

Full Length Article

A high order accurate space-time trajectory reconstruction technique for quantitative particle trafficking analysis

Eloina Corradi ^{a,*}, Maurizio Tavelli ^b, Marie-Laure Baudet ^{a,*,1}, Walter Boscheri ^{c,d,1}^a Department of Cellular, Computational and Integrative Biology (CIBIO), University of Trento, Italy^b Faculty of Computer Science, Free University of Bozen, Italy^c Laboratoire de Mathématiques UMR 5127 CNRS, Université Savoie Mont Blanc, 73376 Le Bourget du Lac, France^d Department of Mathematics and Computer Science, University of Ferrara, Italy

ARTICLE INFO

Keywords:

Trajectory reconstruction
Particle trafficking
Transport analysis
High order polynomial reconstruction
Kinematics

ABSTRACT

The study of moving particles (e.g. molecules, virus, vesicles, organelles, or whole cells) is crucial to decipher a plethora of cellular mechanisms within physiological and pathological conditions. Powerful live-imaging approaches enable life scientists to capture particle movements at different scale from cells to single molecules, that are collected in a series of frames. However, although these events can be captured, an accurate quantitative analysis of live-imaging experiments still remains a challenge. Two main approaches are currently used to study particle kinematics: kymographs, which are graphical representation of spatial motion over time, and single particle tracking (SPT) followed by linear linking. Both kymograph and SPT apply a space-time approximation in quantifying particle kinematics, considering the velocity constant either over several frames or between consecutive frames, respectively. Thus, both approaches intrinsically limit the analysis of complex motions with rapid changes in velocity. Therefore, we design, implement and validate a novel reconstruction algorithm aiming at supporting tracking particle trafficking analysis with mathematical foundations. Our method is based on polynomial reconstruction of 4D (3D+time) particle trajectories, enabling to assess particle instantaneous velocity and acceleration, at any time, over the entire trajectory. Here, the new algorithm is compared to state-of-the-art SPT followed by linear linking, demonstrating an increased accuracy in quantifying particle kinematics. Our approach is directly derived from the governing equations of motion, thus it arises from physical principles and, as such, it is a versatile and reliable numerical method for accurate particle kinematics analysis which can be applied to any live-imaging experiment where the space-time coordinates can be retrieved.

1. Introduction

The use of fluorescent labels and the advance of imaging techniques over the last decades have enabled to capture several cellular and intracellular dynamic events in living samples. The space-time history of the position of moving particles is typically stored in images at different times, which are referred to as frames. Each frame is thus an image that captures particles and their surroundings,

* Co-corresponding authors.

E-mail addresses: eloina.corradi@unitn.it (E. Corradi), marielaura.baudet@unitn.it (M.-L. Baudet).

¹ Contributed equally to the paper as co-senior authors.

<https://doi.org/10.1016/j.amc.2024.128902>

Received 8 November 2023; Received in revised form 21 May 2024; Accepted 16 June 2024

Available online 27 June 2024

0096-3003/© 2024 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

at a given time. Along with the improvement of live-imaging techniques, several approaches, algorithms and analysis tools have been implemented to study moving particles, tracking their direction and quantifying their velocity. The current available tools for studying particle trafficking are kymograph and single particle tracking (SPT) followed by linear linking.

Kymographs are 2D images obtained by stacking the intensity profile along a manually defined path. The result is a graphical representation of spatial motion over time, where the slope of the lines represents the particle velocity along that given path. Kymographs are largely used to investigate particle velocity [1–5]. Although being a widespread technique, kymograph analysis simplifies the complexity of a tracking problem by approximating particle movements along a fixed spatial path, thus two disadvantages intrinsically arise. i) Spatial resolution reduction: by projecting a 3D trajectory onto one single direction and by approximating the kymograph traces with linear segments. ii) Temporal resolution reduction: the dynamics in time is described by tracks with constant behavior (e.g. constant segmental velocity over several acquisitions). All-in-all both space and time resolution are reduced.

Contrary to kymograph, tracking approaches aim at identifying the particle position in each acquisition of a time lapse and at connecting the identified particles over frames, preserving all the spatial information of the dynamics [6–11]. Indeed, most of the single-particle tracking (SPT) methods are based on a three-step process: i) detection of particles frame by frame (detection algorithms), ii) particle linking between consecutive frames [8,9] by solving an assignment problem [12,9] (linking/tracking algorithms), iii) reconstruction of the particle trajectory. Detection and tracking algorithms are continuously refined in order to increase the performance among different scenarios (e.g. high particle density, low signal-to-noise ratio, motion heterogeneity, blinking signal, merging or splitting events) [13–15]. Although none of the existing SPT tools tackles all these aspects optimally [10,11], the selection of one approach or the other enables scientists to study a huge variety of particle dynamics in different biological contexts preserving all the spatial information. Nevertheless, most current tracking techniques carry out the final particle trajectory reconstruction by performing a linear linking of identified particles, thus assuming a constant particle velocity between frames. This approximation might represent a limiting step in imaging analysis, especially for particles showing rapid changes in velocity, and it precludes the accurate analysis of particle co-trafficking, i.e. the analysis of a possible correlation of motion among different particles at the same time.

Therefore we develop and validate a novel algorithm, SHOT-R (Space-time High-Order Trajectory Reconstruction), for high-order particle trajectory reconstruction and kinematic analysis. While SPT followed by linear linking approximates the original trajectory by a piecewise first degree polynomial (P1) reconstruction (i.e. connecting dots with linear segments), the SHOT-R algorithm solves particle kinematics equations relying on higher order polynomials. Polynomial reconstruction is a rather general approach in the numerical solution of hyperbolic partial differential equations, mainly focused on advection equations [16–23], which is here employed for the first time to the live-imaging context. It is unknown how particles move between one acquisition and the next, and any kind of trajectory is an approximation of the real particle path. Nevertheless, here we show through mathematical tests and by applying SHOT-R on a real biological sample, that our method improves the accuracy of the reconstruction and, consequently, the reliability of the kinematic analysis. Indeed, the velocities computed with our algorithm, which accounts for curvilinear trajectories, result to be more accurate than the current SPT strategy. Moreover, our algorithm still achieves slightly better accuracy with respect to curvilinear spline reconstruction. Overall, we provide an accurate analysis platform for particle trafficking, which can be applied to any dataset where the space-time coordinates can be retrieved, thus being extremely versatile. Moreover, thanks to the SHOT-R unique feature of a high order continuous trajectory, novel quantitative information can be extracted (i.e. instantaneous velocity and acceleration).

Tracking and trajectory reconstructions have been investigated also in different fields of application not only restricted to biology. In [24], a nonlinear reconstruction algorithm is applied to flight paths, where the non-linearity is treated by means of a predictor-corrector approach in the numerical integration of the governing systems of ordinary differential equations describing the total aerodynamic forces, moments and angles. Local third order accurate interpolation techniques and second order finite difference operators have been forwarded in [25] with applications to animal flight paths, assuming that the acceleration of a particle is locally constant and avoiding the solution of the equations of motion. To reconstruct the experimental images related to bat flight, in [26] fifth order interpolations have been employed for retrieving the kinematics to fill gaps with no data available. Particle reconstruction techniques have been also forwarded in [27] in the context of experimental particle tracking velocimetry, proposing linear linking algorithms with the definition of a correlation radius that determines the neighborhood of a particle to be used for computing its expected value at the next time level.

In all the above mentioned works, linear reconstructions in the sense of Godunov [28] were used, meaning that the reconstruction of discontinuous solutions is not monotone. Since experimental data are typically not regularly distributed, they can be reasonably considered as a discontinuous function. Therefore, to ensure monotonicity, linear linking algorithms are preferred in the literature, because they are compliant with the discrete maximum principle by construction. However, they are restricted to second and first order of accuracy for position and velocity, respectively. To circumvent Godunov theorem and ensure essentially non-oscillatory properties, the reconstruction procedure must be nonlinear [29,30], thus data-dependent. Here, we present an interdisciplinary work that connects state-of-the-art techniques of numerical analysis with cutting-edge live-imaging analysis, that is currently demanding more sophisticated tools to post-process the acquired data. Indeed, the high order accuracy of our method potentially matches the high resolution acquisition techniques that are currently used to maximize the quality of the acquisitions in live-imaging [31–33]. Our method will therefore be presented and validated with the aim of being appreciated from both the mathematical and the biological angle of attack.

We remark that the method presented in this work is concerned with the third step of the kinematics analysis aforementioned, that is the reconstruction of the particle trajectory. This is a completely independent process from the experimental data acquisition or the tracking techniques which permit to connect frames in a common path related to a particle. Consequently, the starting

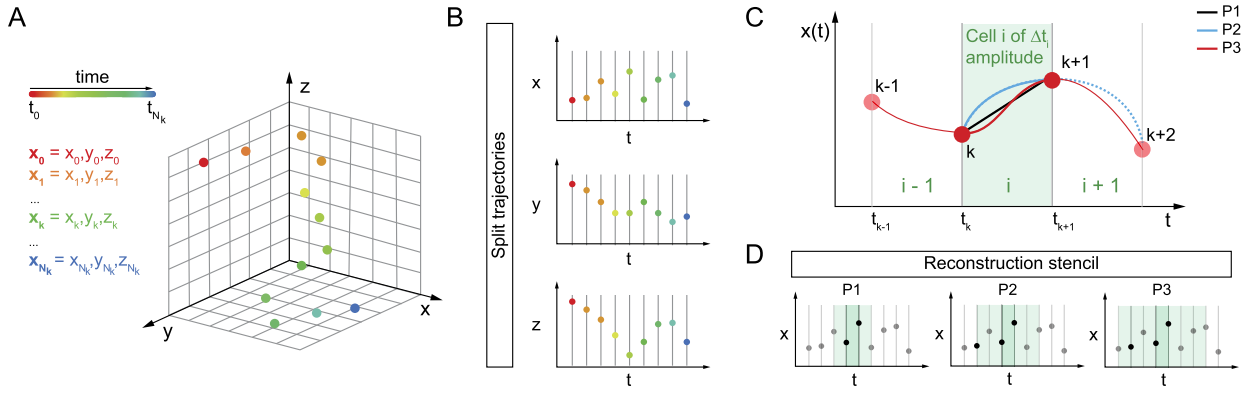


Fig. 1. Space-time high order trajectory reconstruction (SHOT-R) method (A) Schematic of SHOT-R 4D input data: particle motion is described as the evolution of the spatial coordinates (x, y, z) in time (from t_0 to t_{N_k}). (B) Schematic of space-time split trajectory strategy. Each direction (x, y, z) is analyzed separately, thus a multidimensional setting can be reduced to multiple one-dimensional problems. (C,D) Schematic of trajectory reconstruction based on polynomials (SHOT-R). An interpolation polynomial $p^N(t)$ of arbitrary degree N is constructed for each cell i (C) by considering the cell itself and a total number $N + 1$ surrounding elements (i.e. the reconstruction stencil) (D). Abbreviations: $P1, P2, P3$, first, second, third degree polynomial.

point of our method is a set of space-time coordinates associated to different particles, which have been obtained by means of experimental data and a subsequent tracking algorithm. We will show that the presented method does indeed not depend on those previous steps, by analyzing different detection and tracking algorithms to generate the input data for our new reconstruction scheme. Furthermore, an application to a particle trajectory in the field of hydraulics is forwarded, demonstrating the versatility of the presented approach.

The rest of the paper is organized as follows. In Section 2 we introduce the governing equations of motion, while the numerical scheme is fully detailed in Section 3. Numerical convergence studies and a large suite of validation tests are then shown in Section 4. A concluding section finalizes the article where we draw some conclusions and present an outlook to future research.

2. Mathematical model

Particle kinematics is described by ordinary differential equations (ODE) governing the motion:

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}(\mathbf{x}, t), \quad \mathbf{x}(t = t_0) = \mathbf{x}_0, \tag{1}$$

where t denotes the time coordinate, $\mathbf{x} = (x, y, z)$ is the position vector of spatial coordinates and $\mathbf{v}(\mathbf{x}, t) = (v_x, v_y, v_z)$ is the velocity vector with components along each direction in space. The trajectory starts at the initial time $t = t_0$ from the position \mathbf{x}_0 (Fig. 1A) and then the particle moves in time according to ODE (1).

The solution of the trajectory equation (1) is a continuous function, whereas experimental data only provide a discrete set of known spatial coordinates \mathbf{x}_k at the corresponding time t_k (Fig. 1A). Therefore, in order to study particle kinematics, a reconstruction of particle trajectories is needed.

3. Numerical scheme

In the sequel, we detail the derivation and implementation of the SHOT-R method. We assume that a series of frames is available, meaning that we can extract a sequence of points in space and time from a dataset composed of image acquisitions, independently of the nature of the phenomenon under consideration. Obviously, we also assume that the particle motion is always governed by the ODE (1). A graphical representation of the reconstruction procedure is shown in Fig. 1.

3.1. Split trajectories

In our method, to faithfully describe particle motions, the input data are the space-time coordinates of particles retrieved by the raw movies (i.e. the acquired images of the experiments) using available algorithms for detection and tracking (here TrackMate [34] is employed). Once the space-time coordinates (\mathbf{x}_k, t_k) are retrieved for each data point $k = 1, \dots, N_K$ (Fig. 1A), we proceed in a dimension-by-dimension manner by splitting the position vector $\mathbf{x} = (x, y, z)$ along each spatial direction (Fig. 1B). This approach speeds up and simplifies the computational process by decreasing the complexity of the problem itself (i.e. from a single 4D problem to multiple 1D ones), and avoiding the use of a 3D mesh, which is difficult to spatially discretize. At the same time, this strategy keeps the desired temporal and spatial resolution of fully 3D procedures.

Specifically, the issue related to a multidimensional spatial domain is tackled by considering the trajectory equations

$$\begin{cases} \frac{dx}{dt} = v_x(x, t), & x(t = 0) = x_0, \\ \frac{dy}{dt} = v_y(y, t), & y(t = 0) = y_0, \\ \frac{dz}{dt} = v_z(z, t), & z(t = 0) = z_0, \end{cases} \tag{2}$$

that directly derive from the original definition (1). Let observe that all equations in (2) are mathematically the same object, therefore we focus on the solution of a generic one-dimensional trajectory equation satisfied by a particle along the direction s , i.e.

$$\frac{ds}{dt} = v_s(s, t), \quad s(t = 0) = s_0, \tag{3}$$

where s can be referred to any spatial direction among \mathbf{x} and the same holds true for the velocity component v_s with \mathbf{v} . This makes our algorithm very general and applicable to a wide range of physical and biological phenomena. We stress that the only information needed as input are the point coordinates \mathbf{x}_k defined along the trajectory that has to be reconstructed.

