

PAPER • OPEN ACCESS

Adaptive incremental stippling for sample distribution in spatially adaptive PIV image analysis

To cite this article: M Edwards and R Theunissen 2019 *Meas. Sci. Technol.* **30** 065301

View the [article online](#) for updates and enhancements.

You may also like

- [Automated scripting of the dosimetric evaluation of adaptive versus non-adaptive radiotherapy](#)
Pratik Samant, Ben George, Tom Whyntie et al.
- [Adaptive input shaping and control for simultaneous precision positioning and vibration suppression of smart composite plates](#)
Kougen Ma and Mehrdad N Ghasemi-Nejhad
- [On the interplay between robustness and dynamic planning for adaptive radiation therapy](#)
Michelle Böck, Kjell Eriksson and Anders Forsgren

Adaptive incremental stippling for sample distribution in spatially adaptive PIV image analysis

M Edwards^{id} and R Theunissen^{id}

Queens Building, University of Bristol, BS8 1TR, United Kingdom

E-mail: m.edwards@bristol.ac.uk

Received 20 November 2018, revised 13 February 2019

Accepted for publication 18 March 2019

Published 10 May 2019



Abstract

Adaptive sampling strategies in PIV have been shown to efficiently combine the need for limited user-dependence with increased performances in terms of spatial resolution and computational effort, thus rendering such approaches of great interest. The allocation of correlation windows across the spatial image domain is dependent on the interpretation of an underlying objective function, and the distribution of windows accordingly. It is important that such allocation is computationally efficient, robust to changing objective functions and conditions, and conducive to high quality sampling. In this paper, an alternative sample distribution method, based on adaptive incremental stippling, is presented and shown to combine the speed of PDF-based methods with the quality of ‘ideal’ spring-force methods. Case-dependent parameter tuning is no longer necessary, thus improving robustness. In addition, an algorithm to adaptively size initial correlation windows is proposed to further minimise user dependence.

Keywords: adaptive incremental stippling, spring force, adaptive sampling

(Some figures may appear in colour only in the online journal)

1. Introduction

Conventional, structured, PIV analysis routines are typically initiated by dividing the image recordings into smaller correlation windows. Starting from an initial size, WS_0 , and mutual overlap ratio, WOR , the correlation areas are subsequently reduced to a final window size, WS_f , over a number of iterative passes [1]. All parameters are user-defined, applied uniformly across the image recording, and define a structured interrogation grid with a resulting uniform vector spacing $h = (1 - WOR) \cdot WS_k$, where WS_k denotes the window size for iteration k . Not only does this approach allow for simple interpolation onto pixel-wise predictors, for image deformation [2] or post-processing purposes, but it is simple to

implement, enabling such an approach to become commonplace within the community.

Although Cartesian grids offer good interpolation properties, the sampling density is unable to be adjusted according to in-homogeneous flow and/or image conditions. Small and large flow scales must consequently be sampled with equal frequency, leading to local under-sampling of the small scales and/or oversampling of the larger scales. The straightforward calculation of derivative information is not, in itself, a sufficient argument to limit image processing schemes to structured interrogation grids, given the availability of interpolation schemes which are able to semi-analytically calculate derivative information from unstructured data [3]. Moreover, the performance of such routines are heavily dependent on user experience through the selection of WS_0 , WS_f , and WOR which, even under expert use, can significantly affect the resulting solution [4, 5].

The aforementioned limitations can be addressed by adaptive sampling (AS) methodologies, wherein interrogation



Original content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](https://creativecommons.org/licenses/by/3.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

parameters and sampling densities (i.e. the positioning and size of correlation windows) are adjusted spatially according to the image and flow conditions. Such routines have been shown in literature to reduce the error when analysing PIV images [6–10] or in point-wise measurement applications [11]. Spatially adaptive algorithms adjust the sampling density according to some objective function, which serves to identify regions of available information in the domain by balancing flow interest with image quality, as well as considering previous sampling efforts and the need to interrogate the entire domain. Optimal construction of the objective function is still under debate; Theunissen *et al* [7] originally used the local spatial standard deviation of the displacement field, whereas Yu *et al* [10] suggest a combination of the curl and gradient magnitude. Vorticity, and various vortex detection criteria, such as the Q-criterion, are also frequently used in literature to identify coherent structures. In addition, it is known that more samples should be attributed to regions of higher flow curvature in order to minimise reconstruction error [11]. This paper does not address such debate, since, irrespective of its origins, it is essential that an efficient and robust distribution method is applied, capable of faithfully reflecting the intentions of the governing objective function while preserving distribution quality.

Theunissen *et al* [7] originally allocated sampling locations by converting the objective function into a probability density function (PDF) which could be sampled according to the inverse of the associated cumulative density function. This approach required additional steps to yield distributions of satisfactory quality, hence Yu *et al* [10] considered the so-called spring force distribution (SFD). While the former offers flexibility and the lowest computational effort, sampling distributions with SFD are of higher quality at the expense of robustness and increased computational effort. In view of maintaining robustness while safeguarding overall processing time, the authors present a new distribution method for adaptive sampling techniques in this paper.

The problem regarding suitable sampling allocation is highlighted in section 2. To allow fair comparison of the distribution methods, and to improve the generality of the SFD method proposed by Yu *et al*, a modification is presented in section 3 together with the recommended method of adaptive incremental stippling. The different distribution schemes are juxtaposed in section 4 with the focus on robustness, accuracy and efficiency. The proclaimed benefits of stippling are shown on the basis of numerical simulation, analysis of synthetic images, and an experimental case of the flow over a backwards facing step.

2. Problem statement

Proper signal reconstruction on the basis of discrete samples, as is the case in PIV, necessitates an adequate allocation of such samples within the spatial domain. This is demonstrated in figure 1 for the case of a sinusoidal signal with unit amplitude, sampled at 60 locations and re-interpolated using

a cubic B-spline kernel. In the first instance (figure 1(a)), samples are equispaced. As per [12], the highest errors are observed in regions of greatest curvature. Small fluctuations in the sampling locations further exacerbate these errors, yielding an increase in global reconstruction error (figure 1(b)). Conversely, adaptively allocating a higher concentration of samples in these regions, using an approach similar to figure 2, reduces the local error (and consequently the global error), while reducing the sensitivity to variations in sampling distribution (figures 1(b) and (d)). Accurate data re-interpolation is of importance when iteratively deforming PIV images to reduce particle image displacement gradients, as inaccuracies in the predicted displacement field can accumulate through the iterations.

While locally varying the concentrations of samples in accordance to signal condition shows to be advantageous, care must be taken to avoid excessive local oversampling (i.e. clustering of samples), particularly if this comes at the expense of poorer domain-wide sampling. Not only does poor distribution quality result in a worse reconstruction, noise in derivative values is also heightened, potentially reducing the efficacy of adaptive strategies [7, 8, 10]. Therefore, sampling locations should be well distributed over the domain according to the desired sampling density, free from excessive local clustering or voids. Furthermore, the distribution method must be computationally efficient, such that it does not hinder adaptive approaches with large overheads, able to handle complex mask geometries typically encountered in PIV experiments [14, 15], and remain robust to a wide range of conditions.

The method adopted by Theunissen *et al* [7] was originally developed by Secord *et al* [16], and is referred to hereafter as the PDF Transform method, or simply the PDF method. Within this approach, sample locations are allocated following a regular sampling of the inverse cumulative density of the objective function, as illustrated for one dimension in figure 2. While this approach is effective in one dimension, extension to multiple dimensions renders the methodology susceptible to significant clustering of samples, as will be shown in the numerical assessment in section 4 (See figures 5(a) and (e)). Laplacian smoothing can be adopted to ameliorate the distribution, yet the results remain unsatisfactory and, as stated by Yu *et al* [17], correspondence between the desired objective function and final sampling distribution is reduced.

