https://doi.org/10.1186/s42774-021-00087-6

Advances in Aerodynamics

RESEARCH Open Access

Particle reconstruction of volumetric particle image velocimetry with the strategy of machine learning

Qi Gao[1], Shaowu Pan[2*], Hongping Wang[3,4], Runjie Wei[5] and Jinjun Wang[6]

*Correspondence:
shawnpan@umich.edu
[2]Department of Aerospace Engineering, University of Michigan, 48105 Ann Arbor, United States
Full list of author information is available at the end of the article

Abstract

Three-dimensional particle reconstruction with limited two-dimensional projections is an under-determined inverse problem that the exact solution is often difficult to be obtained. In general, approximate solutions can be obtained by iterative optimization methods. In the current work, a practical particle reconstruction method based on a convolutional neural network (CNN) with geometry-informed features is proposed. The proposed technique can refine the particle reconstruction from a very coarse initial guess of particle distribution that is generated by any traditional algebraic reconstruction technique (ART) based methods. Compared with available ART-based algorithms, the novel technique makes significant improvements in terms of reconstruction quality, robustness to noise, and at least an order of magnitude faster in the offline stage.

Keywords: Particle reconstruction, Volumetric particle image velocimetry, Convolutional neural network

1 Introduction

Particle image velocimetry (PIV) is a widely used technique for measuring velocity fields [1, 2]. With volumetric PIV measurement, complex flows can be investigated regarding their three-dimensional three-component (3D3C) flow structures. Among all the 3D3C measurement methods, tomographic PIV (Tomo-PIV) proposed by [3] has been proved on its success of making an accurate measurement with fine spatial resolution under a fairly high particle seeding density of 0.05 ppp (particle per pixel). The key procedure of Tomo-PIV is the particle reconstruction (PR), which is a process of solving inverse projection problem from two-dimensional particle images to 3D intensity distribution of particles. In the original article of Tomo-PIV by [3], the multiplicative algebraic reconstruction technique (MART) based on the maximum entropy criterion was introduced to reconstruct the 3D particle field. Since then, numerous advanced techniques have been developed to optimize the 3D particle reconstruction for improving either accuracy or efficiency, which has been well-reviewed by [4] and [5]. Most available particle reconstruction techniques are based on MART algorithms, such as the spatial filtering MART (SF-MART), which applies spatial filtering on the reconstructed particle intensity field after each MART iteration [6]. SF-MART provides a better reconstruction quality than the traditional MART algorithm, which will be tested and compared with the new technique in the current work.

For the PR problem, with the increase of particle seeding density, the reconstruction quality decreases rapidly due to the issue of ghost particles, which is a fake particle unexpectedly generated at the intersections of light of sight (LOS). Many algorithms were proposed to accelerate the optimization of PR by providing a good initialization: [7] used multiplicative first guess (MFG) as a precursor to the standard MART approach, which can provide a reasonably accurate solution as the initial condition for MART iteration and also accelerate the convergence. [8] further proposed a multiplicative LOS (MLOS) estimation to determine the possible particle locations without requiring the weighting matrix as MFG. Besides having a good initialization, the removal of ghost particles can substantially improve the reconstruction quality. The joint distribution of peak intensity and track length can be used to successfully separate ghost particles and actual particles in certain cases [9]. A simulacrum matching-based reconstruction enhancement (SMRE) technique proposed by [10] utilizes the characteristic shape and size of actual particles to remove ghost particles in the reconstructed intensity field. The Shake-The-Box (STB) approach [11, 12] estimates trajectories based on previous time steps. The particle locations are consequently corrected by the Iterative Reconstruction of Volumetric Particle Distribution (IPR) proposed by [13]. STB has a considerable improvement compared to MART in both accuracy and particle concentration. For time-resolved image acquisition, sequential motion tracking enhancement MART (SMTE-MART) proposed by [14] also produces a time-marching estimation of the object intensity field based on an enhanced guess, which is built upon the object reconstructed at the previous time instant. This method yields superior reconstruction quality and higher velocity field measurement precision when compared with both MART and MTE-MART [15]. For single volume reconstruction, some new reconstruction schemes were developed. Intensity-enhanced MART (IntE-MART) uses a histogram-based intensity reduction to suppress the intensity of ghosts [16]. Gesemann et al. [17] solved the volume intensity using an optimization algorithm based on constrained least squares strategies and L1-regularization. Ye et al. [18] proposed a dual-basis pursuit approach for particle reconstruction, which yielded higher reconstruction quality compared with MART in 2D simulations. In order to reduce the computational time, [19] presented a memory-efficient and highly parallelizable method based on a homography fit synthetic aperture refocusing method. Rather than a ‘voxel-oriented’ approach, [20] proposed an ‘object-oriented’ approach called Iterative Object Detection-Object Volume Reconstruction based on Marked Point Process (IOD-OVRMPP) for the reconstruction of a population of 3D objects. The particle position can be directly obtained using this method.

With the development of machine learning in the field of image processing, designing a model based on machine learning to deal with various image-related tasks has become a hot topic. In the past few years, neural networks have been applied to particle image velocimetry. Machine learning has been utilized to replace traditional cross-correlation for velocity deduction with dense particle seeding [21, 22]. Recently, a series of work has been presented in a conference, ‘13th International Symposium on Particle Image Velocimetry’ (ISPIV 2019, Munich, Germany, July 22-24). For example, [23] applied convolutional neural networks (CNN) to PIV and achieved similar effects of traditional cross-correlation algorithms. Liang et al. [24] used CNN as a filtering step after several

MART iterations for particle reconstruction. However, at the moment, most existing works on applying machine learning to PIV are two dimensional while an investigation of applying machine learning on particle reconstruction, as a fully three-dimensional application, is still lacking. In this work, we present a novel machine learning framework ('AI-PR') using CNN [25] for 3D particle reconstruction problems.

This paper is organized as follows. In Section 2.1, the mathematical formulation of particle reconstruction is presented. In Section 2.2–2.3, the proposed architecture of AI-PR is described. In Section 3, as a preliminary study, comparison of AI-PR against traditional SF-MART based algorithms on synthetic point cloud data is presented in terms of reconstruction quality, computational efficiency and robustness to noise. Finally, conclusions and future directions are summarized in Section 4.