Let D represent the spatial dimensionality of the problem under consideration, i.e. $D \in [1, 3]$. Consequently, $\mathbf{x}_k \in \mathbb{R}^D$, that is the spatial coordinates are given by real numbers in dimension D . Along each spatial direction we have the set of space-time coordinates (s_k, t_k) that are extracted from (\mathbf{x}_k, t_k) . What carried out for the generic direction s will be performed as many times as the dimensionality of the problem is matched, thus a genuinely multidimensional trajectory defined in dimension D is reduced to a total number of D one-dimensional trajectories. The full multidimensional trajectory can then be simply gathered back by collecting all one-dimensional contributions.

3.2. Computational mesh

The time coordinate t is defined in the time interval $T = [t_1; t_{N_K}]$, i.e. $t \in T$, while the space coordinate s is bounded in $\Omega = [s_1; s_{N_K}]$, i.e. $s \in \Omega$. The computational domain Ω is discretized with a total number of $N_J = N_K - 1$ cells with size Δt_i and barycenter time coordinate t_i given by

$$\Delta t_i = t_{i+1/2} - t_{i-1/2}, \quad t_i = \frac{t_{i+1/2} + t_{i-1/2}}{2}, \tag{4}$$

for $i = 1, \dots, N_J$, where the values at the interfaces are directly available from $t_{i-1/2} = t_k$ and $t_{i+1/2} = t_{k+1}$ (Fig. 1C). Let observe that index k is used for cell interfaces $i - 1/2$ in order to obtain a compatible notation on the computational mesh, thus one has the equivalence $k = i - 1/2$. Notice that we do not assume equidistant points, thus the trajectory $s(t)$ can be defined by a sequence of space-time coordinates (s_k, t_k) that are not regularly extracted in time. This does not cause any problem in the algorithm since each cell is assigned its own width Δt_i . The resulting computational mesh is referred to as staggered mesh, because the known quantities are defined at the interfaces $(k, k + 1) = (i - 1/2, i + 1/2)$ of each cell i whereas the sought unknown reconstruction polynomial $p_i^N(s, t)$ will be discretized at the cell center s_i . In other words, known data and unknown reconstruction polynomials adopt staggered definitions on the computational mesh.

3.3. Reconstruction algorithm

The aim of the reconstruction procedure is to obtain a piecewise high order polynomial approximation of the particle trajectory. The result is then a set of polynomials $p_i^N(s, t)$ of degree $N \geq 1$ that are defined within each computational cell $i \in [1, N_J]$ for each spatial direction s . The polynomial of arbitrary degree N is written in terms of a normalized Taylor series expanded around the barycenter coordinate t_i , that is

$$p_i^N(s, t) = \sum_{\ell=0}^N \frac{(t - t_i)^\ell}{\ell! \Delta t_i^\ell} \hat{s}_{i,\ell} := \phi_\ell(t) \hat{s}_{i,\ell}, \tag{5}$$

with Δt_i^ℓ being a normalizing factor necessary to avoid ill-conditioned reconstruction matrices and $\ell!$ represents as usual the factorial of ℓ . The unknown degrees of freedom are denoted by $\hat{s}_i = \{\hat{s}_{i,\ell}\}_{\ell=0}^N$ which have to be determined by the reconstruction algorithm. Let $\{\phi_\ell\}_{\ell=0}^N$ denote a basis for the polynomial space \mathbb{P}^N of degree N , the polynomial (5) can also be written as a linear combination of basis functions $\phi_\ell(t) \in \mathbb{P}^N$ belonging to the space of polynomials of degree N , i.e. \mathbb{P}^N . Einstein summation convention is adopted implying summation over repeated indexes, thus the reconstruction polynomial for cell i compactly writes $\sum_{\ell} \phi_\ell(t) \hat{s}_{i,\ell}$ as $\phi_\ell(t) \hat{s}_{i,\ell}$.

To retrieve a polynomial of degree N , a total number of $N + 1$ unknowns \hat{s}_i must be uniquely determined, as follows from the definition (5). Depending on the chosen degree $N \geq 1$, the piecewise high order polynomial is reconstructed for each cell i by considering the surrounding known points, that form the so-called reconstruction stencil $\mathcal{N}_k = \{\dots - 3/2, i - 1/2, i + 1/2, i + 3/2 \dots\}$ (Fig. 1D). The stencil size depends on the polynomial degree used for the reconstruction (Fig. 1D). Specifically, the stencil size comprises a total number of cells which is at least equal to the number of degrees of freedom of the chosen polynomial degree. Therefore, the higher the polynomial degree the wider the stencil, i.e. the more information from neighboring cells is required for the reconstruction. The stencil counts a total number of $2(N + 1)$ cell interfaces for even and odd degree N , so that the stencil is always symmetric with respect to the cell i under consideration. We address with N the polynomial degree, while $\mathcal{O}(N) = N + 1$ is the order of accuracy of the method. For instance, a polynomial $P1$ of degree $N = 1$ is second order accurate, i.e. $\mathcal{O}(1 + 1) = 2$,

because it is exact up to degree 1 and the approximation errors arise starting from $N = 2$, thus the accuracy is of second order. The safety factor of 2 in the total number of cells contained in the stencil leads to an overdetermined linear system for the degrees of freedom \hat{s}_i that allows for including linear constraints in the reconstruction polynomials. Indeed, we also want to ensure continuity of the reconstructed polynomials across cell interfaces located at $i \pm 1/2$. This would generate a continuous profile for the trajectory, while keeping the reconstruction algorithm local to each cell and therefore computationally efficient.

The reconstruction is designed in such a way that the trajectory coordinates $s_k = s_{i \pm 1/2}$ are exactly matched by the polynomial $p_i^N(s, t)$. For each cell i the reconstruction system is constructed as follows:

$$\phi_{\ell}(t_k) \hat{s}_{i,\ell} = s_k, \quad k \in \mathcal{N}_k, \tag{6}$$

with the linear constraints

$$\phi_{\ell}(t_{i-1/2}) \hat{s}_{i,\ell} = s_{i-1/2}, \quad \phi_{\ell}(t_{i+1/2}) \hat{s}_{i,\ell} = s_{i+1/2}. \tag{7}$$

The linear reconstruction system (6)-(7) set up above can be written in a compact matrix-vector form as

$$\begin{cases} \mathbf{MS} = \mathbf{B} \\ \mathbf{CS} = \mathbf{DB} \end{cases}, \tag{8}$$

with the definitions

$$\mathbf{B}_{[S \times 1]} = \begin{bmatrix} \vdots \\ s_{k-1} \\ s_k \\ s_{k+1} \\ \vdots \end{bmatrix}, \quad \mathbf{S}_{[(N+1) \times 1]} = \begin{bmatrix} \hat{s}_{i,1} \\ \vdots \\ \hat{s}_{i,N+1} \end{bmatrix},$$

$$\mathbf{M}_{[S \times (N+1)]} = \begin{bmatrix} \phi_1(t_1) & \dots & \phi_{N+1}(t_1) \\ \vdots & \vdots & \vdots \\ \phi_1(t_k) & \dots & \phi_{N+1}(t_k) \\ \vdots & \vdots & \vdots \\ \phi_1(t_S) & \dots & \phi_{N+1}(t_S) \end{bmatrix}. \tag{9}$$

The vector of unknown coefficients is \mathbf{S} , the right hand side which contains the known position values is \mathbf{B} and the reconstruction matrix is given by \mathbf{M} . The constrained matrix \mathbf{C} in (8) is a sub-partition of \mathbf{M} and contains those rows which correspond to the stencil elements $i \pm 1/2$, while the vector \mathbf{D} is zero everywhere apart from the two entries related to the constraints (7), where it is set to 1. The constrained least squares (CLSQ) method is adopted for solving system (8), therefore a functional $g(\mathbf{S})$ is defined and minimized by requiring its derivatives to vanish, thus

$$g(\mathbf{S}) = (\mathbf{MS} - \mathbf{B})^T (\mathbf{MS} - \mathbf{B}) - \boldsymbol{\mu}^T (\mathbf{CS} - \mathbf{DB}), \tag{10}$$

$$\frac{\partial g}{\partial \mathbf{S}} = 2\mathbf{MMS} - 2\mathbf{MB} - \mathbf{C}\boldsymbol{\mu} = 0, \tag{11}$$

$$\frac{\partial g}{\partial \boldsymbol{\mu}} = -(\mathbf{CS} - \mathbf{DB}) = 0, \tag{12}$$

where $\boldsymbol{\mu}$ is the vector of Lagrange multipliers to enforce the linear constraints. The associated enlarged linear system of normal equations then reads

$$\begin{pmatrix} 2\mathbf{MM} & -\mathbf{C} \\ \mathbf{C} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{S} \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} 2\mathbf{M} \\ \mathbf{D} \end{pmatrix} \mathbf{B}, \tag{13}$$

whose solution can be simply obtained as

$$\begin{pmatrix} \mathbf{S} \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} 2\mathbf{MM} & -\mathbf{C} \\ \mathbf{C} & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} 2\mathbf{M} \\ \mathbf{D} \end{pmatrix} \mathbf{B} := \mathbf{R}^L \mathbf{B}. \tag{14}$$

We are only interested in the solution of the expansion coefficients \mathbf{S} , therefore the final reconstruction matrix \mathbf{R} for the trajectory in the s dimension is a subset of the CLSQ matrix \mathbf{R}^L , that is $\mathbf{R} := \mathbf{R}_{[(N+1) \times \mathcal{N}_i]}^L$. Once the reconstruction matrices have been computed for all cells, the reconstruction algorithm is efficiently carried out by one matrix-vector product, namely

$$\mathbf{S} = \mathbf{R}_{[(N+1) \times \mathcal{N}_i]}^L \mathbf{B}, \tag{15}$$

which permits to obtain the sought degrees of freedom $\hat{s}_{i,\ell}$ and thus to uniquely define the reconstruction polynomial $p_i^N(s, t)$ according to (5) within each cell i .

3.4. Nonlinear limiting with CWENO strategy

In the presence of discontinuities in the trajectory, which might be due either to the real particle motion or to the casualty of the acquisition process, a limiting technique must be introduced in the reconstruction procedure. The limiter is needed to avoid the generation of spurious oscillations that arise when computing a high order interpolation of the trajectory. Since high order polynomials are not monotone, the reconstruction polynomial $p_i^N(s, t)$ could describe nonphysical oscillations that we want to eliminate. To that aim, we rely on the Central Weighted Essentially Non-Oscillatory (CWENO) strategy firstly proposed in [35] for hyperbolic systems of conservation laws.

Differently from [35], here we apply the CWENO strategy in a pointwise manner, hence resorting to a kind of finite difference WENO scheme [36]. In the CWENO framework the polynomial $p_i^N(s, t)$ given by (5) is called *optimal polynomial*, because among all the possible polynomials of degree N , it is close in the least-square sense to the point values s_k in the stencil \mathcal{N}_k , while satisfying the linear constraints (7). This reconstruction is linear in the sense of Godunov [28], thus it must be limited and stabilized via a nonlinear scheme. The nonlinearity of the reconstruction lies in the weights used to combine different reconstruction polynomials. According to [35,37], the central polynomial $p_{i,0}^N(s, t)$ is evaluated as the difference between the optimal polynomial $p_i^N(s, t)$ and the linear combination of the two one-sided polynomials $p_{i,1}^1(s, t)$ (left) and $p_{i,2}^1(s, t)$ (right) of degree $N = 1$ [38], that is

$$p_{i,0}^N(s, t) = \frac{1}{\lambda_0} \left(p_i^N(s, t) - \lambda_1 p_{i,1}^1(s, t) - \lambda_2 p_{i,2}^1(s, t) \right), \tag{16}$$

where $\lambda_0, \lambda_1, \lambda_2$ are positive coefficients such that

$$\lambda_0 + \lambda_1 + \lambda_2 = 1. \tag{17}$$

The one-sided second order polynomials are simply computed as the linear polynomial connecting the point value s_k with the direct left or right neighbor point, namely s_{k-1} or s_{k+1} , respectively:

$$p_{i,1}^1(s, t) = s_k + \frac{s_k - s_{k-1}}{t_k - t_{k-1}} \cdot t, \quad p_{i,2}^1(s, t) = s_k + \frac{s_{k+1} - s_k}{t_{k+1} - t_k} \cdot t. \tag{18}$$

Let us notice that the above relations can still be represented in terms of the expansion given by (5) by setting $N = 1$. These lower polynomials are nothing but the equivalent of the linear linking used in SPT methods. Here, the second order reconstructions (18) are instead used to limit the central high order reconstruction polynomial $p_{i,0}^N(s, t)$. The linear weights in (17) are a normalization which sums up to unity and we set $\lambda_0 = 200/\lambda_{sum}$ for the central polynomial, while we take $\lambda_1 = \lambda_2 = 1/\lambda_{sum}$ for the left and right one-sided polynomials, hence $\lambda_{sum} = 202$. The nonlinear data-dependent hybridization among the three polynomials obtained for each stencil is given by

$$\tilde{p}_i^N(s, t) = \omega_0 p_{i,0}^N(s, t) + \omega_1 p_{i,1}^{1,\ell}(s, t) + \omega_2 p_{i,2}^{1,r}(s, t), \tag{19}$$

where the nonlinear weights $\{\omega_m\}_{m=0}^2$ are given by

$$\omega_m = \frac{\tilde{\omega}_m}{\sum_{m=0}^2 \tilde{\omega}_m}, \quad \text{with} \quad \tilde{\omega}_m = \frac{\lambda_m}{(\sigma_m + \epsilon)^r}. \tag{20}$$

The parameter $\epsilon = 10^{-14}$ avoids division by zero and the exponent $r = 4$ is chosen according to [37]. The oscillation indicators σ_m are computed according to [39] as

$$\sigma_m = \sum_{\alpha=1}^N \int_{t_k}^{t_{k+1}} \frac{\partial^\alpha \phi_\ell(t)}{\partial t^\alpha} \hat{s}_{m,\ell} \cdot \frac{\partial^\alpha \phi_r(t)}{\partial t^\alpha} \hat{s}_{m,r} dt. \tag{21}$$

Eventually, the unlimited reconstruction polynomial (5) is overwritten as follows:

$$p_i^N(s, t) = \tilde{p}_i^N(s, t), \tag{22}$$

which results in a stable and limited high order approximation.

