



A Probabilistic Particle Tracking Framework for Guided and Brownian Motion Systems with High Particle Densities

Sebastian Herzog^{1,2,3} · Daniel Schiepel¹ · Isabella Guido² · Robin Barta¹ · Claus Wagner^{1,4}

Received: 26 April 2021 / Accepted: 14 September 2021 / Published online: 17 October 2021
© The Author(s) 2021

Abstract

This paper presents a new framework for particle tracking based on a Gaussian Mixture Model (GMM). It is an extension of the state-of-the-art iterative reconstruction of individual particles by a continuous modeling of the particle trajectories considering the position and velocity as coupled quantities. The proposed approach includes an initialization and a processing step. In the first step, the velocities at the initial points are determined after iterative reconstruction of individual particles of the first four images to be able to generate the tracks between these initial points. From there on, the tracks are extended in the processing step by searching for and including new points obtained from consecutive images based on continuous modeling of the particle trajectories with a Gaussian Mixture Model. The presented tracking procedure allows to extend existing trajectories interactively with low computing effort and to store them in a compact representation using little memory space. To demonstrate the performance and the functionality of this new particle tracking approach, it is successfully applied to a synthetic turbulent pipe flow, to the problem of observing particles corresponding to a Brownian motion (e.g., motion of cells), as well as to problems where the motion is guided by boundary forces, e.g., in the case of particle tracking velocimetry of turbulent Rayleigh–Bénard convection.

Keywords Brownian motion · Turbulent flow · Particle tracking · Gaussian mixture model

Introduction

In many areas of science, theories and experiments are in constant interaction. Theories are based on experimental designs and the insights gained with experiments can change theories or create new ones. However, quantitative key indicators cannot be directly derived from any experiment. For the quantitative analysis of different dynamic processes

from sequential image, data particle tracking [27, 29, 36] is important. An overview about the challenges of developing robust visual tracking systems and some state-of-the-art techniques is given by [13, 22]. To detect and follow large numbers of individual particles, many automated computational methods have been developed. One of the most prominent method is the so-called *Shake-the-Box* algorithm [37, 45], which is commercial software that is not freely available. Besides this approach, other promising techniques have been developed, e.g., a non-iterative double frame algorithm [17], an improved iterative particle reconstruction algorithm [23], a high-speed particle tracking algorithm [24] for image processing in real-time and tracking algorithms for deforming [39] or non-uniform [51] particles. Generally speaking, these methods rely on two steps: particle detection where spatial information is extracted based on spots that protrude from the background using certain criteria to identify the particle and its position in every frame of the image sequence. The second step is the so-called particle linking by where the temporal relation between the particles from frame to frame is determined using another set of criteria to form tracks. These two components result in

✉ Sebastian Herzog
sebastian.herzog@dlr.de

¹ German Aerospace Center, Institute for Aerodynamics and Flow Technology, Bunsenstr. 10, 37073 Göttingen, Germany

² Max Planck Institute for Dynamics and Self-Organization, Am Fassberg 17, 37077 Göttingen, Germany

³ Third Institute of Physics-Biophysics, Department for Computational Neuroscience, University of Göttingen, Friedrich-Hund-Platz 1, 37077 Göttingen, Germany

⁴ Technische Universität Ilmenau, Institute for Thermodynamics and Fluid Mechanics, Helmholtzring 1, 98693 Ilmenau, Germany

different approaches to visualize the movements, such as optical flow methods [6], Particle Image Velocimetry [4] (PIV), and Particle Tracking Velocimetry (PTV) [37, 45]. While particle detection is highly task specific, the linking procedure has a more general character. This work presents an approach based on a Gaussian Mixture Model (GMM) [18, 40]. GMMs are used for many different problems, e.g., background subtraction in images [30], superpixel segmentation [5], image denoising [59], or density estimation in atmospheric Lagrangian particle dispersion models [10]. In this work, a modified GMM is introduced as a probabilistic model to describe the movement of a particle. The main focus of this method is: (i) to model trajectories with coupled dimensions (i.e., position and velocity) where changing one dimension leads to changes in the other dimensions, resulting in a high prediction accuracy, (ii) the ability to change the start/end point without a total recalculation of the whole track allowing high computational efficiency, (iii) creating a compact representation of the determined tracks requiring less memory space, and (iv) to be able to model periodic trends in the particle motion. After validating the approach in a synthetic case providing a ground truth for the quantitative comparison, the functionality is demonstrated by means of two different applications. The first application is the tracking of cell migration. The capacity of cells to migrate can be considered as a fundamental mechanism that is physiologically essential for biological processes, which include embryonic development, immune response, wound healing, and spread of pathological conditions like cancer [52]. More general cells are able to sense external stimuli of different nature, such as chemical gradients [54], electric fields [19, 31], substrate stiffness [25], and to respond with a directed movement toward or away from the stimulus source depending on the reaction transduced by the cells. Analyzing the dynamics of cells during migration by following them over time provides information about trajectories and velocities that allow the characterization of fundamental cellular mechanisms. Cell imaging is the most appropriate approach to get access to cell behavior and the automation of microscopes has enabled the acquisition of time-lapse images with fluorescence or bright-field microscopy of big samples of cells at single-cell resolution [14]. However, image analysis is a bottleneck for biological advances as there is a gap between the advanced techniques to collect data and the ability to analyze it. Manual analysis of big data sets is time-consuming. Additionally, it introduces bias of the operator and results that cannot be reproduced. This applies especially to the linking of single cells to tracks. Therefore, automated methods for cell tracking represent a vivid field of research as there is considerable interest in an efficient and accurate cell-tracking method that can overcome problems such as low signal-to noise ratio, poor staining, variable fluorescence in cells, low contrast, stain-free cell images, high

cell density, deformable cell shapes [39], sudden changes in motion direction and speed, and temporary drop out of focal plane [53]. The second application belongs to the domain of experimental fluid mechanics, where methods like PIV and PTV are the most prominent flow field measurement techniques yielding velocity vectors within observation planes or volumes. They have in common that the velocity vectors in a moving fluid are determined from the displacement of seeding particles transported by the flow during a prescribed time interval. For many years, the main difference was that PIV [1] yielded two-dimensional (2D) velocity vectors in planes at particle image densities of $\approx 0.03 - 0.05$ particles per pixel (ppp), whereas PTV provided three-dimensional (3D) velocity vector fields in volumes based on particle triangulation and nearest-neighbor searches [27] for one magnitude lower tracer particle densities. The game changer for PIV was the introduction of tomographic reconstruction of particle distributions followed by three-dimensional cross-correlation [15] which were prerequisite for the time-resolved 3D tomographic PIV technique. In the latter, the particle positions are iteratively reconstructed as intensity peaks in a 3D Eulerian voxel space using algorithms like the multiplicative algebraic reconstruction technique (MART) or simultaneous MART (SMART) [4, 21]. This made tomographic PIV a reliable and robust 3D flow field measurement technique. Additionally, negative effects introduced by ghost particles are less significant than in PTV, since the 3D correlation is performed after the reconstruction process. However, both reconstruction and cross-correlation are computationally expensive and the latter reduces the spatial resolution by relying on the mean displacement of particles within interrogation volumes. This is especially significant for measurements of turbulent flows, since the cross-correlation filters out small-scale structures and high velocity gradients due to the inherent spatial averaging over interrogation volumes. Unlike in tomographic PIV, in 3D PTV, the particle positions are first identified in a number (more than two) of 2D camera images and subsequently matched—typically by triangulation [20]—to obtain the 3D positions in the measurement volume for each time step in a Lagrangian reference systems. Subsequently, trajectories of individual particles are determined by matching particle positions of the successive time step [27, 28, 36]. Based on these trajectories, the velocity and acceleration fields can be determined more precisely than in PIV, since spatial averaging is not involved.

The presented tracking method shall be considered as an extension to existing frameworks. For the cell-tracking case, the input for the presented approach is segmented images from microscope images, where the segmentation can be achieved with the open source software ImageJ [49]. In the domain of Particle Tracking Velocimetry (PTV), the input images are achieved by the particle displacement per time interval typically prescribed by the pulse illumination

frequency. This frequency can be increased using high-speed lasers and high-speed cameras for measurements of high-speed flows. However, until a few years ago, the downside of the PTV approach used to be the limitation of the triangulation and matching process restricting the technique to low particle image densities, i.e., in the order 0.005 particles per pixel (ppp). To match particles to tracks, nearest-neighbor methods were used which could not cope with such dense (and possibly false) particle distributions in volumes. To relax the density limitations, approaches for tracking single particles like Enhanced Particle Tracking Velocimetry (EPTV) [9, 32] are multi-parametric as they consider the particle size or local particle concentration. Mikheev and Zubtsov [32] propose to improve the tracking procedure by proper pre-conditioning of the particle displacement which is realized in the time-resolved 3D tracking method of the commercially distributed *Shake-The-Box* (STB) software. With the latter, particle positions in subsequent images are predicted by extrapolating trajectories generated using former images with a third-order polynomial [45]. A prerequisite for the latter is the combination of calibration methods like volume self-calibration [56] developed for tomographic PIV with iterative particle reconstruction (IPR) and image matching by shaking proposed in [57]. In contrast to the STB software, the here presented probabilistic particle tracking approach does not rely on a tomographic PIV evaluation for initialization or a Particle-Space Correlation as used for multi-pulse applications [37].

Methods

The approach proposed below includes a triangulation step and a modeling step with a new method for tracking and predicting the motion of 3D particles. The triangulation is based on an iterative approach inspired by [57] to triangulate points in a three-dimensional volume, based on a number of two-dimensional images. The modeling step tracks and predicts the motion of a reconstructed 3D particle. With the proposed model, any particle trajectory can be approximated by a GMM [40]. The outstanding feature of this method is that no initial 3D particle distribution is required. The presented method uses the *Soloff model* [50] as an optical transfer function (OTF) [58], whose parameters are determined during a volume self-calibration [56]. In contrast to [57], the particle trajectories of any reconstructed particle are predicted using a probabilistic model based on GMMs [40] allowing to determine the local velocity and acceleration vectors as derivatives of basis functions. The procedure of the resulting framework is summarized below together with references to sections and equations which provide more details of the respective steps.

Algorithm 1: procedure of the framework

```

Data: Images
Result: Tracks
initialization;
for  $t = 1, 2$  do
    Image processing (sec. 2.1);
    Initial points  $\leftarrow$  Triangulate 3D points for  $t$  (sec.
    2.2)
end
Calculate initial velocities based on initial points
(sec. 2.3);
All tracks  $\leftarrow$  Link initial points based on initial
velocities ;
Predict all tracks to update all velocities (sec. 2.4);
for  $i = 3, \dots, T_{max}$  do
    Image processing (sec. 2.1);
    Extend all tracks (Eq. (2.5));
    Remaining points  $\leftarrow$  Back projection of all
    extended Tracks and remove 2D points when
    back projection matches;
    New 3D points  $\leftarrow$  Triangulate 3D points for  $t$ 
    (sec. 2.2);
    Calculate new velocities based on new 3D points
    and non extend 3D points (sec. 2.3);
    All tracks  $\leftarrow$  Link new points based on the new
    velocities;
end

```

Image Processing

To reconstruct the three-dimensional particle distribution \mathcal{P}_t , where $t \in \mathbb{N}$ is the time, from the individual perspectives $c \in \mathbb{N}$, it is necessary to remove possible noise sources from the camera images. A camera image is considered as a set of pixels with intensities $I_{t,c} \in \mathbb{R}^{r \times c}$, where $r \in \mathbb{N}$ is the number of rows and $c \in \mathbb{N}$ the number of columns of the image. After processing the image $I_{t,c}$ the sub-pixel localization of imaged particles is calculated. By doing so, the individual pictures are reduced to a set of coordinates $C_{t,c}$ and a set of intensities $\mathcal{I}_{t,c}$ in the following four steps:

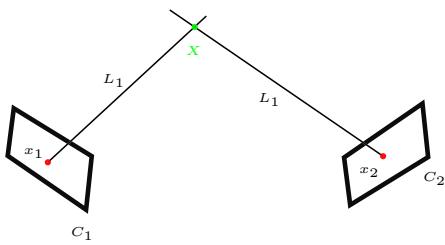
- 1. Masking:** For the masking step, a predefined mask M_c is placed over the image $I_{t,c}$, such that $I'_{t,c} = M_c \cap I_{t,c}$ removing all irrelevant or physically meaningless parts of the images.
- 2. Richardson–Lucy deconvolution:** To reverse blurring from the input images and to amplify the noise, the Richardson–Lucy deconvolution [26, 41] is used.
- 3. Background subtraction:** To remove all parts of $I'_{t,c}$ which do not change in time a, the minimum image $\min\{I'_{t,c}, I'_{t+1,c}, I'_{t+2,c}\}$ is calculated and subtracted pixel-wise, such that:

$$I''_{t,c} = I'_{t,c} - \min\{I'_{t,c}, I'_{t+1,c}, I'_{t+2,c}\}.$$
- 4. Thresholding:** In the next step, all remaining intensities in the image below a certain threshold are removed. This

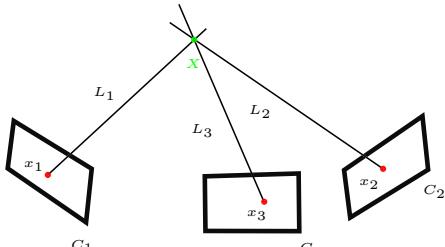
- is done by simply checking whether the intensity of a pixel on an image is above a threshold ϵ_{thres} , such that $I''_{t,c} = \text{thres}(I''_{t,c}, \epsilon_{thres})$.
5. Calculating sub-pixel particle localization: In the last step, the clustered pixels which belong to one real particle on the camera images are reduced to coordinates by applying the analytic, non-iterative radial symmetry center method RSC(.) presented in [38], such that $C_{t,c} = \text{RSC}(I''_{t,c})$. The RSC method tries to fit a radial symmetry function on a cluster of pixels. The radial symmetry center is then the sub-pixel particle localization for the clustered pixels, where the corresponding intensity values $\mathcal{I}_{t,c}$ are integrated over the radial symmetry.

