Experiments in Fluids (2020) 61:47
https://doi.org/10.1007/s00348-019-2875-2

RESEARCH ARTICLE

Introducing OpenLPT: new method of removing ghost particles and high-concentration particle shadow tracking

Shiyong Tan[1] · **Ashwanth Salibindla**[1] · **Ashik Ullah Mohammad Masuk**[1] · **Rui Ni**[1]

Received: 21 August 2019 / Revised: 20 December 2019 / Accepted: 22 December 2019

Abstract

We developed an open-source Lagrangian particle tracking (OpenLPT) based on the Shake-the-Box (Schanz, Gesemann, and Schröder, Exp. Fluids 57.5, 2016) method. The source code of OpenLPT is available on GitHub repository (@JHU-NI-LAB). The code features a new method that removes the majority of ghost particles at a high particle image density. The resulting percentage of ghost particles drops from 110% to 26% for image density at 0.125 ppp—nearly 84% of ghost particles are removed. Extensive tests of OpenLPT using synthetic data sets show that the code produces tracks with accuracy and processing time similar to the previously-reported values. In addition, OpenLPT has been parallelized to run on high-performance computing clusters to drastically increase its processing speed. To examine the code's capability of tracking shadows of small tracers for backlit experiments, the blurred-particle effect was also included on synthetic images and OpenLPT was tested to process these noisy images. The results show that OpenLPT can also track shadows of a high-concentration of particles reliably in 3D. Based on the test, the optimal depth of field (DoF) and particle concentration for future experiments using Lagrangian shadow tracking are provided. For example, DoF controlled by the aperture should be set at around half of the size of the view area. At this DoF, most particles in the interrogation volume can be tracked, whereas particles outside the interrogation volume become too dim to affect results. 40 experimental data sets for a wide range of particle concentrations were also used for evaluating the code, and the results show a nice agreement with the synthetic tests.

List of symbols

I^i_{part}	Intensity of the *i*th particle
C	Track coverage (Fig. 3)
F	Track fragmentation (Fig. 3)
Cr	Track correctness (Fig. 3)
ϵ_r	Mean position error for reconstructed raw tracks
T	Processing time per frame
DoF	Depth of field
CoC	Circle of confusion
d_{o}	Distance between an object and the lens
d_{i}	Distance between the image plane and the lens
f	Focal length of the lens
D_a	Aperture size
f_L	DoF/L
L	Size of the interrogation volume
M	Magnification ratio of the lens
d_p	Physical diameter of a particle
d_e	Diameter of the projected particle image
d_s	Diameter of a particle image due to diffraction
d_f	Diameter of a particle image due to the defocussing effect
z	Distance of a particle to the focal plane in the depth direction
L_n	Size of the noise zone
N_{t}	Number of all trackable particles
N_{tf}	Number of in-focus trackable particles
N_{tb}	Number of blurred trackable particles
N_{n}	Number of out-of-focus non-trackable particles
d	D for only trackable particles
D	Average diameter of all particles in pixels on images
ϕ	Total image density
ψ	Effective image density (only for trackable particles)
C_ϕ	Fraction of pixels occupied by all particles
C_ψ	Fraction of pixels occupied by trackable particles
R	C_ψ/C_ϕ

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s00348-019-2875-2) contains supplementary material, which is available to authorized users.

✉ Rui Ni
rui.ni@jhu.edu

[1] Department of Mechanical Engineering, Johns Hopkins University, Baltimore, USA

ϵ_s	Mean position error for smoothed tracks
e	Calibration error
Δ_ϵ	Mean distance between reconstructed tracks and filtered tracks
ϵ_Δ	Triangulation error
l	Average track length

1 Introduction

Multiphase flow is more ubiqitous in nature and in industrial applications than single-phase flow, as most flows are always seeded or contaminated with elements of a different phase, such as gas bubbles entrained in the ocean or oil droplets in chemical reactors. Understanding the interaction between two phases requires simultaneous measurements of two phases. However, in multiphase flow, sometimes two different phases have different or even conflicting requirements for measurements that have to be resolved first.

For the carrier phase, small-size neutrally buoyant particles are typically used as tracers. To view them clearly on cameras, laser illumination is favorable because it not only makes particles appear to be bright dots in front of a dark background but also helps to select only the particles in the volume of interest. However, for flows that contain bubbles or droplets, the interface between two phases is highly reflective, leaving shadows and light stripes in the view (Jordt et al. 2015). When the concentration of the second phase becomes higher, this problem will severely impact the ability to acquire accurate tracer data around bubbles or droplets. This issue has been resolved in three different ways: refractive index matching (RIM) (Budwig 1994), laser-induced fluorescence (LIF) (Dovichi et al. 1984), and shadow imaging (Zaruba et al. 2007). The RIM technique relies on a careful match of refractive indices of two phases, which is not suitable for bubbly flow. Although the laser-induced fluorescence (LIF) can help to distinguish bubbles from fluorescent particles. The possible reflection of tracer images via the bubble interface can cause large uncertainties of tracer velocity (Lindken and Merzkirch 2002). Finally, compared with all other methods, the back illumination can achieve a clear, sharp shadow projection of a bubble and is proven to be an economic and effective way to acquire bubble-phase features (Nishino et al. 2000).

The back illumination has also been used as an alternative to view tracer particles. Particles block light and their shadows appear on images in front a bright background. For example, particle shadow velocimetry (PSV) (Estevadeordal and Goss 2005) uses particle shadow to perform the PIV algorithm. Compared with the scattering light, in-line back illumination requires much lower light intensity. Time-resolved measurements at 10 kHz can be reached even with a simple LED panel. The technique of PSV was also successfully applied to the near-wall velocity measurements as a way to avoid surface glare encountered in the laser illumination (Goss et al. 2007). In addition to the single-phase measurements, PSV has also been attempted in the multiphase-flow measurements. Khodaparast et al. (2014) demonstrated the benefits of PSV in both gas–liquid and liquid–liquid flows, including a clear detection of both phases, the avoidance of strong reflection, as well as the low cost. Hessenkemper and Ziegenhein (2018) then extended the back illumination to the particle shadow tracking velocimetry (PSTV) in bubbly flows, and pointed out that PSTV is able to obtain more accurate velocity close to bubbles and handle images with a higher particle concentration than that in PSV.

For back illumination, the concern is always associated with particles outside the interrogation volume but still projected on images. This effect has been quantified using the idea of depth of correlation (DoC) defined as the depth within which the out-of-focus particles still contribute to the cross-correlation analysis for PSV (Meinhart et al. 2000). The expression of DoC was then derived by Olsen and Adrian (2000). Blurring particles introduces non-negligible noise to the determined 2D velocity field. In 3D systems, although the blurring particles may still pose challenges to particle identification and tracking, there is no systematical work to evaluate the feasibility and problems.

In this paper, PSV will be extended to the shadow-based 3D particle tracking. There are two different ways of pursuing it: the particle tracking velocimetry (PTV) (Papantoniou and Dracos 1989; Nishino et al. 1989) and the Lagrangian particle tracking (LPT) (Ouellette et al. 2006) method. There is a subtle but important difference between PTV and LPT: LPT intends to obtain long particle trajectories for studying particle dispersion, mixing, and other transport-related quantities, whereas PTV focuses on acquiring the Eulerian velocity field.

Recently, this distinction between these two methods becomes smeared by the advent of a technique called iterative particle reconstruction [IPR, Wieneke (2012)], which can iteratively optimize and locate the 3D position of particles. This method transforms the way that particle images are used and it helps one to identify individual particles for a particle concentration similar to that in a Tomographic PIV (Tomo-PIV) system without many ghost particles. With 3D positions, particles can be linked from frame to frame to construct trajectories, either using the minimal change of acceleration (Malik et al. 1993) or the four-frame best estimate (4BE) methods (Ouellette et al. 2006).

This idea of triangulation-then-tracking was changed by the recently-developed method called Shake-the-Box (STB) (Schanz et al. 2016). This method reversed the order of procedures into tracking-then-triangulation [first introduced by Attanasi et al. (2015)] using the existing tracks to predict the particle location in the next frame and use the predicted location to find the possible match in 3D. Making use of the temporal coherence of tracks STB outperforms the Tomo-PIV at a high image density (Kähler et al. (2016)). Because STB can obtain both the Eulerian and Lagrangian information, it has been extensively used in the acquisition of different types of flow statistics such as velocity (Schröder et al. 2015; Schneiders and Scarano 2016), vorticity, dissipation (Schneiders et al. 2017), pressure field (Van Gent et al. 2017; Huhn et al. 2018), and coherent structures (Schlueter-Kuck and Dabiri 2017; Huhn et al. 2017).

In our study, an open-source STB code, so-called open-source Lagrangian Particle Tracking (OpenLPT), has been developed and will be used for testing Lagrangian shadow tracking (LST) in our experimental setup. This paper will address the concerns of blurred particles and provide clear guidelines for future experimental setups.

2 OpenLPT and its performance evaluation

OpenLPT has been developed primarily to provide the community with an open-sourced STB (Schanz et al. 2016) that can accelerate the data processing on high-performance computing clusters efficiently. Sharing its source code will also help the community to have a common basis of code to conduct future method development and uncertainty quantification. In this paper, some key steps of OpenLPT will be introduced, and one particular step that helps to remove most ghost particles will be discussed in detail. Tests using synthetic datasets will also be presented and compared to the STB results (Schanz et al. 2016).

