SoftwareX 19 (2022) 101204

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

SerialTrack: ScalE and rotation invariant augmented Lagrangian particle tracking

Jin Yang [a], Yue Yin [a,b], Alexander K. Landauer [c], Selda Buyukozturk [a,d], Jing Zhang [a], Luke Summey [a], Alexander McGhee [a], Matt K. Fu [e], John O. Dabiri [e], Christian Franck [a,*]

[a] University of Wisconsin-Madison, Mechanical Engineering, Madison, WI, USA
[b] Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA
[c] National Institute of Standards and Technology, Gaithersburg, MD, USA
[d] School of Engineering, Brown University, Providence, RI, USA
[e] Graduate Aerospace Laboratories, California Institute of Technology, Pasadena, CA, USA

ARTICLE INFO

Article history:
Received 21 March 2022
Received in revised form 15 July 2022
Accepted 29 August 2022

Keywords:
Particle tracking
Topology-based feature vector
Augmented Lagrangian
Finite deformation

ABSTRACT

We present a new particle tracking algorithm for accurately resolving large deformation and rotational motion fields, which takes advantage of both local and global particle tracking algorithms. We call this method ScalE and Rotation Invariant Augmented Lagrangian Particle Tracking (SerialTrack). This method builds an iterative scale and rotation invariant topology-based feature vector for each particle within a multi-scale tracking algorithm. The global kinematic compatibility condition is applied as a global augmented Lagrangian constraint to enhance tracking accuracy. An open source software package implementing this numerical approach to track both 2D and 3D, incremental and cumulative deformation fields is provided.

Code metadata

Current code version	v1.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-22-00070
Code Ocean compute capsule	
Legal code license	MIT license
Code versioning system used	git
Software code languages, tools, and services used	MATLAB[a]
Compilation requirements, operating environments & dependencies	MATLAB with the following toolboxes: Curve Fitting Toolbox, Image Processing Toolbox, Parallel Computing Toolbox, Statistics and Machine Learning Toolbox, Wavelet Toolbox
If available link to developer documentation/manual	https://github.com/FranckLab/SerialTrack/manual.pdf
Support email for questions	cfranck@wisc.edu

[a]Certain commercial equipment, software and/or materials are identified in this paper in order to adequately specify the experimental procedure. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the equipment and/or materials used are necessarily the best available for the purpose.

1. Motivation and significance

Single particle tracking (SPT) and particle tracking velocimetry (PTV) methods provide quantitative, temporally resolved measurements of motions and deformations to investigate complex dynamics at the resolution of single tracking features by automatically localizing and tracking individual particles [1–3]. This is in contrast to digital image/volume correlation (DIC/DVC) and particle image velocimetry (PIV), which are subset-correlation-based techniques (see a summary of various full-field tracking methods in Table 1). The increased specificity of the tracked particles can be beneficial to applications such as quantitative biological motion tracking [4–9] and fluid mechanics for flow measurements [2,10–12]. However, it remains a challenge to uniquely and robustly match particles throughout an image sequence.

Particle tracking methods have been used to study multiple length and time scales in soft materials and rheology [13,14],

* Corresponding author.
E-mail address: cfranck@wisc.edu (Christian Franck).

https://doi.org/10.1016/j.softx.2022.101204

Table 1
Comparison of different full-field measurement methods.

Technique name	PIV	DIC	DVC	PTV	SPT	*SerialTrack (this work)*
Considering particle shape distortion or not	No	Yes	Yes	No	No	Yes
Matching continuous patches (C) or discrete feature points (D)	C	C	C	C	D	D
2D image sequence or 3D volumetric image	2D & 3D	2D	3D	2D & 3D	2D & 3D	2D & 3D
Tracking velocity field (v) or cumulative displacement (u)	v	u	u	v	u	v & u

Table 2
Comparison of different open-source particle tracking codes.

Name	Refs	Language	Dim.	Particle linking algorithm
PTVlab	[29,30]	MATLAB	2D	Integrated cross-correlation and relaxation algorithm
OpenPTV	[31]	C & Python	3D	A spatio-temporal matching algorithm [32]
TrackMate	[33,34]	MATLAB & Fiji	2D & 3D	LAP, u-track, Kalman filter, etc [35].
TracTrac	[36,37]	MATLAB & Python	2D	K-dimensional trees to search and compute statistics around neighboring objects
TPT	[27,38]	MATLAB	3D	Topology-based matching and iterative deformation warping (IDM)
FM-Track	[39]	Python	3D	Rotation-invariant topology-based matching
Part2Track	[40]	MATLAB	2D	Nearest neighbor searching or histogram matching
KNOT	[41]	Python	2D & 3D	Adaptive analysis on the single frame displacements produced from point clouds
Trackpy	[42,43]	Python	2D & 3D	Finds the most likely set of assignments that match each feature in the previous frame [13]
SerialTrack (this work)	[44]	MATLAB	2D & 3D	Scale and rotation-invariant topology-based matching and augmented Lagrangian global kinematic compatibility constraint

experimental physics [15], materials science [16,17], and geophysics [18]. There are a number of SPT and PTV algorithms created for various applications [19,20] but they are often specialized, and typically require either small inter-frame deformations [21–23] or sparse [24–26] /dense [27,28] particle seeding volumes to perform well.

Particle tracking procedures can generally be divided into two steps: (i) *particle localization*, where coordinates of individual particles are extracted from each frame of an image sequence, and (ii) *particle linking*, where detected particles are uniquely matched from frame to frame to construct a motion field.

Particle localization algorithms often decompose the process into particle detection followed by subpixel centroid localization. For example, images are pre-processed to reduce noise and selectively enhance objects, then particle spots or feature locations are detected by applying image segmentation, local-maxima finding, or other thresholding criteria. Particle centroid locations are often estimated by applying Gaussian fitting [45,46] or intensity-based centroid measurements [47,48]. In general, all of the above image processing-based methods perform well for images with sufficiently high signal-to-noise ratios (SNR $\geq$ 5). Recently, machine learning-based methods have been developed that can potentially improve performance in images with spatiotemporal heterogeneity and poor signal-to-noise ratios [33,41,49].

Various algorithms have been created to detect and track individual particles [1] such as the straightforward *k*-nearest-neighbor (*k*NN) searches, topology-based approaches where neighboring particles are employed to construct local surrounding topology features [27,36,50–52], globally optimized search problems — including linear assignment programming [26], Kalman filtering [53], relaxation methods [19,54], and feature vector-based techniques [22,25] (see a brief summary of particle tracking open-source codes in Table 2). Among these methods, the nearest neighbor-type search algorithms are typically suitable for relatively low numbers of particles that undergo displacements smaller than the typical interparticle separation distance. The more robust topology-based and feature-based particle tracking algorithms are able to resolve large deformation fields but favor large particle numbers. The relaxation-based approaches perform well on highly stochastic motion fields but require small inter-frame motions. The nearest-neighbor and local topology-based methods are computationally efficient and can be easily implemented in parallel. However, they both have limitations in regard to particle seeding densities and there is no guarantee that the final tracked motion fields are unique and kinematically admissible. Global optimization particle tracking methods can guarantee the uniqueness and kinematic admissibility of the tracked motion, but are typically computationally expensive.

Here we present a new particle tracking algorithm, called the **S**cal**E** and **R**otation **I**nvariant **A**ugmented **L**agrangian Particle Tracking (SerialTrack) method, which takes advantage of both local (i.e., nearest-neighbor search and local-topology-based feature tracking [27,40]) and global optimization to reconstruct motion fields in either 2D or 3D, and with large, complex deformations for both sparse and dense particles efficiently, robustly and accurately. This new method first builds a local scale and rotation invariant topology-based feature vector for each particle, then iteratively tracks these within a global multiscale algorithm. The global kinematic compatibility condition is applied as an augmented Lagrangian constraint [55,56] to enhance the tracking accuracy. In addition to tracking incremental deformation between two subsequent frames, SerialTrack can track large cumulative deformations where the initial guess of each tracking displacement field has been improved by a data-driven reduced order modeling method [57]. The new method includes both particle localization and particle linking processes, and also may optionally account for shape distortion of particles due to large deformations.

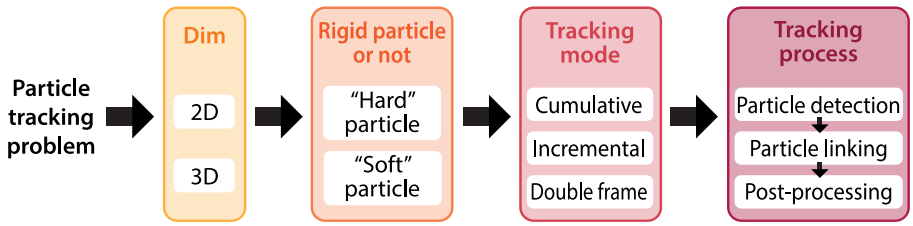

Fig. 1. SerialTrack's Workflow. The first step in SerialTrack's workflow is the specification of the application's dimensionality, i.e., 2D (pixel image) or 3D (volumetric voxel image). Hard vs. soft particles further specify whether tracker particle deformations are to be expected. The tracking mode defines the scheme for selecting images from the experimental image sequence. Finally, the particles detection, linking, and post-processing preferences are specified by the user based on the particular experimental configuration at hand and the desired output data.