Triangulation

To reconstruct 3D points from $C_{t,c}$ and $\mathcal{I}_{t,c}$, a camera calibration is required to provide intrinsic and extrinsic parameters to set up mapping function $F_c : \mathbb{R}^3 \rightarrow \mathbb{R}^2$, $F_c(X) = x_c$, where $X \in \mathbb{R}^3$ is the position in the 3D space and $x \in \mathbb{R}^2$ is the position on some camera images with index c . The general case of triangulation deals with the problem of reconstructing 3D objects from a series of perspective shifted images in the real world. To compensate for different distortions occurring in experiments, we make use of the Soloff model [50] for volume self-calibration [56] to obtain the parameters for the mapping function F_c . Figure 1 illustrates the relations of a 3D point $X \in \mathbb{R}^3$ and



(a) Relation for two perspectives, $c = 1, 2$



(b) Relation from three perspectives, $c = 1, 2, 3$

Fig. 1 Visualization of the projection lines for two different cases. The first one with two perspectives and the second one with three perspectives. The point X in 3D projected onto x_c lying in two and three images

its projection on to the images from $\mathbb{R}^3 \rightarrow \mathbb{R}^2$, such that $L_1 : F_1(X) = x_1$ and $L_2 : F_2(X) = x_2$.

For the triangulation of each $x \in \mathcal{C}_{t,c}$, two points $Y, Y' \in \mathbb{R}^3$ are required, such that $F_c(Y) = F_c(Y') = x_c$. The straight line through Y and Y' defines the line L_c . However, to triangulate the desired point X , at least two projection lines $L_c, L_{c'}$, where $c \neq c'$, are needed. To construct these lines, the correspondence between at least two $x_c \in \mathcal{C}_{t,c}$ is necessary. To determine this correspondence, the two points Y and Y' are projected onto another image, such that $F_{c'}(Y) = x_{c'}$ and $F_{c'}(Y') = x'_{c'}$. The line through $x_{c'}$ and $x'_{c'}$ on $I_{t,c'}$ is called the epipolar line, $E_{c \rightarrow c'}$. By calculating the Euclidean distance for all possible coordinate $x \in \mathcal{C}_{t,c'}$ to the line $E_{c \rightarrow c'}$, given by $d(E, x)$, all coordinates, where $d(E, x) < \epsilon_E$, are considered as possible candidates and ϵ_E is a control parameter. If the number of candidates is higher than 1, new epipolar lines are calculated for all possible candidates and are then projected onto the next image. In the next image, only the intersections of epipolar lines define possible areas for corresponding coordinates. If more lines intersect in an area, the correspondence for a point is considered higher. Finally, the points with the minimal distance and highest number of intersections are used for triangulation, as illustrated in Fig. 2. Remark: If several points have the same number of intersections and the same distance, all points are triangulated.

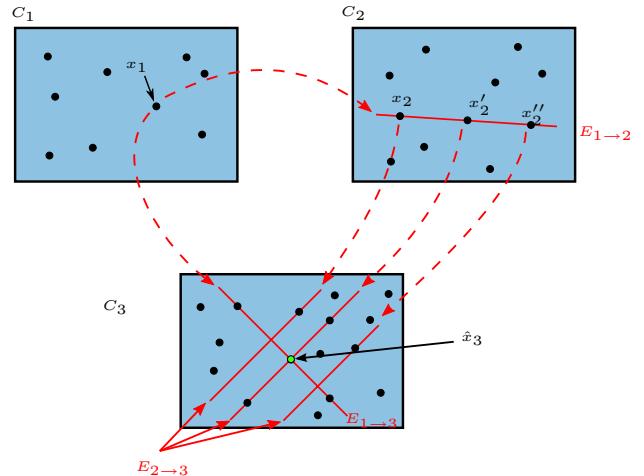


Fig. 2 Starting from C_1 , a point x_1 is selected for which the corresponding points must be found on other images. By calculating two 3D points Y, Y' , the epipolar line $E_{1 \rightarrow 2}$ in C_2 is obtained. Based on $E_{1 \rightarrow 2}$, three possible candidates need to be considered: x_2, x'_2, x''_2 . For each of these points, new epipolar lines are calculated and projected onto the next image C_3 together with the epipolar line from C_1 . This leads to three intersection points in C_3 , where the nearest point to one of this intersections is \hat{x}_3 . Going the way back, the corresponding points for x_1 from C_1 are x'_2 on C_2 and \hat{x}_3 on C_3

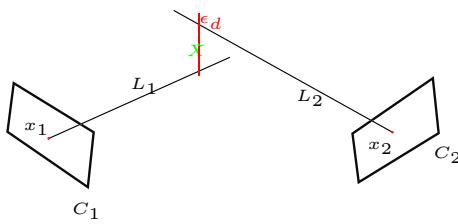


Fig. 3 Illustration of a misalignment of the projection lines L_c , where the distance ϵ_d is the maximum distance that is still accepted as the intersection point

After the corresponding points on different images are found, it is possible to use these points for triangulation of the 3D point. Considering the associated projection lines L_c , the analytical solution of this triangulation problem is the point of intersection of all L_c 's. If the lines are not intersecting $X_{est} \sim X$ is used where X_{est} is the point which minimizes the sum of the distances $\sum_{\forall c} d(L_c, X)$ to approximate X .

With the above discussed approach, it is possible to triangulate any number of points X_{est_t} for any time step t . Furthermore, the intensity I of X_{est_t} is obtained using $F_c(X_{est_t}) = x_{est_t}$ for back projection into the images $I_{t,c}$. The nearest point in a ϵ_E space on $I_{t,c}$ is then considered as the intensity value for X_{est_t} , denoted by $I(X_{est_t})$.

Local Optical Transfer Function

The local optical transfer function by Soloff [50] is a non-linear function $F_a : \mathbb{R}^3 \mapsto \mathbb{R}$

$$\begin{aligned} F_a(X, Y, Z) := & a_0 \\ & + X(X(a_9X + a_{11}Y + a_{14}Z + a_4) \\ & + a_{13}YZ + a_6Y + a_7Z + a_1) \\ & + Y(Y(a_{12}X + a_{10}Y + a_{15}Z + a_5) + a_8Z + a_2) \\ Z(Z(a_{17}X + a_{18}Y + a_{16}) + a_3), \end{aligned} \quad (1)$$

which maps a 3D point to one coordinate where (X, Y, Z) is the 3D position of the point, and a are the parameters of the Soloff model. Using two different parameter sets, i.e., a, b , it is possible to match the 3D point to a 2D image position. Furthermore, this function is used to generate the lines L_c . L_c is defined as the line by passing through two 3D points. To find this two points, the partial derivatives of F , which are

$$\begin{aligned} \frac{\partial}{\partial X} F_a(X, Y, Z) &= 3a_9x^2 + 2a_{11}xy + 2a_{14}xz \\ &+ 2a_4x + a_{12}y^2 + a_{13}yz + a_6y + a_{17}z^2 + a_7z + a_1, \\ \frac{\partial}{\partial Y} F_a(X, Y, Z) &= a_{11}x^2 + 2a_{12}xy + a_{13}xz \\ &+ a_6x + 3a_{10}y^2 + 2a_{15}yz + 2a_5y + a_{18}z^2 + a_8z + a_2, \\ \frac{\partial}{\partial Z} F_a(X, Y, Z) &= 2z(a_{17}x + a_{18}y + a_{16}) \\ &+ x(a_{14}x + a_{13}y + a_7) + y(a_{15}y + a_8) + a_3 \end{aligned} \quad (2)$$

are used to solve the system

$$\left(\begin{array}{ccc} \frac{\partial F_a}{\partial X} & \frac{\partial F_a}{\partial Y} & \frac{\partial F_a}{\partial Z} \\ \frac{\partial F_b}{\partial X} & \frac{\partial F_b}{\partial Y} & \frac{\partial F_b}{\partial Z} \end{array} \right) \left(\begin{array}{c} X^* \\ Y^* \\ Z^* \end{array} \right) = \left(\begin{array}{c} (F_a(X, Y, Z) - x) \\ (F_b(X, Y, Z) - y) \end{array} \right), \quad (3)$$

Algorithm 2: Determining two points on a line of sight

```

Data: a, b, (x, y), (X, Y, Z)
Result: (X + h_x, Y + h_y, Z)
(X_i, Y_i, Z_i) = (X, Y, Z);
f = (f1, f2) = (max value, max value );
while ||f||_2 > float precision do
    A = 
        ( ∂/∂X F_a(X_i, Y_i, Z) ∂/∂Y F_a(X_i, Y_i, Z) ∂/∂Z F_a(X_i, Y_i, Z) )
        ( ∂/∂X F_b(X_i, Y_i, Z) ∂/∂Y F_b(X_i, Y_i, Z) ∂/∂Z F_b(X_i, Y_i, Z) )
    b = ( (x - F_a(X_i, Y_i, Z)) )
        ( (y - F_b(X_i, Y_i, Z)) );
    Solve Ah - b = f;
    Update X_i = X_i + h_x;
    Update Y_i = Y_i + h_y;
end

```

where (x, y) is a selected point on an image and (X, Y, Z) are arbitrary start values for the algorithm. To determine the line L_c , the algorithm 1 must be executed twice for the same 2D point but with different initial values (X, Y, Z) . We suggest selecting the values for (X, Y, Z) , so that they are on the opposite side of the object to be examined. L_c is the line between the two new 3D points.

Generation of Initial Tracks

The previously introduced triangulation is used to triangulate all possible points from the first two time steps t_0, t_1 . This serves as an initialization for the tracking approach described in section “Probabilistic Motion Prediction”. To form tracks from the two point sets at t_0 and t_1 , it is necessary to first estimate velocities for the points at t_0 . To calculate the initial displacement, first the sets $\forall \omega_{t_0} \in \Omega_{t_0}$ and $\forall \omega_{t_1} \in \Omega_{t_1}$, where

$$\Omega_{t_0} = \{p_{t_0} \in \mathcal{P}_{t_0} : \|X_{est_{t_0}} - p_{t_0}\|_2 < \epsilon_\Delta/2\} \quad (4)$$

and

$$\Omega_{t_1} = \{p \in \mathcal{P}_{t_1} : \|X_{est_{t_1}} - p_{t_1}\|_2 < \epsilon_\Delta/2\}, \quad (5)$$

are determined. Second, the Cartesian product $\Omega_{t_0} \times \Omega_{t_1}$ is used to calculate histograms for each displacement direction. The most frequent values in each histogram form $\Delta X_{est_{t_0}}$. Both steps are illustrated in Fig. 4. Two points are forming a track if the

$$\|(X_{est_{t_0}} + \Delta X_{est_{t_0}}) - X_{est_{t_1}}\|_2 \leq \epsilon_s, \quad (6)$$

if there are several candidates for $X_{est_{t_1}}$ that fulfill this requirement, the $X_{est_{t_1}}$ with the smallest deviation is taken. ϵ_s and ϵ_Δ are user-defined parameters.

Probabilistic Motion Prediction

To track individual particles X in time t , three (one for each dimension) Gaussian mixture models (GMMs) [40] are used to approximate the trajectory of X . The aim of this approximation is to determine a function mapping based on the currently observed history for predicting the most likely particle motion.