2.1 OpenLPT

Before running OpenLPT, camera images have to be preprocessed to remove noise, and for Lagrangian shadow tracking (LST) images have to be reversed so particles become bright in front of a dark background. The key steps of OpenLPT are briefly summarized in part A of Fig. 1. Within the part A, both the original images for the first four frames and the residual images will be processed by the iterative particle reconstruction (IPR) (part B). IPR runs the 3D particle reconstruction and refinement (3D-PR, part C) step iteratively. For each iteration, particles that have been triangulated from the previous 3D-PR will be removed, and the residual images are left for the next iteration. In 3D-PR, the main step includes finding 2D particle center by fitting a 1D Gaussian profile around the central pixel in each of the two directions (Ouellette et al. 2006). Triangulating these detected 2D particle positions results in a large list of matched candidates, many of which are ghost particles, especially when the particle image density is high. In OpenLPT, a new pruning method has been developed (highlighted box in part C) to remove the majority of these ghost particles; details of this step will be introduced in Sect. 2.2.

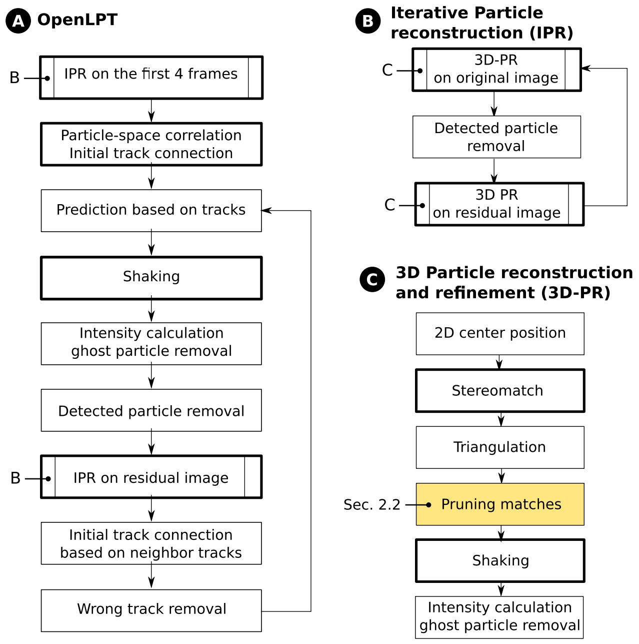

Fig. 1 Flow chart of the key steps of OpenLPT

One step that has been repeatedly used in both part A and C is the so-called shaking. In brief, shaking refers to the refinement of 3D particle positions. For every shake, the particle position in 3D will be modified in a small width and then projected onto all cameras. The difference between the projected image and the actual particle image is calculated, and the 3D position with the least difference summing over all pixels will be selected as the particle position. In the very beginning, this method starts with a shake width, which can be determined based on both the calibration error and particle acceleration. After each shake, the same process is repeated using a smaller and smaller shake width. Every new shake width is half of that used in the previous iteration. After shaking, intensity check, similar to the one introduced by Schanz et al. (2016), is performed to separate real particles from ghost ones by calculating the ratio between the projection and the actual image intensity averaged over all cameras. If this ratio drops below a set threshold, this candidate is believed to be a ghost particle and will be removed from the triangulation list. Particles passed the intensity check are then subtracted from images. The remaining particles in the residual images have a much lower effective image density; they will be processed by IPR (part B and C in Fig. 1) through multiple iterations so most of them can be recovered.

Particles from the first four frames are connected using the velocity field obtained by particle–space correlation

(Novara et al. 2016). From these frames, a number of short particle tracks will be established, and these short tracks will be fitted with the Wiener filter (Wiener 1949) to predict the locations of particles in the next frame. The predicted 3D position is "shaken" again locally to seek the closest candidate. This breaks the classical triangulation-then-tracking scheme used in most particle tracking methods. This new algorithm relies on the prediction from an existing track to find the 3D position of the particle in the next frame, i.e., tracking-then-triangulation. The key difference is that the tracking-then-triangulation method makes use of the temporal coherence of a track as an additional constraint, whereas the other scheme does not.

The same intensity check is then conducted for all particles to remove ghost particles. These connected particles will then be subtracted from images to acquire the residual images for IPR to detect more particles, These new particles will start their own tracks. All tracks are subjected to a linear-fit check (Schanz et al. 2016). Any track longer than four frames with strong position oscillation or a sudden deviation from the original track is treated as possible source of error and gets deleted in this step. This whole process is depicted in part A of Fig. 1. Finally, OpenLPT allows users to track particles from the last frame backward to further connect and extend the existing tracks.

The processing speed of OpenLPT was carefully examined and every step that requires many iterations, such as stereomatching, particle-space correlation, and shaking, is parallelized to optimize the code performance. The details of the improvements of OpenLPT can be found in Tan et al. (2019). OpenLPT is written in C++, and proven compatible on different operation systems and can be easily executed on high-performance computing cluster.

2.2 A new algorithm for removing ghost particles

For a high particle image density, particles will first be triangulated, and all matches will be put in a list. The size of this match list could be as large as 50 times of the number of 2D particles on images, which indicates that each particle image can be triangulated for several times. This produces a significant number of ghost particles.

The intensity check step that was proposed in IPR and STB can effectively remove some ghost particles. However, as shown by Schanz et al. (2016), in the first four frames, the remaining number of ghost particles can still be about 4.4 times larger than that of the real particles. These ghost matches will inevitably introduce uncertainties of velocity estimation, which will impact the following procedures. Here we introduce a new algorithm to remove most ghost particles right after triangulation, which also improves the detection of real ones.

The key of this new algorithm is to use two important conditions: (a) every 2D particle position can only be used once; and (b) the match needs to be optimal for all cameras. These two conditions are not always compatible. So the first condition is given a higher priority than the second. To demonstrate this method, an example is shown in Table 1. In this particular example, two particles are detected in each camera. But four matches are identified so only two are real and the remaining two are ghost matches. The code is designed to select the right match out of this list and remove the ghost particles.

OpenLPT goes through the entire match list twice. During the first time, it does not remove any match, only calculating the preference vector P. P marks the number of times that this match is selected after comparing with all other competing matches. P varies between 0 to the total number of cameras, four in our case. For example, match A has a smaller triangulation error ϵ_Δ for particle 1 in three cameras: II, III, and IV. But match A loses one competition to match B in camera I so the total P for Match A is 3. After repeating this calculation for all matches in the list, the algorithm will go through the entire list again. During the second time, P and ϵ_Δ are used to select the right matches and remove the rest. Note that between match A and B, match A is selected even though match B has a smaller ϵ_Δ because P is given a higher priority to optimize the results for all cameras. After match A is chosen, match B is removed from the list as we enforce the first condition that each 2D particle position can only be used once very strictly. Eventually, for this test case, match A and C are selected. It can be seen that this algorithm does not necessarily pick candidates with the smallest ϵ_Δ; it seeks the optimal results by accounting for all four cameras and enforcing one-particle-one-match criteria strictly.

As shown in Fig. 2, our advanced pruning algorithm is compared with the alternative method that only uses the 3D triangulation error with a threshold at 0.6 px. To show the difference, we plot the fraction of both real and ghost particles as a function of the image density. As the image

Table 1 An example for the algorithm to remove ghost particles

Cam ID	I	II	III	IV	ϵ_Δ (px)	P
Match A	1	1	1	1	0.1	3
Match B	1	2	2	2	0.08	1
Match C	2	2	2	2	0.05	4
Match D	2	1	1	1	0.15	0

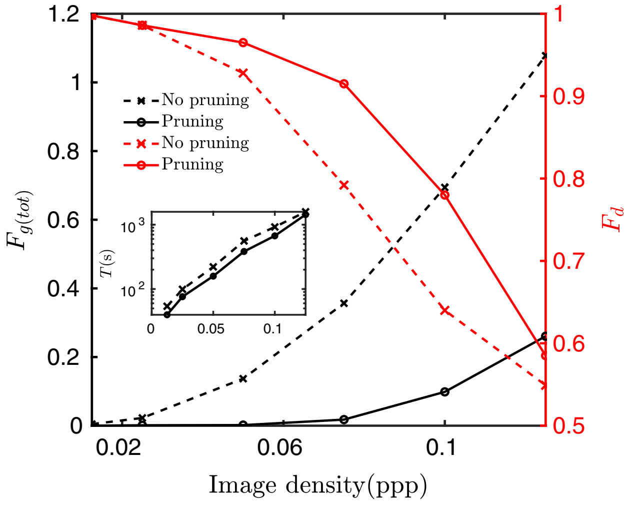

Fig. 2 Comparison of ratio of ghost and true particles, $F_{g(\mathrm{tot})}$ (left y axis), percentage of detected true particles, F_d (right y axis), IPR processing time per frame (inset plot) with and without pruning algorithm

density increases, the number of detected ghost particles (black lines) increases and the number of real particles (red lines) decreases for cases both with (solid lines) and without (dashed lines) our pruning method. However, our method shows a significant improvement as it systematically reduces the percentage of ghost particles. The percentage of ghost particles drops from 110 to 26% for image density at 0.125 ppp after implementing our pruning method.

In addition, it is interesting to observe that the number of detected real particles increases by at least 10% after implementing our algorithm. Through each iteration in IPR, fewer ghost particles result in fewer real particles getting incorrectly removed, and thus more real particles can be identified. Although 10% seems to be a small number, it is actually a significant improvement, as the probability of successfully identifying short tracks scales with $(N_t/N)^4$ in the first four frames. 10% more reconstructed real particles for one frame translates to over 46% more real tracks after four frames, which is crucial for estimating the velocity field for extending short tracks into long ones.