2. Software description

2.1. Software architecture

The basic workflow of the SerialTrack implementation is summarized in Fig. 1. SerialTrack requires the users to provide their captured 2D and 3D (volumetric) image pairs or image sequence. In solid mechanics and material sciences, 3D image volumes can be scanned in multiple layers using confocal cameras, X-ray CT, MRI, or other imaging modalities. In fluid mechanics, these 3D volumes are typically reconstructed from several camera POVs using multi-view stereoscopy with one or more sensors, see for example [10,11]. The code package includes both 2D (SerialTrack2D) and 3D (SerialTrack3D) particle tracking examples and three executing modes: (i) *incremental*, (ii) *cumulative*, and (iii) *double frame*. In incremental mode, two consecutive frames are compared to infer incremental motion between frames, while in cumulative mode, later frames are compared with the first, undeformed reference frame to reconstruct their total motions. In 2D cases, we also include the "double frame" mode where two frames are taken under every single exposure with a temporal delay and each odd number frame is compared to its subsequent even number frame. One advanced feature of the 2D cumulative mode is that the effect of particle shape distortion can be considered. We consider particles to be "soft" if any particle shape change coincides with the local material deformation gradients, and consider particles "hard" if the shape is effectively rigid and invariant throughout the experiment.

During the tracking process, we detect each individual particle in both reference and deformed images and then link them to obtain the full-field motions throughout an image sequence. We will discuss these functionalities in Sections 2.2–2.4.

2.2. Software functionalities: particle detection

In the SerialTrack implementation, we leverage state-of-the-art particle detection methods where the minimum size of particles that can be effectively detected are usually 3 pixels by 3 pixels (see more details in the description of each detection method below and references therein). In the first of two approaches to detect individual single particles, we employ the same method used in Patel et al. [27] where images are thresholded based on a user-specified cutoff to segment particles from background as binary connected components, and then rapidly localize particle centroids with sub-pixel accuracy using the radial symmetry method [47,48]. In the second method, particles are detected by a Laplacian of Gaussian image filtering technique, followed by a Gaussian interpolation of the particle peak intensities [36,40]. Numerous techniques exist for particle segmentation and centroid localization — while these two perform well in our test cases, for different imaging modalities or particle types other methods may be more appropriate. Good particle detection, segmentation, and localization are critical for accurate tracking. In this regard, the code is highly extensible, and other algorithms can easily be added since the core algorithm simply expects a list of centroid coordinates for each image as input.

2.3. Software functionalities: particle linking

We describe our particle linking process in this section. We summarize the code outline in Algorithm 1. A mathematical formulation of the particle tracking problem is summarized in Appendix A, where a regularizer is added to the particle matching optimization functional (A.4) to enforce the uniqueness of the solution. In our implementation, this optimization problem is solved by the alternating direction method of multipliers (ADMM) scheme summarized in Appendix B. In each ADMM iteration, there are two displacement $\mathbf{u}$ and $\hat{\mathbf{u}}$. The vector $\mathbf{u}$ is the locally solved displacement from matching each individual particle's feature descriptor (see Section 2.3.1). The $\hat{\mathbf{u}}$ vector is the global displacement, obtained from projecting the local $\mathbf{u}$ vector onto the global, kinematically compatible space (see Appendix Eq. (A.4)). To solve the ADMM local step of Eq (B.2), particles are linked between frames using a new particle descriptor (i.e., feature vector), which is defined based on the topological arrangement of randomly located neighbors in a framework similar to those described in Patel et al. [27], Janke et al. [40], and Lejeune et al. [39]. Here we improve the previous topology-based relative neighbor feature to be 2D/3D scale- and rotation-invariant, as shown in Section 2.3.1.

During the iterative matching process, a universal outlier removal scheme [6] is also incorporated to enhance the quality of the reconstructed displacement field. During each ADMM iteration, we also detect and remove ghost particles, i.e., particles that are only detected once in the two compared frames, to improve the robustness of each particle's topology-based feature vector (see Section 2.3.2).

To track total, cumulative displacement fields, sometimes called Lagrangian particle tracking, we describe two strategies in Section 2.3.3. Finally, we also discuss minimizing errors due to particle shape distortions in Section 2.3.4, since in some experimental instances particles may deform significantly according to local deformation gradients, and thus shape change must be accounted for in the localization process.

2.3.1. Topology-based scale and rotation-invariant particle descriptor

For each individual particle, the relative position between the k nearest neighbor particles and the selected particle is encoded into a complete particle descriptor consisting of two feature vectors. For the 2D case, an angle-based feature is defined as an array of polar angles between each of the k neighboring particles, i.e., $[\theta_1, \theta_2, \ldots, \theta_k]^T$, as shown in Fig. 2(a–b). An array of interparticle Euclidean distances is also constructed as a distance-based feature, i.e., $[r_1, r_2, \ldots, r_k]^T$, where distances are normalized by

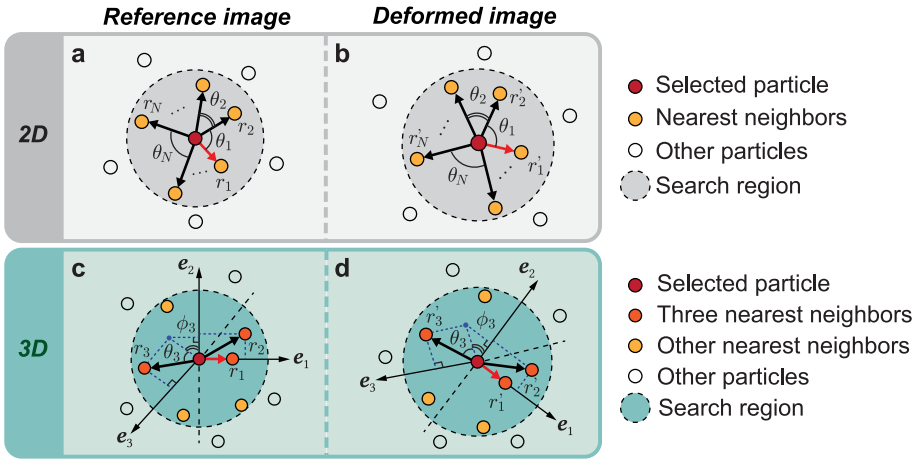

Fig. 2. Diagram outlining the descriptor generation process. (a) The k radii and angles to nearest neighbor particles within the search distance for each particle (b) The same computation is performed in the deformed image. By simultaneously minimizing Euclidean distance for angular and distance mismatch, we achieve a fast linking procedure that is scale and rotation invariant, and thus robust under most kinematically admissible deformations. (c & d) The analogous process for a 3D (volumetric) case, where the basis space for the descriptors is computed locally from each particle's three nearest neighbors.

the first nearest neighbor particle distance. For the 3D case, analogous to the 2D descriptor, 3D radial distances (r), polar angles (θ), and azimuthal angles (ϕ) of the stored k neighboring particles are used to construct particle descriptors for each particle. To establish a coordinate system, $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$, for each particle, we define the first nearest neighbor particle direction as the $\mathbf{e}_1$ axis. The $\mathbf{e}_3$ direction is defined to be perpendicular to the first and second nearest neighbor particles, and must satisfy $\mathbf{e}_3 \cdot \mathbf{r}_3 > 0$ where $\mathbf{r}_3$ is the third nearest neighbor particle direction. The $\mathbf{e}_2$ axis is defined as $\mathbf{e}_3 \times \mathbf{e}_1$ where " $\times$ " is the cross product. As in 2D, the radial distance (r) feature is the Euclidean interparticle distance normalized by the first nearest neighbor particle distance. The design of these descriptors is advantageous since they fully encode the relative spatial positions of neighboring particles. They can also be cheaper to compute compared to normalized correlation-based tracking algorithms, with a possible reduction in the computational cost on the order of (# of image pixels)/(# of particles). The constructed particle descriptors are scale and rotation invariant, which allows for the proper reconstruction of large deformations and rotations while retaining similarity between descriptors during tracking.

To establish successful particle matches between frames, the topology-based descriptor for each identified particle is computed by independently minimizing the Euclidean distance (summation of squared differences) between the distance-based feature and angle-based features, respectively. We consider two particles to be *matched*, i.e., to be the same physical particle in both frames, if they attain minimum radial and angular descriptors simultaneously. During the ADMM iterations, we apply a particle count scaling strategy, such that the number of nearest neighboring particles for each local matching step, k, is exponentially decreased from a user-selected starting value (typically 10's of particles) to 1. When k equals 1, our method is identical to the nearest neighbor search [40]. This scaling strategy helps to address variable particle densities — we have found that features relying on many local particles (high k) perform well for densely seeded regions, whereas features with lower k are better performing for sparsely seeded regions. In both cases, a maximum search radius for particles to include can be specified to reduce the overall computational cost while building the features.

2.3.2. Removing ghost particles

There may exist particles that are only detected once in the two frames, which are termed ghost particles [23]. These may occur when part of the sample moves out of the field of view, or due to experimental noise or occlusion. The existence of ghost particles has a two-fold effect on the local topology-based matching: the descriptor built in one frame for a ghost particle will not have a corresponding correct match in the second frame, and the feature descriptor of particles adjacent to the ghost particle will be degenerate as they will lack that particle in the subsequent frame. Both phenomena can have a deleterious effect on the accuracy of the particle linking and tracking steps. Therefore after solving each global ADMM iteration step of Eq (B.3), we attempt to detect and remove ghost particles and re-collect the centroid locations of correctly detected particles in both frames using the following criteria:

$$\begin{aligned} \mathbb{P}_n &\leftarrow \bigcup_{P_n} \left\{ \left(\min_{P_{n+t} \in \mathbb{P}_{n+t}} |P_n - P_{n+t}(\hat{\mathbf{u}})| \right) < \varepsilon_d \right\}, \\ \mathbb{P}_{n+t} &\leftarrow \bigcup_{P_{n+t}} \left\{ \left(\min_{P_n \in \mathbb{P}_n} |P_n - P_{n+t}(\hat{\mathbf{u}})| \right) < \varepsilon_d \right\}, \end{aligned} \tag{1}$$