2 Principle of particle reconstruction with machine learning

2.1 Particle reconstruction in TPIV as an inverse problem

Since we cannot directly measure the 3D discrete particle field, we consider recovering the continuous 3D light intensity distribution resulting from the scattering by particles [3] from several 2D projections as an inverse problem [26]. For simplicity, we refer to such an inverse problem as particle reconstruction. Consider a fixed three dimensional orthogonal coordinate system, $(x, y, z) \in \mathbb{R}^3$, the unknown light intensity field can be viewed as a continuous source function $f \in C^0(\mathcal{D})$ satisfying,

$$f(x,y,z) \geq 0, \quad \iiint_{\mathcal{D}} f(x,y,z)dxdydz = 1, \tag{1}$$

where $\mathcal{D} \subset \mathbb{R}^3$ is a compact support of f.

Assuming parallel projection (or point spread function), without loss of generality, a view can be defined as a rotation of coordinate system with respect to some certain origins. One can further introduce different translations for cameras but it is ignored in the context for better illustration. The coordinate in the rotated system is $(x', y', z') \in \mathbb{R}^3$ where $x' - y'$ plane is parallel to the projection plane of the view, i.e., z' is parallel to the line of sight, determined by the following relation,

$$\begin{bmatrix} x' & y' & z' \end{bmatrix}^\top = \mathbf{T} \begin{bmatrix} x & y & z \end{bmatrix}^\top, \tag{2}$$

where the rotation matrix $\mathbf{T}$ specified by three Euler angles α, β, γ is defined as,

$$\mathbf{T} = \begin{bmatrix} \cos\alpha\cos\gamma - \sin\alpha\cos\beta\sin\gamma & -\cos\alpha\sin\gamma - \sin\alpha\cos\beta\cos\gamma & \sin\alpha\sin\beta \\ \sin\alpha\cos\gamma - \cos\alpha\cos\beta\sin\gamma & -\sin\alpha\sin\gamma + \cos\alpha\cos\beta\cos\gamma & -\cos\alpha\sin\beta \\ \sin\beta\sin\gamma & \sin\beta\cos\gamma & \cos\beta \end{bmatrix}. \tag{3}$$

In practice, there are J views, i.e., the number of cameras, usually ranging from 4 to 6. For each j-th view, the two dimensional projection field $g_j(x', y')$ is given as,

$$g_j(x', y') = \int_{-\infty}^{\infty} f\left(\mathbf{T}_j^{-1} \begin{bmatrix} x' & y' & z' \end{bmatrix}^\top\right) dz'. \tag{4}$$

As illustrated in Fig. 1, the goal of the inverse problem is to find the source function f, given projection data $\{g_j\}_{j=1}^{J}$ in the discretized form, i.e., f is pixelized as a function dealing with 3D matrix and g_j as 2D images. Unfortunately, it is known to have an infinite number

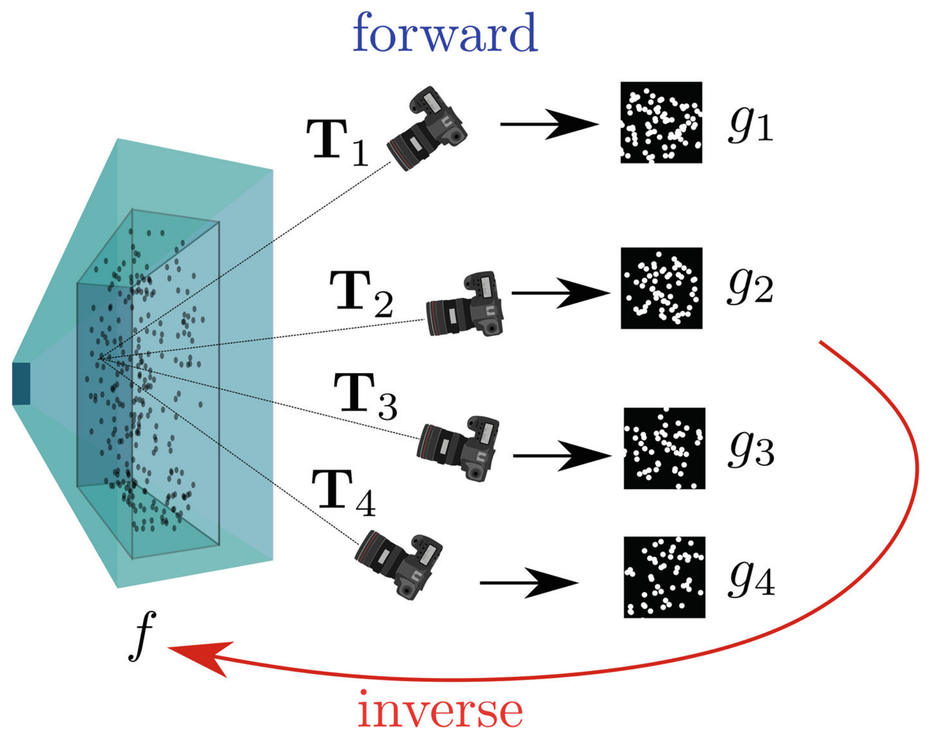

Fig. 1 Illustration of particle reconstruction as an inverse problem

of solutions satisfying all the above conditions [27, 28]. Most often, additional conditions, e.g., entropy maximization [26], are considered to enforce uniqueness.

2.2 Learning particle reconstruction field via CNN

2.2.1 CNN as a general and powerful field processing tool

In recent years, with the increasing amount of data and computational power, CNN has become quite popular in many science and engineering communities with remarkable performance against traditional methods. Several examples include image processing: classification [29, 30], object recognition [31], segmentation [32], inverse problem [33]; prediction of aerodynamics [34, 35]; model-order-reduction of fluid flows [36]. The main idea is to process the field with *convolution* with non-linear activation that leverages the locality (translation-equivariance) of the solution for many problems involving mapping on the spatial field, e.g., computing spatial derivative or average of a nonlinear function of the field. In Fig. 2 we give an example of 2D *linear* convolution on images. Note that by performing the convolution operation, the original 5×5 image is transformed into a 3×3 image. Such shrinkage in the image size is not favored in the context of deep CNN [37] since one would prefer the size of the output of CNN to be the same as the input size, which is the case in the particle reconstruction problem. In this case, we consider padding zeros [37] around the original images so that the convoluted image would have the same size as the original image. Further, one can apply element-wise operation of a known non-linear activation function $\sigma(\cdot) : \mathbb{R} \mapsto \mathbb{R}$ to make convolution nonlinear. Typical activation functions can be tanh, ReLU [37], etc. In this work, we use the ReLU activation function that is defined as,