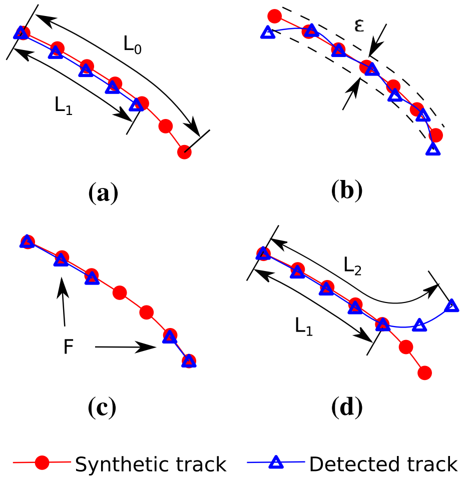

Fig. 3 Schematics of four key metrics to evaluate the quality of a track: **a** coverage $C = L_1/L_0$, **b** position error ϵ, **c** track fragmentation F, **d** correct connection $\mathrm{Cr} = L_1/L_2$

In addition, we also compare the processing time with or without our pruning method. As aforementioned, this step is computationally expensive, because (a) it requires going through a large list of matches twice to remove ghost particles; and (b) the code has to find out all matches that use repeated particle images in the first iteration for four cameras. Surprisingly, after we parallelized this step and implemented an algorithm to sort the list, the code becomes so efficient that the total processing time with this pruning step becomes even shorter than the one without. It indicates that the time that this pruning step saves by reducing the number of ghost particles is larger than how long this step takes. This makes this new algorithm an important improvement of IPR and STB methods.

We also compare the final results from OpenLPT with the STB method by Schanz et al. (2016). The details of the comparison can be found in the supplemental materials. One of the key findings is that the percentage of ghost particles in the first few frames is significantly lower than that in STB, i.e., OpenLPT starts with 26% ghost particles and original STB starts with 400% ghost particles.

2.3 Performance evaluation

OpenLPT is first tested with ideal synthetic data sets with zero noise and no blurring particles. The goal is to show that OpenLPT can work well with standard laser-illuminated experiments with relatively small image noise. We also want to demonstrate that the processing speed of the code is similar to the reported values (Schanz et al. 2016). Three sets of synthetic images with different image densities from 0.0125 particle-per-pixel (ppp) to 0.05 ppp were tested. The code can run at a much higher image density (as shown in the supplemental materials), but we did not push the density any higher in this section as it is challenging for the actual experiments to exceed 0.05 ppp. To demonstrate that the code can operate on different platforms, it was tested on both a standard desktop (Intel i7-8700 six cores at 3.2 GHz frequency) and the Maryland Advanced Research Computing Center (MARCC), a high-performance computing cluster (HPC) consisted of 23,000 cores, each with 2.6 GHz frequency.

2.3.1 Evaluation parameters

Figure 3 illustrates four typical errors induced by imperfect tracks, which will be calculated to evaluate the quality of tracking results from OpenLPT: (i) Coverage C defines the ratio between the number of particles detected by OpenLPT and the total number of particles within one track, L_1/L_0, as shown in Fig. 3a. (ii) Mean position error ϵ measures the deviation of the detected particle positions from the actual ones (Fig. 3b). Since each detected track will be processed with a Gaussian filter to remove noise, the position error is defined for both raw tracks (ϵ_r) and smoothed ones (ϵ_s). (iii) Track fragmentation F quantifies the number of detected segments composing of an original synthetic track (Fig. 3c). For the same coverage C, the larger F is, the more short tracks that one data set would have. (iv) Correct connection Cr defines the portion of a detected track that actually belongs to the same track in case that the code incorrectly connects two separate tracks into one.

2.3.2 Generation of synthetic images

Synthetic images were generated from an isotropic-turbulence data set from the Johns Hopkins Turbulence Databases (JHTDB) (Li et al. 2008). The adopted camera configuration is the same one used in experiments, which will be introduced in Sect. 5. The same data were down-sampled to different image densities at around 0.0125, 0.025, and 0.05 ppp. The details of tests for image density >0.1 ppp are shown in the supplemental materials. Each camera has a fixed resolution of 1024 × 1024 pixels.

To generate the synthetic images, a Gaussian intensity profile is used:

$$I^i_{\text{part}}(x_i, y_i, p) = ae^{-(bx'^2+cy'^2)} \tag{1}$$

where,

$$x' =(x_i - x_{ip}) \cos \alpha + (y_i - y_{ip}) \sin \alpha \tag{2}$$

$$y' = -(x_i - x_{ip}) \sin \alpha + (y_i - y_{ip}) \cos \alpha \tag{3}$$

(x_i, y_i) and $(x_{\text{ip}}, y_{\text{ip}})$ are the 2D positions of all pixels and only the center pixel of the ith particle on 2D images, respectively. For generating the ideal synthetic images, constants $a = 125$, $b = 1.5$, $c = 1.5$ and $\alpha = 0$ were used.

2.3.3 Evaluation results

A list of configuration parameters for applying OpenLPT is shown in Table 2. The evaluation results for OpenLPT are listed in Table 3. If the results were perfect, i.e., all particles can be tracked with no wrong or missing connections within the view volume, C, F, and Cr should all be one, and ϵ_r should be zero. For this ideal synthetic data, C is about 0.9991 even at the highest image density of 0.05 ppp, which is slightly better than the reported value of 0.9986 by Schanz et al. (2016), indicating that the code works almost perfectly for the ideal synthetic data set. The coverage varies very little from 0.9996 to 0.9991 as the image density is quadrupled, so OpenLPT probably has not reached its limit. As a result, other three metrics are all nearly perfect as well, with very weak dependence on the image density. In particular, Cr = 1 for all image densities because the code never incorrectly connects two tracks into one. The position error ϵ_r of raw tracks is about 0.006 px.

The processing time is important for evaluating the performance of the code, especially when a large amount of data needs to be processed. The typical processing time per frame for 12,500 particles is around 5 s for OpenLPT running on a standard desktop (Intel i7-8700 six-core 3.2 GHz processors). As reported by Schanz et al. (2016), 12,800 particles took a server (dual Intel Xeon E5-2680 ten-core processors at 2.8 GHz) around 8 s to analyze one frame. The

Table 2 The configuration parameters for tests with OpenLPT

Configuration for IPR	
Particle projection size	4 × 4 px
Number of total iterations	4(2)[1]
Number of iterations for shaking	6
2D intensity threshold[2]	30
Maximum 3D triangulation error	0.6 px
2D search radius for matching	0.5–2 px
Configuration for OpenLPT	
Shake width	0.5–0.025 px
Search radius without predictor	10 px
Search radius with predictor	0.5 px
Intensity threshold for identifying ghost particles	0.1 I_{avg}
Projection factor	1
Predictor for initialization	Particle–space correlation
Number of passes	1

[1] The number in brackets is the iteration times for reduced camera.

[2] The maximum intensity on image is 255

Table 3 Results of performance evaluation for the OpenLPT

Concentration (ppp)	0.0125	0.025	0.05
C	0.9996	0.9995	0.9991
ϵ_r (px)	0.006	0.006	0.0065
F	1.0309	1.0313	1.0020
Cr	1	1	1
T^1 (desktop, s)	5	13	40
T^1 (MARCC, s)	1.2	3.9	13.5

[1] The average processing time for one frame

processing time is similar, even though it is not conclusive since many conditions are not kept the same in two tests.

The same data sets were also processed by running OpenLPT on a high-performance computing cluster: MARCC. For our tests, the processing time per frame for the data set with 12,500 particles can be reduced to 1.2 s using 240 cores. Note that the computing power of individual processor is not fast as that in a desktop because MARCC has been built for over 4 years.

3 Blurring effect

In this section, we will focus on reproducing the blurring particle effect in synthetic data sets and use these data sets to test OpenLPT. Fig. 4 illustrates the ideal optical configuration, including one convex lens with the focal length of *f* and one aperture with the pinhole size of D_a. d_i and d_o are the image distance and object distance, respectively. One particle on the focal plane (point a on Fig. 4) appears to be focused, bright, and sharp (image intensity is inversed for a better illustration). Particles out of the focal plane (points b and c on Fig. 4) become blurred—the particle image size increases and the intensity decreases.

The largest acceptable size of a particle image is referred to as the circle of confusion (CoC). For a given optical setup, CoC can be translated to the depth of field (DoF), which is illustrated as green shaded area in Fig. 4. A particle within DoF appears on images with its size smaller than CoC, whereas the image of a particle outside DoF appears to be larger.

DoF is related to not only CoC but also the aperture size D_a, which can be approximated by (Allen and Triantaphillidou 2012):

$$\mathrm{DoF} \approx \frac{2d_\mathrm{o}^2 \cdot \mathrm{CoC}}{fD_\mathrm{a}} \tag{4}$$

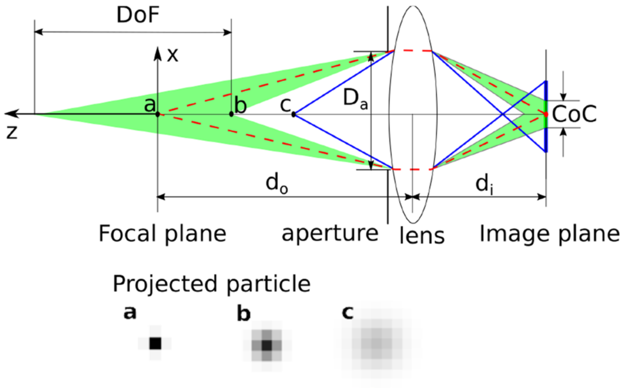

Fig. 4 Illustration of the optical setup to show different key components, including the depth of field and three particles (**a**–**c**) at different locations and their, respective images

It can be noted from Eq. 4 that DoF changes inversely with D_a. In experiments, D_a can be adjusted to obtain a desired DoF. In practice, each experiment will choose its own view area of size *L*, which ranges from micron meters for a micro-PIV system to meters for a large-scale PIV. Since different types of lens with different *f* are required in these applications, DoF tends to have a large range accordingly. For the rest of the discussion, instead of DoF, the dimensionless number $f_L = \mathrm{DoF}/L$ (*L* in our case is fixed at 40 mm) will be used to make our test results more general and useful for other experiments.