3.5. Computation of the length of curvilinear trajectory

The evaluation of the length L of the particle trajectory is needed in order to compute average quantities over the entire path. The length is defined as the integral along the line drawn by the particle trajectory. The integral can be rewritten as the sum of the integrals over all computational cells N_I , that is

$$L = \int_{x_1}^{x_{N_K}} dl = \sum_{I=1}^{N_I} \int_{x_{i-1/2}}^{x_{i+1/2}} dl. \tag{23}$$

The coordinates $(\mathbf{x}_1, \mathbf{x}_{N_K})$ denote the starting and the ending point of the trajectory in the space of dimension $D \in [1; 3]$. To perform the integration (23) at high order of accuracy, it is not sufficient to split the trajectory into N_I segments according to the number of cells and to sum up all the linear contributions. Indeed, this would lead to a second order accurate evaluation of L , since piecewise linear polynomials $P1$ are used and (23) reduces to the classical trapezoidal rule.

To obtain a consistent high order computation of the trajectory length, isoparametric discretizations must be adopted along the generic spatial direction s . To this aim, let us first define a reference coordinate system $\xi \in [0; 1]$ which is used to map a generic cell i , hence

$$s = s_{i-1/2} + \xi (s_{i+1/2} - s_{i-1/2}), \tag{24}$$

so that for $\xi = 0$ the left interface coordinate $s_{i-1/2}$ of the cell is retrieved, while for $\xi = 1$ the right interface $s_{i+1/2}$ is obtained. In the reference system we then define a set of nodal basis functions $\theta_m(\xi)$ which are chosen to be the Lagrange polynomials passing through the set of nodes $\hat{\xi}_m = m/N$ with $0 \leq m \leq N$. Furthermore, the standard interpolation condition $\theta_q(\hat{\xi}_m) = \delta_{qm}$ holds for the basis, with δ_{qm} being the usual Kronecker symbol. Up to degree $N = 3$ the nodal basis functions $\theta_m(\xi)$ with associated nodes $\hat{\xi}_m$ explicitly write as follows:

• $N = 1$

$$\begin{aligned} \theta_1 &= 1 - \xi & \theta_2 &= \xi \\ \hat{\xi}_1 &= 0 & \hat{\xi}_2 &= 1 \end{aligned} \tag{25}$$

• $N = 2$

$$\begin{aligned} \theta_1 &= 1 - 3\xi + 2\xi^2 & \theta_2 &= 4\xi - 4\xi^2 & \theta_3 &= -\xi + 2\xi^2 \\ \hat{\xi}_1 &= 0 & \hat{\xi}_2 &= 1/2 & \hat{\xi}_3 &= 1 \end{aligned} \tag{26}$$

• $N = 3$

$$\begin{aligned} \theta_1 &= 1 - \frac{11}{2}\xi + 9\xi^2 - \frac{9}{2}\xi^3 & \theta_2 &= 9\xi - \frac{45}{2}\xi^2 + \frac{27}{2}\xi^3 \\ \theta_3 &= -\frac{9}{2}\xi + 18\xi^2 - \frac{27}{2}\xi^3 & \theta_4 &= \xi - \frac{9}{2}\xi^2 + \frac{9}{2}\xi^3 \\ \hat{\xi}_1 &= 0 & \hat{\xi}_2 &= 1/3 \\ \hat{\xi}_3 &= 2/3 & \hat{\xi}_4 &= 1 \end{aligned} \tag{27}$$

The integral over cell i in (23) can be performed in the reference system at the aid of a change of variables $\mathbf{x} \rightarrow \xi$, hence obtaining

$$\int_{\mathbf{x}_{i-1/2}}^{\mathbf{x}_{i+1/2}} dl = \int_0^1 |J_\xi| d\xi, \tag{28}$$

with the Jacobian determinant given by

$$|J_\xi| = \sqrt{\left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2 + \left(\frac{\partial z}{\partial \xi}\right)^2}. \tag{29}$$

The computation of the derivatives in (29) is done as usual by exploiting the expansion of the basis functions $\theta(\xi)$, thus for a generic spatial coordinate s it holds

$$\frac{\partial s}{\partial \xi} = \frac{\partial \theta_m(\xi)}{\partial \xi} \hat{s}_m, \tag{30}$$

where the degrees of freedom \hat{s}_m represent the value of the coordinate s corresponding to the reference node $\hat{\xi}_m$. To be precise, the values \hat{s}_m are computed according to (31), relying on the SHOT-R reconstruction polynomial. Finally, the integration in (28) is numerically approximated by Gaussian quadrature formulae of suitable order of accuracy [40].

3.6. Computation of position, velocity and acceleration

The reconstruction polynomials $p_i^N(s, t)$ provide a piecewise continuous representation of the trajectory within each cell i along the generic spatial direction s as a function of time t . Thus, the kinematic description (position s , velocity v_s and acceleration a_s) of the particle at any given time \tilde{t} can be easily computed in two steps:

- find the computational cell containing \tilde{t} , i.e. $i \in [1, N_I]$ such that $t_{i-1/2} \leq \tilde{t} \leq t_{i+1/2}$, so that the associated reconstruction polynomial $p_i^N(s, t)$ can be identified;

- extract the kinematic quantities

$$s(\tilde{t}) = p_i^N(s, \tilde{t}) = \sum_{q=0}^N \frac{(\tilde{t} - t_i)^q}{q! \Delta t_i^q} \hat{s}_i, \tag{31}$$

$$v_s(\tilde{t}) = \frac{d p_i^N(s, \tilde{t})}{dt} = \sum_{q=1}^N \frac{q(\tilde{t} - t_i)^{q-1}}{q! \Delta t_i^q} \hat{s}_i, \tag{32}$$

$$a_s(\tilde{t}) = \frac{d^2 p_i^N(s, \tilde{t})}{dt^2} = \sum_{q=2}^N \frac{q(q-1)(\tilde{t} - t_i)^{q-2}}{q! \Delta t_i^q} \hat{s}_i. \tag{33}$$

The above definitions are referred to as instantaneous quantities. We also need to give explicit formulation for the global average speed v_L , the displacement velocity v_D and the mean of instantaneous velocities within each frame \bar{v}_M . These definitions are mathematically described as follows:

$$v_L = \frac{L}{t_{N_K} - t_1}, \quad L = \int_{\mathbf{x}_1}^{\mathbf{x}_{N_K}} dl, \tag{34}$$

$$v_D = \frac{s_{N_K} - s_1}{t_{N_K} - t_1}, \tag{35}$$

$$v_M = \frac{1}{N_I} \sum_{i=1}^{N_I} v_i, \quad v_i = \frac{s_{i+1} - s_i}{t_{i+1} - t_i}, \tag{36}$$

with s denoting as usual a generic spatial direction among $\mathbf{x} = (x, y, z)$. The length of the trajectory L in (34) is computed as the integral along the curvilinear path from \mathbf{x}_1 to \mathbf{x}_{N_K} according to (23). Finally, the instantaneous velocity at a generic space-time coordinate (s, \tilde{t}) can be simply evaluated relying on the velocity reconstruction polynomial, thus according to (32).

In order to fully exploit the piecewise continuous information provided by the high order reconstruction, the evaluation of all instantaneous kinematic quantities (position, velocity and acceleration) is extracted on a very fine computational mesh given by $N + 1$ Gauss points within each computational cell, hence leading to exact integration up to order $2(N + 1) - 1$ (see [40]) within each cell.

4. Numerical results

The test suite is chosen such that we rigorously validate the accuracy of our method from the mathematical viewpoint, but we also propose some test cases to demonstrate the applicability of the algorithm to biological datasets.

All data are analyzed with Prism (GraphPad 6 or 7) and for all statistical tests the significance level is $\alpha = 0.05$. The exact number of replicates, the type of statistical tests and p-values are reported in Figure legends. We use the abbreviation SPT for referring to SPT followed by linear linking, while our new reconstruction algorithm is addressed with SHOT-R.

4.1. Numerical convergence studies

The order of accuracy $p > 0$ of a numerical method is also defined as the largest value of p for which the following inequality holds [41]:

$$\varepsilon(\Delta t) \leq C \Delta t^p, \tag{37}$$

where $\varepsilon(\Delta t)$ is the error of the numerical solution, which depends on the mesh size Δt . The constant $C > 0$ depends on the problem but is independent of Δt , this is also written thus

$$\varepsilon = \mathcal{O}(\Delta t^p), \quad \text{as } \Delta t \rightarrow 0. \tag{38}$$

In other words, the numerical error must decrease with convergence rate of $\mathcal{O}(\Delta t^p)$ as the mesh size decreases. Given two regular meshes with mesh sizes Δt_1 and Δt_2 , respectively, and the same numerical method, we have a fixed relationship between the meshes which writes

$$\varepsilon(\Delta t_1) = C \Delta t_1^p, \tag{39}$$

$$\varepsilon(\Delta t_2) = C \Delta t_2^p. \tag{40}$$

Dividing (40) by (39) we obtain

$$\left(\frac{\Delta t_2}{\Delta t_1} \right)^p = \frac{\varepsilon(\Delta t_2)}{\varepsilon(\Delta t_1)}, \tag{41}$$

which leads to

$$p = \frac{\ln\left(\frac{\varepsilon(\Delta t_2)}{\varepsilon(\Delta t_1)}\right)}{\ln\left(\frac{\Delta t_2}{\Delta t_1}\right)}, \tag{42}$$

that is the empirical order of accuracy. It must be $p = N + 1$, meaning that a numerical scheme with reconstructions of degree N introduces numerical errors starting from degree $N + 1$, that is a polynomial of degree N can be exactly reconstructed. In practice one performs numerical experiments for a sequence of refined meshes $\{\Delta t_1, \Delta t_2, \dots, \Delta t_K\}$ with usually a fixed ratio between successive meshes of sizes Δt_k and Δt_{k+1} . For example $\Delta t_{k+1} = \frac{\Delta t_k}{2}$. A sequence of numbers $\{p_2, p_3, \dots, p_K\}$ is then retrieved which is given by

$$p_k = \frac{\ln\left(\frac{\varepsilon(\Delta t_k)}{\varepsilon(\Delta t_{k-1})}\right)}{\ln\left(\frac{\Delta t_k}{\Delta t_{k-1}}\right)}, \quad k = 2, \dots, K. \tag{43}$$

To carry out the convergence studies, we consider the time interval $T = [-1; 1]$ and we prescribe the following analytical trajectory in 3D:

$$\mathbf{x} = (x, y, z) = (\sin(\pi x) \cos(2\pi x), 3 \cos(2\pi y) - 2 \sin(\pi y), -6 \sin(\pi z) + 2 \cos(3\pi z)). \tag{44}$$

The new SHOT-R method is then employed to reconstruct the trajectory along each direction and the errors ε are measured for each spatial direction (x, y, z) in L_1 , L_2 and L_∞ norms according to (B.1). The exact solution is explicitly computed with (44), while the reconstructed position is evaluated according to (31). A sequence of four successively refined meshes is used with $\Delta t = \{100, 200, 400, 800\}$ and the results are reported in Table 1. The errors are measured in the error norms described in Appendix B (Fig. 2A), and the results numerically confirm that the SHOT-R algorithm is actually high order accurate ($N + 1$) for an arbitrary polynomial degree N .

4.2. Comparison test against the SPT linear linking approach

This test is set up aiming at comparing the SHOT-R method against the SPT linear linking technique, demonstrating an improvement over state-of-the-art linear tracking approaches. In particular, in a time computational domain $T = [t_0; t_f] = [0; 2]$, we prescribe the particle velocity given by

$$v_x(t) = 1 + \tanh(5(t - 1)), \quad v_y(t) = 2\pi \cos(2\pi t), \tag{45}$$

from which it is possible to evaluate an analytical expression for the particle trajectory by integrating (45) over time t , thus

$$x(t) = \int_t v_x(t) dt = 2t - \frac{\log(\tanh(5(t - 1)) + 1)}{5}, \tag{46}$$

$$y(t) = \int_t v_y(t) dt = \sin(2\pi t). \tag{47}$$

As such, we describe through known functions (45)-(47) the shape of an hypothetical complex xy -trajectory: along the x -axis the particle stalls, then accelerates and eventually moves at a constant velocity; while along y -axis it oscillates regularly (Fig. 2B-E). Three different mesh refinements are considered (Δt_1 , Δt_2 and Δt_3), for a total number of $N_1 = 21$, $N_2 = 41$ and $N_3 = 81$ equidistant points of coordinates $\mathbf{x}_k = (x_k, y_k)$, simulating increasing frequencies of acquisition (increased input points \mathbf{x}_k) and thus increasing temporal resolution (Fig. 3C). The particle trajectory is reconstructed and the velocity computed with both the linear linking (SPT) and SHOT-R.

While with an intermediate mesh density (Δt_2) the position is overall nicely reconstructed by both SPT and SHOT-R, the velocity vector is more accurate using a higher order reconstruction (Fig. 2B-E, 3A-C). Noteworthy, in the reconstruction of the v_x component, the SPT solution is not capable of reproducing the curve accurately, specifically where the velocity drastically increases (Fig. 2E). As measured in L_1 , L_2 and L_∞ norms according to (B.1) (Fig. 2A), the SHOT-R reconstruction systematically achieves better results with errors of about two orders of magnitude smaller with respect to the linear reconstruction SPT (Fig. 3A-C). Moreover, the SHOT-R method results to be consistent and convergent, with decreasing errors by increasing the computational mesh resolution (i.e. the more known points \mathbf{x}_k are given as input, the lower are the errors generated by the reconstruction procedure).

In conclusion, the convergence analysis and the mathematical test validate the accuracy and the robustness of SHOT-R, and show its increased accuracy compared to SPT in computing velocity.

4.3. Velocity reconstruction and backward trajectory integration on biological datasets

As expected, SHOT-R exhibits higher resolution in the reconstruction of the velocity on a mathematical test case (Fig. 2E, 3A-C), thus finally we investigate whether our numerical model applied to a biological dataset will produce similar results. In particular, the novel method is tested on a recent study of pre-miR-181a-1 trafficking along axons [5]. Previously, pre-miR-181a-1 kinematics

Table 1

Convergence table Numerical convergence rates obtained with high order accurate reconstruction of the velocity field given in Section B of the article. Errors in L_1 , L_2 and L_∞ norms are reported for all position components (x, y, z) with associated order of accuracy \mathcal{O} on a sequence of refined mesh with characteristic size Δt .