However, since the trajectories may perform a chaotic movement in a turbulent flow [2], the prediction procedure is a not generalized problem [35]. To deal with it, we suggest using a probabilistic prediction based on the GMM instead of the extrapolation used in [57]. More precisely, we consider a GMM which can model both the position

q_0, q_1, \dots, q_{n-1} and velocity $\dot{q}_0, \dot{q}_1, \dots, \dot{q}_{n-1}$ of the particle. Considering the density of a Gaussian mixture model [40] given by

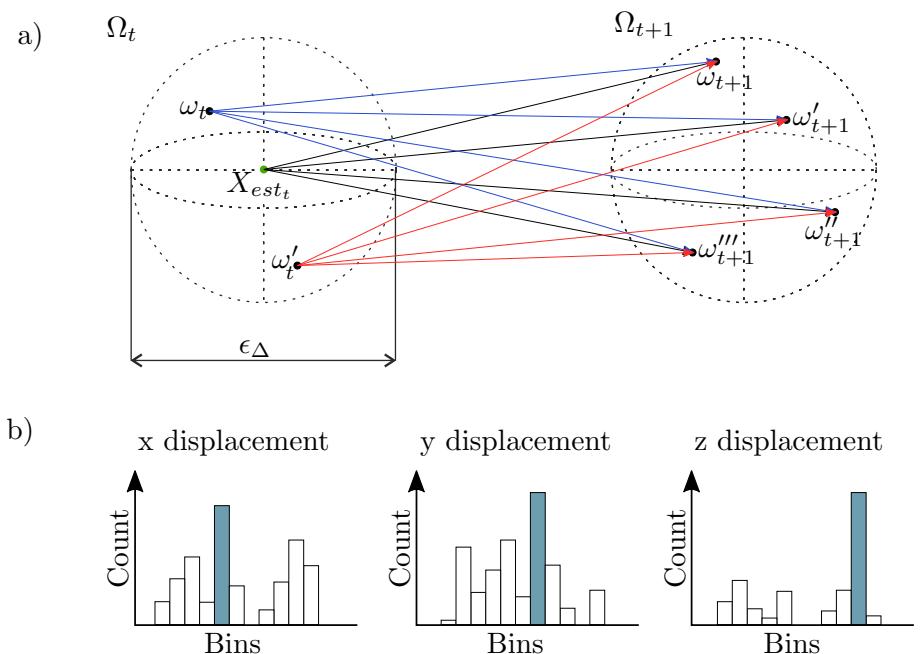
$$p(x) = \sum_{k=1}^K \pi_k \mathcal{N}(x|\mu_k, \Sigma_k), \quad (7)$$

where x is the d -dimensional random variable, and $\mathcal{N}(x|\mu_k, \Sigma_k)$ is a multivariate normal distribution, with mean μ_k and a covariance matrix Σ_k . The coefficients of Σ_k termed by π_k are the components of the density distribution $p(x)$, which have to satisfy $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^K \pi_k = 1$. To derive the parameters for Eq. (7), the Expectation–Maximization (EM) algorithm [34] is used. It is a two-step iterative algorithm which is identifying the maximum-likelihood solution by computing the expectation step (E-step) of the log-likelihood evaluated with the current parameter estimation followed by the maximization step (M-step). This step estimates parameters that maximize the expected log-likelihood obtained with the E-step. Applying the above to the prediction leads to

$$p(x_f, x_h) = \sum_{k=1}^K \pi_k \mathcal{N}(x_h, x_f | \mu_k, \Sigma_k), \quad (8)$$

by inferring a joint Gaussian mixture distribution, where x_h is the history of the trajectory and x_f the future. The prediction is then performed by calculating the conditional mixture density

Fig. 4 a Visualization of the Cartesian product $\Omega_t \times \Omega_{t+1}$, where $\Omega_t = \{\omega_t, X_{est_t}, \omega'_t\}$ and $\Omega_{t+1} = \{\omega_{t+1}, \omega'_{t+1}, \omega''_{t+1}, \omega'''_{t+1}\}$. The blue arrows starting from ω_t exemplify all possible displacements for ω_t , and the same applies to for the black arrow starting from X_{est_t} and red arrow starting at ω'_t . Based on all this displacements, the histograms in b are set up. The bins with the highest counts (colored in cyan blue) in each histogram are the estimation for ΔX_{est_t}



$$p(x_f, x_h) = \frac{p(x_h, x_f)}{\int p(x_h, x_f) dx_f} = \sum_{k=1}^K \hat{\pi}_k \mathcal{N}(x_f | x_h, \hat{\mu}_k, \hat{\Sigma}_k), \quad (9)$$

which is again a GMM, with the parameters

$$\hat{\pi}_k = \frac{\pi_k p(x_h | \mu_{k,x_h}, \Sigma_{k,x_h x_h})}{\sum_{j=1}^K \pi_j p(x_h | \mu_{j,x_h}, \Sigma_{j,x_h x_h})} \quad (10)$$

$$\hat{\mu}_k = \mu_k + \Sigma_{k,x_f x_h} \Sigma_{k,x_h x_h}^{-1} (x_h - \mu_{k,x_h}) \quad (11)$$

$$\hat{\Sigma}_k = \Sigma_{k,x_f x_f} - \Sigma_{k,x_f x_h} \Sigma_{k,x_h x_h}^{-1} \Sigma_{k,x_h x_f}, \quad (12)$$

where

$$\mu_k = \begin{bmatrix} \mu_{k,x_h} \\ \mu_{k,x_f} \end{bmatrix} \quad (13)$$

$$\Sigma_k = \begin{bmatrix} \Sigma_{k,x_h x_h} & \Sigma_{k,x_h x_f} \\ \Sigma_{k,x_f x_h} & \Sigma_{k,x_f x_f} \end{bmatrix} \quad (14)$$

is the partitioning of the means and covariance matrices of the GMM. Equation (9) defines the future trajectories for which expectations can be evaluated, where the mean and covariance of a GMM is given by

$$\mu = \sum_{k=1}^K \pi_k \mu_k \quad (15)$$

$$\Sigma = \sum_{k=1}^K \pi_k (\Sigma_k + (\mu_k - \mu)(\mu_k - \mu)^T) \quad (16)$$

and the probabilistic prediction can be applied. However, to obtain a parametric representation of each trajectory, a compact representation of a single trajectory is desired. In addition, the dimensions of the position and the velocity should be coupled. By defining $\phi_t = [\phi_t, \dot{\phi}_t] \in \mathbb{R}^{K \times 2}$ as the time-dependent basis matrix for q_t and \dot{q}_t and some noise $\epsilon_{Noise} \sim \mathcal{N}(0, \Sigma_k)$, a trajectory τ can be defined as

$$\tau = \begin{bmatrix} q_t \\ \dot{q}_t \end{bmatrix} = \phi_t^T w + \epsilon_{Noise}, \quad p(\tau | w) = \prod_k \mathcal{N}(x | \mu_k, \Sigma_k). \quad (17)$$

Equation (17) represents a linear basis function model with weights $w \in \mathbb{R}^K$, where the basis functions for the position are defined by the product of normal distributions and the basis functions for the velocity are the time derivatives of the basis functions for the position. Using this representation, only the weights w needs to be stored for each trajectory. This process is illustrated in Fig. 5.

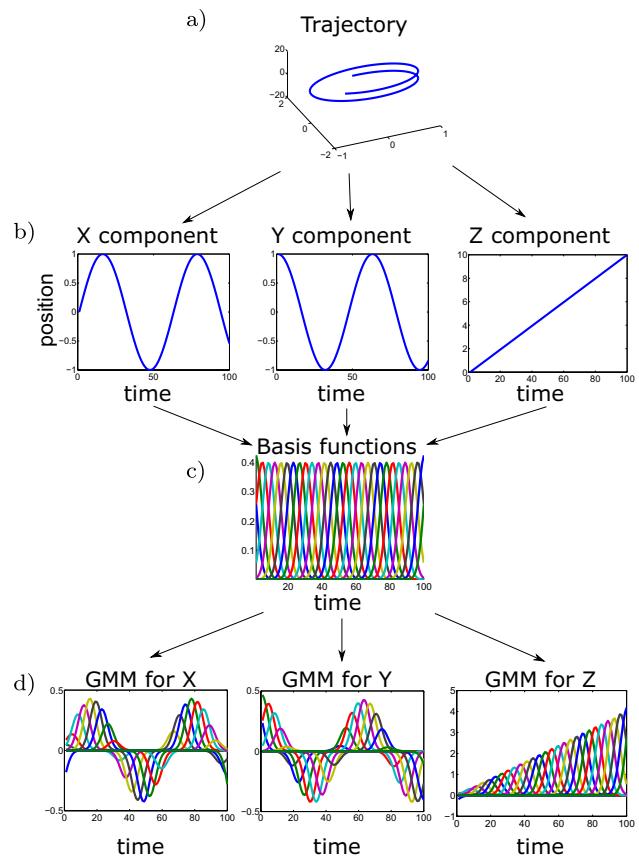


Fig. 5 Representation of a three-dimensional trajectory with three GMMs. The first line **a** shows an example of a trajectory and the second line **b** the x , y and z components of the trajectory. Line **c** represents the set of basis functions from Eq. (7). Using the basis functions and the components from line **b**, w is calculated by Eq. (17). By scaling the basis functions with the obtained w , the basis functions preserve the shape of the desired component. This is illustrated in line **d**

Extending Trajectories and Finding New Trajectories

After the generation of the initial trajectories, these can be used to predict the position of the traced particles at the next time step $t + 1$, denoted by X_{pred} , by adding the velocity at time t to the position at time t . The predicted position at $t + 1$ is then projected back onto the images $I_{c,t+1}$ and evaluated in such way that all points $x_{I_{c,t+1}}$, but at least two points $x_{I_{c,t+1}}$ and $x_{I_{c',t+1}}$, where $c \neq c'$, are found, such that

$$\left(\sum_c \|x_{I_{c,t+1}} - F_c(X_{pred})\|_2 * \left| 1 - \frac{I(x_{I_{c,t+1}})}{I(F_c(X_{pred}))} \right| \right) < \epsilon_{back}, \quad (18)$$

where ϵ_{opti} is a user-defined parameter. However, if such a point is found, the trajectory is extended to this point. If this is not the case, time step $t + 1$ is skipped for the trajectories and the process is repeated for time step

$t + 2, t + 3, \dots, t + gap$, where gap is a user-defined parameter. If still no matching is found the trajectory is considered to be terminated. To further minimize Eq. (18), X_{pred} is varied by component-wise moving X_{pred} , such that the projection of X_{pred} only changes by one pixel on $I_{c,t+1}$. The variation ends when Eq. (18) does not change anymore or a maximum number of variation steps is reached. Remaining points which are not part of any trajectory are considered as positions of not yet identified or new trajectories and the above-described steps are repeated, until no more points remain or until a maximum number of iterations is reached. Remark: In general, it would be possible to formulate Eq. (18) as a multi-objective optimization, so that the distance for the deviation in distance as well as the deviation of intensities are optimized independently, building a pareto front. This leads to the problem that the dimensions for the deviation in distance and the deviation of intensities need to be normalized. Instead of finding a meaningful and generally valid normalization for these independent physical deviations, we decided to use a formulation for a scalarization for the intensity deviation and to scale the deviation in distance by this scalar resulting in to a single-objective optimization problem [Eq. (18)].

Implementation Details

The PTV framework described here was implemented in C++ 2020 using the linear algebra library Armadillo [43, 44]. OpenCV [8] was used for the image processing of the PTV case and ImageJ [48] for the amoba case. For finding the nearest neighbors both in 2D and 3D, the KD-Tree implementation from mlpack [11] was used. The trajectories are processed in parallel using OpenMP [12], with which the main loop is executed in parallel. HDF5 [16] is used as the storage format.

Synthetic Pipe Flow

To validate and benchmark our framework, we set up a synthetic case of a generalized pipe flow. The simulated flow is used as a ground truth case, so that we can directly compare our reconstruction results. First, in subsection “[Particle Generation](#)”, the generation of the synthetic particles is described, whereas in “[Particle Tracking](#)”, the validity of the results obtained from our framework is proven.

Particle Generation

A synthetic case is defined with the aim to provide data sets describing the ground truth, based on which the number and accuracy of the generated trajectories can be evaluated. The case is designed to evaluate the capability of

the new framework to fulfill five requirements related to: three-dimensional particle movement, temporal resolution, particles leaving the domain, contradicting movement of the particle images between the time steps, and varying ppp densities. The case represents a generalized pressure-driven turbulent flow through a pipe.

