Experiments in Fluids (2026) 67:110
https://doi.org/10.1007/s00348-026-04253-8

RESEARCH ARTICLE

STELLA: a modular framework for SpatioTemporal Event-based Lagrangian particLe trAcking

Sebastian Sachs[1,2] · Steffen Jung[3,4] · Max Kahl[3,4] · Margret Keuper[3,4] · Christian Willert[5] · Christian Cierpka[1]

Received: 15 February 2026 / Revised: 23 April 2026 / Accepted: 24 May 2026

Abstract

Event-based cameras have emerged as a powerful tool for object detection and tracking in autonomous driving, robotics, and experimental physics. In particular, they facilitate the study of complex turbulent fluid flows by enabling the tracking of numerous tiny tracer particles, while benefiting from the superior temporal resolution, high dynamic range, and low data rate of the asynchronous event stream. However, exploiting the sparse event stream requires precise and efficient data processing pipelines that either accumulate events into a dense representation or process them directly by clustering algorithms. In this study, we present a modular framework for SpatioTemporal Event-based Lagrangian particLe trAcking (STELLA), which integrates detection and tracking strategies from both pipelines into a unified tracking system. To benchmark the proposed framework, we introduce demanding synthetic and experimental datasets covering the motion of numerous particles, which are made publicly available. Leveraging the rich ground truth of these datasets, established recurrent vision transformer and heat conduction-based detection architectures are trained and applied to particle tracking in fluid flows for the first time. Using STELLA, robust and reliable particle tracking is demonstrated, achieving subpixel-accurate tracks and a mean absolute error in the predicted velocity down to 1.9 % of the peak velocity. Which is the best-performing processing pipeline strongly depends on the dynamics and composition of the considered dataset. In particular, approaches based on dense representations yield accurate tracks for high-frequency periodic particle motions. Conversely, direct processing of the event stream enables simultaneous tracking of more than 900 particles in the wake of a cylinder, with uncertainties comparable to state-of-the-art particle tracking velocimetry (PTV) using a high-speed camera. Despite significant spatial and temporal velocity gradients, slow- and fast-moving particles are precisely tracked in the event stream, challenging conventional approaches using frame-based cameras. Hence, the openly available framework STELLA paves the way for a versatile and easily accessible application of event-based cameras for flow diagnostics.

1 Introduction

The application of event-based cameras for fluid flow diagnostics sparked a lot of interest in the scientific community, as it allows for cost- and energy-efficient experimental setups (Tayarani-Najaran and Schmuker 2021). By combining superior temporal resolution with very low latency ($< 100\,\mu s$) and a high dynamic velocity range ($> 30\,\text{pixel/ms}$), event-based cameras are particularly attractive for measuring many turbulent flows with large velocity gradients (Willert 2025). To measure the velocity field within the fluid, tiny particles with high flow fidelity are seeded into the fluid. To make the particles visible, they get typically illuminated by laser light and captured in a series of images using fast cameras. Based on the individual or mean particle motion in smaller interrogation windows,

✉ Sebastian Sachs
sebastian.sachs@tu-ilmenau.de

✉ Christian Cierpka
christian.cierpka@tu-ilmenau.de

1 Institute of Thermodynamics and Fluid Mechanics and Institute of Micro- and Nanotechnologies, Technische Universität Ilmenau, Ilmenau 98693, Germany

2 Visiting researcher at Department of Industrial Engineering, Alma Mater Studiorum-University of Bologna, Bologna 40126, Italy

3 Data and Web Science Group, University of Mannheim, Mannheim 68161, Germany

4 Max-Planck-Institute for Informatics, Saarland Informatics Campus, Saarbrücken 66123, Germany

5 Institute of Propulsion Technology, Deutsches Zentrum für Luft- und Raumfahrt (DLR), Köln 51147, Germany

the underlying fluid flow is inferred using particle tracking velocimetry (PTV) or particle image velocimetry (PIV), respectively (Kähler et al. 2012, 2016). In this field, commercially available event-based cameras with resolutions up to 1 MP came into focus in recent years, as they remove the restriction to a fixed frame rate. In particular, this enabled a high dynamic velocity range, as fast and slow moving particles leave an individual footprint in the data obtained by the event-based camera. Moreover, the need for expensive high-speed cameras, pulsed lasers with high power and precise synchronization is obviated, as event-based cameras allow for low-cost constant illumination due to their continuous recording with a high dynamic range of up to 130 dB.

At the heart of the technique, event-based cameras are based on biologically inspired neuromorphic vision sensors, which capture pixelwise intensity changes as asynchronous events (Mahowald and Mead 1991). Hence, each pixel operates independently and returns an event whenever the intensity change is greater than an adjustable threshold on a logarithmic scale, which contrasts with conventional synchronous frame-based cameras. In this way, an event stream is recorded, containing the position (x, y) of the event within the pixel array, the timestamp (t) with microsecond resolution, and the polarity (p) of the event, i.e., indicating an increasing or decreasing intensity. As the majority of the imaged scene is usually static in nature, only the changes are transmitted such that the event stream inherently possesses a sparse nature in time and space, which reduces the acquired data rate significantly and enables real-time processing (Drazen et al. 2011; Rusch and Rösgen 2023). Furthermore, motion blur is avoided as long as particles do not approach a velocity of 10^3 pixel/ms , i.e., moving more than one pixel within a microsecond. In typical macroscopic and microscopic measurement setups with magnifications of 0.1 and 20, respectively, this corresponds to approx. 48.6 m/s and 0.243 m/s, respectively.

However, the successful establishment of event-based particle tracking based on the essentially binary nature of the event stream faces inherent challenges that are less prominent in other disciplines. First, the objective is to identify and track a large number of very small and similar appearing objects within the field-of-view (FOV). This contrasts with typical object tracking tasks of cars, aircrafts, or pedestrians, which occupy fewer but larger areas of the FOV. At this point, resolution becomes a crucial factor, which is still low compared to frame-based cameras with usually several megapixel. Second, events are read out line by line by an arbiter, which can lead to readout errors at high event rates. The arbiter works according to the first-in, first-out principle (Litzenberger et al. 2006), while each event of an active line is assigned a timestamp. However, if many events occur simultaneously, individual timestamps or even entire lines of data are irretrievably lost as the arbiter becomes saturated (Willert 2023; Cao et al. 2025). Due to this loss of information, continuous tracking of particles is unfeasible, which defines natural limits of the application. Third, the event stream possesses a unique data format that differs from familiar grayscale images. While algorithms to precisely determine particle image positions have been developed and established over decades, customized algorithms to track objects directly in the event stream are necessary to exploit the outlined potential for flow diagnostics.

Various processing pipelines have emerged to handle the data obtained from event-based cameras (Wang et al. 2024), as shown schematically in Fig. 1. In the upper pipeline, the recorded events are accumulated across an adjustable time interval Δt to create so-called pseudo-frames. In various variants, the number of events within Δt, the last timestamp of the events, or the mere occurrence of events can be encoded pixelwise (Zubic et al. 2023). This provides a dense representation of the data, which can be processed by conventional image-based processing algorithms for synchronous detection and tracking of particle images. Successful implementations demonstrated both PIV (Willert and Klinner 2022; Willert 2023, 2025; Franceschelli et al. 2025) and PTV (Wang et al. 2020; Howell et al. 2020; Zeng et al. 2025; Cao et al. 2025) in various flow scenarios with fast image evaluation. Furthermore, artificial neural networks (ANN) achieved great success in particle detection based on grayscale images (König et al. 2020; Sachs et al. 2023; Ratz et al. 2023; Godbersen et al. 2024; Coutinho et al. 2025), extending the limits of conventional algorithms. By incorporating tailored ANN architectures into this processing pipeline, the higher precision regarding particle position and velocity as well as reduced inference time can be transferred to pseudo-frames. In particular, approaches based on pseudo-frames such as the Mixture-of-Experts heat conduction-based detection (MvHeatDET (Wang et al. 2024)) demonstrated reliable detection of, e.g., pedestrians, table tennis balls, or bicycles. For quantification, a mean Average Precision (mAP) of 52.9 was achieved, which measures how reliably an object-detection algorithm finds and correctly localizes objects by balancing how many predicted objects are correct against how many true objects are successfully detected, averaged across all object categories. In comparison, the mAP achieved falls within the range of modern object detection algorithms benchmarked on the frame-based COCO dataset, peaking at a mAP of 63.1 when using transformer-based architectures (Lin et al. 2014; Amjoud and Amrouch 2023). When combined with Siamese networks or tracklet matchers, detection and tracking can be further integrated in one pipeline (Wang et al. 2025). However, individual timestamps of events are lost during accumulation, and image-based algorithms do not benefit from the spatial and temporal sparsity of the event stream.

The associated information loss can be reduced by accumulating events over shorter time intervals δ in n microbins, which are merged into an event-tensor covering $\Delta t = n\delta$ (second row in Fig. 1). When used as input for recurrent neural networks (RNN), the encoded temporal domain is considered by, e.g., a Long Short-Term Memory (LSTM (Hochreiter and Schmidhuber 1997)) network as used in the Recurrent Vision Transformer (RVT (Gehrig and Scaramuzza 2023)). Besides, temporal correlations can be considered by utilizing the residual voltage memory of leaky integrate-and-fire neurons, which are used in the spiking vision transformer (SpikingViT (Yu et al. 2025)). Both approaches perform considerably well when detecting large objects such as cars or pedestrians in complex traffic data, achieving a mAP of up to 47.4.

In the lower pipeline, the event stream is processed directly by clustering events associated to an individual particle based on their spatiotemporal coherence. This approach was implemented by Rusch and Rösgen (2023) in an algorithm for three-dimensional PTV, which is reported to work in real-time for a limited number of particles to be tracked simultaneously. Based on the event clusters, particle tracks can be obtained by fitting individual path lines (Rusch and Rösgen 2023; Cierpka et al. 2013) or applying a Kalman filter (Borer et al. 2017; Wang et al. 2020; Reed et al. 2025; Gunasekaran et al. 2025). By directly processing the event stream, spatiotemporal sparsity and high temporal resolution are exploited to achieve a high dynamic velocity range of the particle tracks, offering unique advantages compared to the other pipelines.

By using spiking neural networks (SNN), direct processing of the event stream is further conceivable for reliable and computationally fast particle tracking (Wang et al. 2024). These networks are inspired by the biological nervous system and communicate via binary spikes, i.e., discrete pulses, which inherently take the temporal domain into account (Cordone et al. 2022). When compiled on neuromorphic hardware, SNNs are a natural match for event-based PTV by leveraging the temporal and spatial sparsity of the event stream in asynchronous processing (Schuman et al. 2022). However, due to a lack of tailored architectures and training strategies, their performance currently falls behind established ANNs in object detection tasks (Wang et al. 2024).