Algorithm 1: Outline of the "hard" particle tracking procedure

Input: Two images f_n and f_{n+t}
Step 1: Detect particles in images f_n and f_{n+t};
Step 2: (Optional) Warp detected particle coordinates in f_{n+t} with a predictor for displacement $\hat{\mathbf{u}}$;
Step 3: Set dual variable $\boldsymbol{\theta}$ to be zero. Set $IterNum = 1$, and $M = 0$;
while $\left|\hat{\mathbf{u}}^{k+1} - \hat{\mathbf{u}}^{k}\right| > \varepsilon$ ***and*** $IterNum < IterMax$ ***and*** $M < 5$ **do**
  %%%%% **Subproblem 1**: fix $\hat{\mathbf{u}}^k$ and solve $\mathbf{u}^{k+1}$ locally %%%%%;
  Step 4: Build a topology-based, scale and rotation invariant descriptor for each particle;
  Step 5: Match particle features and calculate *MatchRatio*;
  %%%%% **Subproblem 2**: fix $\mathbf{u}^{k+1}$ and solve $\hat{\mathbf{u}}^{k+1}$ globally %%%%%;
  Step 6: Solve Eq (B.3);
  Step 7: (Optional) Remove ghost particles using Eq (1);
  Step 8: Update dual variable: $\boldsymbol{\theta}^{k+1} \leftarrow \boldsymbol{\theta}^{k} + \hat{\mathbf{u}}^{k+1} - \mathbf{u}^{k+1}$;
  if *MatchRatio* $== 1$ **then**
    $M \leftarrow M + 1$ (M: count of "*MatchRatio* $== 1$")
  end
end
Output: Deformation displacement fields from image f_n to f_{n+t}: locally solved displacement $\mathbf{u}$ and final global, kinematically compatible $\hat{\mathbf{u}}$

where P_n stores all the coordinates of the detected particles in frame n; $\hat{\mathbf{u}}$ is the current, solved displacement field after the ADMM global step; $P_{n+t}(\hat{\mathbf{u}})$ represents all the coordinates of the detected particles in frame $n+t$, which are further warped with displacement $(-\hat{\mathbf{u}})$; ε_d is a user specified critical distance for a detected particle in one frame (i.e., frame n) to be diagnosed as a "ghost particle" if there does not exist any particle in the other comparing frame (frame $n+t$) within ε_d. All the detected ghost particles will be removed and remaining particles will form the updated particle collections $\mathbb{P}_n$ and $\mathbb{P}_{n+t}$ for frame n and frame $n+t$, respectively.

2.3.3. Tracking cumulative displacements

Two strategies are provided by the SerialTrack algorithm to reconstruct total, cumulative displacements at each step in an image sequence — namely both cumulative and incremental modes of tracking. In the cumulative mode, subsequent frames are always compared to the undeformed reference (first) frame, through which cumulative displacement fields can be obtained directly. For large deformations, we employ the tracked results from the previous frames and leverage a machine learning method to estimate a displacement predictor to further improve the tracking accuracy within subsequent frames [57]. In the incremental mode, each frame is compared to its subsequent frame. The tracked incremental displacement trajectory segments from each image pair in the sequence can then be merged to compute the final cumulative displacements at each time step (see Section 2.4 for more details).

2.3.4. Effect of particle shape distortion

We optionally consider the effect of particle shape distortion, e.g., for painted circular speckle dots on a 2D sample surface that deform into ellipses during a uniaxial compression/tension test, which can degrade the particle detection ability and thus decrease the overall tracking accuracy. Distinguishing it from Algorithm 1 where particles are defined as "hard", i.e., their shape is assumed to be non-deforming or rigid, we call these particles "soft" and assume their shape deformations follow their local, underlying material deformation gradients. A modified algorithm to better track these distorted particles is summarized in Algorithm 2. Different from tracking "hard" particles, in the "soft" particle tracking algorithm, particle centroid locations need to be re-detected using the actual, warped images during the ADMM iterations (see Step 4 in Algorithm 2).

2.4. Software functionalities: Post-processing

After all particles have been tracked, we provide post-processing functions to interpolate the tracked displacement fields onto a regularly spaced mesh to calculate deformation gradients and strain fields [4]. In the incremental tracking mode, the direct, tracked displacement field for each frame is in its current deformed configuration, or in an Eulerian coordinate frame [58]. Alternatively, we also include post-processing to determine individual particle trajectories throughout the image sequence in a Lagrangian coordinate frame. Particles are linked and all the displacement trajectories are merged [17,40] to obtain total particle displacements at each frame for each identified particle. To further improve the cumulative tracking ratios, all the tracked incremental displacement trajectory segments are extrapolated both *before* the segment's starting time point and *after* the segment's ending time point. Then we find and join trajectory segments at corresponding time points that are likely from the same particles, which can be merged together. We perform this "extrapolation and finding" scheme three to five times or until we have successfully merged all trajectory segments.

Algorithm 2: Outline of the "soft" particle tracking procedure

Input: Two consecutive images f_n and f_{n+t}
Step 1: Detect particles in reference image f_n;
Step 2: (Optional) Predict a displacement predictor;
Step 3: Set dual variable $\boldsymbol{\theta}$ to be zero. Set $IterNum = 1$, and $M = 0$;
while $\left|\hat{\mathbf{u}}^{k+1} - \hat{\mathbf{u}}^{k}\right| > \varepsilon$ ***and*** $IterNum < IterMax$ ***and*** $M < 5$ **do**
  Step 4: Warp image f_{n+t} with current displacement $\hat{\mathbf{u}}^k$ and detect particles in images f_{n+t};
  %%%%% **Subproblem 1**: fix $\hat{\mathbf{u}}^k$ and solve $\mathbf{u}^{k+1}$ locally %%%%%;
  Step 5: Build a topology-based, scale and rotation invariant feature for each particle;
  Step 6: Match particle features and calculate *MatchRatio*;
  %%%%% **Subproblem 2**: fix $\mathbf{u}^{k+1}$ and solve $\hat{\mathbf{u}}^{k+1}$ globally %%%%%;
  Step 7: Solve Eq. (B.3) for $\hat{\mathbf{u}}^{k+1}$;
  Step 8: (Optional) Remove ghost particles using Eq. (1);
  Step 9: Update dual variable: $\boldsymbol{\theta}^{k+1} \leftarrow \boldsymbol{\theta}^{k} + \hat{\mathbf{u}}^{k+1} - \mathbf{u}^{k+1}$;
  if *MatchRatio* $== 1$ **then**
    $M \leftarrow M + 1$ (M: count of "*MatchRatio* $== 1$")
  end
end
Output: Deformation displacement fields from image f_n to f_{n+t}: locally solved displacement $\mathbf{u}$ and final global, kinematically compatible $\hat{\mathbf{u}}$

3. Illustrative examples

We assess the SerialTrack method with both synthetic and experimental data sets at various particle seeding densities, as shown in Figs. 3–6. Although the synthetic image generation model may not capture all noise and error sources present in experimental images, it serves as a verification and validation for the algorithm where direct, quantitative error measurements can be made between tracking results and ground truth data. Several experimental test cases demonstrate the applicability of this technique for permutations of seeding density and dimensionality, and for both fluid and solid materials. All the used code parameters are summarized in Table 3. A permanent copy of the datasets for the examples can be found on the MINDS@UW open access institutional data repository.[1]

3.1. Synthetic examples

For both 2D and 3D synthetic test cases, we generate synthetic images with bead patterns following the steps described in Appendix C, and their quantitative evaluation results in Fig. 3 are shown using tracking ratios and root-mean-squared displacement error. As a baseline, we applied a small first-order motion field in the form of rigid body translation in the x-direction from 0 pixels to 4 pixels in 0.1 pixel increments using cumulative tracking. The typical tracking ratios in the 2D and 3D rigid translation are above 95% of particles tracked and 85% of particles tracked, respectively. Displacement root mean square (RMS) errors were computed on the measured centroid locations of each Lagrangian-tracked particle on the order of 10^{-2} pixels. This error level is comparable to the particle localization uncertainty [47,48]. The deformation gradient tensor, $\mathbf{F}$, of any homogeneous deformation follows the multiplicative decomposition $\mathbf{F} = \mathbf{R} \cdot \mathbf{U} = \mathbf{V} \cdot \mathbf{R}$ where $\mathbf{U}$ and $\mathbf{V}$ are the right and left stretch tensors and $\mathbf{R}$ is a pure rotation of the polar decomposition. Therefore, we include test cases for large, rigid body rotations (rotation angle θ from 0° to 180° in increments of 10°), finite uniaxial stretches (stretch ratios λ from 1 to 3 in increments of 0.1 and simple shear (shear angle γ

[1] https://minds.wisconsin.edu/handle/1793/82901

Table 3
SerialTrack code parameters for synthetic ("syn") and experimental ("exp") test cases.