For the synthetic case, N_p particles are initially distributed within a cubic volume with dimensions of $W \times H \times D = 500 \times 500 \times 500 \text{ mm}^3$. Each particle i has the attributes

$$P_i = \begin{pmatrix} X_i \\ I_i \end{pmatrix} \quad (19)$$

with the position $X_i \in \mathbb{R}^3$ in 3D space and the intensity $I_i \in \mathbb{R}^{n_c}$ for the individual intensities on the n_c camera images. All I_i are initialized with random uniform values between $I_{\min} = 800$ and $I_{\max} = 1200$ to prescribe a certain signal-to-noise ratio. In the considered cases $n_c = 4$, virtual cameras observe the particle distributions within the 3D cubic volume centered around $(0, 0, 0)$ corresponding to boundaries $X_{\min/\max} = \pm(250, 250, 250)$.

In a real experiment, the illuminated particles are imaged with cameras. Here, this is simulated using the transfer functions defined in Eq. (1) to project the particles onto the camera images. The particle positions are generally represented by floating point values and are accordingly blurred on the projected image in a way that they cover 2×2 pixels. Furthermore, intensity fluctuations of the particle images are taken into account. In addition, camera noise is simulated by adding a random intensity amplitude for every pixel to simulate different signal-to-noise ratios. For the cases considered below, a moderately high signal-to-noise ratio of $SNR = 5 : 1$ is prescribed, see, e.g., [60].

The generated particles are positioned in clusters forming superordinate structures A_O and advected for a prescribed number of time steps by solving analytical functions specific to A_O . For a particle i belonging to A_O at position $X_{A_O,i}^t$ at time step t , the position in the next time step $X_{A_O,i}^{t+1}$ is calculated by determining the movement direction $D_i \in \mathbb{R}^3$ defined later in Eq. 24. For two particle clusters, $O \in \{1, 2\}$, the particle velocities are computed solving

$$V_i = \delta_p \cdot S_i \circ D_i, \quad (20)$$

where $S_i \in \mathbb{R}^3$ is a random, uniformly distributed scaling factor with the operator \circ denoting component-wise multiplication. S_i in Eq. (20) are distributed between 0.95 and 1.05. This distribution is required to generate non-uniform particle motions with different particle velocities. The factor $\delta_p \in \mathbb{R}$ is an input parameter and controls the step width per time step of the particles in the artificial measurement volume

$$\delta_p \propto \left| \frac{\Delta X}{\Delta t} \right|. \quad (21)$$

Yet, a limiting factor for the framework is the step width per time step $\delta_S \in \mathbb{R}$ on the camera image

$$\delta_S = \left| \frac{\Delta x}{\Delta t} \right| \quad (22)$$

with the traveled distance of a particle $\Delta x \in \mathbb{R}^2$ measured in pixels. For a particle moving parallel to the plane of the camera sensor, both quantities are proportional $w \propto \delta_S$. A large step width δ_S may reduce the accuracy of the prediction, since a particle is more likely to deviate from the path predicted by the framework. The position in the next time step $X_{A_O,i}^{t+1}$ is then given by

$$X_{A_O,i}^{t+1} = X_{A_O,i}^t + V_i. \quad (23)$$

The synthetic case mimics a pipe flow by forming two super-ordinate aligned annular structures with particle numbers N_{A_1} and N_{A_2} moving predominantly in X direction. They are Gaussianly distributed with a spread of $\mu_A = 25$ around the center line of the respective structures to simulate a generalized turbulent pipe flow. The two structures are located around the center line of the cube normal to the X axis with different radii $r_{A_1} = 175$ mm and $r_{A_2} = 105$ mm. The direction of movement $D_i \in \mathbb{R}^3$ of a particle i is computed solving

$$D_i = (2, \cos(\alpha), \sin(\alpha + \pi)) \quad (24)$$

with

$$\alpha = \arctan((X_{A_O,i})/r_A, (Y_{A_O,i})/r_A) \quad (25)$$

with the angle α . By solving Eqs. (20), (24) and (25) with opposite signs for δ_p in Eq. (20), two particle clusters A_O with opposite mean flow directions and rotation senses are generated. The subsequent particle position at time $t + 1$ is obtained from Eq. 23 which depends on V_i defined in Eq. (20) with the above-given D_i . To keep the particle number and the resulting seeding density constant, particles leaving the domain in the outflow cross-sections are reset to their starting position in the corresponding inflow cross-section mimicking a periodic boundary condition.

An exemplary particle distribution is illustrated in Fig. 6. For the particle numbers $N_{A_1} = 18000$ and $N_{A_2} = 10000$ and a step width of $\delta_S = 7px$, structure A_1 is reflected by the green particle distribution and A_2 by the yellow one. The front view in Fig. 6a shows the annular structures with different radii and the rotational movement around the axis by accordingly colored arrows. The particle motion in longitudinal direction is illustrated by the side view in Fig. 6b and the dominant movement direction is indicated by arrows.

Additionally, in Fig. 7, the discussed flow structures are shown and the position as well as the orientation of the cameras are highlighted in blue. For this test case, three cameras are installed half-height on the horizontal plane with the same distance R_c to the cube's center and an angular displacement of 120° . One of the cameras is positioned at the front of the volume, normal to the X - Y -plane, and the other two cameras are arranged accordingly. The fourth camera is located above the synthetic experiment with the central line of sight in Y -direction, also with the distance R_c to the cube's center.

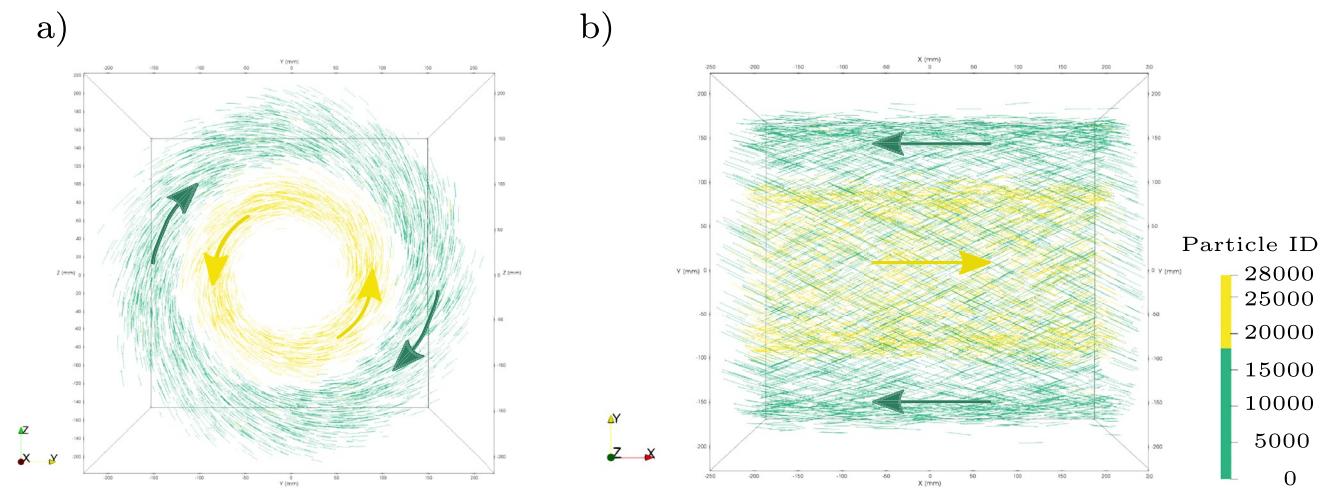
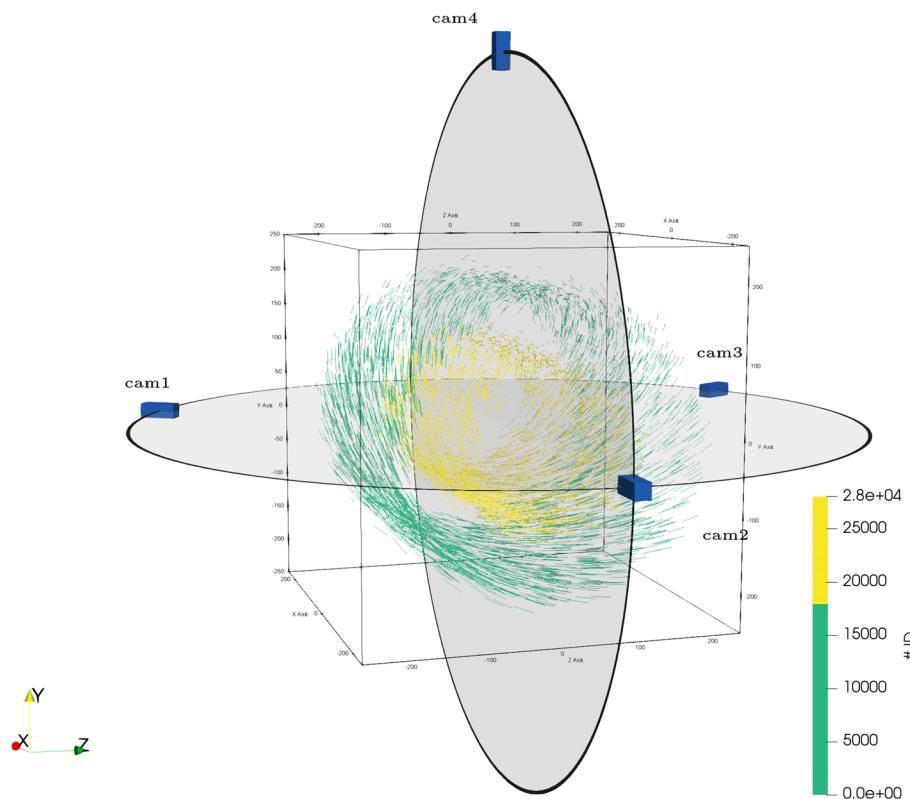


Fig. 6 Particle distribution organized in two counter-rotating aligned annular structures with opposite mean flow directions. Particles belonging to structure A_1 are colored in green and those forming A_2 are highlighted in yellow. In **a**, the perspective is chosen along the

X axis to illustrate the rotational movement around the axis which is also indicated by accordingly colored arrows. In **b**, the view along the Z -axis reflects the movement in X direction with the dominant movement direction indicated by arrows

Fig. 7 Four cameras shown in blue located on planes indicated by large circles surrounding the artificial flow structures of the synthetic case. Particles belonging to structures A_1 and A_2 are colored in green and yellow, respectively



A representative value of the ppp density is determined counting the number of particles in an area of 100×100 pixels. Note that the ppp densities determined with the images of the other cameras lead to nearly the same ppp values. Starting from 0.01 ppp for the lowest number of particles (7500), a particle density of 0.106 ppp is reached for the highest number of particles (45000).

In this synthetic experiment, the mean velocity magnitude is $\bar{V} = 3.8$ mm/time step. Considering, a typical PIV experiment of a turbulent pipe flow relying on a PCO Dimax.HS4 camera operating at 7000 Hz in rolling shutter, the above-mentioned mean velocity can be specified by $\bar{V} = 21$ m/s. For a turbulent flow of air in a pipe with a diameter of 0.35 m, this leads to a bulk Reynolds number, see, e.g., [7], of $Re = 440493$, which is in the range of state-of-the-art turbulent pipe flow experiments.

Particle Tracking

The proposed particle tracking approach is used to track the particle motion in comparison to the ground truth of the particle trajectories generated in the above-defined synthetic test case. To evaluate the performance of the method, the percentage of matched particles (pmp) is introduced and analyzed for particle densities between 0.015 ppp and 0.107 ppp. Two pixel step widths per time step, $\delta_s = 7$ and 14 px, which represent the mean particle velocities and thus

the temporal resolution, are investigated. Each reconstructed and true particle is identified by a unique ID. The earlier is classified as matched, if it is within 1.5 mm of a true particle and both particle IDs persist over time. The pmp is then defined by the number of matched particles compared to the true total particle number, and a reconstructed particle $X_{\text{rec}} \in \mathbb{R}^3$ is called matched if there $\exists : X_{\text{truth}} \in \mathbb{R}^3$ in the ground truth data, such that: $\|X_{\text{rec}} - X_{\text{truth}}\|_2 \leq 10^{-3}$, where $\|\cdot\|_2$ denotes the euclidean distance. A double assignment of an X_{truth} is not permitted.

The employed input parameters used for the framework are summarized in Table 1 including links to their definition in “Methods”.