Furthermore, precise detection of particle positions with high temporal resolution was demonstrated in hybrid approaches (Zeng et al. 2025, 2025b). For this, objects are tracked based on data from an event-based camera, while features from images obtained by a frame-based camera at lower frame rate are integrated to enhance detection precision and compensate for the limited contextual information in the event stream (Gehrig and Scaramuzza 2024). Although the position uncertainty is reduced by merging the data from the frame-based and event-based cameras, this diminishes the advantages of low data rates and cost- and

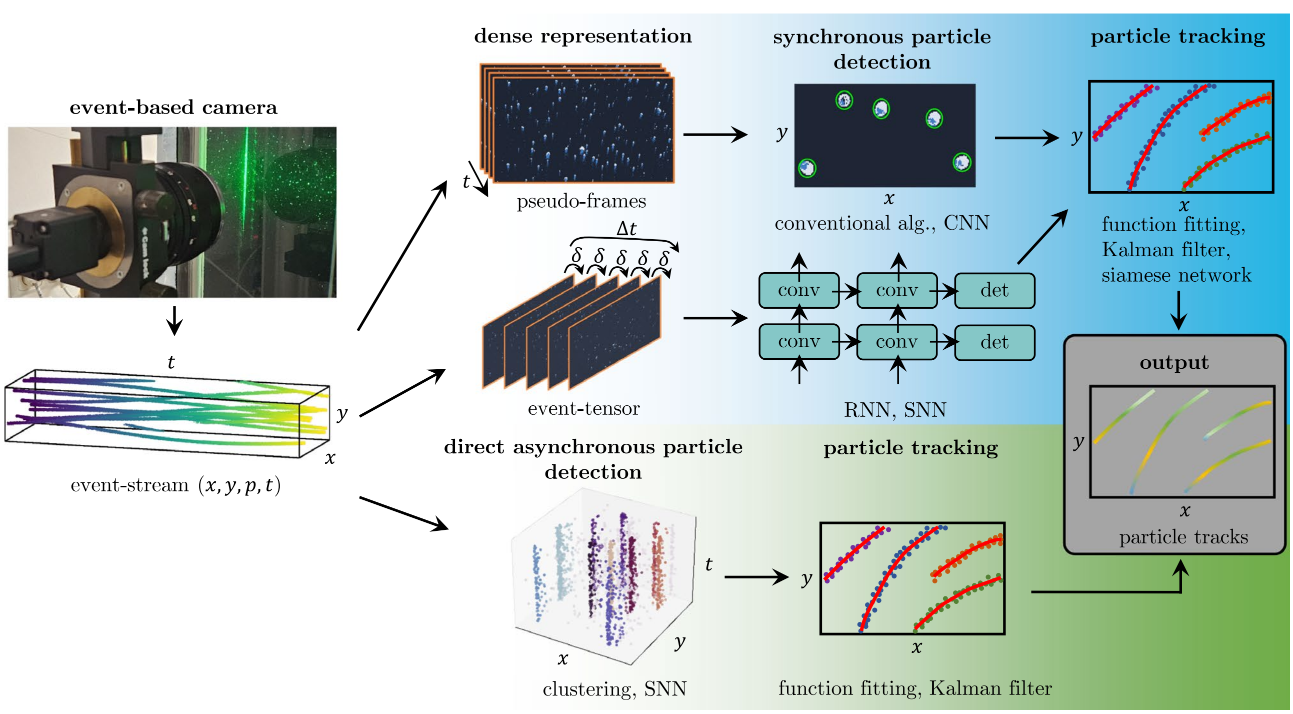


Fig. 1 Schematic representation of established processing pipelines using either a dense representation or direct processing of events

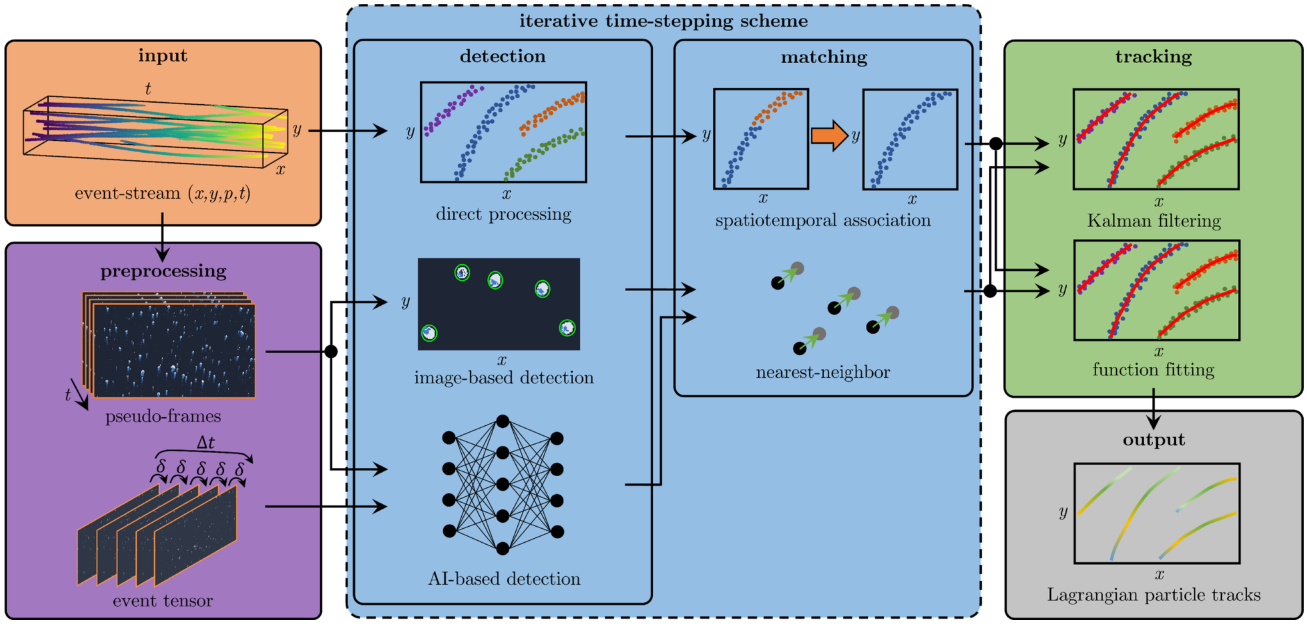


Fig. 2 Schematic workflow of the proposed modular framework STELLA, integrating an iterative time-stepping scheme to enable data processing within the proposed processing pipelines

energy-efficient measurement setups. In addition, a precise synchronization between both cameras increases complexity.

Despite the advances in various processing pipelines, a systematic evaluation of the relative strengths and limitations of the different strategies for event-based particle tracking is lacking. As existing studies typically focus on either dense representations or direct processing, it remains unclear under which flow conditions a given approach is preferable, and how different detection and tracking algorithms perform in combination.

In this work, a modular framework for SpatioTemporal Event-based Lagrangian particLe trAcking (STELLA) is introduced to provide a unified platform and evaluate the performance of algorithms from different processing pipelines on demanding benchmark datasets. The framework integrates algorithms for detecting, matching, and tracking particles in the *xy*-plane, which can be combined in different ways to effectively adapt for different dynamics and compositions of experimental data. This modularity further allows to evaluate the impact of each processing step on the overall performance. In addition, both conventional and artificial intelligence (AI)-based algorithms are incorporated in the evaluation to shed light on underlying characteristics of the proposed processing pipelines, highlighting their potential and current limitations for particle tracking. The algorithms are evaluated for different flow regimes in newly generated synthetic and experimental benchmark datasets that offer a rich ground truth. Hence, the main contribution of this work is not a single algorithm, but a comprehensive framework and datasets that provide new insights into the performance of different processing pipelines for event-based particle tracking. The framework STELLA as well as the datasets used with and without ground truth are available as open source at https://github.com/sesa1504/STELLA.

Table 1 Combinations of detection, matching and tracking strategies used in STELLA along with the associated input event representation

Detection	Event representation	Matching	Tracking
image-based	pseudo-frames	nearest-neighbor	Kalman / spline / hybrid
DBSCAN	event stream	nearest-neighbor	Kalman / spline / hybrid
kd-tree	event stream	query-based assignment to previous clusters	Kalman / spline / hybrid
pixelwise extension	pseudo-frames & event stream	cluster continuation by temporal overlap	Kalman / spline / hybrid
MvHeatDET (Wang et al. 2024)	pseudo-frames	nearest-neighbor	Kalman / spline / hybrid
RVT (Gehrig and Scaramuzza 2023)	event-tensors	nearest-neighbor	Kalman / spline / hybrid

2 Methods

2.1 Modular particle tracking framework

The proposed framework STELLA combines various algorithms in a modular way, which can be combined in the most appropriate way for a current particle tracking problem. In this way, the entire framework or only selected components can be utilized. The event stream obtained directly from the event-based camera or pseudo-frames using a user-defined image processing pipeline can be used as input.

Within STELLA, the processing pipelines discussed in Sect. 1 and shown in Fig. 1 are implemented by different algorithms. In this paragraph, a brief overview is given on the structure of STELLA, while a detailed description of the algorithms can be found in subsequent sections. To process the event stream, the algorithms are organized in a succession of preprocessing, detection, matching, and tracking as depicted schematically in Fig. 2. During preprocessing, the event stream is processed directly or converted into a dense representation, i.e., pseudo-frames or event-tensors. Subsequently, an iterative time-stepping scheme is used to assign incoming events that belong to individual particles to local clusters within the actual time step. This is achieved by means of spatiotemporal coherence in direct processing, with image-based detection or with AI-based algorithms. Once all local event clusters are identified within the actual time step, the local clusters are matched to existing global event cluster that persist through multiple time steps. This is done by spatiotemporal association or using a nearest-neighbor algorithm. Upon completion of the time-stepping scheme, particle tracking is performed for each global event cluster using Kalman filtering, function fitting, or hybrid approaches. Finally, Lagrangian particle tracks are generated as output. An overview on possible combinations of the different detection and tracking algorithms is given in Table 1.

2.2 Preprocessing

The asynchronously recorded event stream contains the position (x, y) of each event in pixel, a timestamp (t) with microsecond resolution, and the polarity (p) of the event, which is either positive (1) or negative (0). To preprocess the event stream, a region of interest (ROI) is initially defined in the image plane (x, y) and events outside the ROI are discarded. Furthermore, only events with positive polarity were considered in this work, as negative events are found not suitable and not required for event-based particle tracking (Willert et al. 2026). Hence, avoiding negative events enabled a higher bandwidth for positive events without facing readout errors. To convert the event stream into an input format suitable for the subsequent algorithms, it can be processed directly or transferred into a dense representation.

2.2.1 Direct processing of the event stream

In direct processing, the event stream is divided into shorter sequences with adjustable time duration Δt. These sequences are loaded and processed iteratively in the subsequent time-stepping scheme by using tailored algorithms as discussed in Sects. 2.3.2 and 2.3.3. However, the events are not preprocessed further and retain their original format. In upcoming architectures based on bio-inspired spiking neural networks (SNN), event streams are expected to be processed directly. However, in the current version, these architectures have not been integrated into this work due to their lower performance compared to architectures that take dense representations as input (Su et al. 2023; Wang et al. 2024, 2024).