3.1 Synthetic data sets with blurred particles

To test LST with OpenLPT, the first step is to generate synthetic data sets, including the effect of blurred particles. The same turbulence data, as the one used in Sect. 2.3.2, was adopted here. The bounding box of this data set was scaled to match the interrogation volume ($40 \times 40 \times 40$ mm^3) in our experiments. For LST, as mentioned in Sect. 3, particles outside the interrogation volume may still appear on images. To reproduce this effect, particles were also put in a volume, marked by the dashed line in Fig. 6, outside the interrogation volume. This volume is about three times larger than the interrogation volume to make sure that the blurred-particle effects for any DoF can be faithfully accounted for in synthetic images. The particle concentration in this volume is kept the same as that in the interrogation volume.

The diameter of a projected particle image on cameras can be estimated by (Olsen and Adrian 2000):

$$d_e = (M^2 d_p^2 + d_s^2 + d_f^2)^{\frac{1}{2}} \tag{5}$$

where *M* is the magnification ratio of the lens, d_p is the diameter of a particle. d_s is the particle image size due to the diffraction effect. d_s is important for scattering light illumination (Ni et al. 2012), in which particles serve as the light source and the diffraction makes particles appear larger on images. For shadow tracking, this effect is less important and hence ignored in the following discussion. d_f is the particle image size due to the defocussing effect, and it can be related to (Rossi et al. 2012):

$$d_f = \frac{MD_a z}{d_\mathrm{o} + z} \tag{6}$$

where *z* is the distance of particles from the focal plane along the optical axis (see Fig. 4).

The intensity profile of the projected particle image is estimated by assuming that the integrated intensity over a particle image should be equal to the power of light blocked by the particle. The particle intensity can be derived as a

Gaussian function in a form similar to the one used for scattering light illumination (Olsen and Adrian 2000):

$$I(\mathbf{r}, z) = A\exp(-B\mathbf{r}^2) = \frac{c}{4\pi d_e^2(d_\mathrm{o}+z)^2}\exp\left(\frac{-4\beta^2\mathbf{r}^2}{d_e^2}\right) \quad (7)$$

where $c = J_p D_a^2 \beta^2$, J_p is the flux of light blocked by a particle, $\mathbf{r} = (x - x_p, y - y_p)$ is the pixel position relative to the particle center (x_p, y_p), and β^2 is an adjustable prefactor.

Rather than setting β^2 and J_p as some arbitrary constants, in our synthetic images, they are determined by making sure that particles on the focal plane will have the peak intensity at 255 ($A = c/(4\pi M^2 d_p^2 d_\mathrm{o}^2) = 255$ for 8-bit images) and size at 3–4 pixels. This essentially sets the highest image intensity and the smallest image size for a focused particle, and particles that are out of the focal plane will be adjusted based on Eq. 7.

The projection of particles throughout the entire volume can be calculated using Eqs. 4–7. In Fig. 5, both a synthetic image and an experiment image at the same image density of 0.016 ppp and $f_L = 0.45$ are shown. These two images are similar: focused particles are sharp and bright, whereas defocused particles are blurred and dim. This suggests that the calculation successfully reproduces some key features encountered in LST.

In Eq. 7, the peak intensity, indicated by the prefactor A, decreases as particles move away from the focal plane. For particles located outside the interrogation volume but with the peak intensity still above the threshold I_min, they will contribute to images as noise, so this range is referred to as the noise zone. Outside the noise zone, particle image intensity falls below I_min—they can no longer be seen or detected. By incorporating Eqs. 5, 6, and 7, the length of the noise zone, L_n can be derived as:

$$L_n = \frac{2}{d_p^2 + D_a^2}\sqrt{d_\mathrm{p}^4 d_\mathrm{o}^2 - \left(d_\mathrm{p}^2 d_\mathrm{o}^2 - \frac{c}{2\pi M^2 I_\mathrm{min}}\right)\left(d_\mathrm{p}^2 + D_\mathrm{a}^2\right)} - L \quad (8)$$

where L is the length of the interrogation volume. It can be seen that $L_n \propto 1/D_a^2$ roughly and DoF $\propto 1/D_a$ (Eq. 4). It indicates that, as one reduces the pinhole aperture size D_a to gain DoF, L_n increases much faster and significantly more blurred particles outside the interrogation volume will get projected on camera imaging planes.

As shown in Fig. 6, the volume can be divided into three regions, including DoF, interrogation volume, and noise zone. The number of particles within these three regions are: N_tf, N_tb, and N_n, respectively. Particles in the first two regions are trackable, but they do not appear as the same image size. The effective image size for the trackable particles, on average, is d, and the average image size for particles in all three regions is D ($D \geq d$). The average range of D/d in all tests is around 1–1.65.

To facilitate the discussion of applying OpenLPT on LST, it is important to introduce parameters similar to the image density that has been used to quantify the seeding density in 3D PIV community. The total or the effective image density, ϕ or ψ, are defined to measure the number of all particles or only the trackable ones divided by the total number of pixels. Since blurred particles do not have the same size as the focused ones, we also need a new parameter, image coverage, defined as the ratio of the number of pixels occupied by particles to the total number of pixels: $C_\phi = \phi D^2$ for all particles and $C_\psi = \psi d^2$ for only the trackable ones. In the extreme limit $C_\phi \approx 1$, all particles overlap with each other and no particle can be tracked. In addition, the ratio between two different image coverage $R = C_\psi / C_\phi$ evaluates the effect of blurred particles to the overall performance of the system.

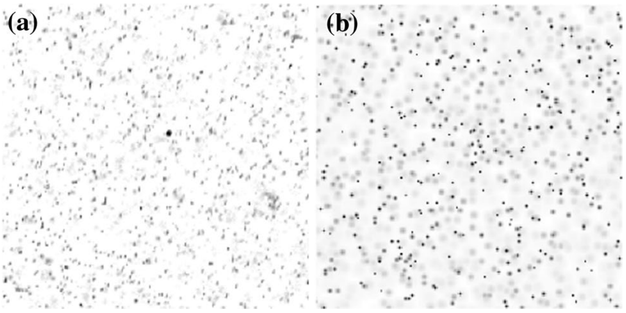

Fig. 5 Examples of **a** an experiment image and **b** a synthetic image at the same image density of 0.016 ppp and $f_L = 0.45$

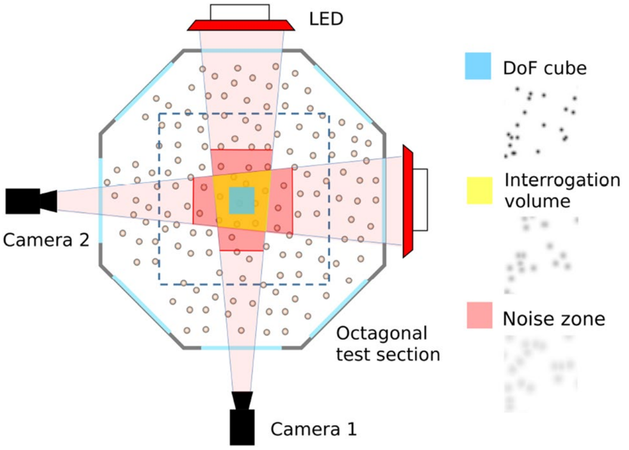

Fig. 6 Top view of three types of volumes in our setup, including (i) volume in focus (within DoF), (ii) interrogation volume, where particles can be tracked, and (iii) noise zone, in which particles are blurred and cannot be tracked

4 Applying OpenLPT on the synthetic data sets with blurred particles

In this section, we will discuss the results of two different tests for two different controlling parameters, the particle concentration and f_L. Particle concentration refers to the number of particles per unit volume, whereas the particle image density is used to quantify the number of particles per pixel on 2D images. In Sect. 4.1, the particle concentration will be fixed, and the effect of f_L on tracking results will be evaluated. In Sect. 4.2, f_L will be fixed at the optimal value and different particle concentrations will be examined. These two parameters are chosen because they directly impact the number of blurred particles, and they can also be controlled in our experiments. Therefore, understanding their relationship will help to seek an optimal set of parameters for future experiments.

4.1 Effect of f_L (fixed particle concentration)

For this test, a fixed number of 12,102 particles were seeded in the interrogation volume. As we mentioned before in Sect. 3.1, eight times more particles were positioned outside the interrogation volume to account for blurred-particle effect. All relevant parameters are included in Table 4.

Four different f_L are compared with each other and also with the laser-illumination case, in which particles outside the interrogation volume cannot be seen. The first section of the table shows the number of particles in three different regions N_{tf}, N_{tb}, and N_{n}. For the smallest $f_L = 0.25$, the blurred-particle effect induced by particles in the noise zone N_{n} is small, but only a small portion of particles in the interrogation volume are in focus. In the other limit for $f_L = 1$, all particles in the interrogation volume are focused. However, it pays the price of including significantly more N_{n}, around 85 times of that at $f_L = 0.25$. This rapid growth of N_{n} as a function of f_L is consistent with the discussion of DoF and L_n in Section 3.