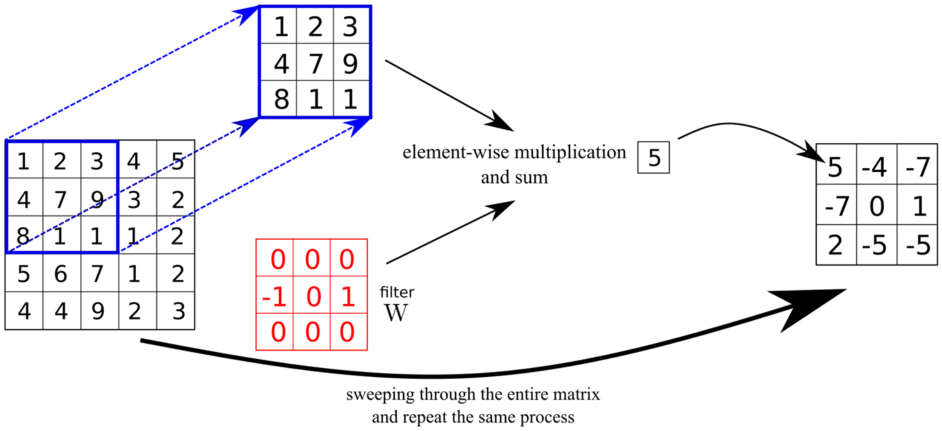

Fig. 2 Illustration of 2D convolution of 3 × 3 kernel *W* on a 5 × 5 matrix

$$\sigma_{\mathrm{ReLU}}(x) = \max(0, x). \tag{5}$$

For more information about CNN, interested readers are referred to the following excellent reviews [33, 38–40].

2.2.2 Mathematical formulation of a single 3D convolutional layer

Recall that we are interested in applying the 3D analogy of 2D convolution introduced in the previous section for particle reconstruction. Instead of an image, the input of 3D CNN is a 3D Cartesian field or a 3D tensor. Note that from the previous section, filter W uniquely determines the convolution. Therefore, it is straightforward to see that if one performs convolution on the same original image with Q different filters, one can end up with Q output images, which are denoted as a 4D tensor. In the community of image processing, a single image at each layer is called a channel, which comes from the RGB channels in digital images [41]. Thus from now on, we denote the shape of a general 3D multi-channel field as a 4D tensor with shape $N_x \times N_y \times N_z \times Q$, with N_x, N_y, N_z representing the size of the 3D Cartesian field and Q representing the number of channels. For convenience, the ensemble of Q filters is called *kernel*, which uniquely determines the above Q convolutions from a 3D tensor to a 4D tensor. This concept can be generalized to convolutions between any 3D multi-channel field with different numbers of channels.

Now we consider the s-strided convolution operation with the generalized kernel $\mathbf{K} \in \mathbb{R}^{L\times M\times N\times Q\times Q^{'}}$ on 3D Q-channel field $\mathbf{V} \in \mathbb{R}^{N_x\times N_y\times N_z\times Q}$ with an output as 3D $Q^{'}$-channel field $\mathbf{Z} \in \mathbb{R}^{N_x^{'}\times N_y^{'}\times N_z^{'}\times Q^{'}}$ as $\mathbf{Z} = c(\mathbf{K}, \mathbf{V}, s)$. L, M, N are positive odd numbers representing the width of the 3D convolutional filter $\mathbf{K}$ in each direction. s is the stride length in convolution. Additionally, this kernel $\mathbf{K}$ contains $Q \times Q^{'}$ filters that are defined in Section 2.2.1. Specifically, for index $1 \le i \le N_x^{'}$, $1 \le j \le N_y^{'}$, $1 \le k \le N_z^{'}$, and channel index $1 \le q' \le Q'$, combining with zero-padding in Eq. 6 to avoid shrinkage of image size so as to enable deeper neural networks,

$$\tilde{\mathbf{V}}(l, m, n; i, j, k) = \begin{cases} \mathbf{V}_{(i-1)s+l,(j-1)s+m,(k-1)s+n,q} & \text{if } 1 \le (i-1)s + l \le N_x \\ & \text{and } 1 \le (j-1)s + m \le N_y \\ & \text{and } 1 \le (k-1)s + n \le N_z \\ 0 & \text{otherwise} \end{cases}, \tag{6}$$

we have the following general expression for zero-padding convolution operation,

$$\mathbf{Z}_{i,j,k,q'} = c(\mathbf{K},\mathbf{V},s)_{i,j,k,q'} \tag{7}$$

$$= \sum_{q=1}^{Q} \sum_{l=\frac{1-L}{2}}^{\frac{L+1}{2}} \sum_{m=\frac{1-M}{2}}^{\frac{M+1}{2}} \sum_{n=\frac{1-N}{2}}^{\frac{N+1}{2}} \tilde{\mathbf{V}}(l,m,n;i,j,k)\mathbf{K}_{l,m,n,q,q'}, \tag{8}$$

where l, m, n are indices of 3D filters, e.g., $-1 \leq l, m, n \leq 1$ for $L = M = N = 3$. It is well-known that convolution operation has a close connection to finite differences. For example, when $L = 3$, such convolution operation contains finite difference approximation of first and second order spatial derivatives.

After obtaining the output field $\mathbf{Z}$, an element-wise nonlinear activation function $\sigma(\cdot) : \mathbb{R} \mapsto \mathbb{R}$ is applied on $\mathbf{Z}$. Finally, the whole process including the nonlinear activation above defined in Eq. 9 is called a *convolutional layer* $\mathcal{C}$ without pooling,

$$\mathbf{V}' = \sigma(\mathbf{Z}) = \sigma(c(\mathbf{K},\mathbf{V},s)) = \mathcal{C}(\mathbf{V}) \in \mathbb{R}^{N_x' \times N_y' \times N_z' \times Q'}. \tag{9}$$

In summary, the above nonlinear convolution transforms a 3D Q-channel tensor into another 3D Q'-channel tensor. It is important to note that, to fully determine such convolution, one just needs to determine the filters in the kernel, which will be discussed in Section 2.3.