2.2.2 Dense representations: pseudo-frames and event-tensors

To perform conventional image-based or AI-based particle detection, the event stream is converted into a dense representation. For this, the events are compressed into either pseudo-frames with a bit depth of 8 bit or event-tensors. In the former, events are accumulated within an adjustable time duration Δt and converted into an image-like representation. Specifically, pixel in which an event occurred within Δt are set to 255, while the rest of the image is assigned a value of 1. In this way, a dataset of pseudo-frames is created, which can be processed in the subsequent time-stepping scheme using conventional image processing methods as discussed in Sect. 2.3.1. The pseudo-frames can also serve as input for various neural network architectures such as MvHeatDET (Wang et al. 2024). However, the recurrent neural network RVT (Gehrig and Scaramuzza 2023) used in this work requires conversion to event-tensors.

In event-tensors, events are accumulated over a shorter time duration δ into bins. A number n of these bins are stacked in an event-tensor covering a larger time interval $\Delta t = n\delta$ and passed on to the neural network (Gehrig and Scaramuzza 2023). In this way, the network is able to learn from the temporal sequences of moving particles and consider them in the detection. However, a lower limit is given for δ by considering the readout latency in the order of 100 µs.

2.3 Detection and matching

Once the event stream is converted into a suitable representation, individual particles are identified based on coherent clusters of events. For this, the data is divided into time sequences of adjustable length Δt and processed in an iterative time-stepping scheme. After detecting local event clusters in the current time step, they are assigned to existing global clusters. If no assignment is possible, new global clusters are initiated. Depending on the event representation, four alternative strategies are distinguished: image-based detection on pseudo-frames, direct event clustering, the pixelwise extension method, and AI-based detection on dense representations.

2.3.1 Image-based detection

For image-based processing, a conventional algorithm is used to detect particle images based on pseudo-frames in two steps. First, a circular averaging filter is applied to transform the pseudo-frames into grayscale images and simplify the detection of the particle image center (Wang et al. 2020). The filter size is set based on the particle image sizes. Second, the grayscale images are segmented using Otsu's threshold value (Otsu 1979) and particle image positions (x, y) are identified. Detected particle images with an area below an adjustable threshold value are discarded. Finally, the found particle image positions are matched to existing global clusters using a simple nearest-neighbor algorithm. However, only the particle image positions are stored and processed for tracking, not the associated events.

2.3.2 Direct event clustering: DBSCAN and kd-tree

Direct processing of the event stream is achieved through clustering based on the spatiotemporal coherence of the events. However, since only short sequences of events are processed iteratively within a relatively small time interval, temporal coherence is intrinsically provided. For spatial clustering, algorithms for density-based spatial clustering (DBSCAN, Ester et al. (1996)) or a two-dimensional search tree (kd-tree, Bentley (1975); Maneewongvatana and Mount (1999)) are used. Using either algorithm, a search radius for neighboring events and a minimum number of events per cluster are specified. In the case of kd-tree, the median values of the local event cluster positions from the previous time step are used as query points, while currently found clusters are assigned to the respective previous clusters directly. However, the remaining unassigned events are used to find new local clusters by applying DBSCAN. When using DBSCAN, local clusters found in the current time step are assigned to existing global clusters by applying a nearest neighbor algorithm based on the median values of the clusters. If no match with a Euclidean distance of less than an adjustable threshold value is found, a new global cluster is assigned.

2.3.3 Direct event clustering: pixelwise extension

In an alternative approach, unfiltered pseudo-frames are used to identify local event clusters. While the detection of particle images is based on the image-based processing described above, the events assigned to a cluster in the pseudo-frame are directly stored and processed further. To simplify assignment to existing global clusters, some of the events from the previous time step are reused in the current time step. In this way, a spatial overlap is created between the clusters found in the current and previous time steps. Local clusters found without overlap to existing global clusters initiate new global clusters. This method is referred to as pixelwise extension in the following.

2.3.4 AI-based detection

STELLA integrates learning-based object detectors to identify particle images in dense representations. In this work, two architectures were considered: the Recurrent Vision Transformer (RVT (Gehrig and Scaramuzza 2023)), and the Mixture-of-Experts heat conduction-based detector MvHeatDET (Wang et al. 2024). These architectures are build for detecting objects such as cars or pedestrians based on the event stream with high precision. As described in Sect. 2.2.2, depending on the architecture, the event stream is converted either to pseudo-frames (for MvHeatDET) or event-tensors (for RVT). A detailed overview of the architectures used can be found in Section II (SI). Using either architecture, the neural network is first trained in a supervised manner based on known ground truth of the particle image positions (x, y) and the corresponding timestamp (t). The preprocessing and structure of the datasets used for training are described in more detail in Sect. 3. Subsequently, inference is performed on synthetic or experimental data without knowing the underlying ground truth. Since the networks output the position (x, y) and timestamp (t) of each detected particle image, a matching step is required. For this, a simple nearest neighbor algorithm is used.

2.4 Tracking and correction

Once the events are clustered and assigned to individual particles, Lagrangian particle tracks are derived by applying a Kalman filter, by function fitting or hybrid approaches of both. Within this step, both the position and velocity of the particles are estimated based on the event positions (x, y) or image-based detections and timestamps (t).

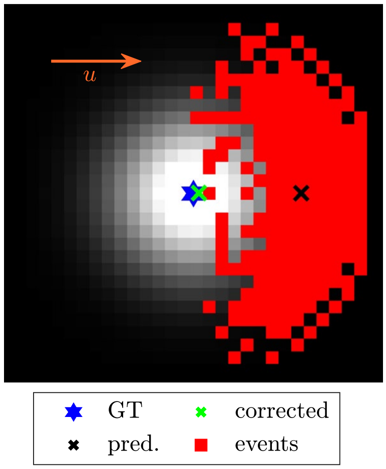


Fig. 3 Synthetic grayscale image of a particle moving at constant velocity to the right. Overlaid are triggered events as well as the ground truth position, predicted position, and corrected position of the particle

2.4.1 Kalman filtering

The Kalman filter works iteratively by updating its state variables with incoming events (Kalman 1960; Julier and Uhlmann 1997; AlSattam et al. 2024). In the chosen model, the position $(x(t), y(t))$, velocity $(u_x(t), u_y(t))$, and acceleration $(a_x(t), a_y(t))$ are predicted independently in x- and y-direction, respectively. After initialization, the state vector $\boldsymbol{X} = [x\ u_x\ a_x]^{\mathrm{T}}$, e.g. in x-direction, is updated in subsequent prediction and correction steps. Within the prediction, the process noise covariance matrix $\boldsymbol{Q} \in \mathbb{R}^{3\times3}$ is introduced to quantify the uncertainty of the underlying motion model. Hence, $\boldsymbol{Q}$ significantly affects the flexibility of the Kalman filter in response to changes in position or smoothing of measurement noise. As the optimal setting $\boldsymbol{Q}$ highly depends on the data under investigation, including the number of data points per track and the particle dynamics, no universal values can be claimed. Instead, $\boldsymbol{Q}$ was carefully fine-tuned for the respective dataset. The values used to process the synthetic benchmark data are listed in Table S2 (SI).

During the correction step, the measurement noise covariance R serves to quantify the uncertainty in the measured values. Similar to $\boldsymbol{Q}$, fine-tuning of the measurement noise covariance R depends on the data under investigation and is affected by experimental parameters such as the optical setup, particle image size, particle seeding concentration, and illumination conditions. Hence, no universal value can be specified and R was chosen for the respective dataset, as exemplified in Table S2 (SI) for synthetic data. By using the Kalman filter, the position and velocity of the particle tracks

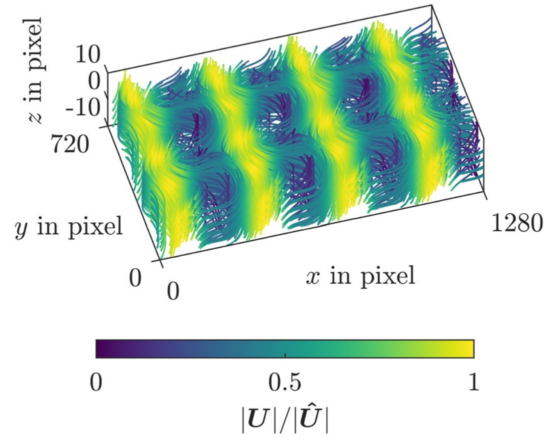


Fig. 4 Synthetic velocity field visualized by streamlines within the computational domain $([\lambda_{xy}, \lambda_z, \hat{A}] = [333.\bar{3}\,\text{pixel}, 0.\bar{3}\,\text{pixel}, 500\,\text{pixel/s}])$. The streamlines are color-coded by the absolute velocity $|\boldsymbol{U}|$

are determined directly without numerical differentiation, while smoothing out measurement noise. Once the Kalman filter has been applied, subsampling is performed with a configurable number of reference points within a time step Δt. Further details and mathematical foundation of the applied Kalman filter can be found in Section III (SI).

2.4.2 Spline fitting

Alternatively, Lagrangian particle tracks $(x(t), y(t))$ are estimated by fitting cubic B-splines. For this, events with the same timestamp within a cluster are initially removed. Then, the fitting procedure is performed on a time grid with adjustable resolution. As a crucial hyperparameter, the smoothing of the B-splines is iteratively reduced until a target root-mean-square error (RMSE) between the fit function and the measured values is achieved. Additional termination criteria include overshooting by limiting the possible fitted position, numerical stability by excluding invalid entries, timeouts, and no significant RMSE improvement. Finally, the velocities $(u_x(t), u_y(t))$ are determined by analytical derivation of the B-splines at the desired temporal resolution.

In both cases, either the clustered events or representative points from image-based or AI-based detection are used for tracking. The input data significantly affects the performance and hyperparameter tuning, as discussed in Sect. 4.1.

2.4.3 Hybrid tracking approaches

Spline fitting was found to yield smooth velocity estimates that are often more accurate than the velocity predicted by the Kalman filter. However, when applied directly to event clusters, there is a risk of smoothing out details in the particle position or of overshooting, especially for rapidly changing tracks. Conversely, Kalman filtering is typically more robust to predict the position of particle tracks. To combine the individual strengths of the Kalman filter and spline fitting in estimating the particle tracks, three hybrid approaches are implemented in STELLA. These hybrid approaches are designed to identify whether spline fitting becomes more reliable when applied to the particle positions and/or velocities predicted by the Kalman filter instead of directly using the event clusters. First, track positions can be generated using the Kalman filter and velocities by using spline fitting, which is referred to as hybrid-($\mathrm{X}^{\mathrm{K}}, \mathrm{u}^{\mathrm{S}}$). Second, track positions and velocities can first be determined using the Kalman filter. Then, the smoothed velocity obtained by the Kalman filter is used for spline fitting, which is referred to as hybrid-($\mathrm{X}^{\mathrm{K}}, \mathrm{u}^{\mathrm{KS}}$). Third, both the positions and velocities obtained by the Kalman filter can be used for spline fitting, which is referred to as hybrid-($\mathrm{X}^{\mathrm{KS}}, \mathrm{u}^{\mathrm{KS}}$). The performance of these hybrid approaches is quantified as part of Sect. 4.