In light of these shortcomings, Yu *et al* modified a mesh distribution method, originally developed by Persson and Strang [18], for PIV application wherein an initial distribution of samples is iterated towards their ideal locations by use of a spring-force analogy, and hence is referred to herein as the spring force distribution (SFD) Method. Repulsive forces between nodes of the Delaunay triangulation are computed by considering the current inter-sample separation with the target separation defined by the objective function. Attractive forces representative of too large inter-sample distances are not allowed. The resulting system of forces then perturbs the nodes over a small timestep, dt , following which the locations and forces are recomputed. Choosing a larger dt allows a speedup in convergence at the expense of stability. The

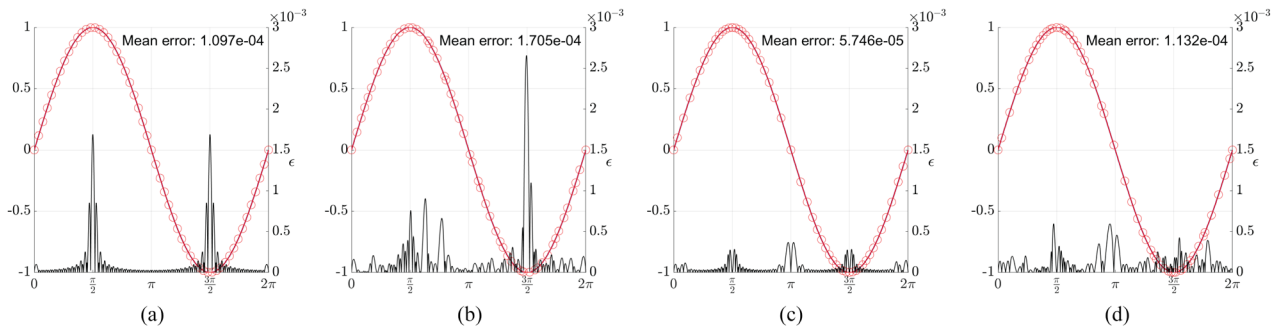


Figure 1. Illustration showing the influence of poor sampling locations, considering a sinusoidal signal of unit amplitude (red) imposing (a) equispaced sampling (red dots) (b) approximately equispaced, with a small random spatial perturbation applied to each sample to mimic poor sampling (c) adaptive sampling as per Secord [13] (d) adaptive sampling with a small spatial jitter applied as in (b). Black lines depict the absolute error between the imposed signal and reconstruction adopting a cubic B-spline. The mean of the error magnitude is presented for each case.

process is repeated until satisfactory convergence is attained¹. The resulting meshes consist of near equilateral triangles² (figure 5) and were shown to reduce the error when used within spatially adaptive routines [10, 17].

Despite the improvement in mesh quality and subsequent reduction of error, the SFD method carries a number of significant challenges, namely user-sensitivity and computational performance. The absence of attractive forces necessitates additional control logic to be imposed at regular iteration intervals, to enforce the addition or removal of samples in regions where the discrepancy between current and target spacing exceeds a predefined threshold. The SFD method is particularly sensitive to the threshold at which points are considered too dense or sparse; too strict and the method becomes unstable, too relaxed and the benefits are negated. Although these thresholds are not quantified by Yu *et al*, the current authors have found target separations greater than 2.2 times the current separation indicative of samples being too close whereas target separations less than 0.6 times the current separation indicate samples are too sparse. Furthermore, the frequency at which the density control is imposed must be tuned; too infrequent and the method becomes slow to converge, too frequent and the method may once again become unstable. In fact since re-calculation of the triangulation is required, increasing the frequency may also slow convergence. Although Yu *et al* do not comment on how they implement adding points or how frequently they test the sampling density, it was found that a density control frequency of eight iterations was suitable to ensure reasonable stability, yielding convergence within 10–20 iterations while restricting computational requirements. Finally, the method lacks guaranteed termination [18] and therefore requires an artificial, user-defined, iteration limit to be imposed. Once again this value has significant influence on the performance of the method and varies depending on the number of samples and the non-uniformity of the target sampling density. To complicate matters further, this iteration

¹ See [18] for a detailed description of the original method along with the source code.

² The ratio between the diameter of the circle inscribed by the triangle and the one circumscribing the triangle, which attains a maximum value of 2, quantifies how equilateral a triangle is and can be used as heuristic for mesh quality.

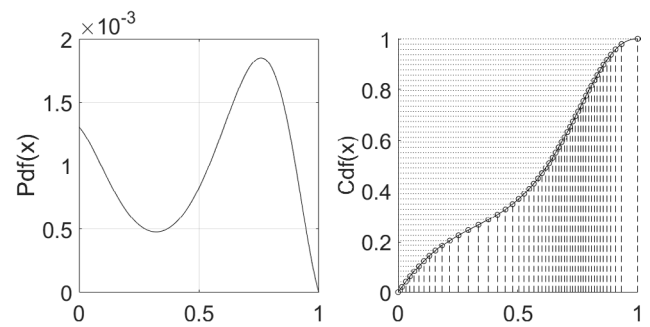


Figure 2. Illustration of the probability density weighting sampling allocation (PDF methodology). The cumulative density (right) is calculated of a given probability density function (left). Regular sampling of the inverse of the cumulative density function then accordingly assigns relevant spatial locations.

limit must also be considered in conjunction with the aforementioned density control frequency, since artificial termination shortly following density control is likely to result in poor quality distributions. This behavioural dependency of the resulting sampling distribution on the number of iterations is illustrated in figure 3 imposing the Franke function, defined in (1) [19], as underlying objective function and a density check every eight iterations.

$$\begin{aligned}
 f(x, y) = & \frac{3}{4} \exp\left(-\frac{(9x-2)^2}{4} - \frac{(9y-2)^2}{4}\right) \\
 & + \frac{3}{4} \exp\left(-\frac{(9x+1)^2}{49} - \frac{9y+1}{10}\right) \\
 & + \frac{1}{2} \exp\left(-\frac{(9x-7)^2}{4} - \frac{(9y-3)^2}{4}\right) \\
 & - \frac{1}{5} \exp\left(-\frac{(9x-4)^2}{4} - \frac{(9y-7)^2}{4}\right). \quad (1)
 \end{aligned}$$

Sub-optimal sampling distributions are obtained when terminating the process too early whereas with increasing iteration number the resulting distribution remains unaltered. The user is therefore left with a choice; either allow many iterations, dramatically increasing execution time, or terminate early at the risk of obtaining unrepresentative sample allocations.

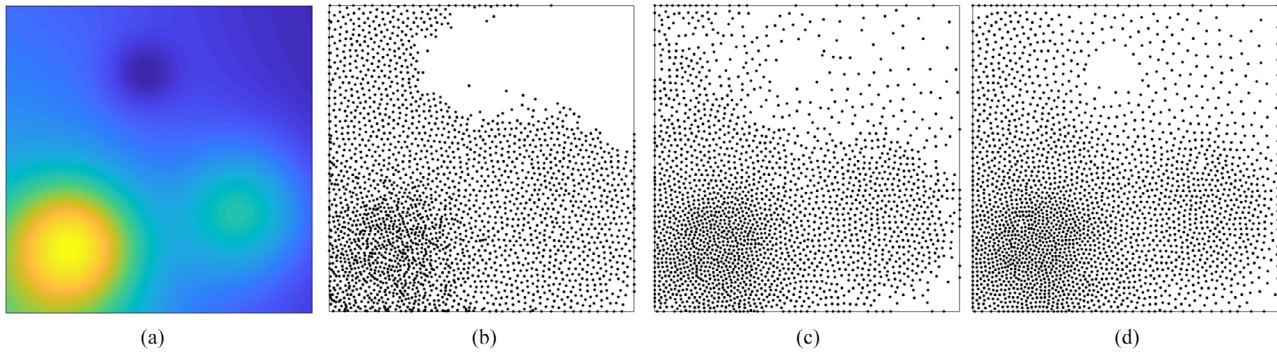


Figure 3. Demonstration of the influence of maximum iteration counter on the resulting distributions obtained through the SFD methodology, imposing the Franke function (a) as objective sampling density with 2500 samples. (b)–(d) Solutions after the 8th, 17th, and 26th iterations, respectively.

Although it is often possible to achieve a good balance of parameters on a case by case basis, it is not possible to obtain optimal parameters for a wide range of applications. User input is therefore required, limiting robustness. Furthermore, given the trend of increasing camera sensor resolution and consequently the total number of samples to be placed, safeguarding computational efficiency becomes pivotal; populating a CCD sensor ranging between 4–29MP³ with correlation windows of 16 px² allowing 50% mutual overlap quickly amounts to 65–450 × 10³ windows respectively.