A basic requirement for the method is the reconstruction of large flow structures. Figure 8 shows the reconstructed structures for the synthetic data set at time step 50, where parts of the reconstructed particle trajectories extending over 10 time steps are depicted. They are color-coded with

Table 1 Summary of the parameter settings for evaluation with the framework

Parameter	Definition	Value
ϵ_d	Fig. 3	1
ϵ_S	Sect. 2.3	1
ϵ_A	Sect. 2.3	3
ϵ_{back}	Eq. 18	1.5
gap	Sect. 2.3	1

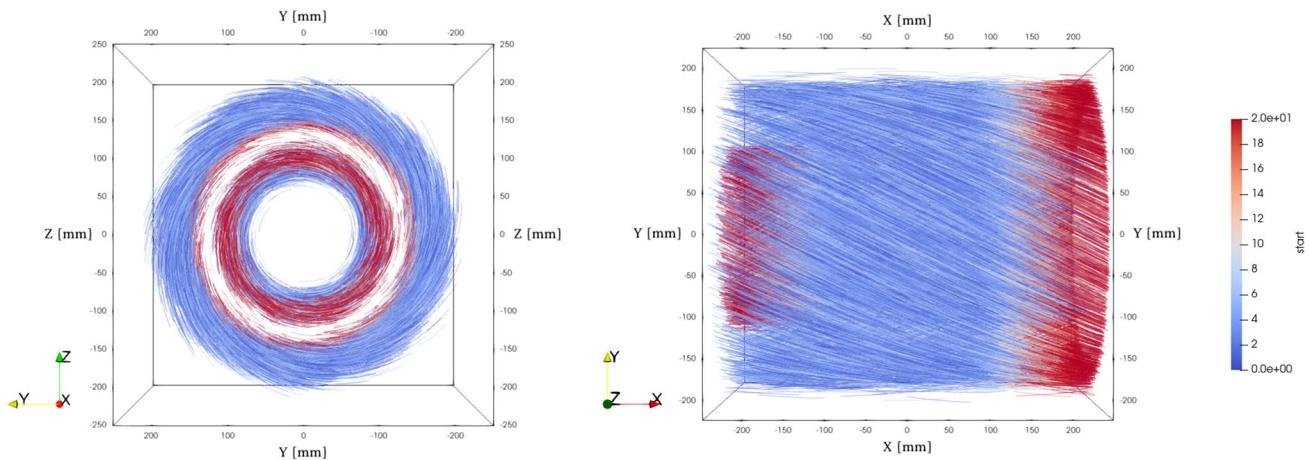


Fig. 8 Reconstruction of the aligned annular structures from the synthetic data set at time step 50. Particle trajectories reflecting 20 time steps are depicted and color-coded with the starting time step of the particle trajectory

starting time step of the particle trajectory. The comparison with the synthetically generated structures shown in Fig. 6 reveals the qualitative agreement. Furthermore, most of the trajectories are colored in blue and started accordingly at a time step close to $t = 0$. This proves the capability of the track initialization and maintaining long trajectories. Only the two regions for the newly entering particles in Fig. 6b (left for outer structure and right for the inner one) have higher starting time steps up to $t = 50$ reflected by the white to red colors.. This highlights the ability of adding new particles and trajectories and at the same time maintaining long trajectories. The quantitative differences are discussed in the following.

It is well known for PTV methods that a certain number of time steps are required to accumulate all particles for the tracking system in the first processing step. Thus, an important quality criterion for those methods is the number of required time steps to acquire the maximum possible

pmp. Figure 9a shows the pmp and the number of matched particles obtained for 28000 true particles as a function of the time step. A close-up view of the first 20 time steps in the lower right corner is presented. After 9 time steps, a plateau of about 90% matched particles corresponding to a total number of 25200 particles is reached. It should be noted that the method does not reconstruct ghost particles. Additionally, a significant length of the reconstructed tracks is necessary to obtain a sufficient amount of connected information for the paths of the particles. Figure 9b shows the frequency distribution of the track lengths measured in time steps. For short track lengths of 0 to 70 time steps, a low frequency of $0.2 \cdot 10^8$ is reported with a large increase to $1.6 \cdot 10^8$ and $1.2 \cdot 10^8$ for longer tracks of 70 and 90 time steps, respectively. Considering an average velocity of 4.1mm/s and a sample length of 500, a maximum of 120 time steps could be reached before the particles leave the domain, indicating that some time steps are necessary to add the particles to stable

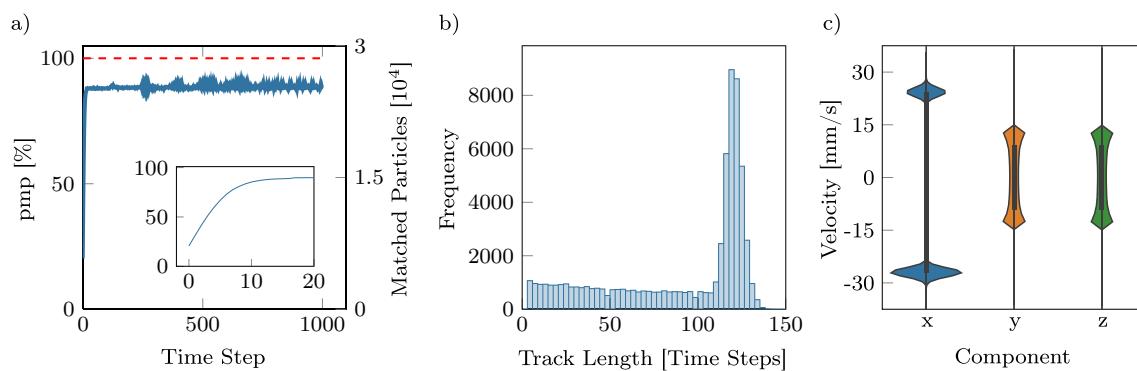


Fig. 9 Statistics for the synthetic data set containing 28,000 true particles. **a** Shows the number of matched tracks and pmp as a function of the time step in blue. The true particles numbers are indicated as a red line. **b** Shows the frequency distribution of the track length. **c**

Shows the violin plot of the velocities of the tracked particles, where the estimated probability density is given by the colored shape, the interquartile range by the thicker black line, and the whole data range by the thinner line

tracks and some tracks break, while the particle is advected through the volume. Figure 9c shows a violin plot of the three velocity components. The colored area is normalized thus showing the relative distribution per component. The X component exhibits two separated agglomerations between ± 3.5 and 4mm/s . The contribution to the negative values is larger compared to the positive values. This results from the higher particle number in the outer structure with negative movement direction in X . The distributions for the Y and Z velocity components are similar with two agglomerations at the higher velocities. This is due to the circular motion coupling both components: When the velocity is high in Y , it is low in Z and vice versa. Both the qualitative behavior and quantitative distribution comply with the ground truth.

To demonstrate the performance of the framework, the matched particles are evaluated in terms of the ppp. In this respect, Fig. 10 reflects the matched particles and pmp as a function of the true particle number. The curve starts close to 100 % pmp for the lowest particle number. For an increasing ppp, a steady decrease of the pmp is observed. For ppp = 0.05, the pmp is about 92.5 %. For a high ppp of 0.09, a fluctuation from the otherwise steady decrease is reported. Finally, for the highest ppp of 0.11, a pmp of 80 % is achieved. An increased particle velocity is investigated represented by the step width per time step of $\delta_S = 14\text{px}$ reflected by the orange line in Fig. 10. The general behavior of this reconstruction efficiency is similar to the previous case. For the lowest particle density of 0.005ppp, an efficiency of 97 % pmp is reported with a steady decrease to 77 % pmp for a high ppp of 0.1. While for the lowest particle number, the difference between the two particle velocities is about 1 % pmp, it increases to about 2.5 % for the highest ppp. The general decrease of the pmp is not surprising, since the association of particles with stable tracks becomes more ambiguous for increasing ppp. Still, while pmp= 99% for ppp= 0.007 corresponds to stable information from 7000

particle tracks, a significant increase in information density is achieved with 34000 stable tracks for the highest ppp.

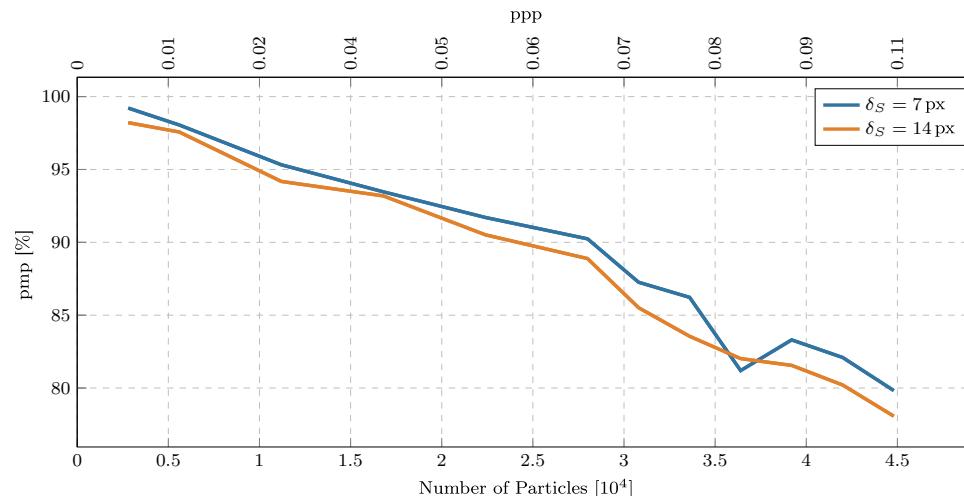
The results were calculated on three different machines, two machines with an AMD Ryzen Threadripper 1950X with 16 (32 threads) cores and 64G random-access memory (RAM) and the third machine with a Intel Xeon Platinum 8268 with 24 cores (48 threads) with 256G RAM. On average (over all cases), about 2981 s are needed to reconstruct one time step, when running on one thread on the Intel Xeon Platinum 8268 and 73.7 s when running on 48 threads. Comparing with the Ryzen Threadripper 1950X, it is 2424 s and 89.7 s. It should be noted, however, that the execution time depends heavily on how often it is possible to extend trajectories and how many particles have to be triangulated per time step. While the extension of n trajectories increases the execution time almost linearly, the execution increases according to a cubic law when n new particles have to be triangulated and assembled into trajectories. Especially in the cases with high particle densities, the execution time per time step can differ by orders of magnitude. During the parallel execution of the code 8G to 12.3 G of RAM was allocated.

In general, the proposed method is applicable for the investigated parameter range of $\text{ppp} \leq 0.100$ and $\delta_S \leq 14\text{px}$. The following section presents the applicability to experimental data.

Applications

In the following subsections, the applicability of the framework to experimental data is demonstrated and the respective results are presented. In subsection “[Dictyostelium Discoideum Motion](#)”, the tracking of the motion of *Dictyostelium discoideum* cells in a 2D micro-scale application is investigated. In subsection “[Turbulent Rayleigh–Bénard](#)

Fig. 10 Number of matched particles and pmp as a function of the true particle number for two $\delta_S = 7$ and 14 px



[Convection](#)”, the results of the flow measurement in 3D confined thermal convection cell are discussed.

Dictyostelium Discoideum Motion

As one of the experimental cases, we investigate the amoeba cells *Dictyostelium discoideum* (*Dd*), a key model organism for the study of eukaryotic migration. Besides chemical gradients, *Dd* cells have the ability to detect electric fields of continuous current and respond to them with a directed migratory movement toward the cathode.

Experiment

In a previous study, we applied the electric field to the cells seeded into a microfluidic channel and reversed the polarity of the electric field every 30 min. We observed a strong cellular response to the electric stimulation [19]. Indeed, the cells show a directed migration toward the cathode and inverted their trajectory when the polarity of the electric field was reversed.

This experiment is an example to illustrate how our framework deals with cells that promptly change their shape and partly overlap. Especially, challenging for a particle tracking algorithm is the abrupt reversal of the movement direction of the cells in addition to the stochastic component of the directed walk. The movement of the *Dd* cells is observed in a total field of view of $660 \times 660 \mu\text{m}^2$ using a bright-field inverted microscope with one mounted camera. The recording frequency is 0.05 Hz, acquiring one image frame every 20 s.

Cell Preparation and Experimental Setup

The cell preparation and the experimental setup are described in detail elsewhere [19]. Briefly, wild-type AX2 *Dictyostelium discoideum* cells were starved in shaking phosphate buffer (PB) for 5 h. During this time, the cell culture was pulsed with 50 nM cAMP (Sigma) every 6 min. An aliquot of the cell suspension was then injected into the experimental chamber for the electrotactic test. The cells were allowed to spread on the glass substrate for 15 min at 22 °C. A direct current voltage of 10 V/cm was applied to the cells through agar bridges and Ag/AgCl electrodes. The polarity of the electric field was reversed every 30 min by a programmable switch device (Siemens). Standard soft lithography was used to produce microfluidic channels, which were 1.5 mm wide, 100 μm high, and 30 mm long. We use an inverted microscope (Olympus IX-71) in bright field with a DeltaVision imaging system (GE Healthcare) to observe the cells in the microfluidic channel. The cellular migration was recorded with a CCD camera (CoolSNAP HQ2, Photometrics). Data acquisition started 20 s after

the application of electric field and the cell images were acquired every 20 s for 2 h. In each experiment, about 50 cells could be tracked during their migration in the electric field.