2.4.4 Position correction

The event-based camera triggers positive events, if the intensity in the respective pixel exceeds the previous value by a defined threshold. As depicted in Fig. 3, this leads to events occurring in the front end of the particle image, i.e., towards the direction of motion. Hence, the geometric center of the positive events is shifted in motion direction, leading to off-center predictions (Reed et al. 2025). To correct for the off-center predictions, a correction step is implemented after generating the particle tracks. For this, the scattering $\sigma_\perp$ of the events perpendicular to the direction of motion of the particle is first determined. Subsequently, the position $[x, y]$ is corrected according to

$$[x^*, y^*] = [x, y] - \beta\sigma_\perp \frac{\boldsymbol{u}}{||\boldsymbol{u}||}, \tag{1}$$

where β denotes an adjustable parameter. By applying the correction, the estimated center of the particle image is shifted in the opposite direction of the direction of motion. In this way, the offset of the positive events, which are mostly recorded in the front end of the particle image, i.e. forming sickle-shaped patterns, is corrected as shown in Fig. 3.

As STELLA is meant as an open framework also other fitting procedures may be applied. In general, event-based recording allows for higher order fitting with all the benefits concerning velocity and position correction, as well as the derivation of accelerations (Cierpka et al. 2013).

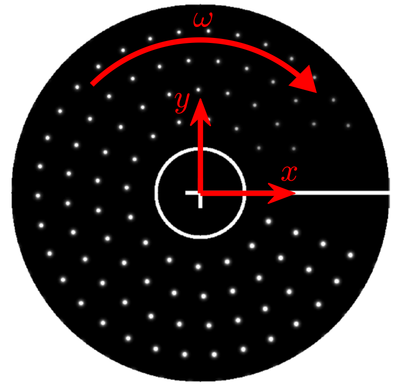


Fig. 5 Printed particle images arranged in concentric circles mounted on a rotating disk

3 Datasets

Three datasets are used to evaluate and compare the performance of the proposed algorithms within the STELLA framework. First, a synthetic dataset is generated that maps the motion of tracer particles in a deterministic velocity field. Since the position and velocity of the particles are well known as ground truth, the dataset allows for a clear comparison. Second, the circular motion of printed particle images on a rotating disc, which was recorded with an event camera, is examined. Although this is experimental data, the position of the printed particle images is known a priori and allows to derive the ground truth. Third, the motion of tracer particles in the wake flow of a cylinder is evaluated. In the case of this purely experimental data, there is no ground truth available. However, a reference measurement is derived by recording the fluid flow simultaneously using a high-speed camera synchronized to the event-based camera. All datasets are openly available for own benchmark tests at https://github.com/sesa1504/STELLA.

3.1 Synthetic vortex flow

The performance of the proposed framework is first evaluated based on the motion of tracer particles in a synthetic fluid flow. In the synthetic data, the particles follow the fluid flow perfectly, which is prescribed by the velocity field

$$\boldsymbol{U} = \hat{A} \begin{bmatrix} \sin\left(\frac{2\pi}{\lambda_z} z\right) + \cos\left(\frac{2\pi}{\lambda_{xy}} y\right) \\ \sin\left(\frac{2\pi}{\lambda_{xy}} x\right) + \cos\left(\frac{2\pi}{\lambda_z} z\right) \\ a\left(\sin\left(\frac{2\pi}{\lambda_{xy}} y\right) + \cos\left(\frac{2\pi}{\lambda_{xy}} x\right)\right) \end{bmatrix}. \tag{2}$$

The velocity field is divergence-free and inspired by an Arnold-Beltrami-Childress flow (Fre and Sorin 2015). However, the velocity component U_z in the out-of-plane direction is scaled by a factor of $a = 0.03$ to obtain elongated particle tracks before the particles leave the thin light sheet. Nevertheless, the chosen flow is of 3D nature and contains highly curved streamlines with significant acceleration and deceleration, as shown in Fig. 4. To obtain a rich dataset for evaluating and training neural networks, the wavelengths $\lambda_{xy} = (333.\bar{3} \pm 56)$ pixel and $\lambda_z = (0.\bar{3} \pm 56 \times 10^{-6})$ pixel as well as the velocity amplitude $\hat{A} = (500 \pm 100)$ pixel/s are varied as depicted in Fig. S1.

With the fluid flow set, 142 sequences with an individual length of 2 s were generated in two steps. First, a number $N = 300 \pm 100$ of tracer particles with a mean size $\bar{d}_\mathrm{p} = (35 \pm 10.5)\,\mu\mathrm{m}$ were randomly positioned in the fluid domain and their motion was calculated based on the underlying fluid flow. To model polydisperse particles, the particle size was further perturbed by additive Gaussian noise with a standard deviation of 20 % of its mean value. The positions and velocities of the particles serve as ground truth. Periodic boundary conditions apply at the edges of the computational domain, which implies that particles leaving the domain are reinitialized on the opposite side keeping the number of particles in the FOV constant. However, if a reinitialized particle would leave the domain again in the next time step, it is reinitialized at a random position. To convert the particle motion into an event stream, grayscale images were first generated with a frame rate of 1 kHz and a bit depth of 8 bit. For this, the particle images possess a Gaussian intensity distribution, whose maximum intensity and standard deviation depend on the particle size and depth position relative to a light sheet. To simulate the latter, the particle intensity was modeled as a Gaussian distribution in depth direction. To further include the influence of different optical setups, the effective particle image size

$$\sigma_\mathrm{p} = \frac{\sqrt{M^2 d_\mathrm{p}^2 + d_\mathrm{opt}^2}}{A_\mathrm{S}}\, p_\mathrm{r} \tag{3}$$

was randomly scaled by a factor p_r uniformly distributed within $\pm 30\,\%$ around the nominal value. Here, the magnification, sensor pixel size, diffraction-limited diameter, aperture number, and light wavelength are denoted as $M = 0.05$, $A_\mathrm{S} = 4.86\,\mu\mathrm{m}$, $d_\mathrm{opt} = 2.44(1 + M) f^{\#} \lambda_\mathrm{L}$, $f^{\#} = 3$, and $\lambda_\mathrm{L} = 532\,\mathrm{nm}$, respectively. The intensity of overlapping particle images was accumulated.

Second, the frame-based data was converted to event streams using the event simulator ESIM (Rebecq et al. 2018). Proven to produce realistic data, ESIM takes the frame-based sequences as input and converts them to events that are used as input for the framework. In this step, measurement noise in the form of randomly triggered events is applied. To train the neural networks, the data was split in a ratio of 70/10/20 for training/validation/testing.

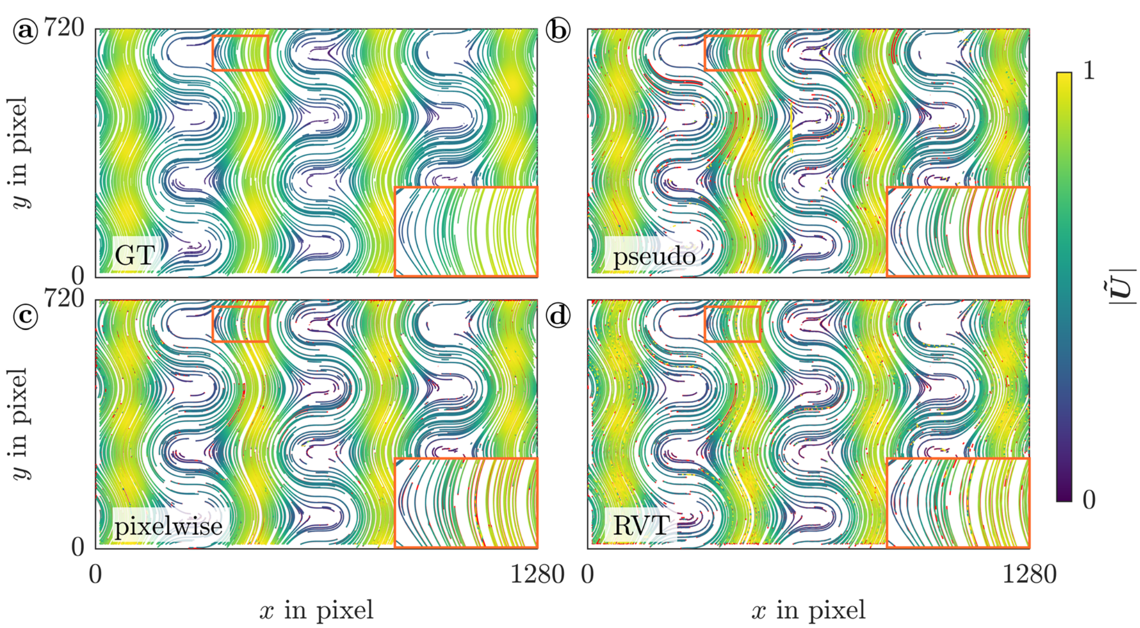


Fig. 6 Particle tracks obtained from the ground truth (**a**), pseudo-frames (**b**), pixelwise extension (**c**) and RVT (**d**) based on synthetic data. The tracks are color-coded by the normalized absolute velocity with the colorbar applying to all sub-plots. In (**b**,**c**,**d**), the ground truth is displayed as red tracks in the background. An enlarged view of the region highlighted by a orange frame is given as inset

3.2 Rotating chopper wheel

To generate a dataset with high similarity to the experimental velocity measurements in fluids and known ground truth, the motion of printed particle images fixed on a rotating disc was recorded using an event-based camera (Prophesee EVK4, 1280×720 pixel) equipped with a lens of focal length 35 mm (Carl Zeiss AG). The particle images possess a Gaussian intensity distribution and are arranged on concentric circles as depicted in Fig. 5. The number of particle images was increased in four different discs, denoted as PI1-PI4. To create a challenging dataset and mimic experimental conditions, the maximum intensity of the particle images decreases in the circumferential direction. The angular position of the rotating disc was marked by a radial line, which was masked in the evaluation. To map the ground truth on the position of the particle images to the event stream, the pivot point was determined and the radial particle positions were scaled. Furthermore, eccentric rotation was unavoidable due to manual assembly of the disc. This was corrected by determining the geometric centers in the event stream along a full rotation. The rotation of the disc was controlled by using an optical chopper wheel rotating at 2.5 Hz to 49.5 Hz. Since the angular speed ω was kept constant within each recording, the circumferential speed of the particle images is given by multiplying with the radial position r according to $u_\phi = r\omega$. Finally, the particle images were illuminated using a high-power LED (Solis525C, Thorlabs Inc.) and recorded perpendicular to the rotating disc for a duration of 30 s per disk and rotation speed. For training the neural networks, the recordings were divided into 2 s long sequences and split in a ratio of 70/10/20 for training/validation/testing.