3. Methodology

3.1. Generalising Yu *et al*'s spring-force distribution method

The SFD method of Yu *et al* requires the target sampling distribution to be defined in terms of the inter-sample spacing, $h(x, y)$, instead of the target sampling density, i.e. smaller values in regions where more samples are desired. In fact, the method requires the relative inter-sample spacing, since the forces are normalised in order to avoid over-constraining the system of forces [18]. Objective functions of the form $\Phi(x, y) = \phi(x, y) \cdot sd(x, y)$, where $sd(x, y)$ is the seeding density over the domain, evaluated using a particle detection routine as in [7], require modification before being suitable for use with the SFD method, since seeding density and, typically, flow metrics correlate positively with the desired sampling density. In this context, $\phi(x, y)$ represents the desired sampling distribution density according to flow metrics alone. For example, this may be the local standard deviation of displacement as in [7], the combination of curl and gradient magnitude as in [10], or some other user desired combination of flow metrics. When coupled with $sd(x, y)$ one obtains the combined objective function $\Phi(x, y)$ indicating the desired sampling density over the domain. Irrespective of the flow metric, the value of this function correlates with the desired sampling density, and thus needs converting into the inter-sample spacing equivalent. Yu *et al* approach this by first obtaining the initial inter-sample spacing, $h_0(x, y)$, according to the seeding density via $h_0(x, y) = (1 - WOR) \cdot \sqrt{NI \times sd(x, y)^{-1}}$ where NI is

the user-defined required number of particles per correlation window. In subsequent iterations, flow adaptivity is incorporated by converting flow metrics, ϕ , into their distance-based counterparts through use of a transformation function, $g(\cdot)$, shown in (2). In this function, large values of $\phi(x, y)$ are converted into small values and vice-versa. The output from $g(\phi)$ is a factor $\phi^*(x, y)$, limited between 0.6 and 1.5, to rescale the existing inter-sample spacing, h_0 , to obtain the new target inter-sample spacing, i.e. $h(x, y) = h_0(x, y) \cdot \phi^*(x, y)$, which now incorporates flow adaptivity.

$$\phi^* = g(\phi) = \begin{cases} -0.5\phi^{0.88} + 1.5, & 0 < \phi < 1 \\ 0.4\phi^{-1.1} + 0.6, & 1 \leq \phi \end{cases} \quad (2)$$

The motivation for limiting refinement due to flow metrics is to limit excessive coarsening or refinement of the mesh. Such limitations may be conducive to producing high quality sampling distributions, which prevent localised clustering and promote a degree of space-filling within the domain. However, the effect is not dissimilar to the effect of retrospective mesh-smoothing, in that the obtained distribution may no longer faithfully reflect the input objective function, as demonstrated by figure 5(g). Furthermore, bounds will have to be selected on a case-by-case basis according to the underlying displacement scales and image quality, requiring significant tuning to obtain an appropriate, smooth, scaling function. Instead, as suggested by Persson [20], robustness and generality should be ensured by controlling the gradient of the inter-sample spacing throughout the domain, instead of imposing absolute limits on the spacing's themselves.

Distribution approaches whereby the samples are positioned in accordance with the objective function, $\Phi(x, y)$, directly, naturally attribute more samples where $\Phi(x, y)$ is greater and therefore do not require modification or transformation of $\Phi(x, y)$. Instead, in its simplest form, the input objective function simply takes the form of the multiplication of the chosen flow metric and seeding density, i.e. $\Phi(x, y) = \phi(x, y) \cdot sd(x, y)$. Due to the limits imposed by (2), significantly different distributions would be obtained despite use of the same flow metric $\phi(x, y)$.

An un-bounded conversion of the objective function, into a suitable form for the SFD, is therefore required to enable a comparable juxtaposition of the distribution methods. This transformation is presented in the appendix. The same

³ See www.dantecdynamics.com/ccd-and-scmos-cameras and www.lavision.de/en/products/cameras/cameras-for-piv/index.php Accessed 15/11/18.

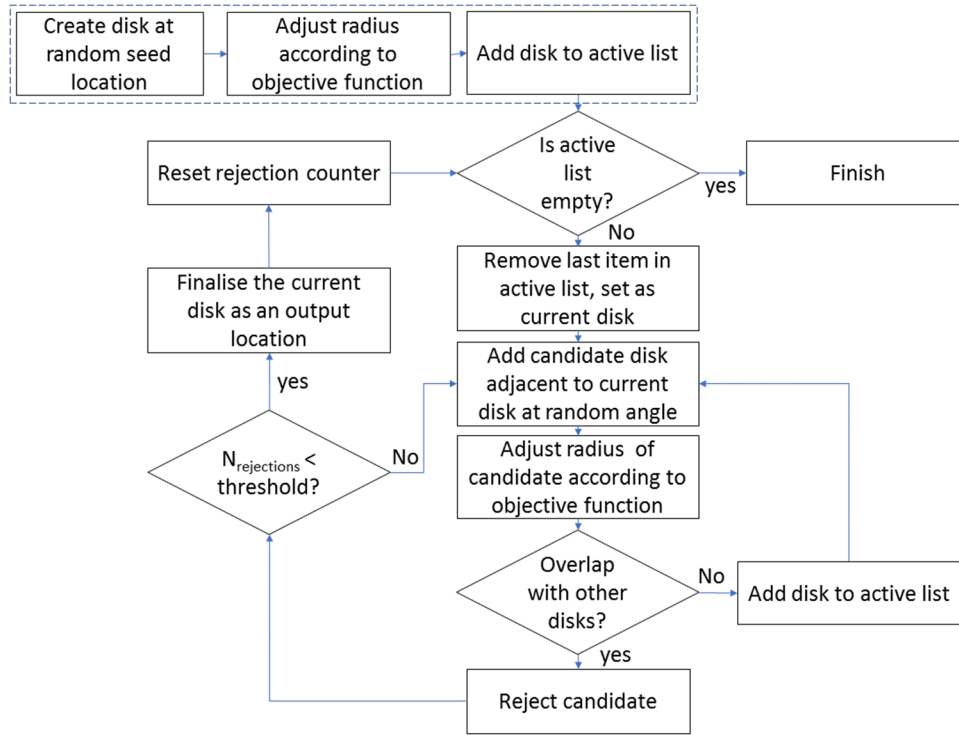


Figure 4. Flow chart related to the AIS process.

premise as used within the proposed stippling method is adopted here, in that each sample location represents a discrete proportion, F , of the underlying objective function, i.e. $N_w \cdot F \approx \iint \Phi(x, y) dx dy$, with $\Phi(x, y)$ as above.

$$h(x, y) = g_2(\Phi(x, y)) = \frac{\sqrt{F/(\Phi(x, y))}}{\pi} \quad \text{with} \quad (3)$$

$$F = N_w^{-1} \iint (\Phi(x, y)) dx dy = N_w^{-1}.$$

Where N_w is the number of windows to be distributed. Now, $\Phi(x, y)$ can be used as input into density based distribution methods, whereas $h(x, y)$ calculated from (3) can be used as the inter-sample spacing as required by the SFD. This conversion is indeed very similar to the approach of Yu *et al* with the absence of the imposed limits on inter-sample spacing.

3.2. Adaptive incremental stippling

Adaptive incremental stippling (AIS) was initially developed by Ascencio-Lopez *et al* [21] as a means to rapidly construct dotted (stippled) images, such that the stipples' spatial density is reflective of the local image intensity. In this method, each stipple is surrounded by a disk of radius h which is not permitted to overlap with any other disk. By varying the radius of the disks according to the local image intensity, the spatial density of the resulting distribution is reflective of the underlying image intensity, while maintaining good distribution qualities free from excessive clustering or voids. A brief description is given below and outlined in figure 4.

To stipple an image $I(x, y)$ of size $L_x \times L_y$ scaled from $[0, 1]$, first we define the image density as $\phi(x, y) = 1 - I(x, y)$.

Next, an initial disk is placed at a random location (x_i, y_i) with radius $h_1 = \sqrt{L_x \cdot L_y / (\iint \phi(x, y) / F)}$, where F is a user-specified constant which represents the amount of density to be conveyed by a single stipple. The radius of the disk is then adjusted until $\iint_{S(h)} \phi(m, n) = F$ where $S(h)$ represents the surface of the disk with radius h , i.e. the amount of image density contained by the disk is equal to F . The disk is then added to a stack of active locations. This initialisation process is distinguished from the main body of the algorithm in figure 4 by dashed lines, since this can be modified to allow for better representation of image boundaries and will be described later.