Table 4 Statistics of trackable and blurred particles for different f_L at the same particle concentration

f_L	Laser	0.25	0.5	0.75	1
N_{tf}	11,975	4,031	8,073	10,950	11,975
N_{tb}	127	5,208	4,029	1,152	127
N_{n}	0	286	5,580	14,893	24,197
D (px)	2.78	4.59	4.52	4.59	4.6
d (px)	2.78	4.55	3.81	3.15	2.78
ϕ (ppp)	0.0115	0.009	0.0169	0.0257	0.0346
C_ϕ(%)	8.92	19.1	34.5	54.2	73.3
ψ (ppp)	0.0115	0.0088	0.0115	0.0115	0.0115
C_ψ(%)	8.92	18.2	16.8	11.4	8.92
R(%)	100	95	49	21	12

One possible concern for applying OpenLPT on LST is that particles may not have the same image size, potentially putting an important step, particle image projection, at risk. In the second section, it is interesting to note that the average diameter of all particles (D) remains almost the same at around 4.6, insensitive to different DoF. Although an increase of f_L reduces the image diameter of trackable particles, it also expands the noise zone—a greater number of blurred particles with larger diameter will appear on images. These two combined opposing effects lead to a similar averaged particle diameter D. The image size of only the trackable particles d indeed decreases as a function of f_L, but only from about 4.55–2.78 in a small range. This range of particle size is still within the limit that STB can handle by adjusting the OTF parameters.

The third section shows the image density and coverage. As f_L increases, the total image density increases rapidly because of the growing N_{n}. At $f_L = 1$, over 70% image is occupied by particles, likely resulting in a large probability of overlapping particle images. At the same time, it can be seen that the effective image coverage of only the trackable particles C_ψ decreases as f_L increases. When f_L becomes larger than 0.5, the total number of particles within the interrogation volume $N_t = N_{\mathrm{tf}} + N_{\mathrm{tb}}$ becomes a constant. But the effective image size of the trackable particles reduces. The combined effect makes C_ψ drop. Finally, the ratio R between the two image coverages decreases significantly as f_L increases, implying that a large f_L may not be preferable in LST measurements. Note that this requirement is opposite to that in a laser-illuminated system, in which a large f_L is preferable so all particles within the interrogation volume are focused.

Results of a randomly-picked track at two different $f_L = 0.25$ and 1 are shown in Fig. 7. Alongside these two cases, the original synthetic track (red) and the tracking data with no blurred particles (green diamond) are also shown. First of all, the test case with no blurred particle shows an excellent match with the raw synthetic data, indicating that our OpenLPT code performs as expected. After introducing blurred particles, some position variations start to appear. The deviation does not seem to depend on f_L for the range of f_L considered.

In Fig. 7b, the position data is processed to obtain the particle velocity by applying either the Gaussian kernel (Mordant et al. 2004; Ni et al. 2012) or the B-spline method (Gesemann 2015) to raw tracks. Here, we choose to use the Gaussian kernel filter. As shown in Fig. 7b, the filtered velocity ($f_L = 0.25$ and 1), indicated by the green and blue symbols, is very close to the simulation results and noise-free test case. The difference between the

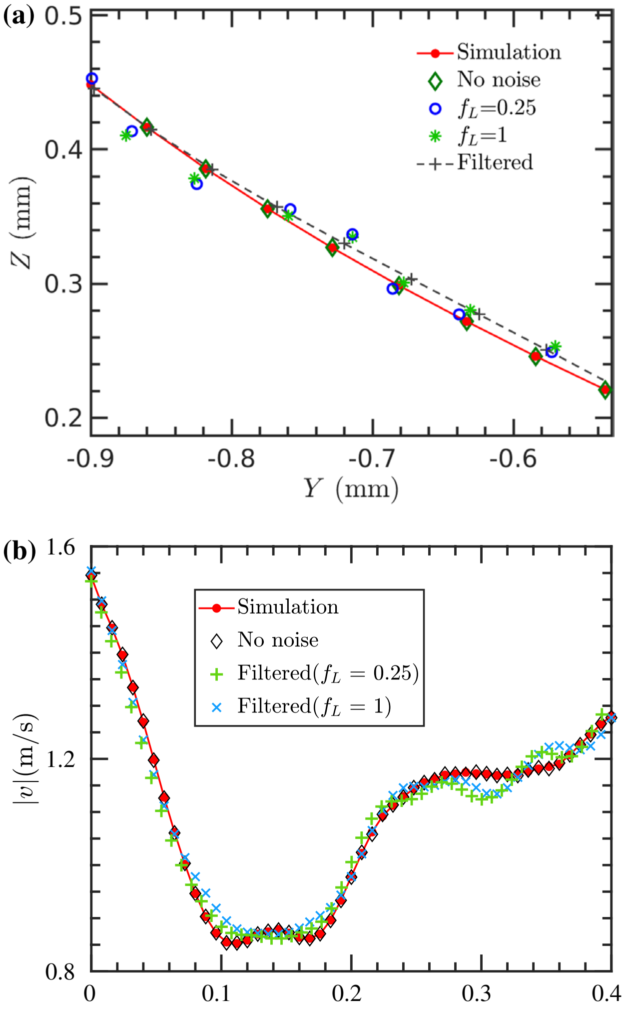

Fig. 7 A comparison of the same track, including both **a** the position on the Y–Z plane (at X = 0) and **b** the velocity magnitude along the Y direction, from data sets synthesized with different optical parameters

calculated velocity to the actual velocity from simulation is about 7.2% of the velocity magnitude in this test case.

Note that both the position and the calculated velocity of the sample track at $f_L = 0.25$ and one exhibit very similar variations, suggesting that the accuracy of the tracked particle position may not depend much on f_L, even though a larger f_L brings in more blurred particles.

Table 5 compares ϵ_r (raw tracks) and ϵ_s (smoothed tracks) averaged over all tracks for the same synthetic data set tested at different f_L. In consistent with the observation in Fig. 7, there is no trend of either ϵ_r or ϵ_s on f_L. In fact, these two parameters are almost constant, suggesting that the track position error is not very sensitive to f_L. Moreover, the relatively large position error due to the presence of blurred particles reduces by half after filtering, which is consistent with the previous finding (Schanz et al. 2016).

Table 5 Position error of unfiltered and filtered tracks

f_L	0.25	0.375	0.5	0.75	1
ϵ_r (px)	0.59	0.52	0.56	0.56	0.50
ϵ_s (px)	0.28	0.28	0.28	0.28	0.28

The effects of f_L on other tracking quality parameters, C, T, F, and Cr (introduced in Sect. 2.3) are illustrated in Fig. 8. For coverage C, at around $f_L = 0.5$, C reaches its maximum close to one, indicating that most tracks in the interrogation volume can be successfully detected at this f_L. For f_L smaller than 0.5, many particles even in the interrogation volume become too blurred to be seen (see Table 4 for details). Missing tracks ($C \ll 1$) is due to, rather than the tracking algorithm, the loss of particles on images. For $f_L > 0.5$, all particles in the interrogation volume can be detected and tracked, but the growing number of blurred particles outside the interrogation volume starts to increase the noise level. However, this effect is not significant because the coverage drops only by 2% as the percentage of blurred particles on images increases from 32 to 67%.

For track fragmentation F and correct connection Cr, both of them approach 1 as f_L increases, suggesting fewer fragmented and more accurate tracks even with more blurred particles appearing on images. This is a little couterintuitive as it seems that adding noise does not affect the result even help to improve it. In fact, this test involves two competing effects. Although increasing f_L will introduce more blurred particles, it also makes more particles in the interrogation volume focused. Since the contrast between the focused bright particles and the blurred dim particles is large for $f_L > 0.5$, OpenLPT has no problem identifying the focused particles and the tracks remain long and intact.

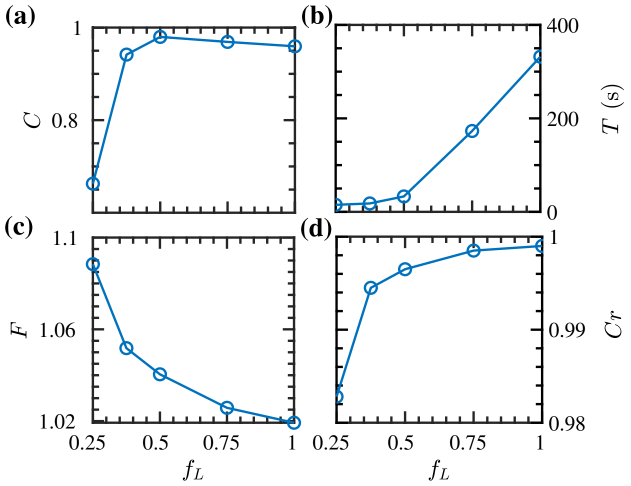

Fig. 8 The dependence of **a** coverage C, **b** processing time T, **c** track fragmentation F, and **d** correct connection Cr on the normalized depth-of-field f_L

Note that both F and Cr do not change much in the range of f_L considered. So they are not the dominant factors to be considered for designing experiments. We also compare the dependence of the processing time on f_L. As expected, the increasing number of blurred particles significantly lengthens the processing time, especially for $f_L > 0.5$.

In sum, the general principle of designing an experiment using OpenLPT for particle shadow tracking is to set DoF smaller than the size of the interrogation volume, ideally a half of it. At this value, it is a good balance between the quality of the tracking results and the total processing time. The noise level remains low and the results are comparable to that in a laser-illuminated case.