2.2.3 Geometry-informed input features

Instead of naively taking input as J images from the cameras, we consider input for the 3D CNN as the particle field generated by MLOS method: $\mathbf{E}_{\mathrm{MLOS}}$ in Eq. 10. Because the geometrical optics information, i.e., directions and positions of all the cameras, is naturally embedded, $\mathbf{E}_{\mathrm{MLOS}}$ is geometry-informed.

$$\mathbf{E}_{\mathrm{MLOS}}(x,y,z) = \prod_{j=1}^{J} g_j\left(\tilde{\mathbf{T}}\begin{bmatrix} x & y & z \end{bmatrix}^{\top}\right), \tag{10}$$

where $\tilde{\mathbf{T}}$ is the first two rows of $\mathbf{T}$.

2.3 Architecture of AI-PR

Unlike traditional MART-based methods which don't require any data, the framework of AI-PR depends on *data*: a particle field f and the corresponding 2D images projected on the cameras. However, in real experiments, it is often impossible to obtain f, i.e., the exact locations and intensity distribution of particles from a measurement. Hence, synthetic random particle field with resolution as 256×256×128 is employed as training and testing data. The synthetic particle fields and their images are generated following a typical way that has been widely used for testing PR algorithms. Details can be found in [16] and [18]. Four projections of particle fields were calculated from given mapping functions to simulate camera imaging. The initial MLOS field was then computed and prepared as input for the aforementioned 3D CNN.

2.3.1 Overcoming memory bottleneck with a divide-and-conquer approach

A key difference between 3D PR and most 2D computer vision problems is that a typical 3D PR usually requires large magnitude of memory usage due to increase from $O(n^2)$ to $O(n^3)$, where n roughly represents the resolution in one direction. For a typical fluid dynamics problem, $n \sim O(10^2)$. While convolutions are highly parallel and optimized on a typical graphical card which often contains limited memory. Then it becomes challenging to perform even mini-batch training [37] with 3D convolution operation on such a huge 4D tensor on GPU especially in our framework where size-reduction operation, i.e., pooling, is absent. One of the direct solutions is to implement parallel 3D convolutions [42, 43]. Instead, we consider a divide-and-conquer approach. We divide the input MLOS field ($256 \times 256 \times 128$) into $4 \times 4 \times 4 = 64$ sub-fields ($64 \times 64 \times 32$) by dividing its length along each direction into 4 equal pieces. Then we assume that f within each subfield can be inferred from the $\mathbf{E}_{\text{MLOS}}$ in that subfield, i.e., assuming the mapping from MLOS field to the actual particle field can be approximated by an affordable non-local mapping, instead of a global mapping. Effectively, the size of input and output for 3D CNN is reduced by a factor of $4^3 = 64$ while the number of data is increased by a factor of 64. Therefore, such a divide-and-conquer approach makes the training for 3D CNN affordable while increasing the number of data for mini-batch training. Finally, we concatenate 64 output sub-field from CNN into the field with the same size as original ($256 \times 256 \times 128$). It should be noted that when we test the model, we have to divide the corresponding testing dataset into several 3D blocks with the shape as $64 \times 64 \times 32$, potentially with some overlapping. In the following section, we will show that we can apply our trained model on MLOS field of an even larger resolution thanks to the divide-and-conquer approach. But again, one must first divide the field into sub-field with the shape of $64 \times 64 \times 32$ in order to apply the trained model.

2.3.2 Learning particle reconstruction with 3D CNN

Given the definition of a *convolutional layer* in Section 2.2.2 as the building block, the architecture of AI-PR is illustrated in Fig. 3. Following three major steps, one can obtain a good approximation of f with 3D CNN output field f_{CNN}:

1 $\mathbf{E}_{\text{MLOS}}$ is first calculated from multiple two-dimensional particle images by camera imaging, which is the same as traditional PR algorithms, while note that MLOS method has been noticed as a very good initial guess of particle field in MART-based algorithms.

2 A 3D CNN is employed with batch normalization [44] such that it takes the input MLOS field $\mathbf{E}_{\text{MLOS}}$ and output f_{CNN}.

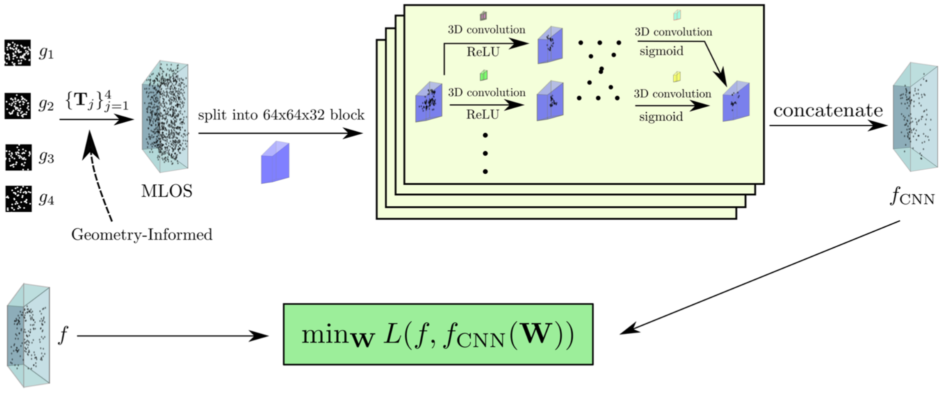

Fig. 3 Schematic diagram of AI-PR

3 Stochastic gradient-based optimization is performed, e.g., ADAM [45], on kernel in all the layers to minimize the difference defined in Eq. 11 between *known f* from the training data and the corresponding f_{CNN}.

$$L(f, f_{\mathrm{CNN}}(\mathbf{W})) = \frac{\sum_{j=1}^{M_{sub}M} \sum_{i=1}^{M_{res}} f^{(j)}(I_i, J_i, K_i) f_{\mathrm{CNN}}^{(j)}(I_i, J_i, K_i; \mathbf{W})}{\epsilon + \sum_{j=1}^{M_{sub}M} \sum_{i=1}^{M_{res}} f^{(j)}(I_i, J_i, K_i) - f_{\mathrm{CNN}}^{(j)}(I_i, J_i, K_i; \mathbf{W})}, \tag{11}$$

where superscript (j) corresponds to j-th synthetic random particle field in the training data, M is the total number of random clouds in the synthetic data, M_{sub} is the number of sub-fields, e.g., 64, and M_{res} is the total number of voxels of the sub-fields, e.g., $64 \times 64 \times 32 = 131072$. I_i, J_i, K_i are spatial indices of i-th voxel and $\mathbf{W}$ is the set of filters in all the kernels in the network. ϵ is a small constant to avoid zero denominator. In addition, batch normalization [44] is used to accelerate the optimization.