Illustrative examples	Syn or Exp	Fig.	Dim.	Particle rigidity	Tracking mode	Bead intensity threshold	Bead radius (px)	Max neighbor #	Size of search field
Translation	Syn	3a	2D	Hard	inc	0.5	3	25	Inf
Rotation	Syn	3b	2D	Hard	inc	0.5	3	25	Inf
Uniaxial stretch	Syn	3c	2D	Hard	accum	0.5	3	25	50
Simple shear	Syn	3d	2D	Hard	accum	0.5	3	25	50
Translation	Syn	3e	3D	Hard	inc	0.5	3	25	Inf
Rotation	Syn	3f	3D	Hard	inc	0.5	3	25	Inf
Uniaxial stretch	Syn	3g	3D	Hard	accum	0.5	3	25	50
Simple shear	Syn	3h	3D	Hard	accum	0.5	3	25	50
DIC Challenge v2	Syn	3i-j	2D	Hard	inc	0.5	3	25	50
Uniaxial stretch	Syn	S3c	2D	Soft	accum	0.5	3	25	50
Simple shear	Syn	S3d	2D	Soft	accum	0.5	3	25	50
Inertial cavitation	Exp	4a	2D	Hard	inc	0.5	2	10	30
Pipe flow	Exp	4b	2D	Hard	inc	0.4	2	25	50
Foam compression	Exp	5	2D	Soft	accum	0.5	3	25	50
Hydrogel shear	Exp	6a	3D	Hard	inc	0.1	20	5	700
Gel indentation	Exp	6b	3D	Hard	inc	0.1	3	25	50

from 0° to 45° in $\tan(\gamma)$ increments of 0.05), as shown in Fig. 3(b–d) for 2D cases and (f–h) for 3D cases. Particle seeding densities (SD) range from 0.003 particles per pixel to 0.012 particles per pixel in the synthetic 2D cases, and 10^{-4} particles per voxel to 10^{-3} particles per voxel in the synthetic 3D cases. PTV methods based on an underlying rectilinear grid (typical for image-based measurements) are often challenged by large rotation angles. For large stretches and shear deformations where motions are greater than the inter-particle spacing, local algorithms, such as kNN or relaxation methods, suffer from non-uniqueness of particle identification leading to poor tracking ratios [27]. In the SerialTrack method, the region-based formulation is designed to minimize these effects and is hybridized with a global optimization strategy that ensures uniqueness and kinematic admissibility of the reconstructed motion field, and thus the tracking ratios remain high and the RMS displacement errors are typically below $O(0.1)$ px. In all cases, the tracking ratio decrease in part reflects that particles are moving out of the field of view in our referential Eulerian frame. This is most clearly illustrated in the rotational case, where the lowest tracking ratios correspond to approximately 45° rotation, where the overlap between reference and deformed configuration frame is likewise at a minimum. For a summary of the overall detection ratio and strain RMS error, see SI Fig. S1 and Fig. S2 for the synthetic 2D and 3D results, respectively. Our typical particle tracking applications use rigid particles, however, deformable or modulus-matched particles can be important to preserve the verisimilitude of the instrumented test. The method, therefore, also optionally accounts for particle shape distortions. A similar summary of the tracking results for this subclass of reconstructions is given in SI Fig. S3.

Inspired by Reu et al. [59], we designed a synthetic 2D example to test the spatial resolution of the SerialTrack algorithm using a "star" pattern displacement field. Both reference and deformed images are 4001 pixels × 501 pixels. The vertical displacement has a varying spatial period, λ, from 10 pixels at $x = 1$ pixels, to 300 pixels at $x = 4001$ pixels according to Eq (2). The spatial period of the vertical sine wave in the assigned displacement field is proportional to the image position across the width of the image. The magnitude of the periodic vertical displacements is ± 2 pixels. The ground truth of horizontal displacement is zero. We test different bead seeding densities (SD) of (0.003, 0.006, and 0.012) beads per pixel. No additional noise is added during the image generation.

$$\lambda = 10 + \frac{300 - 10}{4001}(x - 1) \tag{2}$$

Both SerialTrack tracked horizontal and vertical displacement fields are summarized in Fig. 3(i). Vertical displacements retrieved along the center row ($y = 251$ pixels) are further summarized in Fig. 3(j). The ground truth of the vertical displacement at $y = 251$ pixels is 2 pixels. First, we find that SerialTrack can resolve heterogeneous deformations well. Particularly, using dense particles where SD > 0.006 particles per pixel, the algorithm can accurately recover the low frequency, large amplitude deformation fields when $x > 500$ pixels, where the wave length is greater than 46 pixels. We also find that the tracked vertical displacement for the very high frequency deformations on the left hand side are underestimated. This is not surprising, because with a topology-based particle tracking approach it is challenging to resolve heterogeneous deformation whose characteristic wavelength (λ) is smaller than half of the averaged nearest-neighbor-particle distance $\sim O(\mathrm{SD}^{-1/2})$. Compared with other subset-based correlation methods [59], the SerialTrack method is not only able to solve dense particles but also sparse particles. Additionally, it can be computationally cheaper, with a potential computational cost reduction on the order of (# of image pixels) / (# of particles).

3.2. Experimental examples

With the synthetic deformation cases showing strong performance across a range of deformation modes and particle densities in both 2D and 3D, we move to a variety of experimental test cases. Test cases are conducted in both 2D and 3D, with sparse and dense particles, and are summarized in Figs. 4–6. Additionally, a 2D large deformation uniaxial compression experiment is employed to demonstrate the "soft particle" implementation in Section 3.2.2 where the ink-printed circular dots undergo significant shape distortion.

3.2.1. 2D "hard" particle examples

Fig. 4a shows an example of sparse 2D particle tracking in a laser-induced cavitation event in a soft material specimen. Following McGhee et al. [60], 15 μm polystyrene particles are embedded into a single, flat plane within the bulk of a gelatin hydrogel. Then the sample is placed into the optical path of a laser-based cavitation system [61,62]. A 6 ns laser pulse is steered through the backport of an inverted microscope and focused via a 20× magnification, 0.5 numerical aperture (NA) (i.e., 20×/0.5NA) imaging objective onto the imaging plane to induce a cavitation bubble on the same plane as the embedded particles. Fig. 4a(i) shows the resulting bubble radius vs. time curve with call-outs for specific frames of interest denoted by a blue star, red diamond, and yellow circle which correspond to the time points of bubble expansion, maximum bubble radius, and nearing the first collapse, respectively. Two images of a typical cavitation event are shown in Fig. 4a(ii). By tracking the 2D motion of the embedded sparsely distributed particles in this image sequence,

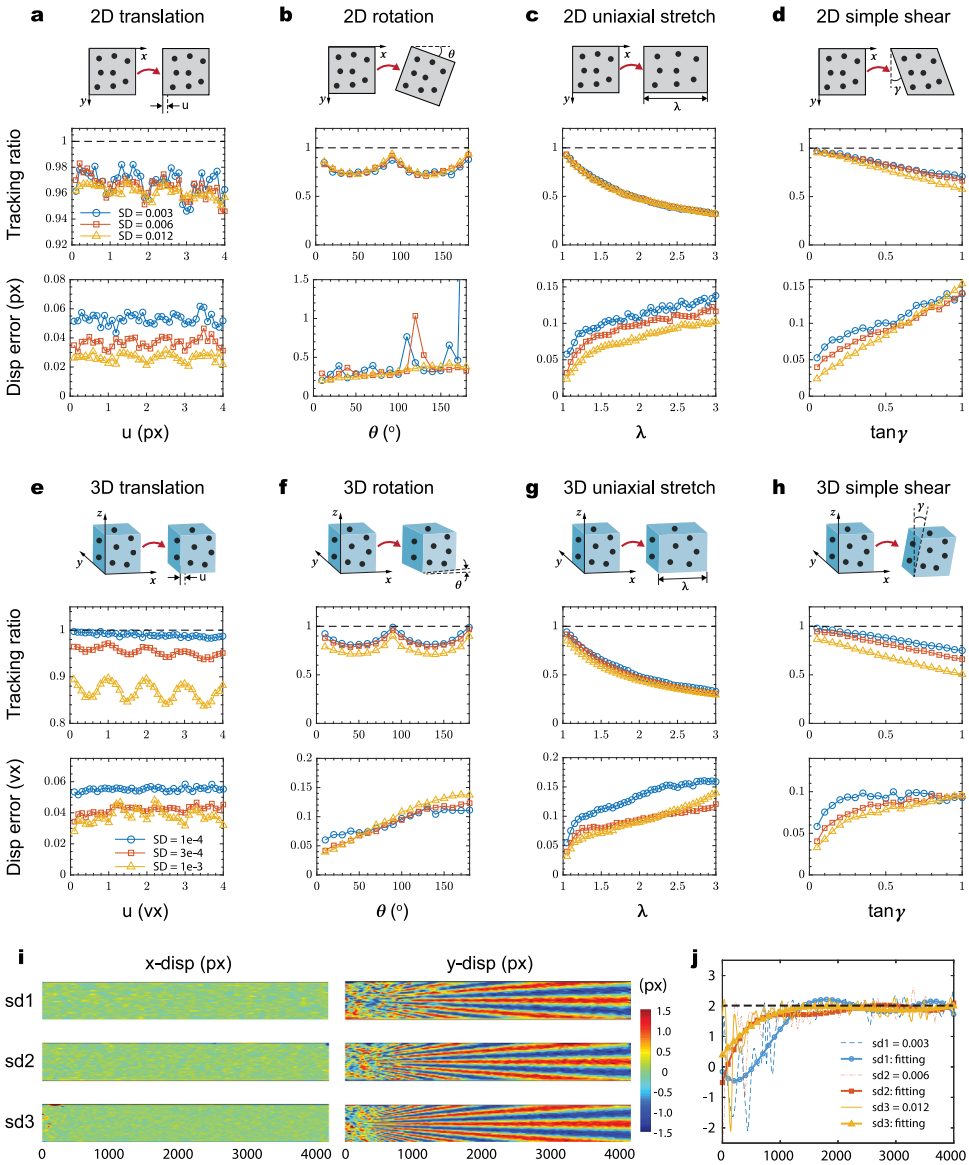

Fig. 3. The overall ratio of successfully tracked particles to total detected particles as a function of increasing applied motion, i.e., image step, and the displacement RMS error at each step for synthetic 2D (top row) & 3D (middle row) homogeneous deformations for the proposed SerialTrack method. (SD: seeding density). A synthetic 2D example using a "star" pattern heterogeneous displacement field is shown at the bottom row. (a,e) Unidirectional translations in the *x*-direction from 0 pixels (or voxels) to 4 pixels (or voxels). (b,f) Rigid body rotation about the *z*-axis with rotation angles from 0° to 180°. (c,g) Uniaxial stretch in the *x*-direction with stretch ratios from 1 to 3. (d,h) Simple *xy*- and *xz*-shear with shear angles from 0° to 45°. (i) Tracked *x* and *y* displacement fields in the synthetic "star" pattern heterogeneous field and their vertical displacements retrieved along the center row are further summarized in (j).

we can reconstruct the evolution of the resulting time-resolved velocity fields. For example, the radial velocity vs. radial distance curves are computed and plotted in Fig. 4a(iii). Velocity fields at the marked expansion (blue star) and collapse (yellow circle) time points in Fig. 4a(i) are summarized in Fig. 4a(iv).