N = 1																		
Δt	horizontal position x						horizontal position y						horizontal position z					
	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$
1.00e-2	1.89e-3	-	1.49e-3	-	1.64e-3	-	5.06e-3	-	4.00e-3	-	4.60e-3	-	7.74e-3	-	6.24e-3	-	7.62e-3	-
5.00e-3	4.73e-4	2.00	3.72e-4	2.00	4.11e-4	2.00	1.27e-3	2.00	1.00e-3	2.00	1.15e-3	2.00	1.94e-3	2.00	1.56e-3	2.00	1.91e-3	2.00
2.50e-3	1.18e-4	2.00	9.31e-5	2.00	1.03e-4	2.00	3.16e-4	2.00	2.50e-4	2.00	2.88e-4	2.00	4.84e-4	2.00	3.90e-4	2.00	4.77e-4	2.00
1.25e-3	2.95e-5	2.00	2.33e-5	2.00	2.57e-5	2.00	7.91e-5	2.00	6.25e-5	2.00	7.20e-5	2.00	1.21e-4	2.00	9.75e-5	2.00	1.19e-4	2.00
N = 2																		
Δt	horizontal position x						horizontal position y						horizontal position z					
	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$
1.00e-2	2.80e-4	-	2.49e-4	-	6.16e-4	-	4.86e-4	-	4.14e-4	-	5.91e-4	-	1.11e-3	-	9.49e-4	-	1.38e-3	-
5.00e-3	3.43e-5	3.03	3.02e-5	3.04	8.16e-5	2.92	5.96e-5	3.03	5.13e-5	3.01	7.39e-5	3.00	1.35e-4	3.04	1.16e-4	3.03	1.72e-4	3.00
2.50e-3	4.23e-6	3.02	3.69e-6	3.03	1.03e-5	2.98	7.42e-6	3.01	6.41e-6	3.00	9.24e-6	3.00	1.68e-5	3.01	1.45e-5	3.01	2.15e-5	3.00
1.25e-3	5.25e-7	3.01	4.55e-7	3.02	1.30e-6	2.99	9.27e-7	3.00	8.01e-7	3.00	1.15e-6	3.00	2.09e-6	3.00	1.81e-6	3.00	2.69e-6	3.00
N = 3																		
Δt	horizontal position x						horizontal position y						horizontal position z					
	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$
1.00e-2	7.66e-5	-	6.60e-5	-	1.04e-4	-	9.05e-5	-	8.24e-5	-	2.27e-4	-	2.99e-4	-	2.69e-4	-	6.88e-4	-
5.00e-3	4.68e-6	4.03	4.01e-6	4.04	4.94e-6	4.39	5.61e-6	4.01	5.00e-6	4.04	1.50e-5	3.92	1.88e-5	3.99	1.68e-5	4.00	4.94e-5	3.80
2.50e-3	2.90e-7	4.01	2.50e-7	4.00	3.10e-7	3.99	3.47e-7	4.01	3.05e-7	4.04	9.51e-7	3.98	1.17e-6	4.01	1.03e-6	4.03	3.19e-6	3.95
1.25e-3	1.81e-8	4.00	1.56e-8	4.00	1.94e-8	4.00	2.16e-8	4.01	1.88e-8	4.02	5.96e-8	4.00	7.29e-8	4.01	6.34e-8	4.02	2.01e-7	3.99
N = 4																		
Δt	horizontal position x						horizontal position y						horizontal position z					
	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$
1.00e-2	1.18e-5	-	1.17e-5	-	4.77e-5	-	9.72e-6	-	8.65e-6	-	2.74e-5	-	5.11e-5	-	4.89e-5	-	1.88e-4	-
5.00e-3	3.68e-7	5.01	3.67e-7	4.99	1.98e-6	4.59	2.76e-7	5.14	2.36e-7	5.20	4.66e-7	5.88	1.42e-6	5.17	1.23e-6	5.32	3.31e-6	5.83
2.50e-3	1.11e-8	5.05	1.06e-8	5.12	6.60e-8	4.91	8.40e-9	5.04	7.21e-9	5.03	9.98e-9	5.55	4.27e-8	5.06	3.66e-8	5.07	5.48e-8	5.92
1.25e-3	3.38e-10	5.04	3.09e-10	5.10	2.09e-9	4.98	2.61e-10	5.01	2.25e-10	5.00	3.12e-10	5.00	1.32e-9	5.01	1.14e-9	5.01	1.57e-9	5.12
N = 5																		
Δt	horizontal position x						horizontal position y						horizontal position z					
	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$
1.00e-2	3.90e-6	-	3.91e-6	-	1.71e-5	-	1.92e-6	-	1.98e-6	-	9.03e-6	-	1.35e-5	-	1.29e-5	-	4.73e-5	-
5.00e-3	5.63e-8	6.12	4.93e-8	6.31	1.57e-7	6.77	2.99e-8	6.01	2.99e-8	6.05	1.70e-7	5.73	2.24e-7	5.91	2.20e-7	5.87	1.19e-6	5.31
2.50e-3	8.54e-10	6.04	7.33e-10	6.07	1.27e-9	6.94	4.58e-10	6.03	4.32e-10	6.11	2.77e-9	5.94	3.47e-9	6.02	3.26e-9	6.07	2.07e-8	5.85
1.25e-3	1.32e-11	6.01	1.14e-11	6.01	1.49e-11	6.42	7.05e-12	6.02	6.39e-12	6.08	4.39e-11	5.98	5.36e-11	6.02	4.85e-11	6.07	3.31e-10	5.97

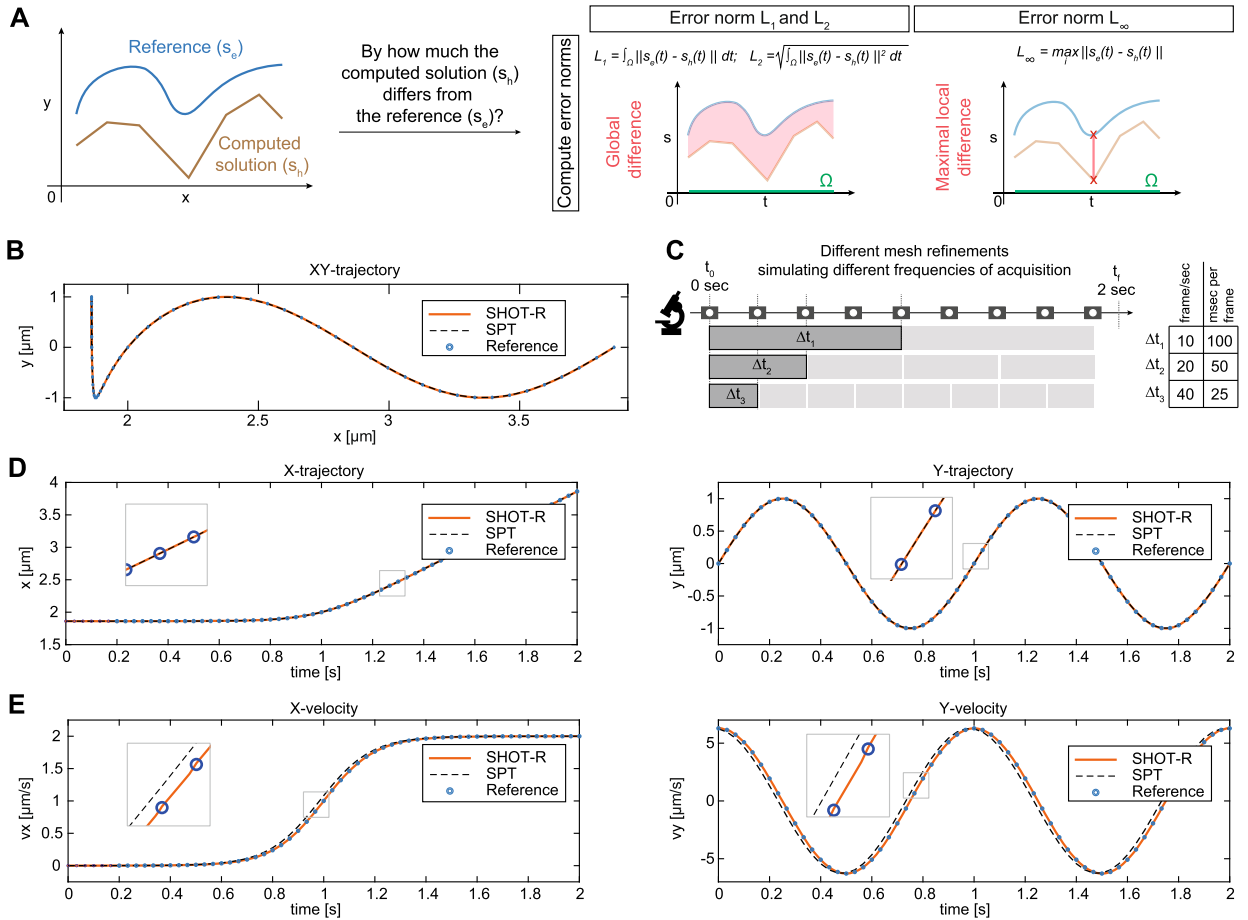


Fig. 2. Comparison test against SPT linear linking. (A) Schematic representation of error norms calculation Eq. (B.1). (B,C) XY-reference trajectory mimicking particle movement (B) and different mesh refinements simulating timelapse experiments at different acquisition frequency (C). Parameters used in the new mathematical test: Eq. (45) and Eq. (47) x- and y-trajectory and velocity in a time computational domain $T = [t_0; t_f] = [0; 2]$. (D,E) Qualitative comparison between SPT and SHOT-R in trajectory reconstruction (D) and in computing velocity (E) on given equations (45) and (47). (D,E) Δt_2 mesh refinement. Abbreviations: P_1, P_2, P_3 , first, second, third degree polynomial; Δt , time intervals (mesh refinement); SHOT-R, Spatiotemporal High-Order Trajectory Reconstruction; SPT, single particle tracking.

was studied by kymograph analysis [5], while here we retrieve particle coordinates with TrackMate [34] and apply SHOT-R method. SHOT-R trajectory reconstruction is compared with the already existing SPT followed by linear linking methods (Fig. 4A,B). As observed for the comparison test (Fig. 2E, 3A-C), biological particle trajectories do not change depending on the polynomial reconstruction (Fig. 4A, Fig. 5A,B). Indeed, the mean values of norm distribution were close to zero (i.e. L_1 , 0.08 μm ; L_2 and L_∞ , 0.03 μm) (Fig. 4A), suggesting a perfect overlap of the SPT and SHOT-R reconstructed trajectories (Fig. 5A,B). On the contrary, the computed velocities differ with mean errors greater than zero (L_1 , 0.65 $\mu\text{m/s}$; L_2 , 0.24 $\mu\text{m/s}$; L_∞ , 0.26 $\mu\text{m/s}$) (Fig. 4B, Fig. 5C,D). Axonal particles moving faster than 0.5 $\mu\text{m/s}$ are considered fast-moving puncta [5,42,43], hence such an average difference between the two approaches might interfere with the final data interpretation.

Investigating which method better approximates the real particle trajectory, and which one is more reliable in studying the particle kinematics remains a major challenge. Indeed, any experimental dataset is always a list of space-time coordinates without a continuous information of particle position during time. As such, it does not exist a reference solution, a known real kinematic behavior of a particle, and thus so far, there has not been a strategy to determine which analysis approach is more accurate over another one on a real experimental dataset. Here, to overcome the problem of the lack of an exact reference solution, we implement a test to check whether our model is significantly more accurate in computing velocities than available strategies (Fig. 6A). We underline that this approach can be applied to any method for particle kinematic analysis and any kind of experimental dataset.

First, xy trajectories of all particles are reconstructed with first and third polynomial degree (i.e. SPT and SHOT-R trajectories) based on the x_k coordinates of all tracked particles (Fig. 6A, ①). Second, velocity in xy is derived from SPT and SHOT-R trajectories (Fig. 6A, ②). Third, the particle trajectories are then retrieved from the computed velocity by backward time integration of the ODE (1) through Runge-Kutta (RK) methods [41] obtaining the “RK-trajectories” (Fig. 6A ③, Fig. 6B). Lastly, the reconstructed trajectories based on velocities for SPT and SHOT-R (i.e. RK-SPT and RK-SHOT-R trajectories) are compared to the real trajectory given by the

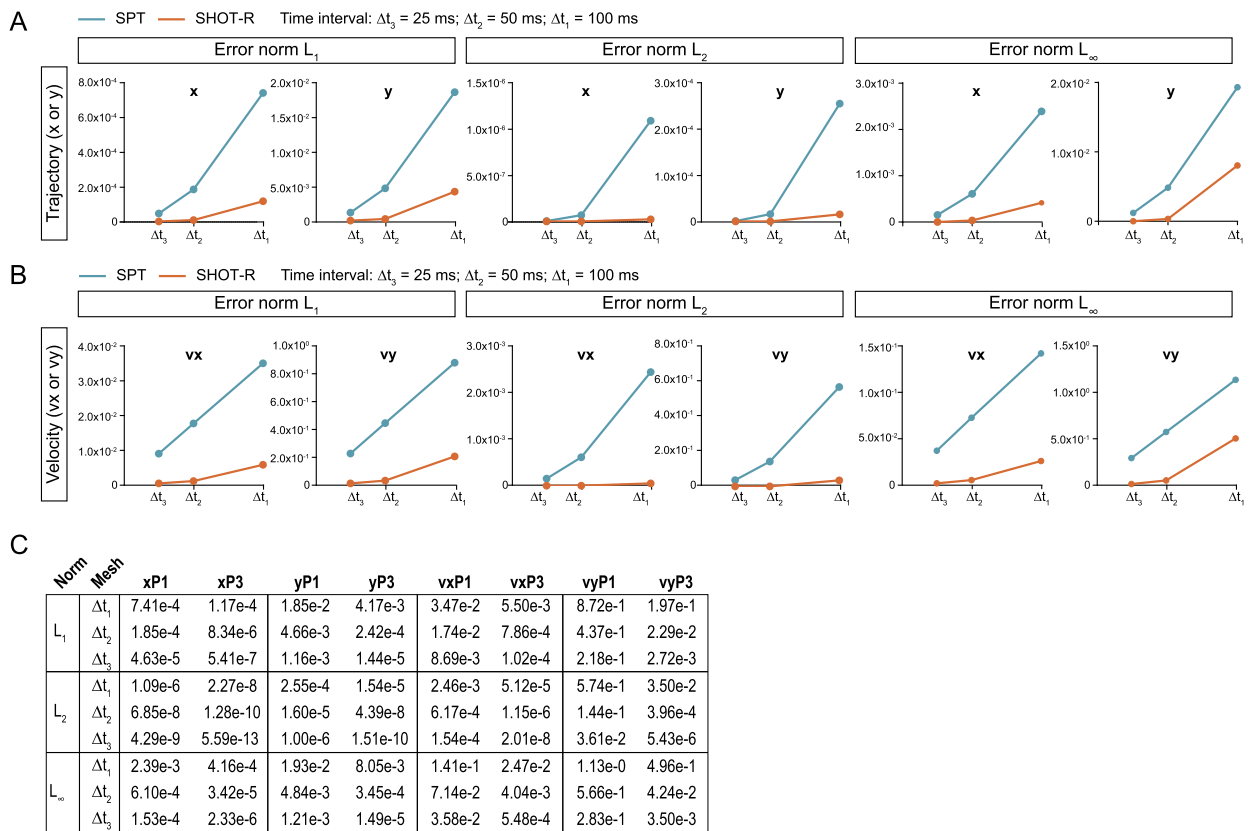


Fig. 3. Quantitative comparison test against SPT linear linking. (A-C) Quantitative comparison between SPT and SHOT-R in trajectory reconstruction (A,C) and in computing velocity (B,C) on given equations Eq. (45) and Eq. (47). Abbreviations: Δt , time intervals (mesh refinement); SPT, single particle tracking.