The main loop begins by taking the topmost disk in the active list, which at this point is the seed location, and calling it D_a . A candidate disk, D_c , with radius $h_c = h_1$ is placed at a random angle, α , adjacent to the active disk, such that $x_c = x_a + (h_a + h_c) \cos \alpha$ and $y_c = y_a + (h_a + h_c) \sin \alpha$. The radius of the disk, and hence the location (x_c, y_c) , is then adjusted until it too encloses an amount F of the image density. Once sized, the disk is checked for overlap with any other disk. If no overlap is detected then D_c is accepted and added to the active stack, otherwise the candidate is rejected and a new α and disk generated. The placing and re-sizing of candidate disks around D_a continues until some threshold of consecutive failures is exceeded, i.e. there is no more room for disks adjacent to D_a . At this point, D_a is placed on the output points list, and the topmost disk in the active list is popped from the list and now becomes the new D_a . The process continues until the active list is empty, at which point the image will be completely stippled.

By replacing the image density with the target objective function, this method can be adapted to create a high quality

unstructured distribution of sample locations for spatially adaptive PIV applications. The original method requires a user-defined input value F , which indicates how much darkness each stipple should represent, and consequently defines the final number of stipples/samples. In the current work the authors instead relate the amount of objective function represented by each sample, F , in a manner similar to (3). This approach implicitly assumes that all pixels will be covered by a disk, which is not valid when packing circles [22–24]. Despite frequent publication, an elegant solution to calculate the ratio of covered to uncovered area for circles of a specific radius is still lacking. In general, optimal packings of uniform circles in a unit square typically have coverage ratios varying between 0.75–0.85. The packing of non-uniform disks further influences this ratio in an unpredictable manner. The effect of this sub-optimal packing is an under-production of samples in the region of 20%–40%, varying according to both the objective function and the number of samples.

A scaling, η , could be applied to F to accommodate such an under-production of samples, i.e. $\eta \cdot F$ where $\eta < 1$ will result in a greater number of samples being produced. Since the value of η will vary according to both the underlying objective function and the number of samples, it is not possible to obtain a globally optimal value for η . However, for a given objective function and N_w samples, η remains approximately constant and therefore can be estimated iteratively. As such, the algorithm is first run using F as per (3), producing $N_{w,1}$ sample locations. Provided $N_{w,1}$ differs from the target N_w beyond a specified tolerance, the algorithm is repeated using $F = \eta \cdot \iint \phi(x, y) dx dy / N_w$, where η is the running average of the packing factor i.e. $\eta = \frac{1}{K} \sum_{k=1}^K N_{w,k} / N_w$. For a tolerance of 3%, this is typically achieved with two or three passes of the algorithm. Despite the need to repeat the algorithm, the method remains computationally efficient with respect to the SFD approach, partly due to its meshless nature, as shown in figure 6(a).

A second modification required to make stippling a suitable sampling distribution method for PIV applications, is the ability to sample the borders of a domain effectively. This is achieved by extracting the objective function along the identified boundaries and allocating samples adopting the 1D PDF transform method of Secord *et al* [16], allocating a proportional amount of the available sample budget. These locations can then serve as the seed locations for the AIS algorithm instead of using a single random location, replacing the aforementioned initialisation process, represented by the dashed borders in figure 4.

3.3. Initial window sizing

To further reduce user dependency, the authors have also addressed the issue of initial correlation window sizing. While the choice of final window size has been extensively investigated in the past leading to various adaptive strategies [7, 9, 25, 26], the choice of initial window size has received little attention and is typically guided by user experience on

a case-by-case basis. Automated initial window sizing can be based on seeding density, for example, ensuring each window contains on average NI particle images (Yu *et al* [17]). While this yields reliable sizes for images with small displacements or relatively low seeding, this approach neglects the correlation dependency on displacements i.e. the $\frac{1}{4}$ -rule [27], thereby causing valid detection rates to potentially drop significantly.

A modified approach is adopted hereafter and referred to as adaptive initial window sizing (AIW). Window sizes are initially calculated based upon seeding and correlation metrics, namely signal-to-noise ratio of the correlation peak, and associated displacement magnitude relative to the imposed window size, are subsequently evaluated. If either the former falls below a threshold or the $\frac{1}{4}$ -rule is not satisfied, then the WS is incremented by a small amount and re-correlated. The amount by which the WS is incremented can be selected according to how accurate the user would like the WS to be. The smaller the WS increment, the less likely to overshoot the optimum WS at the expense of more failed correlations to reach such a WS. An increment of 6 px has been found to be a good balance. Incrementing by a percentage change causes drastic changes in WS for larger values, potentially leading to significant overshoot of the optimal WS. Provided NI is suitably chosen, the WS based upon seeding will be sufficient for the majority of locations, thus requiring few additional correlations. A conservatively large NI in the region of 20–30 is suggested to accommodate variations in image quality, particle detection performance, out-of-plane motion, etc.

4. Results

4.1. Spatial sample allocation

The performance of each of the sample distribution methods is assessed in four ways; visual comparison, numerical assessment, application in synthetic PIV image analysis, and finally, application in experimental PIV image analysis. In the visual comparison two objective functions are adopted, one being homogeneous (uniform) while the other follows the Franke function defined in (1) (figure 3(a)). In (1), x and y represent the normalised spatial location in the domain. The susceptibility of the PDF transform method to clustering is apparent in figure 5, whereas such locally heightened concentrations of samples are not present with the other distributions.

When applied to the spatially varying Franke function, (1), a globally higher density of samples can be discerned in correspondence with the objective function when applying the PDF approach, although clustering is once again present. The SFD methodology produces a more conducive distribution of samples, although use of (2) reduces agreement between the imposed sampling density (figure 3(a)) and resulting distribution (figure 5(g)). By means of the transformation method proposed in appendix (see (3)) no limitations are imposed on the objective function, thereby relaxing the constraint on inter-sample spacing dynamic range. This results in a distribution, which is more akin to the objective function (figure 5(d)).