4.2 Effect of particle concentration (fixed $f_L = 0.5$)

The key benefit of OpenLPT and STB is to track a high concentration of particles. In this section, we want to test if the presence of blurred particles will affect that capability of OpenLPT. For this test, f_L is fixed at the optimal value at 0.5, so the noise zone length L_n is also a constant. As we increase the number of particles in the interrogation volume, the blurred particles outside increases accordingly to keep a constant particle volume concentration. The ratio between the number of trackable particles and blurred particles is around 2.16, as shown in Table 4.

The maximum total number of particles inside the interrogation volume is set at 27,500, which corresponds to $\phi = 0.038$ ppp and $C_\phi = 78\%$. The effective image density is $\psi = 0.026$ ppp, which is lower than the upper limit of 0.08 ppp for some recent laser-illuminated experiments (Schanz et al. 2016). Nevertheless, after the tests, it seems that we have not reached the maximum particle concentration that OpenLPT can handle. But we capped at this number to be consistent with experiments.

Figure 9 shows that the number of detected particles by OpenLPT, N_d, and the track coverage, C, as a function of the total number of particles in the interrogation volume $N_t = N_{\mathrm{tf}} + N_{\mathrm{tb}}$. N_d monotonically increases with N_t linearly, and it does not seem to deviate from the linear relationship for the range of N_t considered. N_d can be fitted with the number of particles on an image N_ϕ (or N_t): $N_d = 0.52N_\phi + 780$, which will be used to compare with the experimental results later. In addition, the black dashed line indicates an ideal case that all particles on images can be tracked, i.e., $N_d = N_t$. Although the data seems to deviate from this ideal line gradually as N_t grows, the deviation remains small compared to the total number of particles, and it did not show any abrupt transition at a large N_t. This suggests that N_t has not reached the limit in the synthetic data.

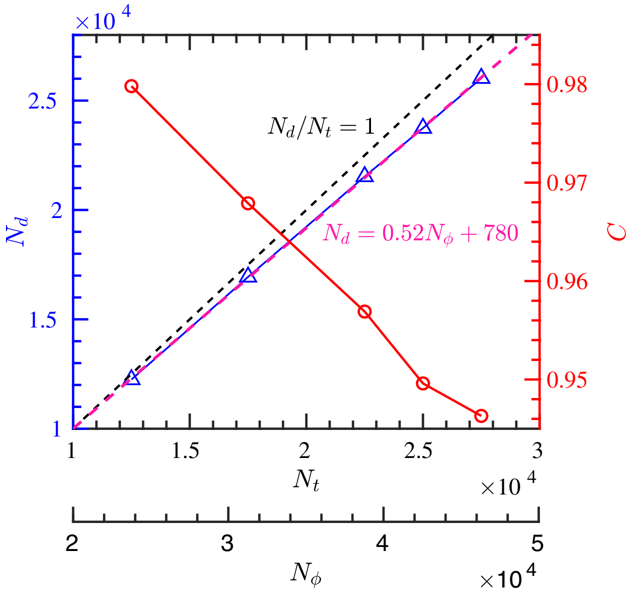

Fig. 9 The number of detected particles N_d and the coverage C as functions of the particle concentration N_t or the number of particles on an image N_ϕ

Table 6 Processing time, position error, track fragmentation and correct connection of tracks obtained from images at different particle concentration

N_t	12,500	17,500	20,000	25,000	27,500
T (s)	33	60	143	171	138
ϵ (raw, px)	0.52	0.52	0.52	0.56	0.51
ϵ (filtered, px)	0.26	0.28	0.28	0.28	0.28
F	1.0404	1.0398	1.0386	1.0416	1.0389
Cr	0.9965	0.9958	0.9952	0.9940	0.9949

The track coverage reduces by around 2% for every increase of 10,000 particles in N_t, whereas it drops only by 0.08% for the same increase of N_t in the zero-noise test case (see Table 3 for details). 2% is still very insiginificant change of coverage as nearly 95% of tracks can be covered at $N_t = 27,500$. This shows that applying OpenLPT on LST does not generate uncertainties as large as one may think. In fact, the quality of tracks is comparable to that from the laser-illuminated cases.

Other parameters such as processing time, position error, track fragmentation, and correct connection are listed in Table 6. As expected, the processing time increases as N_t grows. However, all other parameters do not depend on N_t. This is promising as the increased number of blurred particles do not necessarily affect the quality of tracks at all for the range of concentration considered.

This suggests that we can simply choose the highest effective image density at $\psi = 0.026$ ppp for our experiments.

5 Experiment results

5.1 Experimental setup

All experiments were conducted in an experimental facility, V-ONSET (Vertical Octagonal Non-corrosive Stirred Energetic Turbulence) (Masuk et al. 2019), as shown in Fig. 10a. This vertical water tunnel was built to study turbulent multiphase flow at high energy dissipation rates, specifically targeting deformable bubbles/droplets interacting with surrounding flow.

The octagonal test section (top view: Fig. 10b and side view: Fig. 10a,c) is chosen to allow optical access from multiple locations covering the entire perimeter. This is designed primarily for reconstructing complex 3D shapes of bubbles and droplets, but it can also be used for LST. Six high-speed cameras were used for experiments. In this paper, only four cameras were used for testing OpenLPT. The camera configuration is shown in Fig. 10b, c, and four LED panels were dedicated for each one of the four cameras to provide the in-line back illumination. On top of the test section, an unique 3D-printed jet array is designed to merge the mean flow coaxially with 88 independently-controlled jets. High-speed jets allow us to vary the turbulence characteristics in the test section. The details of this facility can be found in Masuk et al. (2019).

5.2 Experimental procedure

Before conducting tests, OpenLPT is very sensitive to the camera calibration. Wieneke (2018) pointed out that the calibration error needs to be controlled within 0.4 pixel for iterative particle reconstruction (IPR). To achieve this goal, a calibration procedure was developed:

1. By imaging a common calibration target, rough estimate of the camera parameters can be acquired using different calibration models. For our code, we chose to use the pinhole Tsai model (Tsai 1987).
2. The calibration results can be improved using the Volume Self-calibration method (VSC). In distinction from the existing VSC (Wieneke 2008), our method is a simple non-linear optimization method to minimize the triangulation error of the chosen particles. For this particular step, a low-concentration of tracer particles (< 0.001 ppp) uniformly distributed in the volume are used. This step allows a large search radius to find all matches, which increases the possibility to find a globally-optimized set of parameters.
3. Finally, the actual experimental data set with a high particle concentration is used for VSC. For both this step and the previous step, we select particles that have long trajectories and reduce the number in some overpopulated regions to ensure a uniform concentration across the entire volume.

5.3 Data analysis

Forty experimental data sets covering a range of particle image density from 0.005 to 0.022 ppp have been collected to evaluate the performance of OpenLPT. Each data set contains 21,800 frames collected at 4000 fps. For all

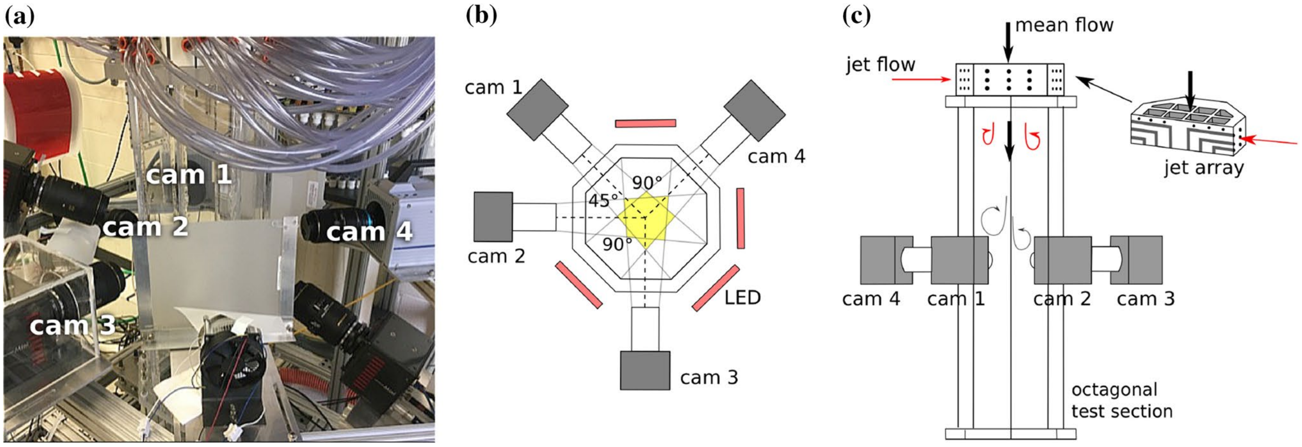

Fig. 10 **a** Picture of test section and camera configuration, **b** top view of all cameras and their positions and orientations, **c** side view of the water tunnel and cutaway view of the jet array experiments, an effective focal length of the lens is around 188 mm, and the aperture size of 8.55 mm was kept constant for all experiments. Based on these parameters and the circle of confusion at four pixels (also used for the synthetic data in Sect. 3), the calculated DoF was about 22 mm, based on Eq. 4. The size of the interrogation volume L is roughly 49 mm, and thus f_L is 0.45, which is close to the optimal value of 0.5 obtained in the synthetic tests. Other experimental configurations were presented in Sect. 5.1, and OpenLPT is applied to processing the experimental data on MARCC.

Trajectories spanning 50 frames from one data set are shown in Fig. 11 with each track color coded by its instantaneous velocity magnitude. Each trajectory was filtered by a Gaussian kernel (Mordant et al. 2004), and the filter length and width are determined to remove noise contained in the acceleration variance (Ni et al. 2012). The difference between the filtered and raw positions is used to quantify the position error ϵ. For this test data set, the average ϵ is about 0.01 mm, which is only half of the calibration error and one order-of-magnitude smaller than the average particle displacement of 0.15 mm between two consecutive frames, indicating that the blurred particles have a weak impact on the triangulation error and the tracking results are reliable at this f_L.