Dictyostelium Discoideum Tracking

The movement of the cells is recorded by a single camera, thus reducing the employed input parameters for the framework as summarized in Table 2 including links to their definition in “[Methods](#)”.

The current position of the cells for the particle tracking is deduced from the particle images with the open source software ImageJ [42], as illustrated in Fig. 11. For the background subtraction, the *Trainable Weka Segmentation* [49] was used on the raw data, where the *Dd* cells were segmented to one class and everything else (like the background) to another. Like this, each frame was segmented individually. The segmentation was then applied as a mask to the raw images followed by the RSC method to determine the centers of the cell clusters.

The experimental applicability of the framework is demonstrated by tracking the *Dd* movement presented in Fig. 12. The results of the particle tracking for three time instants, $t = 2000$ s, $t = 6000$ s, and $t = 10000$ s, are shown.

To make quantitative statements on the *Dd* motion, the trajectories resulting from the tracking are further analyzed. As cells are no solid particles, they tend to aggregate with neighbor cells when they come close to each other. Subsequently, they are likely to split again and separately continue migrating and they may move out of the field of view. For this reason, the number of simultaneously tracked cells is not constant over time, see Fig. 13a. Starting at about 90 cells for $t = 0$ min, an increase in simultaneously tracked cells is reported up to 113 cells at frame $t = 65$. Afterward, the events described above yield fluctuations in the number, and on average, a decrease in simultaneously tracked cells down to 76 for frame $t = 474$ is reported. A total of 1059 tracks were generated and their distribution is shown in Fig. 13 b). 566 tracks have a length of ≤ 20 time steps, and the average track length is ≈ 48.3 with a standard deviation of ≈ 84 . The median track length is 17. 160 tracks have a length of more than 80 time steps.. As mentioned above, *Dd* cells have the ability to detect continuous current electric fields. Interestingly, this characteristic becomes evident in

Table 2 Summary of the parameter settings for evaluation with the framework

Parameter	Definition	Value
ϵ_4	Sect. 2.3	1.5
n_p	Sect. 2.1	5
ϵ_{back}	Eq. (18)	1.5
gap	Sect. 2.5	3

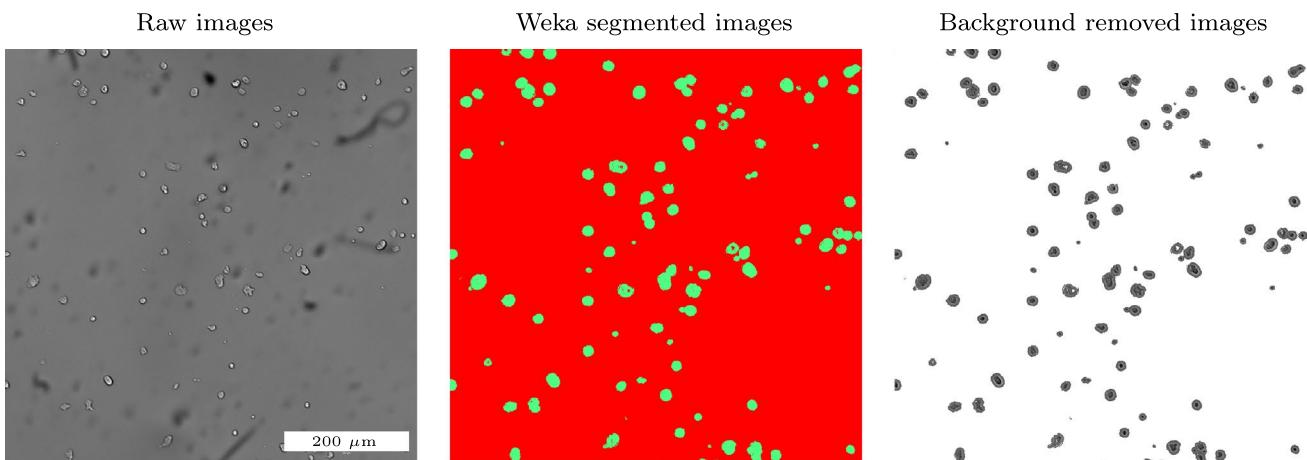


Fig. 11 Image preprocessing steps for the amoeba case. The raw data are shown on the right. To remove the background, the Weka segmentation [3] was applied to the raw data with default parameters. The green segments in the image are the amoeba, while the red seg-

ment is considered as the background or other irrelevant parts. The image on the right shows the background removed images, achieved using the segmented images as masks on the raw images

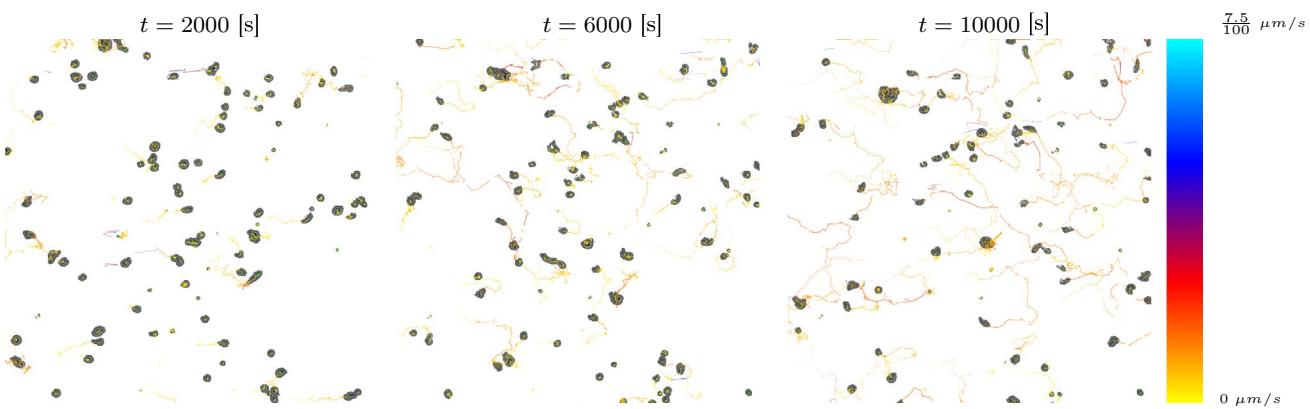


Fig. 12 Snapshots of the reconstructed tracks. The dots are the centers of the segments used for tracking. The segments of the tracks are color-coded in accordance with mean velocity of the track

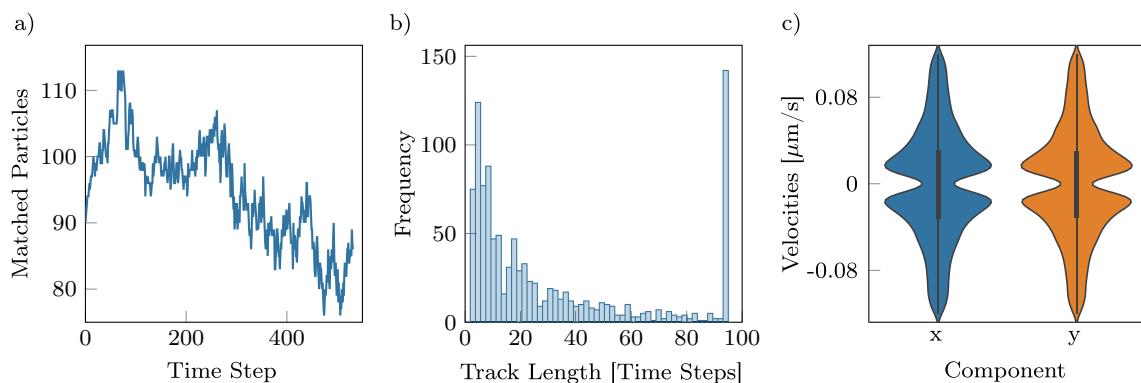


Fig. 13 **a** Number of simultaneously tracked amoebae depending on the time step. **b** Frequency distribution of the track length, whereas **c** shows a violin plot of the velocities

the velocity distribution of the pursued amoebas. The violin plot in Fig. 13c shows that the density distributions exhibit two extreme values. From this, it can be interpreted that the cells do not have one expected value in terms of velocity, but two. This is probably due to the reversed polarity of the electric field during the experiment. At this point, however, it should be noted that this work is not about generating insight into *Dd* dynamics and that the interpretation and validation of the two expected values is not part of the present study. We only want to demonstrate that our approach is suitable for evaluating this kind of data.

This case was calculated on workstation computer with an Intel Xeon Platinum 8268 with 24 cores and 256G of random-access memory (RAM). The time needed to reconstruct one time step was 8.3 s on one thread and 0.2 s on 48 threads. A total of 2.3G RAM was allocated during the reconstruction. A total of 496 time steps were evaluated in parallel in 86 s.

Turbulent Rayleigh–Bénard Convection

The other experimental case investigates PTV data acquired in a cubic Rayleigh–Bénard convection (RBC) sample with side length of $l = 500$ mm filled with water.

Experimental Setup

A sketch of the sample is shown in Fig. 14. The top and bottom of the experiment are embedded in an insulation mantle indicated by (a). The temperature difference ΔT is generated by the cooling and heating elements, (b) and (f), whose temperatures, T_C and T_H , are controlled by two water circuits. The developing fluid flow is measured between two anodised black aluminum plates, (c) and (e), enclosing the sample. The side walls (d) are made of 10 mm thick glass yielding high optical accessibility of the measurement volume. As light source, a high-power white-light LED array is used to

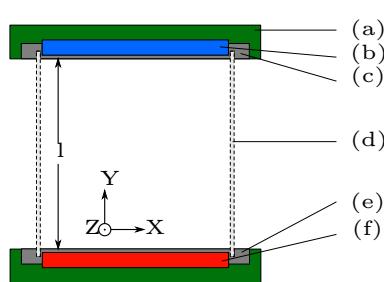


Fig. 14 Schematic of the cubic RBC sample along Z-direction. (a) marks the insulation layer, (b) and (f) indicate the cooling and heating elements, and (c) and (e) mark the anodised aluminum plates enclosing the sample. The glass cube (d) and the characteristic length l of the sample are indicated

illuminate TiO_2 -coated latex particles as flow tracers imaged by four cameras with a recording rate of 5Hz.

For the here considered case, the experimental settings are $T_C = 18^\circ\text{C}$, $T_H = 24^\circ\text{C}$, with an average sample temperature of $\bar{T} = 21^\circ\text{C}$ and $\Delta T = 6\text{K}$. The corresponding characteristic numbers are a Prandtl number of 6.9 and a Rayleigh number of $1.0 \cdot 10^{10}$. A full description of the experimental apparatus and setup can be found in [47].

Particle Tracking

The employed input parameters for the evaluation of this turbulent 3D flow with the HD-PTV method are summarized in Table 3 including links to their definition in “Methods”. Statistics are acquired by evaluating 250 time steps.

Here, the applicability of the framework is proven by the obtained particle trajectories presented in Fig. 15. For time step $t = 100$, the particles are presented as dots with the particle path for the previous 50 time steps attached as a tail. The latter is color-coded by the velocity magnitude. In Fig. 15a, a view along the Z-axis is chosen to demonstrate the reconstruction of particle tracks in the entire volume. In addition, a large-scale flow structure filling the convection sample is visible. This was also found for the same data set by [47] using tomographic PIV. New insight is gained by the reconstruction of a previously unresolved smaller corner flow reported at X and $Y \approx 50\text{mm}$. In Fig. 15b, the perspective is rotated to a diagonal plane of the sample to reflect the orientation of the large-scale circulation (LSC) more clearly. This perspective can also be seen in Fig. 16, where the velocity components are projected onto a Cartesian grid. For this projection, the mean velocities in a radius of 50mm are projected onto an equidistantly distributed grid over the whole volume. The grid points are illustrated as glyphs, where the orientation is given by the velocity and the size corresponds to the magnitude. The instantaneous velocity magnitudes range up to 20mm/s with the majority being significantly slower as reflected by the dominantly blue trajectories.