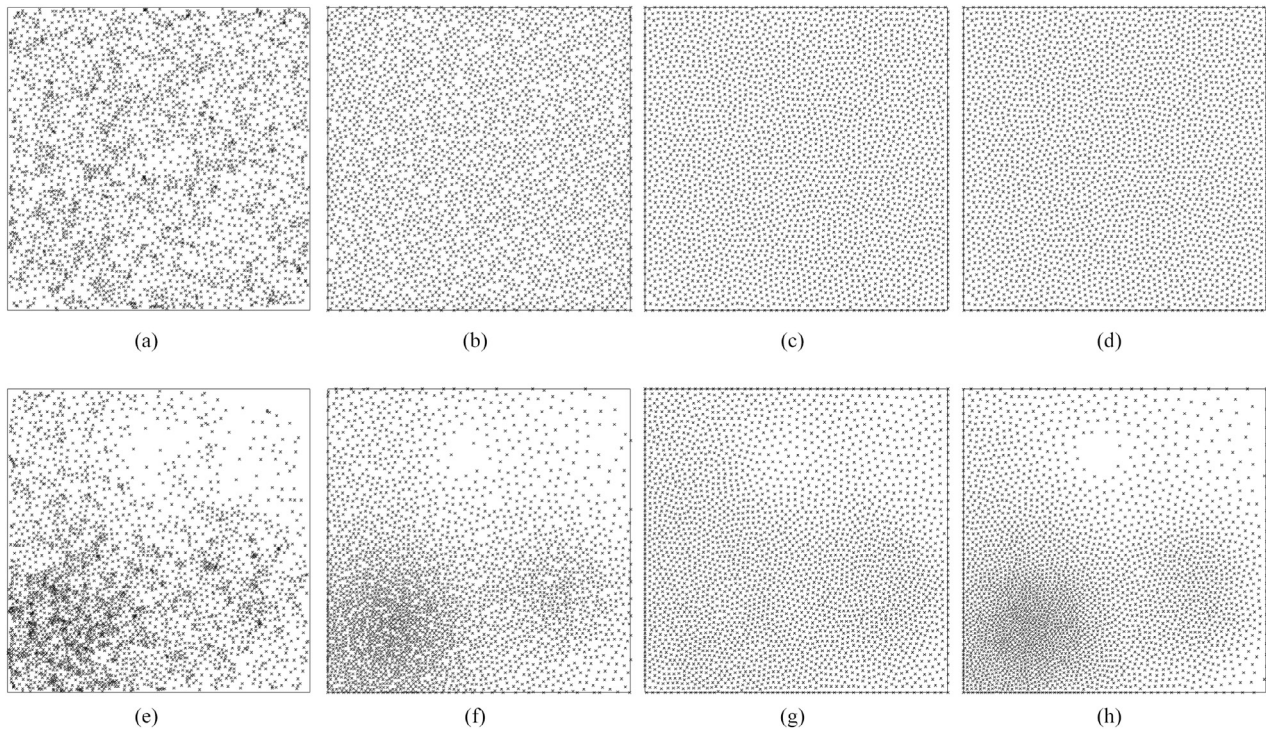


Figure 5. Comparison of sampling distribution methodologies imposing an objective function which is ((a)–(d)) uniform and ((e)–(h)) reflects the Franke function (see (1)). (a)–(e) PDF weighted sample distribution, (b)–(f) Adaptive stippling, (c)–(g) spring-force distribution adopting (2) and (d)–(h) (3).

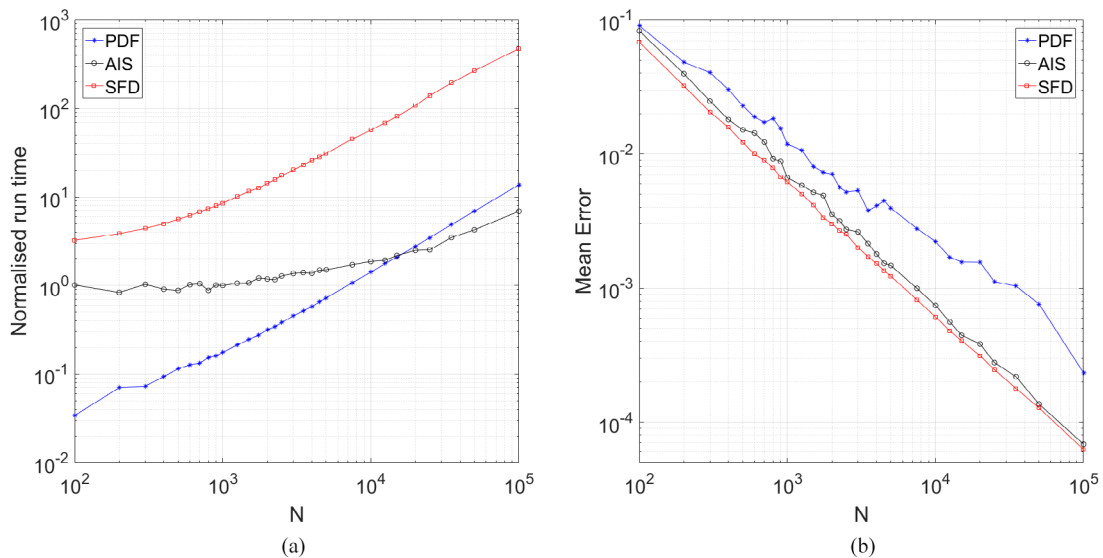


Figure 6. Numerical comparison of the distribution methods in terms of (a) average run-time required to distribute samples according to a spatial sampling density determined by (1), normalised by the time to distribute 1000 samples using adaptive stippling (AIS) and (b) (ensemble) average interpolation error from homogeneously distributed samples.

For this reason, the remaining results pertaining to the SFD method will utilise (3).

4.2. Computational effort and accuracy

The three methods, AIS, PDF, and SFD, as per (3), are assessed in terms of computational requirements and interpolation accuracy on the basis of Monte Carlo simulations. Run-time performance was assessed for a range of sample quantities, N ,

distributed according to Franke’s function over a domain of size 1000 px × 1000 px. For each N , 300 distributions were created and, subsequently, the trends depicted in figure 6(a) represent the average of 300 run-times. While N is small, the PDF method is the fastest of all three methods, followed by AIS. With increasing N this difference reduces and beyond $N = 10^4$, which is a typical number of interrogation windows in PIV analyses, AIS presents itself as the fastest alternative. The AIS method is faster than the SFD method for all N tested

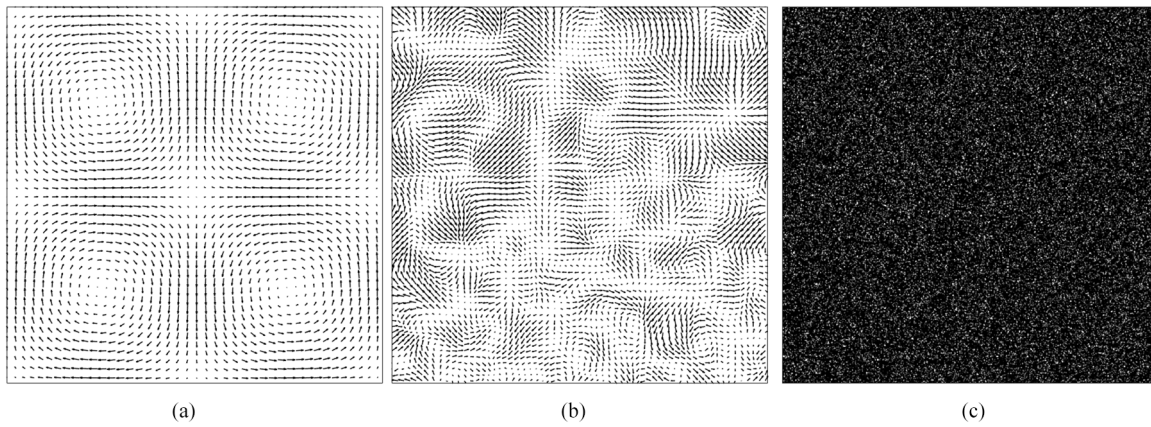


Figure 7. Imposed displacement fields used in the synthetic PIV image analyses. (a) Vortex array (4), subsampled by a factor 24 and scaled by a factor 2 for readability, (b) Gaussian smoothed velocity field, subsampled by a factor of 16 and scaled by 2. (c) Exemplary PIV image.

by at least a factor of three, increasing to two orders of magnitude for 10^5 samples. This performance difference is amplified by the need to re-evaluate the sampling distribution for each iteration in a multi-pass analysis. It is important to note as well that the computational requirements for the SFD depend on the uniformity of the objective function, since more iterations will be required for less uniform objective functions.

The interpolation accuracy is quantified by considering the average error magnitude between an interpolated surface and the known Franke function. For each distribution method, N samples are distributed homogeneously over the domain. The sampled values are then interpolated onto a pixel-wise grid using a natural neighbour interpolation scheme, which has shown to perform well in both PIV using feature tracking [28], and particle tracking velocimetry [29]. At each pixel, the magnitude of the error relative to the reference solution was calculated and spatially averaged over the domain. This process was repeated 300 times for each value of N samples, and averaged over the ensemble to obtain a single value representative of the interpolation performance. The results are presented in figure 6(b).

Differences between AIS and SFD can be seen to be marginal and the interpolation error decreases following approximately $N^{-1.05}$. The PDF approach deviates from this tendency and evolves as roughly $N^{-0.8}$, indicating the extra benefit of each additional sample in an AIS or SFD distribution relative to that of a PDF distribution. The majority of the increased error arises from poor interpolation quality near regions of clustering. In addition, a further source of error comes from the inability to effectively sample the borders of the domain in the PDF method. It is possible to place samples along borders prior to the main algorithm, in a similar manner to the approach discussed in the AIS method, however, it is not then possible for the remaining sampling locations to adjust their locations accordingly. Therefore, with or without the additional sampling, the distribution quality near the boundaries is further decreased resulting in increased interpolation error. While the PDF method might therefore seem advantageous when dealing with fewer samples, it leads to higher levels of interpolation error due to clustering. Distributing samples using the spring-force method on the other hand leads to lower

errors but drastic increases in computational effort. The proposed stippling methodology offers a combination of speed-up while retaining interpolation accuracy.

