 s</td>
<td>399 s</td>
<td>400s</td>
<td>382 s</td>
</tr>
</tbody>
</table>

Table 2: Matlab processing time in seconds as a function of ppp. 512X512 images. 627x591x201 voxels for v/p=1 and 1254x1182x401 voxels for v/p=1/2.
5.6. Particle visibility issues

As studied in more details in our companion paper [7], the particles that are not seen by all cameras represent a nuisance factor for the reconstruction. Assuming a random uniform particle distribution in space, the proportion of particles that are seen by all the cameras is equal to the ratio of the volume seen by all the cameras to the volume seen by at least one camera:

\[ \frac{\text{intersection volume}}{\text{union volume}} = R_{IU} \].

For given cameras' positions, this ratio depends on the field of view, and thus, with a fixed focal length and pixel size, it directly depends on the image size (the bigger the image size, the greater the ratio).

To study the influence of \( R_{IU} \), one considers the same particle volumic distributions with an increasing image size. The ppp is thus practically the same for all image sizes. It can be seen in figure 9 that whatever the algorithm, Q and Recall are strongly correlated to \( R_{IU} \). In other words, the less there are particles not seen by all the cameras, the better the results.

For this simulation, \( v/p=1/2 \) implements a coarse to fine approach. MLOS local maxima are first detected for \( v/p=1 \) and a sparse volume around local maxima is computed for \( v/p=1/2 \). It results in a very slight performance drop but is far less memory consuming, especially when the image size is 1024x1024.

![Figure 9](image.png)

**Figure 9**: Recall (left) and Q (right) as a function of images size for a fixed ppp = 0.058. \( R_{IU} \) is the ratio between the intersection and union volumes, see the text for more details. Reconstructions are performed with the same \( \sigma_{psf} \) as for image generation.

It can be seen in Figure 9 that LocM-CoSaMP is less robust than LocM-SMART when the proportion of partially seen particles is high (low \( R_{IU} \)). As soon as the \( R_{IU} \) ratio reaches 0.4, LocM-CoSaMP outperforms the other methods. Real experiments on a cylindrical air jet currently investigated at ONERA correspond to a ratio close to 0.6, which is a rather favourable situation for LocM-CoSaMP.

Classical SMART performs better than LocM-SMART for \( v/p=1 \) as soon as \( R_{IU} \) is greater than 0.4, however at a much higher computational cost, but performs lower than LocM-SMART for \( v/p=1/2 \). It is worth noting that LocM-CoSaMP performs better than classical SMART whatever the \( v/p \) ratio.

6. Conclusion

In this paper, we have introduced algorithms which fully exploit the specificities of performing a pointwise particle reconstruction in the context of 3D PIV, instead of the traditional blob-like reconstruction. The particle approach leads indeed to very sparse and spiky reconstructions, where only the voxels supposed to contain a particle have nonzero values. We have shown that true particles are located on, or close to, local maxima of the MLOS result. Only columns of the weight matrix \( W \) corresponding to these local maxima of MLOS need to be retained. This dramatically reduces the dimensionality of the linear system to solve. We have proposed LocM-XX methods that aim at exploiting the PSF model capability to render the image patterns of a particle using only one voxel. We have introduced the Compressive Sampling Matching Pursuit algorithm and shown that LocM-CoSaMP clearly outperforms classical SMART if a voxel to pixel ratio equal \( \frac{1}{2} \) is adopted. In addition, LocM-CoSaMP is computationally very efficient, as it processes a 1254x1182x401 voxels volume with Matlab in about 5 minutes on a modern 8 core workstation. A great advantage of LocM-CoSaMP over LocM-SMART is also its lesser sensitivity to the uncertainty on PSF parameters used for reconstruction.

At the time of this writing, LocM-CoSaMP has been applied to real datasets obtained on the near-field of a cylindrical
air jet. Physically consistent preliminary vector fields have been obtained with a 3D motion estimation using volumes reconstructed with LocM-CoSaMP. Our ongoing work now concerns more particularly such vector field computations using sparse and spiky reconstructions.

REFERENCES