To make quantitative statements on the flow occurring within RBC, the trajectories resulting from the tracking are further analyzed. Contrary to the amoeba example from the previous case, the particle number for the enclosed RBC case is expected to be constant. This is confirmed in

Table 3 Summary of the parameter settings for the evaluation with the framework

Parameter	Definition	Value
ϵ_d	Fig. 3	2
ϵ_s	Sect. 2.3	1.7
ϵ_4	Sect. 2.3	3
ϵ_{back}	Eq. (18)	2.5
gap	Sect. 2.5	1

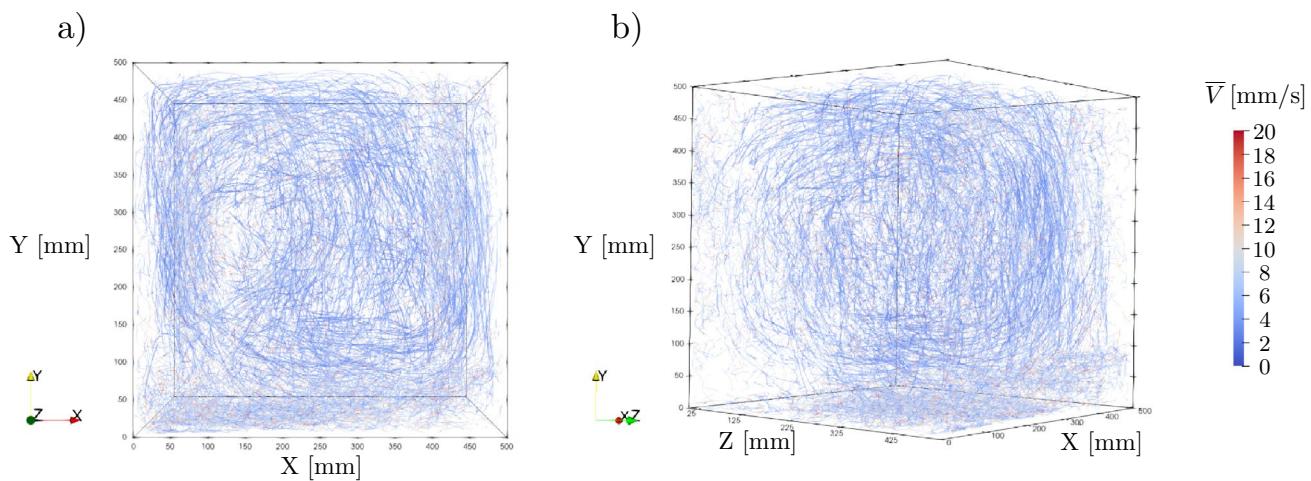


Fig. 15 Reconstruction of the RBC data at time step 100. Particle trajectories reflecting 50 time steps are depicted and color-coded with the velocity magnitude

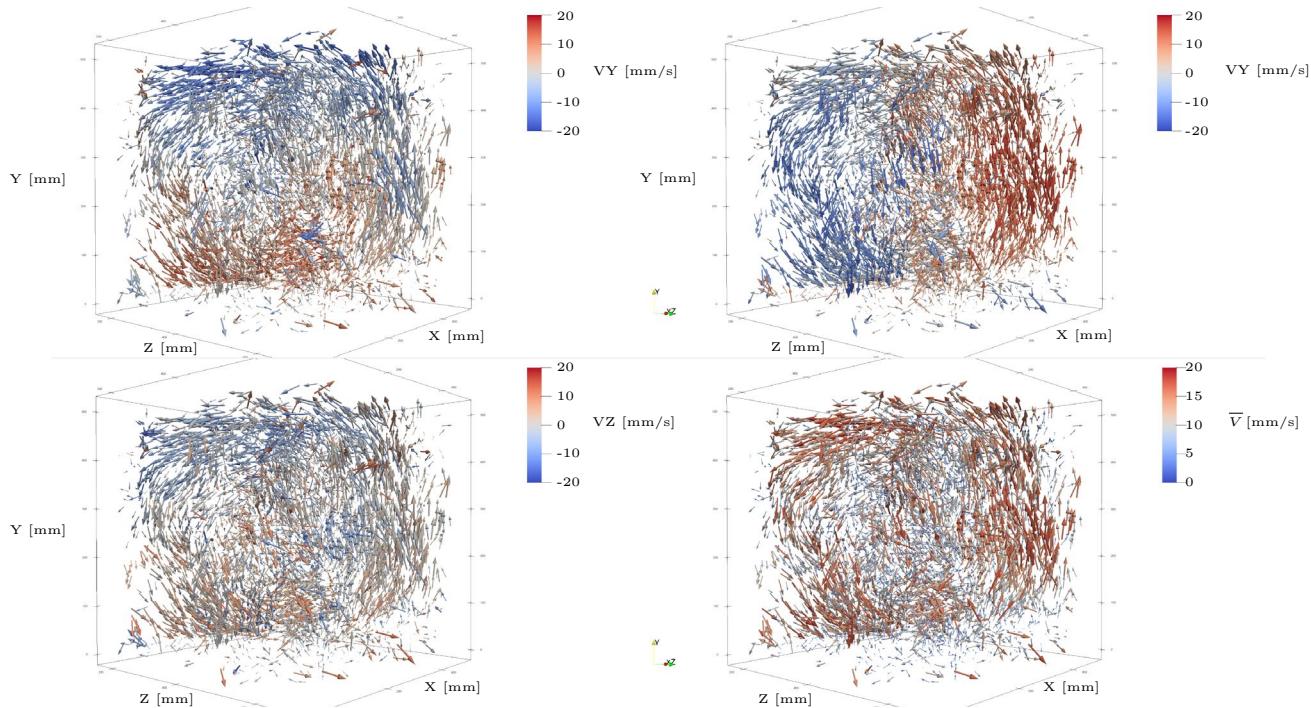


Fig. 16 Projection of the velocity components and the velocity magnitude onto a Cartesian grid, where the volume is clipped along the LSC

Fig. 17a. After only six time steps, the number of simultaneously tracked particles is 27472 with a stable plateau of about 28500 particles reached at time step 10.

For PTV methods, typically long trajectories are desired. Figure 17b reflects the frequency distribution of the track length. A distribution with a track length of up to 250 time steps and with the strongest contribution to smaller track length (≤ 50 time steps) is obtained. The evaluated data set

contained 250 time steps thus determining the limit which indicates that the longest tracks reached the potential distance. While the synthetic case showed a clear indication at a maximum track length determined by the system, the experimental case is expected to show less-pronounced dominant track length. The mean track length is 10.04 ± 10.03 and the median length is 7. Turbulent RBC is characterized by velocity distribution and can therefore be used as a

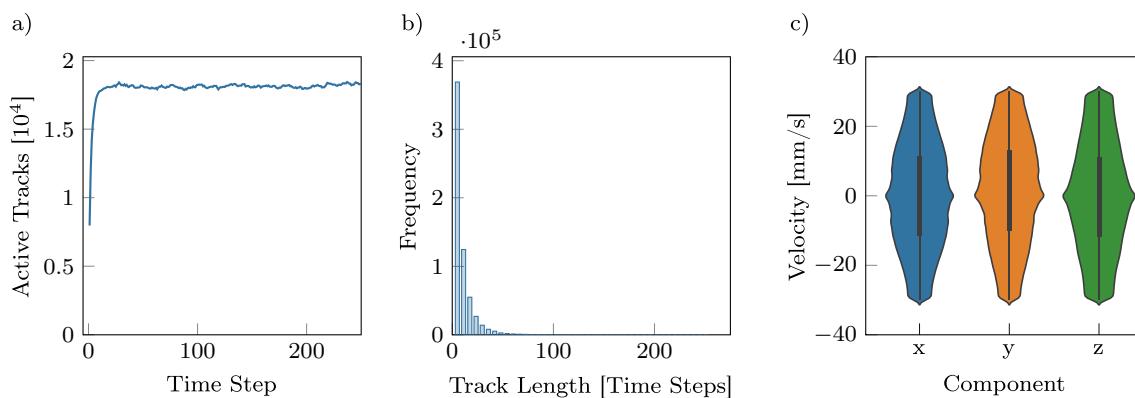


Fig. 17 Same figure structure as for Fig. 13 but for the RBC case. **a** Number of simultaneously tracked tracer particles in each time step. **b** Frequency distribution of the track length, whereas **c** shows a violin plot of the velocities

validation case for the presented PTV framework. A violin plot of the velocities is presented in Fig. 17c. Each velocity component shows a symmetric distribution. This is in agreement with the expectations in terms of a flow within an enclosed system driven by a dominant circular structure. The fact that the LSC is not exactly aligned with the diagonal plane explains the different spread in the frequency distribution of the velocity components.

The execution times per time step were also recorded for this case. The Intel Xeon Platinum 8268 achieved a average execution time of 2871 s per time step when running on one thread and 44 s when running with 48 threads. The comparatively higher execution time can be explained by the fact that much shorter trajectories are reconstructed for the experimental case than in the synthetic case. This implies that new particles have to be triangulated more often, which is more costly. In total, 10.5G RAM was allocated during the parallel execution.

The results justify the applicability of the developed framework for turbulent flows as it allows to reliably measure particle position, velocities, and their associated trajectories.

Conclusion

A new framework approach for particle tracking based on a Gaussian Mixture Model (GMM) is presented. It is divided into an initialization step and a processing step. In the first step, the velocities at the initial points are determined after iterative reconstruction of individual particles of the first four images in order to generate the tracks between these initial points without using tomographic particle image velocimetry. Subsequently, the tracks are extended by searching and including new points obtained from consecutive images using iterative reconstruction of individual particles in combination with continuous modeling of the particle

trajectories with a Gaussian mixture model considering the position and velocity as interdependent quantities.

The presented approach is validated and benchmarked by tracking particles generated in a synthetic generalized turbulent pipe flow which defines the ground truth. Considering this synthetic case, the approach returns about 90% matched particles after 9 time steps without generating any ghost particles. Furthermore, starting with 100% of matched particles for the lowest particle density considered, the percentage of matched particles (pmp) decrease from 92.5% for a particle density in terms of particle per pixels (ppp) of 0.05 to a pmp of 80% at the highest ppp of 0.11 for a step width per time step of $\delta_S = 7px$. Increasing the step width per time step from $\delta_S = 7px$ to $\delta_S = 14px$ displacement results in a similar declining curve and pmp values that are generally 5% lower.

Finally, the approach is successfully applied to two well-known tracking problems. The first one is tracking the eukaryotic migration of amoeba cells for which 1059 tracks were generated with tracks lengths of up to 100 time steps. The second one is turbulent Rayleigh–Bénard convection for which the motion of about 28500 particle is visualized using tracks of lengths up to 250 times steps.

The hyperparameters presented in this work are threshold values defining spherical search environments to limit the number of candidates for trajectory extension as well as for triangulation task of new particles. Thus, the hyperparameters are currently determined with the aim to maximize the trajectory length. In future investigations, however, the hyperparameters will be optimized regarding the multi-objectives trajectory length, smoothness of the trajectory, and computational requirements like memory and execution time by developing a method for estimating the parameters automatically, e.g., via a Kalman filter [55]-based algorithm like in this example on robot motion planning [33], so that less experts knowledge have to be provided. Classical approaches like grid search or gradient-based approaches would require a ground truth to

calculate possible errors metrics, but such a ground truth is not present in experimental data.

Considering the presented results, the introduced framework can reconstruct trajectories from both, guided and Brownian motion systems, even under high particle densities. The introduced GMM may be more difficult to implement as a simple regression method or polynomials, but it also have some significant advantages over the classical representation of trajectories by, e.g., polynomials (like in [46]) or linked lists. The GMM we use is designed to be as data-driven as possible, thus preventing the user from adding too much expert knowledge; leading to a mathematical model of the data with normal distributions. Connecting the field of particle tracking and Gaussian regression gave us several advantages which can be divided into three categories.

Computational efficiency The way the GMM is formulated allows both the start and end points to be changed without recalculating the whole system. This allows that in a case where there are several candidates for a trajectory, they can be tried efficiently. Especially in cases where there is a lot of ambiguity, as in the PTV cases presented, this has a very big advantage on the overall runtime of the system.

Compact representation As described in Eq. (17) a trajectory can be described completely through its weight vector. Starting from a given start point, only the weight vector needs to be saved to generate trajectories with arbitrary time steps.

Robustness and uncertainty quantification As described in the Methods section, particles are first triangulated and then assembled into trajectories. The triangulation itself is already dependent on the image processing, followed by the epipolar search and finally the numerical solution of the triangulation problem. In each of these steps, errors can occur both algorithmically (e.g., the quality of the raw data is not sufficiently high) and numerically due to the float arithmetic. The here presented method bypasses the problem of the error propagation by inferring probability distributions as a motion model. This makes additional algorithmically steps for smoothing or estimating the errors unnecessary. Conversely, the estimated probability distributions can also be used for uncertainty estimation.