Finally, after f_{CNN} is trained (loss is sufficiently minimized over training data), to obtain a particle reconstruction for an unknown particle field f, one just needs to compute MLOS field from the camera projections of f and then take $\mathbf{E}_{\mathrm{MLOS}}$ as input of the trained CNN to obtain f_{CNN}, which is supposed to be a good approximation of f.

2.3.3 Structure of 3D CNN

In this work, a deep 3D CNN with 12 hidden layers is employed. The size of input/output layer is $64 \times 64 \times 32$ while that of hidden layers is $64 \times 64 \times 32 \times 16$, i.e., each hidden layer has 16 channels with size unchanged. We consider the ReLU activation function in Eq. 5 in each layer except the output layer where Sigmoid defined in Eq. 12 is considered to ensure the output is bounded between 0 and 1. The convolution kernel of the input/output layer has a size of $3 \times 3 \times 3 \times 16$, while the other layers have a size of kernel as $3 \times 3 \times 3 \times 16 \times 16$.

$$\sigma_{\mathrm{sigmoid}}(x) = \frac{1}{1 + e^{-x}}. \tag{12}$$

2.3.4 Improving robustness with additive noise

Adding artificial noise to the synthetic data for assessment is a key issue of the evaluation. There are many types of noise such as white noise, Gaussian noise, Poisson noise and salt-and-pepper noise. White noise and salt & pepper noise are discrete signals whose samples are regarded as a sequence of serially uncorrelated random variables. Normally, this type of noise will not significantly affect PIV related algorithms, which can be easily reduced by applying pre-processing filter e.g. median filter. Poisson noise is commonly in weak light illumination during imaging, which is not a major noise under laser illumination of PIV measurement. On the other hand, Poisson noise can normally be approximated by Gaussian noise. Hence, Gaussian noise is the most concerned type. Adding Gaussian noise to the dataset has been widely applied in many other seminal works [46, 47]. To investigate the robustness of AI-PR, we consider 20% of the total M training particle images biased with Gaussian noise. Different degrees of Gaussian noise is added to the four particle images. Following the typical way of adding noise [16, 21], the standard deviation σ of the image noise is calculated with levels of $n\sigma$ for PR testing, where n is from 0 to 0.2 with an interval of 0.05. It is noticed that the performance of the new algorithm is stable and accurate enough when the size of training data M is over 500.

However, it should be noted that calibration error also contributes significantly to the particle reconstruction. For the volumetric PIV calibration, one needs 'self-calibration'

[48] to significantly improve the accuracy of mapping functions and make the uncertainty down to 0.1 pixels, which is small enough to guarantee the accuracy of particle reconstruction. In practice, this step is performed before we apply the new proposed particle reconstruction algorithm. Therefore, we believe that error from calibration is extremely reduced and is decoupled with the noise from the raw particle images. Compared with the noise of particle images, the error of calibration after 'self-calibration' is negligible. For this reason, we did not consider the negative effect of calibration error in this study, but only focused on the noise from particle images, which is an acceptable approach for studying particle reconstruction [16, 47].

3 Results and discussions

3.1 Comparison setup

In this section, we briefly describe the performance of AI-PR against the traditional SF-MART method [6] in terms of reconstruction quality, computational efficiency and robustness to noise. Again, recall that it is difficult to obtain the true particle field in a real experiment. Hence, as a preliminary study, the comparison between AI-PR and SF-MART method is conducted on synthetic random particle field data. The testing particle fields are generated in the same manner as the training set but with a different size of $780 \times 780 \times 140$. Note that since AI-PR is trained on the sub-fields anyway instead of the original field, we divide the $780 \times 780 \times 140$ into sub-fields with size $64 \times 64 \times 32$ with some overlap. Seeding density in the generation of random particle fields varies from ppp = 0.05 to 0.3 with an interval of 0.05. Noise level ranges from n = 0.05 to 0.3 with an interval of 0.05. It is important to note that we only use the synthetic random particle field generated at ppp=0.2 with different noise levels for training AI-PR while the rest is for testing. Traditional PR methods: SF-MART with five and ten times iterations, together with the proposed AI-PR and its input as MLOS field are considered for comparison against each other. The codes for the training and testing on AI-PR are developed with Tenserflow$^{\mathrm{TM}}$ V1.13.1 [49] in Python (www.python.org) while MLOS and MART are developed with Matlab$^{®}$ (MathWorks, Inc.). The computer used is an Intel x99 workstation with one CPU of E5-2696 V4, 64GB DDR4 memory and a RTX2080ti graphics processing unit.

3.2 Comparison on cross-section of particle field

Figure 4 provides a central cross-section of a reconstructed particle field with ppp = 0.15 which is in the testing range. It is obvious that MLOS only gives a very coarse initial guess of potential particle location and intensity distribution, while AI-PR and SF-MART can recover better particle fields. Comparing further between AI-PR and MART methods, it is notable that SF-MART generates more ghost particles and has worse intensity distribution than AI-PR does. If the particle shape is looked closer, it can be found that MART-reconstructed particles have more ellipsoid shape, when AI-PR restores the spherical shape better.

3.3 Comparison on reconstruction quality, noise-robustness and computational efficiency

In terms of reconstruction quality, AI-PR shows its superiority to SF-MART methods as shown in Figs. 5 and 6. The quality factor Q defined in Eq. 13 is utilized for evaluating the accuracy and stability of the new technique, which is the correlation coefficient between the synthetic and reconstructed fields.