The relationship between the average number of detected 3D particles N_d per frame and the total number of particles on each image N_ϕ is shown as closed blue circles in Fig. 12. The data are scattered probably because of the varying calibration error. Since the non-linear optimization used in VSC depends on the initial parameters and the particles used, the VSC results are slightly different for every high-concentration data set. The calibration error, e, for each data set is shown in Fig. 12. As N_ϕ increases, there is little change of e, implying a very weak dependence of e, if at all, with the total number of particles. This observation is consistent with our expectation, as the entire two-steps VSC (low and high concentrations) is designed to make sure the particles selected for calibration are accurate and uniformly distributed over the entire volume, and most important, not sensitive to the concentration.

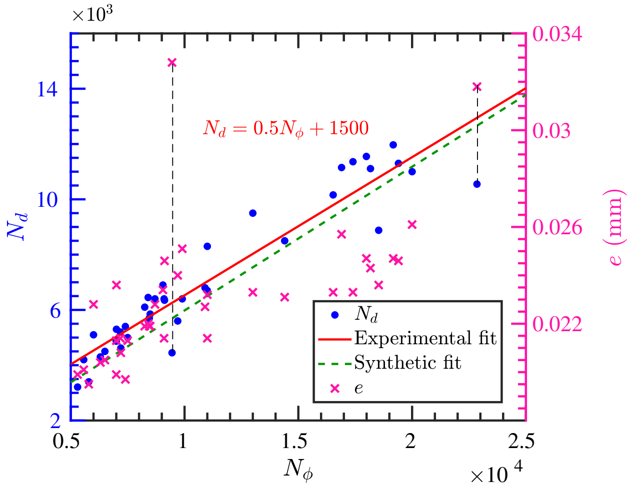

Fig. 12 The number of detected particles N_d (left axis) and the calibration error e (right axis) as functions of the total number of particles on each image N_ϕ

Nevertheless, we occasionally will have data sets that tend to have a large e even after VSC. Two cases are highlighted by two vertical dashed lines, which help to connect high e to their corresponding N_d. It can be seen that, when e is large, the number of particles detected becomes smaller because not all particles in the volume can be identified.

Despite the variations, N_d increases monotonically with N_ϕ, and this relationship can be fitted with a linear relationship, $N_d = 0.5N_\phi + 1500$, as shown by the red line. The green dashed line is acquired from the synthetic data set

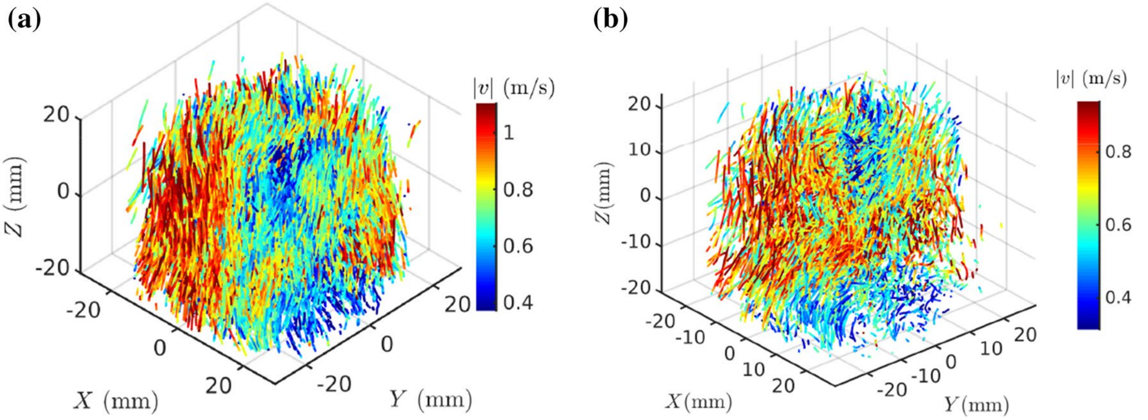

Fig. 11 Sample trajectories from one experimental data **a** with and **b** without the mean flow (the actual particle size is only about 1/15 of the line width). Each track is color coded with the instantaneous velocity magnitude

that has already been shown in Fig. 12. It is quite surprising that the red line and the green dashed line agree to each other extremely well, despite the clear distinction between the experimental data and the synthetic data. For example, the synthetic data set does not include many effects, such as the optical aberration, calibration error, non-uniform light intensity, and Gaussian random noise on images. The only effect that was included in the synthetic data set is the blurred-particle effect. The nice agreement between two data sets implies that the blurred-particle effect is the strongest noise in LST, and the way that the synthetic data were generated successfully captured that. At $f_L = 0.5$, only half of the visible particles on images are actually trackable. The slope of the experimental data is a perfect 0.5, which indicates that most of the trackable particles have been successfully identified by the code despite the noise created by the other half. This powerful technique suggests that the code can deal with around 13,000 particles using 1M pixels cameras very easily, and the limit of our OpenLPT coupled with LST has not been reached.

Finally, the quality of trajectories was examined by determining two different metrics: (i) the position error $\Delta\epsilon = \epsilon_r - \epsilon_s$ between the raw and filtered tracks (for details, see Sect. 2.3.1) and (ii) the average trajectory length l. For experiments, since it is impossible to know the true particle positions in a track *a priori*, the filtered track is used as a reference to calculate $\Delta\epsilon$, which essentially assumes that most position noise is random and can be removed by a low-pass filter efficiently. This assumption has been supported by our tests using the synthetic data set, as shown in Fig. 7.

$\Delta\epsilon$ seems to grow gradually as N_ϕ increases, whereas the synthetic data suggest that ϵ does not change with N_ϕ (Table 6). This difference could be a result of either the variation of calibration error e in experiments or other types of noise that was not included in the synthetic data. To test which one is the dominant effect, $\Delta\epsilon$ is also plotted against the calibration error e, as shown by the red cross in Fig. 13. The data show a similar trend and even a similar level of variations, suggesting that the observed trend is likely to be a result of the variations of the calibration error e in experiments.

Fig. 13b shows the dependence of the average track length l with N_ϕ. Again, similar to the discussions on $\Delta\epsilon$, the scatter of the data is clear and probably also because of the calibration error. The decrease of the mean track length as N_ϕ increases is very clear, which is not surprising as the track length is always sensitive to noise. Such an adverse effect can hardly be seen from the results in the synthetic cases in Sect. 4, which suggests that this may not be due to the noise induced by the blurred particles but other types due to non-uniform light intensity or random noise. One way to solve this problem is to use the postprocessing step to splice this broken tracks into longer ones using the six-dimension matching algorithm (Xu 2008) that has been developed before for Lagrangian particle tracking.

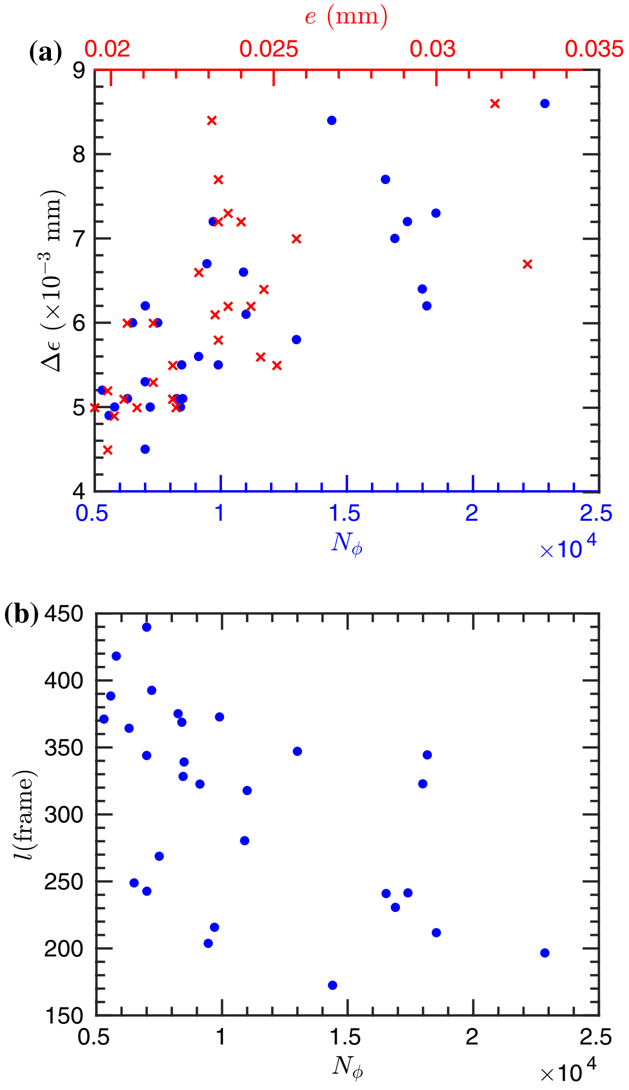

Fig. 13 **a** Position error between the raw and filtered tracks, $\Delta\epsilon$, as a function of the total number of particles on each image N_ϕ and calibration error e, **b** The averaged length of tracks l (frames) as a function of N_ϕ

6 Conclusion

In this paper, we first introduce the algorithm and structure of OpenLPT. Although it is based on the shake-the-box method by Schanz et al. (2016), we improved the part how ghost particles are identified and removed. At the highest particle image density at 0.125 ppp, the new algorithm reduces the ghost particle from 110% to 26%. This was achieved based on a new pruning algorithm that optimizes the match list so each 2D particle image can only be used once and the match that is ideal in most cameras is selected. This step not only reduces the number of ghost particles and increases the number of detected true particles; it also shortens the processing time. OpenLPT has also been tested extensively on multiple synthetic data sets with a range of image densities from 0.0125 to 0.125 ppp. The details of the comparison are shown in the supplemental materials. In addition, one of the key motivations of developing OpenLPT is to parallelize the code for running on high-performance computing (HPC) clusters. In this paper, we clearly show that running on HPC can speed up the process significantly.