3.3 Flow in the wake of a cylinder

The fluid flow in the wake of the cylinder with a diameter of $d_{\text{cylinder}} = 21\,\text{mm}$ positioned in a water channel was investigated in a Reynolds number range of $Re = u_\infty d_{\text{cylinder}}/\nu \in [3.1, 20.6] \times 10^3$, where the free stream velocity in main flow direction and fluid viscosity are given by u_∞ and ν, respectively. To visualize the fluid flow, the water was seeded with polydisperse tracer particles (polyamide, VESTOSINT® 2070, Evonik Industries AG) with a mean diameter of 6 µm. Furthermore, a continuous wave laser (12 W, $\lambda_{\text{L}} = 532\,\text{nm}$, Frankfurt Laser Company) equipped with a cylindrical lens (focal length 10 mm) was used to create a light sheet. The motion of the tracer particles was captured by a high-speed camera (Imager HS 4M, 2016×2016 pixel, 12 bit, LaVision GmbH) and an event-based camera (Prophesee EVK4, 1280×720 pixel). Both cameras were aligned almost perpendicular to the light sheet at a distance of approx. 0.75 m, while small differences in the angle of observation were corrected by image dewarping based on a calibration. The frame rate $f_{\text{img}} \in [129, 700]\,\text{Hz}$ and exposure time $T_{\text{exp}} \in [79.6, 325]\,\mu\text{s}$ of the high-speed camera were set in dependence on the respective Reynolds number. Due to the decreasing exposure time, the achievable signal-to-noise ratio (SNR) of the high-speed camera decreased simultaneously and restricted the chosen measurement range. However, using only the event-based camera, a larger range up to $Re = 41.6 \times 10^3$ was recorded.

The FOV of the high-speed camera covered an area of approx. $160 \times 160\,\text{mm}^2$ by mounting a lens with a focal length of 100 mm (Nikon). The event-based camera was equipped with an 35 mm lens (Carl Zeiss AG) to achieve a FOV with a size of approx. $193 \times 109\,\text{mm}^2$. Although the FOV of both cameras overlapped, the high-speed camera provides a higher spatial resolution with 12.54 pixel/mm

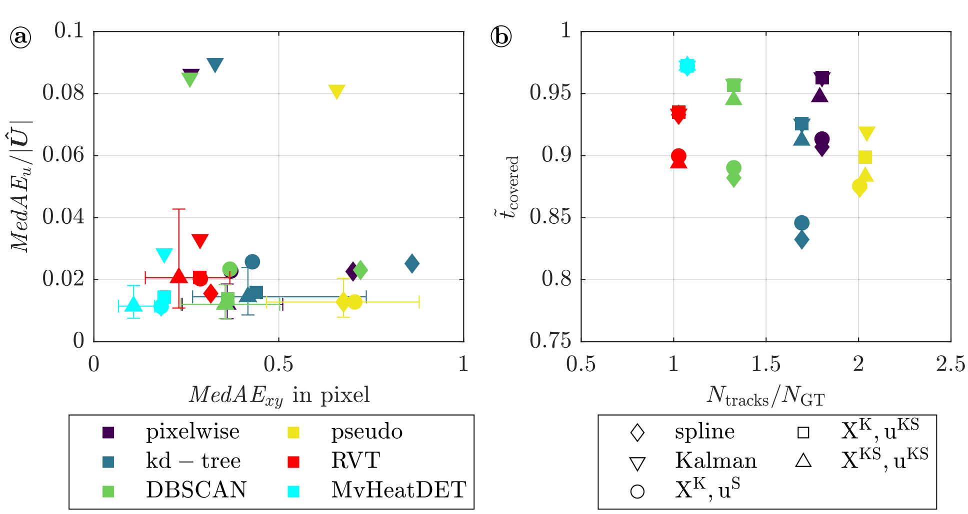


Fig. 7 Relative MedAE in the velocity $MedAE_u/|\hat{U}|$ as a function of $MedAE_{xy}$ for various combinations of detection (indicated by the color) and tracking (indicated by the shape) algorithms (**a**). In (**b**), the relative proportion $\tilde{\iota}_{\text{covered}}$ is depicted against the ratio $N_{\text{tracks}}/N_{\text{GT}}$ for the same combination of algorithms

instead of 6.62 pixel/mm when using the event-based camera. This improves the fidelity of the reference measurement using conventional PTV, which needs to be considered when comparing results from conventional and event-based PTV. Both cameras were synchronized using a programmable timing unit (PTU X, LaVision GmbH). For this, trigger events were recorded in the event stream, which indicate the timing of image capture by the high-speed camera. To compare the data from both cameras, scaling to physical units and image dewarping were performed using images of a calibration target (LaVision GmbH). The static calibration target was illuminated with a triggered white light source to record events. Based on the position of the calibration target, a common coordinate system was defined for both cameras.

4 Results and discussion

4.1 Performance metrics under known ground truth

The performance of the processing pipelines currently implemented in STELLA is first evaluated based on synthetic data with known ground truth. For this, the dataset described in Sect. 3.1 is used, which provides particle tracks with known position and velocity. The hyperparameter used for evaluation are listen in Table S2 (SI). In Fig. 6a, the stacked ground truth tracks of one sequence are depicted in a top view, while the normalized absolute velocity $|\tilde{\boldsymbol{U}}| = |\boldsymbol{U}|/|\hat{\boldsymbol{U}}|$ using the velocity $|\hat{\boldsymbol{U}}| = 600\,\mathrm{pixel/s}$ is color-coded. The inset provides an enlarged view on the area highlighted by the orange frame. In the dataset, the particle tracks are highly curved and experience significant acceleration and deceleration along their path.

A qualitative comparison of the tracking results is provided in Fig. 6b–d, which shows the predicted particle tracks using pseudo-frames (b), direct processing with pixelwise extension (c), and AI-based processing with RVT (d). Furthermore, a lively representation can be found in video S1. In all evaluations depicted, hybrid-($\mathrm{X}^{\mathrm{KS}}, \mathrm{u}^{\mathrm{KS}}$) was used as tracking method and no additional outlier filters were applied. The tracks are color-coded according to their predicted normalized velocity, while the ground truth is indicated by red tracks in the background. Using either algorithm, reliable particle tracks were predicted that cover the majority of the ground truth and provide consistent velocity estimates. Just one apparent erroneous track can be found near the center in Fig. 6b using pseudo-frames for detection.

Visible red tracks of the ground truth indicate particles that were not tracked by the framework over the full duration or with higher uncertainty in position. Qualitatively, this occurs more frequently in image-based evaluation than in direct processing of the event stream. When using AI-based processing with RVT, red tracks are particularly evident at the beginning and end of the tracks. This is caused by the finite time resolution of the event-tensors, which cover time intervals of 10 ms to preserve meaningful representations in the bins. In contrast, time intervals of only 4 ms were used in the evaluation based on pseudo-frames. In the enlarged views, tracks predicted based on pseudo-frames or RVT are visibly smoother than those processed directly. Besides a more straightforward hyperparameter search for tracking, this is mainly due to smoothing effects during detection. These reduce the dispersion of representative points used for tracking compared to processing the pure event stream.

To provide a quantitative comparison of the processing pipelines, the predicted tracks were matched to the tracks in the ground truth. For this, a track was declared correctly assigned if an Euclidean distance of less than 5 pixel to a track of the ground truth in the xy-plane was found at the respective time instance. Predicted tracks that exceed the median absolute velocity of all tracks by a factor of 2.5 were discarded. This primarily affects tracks that contain high velocity values due to overshooting of the fit functions. Subsequently, the median absolute error (MedAE) in position ($MedAE_{xy}$) and velocity ($MedAE_{u}$) was determined along all tracks and plotted in Fig. 7a with normalization by $|\hat{\boldsymbol{U}}|$. The color of the markers refers to the detection algorithm used, while the shape of the marker reflects the tracking algorithm. As the MedAE in velocity is plotted as a function of the MedAE in position, the most precise tracks are expected towards the lower left corner. Overall, subpixel-accurate tracking positions were achieved, while the $MedAE_{u}$ of the velocity remained below 10 % of $|\hat{\boldsymbol{U}}|$. However, both the detection and tracking algorithm significantly affect the resulting outcome. In the analyzed dataset and using conventional algorithms, the most accurate tracks were achieved by directly processing the event stream using DBSCAN or pixelwise extension, reaching $MedAE_{xy} < 0.4\,\mathrm{pixel}$ and $MedAE_{u}/|\hat{\boldsymbol{U}}| \approx 1.2\,\%$. When evaluated using pseudo-frames, the $MedAE_{xy}$ increases to approx. 0.68 pixel. Considering that only positive events were used, the $MedAE_{xy}$ could be further reduced by fine-tuning the parameter β to correct the offsets of the particle images. However, $MedAE_{u}$ still robustly reaches values below 2 % of $|\hat{\boldsymbol{U}}|$ in terms of velocity, even when using pseudo-frames in the evaluation.

The proposed tracking algorithms provide reliable predictions with an $MedAE_{u}/|\hat{\boldsymbol{U}}|$ below 3.5 %, with the most accurate results achieved using hybrid approaches. An exception is the tracking via Kalman filter, both for detection based on direct or image-based processing. This is due to manual adjustment of the hyperparameters $\boldsymbol{Q}$ and R, which remains challenging due to the high dimensionality. This clearly represents a drawback compared to fit functions or hybrid approaches, which allow for a broader range of hyperparameters. Automated optimization methods, such as Bayesian optimization schemes, significantly streamline fine-tuning. However, as the computational effort is increased simultaneously, the implementation of optimization methods is kept for future studies.

The 25th and 75th percentiles of the absolute error are shown as error bars for the best performing tracking algorithm for each detection algorithm. Across all algorithms, the unequal spread of the percentile ranges indicates strongly positively skewed distributions of absolute errors, consistent with a median skewness of 13.8 and 11.4 in position and velocity, respectively. Hence, most results are concentrated at small errors, while a few large errors produce a long right tail. Applying additional outlier criteria may therefore further reduce $MedAE_{xy}$ and $MedAE_u$.

However, AI-based algorithms (RVT, MvHeatDET) achieved metrics in a similar range of $MedAE_{xy} < 0.35$ pixel and $MedAE_u/|\hat{U}| \approx 2\,\%$. The most precise results among all algorithms were obtained by using MvHeatDET with $MedAE_{xy} = 0.11$ pixel and $MedAE_u/|\hat{U}| = 1.2\,\%$. The latter was fine-tuned on dataset EvDET200K (Wang et al. 2024), which contains rather small and fast moving objects such as table tennis balls. Conversely, the application of RVT was originally intended for larger objects such as cars or pedestrians. However, the high number of individual moving objects in the FOV remains challenging and deviates from conventional object detection tasks with fewer objects. Since all results were achieved without fine-tuning the hyperparameter of the neural networks for particle tracking, a potential for more precise tracks can be indicated. Hence, the use of AI-based processing constitutes a promising approach regarding the achievable precision and low inference time.

In particular, the computational effort in the different processing pipelines is decisive in event-based particle tracking. Required processing times for all algorithms are listed in Table S1 (SI) for a reference case. In summary, the implemented algorithms are not yet optimized regarding processing speed, but already reveal clear differences between the considered approaches. Among the conventional algorithms, kd-tree achieved the lowest processing time for detection and matching, making it nearly 40 % faster than the detection based on pseudo-frames. However, the AI-based detection using RVT further reduced the processing time by 81 %. Applying flatter architectures with less trainable parameters and tailored for particle detection might further reduce the processing time while maintaining precision. Moreover, the computational effort in the subsequent tracking step is significantly reduced when applying image-based or AI-based detection, as only particle positions at discrete time steps are used for tracking. Hence, the chosen processing pipeline influences not only the precision of the results, but also the required processing time. This trade-off is particularly relevant for future real-time applications of event-based PTV, while the results indicate high potential for AI-based detection.