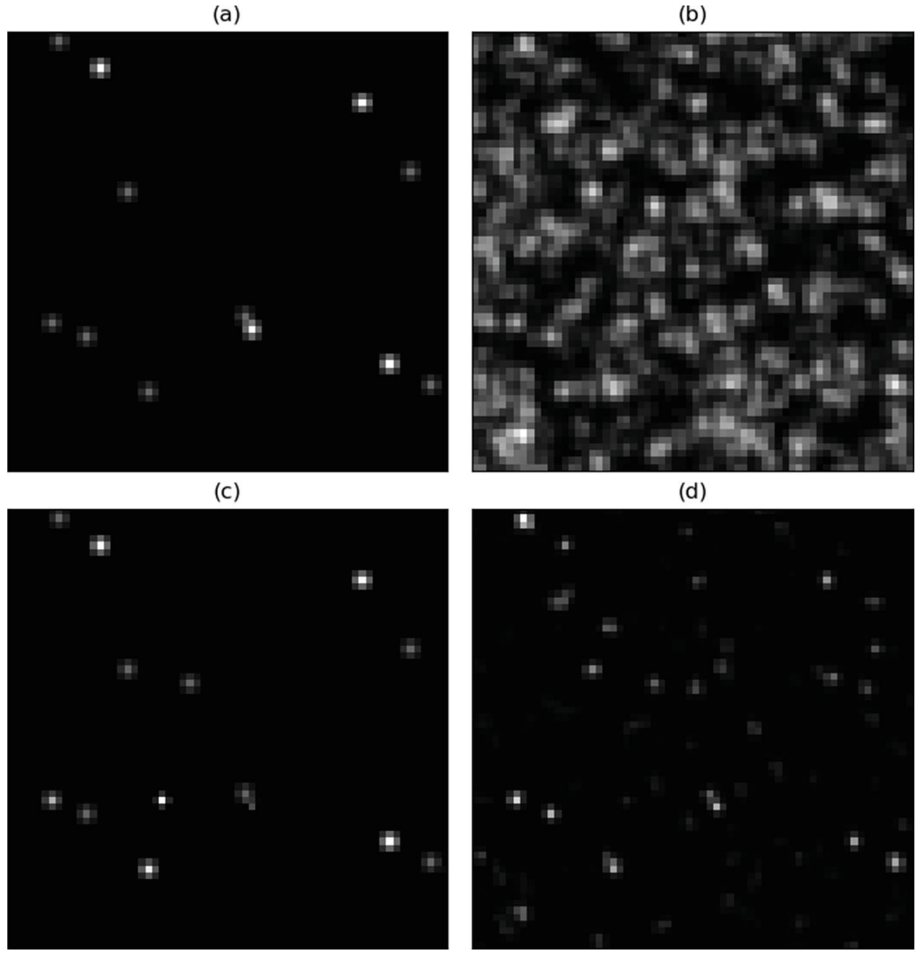

Fig. 4 Cross-sections of particle field, **a** Synthetic field, **b** MLOS field, **c** AI-PR, **d** SF-MART field with 10 iterations.

$$Q = \frac{\sum_{i=1}^{M} f(I_i, J_i, K_i) f_{\mathrm{CNN}}(I_i, J_i, K_i; \mathbf{W})}{\sqrt{\sum_{i=1}^{M} f^2(I_i, J_i, K_i)}\sqrt{\sum_{i=1}^{M} f_{\mathrm{CNN}}^2(I_i, J_i, K_i; \mathbf{W})}}, \tag{13}$$

In Fig. 5, all the methods are tested with varying particle density while without noise. It is shown that AI-PR can recover the particle with significant improvements from MLOS field. Reconstruction quality Q of AI-PR is much better than that of SF-MART methods. When ppp reaches 0.25, the Q remains at around 0.7 for AI-PR, while SF-MART-10 reduces below 0.6. Next, the effect of noise is parameterized in Fig. 6 at a fixed particle density ppp = 0.15, the Q reduces with the increase of noise level for all methods, but AI-PR has the best stability against the biases.

As shown in Table 1, the algorithms of MLOS, SF-MART-5, SF-MART-10 and AI-PR take wall-time as 512.5s, 5333.5s, 9881.5s, and 524.5s, respectively. Since AI-PR processing included the computing cost of MLOS and CNN, the actual computing time was only about 12s. However, it should be noted that although both SF-MART and the 3D CNN part in AI-PR share the same computational complexity, leveraging the state of the art GPU computing power, the latter can be highly parallel on GPU thus extremely fast while

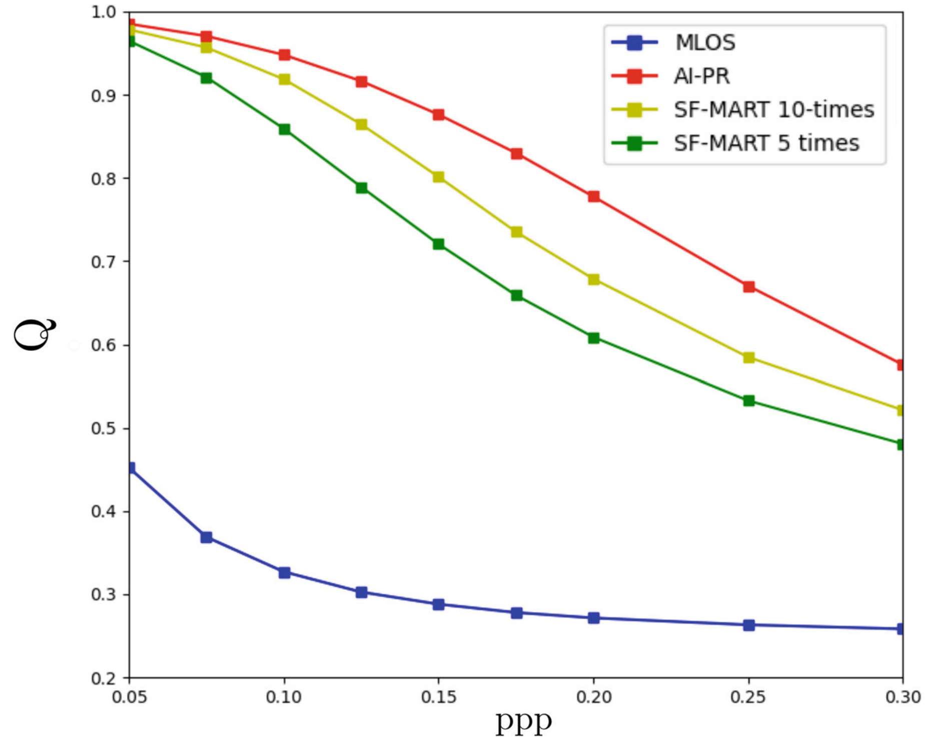

Fig. 5 Quality factor *Q* of different methods with varying seeding density from 0.05 to 0.3