As an example of a densely populated, 2D particle tracking scenario, we examine a case where the data originates from high-speed PIV measurements of flow in a tube [40] as shown in Fig. 4b(i). We use the same Laplacian of Gaussian image filtering technique as described [40] to detect single particles (see Fig. 4b:ii–iii). We tested both the incremental and cumulative modes (see Fig. 4(b:v–vii)). In the cumulative mode, we directly track the total, cumulative displacement of each individual particle. In the incremental mode, the cumulative displacements are computed by merging trajectory segments (refer to Section 2.4). The final cumulative tracking ratio is given in each case in Fig. 4(b:iv). The reconstructed cumulative displacement fields for the first and ninth frames are visualized in Fig. 4(b:viii–ix), where the cumulative displacement in the ninth frame is large but still well-tracked by SerialTrack.

3.2.2. 2D "soft" particle tracking example

Here we apply our SerialTrack code to a large deformation, uniaxial compression experiment on an open-cell polyurethane foam sample with a nominal density of 240 kg/m^3. The dimensions of the foam specimen were approximately 12.7 mm × 12.7 mm × 12.7 mm. The experimental setup and other experimental details can be found in Yang et al. [63]. The reference and deformed images at compression ratios of 7.3%, 16.6%, 25.9%, and 37.4% are shown in Fig. 5a, with magnified insets shown in Fig. 5b. Three dashed-line ellipses and three rectangles are marked to highlight the same locations on the front surface of the testing specimen that underwent large deformations.

As described in Algorithm 2, deformed images are iteratively warped, and single particles are detected during each ADMM

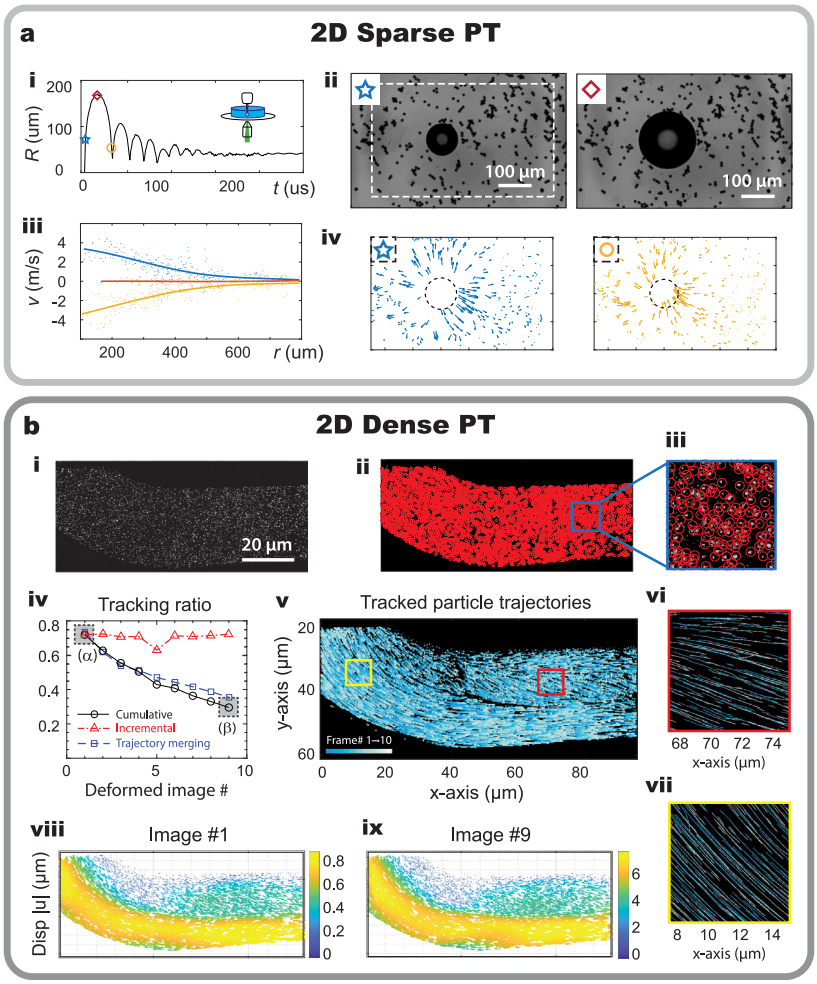

Fig. 4. 2D, "hard" particle tracking experimental examples. (a) Tracking laser-induced inertial cavitation in a soft material with sparsely seeded particles [60]. (a:i) Experimentally measured bubble radius vs. time curve. Three specific frames of interest are denoted by a blue star, red diamond, and yellow circle, which corresponds to the time points of bubble expansion, maximum bubble radius, and nearing the first collapse point, respectively. (a:ii) Raw high-speed camera frames at the blue star and red diamond time points. (a:iii) Reconstructed radial velocity vs. radial distance curves at the three marked time points in (a:i). (a:iv) Reconstructed velocity fields at the blue star and yellow circle time points. (b) 2D particle tracking example of flow through a bent pipe using densely seeded particles [40]. (b:i) One typical frame in the time-resolved image sequence. (b:ii-iii) Detected single particle centroids using the Laplacian of Gaussian filtering technique are circled in red. (b:iv) Particle tracking ratios. (b:v-vii) Tracked trajectories. (b:viii-ix) Tracked cumulative displacement fields for the first and ninth frames.

iteration. Here we present the final warped images in Fig. 5c. The corresponding detected particle centroids are marked by red dots and shown under magnification in Fig. 5d. The final tracked cumulative displacements are visualized via cone plots and summarized in Fig. 5e.

3.2.3. 3D examples

As a sparsely seeded, 3D tracking example case, inspired by biological applications such as traumatic brain injury [64,65], we seeded 5 µm fluorescent microparticles at a 1.5% vol/vol fraction in a soft polyacrylamide hydrogel and deformed the hydrogel in a simple-shear-like mode on a confocal laser point scanning microscope using a 20×/0.5NA (approximately 1 µm voxel size) imaging objective. The shear deformation was imposed quasi-statically (1 minute per step) in 10 steps in nominally 4% engineering shear strain increments from 0% strain to 40% strain, and a total of $O(100)$ particles were tracked as shown in Fig. 6a(i) where the artificial color of the particles depends on the z-coordinate. Results, including the particle tracking ratios, nominal crosshead, and reconstructed deformation gradient tensor components, are shown in Fig. 6a(ii,iii). In addition, note that the final cumulative tracking ratio obtained by merging incrementally tracked trajectory segments is higher than the direct cumulative mode (see Fig. 6a(ii)).

We also test our SerialTrack method for tracking densely seeded particles in 3D. In this experiment, a 1 mm diameter stainless steel sphere with a density of 7750 kg/m^3 was placed onto the surface of a submerged soft polyacrylamide (PA) hydrogel to perform spherical indentation under the force of gravity (g), as shown in Fig. 6b(i-ii). 3D volumetric image stacks (image size: 1024 voxels × 1024 voxels × 445 voxels) containing fluorescent beads were scanned before and after the indentation deformation near the hydrogel surface using multiphoton microscopy and a 25×/1.15NA water immersion objective [56]. All other experimental parameters can be found in [56]. Three-dimensional cone plot and the xz-plane projection of the tracked 3D deformation field are shown in Fig. 6b(iii) and (iv), respectively.

4. Impact and conclusions

Particle tracking, often called single particle tracking (SPT) or particle tracking velocimetry (PTV), is a widely used methodology for quantitative, full-field analysis of dynamic processes, typically from time-lapse image data. SPT methods operate by detecting and tracking individual tracer particles or fiducial markers during

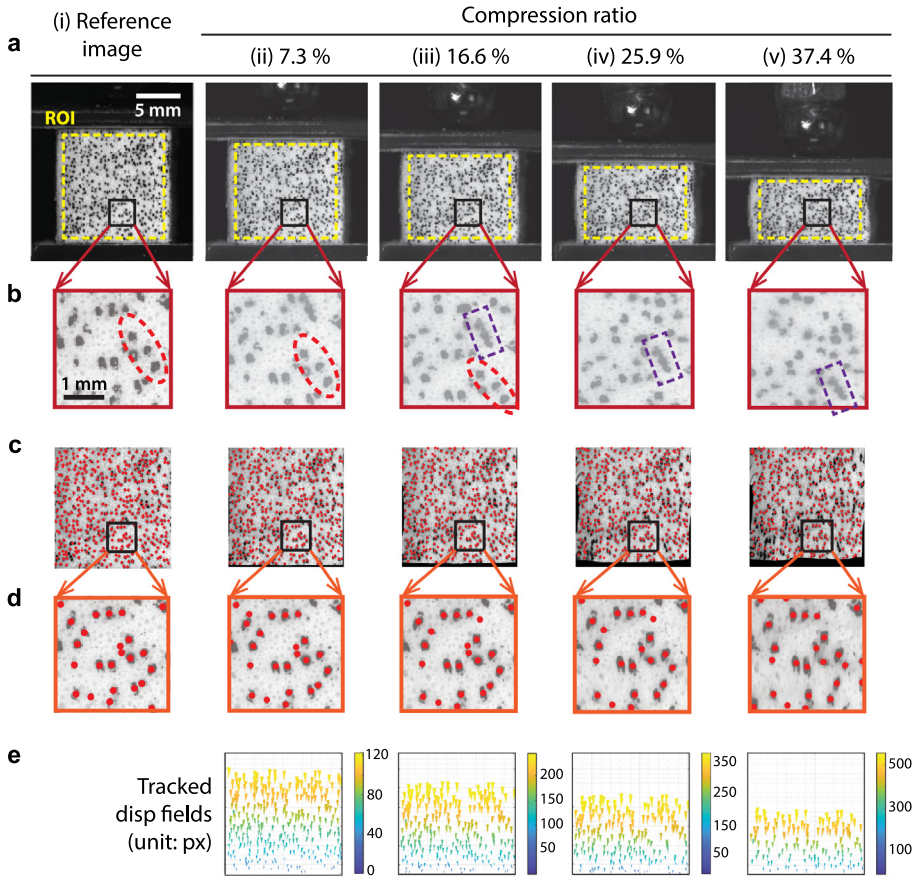

Fig. 5. Demonstration of the "soft" particle tracking (Algorithm 2) via a large deformation uniaxial compression experiment. (a) Experimental reference and selected deformed images where regions of interest (ROIs) are marked using dashed line rectangular boxes. Insets are magnified and shown in (b) where three ellipses and three rectangles are marked to highlight the same locations on the surface of the specimen that underwent large deformations, respectively. During the "soft" particle tracking process, deformed images are iteratively warped, and particles are detected as shown in (c-d). Final tracked cumulative displacement field cone plots are summarized in (e).