4.3. Synthetic PIV image analysis

Sample distribution methods were implemented in a PIV image analysis routine and used to analyse synthetic images. Two flow fields were tested (figure 7); a 2×2 array of contra-rotating vortices [30], defined in (4), and a Gaussian smoothed velocity field. In (4) A_0 is the maximum single-component displacement and $L_x = 1000$ px and $L_y = 1000$ px represent the domain dimensions. The second displacement field was obtained by applying a moving Gaussian filter to isotropic random noise, in order to produce a displacement field more analogous of turbulence. The Gaussian filter had linearly varying kernel size, from 51 px at the top of the domain to 15 px at the bottom. A maximum displacement of 10 px was enforced. Both flow fields are displayed in figure 7. For each flow field, a total of 750 images were created with a seeding density of approximately 0.05 particles per pixel. Particle images followed a Gaussian intensity profile with a uniform diameter of 3 px, equivalent to an intensity ratio $\frac{I}{I_{\max}} = e^{-\frac{1}{2}}$, and a uniformly random location within a Gaussian light sheet to mimic experimental conditions. Intensities were subsequently integrated across the pixels with unity pixel fill ratio and in absence of image noise. An exemplary image is shown in figure 7.

$$u = A_0 \cos\left(\frac{2\pi x}{L_x} + \frac{\pi}{2}\right) \cos\left(\frac{2\pi y}{L_y}\right) \quad (4a)$$

$$v = A_0 \sin\left(\frac{2\pi x}{L_x} + \frac{\pi}{2}\right) \sin\left(\frac{2\pi y}{L_y}\right). \quad (4b)$$

In the first iteration, 2500 windows were distributed using one of the three scrutinised distribution methods, utilising only the seeding density as initial objective function $\Phi(x, y) = sd(x, y)$. At each sample location, the AIW algorithm determines the appropriate window size, according to the seeding and displacement magnitude, which is then interpolated onto a pixel wise grid to obtain $WS_0(x, y)$.

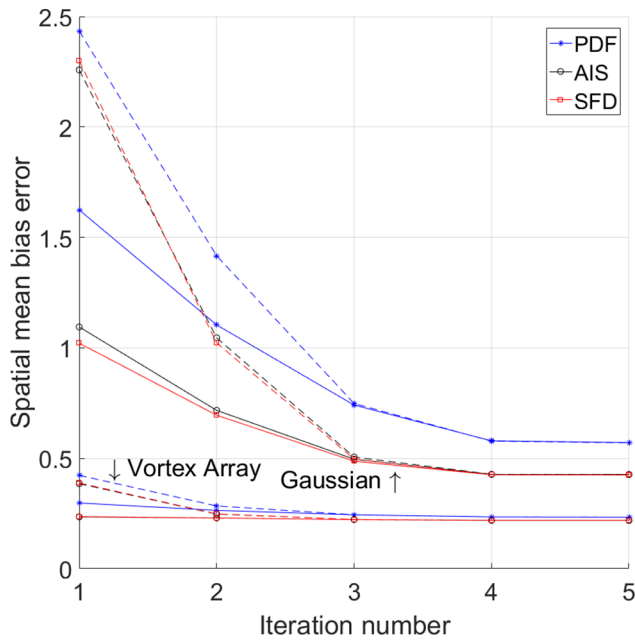


Figure 8. Evolution in $\bar{\epsilon}_k$ with iteration number in the analysis of the (bottom lines) vortex array (see figure 7(a)) and (top lines) Gaussian smoothed random velocity field (see figure 7(b)), adopting different sample distribution methods. Solid lines refer to using the adaptive initial window sizing algorithm. Dashed lines relate to adopting an initial window size of 97 px. In all cases a final WS of 15 px is imposed uniformly across the domain.

Interrogation areas were cross-correlated and sub-pixel accurate displacements were obtained using Gaussian regression [31]. Displacement vectors were then validated by means of the universal outlier detection algorithm [32]. Following a top-hat predictor filter [33], displacements were interpolated using a 3rd order polynomial fit for image deformation or post-processing [34]. Each iteration, k , the budget of correlation windows was set to $2500 \times k$. The objective function consisted of a combination of the local spatial standard deviation of the flow and seeding density, i.e. $\Phi(x, y) = \phi(x, y) \cdot sd(x, y)$ [7]. The window size in iteration k was calculated as $WS_k(x, y) = WS_0(x, y) - (WS_0(x, y) - WS_f) * (k - 1) / (K - 1)$ where WS_f was the final window size, set to 15 px² in this case, and K is the number of iterations (currently $K = 4$). A fifth and final refinement iteration was subsequently performed, without adjusting the window sizes or locations to minimise residual displacements.

To obtain a single heuristic for comparison, each of the 750 displacement fields, at each iteration, were first combined to obtain the pixel-wise-defined ensemble mean displacement field and standard deviation. The bias magnitude, $\epsilon_k(x, y)$, was then calculated and spatially averaged to obtain a single value representing the mean bias magnitude, $\bar{\epsilon}_k$. The mathematical definition is presented in (5), where $\langle U_{m_k} \rangle(x, y)$ is the interpolated, ensemble averaged, measured displacement at pixel location (x, y) for iteration k , and $U_t(x, y)$ is the imposed displacement field.

$$\bar{\epsilon}_k = \frac{1}{L_x L_y} \sum_i \sum_j \epsilon_k(x, y) = \frac{1}{L_x L_y} \sum_i \sum_j |\langle U_{m_k} \rangle(x, y) - U_t(x, y)|. \quad (5)$$

Figure 8 shows the evolution in $\bar{\epsilon}_k$ for each distribution method throughout the iterative analysis procedures. The bias magnitude for the Gaussian smoothed flow field is considerably greater than the vortex array due to the increased complexity of the flow. Ideally, this should be sampled by a greater number of correlation windows to properly spatially sample flow structures and minimise interpolation errors. Furthermore, little reduction in bias magnitude for the vortex array flow field can be noticed beyond the second iteration, indicating that fewer correlation windows may have been sufficient in this case. For both cases, the AIS and SFD performed almost identically, with the only discernible differences occurring in the first and second iterations. Performances attributed to the PDF transform method were consistently inferior, particularly for the Gaussian flow field. While the overall computation time for both the PDF and AIS approaches were similar (within 2%), with the AIS offering a speed up of 1%, the SFD method was approximately 35%–40% slower compared to the PDF approach.

To demonstrate the benefit of the AIW sizing algorithm, also plotted in figure 8 is the bias magnitude versus iteration for each distribution method with a uniform initial window size of 97 px², linearly reducing to the same final WS of 15 px². While there is almost no difference in the final solution, the error in the early iterations is significantly reduced. This is a beneficial trait for adaptive routines which rely on the accuracy of previous iterations to guide sampling. The additional computational cost for performing excess correlations to determine reliable window sizes, are outweighed by the computational savings in correlating overall smaller interrogation areas. In fact, the overall run-times when using the AIW algorithm were between 5%–10% faster than adopting uniformly sized correlation windows. Furthermore, the AIW requires no user input to determine a reliable initial window size and thus represents increased robustness by reducing user dependence.

4.4. Experimental application

The previous assessments have illustrated the ability to distribute correlation windows by means of adaptive stippling. Distributions reflect the imposed objective function and are void of clustering, contrary to those produced using the PDF method. Compared to the spring-force approach, adaptive stippling represents less computational complexity while maintaining accuracy and thus reveals itself as the favoured sample allocation methodology.