The second part of the paper is to test the performance of the method on particle shadow tracking. Images with blurring particles were synthesized and processed using OpenLPT. From the calculation, it becomes obvious that, as the aperture size drops, the size of the noise zone expands faster than DoF, and more blurred particles outside the interrogation volume start to appear on images.

Tests with blurring particles were carried out at different DoF and a range of particle concentrations. As DoF increases, although the track qualities, including track fragmentation and correctness, have improved as particles in the interrogation volume become more focused; the number of blurred particles increase substantially—processing time soars and the track coverage starts to drop for $f_L > 0.5$. As a result, we recommend future experiments use $f_L = 0.5$ for a balance between the processing speed and accuracy for OpenLPT or other 3D particle shadow tracking methods.

Finally, following the lesson learned from the synthetic data, experiments with $f_L = 0.45$ (close to the optimal value of $f_L = 0.5$) and $L = 49$ mm were adopted for tests. Results of the detected particles as a function of the total number of particles on images show an excellent agreement with the synthetic data sets, despite a scatter of the experimental data due to the calibration uncertainty. This test also suggests that OpenLPT can track shadows of a high concentration of particles, and the blurring-particle problem can be managed as long as one pays attention to the selection of the aperture size.

Acknowledgements Acknowledgment is made to the Donors of the American Chemical Society Petroleum Research Fund for partial support of this research. This study was also financially supported by National Science Foundation under the Award Numbers: 1705246 and CAREER-1653389. Partial support is provided by the Oak Ridge Institute for Science and Education (ORISE) professorship to Rui Ni. The authors also thank Daniel Schanz, from German Aerospace Center (DLR), for his valuable suggestions.

References

Allen E, Triantaphillidou S (2012) The manual of photography and digital imaging. Focal Press

Attanasi A, Cavagna A, Del Castello L, Giardina I, Jelić A, Melillo S, Parisi L, Pellacini F, Shen E, Silvestri E et al (2015) Greta-a novel global and recursive tracking algorithm in three dimensions. IEEE Trans Pattern Anal Mach Intell 37(12):2451–2463

Budwig R (1994) Refractive index matching methods for liquid flow investigations. Exp Fluids 17(5):350–355

Dovichi NJ, Martin JC, Jett JH, Trkula M, Keller RA (1984) Laser-induced fluorescence of flowing samples as an approach to single-molecule detection in liquids. Anal Chem 56(3):348–354

Estevadeordal J, Goss L (2005) PIV with LED: particle shadow velocimetry (PSV) technique. In: 43rd AIAA aerospace sciences meeting and exhibit, p 37

Gesemann S (2015) From particle tracks to velocity and acceleration fields using b-splines and penalties. arXiv preprint arXiv:1510 09034

Goss L, Estevadeordal J, Crafton J (2007) Velocity measurements near walls, cavities, and model surfaces using particle shadow velocimetry (PSV). In: 2007 22nd International Congress on Instrumentation in Aerospace Simulation Facilities, IEEE, pp 1–8

Hessenkemper H, Ziegenhein T (2018) Particle shadow velocimetry (PSV) in bubbly flows. Int J Multiph Flow 106:268–279

Huhn F, Schanz D, Gesemann S, Dierksheide U, van de Meerendonk R, Schröder A (2017) Large-scale volumetric flow measurement in a pure thermal plume by dense tracking of helium-filled soap bubbles. Exp Fluids 58(9):116

Huhn F, Schanz D, Manovski P, Gesemann S, Schröder A (2018) Time-resolved large-scale volumetric pressure fields of an impinging jet from dense lagrangian particle tracking. Exp Fluids 59(5):81

Jordt A, Zelenka C, von Deimling JS, Koch R, Köser K (2015) The bubble box: Towards an automated visual sensor for 3d analysis and characterization of marine gas release sites. Sensors 15(12):30716–30735

Kähler CJ, Astarita T, Vlachos PP, Sakakibara J, Hain R, Discetti S, La Foy R, Cierpka C (2016) Main results of the 4th international PIV challenge. Exp Fluids 57(6):97

Khodaparast S, Borhani N, Thome J (2014) Application of micro particle shadow velocimetry μPSV to two-phase flows in microchannels. Int J Multiph Flow 62:123–133

Li Y, Perlman E, Wan M, Yang Y, Meneveau C, Burns R, Chen S, Szalay A, Eyink G (2008) A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence. J Turbul 9:N31

Lindken R, Merzkirch W (2002) A novel PIV technique for measurements in multiphase flows and its application to two-phase bubbly flows. Exp Fluids 33(6):814–825

Malik N, Dracos T, Papantoniou D (1993) Particle tracking velocimetry in three-dimensional flows. Exp Fluids 15(4–5):279–294

Masuk AUM, Salibindla A, Tan S, Ni R (2019) V-ONSET (vertical octagonal noncorrosive stirred energetic turbulence): a vertical water tunnel with a large energy dissipation rate to study bubble/droplet deformation and breakup in strong turbulence. Rev Sci Instrum 90(8):085105

Meinhart C, Wereley S, Gray M (2000) Volume illumination for two-dimensional particle image velocimetry. Measur Sci Technol 11(6):809

Mordant N, Crawford AM, Bodenschatz E (2004) Experimental Lagrangian acceleration probability density function measurement. Physica D Nonlinear Phenomena 193(1–4):245–251

Ni R, Huang SD, Xia KQ (2012) Lagrangian acceleration measurements in convective thermal turbulence. J Fluid Mech 692:395–419

Nishino K, Kasagi N, Hirata M (1989) Three-dimensional particle tracking velocimetry based on automated digital image processing. J Fluids Eng 111(4):384–391

Nishino K, Kato H, Torii K (2000) Stereo imaging for simultaneous measurement of size and velocity of particles in dispersed two-phase flow. Measur Sci Technol 11(6):633

Novara M, Schanz D, Gesemann S, Lynch K, Schröder A (2016) Lagrangian 3D particle tracking for multi-pulse systems: performance assessment and application of Shake-The-Box. In: 18th international symposium on applications of laser techniques to fluid mechanics, pp 4–7

Olsen M, Adrian R (2000) Out-of-focus effects on particle image visibility and correlation in microscopic particle image velocimetry. Exp Fluids 29(1):S166–S174

Ouellette NT, Xu H, Bodenschatz E (2006) A quantitative study of three-dimensional Lagrangian particle tracking algorithms. Exp Fluids 40(2):301–313

Papantoniou D, Dracos T (1989) Analyzing 3-D turbulent motions in open channel flow by use of stereoscopy and particle tracking. In: Fernholz HH, Fiedler HE (eds) Advances in Turbulence 2. Springer, Berlin, Heidelberg, pp 278–285

Rossi M, Segura R, Cierpka C, Kähler CJ (2012) On the effect of particle image intensity and image preprocessing on the depth of correlation in micro-PIV. Exp Fluids 52(4):1063–1075

Schanz D, Gesemann S, Schröder A (2016) Shake-The-Box: Lagrangian particle tracking at high particle image densities. Exp Fluids 57(5):70

Schlueter-Kuck KL, Dabiri JO (2017) Coherent structure colouring: identification of coherent structures from sparse data using graph theory. J Fluid Mech 811:468–486

Schneiders JF, Scarano F (2016) Dense velocity reconstruction from tomographic PTV with material derivatives. Exp Fluids 57(9):139

Schneiders JF, Scarano F, Elsinga GE (2017) Resolving vorticity and dissipation in a turbulent boundary layer by tomographic PTV and VIC+. Exp Fluids 58(4):27

Schröder A, Schanz D, Michaelis D, Cierpka C, Scharnowski S, Kähler CJ (2015) Advances of piv and 4d-ptv shake-the-box for turbulent flow analysis-the flow over periodic hills. Flow Turbul Combust 95(2–3):193–209

Tan S, Salibindla A, Masuk AUM, Ni R (2019) An open-source Shake-the-Box method and its performance evaluation. In: 13th international symposium on particle image velocimetry

Tsai R (1987) A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. IEEE J Robot Autom 3(4):323–344

Van Gent P, Michaelis D, Van Oudheusden B, Weiss PÉ, de Kat R, Laskari A, Jeon YJ, David L, Schanz D, Huhn F et al (2017) Comparative assessment of pressure field reconstructions from particle image velocimetry measurements and Lagrangian particle tracking. Exp Fluids 58(4):33

Wieneke B (2008) Volume self-calibration for 3D particle image velocimetry. Exp Fluids 45(4):549–556

Wieneke B (2012) Iterative reconstruction of volumetric particle distribution. Measur Sci Technol 24(2):024008

Wieneke B (2018) Improvements for volume self-calibration. Measur Sci Technol 29(8):084002

Wiener N (1949) Extrapolation, interpolation, and smoothing of stationary time series, vol 2. MIT Press, Cambridge

Xu H (2008) Tracking Lagrangian trajectories in position-velocity space. Measur Sci Technol 19(7):075105

Zaruba A, Lucas D, Prasser HM, Höhne T (2007) Bubble-wall interactions in a vertical gas-liquid flow: Bouncing, sliding and bubble deformations. Chem Eng Sci 62(6):1591–1605