a time-resolved image sequence. Here we present a new hybrid local–global tracking algorithm that builds an iterative scale and rotation invariant topology-based feature vector for each particle within a multi-scale tracking process, where the global kinematic compatibility of the final tracked displacement field is optimized. SerialTrack is able to track particles in both 2D and 3D images with both sparse and dense particle seeding densities. It can accurately reconstruct large, finite deformation and complex velocity fields. We also consider the effect of shape distortion on the embedded particles due to presence of local, material deformation gradients. We used synthetic examples to verify and validate the implementation of SerialTrack, and provide an estimate of the spatial resolution capabilities. We then demonstrate the performance and post-processing routines on a variety of experimental test cases including 2D and 3D examples featuring sparsely and densely populated, soft and hard particles. Of final note, the current implementation of SerialTrack assumes the overall deformation field to be continuous (see Eq. (A.4)). However, the proposed optimization problem can be further modified to consider discontinuous deformations [57], which will be added to future versions of the code. The provided open-source code package implements the proposed SerialTrack method and allows users to apply the code directly to their own research.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors thank the U.S. Office of Naval Research for research support under the "PANTHER" program, award number N000142112044 through Dr. Timothy Bentley. This research was performed in part while A.K.L. held an NRC Research Associateship award at the National Institute of Standards and Technology.

Appendix A. Particle tracking problem formulation

We assume that each image is in a sequence of images from the 0th (reference configuration) to Nth (final, fully deformed) configuration, where each image is defined by a grayscale intensity field $f_n(\mathbf{x})$, consisting of multiple superposed intensity sources ($\mathcal{N}$, e.g., from fluorescent particles) in the image domain as:

$$f_n(\mathbf{x}) = \sum_{\mathbf{P}\in\mathbb{P}_n} \mathcal{N}\left(\mathbf{x}; A(\mathbf{P})\exp\left(-\frac{|\mathbf{x}-\mathbf{P}|^2}{2\sigma(\mathbf{P})^2}\right)\mathbf{I}\right), \tag{A.1}$$

where $\mathbf{x}$ denotes each image pixel; $\mathbf{P}$ is the coordinate of each individual particle centroid; A is the maximum particle intensity with $\sigma(\mathbf{P})$ the standard deviation of intensity decay for each particle; $\mathbf{I}$ is an identity matrix. $\mathbb{P}_n$ is the collection of all particles in image f_n, $C(n)$ is the total number of particles in image f_n, and

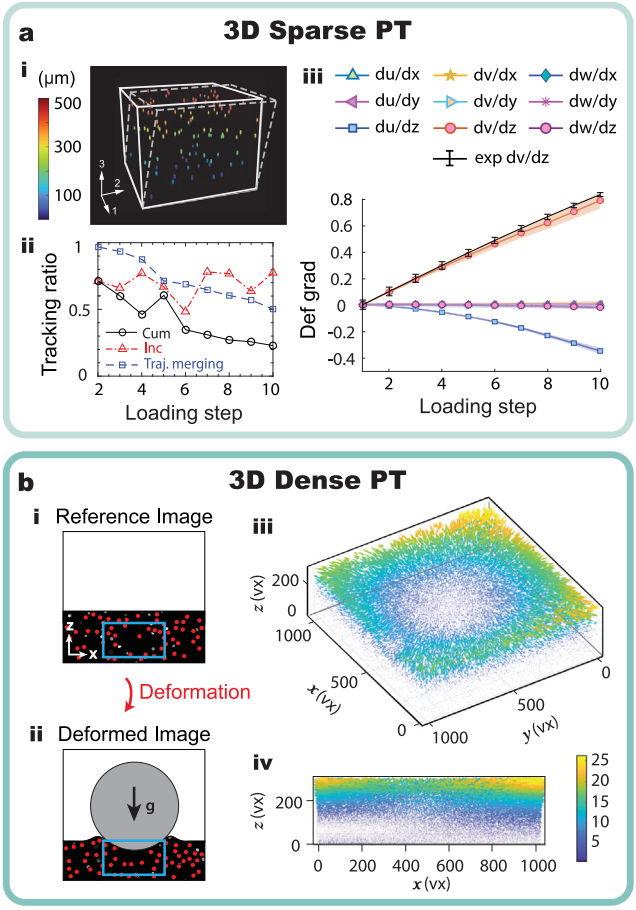

Fig. 6. Tracking 3D deformations. (a) A 3D shear example (du/dz $\neq 0$, dv/dz $\neq 0$) with sparsely distributed particles. (a:i) The shear deformation was imposed quasi-statically on the hydrogel sample. All detected particles are color-coded depending on the z-coordinate of their centroids. (a:ii) Tracking ratios. (a:iii) Reconstructed deformation gradients, which agree well with the nominal crosshead motion (coded as "exp"). (b) A 3D spherical indentation example where particles are densely seeded. (b:i–ii) Sketch of the reference and deformed configurations of the spherical indentation experiment (not to scale). (b:iii–iv) 3D cone plot and the xz-plane projection of the tracked 3D deformation.

we define the operator $\mathcal{E}$ as the *particle detection* process that extracts the centroid position ($\mathbf{P}_i$) of each particle such that

$$\mathbb{P}_n = \left\{\mathbf{P}_0, \mathbf{P}_1, \ldots, \mathbf{P}_{C(n)}\right\} := \mathcal{E}(f_n(\mathbf{x})) \tag{A.2}$$

The subsequent *particle linking* process solves for the unknown displacement field $\mathbf{u}_i$ from image n in the sequence to a later image $n+t$ in the sequence. For *incremental* mode we set $t \triangleq 1$, and for *cumulative* mode $n \triangleq 0$ and $t \in N$, where t is in general a positive integer number of image frames. The linking process solution minimizes a cost function, for example, in a simple case the sum of squared differences,

$$\min_{\mathbf{u}_i} \int_{\mathbf{x}} |\mathcal{E}(f_n(\mathbf{x})) - \mathcal{E}(f_{n+t}(\mathbf{x}-\mathbf{u}_i))|^2 \, d\mathbf{x} \tag{A.3}$$

However, the above optimization problem is ill-posed. In general, the cost function is not convex and the solution is not unique, since the matching of particles $\mathbb{P}_n$ to $\mathbb{P}_{n+t}$ has no guarantee of uniqueness and in practices mismatches and non-matching particle links frequently exist. To reduce displacement noise introduced by mis-linked particles, global regularization penalties are further added to the optimization cost function. This optimization problem can be efficiently solved by the *alternating direction method of multipliers* (ADMM) after adding a global slack variable, $\hat{\mathbf{u}}_i$. The modified optimization problem is:

$$\min_{\mathbf{u}_i} \int_{\mathbf{x}} |\mathcal{E}(f_n(\mathbf{x})) - \mathcal{E}(f_{n+t}(\mathbf{x}-\mathbf{u}_i))|^2 + \frac{\alpha}{2}\left|\nabla\hat{\mathbf{u}}_i\right|_F^2 \, d\mathbf{x}, \tag{A.4}$$

subject to $\mathbf{u}_i = \hat{\mathbf{u}}_i$

where the coefficient α is a positive weight of the added regularizer and $|\cdot|_F$ is the Frobenius norm for tensors such that $|\mathbf{A}|_F^2 := \sum_i \sum_j |A_{ij}|^2$. The first term in (A.4) is the displacement from particle matches using the *local* linking algorithm information, which can be solved quickly and in parallel as discussed in Sect 2.3.1 for our implementation. The second term in Eq (A.4) penalizes *global* displacement variance and noise. Other global regularization schemes can also be applied if there is additional, *a priori* known information about the physics of the problem. The combined local and global optimization solution is implemented in an iterative fashion, such that local particle matching is informed by the globally refined displacement field to yield a final unique and kinematically admissible displacement field $\mathbf{u}_i$ with local accuracy and resolution from individually tracked centroid locations.