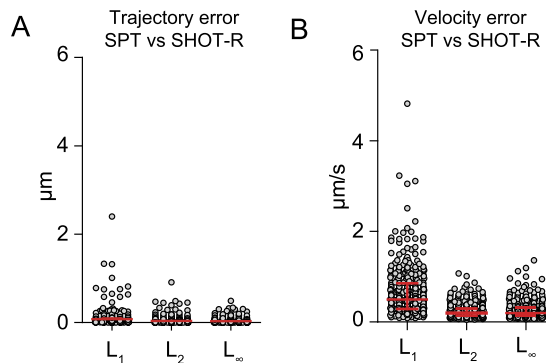


Fig. 4. Comparison of SPT and SHOT-R trajectories and computed velocities on a biological dataset. (A,B) Difference between SPT and SHOT-R reconstructed trajectory (A) or computed velocity (B) measured as error norms (L_1 , L_2 , L_∞) with (B.1). Abbreviations: SHOT-R, Space-time High-Order Trajectory Reconstruction; SPT, single particle tracking. Data information: each data point corresponds to a particle. Total number of analyzed particles: 372.

known data points x_k (Fig. 6A ④, Fig. 6B). The difference between the reference trajectory and the RK-SHOT-R trajectory, measured as error norm, is significantly reduced compared to the difference between the reference trajectory and the RK-SPT (Fig. 6C-E), both considering globally the curve (Fig. 6C,D), as well as the maximal difference in position obtained for each trajectory (Fig. 6E). This means that the velocities computed by SHOT-R are more accurate than those obtained by SPT, indeed the trajectories retrieved by backward time integration of SHOT-R velocities are more accurate than those retrieved with SPT (Fig. 6C,E). The Runge-Kutta schemes used for carrying out the time stepping are reported in Appendix C.

In summary, we have shown that, likewise for the mathematical test, SHOT-R method increases the accuracy in velocity reconstruction on this real biological dataset.

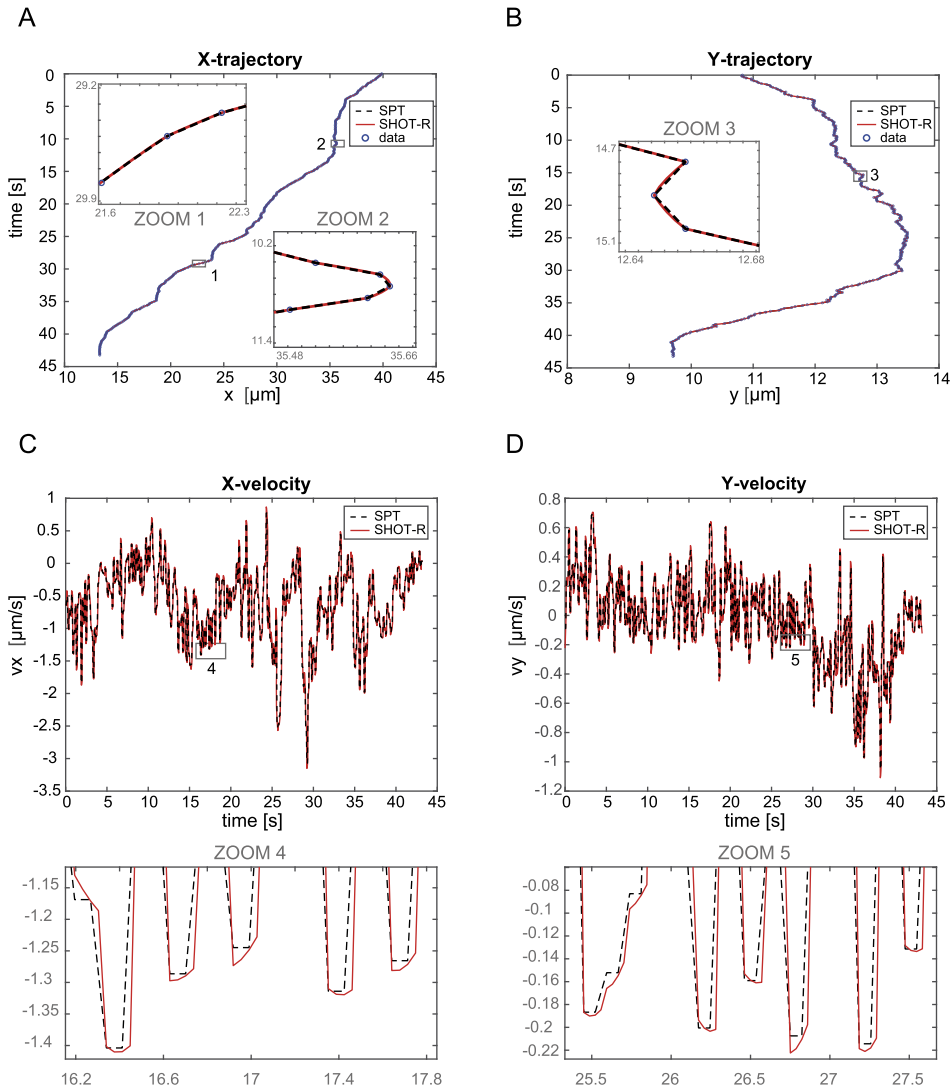


Fig. 5. Velocity reconstruction on a biological dataset. (A-D) Representative comparison of SPT and SHOT-R in xy trajectory reconstruction (A,B) and in computing velocity (C,D).

4.4. Investigation of SHOT-R dependency on SNR and different detection algorithm

Any dataset can be affected by noise due to the experimental conditions and to the setup used for data acquisition. In turn, noisy data affect the performance of detection and tracking algorithms [11], leading to shorter or fragmented tracks due to missed detection, and they are prone to errors in case of false positive detected particles. All-in-all noisy data lead to different space-time coordinates of the particles under consideration. Detection and tracking steps are independent of the chosen trajectory reconstruction technique, because they are ultimately responsible to only provide the input space-time coordinates upon which the trajectory is reconstructed. Therefore, the lower the quality of input data, either because of a high background noise or because of a less accurate detection and tracking procedure, the lower the quality of the reconstructed trajectory. The opposite holds true as well, making it clear that the trajectory reconstruction is an independent step.

Nevertheless, in this section we compare different algorithms of particle detection and of trajectory reconstruction on data with an increased background noise. The aim is to investigate how a variation in the input data impacts the different reconstruction methods. The background noise is measured by the signal-to-noise ratio (SNR) which is computed as

$$\text{SNR} = 10 \log \left(\frac{\sigma_S}{\sigma_N} \right), \tag{48}$$

where σ_S and σ_N denote the variance of the signal and the noise, respectively. The variance is evaluated as usual as the square of the standard deviation. The software TrackMate v5.2.0 [34] is then used to perform detection and tracking. Finally, the obtained space-time coordinates undergo a reconstruction algorithm with linear linking (SPT), SHOT-R and cubic spline. Cubic splines are

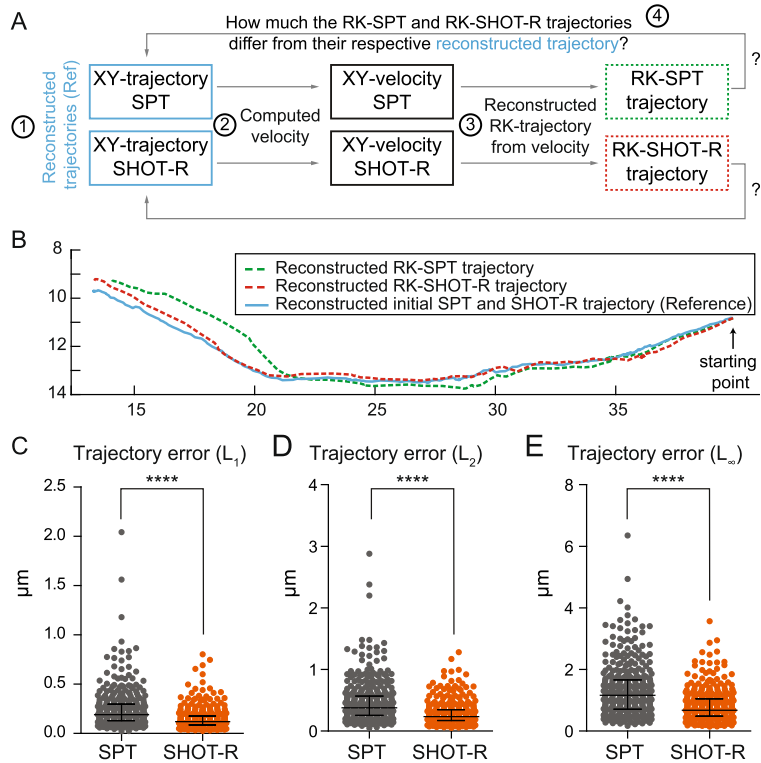


Fig. 6. Backward trajectory integration on a biological dataset. (A) Schematic of the pipeline to test SPT and SHOT-R accuracy in calculating velocity. Color code matches panel B. (B) Representative comparison of RK-SPT and RK-SHOT-R trajectory obtained by backward integration of the reconstructed velocity of a retrograde moving particle. (C-E) Difference between RK-SPT and RK-SHOT-R measured as error norms L_1 (C), L_2 (D), L_∞ (E) of the reconstructed trajectory based on velocity trace. Abbreviations: SHOT-R, Spatiotemporal High-Order Trajectory Reconstruction; SPT, single particle tracking; RK, Runge-Kutta method for backward time integration. Data information: **** $P < 0.0001$. Values are median with interquartile range. Data are not normally distributed (Shapiro-Wilk test), two-tailed Wilcoxon matched pairs test. Each data point corresponds to a particle. Total number of analyzed particles: 372 (C-E).

widely employed in the reconstruction of particle trajectories [44,45], and they are based on a global reconstruction enforcing global continuity of the first derivative, thus making the obtained polynomial C^1 -differentiable.

To generate images having different signal to noise ratio (SNR), a random noise is applied to a representative timelapse of three moving particles (Fig. 7A). The standard deviation is assessed using the ImageJ “Measure” tool on a manually traced rounded region of interest (ROI) corresponding to each particle, and to the background. The SNR for the whole image is then obtained by averaging the SNR values of all the single particles contained in the field of view. To gradually decrease the SNR of the image from 16 to 8, 4 or 2, a Gaussian-distributed random noise is added to the entire image using the “Add specific noise” option in ImageJ. This is achieved by increasing the standard deviation of the Gaussian-distributed noise. The intensity profiles of the three particles further illustrate how the random noise decreases the SNR of all of them (Fig. 7B). The intensity of the particles (i.e. gray levels in the plot profiles, Fig. 7B) is not homogeneous in the field of view and depending on the quality of the starting signal the decreased SNR differently affects particles detection and tracking (Fig. 7B). As expected, particles with low SNR values are difficult to be detected, giving rise to shorter or fragmented tracks (Table 2), as analyzed in [10].

Next, the obtained timelapse at different SNR is used as input for TrackMate. Particle coordinates are then obtained in TrackMate employing three different detection algorithms named DoG (differences of Gaussian), Hessian, and LoG (Laplacian of Gaussian). A different quality threshold is applied depending on the SNR of the tested images. The estimated diameter is set to $0.6 \mu\text{m}$, median filter and sub-pixel localization options are selected in the detection step. The tracking settings are the same for all particles (detailed in Appendix A). In all the images, independently of the used detection technique, a Simple Linear Assignment Problem (LAP) algorithm performs the tracking. The obtained particle coordinates are used as input for the reconstruction algorithms: linear linking (SPT), SHOT-R and cubic spline.

We perform the test case described in Section 4.3, thus the reconstruction polynomial is employed to evaluate the velocity field in order to perform a background time integration of the trajectory for each particle under consideration with a time step of $d\tau = 0.7$. For linear linking (SPT) we use the second order Runge-Kutta scheme, while for SHOT-R and cubic spline we rely on the fourth order time integrator, see Appendix C. Errors are measure in L_1 , L_2 and L_∞ norms, see Appendix B. Being the reconstruction algorithms not designed for particle detection and tracking, we evaluate the impact of SNR variation in the reconstruction of the trajectories. Clearly, the missed detections due to low SNR (Table 2) are not recovered by applying any trajectory reconstruction algorithm. However, independently of the detection algorithm used to retrieve the particle spatio-temporal coordinates, the SHOT-R algorithm shows a more accurate trajectory reconstruction over both the linear linking (SPT) and the cubic spline (Fig. 7 C, Table 3).