The advantage of using AIS over the PDF transform is lastly demonstrated on the basis of experimental PIV images. The case chosen is the flow over a backwards facing step with an expansion ratio of 1.2 at a step height-based Reynolds number of around 5000. In total 250 images were analysed using the same approach as above, i.e. a sampling distribution according to the local spatial standard deviation of the flow field and the seeding density. The temporal standard deviation of the u -component of the displacement field is calculated as $\sigma_{u_i}(x, y) = \sqrt{\sum_i (u_i(x, y) - \overline{u(x, y)})^2 / (N - 1)}$ where $u_i(x, y)$ is the horizontal displacement component at the location (x, y)

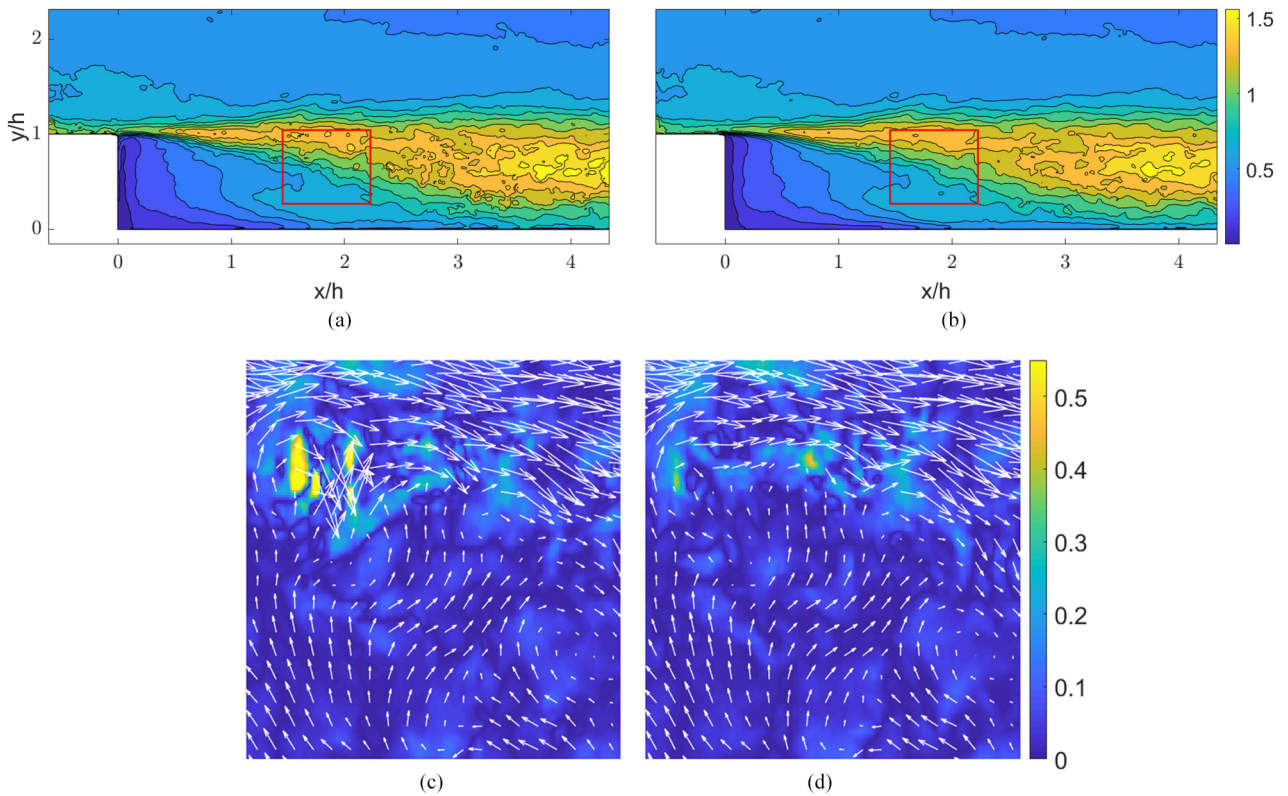


Figure 9. Pixel wise standard deviation, σ_u , over 250 displacement fields in horizontal displacement component for the flow over a backwards facing step at $Re_h \approx 5000$, using (a) the PDF transform method and (b) AIS method. (c) and (d) Show close-ups of an instantaneous vector field, overlaid on the instantaneous vorticity magnitude, for the regions bounded by the rectangles in (a) and (b) respectively, demonstrating the poor vector validation encountered as a consequence of using the PDF transform approach.

for the i th snapshot out of a total of N snapshots, and $\overline{u(x, y)}$ is the ensemble mean displacement component at the location (x, y) . Since both the stippling and spring-force approaches produced similar results, only the PDF and AIS results are presented hereafter. Despite similar results, the total run-time for the SFD was 60% greater than the stippling approach.

Figure 9 shows the standard deviation of the measured horizontal displacement component for both the PDF and AIS approaches, revealing considerably more localised spikes in magnitude, i.e. noise, in the PDF results compared to the approach using AIS. A similar story, though not depicted, is observed in the v -component standard deviation. One contributor to this noise related to interpolation accuracy; a worse interpolation results in greater instantaneous error and thus leads to increased σ in the displacement field. A second, perhaps more significant, contributor to these spikes in standard deviation values come from clusters of outliers, which fail to be detected. An example is presented within figure 9 where a region of one of the instantaneous displacement fields is shown overlaid on the local instantaneous vorticity. Dedicated outlier detection algorithms do exist, such as the works of Masullo *et al* [35], Higham *et al* [36], and Wang *et al* [37]. Nevertheless, such methods are either computationally intensive, rely on proper orthogonal decomposition requiring multiple flow fields, or both as is the case for Wang *et al* whereby they recommend their algorithm only for post-processing given it is too computationally intensive for a multi-pass

routine. While POD based outlier detection can be applied to an instantaneous flow field by sub-dividing the domain into a number of sub-regions, the size and number of sub region become important parameters to be tuned. Furthermore as the results of Higham show, these methods are not totally robust to outliers, particularly as cluster size increases and remains a difficult topic. The improved distribution quality resulting from AIS and SFD ease this challenge, by reducing the possible size of outlier clusters, ensuring a smoother interpolation, and thus significantly reducing the chance of irrecoverably distorting the underlying image and propagating outliers into the final solution. This can be considered to be an additional benefit of the proposed stippling distribution.

5. Conclusions

Adaptive PIV analysis routines require a fast and robust distribution method to obtain a spatially varying sampling density, faithful to some imposed objective function, without compromising interpolation reconstruction or hindering vector validation routines. Distribution methods presented in existing literature sacrifice either distribution quality or computational efficiency and robustness. In this work, a method based on adaptive incremental stippling (AIS) has been presented, producing sampling distributions of high quality while remaining computationally efficient without the need for case dependent parameter tuning.

Distributions have been compared both numerically stand-alone and within PIV analysis routines using synthetic and experimental images. The AIS methodology is shown to yield interpolation errors equal to the (ideal) spring-force distribution approach, while offering a speed up of at least a factor 2. Simultaneously, contrary to allocation of samples on the basis of probability density transformations, AIS no longer suffers from sample clustering. This is shown to improve accuracy due to improved outlier detection and reduced levels in standard deviation, which were artificially introduced by undetected erroneous vector clusters. By maintaining both quality and computational efficiency, the proposed AIS sample distribution approach enables spatially adaptive routines to become increasingly desirable over their structured counterparts.

Acknowledgment

The authors would like to thank EPSRC for Matthew Edwards' financial support (EP/M507994/1).

Data access statement

All underlying data are described and provided in full within this paper.

Appendix

Sampling locations in Yu *et al* [10] are based initially on seeding ($h_0(x, y) = (1 - WOR) \cdot \sqrt{NI \times sd(x, y)^{-1}}$) and adapted to flow conditions in subsequent iterations by scaling the sample spacing ($h(x, y) = h_0(x, y) \cdot \phi^*(x, y) = h_0(x, y) \cdot g(\phi)$). The function $g(\phi)$ converts the flow metrics, which naturally correlate directly with the intended sampling density, into a distance-based format. In this approach, $g(\phi)$ is defined in (2), representing a bounded inverse function. These restrictions hamper a general, objective, assessment of the different sampling distribution method's performance, for a given objective function, since bounds in ϕ^* translate into effective alterations of the objective function. The current authors circumvent this restriction by introducing a transformation of imposed objective functions (used in AIS) into equivalent sample spacing distribution (used in SFD).