Besides the precision in position and velocity, the relative proportion $\tilde{t}_{\mathrm{covered}} = T_{\mathrm{matched}}/T_{\mathrm{gt}}$ of the duration during which the ground truth tracks are covered by the predicted tracks T_{matched} to the total duration of the tracks in the ground truth T_{gt} is crucial. In Fig. 7b, $\tilde{t}_{\mathrm{covered}}$ is depicted as a function of the ratio of the number of predicted tracks N_{tracks} to the number of tracks in the ground truth N_{gt}. Hence, the optimal result is found at [1,1]. Conversely, the ratio $N_{\mathrm{tracks}}/N_{\mathrm{gt}}$ is an indication on the share of tracks that are interrupted or don't match to the ground truth. Hence, $N_{\mathrm{tracks}}/N_{\mathrm{gt}} > 1$ refers to a result with interrupted or unmatched tracks, while $N_{\mathrm{tracks}}/N_{\mathrm{gt}} < 1$ infers that ground truth tracks are not fully covered. In each combination of algorithms, more than 83 % of the total duration of the tracks in the ground truth was covered by the predicted tracks. The highest relative proportion $\tilde{t}_{\mathrm{covered}}$ among the conventional algorithms was achieved at 96 % by directly processing the event stream with pixelwise extension combined with tracking by Kalman filter or hybrid-($\mathrm{X^K}, \mathrm{u^{KS}}$). However, 1.8 times the number of tracks in the ground truth was predicted in this case, mainly due to interrupted tracks. This share is significantly reduced when using DBSCAN or AI-based processing, reaching 1.03 for RVT and 1.07 for MvHeatDET. The latter achieved a relative proportion $\tilde{t}_{\mathrm{covered}} = 97\,\%$, proving that the majority of tracks in the ground truth were covered without significant interruptions.

While the ratio $N_{\mathrm{tracks}}/N_{\mathrm{gt}}$ is mainly determined by the detection and matching algorithm, $\tilde{t}_{\mathrm{covered}}$ is significantly influenced by the selected tracking algorithm. Overall, fit functions and hybrid-($\mathrm{X^K}, \mathrm{u^S}$), in which the velocity of the particles is fitted directly on the event clusters, achieved the lowest values of $\tilde{t}_{\mathrm{covered}}$. This is caused by a possible overshoot in the predicted velocity, which leads to exclusion of the respective track. Conversely, the Kalman filter, hybrid-($\mathrm{X^K}, \mathrm{u^{KS}}$) and hybrid-($\mathrm{X^{KS}}, \mathrm{u^{KS}}$) deliver reliable tracks regardless of the detection algorithm used. However, the results were achieved using the synthetic dataset, whereas specific combinations of detection and tracking algorithms such as kdtree with function fitting can work more reliably in different scenarios, as demonstrated in the following sections.

4.2 Rotating chopper wheel

To extend the results gathered from synthetic data to experimental data, printed particle images were fixed on a rotating disc and recorded with an event-based camera, as described

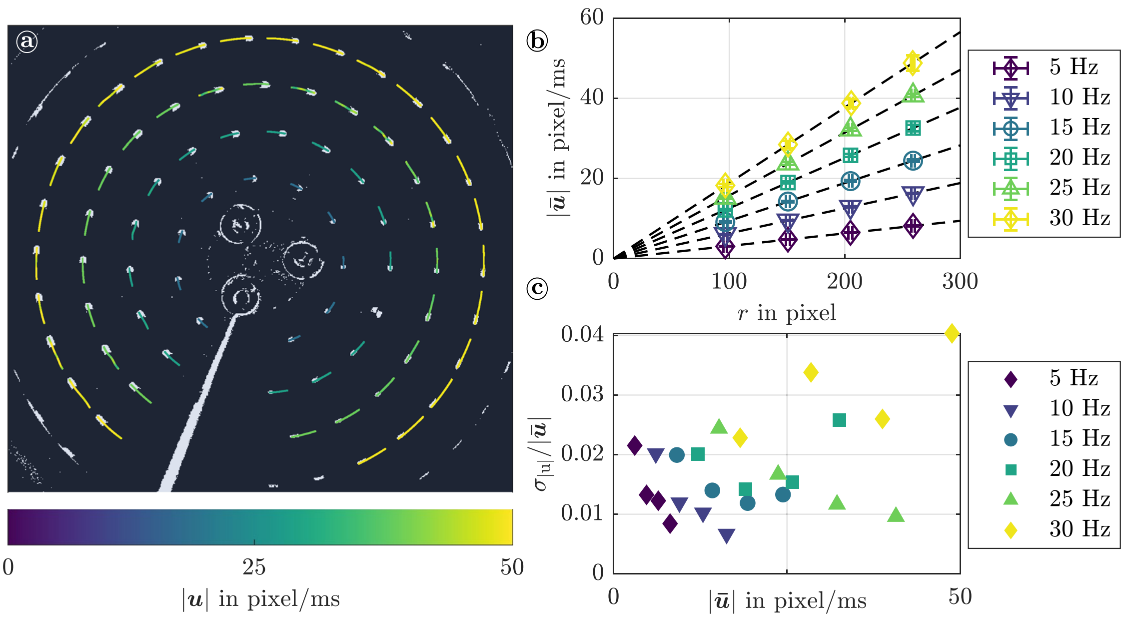
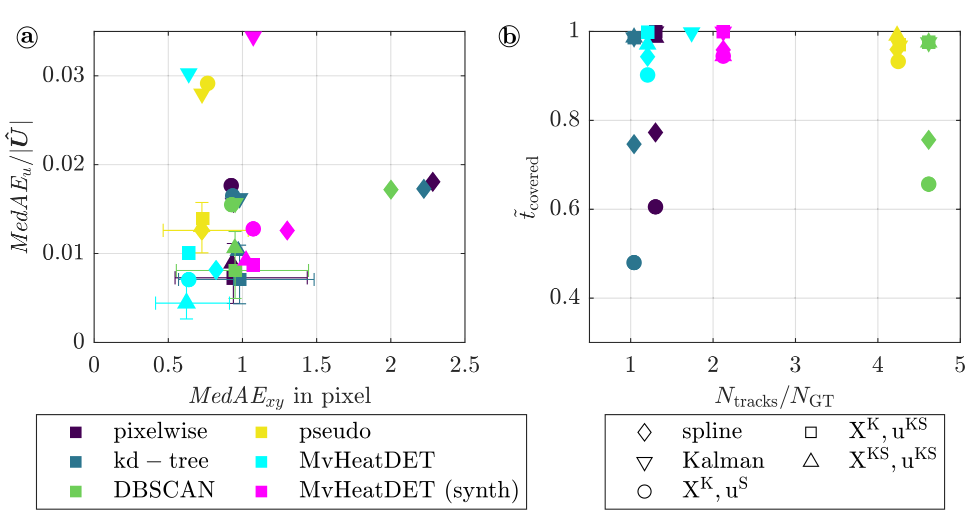


Fig. 8 Particle tracks predicted by direct processing depicted in front of a pseudo-frame of the disc (PI2) rotating at 30 Hz (**a**). In (**b**), the mean absolute velocity measured for various rotational speeds are shown against the analytical ground truth. The corresponding relative standard deviation is given in (**c**) against the mean absolute velocity

in Sect. 3.2. The given positions of the particle images and the rotational speed of the disc define the ground truth of the particle tracks. Although an eccentric mounting of the particle images was corrected in post-processing, minor deviations between the ground truth tracks and recordings may occur, which contrasts with the synthetic data. The dataset contains four disks with an increasing number of particle images (see Fig. S2), which were recorded at rotational speeds ranging from 2.5 Hz to 49.5 Hz. However, with increasing event rate and depending on the number of lines to be read out in the pixel array, i.e., due to the increasing number of particle images, readout errors occurred (see Fig. S3). This is a well-known challenge (Willert 2023; Cao et al. 2025; Maya et al. 2025), which manifests in uniform timestamps along individual lines and, in severe cases, irretrievably loss of events. As discussed in detail in Section V (SI), the dataset was restricted to sequences in which the average ratio of unique timestamps per line N_{unique} to the total number of events per line N_{total} exceeded 0.2.

In Fig. 8a, tracks predicted by direct processing (pixelwise extension) and the tracking algorithm hybrid-(X^{KS}, u^{KS}) are depicted for a period of 800 µs at a rotational speed of 30 Hz. Additionally, the motion of the predicted tracks is shown in video S2. In the evaluation, the radial line

Fig. 9 Relative MedAE in velocity $MedAE_u/|\hat{U}|$ against $MedAE_{xy}$ for different detection (indicated by the color) and tracking (indicated by the shape) algorithms (**a**), evaluated based on experimental data obtained by the rotating chopper wheel. The neural network MvHeatDET was either trained on synthetic (synth) or experimental data. The relative proportion $\tilde{t}_{\text{covered}}$ is depicted as a function of the ratio $N_{\text{tracks}}/N_{\text{GT}}$ for the same combination of algorithms in (**b**)



indicating the angular position as well as a particle image that appeared twice in the event stream due to a printing error were masked. Tracks with a predicted velocity greater than 2.5 times the median absolute velocity of all tracks were discarded. The results demonstrate a reliable detection and tracking of the particle images, even with decreasing intensity of the printed particle images in circumferential direction. The predicted absolute velocity is color-coded, indicating a radial increase of up to approx. 50 pixel/ms.

Since the angular velocity was kept constant in each measurement, the absolute velocity of the particle images scales linearly with the radius r. This correlation allows for validation of the mean absolute velocity $|\bar{\boldsymbol{u}}|$, as illustrated in Fig. 8b using dashed lines for analytical solutions at different rotational speeds. For each measurement, the four markers indicating the mean absolute velocity of all particle images at the respective radial position coincide with the dashed lines. The standard deviation in position and velocity is indicated by error bars. A quantitative comparison of the standard deviation achieved for the absolute velocity reflects a precise evaluation with $\sigma_{|\bar{\boldsymbol{u}}|}/|\bar{\boldsymbol{u}}| < 4\,\%$ (see Fig. 8c). Up to a rotational speed of 15 Hz, a generally decreasing relative standard deviation $\sigma_{|\bar{\boldsymbol{u}}|}/|\bar{\boldsymbol{u}}|$ can be observed at increasing absolute velocity. This is caused by the increasing reference value $|\bar{\boldsymbol{u}}|$ and the increasing number of particle images at increasing radius. However, the results deviate from this trend at rotational speeds above 20 Hz, as the relative standard deviation increases with r. As shown in Fig. S3, ratios $N_{\text{unique}}/N_{\text{total}} < 0.3$ were obtained in the respective measurements. The line-by-line unification of the timestamps thus affects the uncertainty of the predicted positions and velocities of the particle tracks and leads to increasing standard deviations. Hence, the usable event rate is restricted by hardware limitations of the event-based camera, depending on the number of lines to be read out of the pixel array. On top of that, less events are triggered for one particle as the velocity increases, leading to an increase in uncertainty.