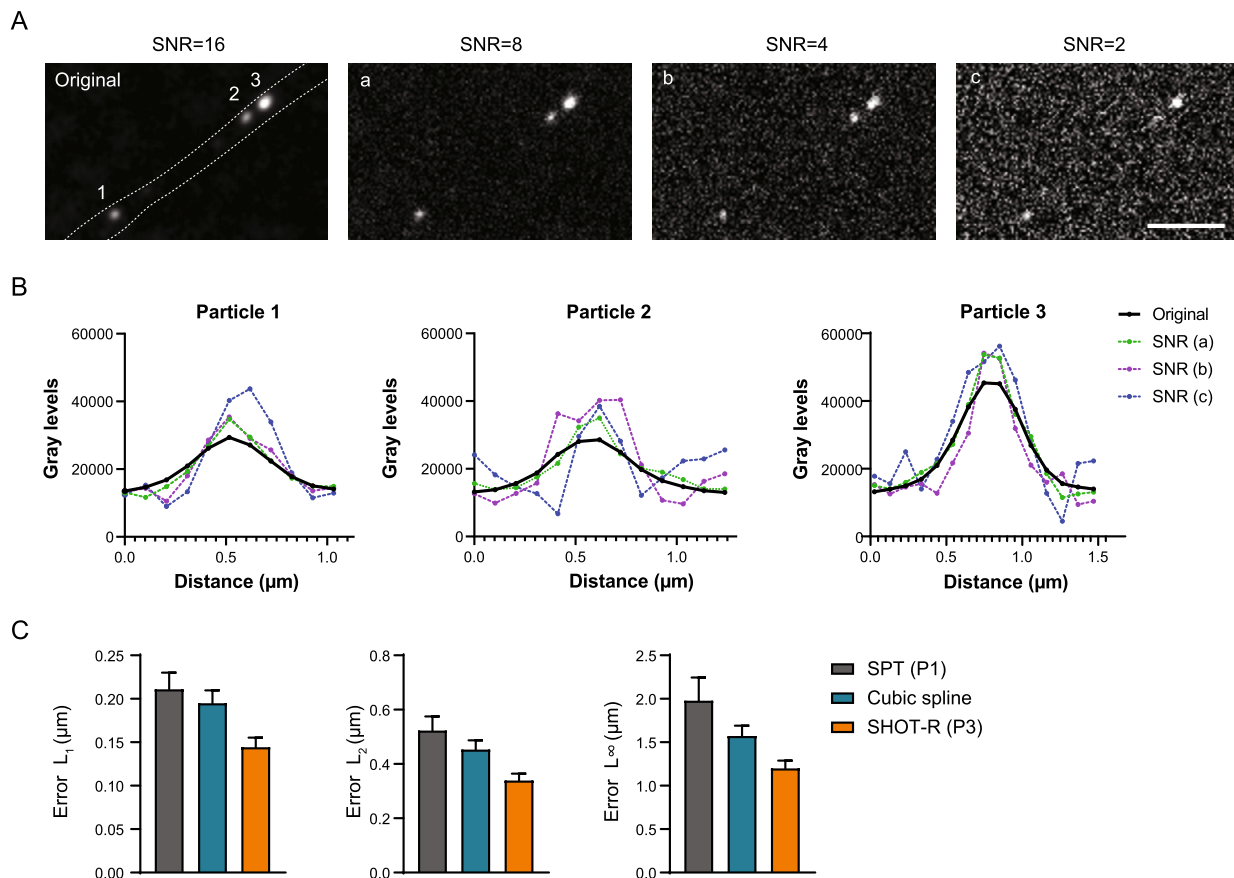


Fig. 7. Investigation of SHOT-R dependency on SNR and different detection algorithms. (A) Selected representative timelapse of three moving particles (1,2,3) at progressively altered SNR (a, b, c). SNR values are the average of the SNR of the three particles. SNR of each single particle is reported in Table 2. Dashed lines represent the boundary of the axon containing the moving particles. Scalebar: 5 μm . (B) Intensity plot profiles of particle 1, 2 and 3 reported in panel A, upon decreasing of the SNR (a, b, c). (C) Difference among SPT, cubic spline and SHOT-R measured as error norms L_1 , L_2 , L_∞ of the reconstructed trajectory based on velocity trace. Abbreviations: SNR, signal-to-noise ratio; SHOT-R, Spatiotemporal High-Order Trajectory Reconstruction; SPT, single particle tracking. Data information: **** P < 0.0001. Values are mean with SEM of all the errors measured for all particles starting from different algorithms. Single values are reported in Table 3.

Table 2

Frames detected per particle depending on SNR. SNR, signal-to-noise ratio; Average SNR, average of the SNR of the single three particles rounded to the nearest integer, Algorithm 1, DoG detector + LAP tracker; Algorithm 2, Hessian detector + LAP tracker; Algorithm 3, LoG detector + LAP tracker. The SNR = 16 corresponds to the original dataset prior to noise application, and the Algorithm 1 is what was used for the tracking in the method validation presented in the previous sections. Therefore, row SNR = 16 and column Algorithm 1 is the reference.

Average SNR	Particle#	SNR Particle	Algorithm 1	Algorithm 2	Algorithm 3
16	Particle 1	14.01	67	28+28	66
	Particle 2	14.73	300	107 + 34	300
	Particle 3	20.51	300	300	300
8	Particle 1	4.75	67	58	67
	Particle 2	7.20	297	102+35	298
	Particle 3	11.49	300	300	300
4	Particle 1	1.65	44	31	64
	Particle 2	4.62	110	52	133
	Particle 3	4.93	298	278 + 23	300
2	Particle 1	-0.68	7	X (not detected at all)	8
	Particle 2	3.05	48	11	47
	Particle 3	4.62	242+7	143+152	255+15

Table 3
Difference among SPT, cubic spline and SHOT-R measured as error norms L_1 , L_2 , L_∞ of the reconstructed trajectory based on velocity trace. Avg SNR, average signal-to-noise of the three particles in the field of view; P#, number of the particle; Alg, algorithms used for the detection and tracker: 1, DoG detector + LAP tracker; 2, Hessian detector + LAP tracker; Algorithm 3, LoG detector + LAP tracker. Abbreviations: SNR, signal-to-noise ratio; SHOT-R, Spatiotemporal High-Order Trajectory Reconstruction; SPT, single particle tracking.

Avg SNR	Par #	Alg	L1 SPT	L2 SPT	L_∞ SPT	L1 Spline	L2 Spline	L_∞ Spline	L1 SHOT-R	L2 SHOT-R	L_∞ SHOT-R
16	1	1	1.56E-01	3.71E-01	1.27E-00	2.06E-01	5.33E-01	2.05E-00	1.04E-01	2.99E-01	1.33E-00
	2	1	9.62E-02	2.47E-01	1.15E-00	1.33E-01	3.13E-01	1.08E-00	1.10E-01	2.68E-01	9.09E-01
	3	1	2.61E-01	6.33E-01	2.35E-00	1.33E-01	3.07E-01	1.05E-00	7.78E-02	1.87E-01	7.58E-01
8	1	1	1.79E-01	4.39E-01	1.72E-00	1.87E-01	4.29E-01	1.31E-00	1.38E-01	3.26E-01	1.06E-00
	2	1	4.58E-01	1.44E-00	7.89E-00	2.21E-01	5.36E-01	2.42E-00	1.82E-01	4.47E-01	2.06E-00
	3	1	1.41E-01	3.70E-01	1.21E-00	1.24E-01	2.66E-01	7.10E-01	7.07E-02	1.55E-01	4.43E-01
4	1	1	2.40E-01	5.93E-01	2.16E-00	3.70E-01	8.63E-01	2.80E-00	3.00E-01	6.93E-01	2.18E-00
	2	1	2.56E-01	6.02E-01	1.96E-00	3.29E-01	7.40E-01	2.45E-00	2.14E-01	4.86E-01	1.68E-00
	3	1	3.60E-01	8.86E-01	3.34E-00	1.79E-01	4.07E-01	1.30E-00	2.12E-01	4.94E-01	1.55E-00
2	2	1	3.99E-01	9.20E-01	2.90E-00	2.54E-01	6.00E-01	2.51E-00	1.74E-01	4.16E-01	1.74E-00
	3	1	1.90E-01	4.42E-01	1.42E-00	1.53E-01	3.71E-01	1.47E-00	1.22E-01	2.61E-01	8.99E-01
16	1	2	1.51E-01	3.59E-01	1.19E-00	1.80E-01	4.62E-01	1.82E-00	9.38E-02	2.72E-01	1.34E-00
	1	2	1.53E-01	3.48E-01	1.26E-00	1.53E-01	3.44E-01	1.14E-00	1.17E-01	2.59E-01	7.29E-01
	2	2	7.09E-02	1.50E-01	3.86E-01	1.11E-01	2.42E-01	6.59E-01	9.99E-02	2.18E-01	5.80E-01
	2	2	6.49E-02	1.60E-01	5.88E-01	9.79E-02	2.21E-01	6.70E-01	1.29E-01	3.02E-01	9.75E-01
8	3	2	8.12E-02	2.26E-01	8.15E-01	1.14E-01	2.54E-01	7.04E-01	1.32E-01	3.09E-01	8.87E-01
	1	2	1.88E-01	4.50E-01	1.77E-00	1.32E-01	3.52E-01	1.68E-00	9.92E-02	2.36E-01	1.05E-00
	2	2	8.18E-02	1.98E-01	8.92E-01	1.11E-01	2.53E-01	8.87E-01	1.05E-01	2.36E-01	7.00E-01
	2	2	1.13E-01	2.67E-01	1.06E-00	1.50E-01	3.49E-01	1.08E-00	1.06E-01	2.35E-01	7.96E-01
4	3	2	3.37E-01	8.25E-01	2.38E-00	3.65E-01	8.24E-01	2.24E-00	2.74E-01	6.28E-01	1.76E-00
	1	2	2.67E-01	6.23E-01	1.90E-00	3.36E-01	7.66E-01	2.23E-00	3.10E-01	7.05E-01	2.08E-00
	2	2	1.19E-01	3.01E-01	1.09E-00	7.79E-02	1.85E-01	7.47E-01	4.20E-02	1.02E-01	3.46E-01
	3	2	2.77E-01	5.88E-01	1.57E-00	3.63E-01	7.63E-01	1.93E-00	1.84E-01	3.90E-01	1.09E-00
2	3	2	1.44E-01	3.22E-01	8.72E-01	1.37E-01	2.97E-01	7.38E-01	8.51E-02	1.86E-01	5.45E-01
	3	2	3.72E-01	8.87E-01	3.11E-00	1.40E-01	3.47E-01	1.68E-00	1.31E-01	3.28E-01	1.52E-00
16	3	2	2.68E-01	6.34E-01	2.62E-00	2.23E-01	5.34E-01	2.53E-00	1.40E-01	3.34E-01	1.45E-00
	1	3	2.65E-01	6.43E-01	2.32E-00	1.14E-01	2.64E-01	1.00E-00	8.41E-02	1.96E-01	7.59E-01
	2	3	1.54E-01	3.66E-01	1.20E-00	2.22E-01	5.65E-01	2.14E-00	1.09E-01	3.12E-01	1.37E-00
8	3	3	4.57E-02	1.25E-01	4.25E-01	7.13E-02	1.55E-01	4.59E-01	6.63E-02	1.51E-01	4.47E-01
	1	3	1.80E-01	4.40E-01	1.73E-00	1.94E-01	4.43E-01	1.32E-00	1.44E-01	3.39E-01	1.12E-00
	2	3	4.34E-01	1.35E-00	7.49E-00	2.10E-01	5.15E-01	2.21E-00	1.78E-01	4.37E-01	2.14E-00
4	3	3	7.86E-02	1.85E-01	5.24E-01	9.44E-02	2.03E-01	6.11E-01	7.78E-02	1.65E-01	4.20E-01
	1	3	2.68E-01	6.36E-01	2.19E-00	3.74E-01	8.78E-01	2.93E-00	2.96E-01	6.91E-01	2.20E-00
	2	3	1.13E-01	2.82E-01	9.85E-01	1.46E-01	3.41E-01	1.17E-00	1.27E-01	3.03E-01	1.05E-00
2	3	3	3.15E-01	7.71E-01	2.96E-00	3.26E-01	7.42E-01	2.54E-00	1.87E-01	4.26E-01	1.49E-00
	2	3	4.28E-01	1.01E-00	3.44E-00	2.34E-01	5.59E-01	2.28E-00	1.60E-01	4.03E-01	1.77E-00
3	3	1.03E-01	2.56E-01	1.09E-00	2.44E-01	5.34E-01	1.61E-00	1.57E-01	3.48E-01	1.13E-00	

Comparison with cubic splines Finally, to statistically compare the errors generated by the SHOT-R and the cubic spline algorithms, we perform the same test illustrated in Section 4.3 considering SHOT-R and cubic spline reconstructions over the entire dataset. The new SHOT-R algorithm always provides lower errors in all norms, thus proving higher accuracy in the tested biological dataset (Fig. 8).

We also measure the computational cost of the SHOT-R algorithm compared against the SPT scheme and the cubic spline. All three methods are run 10 times on the entire biological dataset used in Section 4.3 and 4.4, and the computational time is measured in seconds. The results are reported in Table 4. For SHOT-R, we separately monitor the cost of three different parts of the algorithm: i) central (unlimited) reconstruction, which corresponds to the solution of system (14); ii) one-sided linear polynomials evaluated with (18); iii) nonlinear WENO weights given by (20)-(21). As expected, the SPT reconstruction is the less expensive method, though the less accurate, since it does not even need any form of nonlinear limiting, being intrinsically compliant with the discrete maximum principle on the interpolated quantity. On the other hand, the unlimited central reconstruction of SHOT-R is always less expensive than the cubic spline, because no global linear system on the entire computational mesh has to be solved in the SHOT-R scheme. If the reconstructed trajectory is smooth enough, one could only use this part of the algorithm to interpolate the desired quantity over the computational domain, hence using the most efficient scheme among the three. However, if discontinuous profiles are present, the nonlinear CWENO reconstruction generates an additional computational cost for the evaluation of the one-sided linear polynomials and the final nonlinear weights. This leads to higher computational efforts compared to cubic splines, which however

Table 4

Computational cost of three different reconstruction algorithms run 10 times on the entire dataset used in Section 4.3 and 4.4. The schemes are as follows: SPT followed by linear linking, SHOT-R with P3, cubic splines. For SHOT-R we distinguish three parts of the algorithm: central reconstruction, one-sided reconstruction, computation of the nonlinear weights for the CWENO limiter. The computational time is given in seconds [s], and the average computational times are provided in the last line of the table.