Acknowledgements The authors like to thank Annika Köhne for proofreading this work.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of Interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Adrian R. Multi-point optical measurements of simultaneous vectors in unsteady flow- a review. *Int J Heat Fluid Flow*. 1986;7:127–45.
- Aref H. Chaotic advection of fluid particles. *Philos Trans R Socf Lond Ser A: Phys Eng Sci* 1990;333(1631):273–88.
- Arganda-Carreras I, Kaynig V, Rueden C, Eliceiri KW, Schindelin J, Cardona A, Seung SH. Trainable weka segmentation: a machine learning tool for microscopy pixel classification. *Bioinformatics*. 2017;33(15):2424–6.
- Atkinson C, Soria J. An efficient simultaneous reconstruction technique for tomographic particle image velocimetry. *Exp Fluids*. 2009;47:563–78.
- Ban Z, Liu J, Cao L. Superpixel segmentation using gaussian mixture model. *IEEE Trans Image Process*. 2018;27(8):4105–17.
- Barron JL, Fleet DJ, Beauchemin SS. Performance of optical flow techniques. *Int J Comput Vision*. 1994;12(1):43–77. <https://doi.org/10.1007/BF01420984>.
- Bauer C, Feldmann D, Wagner C. On the convergence and scaling of high-order statistical moments in turbulent pipe flow using direct numerical simulations. *Phys Fluids*. 2017;29(12):125105. <https://doi.org/10.1063/1.4996882>.
- Bradski, G. The OpenCV Library. Dr. Dobb's Journal of Software Tools. 2000
- Cardwell N, Vlachos P, Thole K. A multi-parametric particle-pairing algorithm for particle tracking in single and multiphase flows. *Meas Sci Technol*. 2011;22:105406.
- Crawford A. The use of gaussian mixture models with atmospheric lagrangian particle dispersion models for density estimation and feature identification. *Atmosphere*. 2020;11(12):1369.
- Curtin RR, Edel M, Lozhnikov M, Mentekidis Y, Ghaisas S, Zhang S. mlpack 3: a fast, flexible machine learning library. *J Open Source Softw*. 2018;3:726. <https://doi.org/10.21105/joss.00726>.
- Dagum L, Menon R. Openmp: an industry-standard api for shared-memory programming. *IEEE Comput Sci Eng*. 1998;5(1):46–55. <https://doi.org/10.1109/99.660313>.
- Dutta A, Mondal A, Dey N, Sen S, Moraru L, Hassanien AE. Vision tracking: a survey of the state-of-the-art. *SN Comput Sci*. 2020;1(1):1–19.
- Eliceiri KW, Berthold MR, Goldberg IG, Ibáñez L, Manjunath BS, Martone ME, Murphy RF, Peng H, Plant AL, Roysam B, Stuurman N, Swedlow JR, Tomancak P, Carpenter AE. Biological

- imaging software tools. *Nat Methods*. 2012;9(7):697–710. <https://doi.org/10.1038/nmeth.2084>.
- 15. Elsinga G, Scarano F, Wieneke B, van Oudheusden B. Tomographic particle image velocimetry. *Exp Fluids*. 2006;41:933–47.
 - 16. Folk M, Cheng A, Yates K. Hdf5: A file format and i/o library for high performance computing applications. In: Proceedings of Supercomputing. 1999; 99:5–33.
 - 17. Fuchs T, Hain R, Kähler CJ. Non-iterative double-frame 2D/3D particle tracking velocimetry. *Exp Fluids*. 2017;58:119. <https://doi.org/10.1007/s00348-017-2404-0>.
 - 18. Garcia V, Nielsen F, Nock R. Levels of details for gaussian mixture models. In: Asian Conference on Computer Vision, pp. 514–525. Springer. 2009.
 - 19. Guido I, Diehl D, Olszok NA, Bodenschatz E. Cellular velocity, electrical persistence and sensing in developed and vegetative cells during electrotaxis. *PLoS ONE*. 2020;15:1–16. <https://doi.org/10.1371/journal.pone.0239379>.
 - 20. Hartley RI, Sturm P. Triangulation. In: Computer Analysis of Images and Patterns: Proc. of 6th International Conference CAIP '95, pp. 190–197. Springer, Prague, Czech Republic. 1995.
 - 21. Herman G, Lent A. Iterative reconstruction algorithms. *Comput Biol Med*. 1976;6:273–94.
 - 22. Huang TS, Tsai R. Image sequence analysis: Motion estimation. In: Image sequence analysis, pp. 1–18. Springer. 1981.
 - 23. Jahn T, Schanz D, Schröder A. Advanced iterative particle reconstruction for lagrangian particle tracking. *Exp Fluids*. 2021;62(8):1–24.
 - 24. Kreizer M, Ratner D, Liberzon A. Real-time image processing for particle tracking velocimetry. *Exp Fluids*. 2010;48(1):105–10. <https://doi.org/10.1007/s00348-009-0715-5>.
 - 25. Lo CM, Wang HB, Dembo M, Wang Yl. Cell movement is guided by the rigidity of the substrate. *Biophys J*. 2000;79(1):144–52. [https://doi.org/10.1016/S0006-3495\(00\)76279-5](https://doi.org/10.1016/S0006-3495(00)76279-5). <http://www.sciencedirect.com/science/article/pii/S0006349500762795>.
 - 26. Lucy L. An iterative technique for the rectification of observed distributions. *Astron J*. 1974;79:745–54. <https://doi.org/10.1086/111605>.
 - 27. Maas H, Gruen A, Papantoniou D. Particle tracking velocimetry in three-dimensional flows. *Exp Fluids*. 1993;15(2):133–46.
 - 28. Malik N, Dracos T, Papantoniou D. Particle tracking in three dimensional turbulent flows—part ii: particle tracking. *Exp Fluids*. 1993;15:279–94.
 - 29. Malik NA, Dracos T, Papantoniou D. Particle tracking velocimetry in three-dimensional flows, Part II: Particle tracking. *Exp Fluids*; 1993.
 - 30. Martins I, Carvalho P, Corte-Real L, Alba-Castro JL. Bmog: boosted gaussian mixture model with controlled complexity for background subtraction. *Pattern Anal Appl*. 2018;21(3):641–54.
 - 31. McCaig CD, Rajnicek AM, Song B, Zhao M. Controlling cell behavior electrically: current views and future potential. *Physiol Rev*. 2005;85(3):943–78. [https://doi.org/10.1152/physrev.00020.2004 \(PMID: 15987799\)](https://doi.org/10.1152/physrev.00020.2004).
 - 32. Mikheev A, Zubtsov V. Enhanced particle-tracking velocimetry (eptv) with a combined two-component pairmatching algorithm. *Meas Sci Technol*. 2008;19:085401.
 - 33. Mohanan M, Salgaonkar A. Probabilistic approach to robot motion planning in dynamic environments. *SN Comput Sci*. 2020;1:1–16.
 - 34. Moon TK. The expectation-maximization algorithm. *IEEE Signal Process Mag*. 1996;13(6):47–60.
 - 35. Muthu JS, Murali P. Review of chaos detection techniques performed on chaotic maps and systems in image encryption. *SN Computer Science*. 2021;2(5):1–24.
 - 36. Nishino K, Kasagi N, Hirata M. Three-dimensional particle tracking velocimetry based on automated digital image processing. *Trans ASME J Fluid Eng*. 1989;111:384–90.
 - 37. Novara M, Schanz D, Gesemann S, Lynch K, Schröder A. Lagrangian 3d particle tracking for multi-pulse systems: performance assessment and application of shake-the-box. In: 18th International Symposium on the Application of Laser and Imaging Techniques to Fluid Mechanics, 2016; 2638–2663.
 - 38. Parthasarathy R. Rapid accurate particle tracking by calculation of radial symmetry centers. *Nat Methods*. 2012;9(7):724–6.
 - 39. Rathi Y, Vaswani N, Tannenbaum H, Yezzi A. Tracking deforming objects using particle filtering for geometric active contours. *IEEE Trans Pattern Anal Mach Intell*. 2007;29(8):1470–5. <https://doi.org/10.1109/TPAMI.2007.1081>.
 - 40. Reynolds D. Gaussian mixture models. Encyclopedia of biometrics. 2015; 827–832.
 - 41. Richardson WH. Bayesian-based iterative method of image restoration*. *J Opt Soc Am*. 1972;62(1):55–9. <https://doi.org/10.1364/JOSA.62.000055>. <http://www.osapublishing.org/abstract.cfm?URI=josa-62-1-55>.
 - 42. Rueden CT, Schindelin J, Hiner MC, DeZonia BE, Walter AE, Arena ET, Eliceiri KW. Image J2: imageJ for the next generation of scientific image data. *BMC Bioinf*. 2017;18(1):529. <https://doi.org/10.1186/s12859-017-1934-z>.
 - 43. Sanderson C, Curtin R. Armadillo: a template-based c++ library for linear algebra. *J Open Source Softw*. 2016;1(2):26. <https://doi.org/10.21105/joss.00026>.
 - 44. Sanderson C, Curtin R. A user-friendly hybrid sparse matrix class in c++. In: Davenport JH, Kauers M, Labahn G, Urban J, editors. Mathematical Software - ICMS 2018. Cham: Springer International Publishing; 2018. p. 422–30.
 - 45. Schanz D, Gesemann S, Schröder A. Shake-the-box: Lagrangian particle tracking at high particle image densities. *Exp Fluids*. 2016;57(5):70.
 - 46. Schanz D, Schröder A, Gesemann S, Michaelis D, Wieneke B. Shake the box: A highly efficient and accurate tomographic particle tracking velocimetry (tomo-ptv) method using prediction of particle positions. In: PIV13. Delft, The Netherlands. 2013.
 - 47. Schiepel D, Bosbach J, Wagner C. Tomographic particle image velocimetry of turbulent Rayleigh-Bénard convection in a cubic sample. *J Flow Vis Image Process*. 2013;20(1–2):3–23.
 - 48. Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, Preibisch S, Rueden C, Saalfeld S, Schmid B, Tinevez JY, White DJ, Hartenstein V, Eliceiri K, Tomancak P, Cardona A. Fiji: an open-source platform for biological-image analysis. *Nat Methods*. 2012;9(7):676–82. <https://doi.org/10.1038/nmeth.2019>.
 - 49. Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, et al. Fiji: an open-source platform for biological-image analysis. *Nat Methods*. 2012;9(7):676–82.
 - 50. Solooff SM, Adrian RJ, Liu ZC. Distortion compensation for generalized stereoscopic particle image velocimetry. *Meas Sci Technol*. 1997;8(12):1441.
 - 51. Tapia HS, Aragon JG, Hernandez DM, Garcia BB. Particle tracking velocimetry (ptv) algorithm for non-uniform and non-spherical particles. In: Electronics, Robotics and Automotive Mechanics Conference (CERMA'06), vol. 2, pp. 325–330. IEEE. 2006.
 - 52. Trepaut X, Chen Z, Jacobson K. Cell Migration, pp. 2369–2392. American Cancer Society. 2012. <https://doi.org/10.1002/cphy.c110012>.
 - 53. Ulman V, et al. An objective comparison of cell-tracking algorithms. *Nat Methods*. 2017;14(12):1141–52. <https://doi.org/10.1038/nmeth.4473>.

54. Van Haastert PJM, Devreotes PN. Chemotaxis: signalling the way forward. *Nat Rev Mol Cell Biol.* 2004;5(8):626–34. <https://doi.org/10.1038/nrm1435>.
55. Welch G, Bishop G, et al. An introduction to the kalman filter. 1995.
56. Wieneke B. Volume self-calibration for 3D particle image velocimetry. *Exp Fluids.* 2008;45(4):549–56. <https://doi.org/10.1007/s00348-008-0521-5>.
57. Wieneke B. Iterative reconstruction of volumetric particle distribution. *Meas Sci Technol.* 2013;24:024008.
58. Williams TL. The optical transfer function of imaging systems. England: Routledge; 2018.
59. Xie X, Huang W, Wang HH, Liu Z. Image de-noising algorithm based on gaussian mixture model and adaptive threshold modeling. In: 2017 International conference on inventive computing and informatics (ICICI), pp. 226–229. IEEE. 2017.
60. Xue Z, Charonko J, Vlachos P. Particle image velocimetry correlation signal-to-noise ratio metrics and measurement uncertainty quantification. *Measurement Science and Technology.* 2014;25(11):115301. <http://stacks.iop.org/0957-0233/25/i=11/a=115301>.

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