In order to provide a quantitative evaluation without the influence of these hardware limitations, the $MedAE_u$ in Fig. 9a was plotted as a function of the $MedAE_{xy}$ for a rotation speed of 10 Hz ($|\hat{\boldsymbol{U}}| = 16.43$ pixel/ms). Based on conventional algorithms, an $MedAE_{xy}$ around 0.93 pixel was achieved, reaching subpixel values by accumulating the events into pseudo-frames. However, when using fit functions for tracking based on event clusters, $MedAE_{xy}$ significantly increases to approx. 2.2 pixel. This is due to the rapid temporal changes in the in-plane coordinates of the tracks, which follow

$$\begin{bmatrix} x_{\text{i}}(t) \\ y_{\text{i}}(t) \end{bmatrix} = \begin{bmatrix} x_0 + r_{\text{i}} \cos(\omega t + \phi_0) \\ y_0 + r_{\text{i}} \sin(\omega t + \phi_0) \end{bmatrix}, \tag{4}$$

using polar coordinates $[r, \phi]$. These rapid temporal changes restrict the suitable smoothing, as B-splines tend to cause increased uncertainty or overshooting when smoothing is excessive or insufficient, respectively. Compared to the

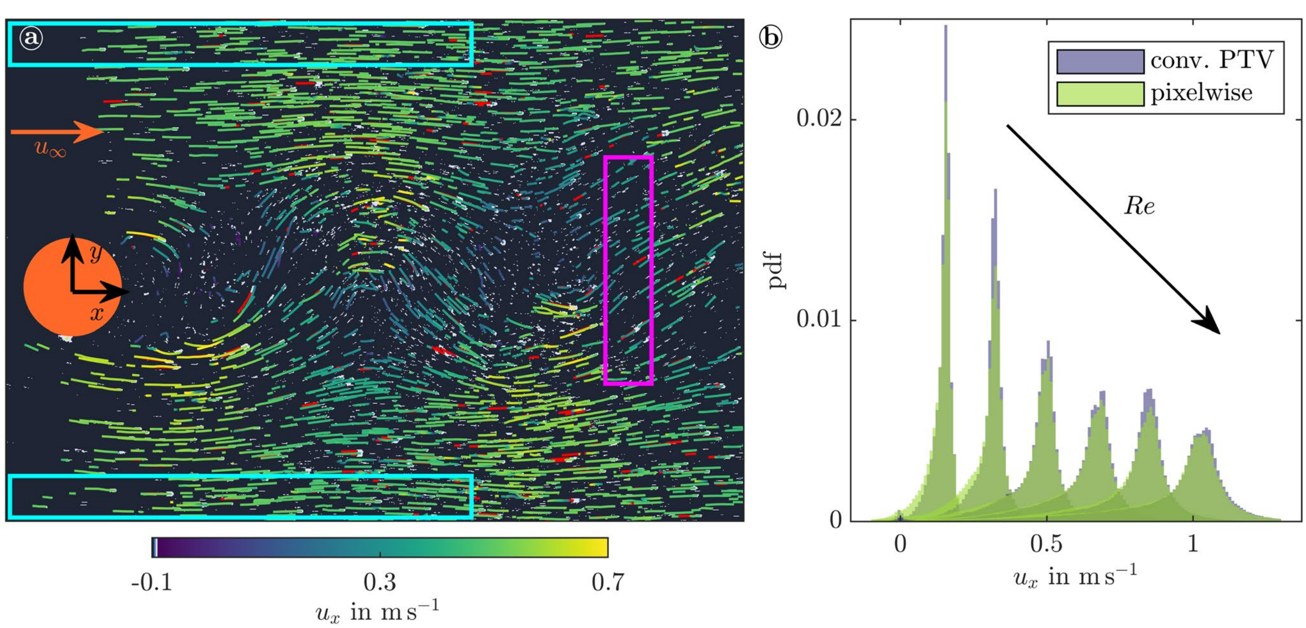


Fig. 10 Particle tracks predicted by using conventional PTV (red) or combining pixelwise extension and hybrid-(X^K, u^S) (colored) in the wake of a cylinder (**a**). The cylinder and inflow direction are highlighted in orange. Peripheral regions and a region in the wake of the cylinder used for evaluation are marked in cyan and pink, respectively. The pdf of the velocity component u_x measured in the whole FOV is depicted in (**b**) for different Reynolds numbers

results based on the synthetic data in Fig. 7a, $MedAE_{xy}$ is overall larger. This trend is attributed not only to the more challenging particle tracking task, but also to the remaining uncertainty of the ground truth. However, $MedAE_u$ remained consistently below 3.1 % of the maximum speed $|\hat{\boldsymbol{U}}|$ for all algorithms. The most precise results in terms of velocity were achieved by direct processing and hybrid-($\mathrm{X^K, u^{KS}}$) as tracking algorithm at $MedAE_u/|\hat{\boldsymbol{U}}| < 1\,\%$. Hence, the proposed algorithms demonstrated a reliable tracking of slow and fast moving particle images in the center and outer areas of the rotating disk, respectively. In turn, this reveals one fundamental advantage of event-based PTV, as every particle possesses its individual footprint in the event stream.

Based on the error bars indicating the 25th and 75th percentile for the best performing combination of detection and tracking algorithms, a similarly skewed distribution of the absolute errors as in the synthetic case is revealed. However, the median skewness in the absolute error of the track positions reduced to 1.27, indicating fewer predictions with large absolute errors occurring.

As the most precise results were obtained by using MvHeatDET in Sect. 4.1, the neural network was further applied to the experimental data. Although trained on synthetic data, the achieved $MedAE_{xy}$ and $MedAE_u/|\hat{\boldsymbol{U}}|$ fall in the range of the results obtained by direct processing of the event stream (see Fig. 9a). Hence, the neural network was able to generalize from synthetic to experimental data, greatly facilitating the application in real-world experiments. However, by further training the neural network on experimental data with known ground truth, both the lowest $MedAE_{xy}$ and $MedAE_u/|\hat{\boldsymbol{U}}|$ were achieved by combining MvHeatDET with hybrid-($\mathrm{X^{KS}, u^{KS}}$), reaching 0.62 pixel and 0.44 %, respectively. As pseudo-frames are used as input, the neural network benefited from the image-based detection in this particular dataset. Furthermore, the reliable detection based on synthetic and experimental data demonstrates the robust application for particle detection using event-based cameras. However, the enhanced performance of MvHeatDET when trained on experimental data revealed further potential for generalization. In particular, by including further variations in the optical settings and fluid flow applied in the synthetic data as well as deriving more realistic particle images, AI-assisted event-based particle tracking could be applied reliably also by non-expert users in a variety of experimental setups.

The relative proportion $\tilde{t}_{\mathrm{covered}}$ of the duration covered by the predicted tracks is depicted in Fig. 9b against the ratio $N_{\mathrm{tracks}}/N_{\mathrm{GT}}$. Analogous to the results discussed in Sect. 4.1, $\tilde{t}_{\mathrm{covered}}$ depends primarily on the tracking algorithm used. Based on the experimental data, reliable particle tracking with $\tilde{t}_{\mathrm{covered}} > 0.98$ was achieved by applying the Kalman filter, hybrid-($\mathrm{X^K, u^{KS}}$) or hybrid-($\mathrm{X^{KS}, u^{KS}}$). Significantly lower values of $\tilde{t}_{\mathrm{covered}}$ are obtained exclusively using fit functions or hybrid-($\mathrm{X^K, u^S}$). In these cases, B-splines were fitted directly to the event clusters, which leads to outliers due to potential overshooting in the predicted velocity as discussed above. Conversely, the ratio $N_{\mathrm{tracks}}/N_{\mathrm{GT}}$ depends primarily on the selected detection and matching algorithm. Direct processing with pixelwise extension or kd-tree as well as MvHeatDET demonstrated reliable particle tracking with a low proportion of interrupted tracks, achieving values of $N_{\mathrm{tracks}}/N_{\mathrm{GT}} < 1.31$. Interestingly, kd-tree performed worse in Sect. 4.1, highlighting the individual strengths of each algorithm depending on the dataset considered. In contrast, the results obtained by DBSCAN or image-based processing contain a larger proportion of interrupted tracks, resulting in a ratio $N_{\mathrm{tracks}}/N_{\mathrm{GT}} > 4.2$. Both approaches utilize a nearest neighbor algorithm to match tracks in the time-stepping scheme, which leads to reduced performance due to the highly varying velocities of the particle images within a sequence.

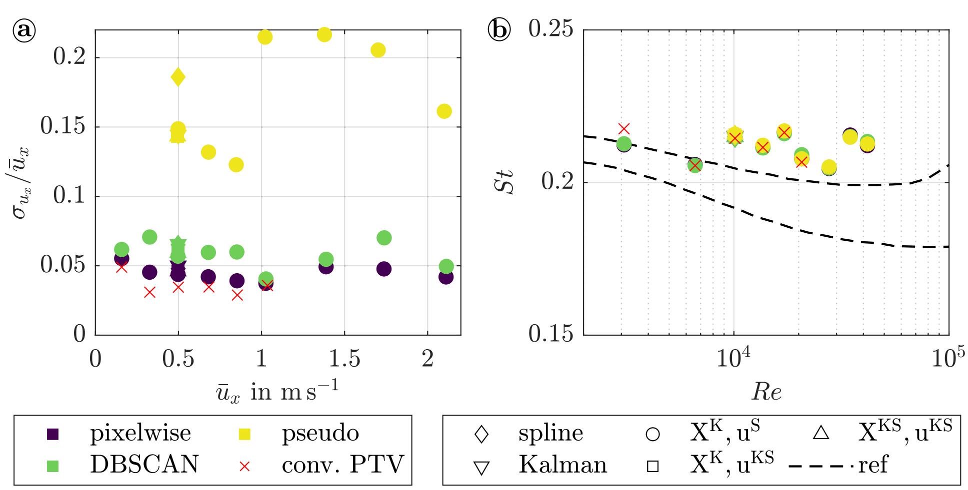


Fig. 11 Standard deviation of the velocity component u_x normalized by the median value $\bar{u}_x$ for different algorithms and Reynolds numbers analyzed in the peripheral regions indicated in cyan in Fig. 10a (**a**). The derived Strouhal number is given in (**b**) as a function of *Re*, using data obtained in the pink frame highlighted in Fig. 10a. Literature values are indicated by the dashed lines (Lienhard 1966)

4.3 Flow in the wake of a cylinder

The main objective of STELLA is the time-resolved tracking of multiple tiny particles in a fluid flow, which is demonstrated below based on the wake of a cylinder. To define a reference, the particles were illuminated using a continuous-wave laser and observed synchronously with an event-based and high-speed camera, as described in section 3.3. To evaluate the particle motion based on images captured by the high-speed camera, particle images were detected by segmentation using a global threshold in the intensity level and tracked using a two-frame predictor, as discussed in Cierpka et al. (2013). By varying the inflow velocity u_∞, a range of $Re \in [3.1, 20.6] \times 10^3$ was investigated. As this dataset is intended as a demanding benchmark, a rather high particle seeding concentration was set, resulting in an average of 926 active tracks found in the event stream, corresponding to an apparent particle image density of 10^{-3} particles per pixel at a mean event rate of about 17×10^6 events per second. However, as the exposure time of the high-speed camera was reduced with increasing *Re* to limit motion blur effects, the SNR decreased concurrently. As a result, only 543 active tracks were found on average based on conventional PTV. To reject spurious tracks detected in the event stream, two outlier criteria were applied. These criteria serve to reject particle tracks based on the quality of the respective track and the velocity distribution of neighboring tracks. A detailed explanation can be found in Section VI (SI).