# Run	SPT	SHOT-R				Cubic Spline
		Central Rec.	One-sided Rec.	CWENO weights	Total time	
1	0.3437	0.4675	0.2812	0.1562	0.9049	0.5781
2	0.2500	0.3594	0.1875	0.1719	0.7188	0.7656
3	0.2031	0.4062	0.2500	0.0781	0.7343	0.6406
4	0.1406	0.2656	0.1875	0.0780	0.5311	0.6719
5	0.1875	0.2344	0.2031	0.1094	0.5469	0.6406
6	0.0937	0.5469	0.2344	0.1719	0.9532	0.6875
7	0.1562	0.5312	0.2187	0.1718	0.9217	0.5625
8	0.1719	0.3906	0.2812	0.1562	0.8280	0.6719
9	0.1719	0.4219	0.2500	0.1562	0.8281	0.5781
10	0.1719	0.4219	0.2188	0.2812	0.9219	0.5469
Avg	0.1890			0.7889		0.6344

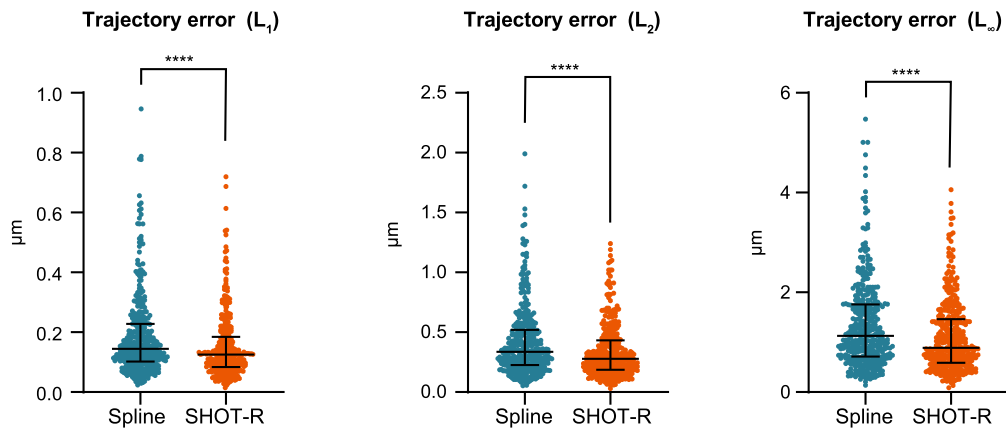


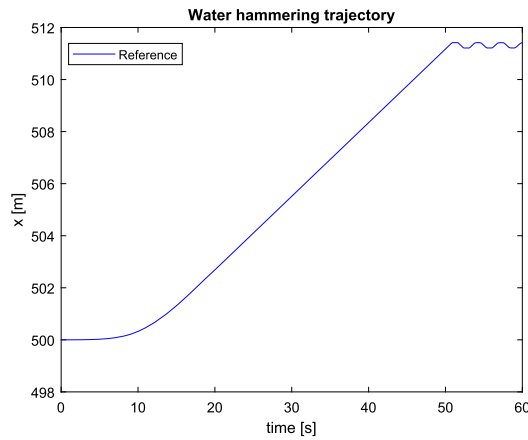
Fig. 8. Comparison SHOT-R and spline reconstruction based on backward trajectory integration on a biological dataset. Difference between SHOT-R and cubic Spline measured as error norms L_1 , L_2 , L_∞ of the reconstructed trajectory based on velocity trace. Abbreviations: SHOT-R, Spatiotemporal High-Order Trajectory Reconstruction. Data information: **** $P < 0.0001$. Values are median with interquartile range. Data are not normally distributed (Shapiro-Wilk test), two-tailed Wilcoxon matched pairs test. Each data point corresponds to a particle. Total number of analyzed particles: 372.

remain unbounded and possibly prone to oscillations. Despite a larger computational effort, SHOT-R exhibits higher accuracy as evident from Table 3 at a computational cost that is 25% larger than the cubic spline.

4.5. An example of application to hydraulics

In order to show the versatility of the SHOT-R method, we consider a completely different application in hydraulics: water hammering. This phenomenon is generated by a high pressure wave traveling in a pressurized pipe, commonly used for instance in water supply systems. The normal operational setting of the pipeline is given by a constant flow, hence leading to a linear pressure drop and a constant velocity in the pipe. If the flow is suddenly forced to stop because of the closure of a valve at the end of the pipeline, a strong pressure wave arises which may yield severe problems like vibrations or even the rupture of the pipe. This huge pressure jump travels backward and forward along the pipe, and it is also called water hammer or hydraulic shock in hydraulics, see [46].

For this test, we trace a particle that travels in a pipe of length 1 km with an equilibrium radius of $R_0 = 0.15$ m. The water demand at the end of the pipe is of 50 liters per seconds, and the pressure prescribed at the beginning of the tube is 0.59 GPa that is equivalent to a water column of 50 m. The viscous flow is modeled by the Darcy-Weisbach relation, and an elastic pipe with rigidity coefficient of $\beta = 1.7e11$ is considered for the solid material. The reference trajectory of the particle is computed numerically by means of a one-dimensional semi-implicit algorithm, similar to the one presented in [47]. The fluid initially accelerates up to the steady velocity of 0.7074 m/s. At $t = 50.5$ s, the water demand is stopped by a very fast operation of 0.1 s, mimicking the sudden closure of a valve in the pipeline. The particle is initially placed at $x = 500$ m and is tracked during the operation. The resulting position is shown in the top left panel of Fig. 9 where the effect of the hydraulic shock leads to a fast change of the particle velocity in the pipeline.



time [0; 49]	L_1	L_2	L_∞
SPT	2.9372E-03	6.0245E-03	1.2984E-02
SHOT-R	3.9047E-05	7.7818E-05	2.1259E-04
Spline	1.2266E-05	4.4507E-05	3.2224E-04

time [50; 60]	L_1	L_2	L_∞
SPT	4.7426E-03	8.7236E-03	1.9047E-02
SHOT-R	2.3396E-04	4.4411E-04	1.1233E-03
Spline	2.6222E-04	5.0501E-04	1.2425E-03

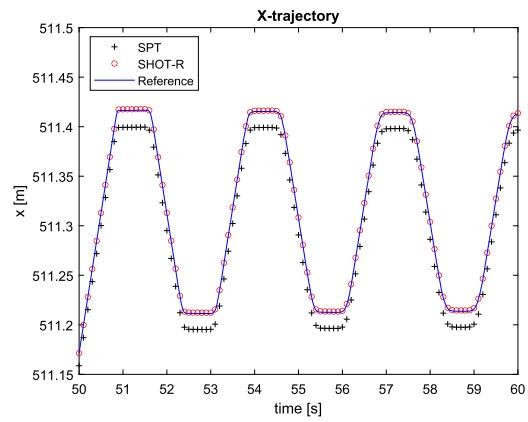
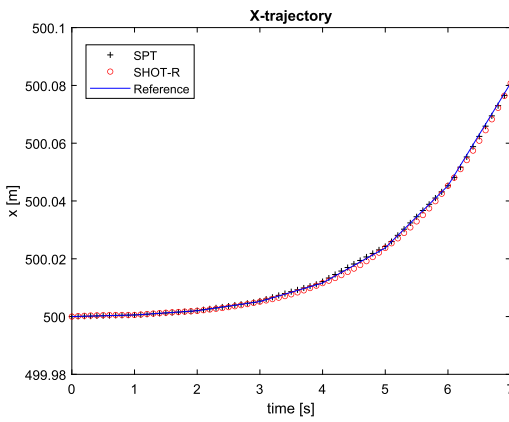


Fig. 9. Water hammering example. Top: reference particle trajectory (left) and errors in L_1 , L_2 and L_∞ norm of the backward integrated trajectory using SPT, SHOT-R and spline reconstructions for the head and the tail of the pipe (time interval [0; 49] for the head and [50; 60] for the tail). Bottom: comparison of the reconstructed trajectory for SPT and SHOT-R against the reference solution with a zoom on the head (left) and on the tail (right) of the pipe.

This dataset is then used to perform a high order trajectory reconstruction with the new SHOT-R method. Both position and velocity of the particle are reconstructed, and the same test presented in Section 4.3 is carried out. Therefore, the trajectory of the particle is integrated backward in time using the reconstructed velocity till the initial time of the reference trajectory is achieved. The same trajectory integration is carried out also employing a second order accurate SPT algorithm as well as a cubic spline reconstruction. In Fig. 9 we plot the comparison of SPT and SHOT-R reconstructed trajectories against the reference solution, where one can clearly notice the better approximation achieved by the SHOT-R method. The improvement is particularly evident at the end of the time series, where the pressure wave causes severe oscillations in the particle trajectory. On the other hand, if the trajectory is almost linear, as before the pipe closure, even a simple SPT method provides very good results since it is capable of exactly reproducing second order trajectories. On the right panel of Fig. 9, a table reports the errors obtained with SPT, SHOT-R and spline reconstruction algorithms, confirming that the SHOT-R method systematically achieves a lower error. In fact, the errors are measured both for the head and the tail of the trajectory by separately analyzing the time intervals [0; 49] s and [50; 60] s, respectively. The results obtained with the spline reconstruction are comparable with the ones of SHOT-R in terms of error norms, while both outperform the SPT linear approach over the entire trajectory.

5. Conclusions

In live-imaging, a series of frames capturing particle position over time is acquired. Retrieving trajectories from these images means approximating the entire particle path. Any linear linking algorithm describes a particle path by joining the known spatial coordinates with straight lines. Here, we reconstruct particle trajectories using polynomials of higher accuracy than second order (i.e. simple straight lines) to better simulate the original trajectory under study.

The reconstruction process is based on the construction of local high order polynomials within each time interval, that are expressed in terms of an expansion with modal Taylor basis functions. To control the spurious oscillations that may arise from the interpolation of discontinuous datasets, we rely on a CWENO limiting strategy. Differently from spline interpolations, that require the solution of a global system on the entire computational grid, we solve local overdetermined linear systems with constraints that ensure the resulting polynomial to be continuous at cell boundaries.

We face two different challenges: i) the reconstruction of 3D curvilinear trajectories in time (i.e. solving a 4D computational reconstruction), ii) the demonstration of more accurate results using SHOT-R over SPT and linear linking considering the lack of a reference solution in real live-imaging experiment. In our method, we solve these two problems, making them two strengths of SHOT-R itself. First, to overcome the computational challenge of solving a 4D problem, SHOT-R reconstructs particle trajectories starting from the raw space-time coordinates and splitting the trajectories along each spatial direction (x , y , z), then it reassembles the 4D information by gathering the reconstruction along each axis in 1D without sacrificing the spatial or temporal resolution. The second challenge is approached by measuring the difference between the initial reconstructed trajectory and the one retrieved from the velocities by backward time integration (Fig. 6). The more accurate the velocity the smaller the difference between the two trajectories. We consider this as an important test to quantify how much our method gains in accuracy compared to SPT in any tested dataset. Indeed, when the particles under study do not move or move at a constant velocity, SHOT-R is not expected to bring more information than what the classical SPT and linear linking do (Fig. 2). On the other hand, when the moving object (e.g. molecules, virus, vesicles, organelles, or whole cells) shows changes in direction and/or velocity, SHOT-R improves the accuracy in the analysis.

Beside checking the accuracy of SHOT-R over SPT current strategies, it is crucial to understand which are the new information that can be extracted using the novel algorithm. Specifically, SHOT-R is based on high order and piecewise continuous trajectory reconstructions, hence broadening the range of information that can be unraveled on particle kinematics. Indeed, a higher order method improves the accuracy in computing particle velocities and would allow to investigate novel kinematics features (i.e. acceleration). Moreover, the SHOT-R continuum trajectory reconstruction would enable i) to assess instantaneous velocity and acceleration at any space-time point, ii) to define phenomena based on time independently of the frequency in acquisition (e.g. pausing defined for a temporal threshold in seconds and not in frames), iii) to study co-trafficking events by automatically identify particles moving together, defining a common reference trajectory thus enabling to uncover the kinematics of co-trafficking.

Our numerical scheme is extremely versatile: it could be applied to any live-imaging experiment where the space-time coordinates can be retrieved. An application to hydraulics has also been shown to further demonstrate the flexibility of the proposed approach which is independent of the tracked physical phenomenon. In order to address different biological questions, SHOT-R could provide an overview on particle dynamics based on the entire trajectory or portions of trajectory, from average to instantaneous behavior, accounting for complex bidirectional motions interspersed with pauses. Moreover, it would enable to study particle directionality along different reference vectors depending on the biological questions, and to investigate the kinematics of co-trafficked particles.

Future research will concern the inclusion of uncertainty quantification that is linked to the acquisition process. To allow for a non-intrusive analysis, one might employ collocation methods to account for the uncertainty in the spatial coordinates, as recently proposed in the context of epidemic data [48].

CRedit authorship contribution statement

Eloina Corradi: Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization. **Maurizio Tavelli:** Methodology, Software, Validation, Writing - Review & Editing, Visualization. **Marie-Laure Baudet:** Resources, Supervision, Writing - Review & Editing, Project Administration, Funding Acquisition. **Walter Boscheri:** Conceptualization, Methodology, Software, Validation, Writing - Original Draft, Writing - Review & Editing, Resources, Project Administration, Funding Acquisition.

Data and code availability

The code is implemented in Matlab programming language, version R2022b. Data and source code are available from the corresponding authors upon request.

Acknowledgements

M.-L.B. acknowledges financial support from the grants: Marie Curie Career Integration (618969 GUIDANCE-miR), The G. Armenise-Harvard Foundation, MIUR SIR (RBSI144NZ4).

W.B. received financial support by Fondazione Cariplo and Fondazione CDP (Italy) under the project No. 2022-1895 and by the Italian Ministry of University and Research (MUR) with the PRIN Project 2022 No. 2022N9BM3N.

M.T. and W.B. are members of the National Group of Scientific Computing GNCS-INdAM.

Appendix A. Particle coordinates

In our biological applications, pre-miRNAs particle xy coordinates are extracted from published movies [5] with TrackMate v5.2.0 [34]. Raw movies are cropped the axonal portion to analyze. The leading tip of the axon, the growth cone, is kept on the right side of each crop, which is not included in the analysis since the high density in particle signal does not allow to recognize single pre-miRNA puncta. Occasionally, background correction with rolling ball pixel size 10 is applied to increase signal to noise ratio in ImageJ. Difference of Gaussian particle detection (DoG) is used as detector in TrackMate. The estimated diameter is set to 0.6 μm and the threshold in the TrackMate detection step is fixed from 10 to 400 depending on the signal to noise ratio of each movie. Median filter, reducing the generation of spurious spots in case of salt and pepper noise, and sub-pixel localization options are selected. The tracking settings are: link max distance “1”; gap closing max distance “2” and gap closing max frame gap “4” (for movies with single

channel acquisition); link max distance “5”; gap closing max distance “2” and gap closing max frame gap “1” (for movies with both green and red channel). Splitting or merging events are not expected, hence Simple Linear Assignment Problem (LAP) Tracker is used. No filters are applied to the track step. The automatically tracked lines are manually checked and minor adjustments applied (i.e. remove signal coming from other axons present in the same field; link tracks belonging to the same particle, only when the particle is visible but not detected due to the applied threshold). Prior to analysis a filter on number of consecutive frames per track is applied in order to clear spurious links done by the automatic tracking. For movies on pre-miRNA endogenous trafficking the minimal number of frames is set at 20, for co-trafficking movies at 5.