For a user-defined number of correlation windows, N_w , a radius, $h(x, y)$, exists which will enclose a constant proportion F of the objective function, which combines information pertaining flow metrics and seeding density, around the location (x, y) . This radius is synonymous with the sample spacing. The value F is calculated such that $N_w \cdot F = \int \int \Phi(x, y) dx dy$;

$$h(x, y) = \frac{1}{\pi} \sqrt{F/\Phi(x, y)} = \frac{1}{\pi} \sqrt{F/(\phi(x, y) \cdot sd(x, y))}. \quad (\text{A.1})$$

Substituting $F = N_w^{-1} \int \int \Phi(x, y) dx dy = K \cdot N_w^{-1}$ yields $h(x, y) = \frac{1}{\pi} \sqrt{K/(N_w \cdot \Phi(x, y))}$. The number of windows can be based solely on the basis of seeding density;

$$N_w = \frac{\int \int (sd(x, y)) dx dy}{(1 - WOR)^2 \cdot NI} = \frac{N_p}{(1 - WOR)^2 \cdot NI}. \quad (\text{A.2})$$

Parameter WOR signifies the mean window overlap ratio, NI the number of particles captured by each correlation window, and N_p symbolises the total number of particles in the image [7, 10]. Substitution into $h(x, y)$ then yields

$$h(x, y) = \frac{1}{\pi} \sqrt{(1 - WOR)^2 \sqrt{NI/sd(x, y)} \sqrt{K/(N_p \cdot \phi(x, y))}}. \quad (\text{A.3})$$

With the expression for $h_0(x, y)$ follows

$$\begin{aligned} h(x, y) &= \frac{1}{\pi} \sqrt{K/N_p} h_0(x, y) \sqrt{1/\phi(x, y)} \\ &\equiv \text{const} \cdot h_0(x, y) \sqrt{1/\phi(x, y)}. \end{aligned} \quad (\text{A.4})$$

Since the SFD solves the force system by computing relative forces, to avoid an implicit constraint with the number of nodes and the objective function, the constant term is irrelevant and this equation is therefore effectively

$$h(x, y) = h_0(x, y) \sqrt{1/\phi(x, y)} = h_0(x, y) \cdot g(\phi). \quad (\text{A.5})$$

Identical to the original SFD method, the rewritten expression for the sample spacing includes a seeding term and a flow term, constituting the objective function, though without imposing limitations in $g(\phi)$. This illustrates the equivalence between AIS and SFD, and permits an unbiased juxtaposition of the obtained sample distributions.

ORCID iDs

M Edwards  <https://orcid.org/0000-0002-7036-8789>

R Theunissen  <https://orcid.org/0000-0001-5844-5158>

References

- [1] Raffel M, Willert C E, Wereley S and Kompenhans J 2007 *Particle Image Velocimetry: a Practical Guide* (New York: Springer)
- [2] Scarano F and Riethmuller M L 1999 Iterative multigrid approach in PIV image processing with discrete window offset *Exp. Fluids* **26** 513–23
- [3] Mackman T J and Allen C B 2010 Investigation of an adaptive sampling method for data interpolation using radial basis functions *Int. J. Numer. Methods Eng.* **83** 915–38
- [4] Stanislas M, Okamoto K, Kähler C J, Westerweel J and Scarano F 2008 Main results of the Third International PIV challenge *Exp. Fluids* **45** 27–71
- [5] Kähler C J, Astarita T, Vlachos P P, Sakakibara J, Hain R, Discetti S, La Foy R and Cierpka C 2016 Main results of the 4th International PIV Challenge *Exp. Fluids* **57** 97
- [6] Scarano F 2003 Theory of non-isotropic spatial resolution in PIV *Exp. Fluids* **35** 268–77
- [7] Theunissen R, Scarano F and Riethmuller M L 2007 An adaptive sampling and windowing interrogation method in PIV *Meas. Sci. Technol.* **18** 275–87

- [8] Theunissen R, Scarano F and Riethmuller M L 2008 On improvement of PIV image interrogation near stationary interfaces *Exp. Fluids* **45** 557–72
- [9] Wieneke B and Pfeiffer K 2010 Adaptive PIV with variable interrogation window size and shape *Proc. 15th Int. Symp. on Applications of Laser Techniques to Fluid Mechanics (Lisbon, Portugal)*
- [10] Yu K and Xu J 2016 Adaptive PIV algorithm based on seeding density and velocity information *Flow Meas. Instrum.* **51** 21–9
- [11] Theunissen R and Gjelstrup P 2018 Adaptive sampling in higher dimensions for point-wise experimental measurement techniques *Meas. Sci. Technol.* **29** 085007
- [12] Theunissen R, Kadosh J S and Allen C B 2015 Autonomous spatially adaptive sampling in experiments based on curvature, statistical error and sample spacing with applications in LDA measurements *Exp. Fluids* **56** 116
- [13] Secord A 2002 Weighted Voronoi stippling *Proc. the 2nd Int. Symp. on Non-Photorealistic Animation and Rendering* p 37
- [14] Dussol D, Druault P, Mallat B, Delacroix S and Germain G 2016 Automatic dynamic mask extraction for PIV images containing an unsteady interface, bubbles, and a moving structure *C. R.—Mec.* **344** 464–78
- [15] Masullo A and Theunissen R 2017 Automated mask generation for PIV image analysis based on pixel intensity statistics *Exp. Fluids* **58** 1–11
- [16] Secord A, Heidrich W and Streit L 2002 Fast primitive distribution for illustration *Proc. EGWR* pp 215–26
- [17] Yu K, Xu J, Tang L and Mo J 2015 New adaptive sampling method in particle image velocimetry *Meas. Sci. Technol.* **26** 037002
- [18] Persson P and Strang G 2004 A simple mesh generator in Matlab *SIAM Rev.* **46** 329–45
- [19] Franke R 1979 A critical comparison of some methods for interpolation of scattered data *Naval Postgraduate School Technical Report*
- [20] Persson P 2004 Mesh generation for implicit geometries *Doctoral Dissertation* Massachusetts Institute of Technology
- [21] Ascencio-Lopez I, Meruvia-Pastor O and Hidalgo-Silva H 2010 Adaptive incremental stippling using the Poisson-disk distribution *J. Graph. GPU Game Tools* **15** 29–47
- [22] Huang W and Ye T 2010 Greedy vacancy search algorithm for packing equal circles in a square *Oper. Res. Lett.* **38** 378–82
- [23] Shewchuk J R 1997 Geometry 1997 *Discrete Comput. Geom.* **363** 305–63
- [24] Boll D W, Donovan J, Graham R L and Lubachevsky B D 2000 Improving dense packings of equal disks in a square *Electron. J. Comb.* **7** 1–9
- [25] Di Florio D, Di Felice F and Romano G P 2002 Windowing, re-shaping and re-orientation interrogation windows in particle image velocimetry for the investigation of shear flows *Meas. Sci. Technol.* **13** 953–62
- [26] Scarano F 2002 Iterative image deformation methods in PIV *Meas. Sci. Technol.* **13** R1–19
- [27] Keane R D and Adrian R J 1990 Optimization of particle image velocimeters. I. Double pulsed systems *Meas. Sci. Technol.* **1** 1202–15
- [28] Miozzi M 2004 Particle image velocimetry using feature tracking and Delaunay tessellation *Proc. of the 12th Int. Symp. on Application of Laser Technique to Fluid Mechanics (Lisbon, Portugal)* pp 1–12
- [29] Lloyd P M, Stansby P K and Ball D J 1995 Unsteady surface-velocity field measurement using particle tracking velocimetry *J. Hydraul. Res.* **33** 519–34
- [30] Shinneeb A M, Bugg J D and Balachandar R 2004 Variable threshold outlier identification in PIV data *Meas. Sci. Technol.* **15** 1722–32
- [31] Nobach H and Honkanen M 2005 Two-dimensional Gaussian regression for sub-pixel displacement estimation in particle image velocimetry or particle position estimation in particle tracking velocimetry *Exp. Fluids* **38** 511–5
- [32] Westerweel J and Scarano F 2005 Universal outlier detection for PIV data *Exp. Fluids* **39** 1096–100
- [33] Schrijer F F J and Scarano F 2008 Effect of predictor–corrector filtering on the stability and spatial resolution of iterative PIV interrogation *Exp. Fluids* **45** 927–41
- [34] Scarano F and Riethmuller M L 2000 Advances in iterative multigrid PIV image processing *Exp. Fluids* **29** S051–60
- [35] Masullo A and Theunissen R 2016 Adaptive vector validation in image velocimetry to minimise the influence of outlier clusters *Exp. Fluids* **57** 1–21
- [36] Higham J E, Brevis W and Keylock C J 2016 A rapid non-iterative proper orthogonal decomposition based outlier detection and correction for PIV data *Meas. Sci. Technol.* **27** 125303
- [37] Wang H P, Gao Q, Feng L H, Wei R J and Wang J J 2015 Proper orthogonal decomposition based outlier correction for PIV data *Exp. Fluids* **56** 1–15