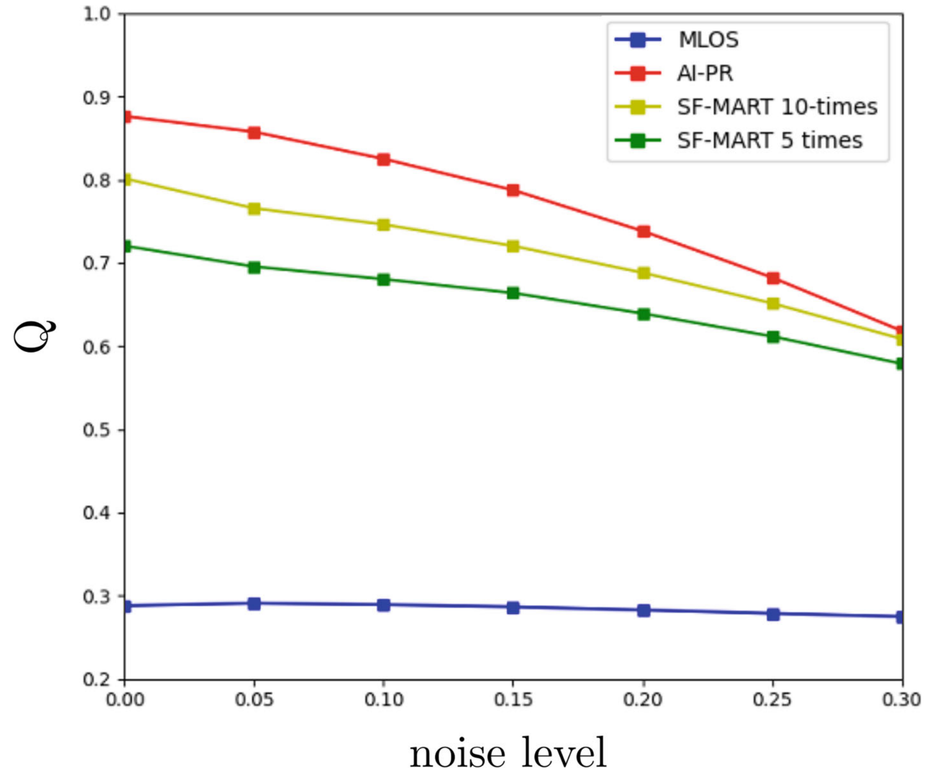

Fig. 6 Quality factor *Q* of different methods with varying noise levels from 0 to 0.3 with an interval of 0.05 at ppp=0.15

Table 1 Summary of time cost for particle reconstruction

	time (sec)
MLOS	512.5
SF-MART 5 iter.	5333.5
SF-MART 10 iter.	9881.5
AI-PR	524.5

the former requires iterative calculation with dependency among voxels/pixels. Moreover, it is noticed that the training of CNN costs about 16 hours for 100 epochs and the SF-MART algorithm is not accelerated with GPU in the current work.

4 Conclusions

Robust and efficient 3D particle reconstruction for volumetric PIV has been a long standing problem in experimental fluid mechanics. Traditional SF-MART-based algorithms either suffer from expensive computational time or sensitivity to noise. As a preliminary study, the newly proposed AI-based technique shows its superior advantages of accuracy, efficiency (x10 faster), and robustness to noise on recovering particle locations and intensities from 2D particle images over traditional SF-MART-based algorithms. Overall, with its superior accuracy and robustness, we believe AI-PR technique is very promising to apply to more realistic experiments by increasing the training dataset. However, as for the current work, the validation of our algorithm is limited to synthetic data rather than real experimental data. Future work should focus on combining calibration of volumetric PIV with AI-PR training, and performing particle reconstruction directly from AI-PR without calibration and additional network training for different real experimental cases.

Acknowledgements
Not applicable.

Authors' contributions
Conceptualization, Methodology: Q. Gao, S. Pan, R. Wei. Software, Simulation: H. Wang. Writing original draft, Q. Gao. Supervision, Funding acquisition: Q. Gao, J. Wang. Reviewing and Editing: S. Pan. The author(s) read and approved the final manuscript.

Funding
This work was supported by the National Key R & D Program of China (No. 2020YFA0405700), the National Natural Science Foundation of China (grant No. 11721202), the Program of State Key Laboratory of Marine Equipment (No. SKLMEA-K201910)

Availability of data and materials
The data and materials are available upon request.

Declarations

Competing interests
The authors declare that they have no competing interests.

Author details
[1]School of Aeronautics and Astronautics, Zhejiang University, Hangzhou, China. [2]Department of Aerospace Engineering, University of Michigan, 48105 Ann Arbor, United States. [3]State Key Laboratory of Nonlinear Mechanics, Institute of Mechanics, Chinese Academy of Sciences, Beijing, China. [4]School of Engineering Science, University of Chinese Academy of Sciences, Beijing, China. [5]MicroVec. Inc., Beijing, China. [6]Key Laboratory of Fluid Mechanics of Ministry of Education, Beihang University, Beijing, China.

Received: 6 July 2021 Accepted: 1 September 2021
Published online: 23 September 2021