Appendix B. Computing error norms

To quantitatively appreciate the accuracy of the results, the errors between the reconstructed and a reference solution for both position and velocity are measured in L_1 , L_2 and L_∞ norms, that for a generic spatial direction s are

$$\begin{aligned}
 L_1 &= \int_{\Omega} \|s_e(t) - s_h(t)\| dt, \\
 L_2 &= \sqrt{\int_{\Omega} \|s_e(t) - s_h(t)\|^2 dt}, \\
 L_\infty &= \max_{\Omega} \|s_e(t) - s_h(t)\|.
 \end{aligned}
 \tag{B.1}$$

The reference solution s_e depends on the specific application. In particular, it is given by (47) and (45) for position and velocity (Fig. 3, 4) or the $P3$ polynomial (Fig. 6). Then, $s_h(t)$ denotes the numerical solution computed with SHOT-R (or the linear reconstruction in Fig. 3, 4). The integrals appearing in (B.1) are evaluated using Gaussian quadrature formulae of suitable accuracy (see [40]).

Appendix C. Backward integration of particle trajectory

The results shown in Fig. 6C-E have been obtained by integrating backward in time the trajectory of the particle using the SHOT-R strategy. From the mathematical viewpoint, this corresponds to the solution of the ODE

$$\frac{d\mathbf{x}}{dt} = -\mathbf{v}(\mathbf{x}, t), \quad \mathbf{x}_0 = \mathbf{x}_{N_K}, \quad t \in [0; t_{N_K} - t_1],
 \tag{C.1}$$

with $D = 2$ and therefore $\mathbf{x} = (x, y)$ and $\mathbf{v} = (v_x, v_y)$. The initial condition \mathbf{x}_0 of the ODE is given by the xy position of the particle at the end of the trajectory, which coincides with the last input point N_K . We proceed as follows. Firstly, SHOT-R method is employed for reconstructing the particle trajectory, i.e. for obtaining $p_i^N(\mathbf{x}, t)$ for all cells i . Secondly, the velocity vector \mathbf{v}_i is computed with (32), hence obtaining a high order velocity field defined within each computational cell. Finally, (C.1) is solved up to the final time $t_f = t_{N_K} - t_1$ in such a way that the position of the particle at the beginning of the trajectory is retrieved. Obviously, due to numerical approximation, this will never coincide with the input coordinates of the first point \mathbf{x}_1 . However, we have shown that higher order reconstructions provide a more accurate results, that is the final reconstructed coordinates which arise from the solution of the ODE (C.1) are closer to the original first point of the trajectory taken as input for the trajectory reconstruction.

In order to solve (C.1) any classical ODE integrator can be used, provided that it satisfies at least the same accuracy property of the polynomial reconstruction. Here, we rely on a very popular numerical technique, namely the class of Runge-Kutta (RK) schemes [41]. For $P1$ reconstruction (SPT) we use a second order RK method, while for $P3$ (SHOT-R) the fourth order version is adopted. Runge-Kutta schemes explicitly write, for every intermediate step, as follows:

- second order Runge-Kutta (RK2)

$$\begin{aligned}
 \mathbf{x}_{n+1} &= \mathbf{x}_n + \frac{d\tau}{2} (\mathbf{k}_1 + \mathbf{k}_2), \\
 \mathbf{k}_1 &= -\mathbf{v}(\tau_n, \mathbf{x}_n), \\
 \mathbf{k}_2 &= -\mathbf{v}\left(\mathbf{x}_n + \frac{d\tau}{2}\mathbf{k}_1, \tau_n + \frac{d\tau}{2}\right);
 \end{aligned}
 \tag{C.2}$$

- fourth order Runge-Kutta (RK4)

$$\begin{aligned}
 \mathbf{x}_{n+1} &= \mathbf{x}_n + \frac{d\tau}{6} (\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4), \\
 \mathbf{k}_1 &= -\mathbf{v}(\tau_n, \mathbf{x}_n), \\
 \mathbf{k}_2 &= -\mathbf{v}\left(\mathbf{x}_n + \frac{d\tau}{2}\mathbf{k}_1, \tau_n + \frac{d\tau}{2}\right), \\
 \mathbf{k}_3 &= -\mathbf{v}\left(\mathbf{x}_n + \frac{d\tau}{2}\mathbf{k}_2, \tau_n + \frac{d\tau}{2}\right), \\
 \mathbf{k}_4 &= -\mathbf{v}(\mathbf{x}_n + d\tau\mathbf{k}_3, \tau_n + d\tau).
 \end{aligned}
 \tag{C.3}$$

The time step $d\tau$ is used to integrate the ODE (C.1) within the time interval $[0; t_{N_K} - t_1]$ and the superscript n denotes the numerical solution at a given time step. $\tau \in [0; t_{N_K} - t_1]$ represents the local time coordinate which advances the solution in time. We choose $d\tau = 0.5$ which corresponds to approximately 3.5 times the time frame of acquisition of the biological dataset used as proof of concept (i.e. 0.144 sec). This has been deliberately chosen in order to highlight even more the benefit of a high order reconstruction, because it contains much more information and thus can produce very good results even with large time steps which cover more than three consecutive frames, i.e. more than three adjacent computational cells. If classical linear tracking is used, i.e. $P1$ reconstruction is applied, the results suffer from lack of resolution (Fig. 6) and the accuracy is drastically reduced.

References

- [1] X. Wang, T.L. Schwarz, The mechanism of Ca²⁺-dependent regulation of kinesin-mediated mitochondrial motility, *Cell* 136 (2009) 163–174.
- [2] J.R. Daniele, R.M. Baqri, S. Kunes, Analysis of axonal trafficking via a novel live-imaging technique reveals distinct hedgehog transport kinetics, *Biol. Open* 6 (2017) 714–721.
- [3] V. Lacovich, et al., Tau isoforms imbalance impairs the axonal transport of the amyloid precursor protein in human neurons, *J. Neurosci.* 37 (2017) 58–69.
- [4] J.D. Fenn, C.M. Johnson, J. Peng, P. Jung, A. Brown, Kymograph analysis with high temporal resolution reveals new features of neurofilament transport kinetics, *Cytoskeleton* 75 (2018) 22–41.
- [5] E. Corradi, et al., Axonal precursor mi rna s hitchhike on endosomes and locally regulate the development of neural circuits, *EMBO J.* 39 (2020) e102513.
- [6] D. Reid, An algorithm for tracking multiple targets, *IEEE Trans. Autom. Control* 24 (1979) 843–854.
- [7] V. Racine, et al., Visualization and quantification of vesicle trafficking on a three-dimensional cytoskeleton network in living cells, *J. Microsc.* 225 (2007) 214–228.
- [8] A. Sergé, N. Bertaux, H. Rigneault, D. Marguet, Dynamic multiple-target tracing to probe spatiotemporal cartography of cell membranes, *Nat. Methods* 5 (2008) 687–694.
- [9] K. Jaqaman, et al., Robust single-particle tracking in live-cell time-lapse sequences, *Nat. Methods* 5 (2008) 695–702.
- [10] N. Chenouard, I. Bloch, J.-C. Olivo-Marin, Multiple hypothesis tracking for cluttered biological image sequences, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (2013) 2736–3750.
- [11] N. Chenouard, et al., Objective comparison of particle tracking methods, *Nat. Methods* 11 (2014) 281–289.
- [12] R. Jonker, A. Volgenant, A shortest augmenting path algorithm for dense and sparse linear assignment problems, *Computing* 38 (1987) 325–340.
- [13] M.K. Cheezum, W.F. Walker, W.H. Guilford, Quantitative comparison of algorithms for tracking single fluorescent particles, *Biophys. J.* 81 (2001) 2378–2388.
- [14] H.Y. Park, A.R. Buxbaum, R.H. Singer, Single mrna tracking in live cells, *Methods Enzymol.* 472 (2010) 387–406.
- [15] E. Meijering, O. Dzyubachyk, I. Smal, Chapter nine-methods for cell and particle tracking, in: P.M. Conn (Ed.), *Imaging and Spectroscopic Analysis of Living Cells Optical and Spectroscopic Techniques*, in: *Methods in Enzymology*, vol. 504, 2012.
- [16] W. Boscheri, A space-time semi-Lagrangian advection scheme on staggered Voronoi meshes applied to free surface flows, *Comput. Fluids* 202 (2020) 104503.
- [17] A. Iske, M. Käser, Conservative semi-Lagrangian advection on adaptive unstructured meshes, *Numer. Methods Partial Differ. Equ.* 20 (2004) 388–411.
- [18] G. Tumolo, L. Bonaventura, M. Restelli, A semi-implicit, semi-Lagrangian, p-adaptive discontinuous Galerkin method for the shallow water equations, *J. Comput. Phys.* 232 (2013) 46–67.
- [19] J.-M. Qiu, C. Shu, Conservative high order semi-Lagrangian finite difference WENO methods for advection in incompressible flow, *J. Comput. Phys.* 230 (2011) 863–889.
- [20] W. Boscheri, M. Tavelli, L. Pareschi, On the construction of conservative semi-Lagrangian IMEX advection schemes for multiscale time dependent PDEs, *J. Sci. Comput.* 90 (2022).
- [21] W. Boscheri, M. Dumbser, M. Righetti, A semi-implicit scheme for 3d free surface flows with high-order velocity reconstruction on unstructured Voronoi meshes, *Int. J. Numer. Methods Fluids* 72 (2013) 607–631.
- [22] S. Clain, S. Diot, R. Loubère, A high-order finite volume method for systems of conservation laws - multi-dimensional optimal order detection, *J. Comput. Phys.* 230 (2011) 4028–4050.
- [23] G. Albi, M. Herty, L. Pareschi, Linear multistep methods for optimal control problems and applications to hyperbolic relaxation systems, *Appl. Math. Comput.* 354 (2019) 460–477.
- [24] J. Mulder, Q. Chu, J. Sridhar, J. Breeman, M. Laban, Non-linear aircraft flight path reconstruction review and new advances, *Prog. Aerosp. Sci.* 35 (1999) 673–726.
- [25] J. Rayner, H. Aldridge, Three-dimensional reconstruction of animal flight paths and the turning flight of microchiropteran bats, *J. Exp. Biol.* 118 (1985) 247–265.
- [26] X. Tian, et al., Direct measurements of the kinematics and dynamics of bat flight, *Bioinspir. Biomim.* 1 (2006) S10.
- [27] N. Malik, T. Dracos, D. Papantoniou, Particle tracking velocimetry in three-dimensional flows: part II: particle tracking, *Exp. Fluids* 15 (1993) 279–294.
- [28] S. Godunov, Finite difference methods for the computation of discontinuous solutions of the equations of fluid dynamics, *Mat. Sb.* 47 (1959) 271–306.
- [29] T. Barth, D. Jespersen, The design and application of upwind schemes on unstructured meshes, *AIAA Paper* 89-0366, 1989, pp. 1–12.
- [30] T. Barth, P. Frederickson, Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction, in: 28th Aerospace Sciences Meeting, January 1990, AIAA paper no. 90-0013.
- [31] C. Manzo, M.F. Garcia-Parajo, A review of progress in single particle tracking: from methods to biophysical insights, *Rep. Prog. Phys.* 78 (2015) 124601.
- [32] M. Lelek, et al., Single-molecule localization microscopy, *Nat. Rev. Methods Primers* 1 (2021) 39.
- [33] P. Parutto, et al., High-throughput super-resolution single-particle trajectory analysis reconstructs organelle dynamics and membrane reorganization, *Cell Rep. Methods* 2 (2022).
- [34] J.-Y. Tinevez, et al., TrackMate: an open and extensible platform for single-particle tracking, *Methods* 115 (2017) 80–90.
- [35] D. Levy, G. Puppo, G. Russo, Central WENO schemes for hyperbolic systems of conservation laws, *Math. Model. Numer. Anal.* 33 (1999) 547–571.
- [36] C.-W. Shu, High-order finite difference and finite volume weno schemes and discontinuous Galerkin methods for cfd, *Int. J. Comput. Fluid Dyn.* 17 (2003) 107–118.
- [37] M. Dumbser, W. Boscheri, M. Semplice, G. Russo, Central weighted ENO schemes for hyperbolic conservation laws on fixed and moving unstructured meshes, *Comput. Fluids* 39 (2017) A2564–A2591.
- [38] I. Cravero, G. Puppo, M. Semplice, G. Visconti, CWENO: uniformly accurate reconstructions for balance laws, *Math. Comput.* 87 (2018) 1689–1719.
- [39] M. Dumbser, C. Enaux, E. Toro, Finite volume schemes of very high order of accuracy for stiff hyperbolic balance laws, *J. Comput. Physiol.* 227 (2008) 3971–4001.
- [40] A. Stroud, *Approximate Calculation of Multiple Integrals*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1971.
- [41] R. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations*, SIAM, Philadelphia, Pennsylvania, 2007.
- [42] S. Maday, A.E. Twelvetrees, A.J. Moughamian, E.L. Holzbaur, Axonal transport: cargo-specific mechanisms of motility and regulation, *Neuron* 84 (2014) 292–309.
- [43] K.-M. Leung, et al., Cue-polarized transport of β -actin mRNA depends on 3' UTR and microtubules in live growth cones, *Front. Cell. Neurosci.* 12 (2018) 300.

- [44] K.S. Lee, T.J. Kim, G. Pratz, Single-cell tracking with PET using a novel trajectory reconstruction algorithm, *IEEE Trans. Med. Imaging* 34 (2014) 994–1003.
- [45] A. Patron-Perez, S. Lovegrove, G. Sibley, A spline-based trajectory representation for sensor fusion and rolling shutter cameras, *Int. J. Comput. Vis.* 113 (2015) 208–219.
- [46] E.B. Wylie, *Fluid Transients*, FEB Pr, 1983.
- [47] M. Dumbser, U. Iben, M. Ioriatti, An efficient semi-implicit finite volume method for axially symmetric compressible flows in compliant tubes, *Appl. Numer. Math.* 89 (2015).
- [48] G. Bertaglia, W. Boscheri, G. Dimarco, L. Pareschi, Spatial spread of COVID-19 outbreak in Italy using multiscale kinetic transport equations with uncertainty, *Math. Biosci. Eng.* 18 (2021) 7028–7059.