In the current algorithm, we only consider continuous deformation cases where certain regularity and smoothness are assumed as prior information (see Eq (A.4)). However, the proposed optimization problem can be further modified to consider discontinuous deformations, e.g., [57], or by enforcing alternative partial differential equations (e.g., a Naiver–Stokes-based global form), which is a potential future work direction

Appendix B. Alternating direction method of multipliers

To efficiently solve the optimization problem posed in Eq. (A.4), it can be rewritten for a given displacement step $\mathbf{u}$ in ADMM form as [66]:

$$\mathcal{L}(\mathbf{u}, \hat{\mathbf{u}}, \boldsymbol{\theta}) = \int_{\mathbf{x}} |\mathcal{E}(f_n(\mathbf{x})) - \mathcal{E}(f_{n+t}(\mathbf{x}-\mathbf{u}))|^2 + \frac{\alpha}{2}\left|\nabla\hat{\mathbf{u}}\right|^2 + \frac{\mu}{2}\left|\hat{\mathbf{u}} - \mathbf{u} + \boldsymbol{\theta}\right|^2 \, d\mathbf{x} \tag{B.1}$$

where μ is a positive coefficient of the added augmented Lagrangian penalty and $\boldsymbol{\theta}$ is an introduced dual variable. During each ADMM iteration step, we first decompose the global minimization problem Eq. (B.1) into independent, local problems (see Sect 2.3.1), then all the local solutions, $\mathbf{u}$, are projected onto a global, kinematically compatible space where the global auxiliary displacement field $\hat{\mathbf{u}}$ is admissible. Mathematically, given the results $\{\mathbf{u}_i^k\}$, $\{\hat{\mathbf{u}}_i^k\}$, $\{\boldsymbol{\theta}_i^k\}$ in the kth step, we solve the $(k+1)$th update using the following steps:

- **Subproblem 1**: Local update. While holding $\{\hat{\mathbf{u}}_i^k\}$ and $\{\boldsymbol{\theta}_i^k\}$ fixed, we minimize Eq. (B.1) over $\{\mathbf{u}_i\}$ to obtain $\{\mathbf{u}_i^{k+1}\}$. Since $\{\hat{\mathbf{u}}\}$ is fixed and μ can be a small value, this problem is broken down into a series of local problems that can be solved independently using local topology-based feature matching to obtain a displacement guess:

$$\mathbf{u}^{k+1} := \arg\min_{\mathbf{u}} \mathcal{L}(\mathbf{u}, \hat{\mathbf{u}}^k, \boldsymbol{\theta}^k) \tag{B.2}$$

- **Subproblem 2**: Global update. While holding $\{\mathbf{u}_i^k\}$ and $\{\boldsymbol{\theta}_i^k\}$ fixed, we minimize $\mathcal{L}$ over $\{\hat{\mathbf{u}}\}$ such that

$$\hat{\mathbf{u}}^{k+1} := \arg\min_{\hat{\mathbf{u}}} \mathcal{L}(\mathbf{u}^{k+1}, \hat{\mathbf{u}}, \boldsymbol{\theta}^k) \tag{B.3}$$

This is a global problem, but is independent of the original image sequence $\mathbf{f}$ since it only relies on the displacements computed from the local particle linking step. Indeed, it leads to a

well-posed linear problem:

$$\left(-\frac{\alpha}{\mu}\nabla\cdot\nabla+\mathbf{I}\right)\hat{\mathbf{u}}=\mathbf{u}^{k+1}-\boldsymbol{\theta}^{k} \tag{B.4}$$

- **Subproblem 3**: Dual variable update. We finally update the dual variable $\{\boldsymbol{\theta}\}$ as follows:

$$\boldsymbol{\theta}^{k+1}:=\boldsymbol{\theta}^{k}+\hat{\mathbf{u}}^{k+1}-\mathbf{u}^{k+1}. \tag{B.5}$$

In practice, the smoothing parameter α is carefully chosen in the range $\alpha/\mu = O(10^{-3}) \sim O(10^{-1})$ based on the expected smoothness of the deformation field. The parameter α/μ can be further tuned using the L-curve method [67].

Appendix C. Synthetic image generation

In each reference image, isolated spherical beads are randomly seeded using a 2D or 3D Gaussian intensity profile as an approximation of a random, isotropic image pattern. A typical Gaussian point spread function (PSF) with amplitude A and spread σ and located at $\mathbf{x}$ is expressed as

$$\mathrm{PSF}(\mathbf{x})=A\exp\left(-\sum_{i=1}^{d}\frac{x_i^2}{2\sigma^2}\right)\mathbf{I} \tag{C.1}$$

where d is the image dimensionality and x_i is the ith-component of $\mathbf{x}$; $\mathbf{I}$ is the identity tensor. We choose $\sigma = 1$ to approximate a circular/spherical particle in the volume image with a diameter of approximately 5 pixels or voxels. All the beads are sampled randomly with seeding density SD which denotes the number of particles per pixel or voxel. To avoid particles overlapping in the synthetic images, a Poisson disc sampling algorithm is used to seed center-point locations with a minimum separation distance between particles equal to the particle diameter [27]. The particle positions in the deformed images are calculated via the imposed displacement field and grayscale values are interpolated into the image. In addition, 5% white Gaussian noise has been added to the synthetic images to roughly approximate the experimental noise in our images.

Appendix D. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.softx.2022.101204.

References

[1] Chenouard N, Smal I, De Chaumont F, Maška M, Sbalzarini IF, Gong Y, et al. Objective comparison of particle tracking methods. Nature Methods 2014;11(3):281–9. http://dx.doi.org/10.1038/nmeth.2808.

[2] Maas HG, Gruen A, Papantoniou D. Particle tracking velocimetry in three-dimensional flows. Exp Fluids 1993;15:133–46. http://dx.doi.org/10.1007/BF00190953.

[3] Meijering E, Dzyubachyk O, Smal I. Chapter nine - Methods for cell and particle tracking. In: conn PM, editor. Methods in enzymology. Imaging and spectroscopic analysis of living cells, vol. 504, Academic Press; 2012, p. 183–200. http://dx.doi.org/10.1016/B978-0-12-391857-4.00009-4.

[4] Hazlett L, Landauer AK, Patel M, Witt HA, Yang J, Reichner JS, et al. Epifluorescence-based three-dimensional traction force microscopy. Sci Rep 2020;10:16599. http://dx.doi.org/10.1038/s41598-020-72931-6.

[5] Ulman V, Maška M, Magnusson KE, Ronneberger O, Haubold C, Harder N, et al. An objective comparison of cell-tracking algorithms. Nature Methods 2017;14(12):1141–52. http://dx.doi.org/10.1038/nmeth.4473.

[6] Ewers H, Smith AE, Sbalzarini IF, Lilie H, Koumoutsakos P, Helenius A. Single-particle tracking of murine polyoma virus-like particles on live cells and artificial membranes. Proc Natl Acad Sci 2005;102:15110–5. http://dx.doi.org/10.1073/pnas.0504407102.

[7] Sbalzarini IF, Koumoutsakos P. Feature point tracking and trajectory analysis for video imaging in cell biology. J Struct Biol 2005;151:182–95. http://dx.doi.org/10.1016/j.jsb.2005.06.002.

[8] Wu M, Roberts JW, Kim S, Koch DL, DeLisa MP. Collective bacterial dynamics revealed using a three-dimensional population-scale defocused particle tracking technique. Appl Environ Microbiol 2006;72:4987–94. http://dx.doi.org/10.1128/AEM.00158-06.

[9] Leggett SE, Patel M, Valentin TM, Gamboa L, Khoo AS, Williams EK, et al. Mechanophenotyping of 3D multicellular clusters using displacement arrays of rendered tractions. Proc Natl Acad Sci 2020;117:5655–63. http://dx.doi.org/10.1073/pnas.1918296117.

[10] Ardekani R, Biyani A, Dalton JE, Saltz JB, Arbeitman MN, Tower J, et al. Three-dimensional tracking and behaviour monitoring of multiple fruit flies. J R Soc Interface 2013;10:20120547. http://dx.doi.org/10.1098/rsif.2012.0547.

[11] Kim J-T, Kim D, Liberzon A, Chamorro LP. Three-dimensional particle tracking velocimetry for turbulence applications: Case of a jet flow. J Visual Exp: JoVE 2016;53745. http://dx.doi.org/10.3791/53745.

[12] Saha D, Babler MU, Holzner M, Soos M, Lüthi B, Liberzon A, et al. Breakup of finite-size colloidal aggregates in turbulent flow investigated by three-dimensional (3D) particle tracking velocimetry. Langmuir 2016;32:55–65. http://dx.doi.org/10.1021/acs.langmuir.5b03804.

[13] Crocker JC, Grier DG. Methods of digital video microscopy for colloidal studies. J Colloid Interface Sci 1996;179(1):298–310. http://dx.doi.org/10.1006/jcis.1996.0217.

[14] Mason TG, Weitz DA. Optical measurements of frequency-Dependent linear viscoelastic moduli of complex fluids. Phys Rev Lett 1995;74:1250–3. http://dx.doi.org/10.1103/PhysRevLett.74.1250.

[15] Manzo C, Garcia-Parajo MF. A review of progress in single particle tracking: From methods to biophysical insights. Rep Progr Phys 2015;78(12):124601. http://dx.doi.org/10.1088/0034-4885/78/12/124601.

[16] Huang PY, Kurasch S, Alden JS, Shekhawat A, Alemi AA, McEuen PL, et al. Imaging atomic rearrangements in two-dimensional silica glass: Watching silica's dance. Science 2013;342(6155):224–7. http://dx.doi.org/10.1126/science.1242248.

[17] Novara M, Scarano F. A particle-tracking approach for accurate material derivative measurements with tomographic PIV. Exp Fluids 2013;54:1584. http://dx.doi.org/10.1007/s00348-013-1584-5.

[18] Smith WN. A submersible three-dimensional particle tracking velocimetry system for flow visualization in the coastal ocean. Limnol Oceanogr: Methods 2008;6:96–104. http://dx.doi.org/10.4319/lom.2008.6.96.

[19] Pereira F, Stüer H, Graff EC, Gharib M. Two-frame 3D particle tracking. Meas Sci Technol 2006;17:1680–92. http://dx.doi.org/10.1088/0957-0233/17/7/006.

[20] Ouellette NT, Xu H, Bodenschatz E. A quantitative study of three-dimensional Lagrangian particle tracking algorithms. Exp Fluids 2006;40:301–13. http://dx.doi.org/10.1007/s00348-005-0068-7.

[21] Boltyanskiy R, W. Merrill J, R. Dufresne E. Tracking particles with large displacements using energy minimization. Soft Matter 2017;13:2201–6. http://dx.doi.org/10.1039/C6SM02011A.

[22] Legant WR, Miller JS, Blakely BL, Cohen DM, Genin GM, Chen CS. Measurement of mechanical tractions exerted by cells in three-dimensional matrices. Nature Methods 2010;7:969–71. http://dx.doi.org/10.1038/nmeth.1531.