References

1. Wang W, Tang T, Zhang Q, Wang X, An Z, Tong T, Li Z (2020) Effect of water injection on the cavitation control: experiments on a NACA66 (MOD) hydrofoil. Acta Mech Sinica 36(5):999–1017
2. Hong J, Abraham A (2020) Snow-powered research on utility-scale wind turbine flows. Acta Mech Sinica 36(2):339–355
3. Elsinga GE, Scarano F, Wieneke B, van Oudheusden BW (2006) Tomographic particle image velocimetry. Exp Fluids 41:933–947
4. Scarano F (2012) Tomographic PIV: principles and practice. Meas Sci Technol 24(1):012001
5. Gao Q, Wang H, Shen G (2013) Review on development of volumetric particle image velocimetry. Chin Sci Bull 58(36):4541–4556
6. Discetti S, Natale A, Astarita T (2013) Spatial filtering improved tomographic PIV. Exp Fluids 54(4):1505
7. Worth NA, Nickels TB (2008) Acceleration of Tomo-PIV by estimating the initial volume intensity distribution. Exp Fluids 45(5):847–856
8. Atkinson C, Soria J (2009) An efficient simultaneous reconstruction technique for tomographic particle image velocimetry. Exp Fluids 47(4):553–568
9. Elsinga GE, Tokgoz S (2014) Ghost hunting-an assessment of ghost particle detection and removal methods for tomographic-PIV. Meas Sci Technol 25(8):084004
10. de Silva CM, Baidya R, Marusic I (2013) Enhancing Tomo-PIV reconstruction quality by reducing ghost particles. Meas Sci Technol 24(2):024010
11. Schanz D, Schröder A, Gesemann S (2014) 'Shake The Box' - a 4D PTV algorithm: Accurate and ghostless reconstruction of Lagrangian tracks in densely seeded flows. In: 17th International Symposium on Applications of Laser Techniques to Fluid Mechanics, Lisbon, Portugal, 7-10 July 2014
12. Schanz D, Gesemann S, Schröder A (2016) Shake-the-box: Lagrangian particle tracking at high particle image densities. Exp Fluids 57(5):70
13. Wieneke B (2013) Iterative reconstruction of volumetric particle distribution. Meas Sci Technol 24(2):024008
14. Lynch KP, Scarano F (2015) An efficient and accurate approach to MTE-MART for time-resolved tomographic PIV. Exp Fluids 56(3):1–16
15. Novara M, Batenburg KJ, Scarano F (2010) Motion tracking-enhanced MART for tomographic PIV. Meas Sci Technol 21(3):035401
16. Wang H, Gao Q, Wei R, Wang J (2016) Intensity-enhanced mart for tomographic PIV. Exp Fluids 57(5):87
17. Gesemann S, Schanz D, Schröder A, Petra S, Schnörr C (2010) Recasting Tomo-PIV reconstruction as constrained and L1-regularized nonlinear least squares problem. In: 15th Int Symp on Applications of Laser Techniques to Fluid Mechanics, Lisbon, Portugal, 5-8 July 2010
18. Ye ZJ, Gao Q, Wang HP, Wei RJ, Wang JJ (2015) Dual-basis reconstruction techniques for tomographic PIV. Sci China Technol Sci 58(11):1963–1970
19. Bajpayee A, Techet AH (2017) Fast volume reconstruction for 3D PIV. Exp Fluids 58(8):95
20. Ben Salah R, Alata O, Tremblais B, Thomas L, David L (2018) Tomographic reconstruction of 3D objects using marked point process framework. J Math Imaging Vision 60(7):1132–1149
21. Cai S, Zhou S, Xu C, Gao Q (2019) Dense motion estimation of particle images via a convolutional neural network. Exp Fluids 60:1–16
22. Cai S, Liang J, Gao Q, Xu C, Wei R (2019) Particle image velocimetry based on a deep learning motion estimator. IEEE Trans Instrum Meas 69(6):3538–3554
23. Lagemann C, Lagemann K, Schröder W, Klaas M (2019) Deep artificial neural network architectures in PIV applications. In: 13th International Symposium on Particle Image Velocimetry, Munich, Germany, 22-24 July 2019
24. Liang J, Cai S, Xu C, Chu J (2020) Filtering enhanced tomographic PIV reconstruction based on deep neural networks. IET Cyber-Syst Robot 2(1):43–52
25. LeCun Y, Bengio Y, et al (1995) Convolutional networks for images, speech, and time series. Handb Brain Theory Neural Netw 3361(10):1995
26. Minerbo G (1979) Ment: A maximum entropy algorithm for reconstructing a source from projection data. Comput Graph Image Process 10(1):48–68
27. Guenther R, Kerber C, Killian E, Smith K, Wagner S (1974) Reconstruction of objects from radiographs and the location of brain tumors. Proc Natl Acad Sci 71(12):4884–4886
28. Huesman R (1977) The effects of a finite number of projection angles and finite lateral sampling of projections on the propagation of statistical errors in transverse section reconstruction. Phys Med Biol 22(3):511
29. Wang J, Yang Y, Mao J, Huang Z, Huang C, Xu W (2016) CNN-RNN: A unified framework for multi-label image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 2285–2294
30. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. pp 1097–1105
31. Liang M, Hu X (2015) Recurrent convolutional neural network for object recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 3367–3375
32. Milletari F, Navab N, Ahmadi S-A (2016) V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV), New York City. pp 565–571
33. McCann MT, Jin KH, Unser M (2017) Convolutional neural networks for inverse problems in imaging: A review. IEEE Signal Process Mag 34(6):85–95
34. Bhatnagar S, Afshar Y, Pan S, Duraisamy K, Kaushik S (2019) Prediction of aerodynamic flow fields using convolutional neural networks. Comput Mech 64:525–545
35. Guo X, Li W, Iorio F (2016) Convolutional neural networks for steady flow approximation. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York. pp 481–490
36. Lee K, Carlberg KT (2019) Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. J Comput Phys 404:108973
37. Goodfellow I, Bengio Y, Courville A (2016) Deep Learning. 3rd edn.. MIT press, Cambridge

38. Rawat W, Wang Z (2017) Deep convolutional neural networks for image classification: A comprehensive review. Neural Comput 29(9):2352–2449
39. Aloysius N, Geetha M (2017) A review on deep convolutional neural networks. In: 2017 International Conference on Communication and Signal Processing (ICCSP), New York City. pp 0588–0592
40. Zhiqiang W, Jun L (2017) A review of object detection based on convolutional neural network. In: 2017 36th Chinese Control Conference (CCC), New York City. pp 11104–11109
41. Baxes GA (1994) Digital Image Processing: Principles and Applications. Wiley, New York
42. Gonda F, Wei D, Parag T, Pfister H (2018) Parallel separable 3D convolution for video and volumetric data understanding. arXiv preprint arXiv:1809.04096
43. Jin P, Ginsburg B, Keutzer K (2018) Spatially parallel convolutions. In: 2018 ICLR workshop
44. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167
45. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980
46. Cai S, Liang J, Gao Q, Xu C, Wei R (2019) Particle image velocimetry based on a deep learning motion estimator. IEEE Trans Instrum Meas 69(6):3538–3554
47. Discetti S, Natale A, Astarita T (2013) Spatial filtering improved tomographic PIV. Exp Fluids 54(4):1–13
48. Wieneke B (2008) Volume self-calibration for 3D particle image velocimetry. Exp Fluids 45(4):549–556
49. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, Kudlur M, Levenberg J, Monga R, Moore S, Murray DG, Steiner B, Tucker P, Vasudevan V, Warden P, Wicke M, Yu Y, Zheng X (2016) Tensorflow: A system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). pp 265–283

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.