[23] Schanz D, Gesemann S, Schröder A. Shake-the-box: Lagrangian particle tracking at high particle image densities. Exp Fluids 2016;57:70. http://dx.doi.org/10.1007/s00348-016-2157-1.

[24] Cierpka C, Lütke B, Kähler CJ. Higher order multi-frame particle tracking velocimetry. Exp Fluids 2013;54:1533. http://dx.doi.org/10.1007/s00348-013-1533-3.

[25] Feng X, Hall MS, Wu M, Hui C-Y. An adaptive algorithm for tracking 3D bead displacements: Application in biological experiments. Meas Sci Technol 2014;25:055701. http://dx.doi.org/10.1088/0957-0233/25/5/055701.

[26] Jaqaman K, Loerke D, Mettlen M, Kuwata H, Grinstein S, Schmid SL, et al. Robust single-particle tracking in live-cell time-lapse sequences. Nature Methods 2008;5:695–702. http://dx.doi.org/10.1038/nmeth.1237.

[27] Patel M, Leggett SE, Landauer AK, Wong IY, Franck C. Rapid, topology-based particle tracking for high-resolution measurements of large complex 3D motion fields. Sci Rep 2018;8(1):1–14. http://dx.doi.org/10.1038/s41598-018-23488-y.

[28] Ohmi K, Panday SP. Particle tracking velocimetry using the genetic algorithm. J Vis 2009;12:217–32. http://dx.doi.org/10.1007/BF03181860.

[29] Brevis W, Niño Y, Jirka G. Integrating cross-correlation and relaxation algorithms for particle tracking velocimetry. Exp Fluids 2011;50(1):135–47. http://dx.doi.org/10.1007/s00348-010-0907-z.

[30] PTVlab software link. 2022, http://ptvlab.blogspot.com. [Accessed 14 September 2022].

[31] OpenPTV software link. 2022, http://www.openptv.net. [Accessed 14 September 2022].

[32] Willneff J. A spatio-temporal matching algorithm for 3D particle tracking velocimetry (Ph.D. thesis), ETH Zurich; 2003.

[33] Ershov D, Phan M-S, Pylvänäinen JW, Rigaud SU, Le Blanc L, Charles-Orszag A, et al. Bringing TrackMate into the era of machine-learning and deep-learning. 2021, http://dx.doi.org/10.1101/2021.09.03.458852, BioRxiv, Cold Spring Harbor Laboratory.
[34] TrackMate software link. 2022, https://imagej.net/plugins/trackmate. [Accessed 14 September 2022].
[35] TrackMate software algorithms. 2022, https://imagej.net/plugins/trackmate/algorithms. [Accessed 14 September 2022].
[36] Heyman J. TracTrac: A fast multi-object tracking algorithm for motion estimation. Comput Geosci 2019;128:11–8. http://dx.doi.org/10.1016/j.cageo.2019.03.007.
[37] TracTrac software link. 2022, https://perso.univ-rennes1.fr/joris.heyman/trac.html. [Accessed 14 September 2022].
[38] TPT software link. 2022, https://github.com/FranckLab/T-PT. [Accessed 14 September 2022].
[39] Lejeune E, Khang A, Sansom J, Sacks MS. FM-Track: A fiducial marker tracking software for studying cell mechanics in a three-dimensional environment. SoftwareX 2020;11:100417. http://dx.doi.org/10.1016/j.softx.2020.100417.
[40] Janke T, Schwarze R, Bauer K. Part2Track: A MATLAB package for double frame and time resolved particle tracking velocimetry. SoftwareX 2020;11:100413. http://dx.doi.org/10.1016/j.softx.2020.100413.
[41] Zepeda O J, Bishop LDC, Dutta C, Sarkar-Banerjee S, Leung WW, Landes CF. Untying the gordian KNOT: Unbiased single particle tracking using point clouds and adaptive motion analysis. J Phys Chem A 2021;8723–33. http://dx.doi.org/10.1021/acs.jpca.1c06100.
[42] Allan DB, Caswell T, Keim NC, van der Wel CM, Verweij RW. Soft-matter/trackpy: Trackpy v0.5.0. 2021, http://dx.doi.org/10.5281/zenodo.4682814, Zenodo.
[43] Trackpy code website. 2022, http://soft-matter.github.io/trackpy/v0.5.0/. [Accessed 14 September 2022].
[44] SerialTrack software link. 2022, https://github.com/FranckLab/SerialTrack. [Accessed 14 September 2022].
[45] Abraham AV, Ram S, Chao J, Ward ES, Ober RJ. Quantitative study of single molecule location estimation techniques. Opt Express 2009;17:23352–73. http://dx.doi.org/10.1364/OE.17.023352.
[46] Small A, Stahlheber S. Fluorophore localization algorithms for super-resolution microscopy. Nature Methods 2014;11:267–79. http://dx.doi.org/10.1038/nmeth.2844.
[47] Liu S-L, Li J, Zhang Z-L, Wang Z-G, Tian Z-Q, Wang G-P, et al. Fast and high-accuracy localization for three-dimensional single-particle tracking. Sci Rep 2013;3:2462. http://dx.doi.org/10.1038/srep02462.
[48] Parthasarathy R. Rapid, accurate particle tracking by calculation of radial symmetry centers. Nature Methods 2012;9:724–6. http://dx.doi.org/10.1038/nmeth.2071.
[49] Newby JM, Schaefer AM, Lee PT, Forest MG, Lai SK. Convolutional neural networks automate detection for tracking of submicron-scale particles in 2D and 3D. Proc Natl Acad Sci 2018;115:9026–31. http://dx.doi.org/10.1073/pnas.1804420115.
[50] Cui Y, Zhang Y, Jia P, Wang Y, Huang J, Cui J, et al. Three-dimensional particle tracking velocimetry algorithm based on tetrahedron vote. Exp Fluids 2018;59:31. http://dx.doi.org/10.1007/s00348-017-2485-9.
[51] Rubbert A, Schroder W. Iterative particle matching for three-dimensional particle-tracking velocimetry. Exp Fluids 2020;61:58. http://dx.doi.org/10.1007/s00348-020-2891-2.
[52] Zhang Y, Wang Y, Yang B, He W. A particle tracking velocimetry algorithm based on the Voronoi diagram. Meas Sci Technol 2015;26:075302. http://dx.doi.org/10.1088/0957-0233/26/7/075302.
[53] Tinevez J-Y, Perry N, Schindelin J, Hoopes GM, Reynolds GD, Laplantine E, et al. TrackMate: An open and extensible platform for single-particle tracking. Methods 2017;115:80–90. http://dx.doi.org/10.1016/j.ymeth.2016.09.016.
[54] Ohmi K, Li H-Y. Particle-tracking velocimetry with new algorithms. Meas Sci Technol 2000;11:603–16. http://dx.doi.org/10.1088/0957-0233/11/6/303.
[55] Yang J, Bhattacharya K. Augmented Lagrangian digital image correlation. Exp Mech 2019;59:187–205. http://dx.doi.org/10.1007/s11340-018-00457-0.
[56] Yang J, Hazlett L, Landauer AK, Franck C. Augmented Lagrangian digital volume correlation (ALDVC). Exp Mech 2020;60:1205–23. http://dx.doi.org/10.1007/s11340-020-00607-3.
[57] Yang J, Rubino V, Ma Z, Tao J, Yin Y, McGhee A, et al. SpatioTemporally adaptive quadtree mesh (STAQ) digital image correlation for resolving large deformations around complex geometries and discontinuities. Exp Mech 2022;62:1191–215. http://dx.doi.org/10.1007/s11340-022-00872-4.
[58] Landauer AK, Patel M, Henann DL, Franck C. A q-factor-based digital image correlation algorithm (qDIC) for resolving finite deformations with degenerate speckle patterns. Exp Mech 2018;58:815–30. http://dx.doi.org/10.1007/s11340-018-0377-4.
[59] Reu PL, Blaysat B, Andò E, Bhattacharya K, Couture C, Couty V, et al. DIC challenge 2.0: Developing images and guidelines for evaluating accuracy and resolution of 2D analyses. Exp Mech 2022;1–16. http://dx.doi.org/10.1007/s11340-021-00806-6.
[60] McGhee AJ, McGhee EO, Famiglietti JE, Schulze KD. Dynamic subsurface deformation and strain of soft hydrogel interfaces using an embedded speckle pattern with 2D digital image correlation. Exp Mech 2021;61:1017–27. http://dx.doi.org/10.1007/s11340-021-00713-w.
[61] Estrada JB, Barajas C, Henann DL, Johnsen E, Franck C. High strain-rate soft material characterization via inertial cavitation. J Mech Phys Solids 2018;112:291–317. http://dx.doi.org/10.1016/j.jmps.2017.12.006.
[62] Yang J, Cramer HC, Franck C. Extracting non-linear viscoelastic material properties from violently-collapsing cavitation bubbles. Extrem. Mech. Lett. 2020;39:100839. http://dx.doi.org/10.1016/j.eml.2020.100839.
[63] Yang J, Tao J-L, Franck C. Smart digital image correlation patterns via 3D printing. Exp Mech 2021;61:1181–91. http://dx.doi.org/10.1007/s11340-021-00720-x.
[64] Summey L. Development of a simple shear impact device for evaluating cellular traumatic brain injury [Master's Thesis], University of Wisconsin-Madison; 2020.
[65] Bar-Kochba E, Scimone MT, Estrada JB, Franck C. Strain and rate-dependent neuronal injury in a 3D in vitro compression model of traumatic brain injury. Sci Rep 2016;6. http://dx.doi.org/10.1038/srep30550.
[66] Boyd S, Parikh N, Chu E, Peleato B, Eckstein J. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends Mach Learn 2011;3(1):1–122. http://dx.doi.org/10.1561/2200000016.
[67] Hansen PC, O'Leary DP. The use of the L-curve in the regularization of discrete ill-posed problems. SIAM J Sci Comput 1993;14(6):1487–503. http://dx.doi.org/10.1137/0914086.