In Fig. 10a, particle tracks found by direct processing of the event stream and conventional PTV based on recordings from the high-speed camera at $Re = 10.1 \times 10^3$ are overlaid on a pseudo-frame. While the tracks derived by conventional PTV are shown in red, the color of the tracks determined by direct processing indicates the velocity component u_x. The position of the cylinder is highlighted in orange. A visual representation of the derived particle tracks can also be found in videos S3 and S4 for different *Re*. Tracks derived from direct processing of the event stream largely cover the tracks found by using conventional PTV. In particular, the relative proportion $\tilde{t}_{\mathrm{covered}}$ of the duration covered by the predicted tracks amounts to 92.4 %. However, nearly twice as many tracks were predicted based on the event stream, which can primarily be attributed to regions of lower velocity in the wake of the cylinder, where fewer tracks were found with conventional PTV. This is due to the constant frame rate of the high-speed camera, which was set to ensure a reasonable displacement of the faster particles in the peripheral regions between subsequent frames to achieve a low uncertainty in the derived velocity. However, the wake flow inherently possesses a 3D nature due to periodically detaching vortices. Hence, particles may leave the light sheet in out-of-plane direction within two consecutive frames at the set frame rate without being tracked. Conversely, faster and slower particles leave individual footprints in the event stream, which enables simultaneous tracking in the peripheral regions and wake of the cylinder.

In Fig. 10b, the probability density function (pdf) of the velocity component u_x is depicted for $Re \in [3.1, 20.6] \times 10^3$ measured using the high-speed and event-based camera. The velocity distributions of u_x determined using either measurement technique largely overlap. Furthermore, peaks in the distributions correspond to similar velocities, which relate to the average velocity in the peripheral regions. Deviations only occur towards lower velocities that tail off more gradually when using the event-based camera, which is in line with the qualitative observation described above.

To compare event-based and conventional PTV, statistics on the derived velocity data in the peripheral regions highlighted in cyan in Fig. 10a are determined below. In Fig. 11a, the standard deviation σ_{u_x} of the velocity component u_x normalized by the median value $\bar{u}_x$ is depicted as a function of the median $\bar{u}_x$ for different algorithms. Since the flow is transient, random measurement noise as well as time-dependent vortex shedding affect the standard deviation σ_{u_x}. For evaluation, detection algorithms based on direct processing (pixelwise extension, DBSCAN) and pseudo-frames were used. AI-based evaluation was omitted, as the number of particles present in the FOV exceeds the maximum amount of predictions given by MvHeatDET. This clearly presents a drawback in the applicability of MvHeatDET to real-world experiments and underlines the demand for tailored architectures. Furthermore, no reliable ground truth is available for training the neural networks, emphasizing the need for generalization from synthetic to experimental data. Various tracking algorithms were compared for $Re = 10.1 \times 10^3$ ($\bar{u}_x \approx 0.5\,\mathrm{m/s}$). In this case, the chosen tracking algorithm influences the achievable standard deviation only marginally, while the detection algorithm affects σ_{u_x} significantly. The most similar result to conventional PTV was achieved by combining pixelwise extension with hybrid-$(\mathrm{X^K}, \mathrm{u^S})$, achieving a difference $\Delta\sigma_{u_x}/\bar{u}_x$ of 0.92 % relative to conventional PTV. Consequently, only the tracking algorithm hybrid-$(\mathrm{X^K}, \mathrm{u^S})$ was used to evaluate the recordings at different Reynolds numbers.

Across the entire range of $Re \in [3.1, 20.6] \times 10^3$, algorithms based on direct processing of the event stream achieved reliable results with differences of $\Delta\sigma_{u_x}/\bar{u}_x < 1.5\,\%$ compared to conventional PTV. Among the algorithms for direct processing, pixelwise extension showed the smallest differences. Due to the high dynamic range of the event-based camera, the measurement range was further extended to $Re = 41.6 \times 10^3$ ($\bar{u}_x \approx 2.1\,\mathrm{m/s}$). In this range, reliable particle tracking was not feasible using conventional PTV due to the decreasing SNR in the utilized setup. However, even beyond the measurement range of conventional PTV, a reliable tracking with $\sigma_{u_x}/\bar{u}_x < 7.1\,\%$ was achieved by direct processing of the event stream. In contrast, the obtained normalized standard deviation $\sigma_{u_x}/\bar{u}_x$ significantly increased when using pseudo-frames. This is caused by the nearest-neighbor algorithm used to match the detections between successive pseudo-frames. Due to the rather high particle image density, spurious matches cannot be avoided entirely, leading to erroneous tracks that limit the performance of this approach. However, these limits could be overcome by advanced techniques using multiframe prediction schemes, as demonstrated in previous work based on conventional PTV (Schröder and Schanz 2023).

The velocity data provided by STELLA enables reliable flow diagnostics, as demonstrated by the derived Strouhal number $St = fd_{\text{cylinder}}/u_\infty$ depicted as a function of Re in Fig. 11b. Here, the vortex shedding frequency f was determined roughly by fitting a periodic function to the velocity component u_y measured in the pink marked region of Fig. 10a along five periods. Across the entire measurement range, the derived Strouhal numbers scatter around a value of $St = 0.21$, averaging only about 4.2 % higher than reported in literature (Lienhard 1966), as indicated by dashed lines. Remaining differences to literature values might be caused by transient vortex shedding, which does not entirely average out when estimating the frequency f across five periods. However, there are only marginal differences between the results obtained by using the event stream and applying conventional PTV on the grayscale images recorded by the high-speed camera. Hence, the proposed algorithms proved an reliable and robust data evaluation, enabling flow diagnostics.

5 Conclusion

In this study, the modular framework STELLA for time-resolved tracking of numerous small particles using an event-based camera was introduced. For this, various algorithms were combined within STELLA, enabling direct processing of the event stream and processing in dense representations, e.g., using pseudo-frames. The performance of the algorithms and proposed processing pipelines was benchmarked against synthetic and experimental datasets. Since the datasets offer a rich ground truth, AI-based algorithms were trained and evaluated for particle tracking. The performance of the algorithms can be summarized as follows:

- Both algorithms based on direct processing and dense representations demonstrated subpixel-accurate tracking of multiple particles in synthetic datasets. The lowest MedAE in terms of position and velocity was achieved at 0.36 pixel and 1.2 % of the peak velocity through direct processing, while up to 95 % of the duration of tracks in the ground truth was covered by predicted tracks.
- Robust and reliable flow diagnostics were demonstrated in demanding experimental datasets, even at high particle seeding concentrations and an average of 926 active tracks. Using event-based PTV, fluid flows with significant velocity gradients were reliably measured with constant illumination, overcoming challenges related to fixed frame rates and decreasing SNR with decreasing exposure time in conventional PTV.
- The most effective combination of detection and tracking algorithms across different processing pipelines could not be identified universally and depends largely on the data under consideration. In general, direct processing approaches delivered reliable results across a wide range of applications. However, processing via pseudo-frames proved beneficial regarding spatial accuracy in experiments with periodic particle tracks at high frequency.
- AI-based processing pipelines achieved promising results on synthetic data, comparable to the most effective algorithms for direct processing. Considering the intended tracking of fewer and larger objects such as cars or pedestrians in current architectures, great potential lies in fine-tuning of hyperparameters and tailored architectures.

The proposed framework STELLA offers a openly accessible, flexible and versatile platform to evaluate complex particle trajectories. As approaches from different processing pipelines are integrated, STELLA allows for a flexible customization to the fluid flow at hand. In addition, the provided datasets constitute a demanding benchmark, which not only enables the training of neural networks but allows for a transparent comparison of emerging algorithms for particle tracking with event-based cameras. In future versions of STELLA, an extension to three dimensional particle tracking by utilizing several camera views as well as deriving particle acceleration will be implemented. Moreover, a possible generalization of neuronal networks trained on synthetic data to an inference on experimental data was demonstrated in this study. However, further enhancing the capabilities for generalization to demanding and versatile experimental data paves the way for subsequent studies.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00348-026-04253-8.

Acknowledgements The authors thank the Federal Ministry of Research, Technology and Space (BMFTR), a ministry of the Federal Government of Germany, for financial support within the project "TrackOpt" (01IS24074C) and the German Research Foundation (DFG) for funding under grant number 577000111. Furthermore, we thank Alexander Thieme and Georg Schulze for technical support and expertise in the experiments, as well as Maryam Al Zaben, who supported us in training of the neural networks. The fruitful scientific discussion and helpful insights into diffusion-based generative models with Prof. Michael Chertkov are greatly appreciated.

Author Contributions S.S.: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization. S.J.: Methodology, Software, Validation, Investigation, Writing - Review & Editing. M.K.: Software, Validation, Investigation, Data Curation. M.K.: Resources, Writing - Review & Editing, Supervision, Project administration, Funding acquisition. C.W.: Software, Writing - Review & Editing, Supervision. C.C.: Conceptualization, Resources, Writing - Review & Editing, Supervision, Project administration, Funding acquisition.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability Synthetic and experimental datasets generated within this study will made publicly available after publication.

Declarations

Conflict of interest There are no conflict of interest to declare.

Ethics approval Not applicable

References

AlSattam O, Mongin M, Grose M, Gunasekaran S, Hirakawa K (2024) Kf-pev: a causal kalman filter-based particle event velocimetry. Experiments in Fluids **65**(9)

Amjoud AB, Amrouch M (2023) Object detection using deep learning, cnns and vision transformers: A review. IEEE Access 11:35479–35516

Bentley JL (1975) Multidimensional binary search trees used for associative searching. Commun ACM 18(9):509–517

Borer D, Delbruck T, Rösgen T (2017) Three-dimensional particle tracking velocimetry using dynamic vision sensors. Exp Fluids 58(12):165

Cao J, Zeng X, Li S, He C, Wen X, Liu Y (2025) A novel event-based ensemble particle tracking velocimetry for single-pixel turbulence statistics. Exp Thermal Fluid Sci 169:111554

Cao J, Zeng X, Lyu Z, Cai W, Liu H, Liu Y (2025) Benchmark evaluation of event-based imaging velocimetry using digital micromirror device. Experiments in Fluids **66**(4)

Cierpka C, Lütke B, Kähler CJ (2013) Higher order multi-frame particle tracking velocimetry. Exp Fluids 54(5):5

Cordone L, Miramond B, Thierion P (2022) Object detection with spiking neural networks on automotive event data. International Joint Conference on Neural Networks (IJCNN), Padua, Italy. http://arxiv.org/pdf/2205.04339v1

Coutinho G, Moita A, Moreira A, Rossi M (2025) Defocustrackerai: Adaptive and automatic detection of defocused particle images. Proceedings of the 16th International Symposium on Particle Image Velocimetry

Drazen D, Lichtsteiner P, Häfliger P, Delbrück T, Jensen A (2011) Toward real-time particle tracking using an event-based dynamic vision sensor. Exp Fluids 51(5):1465–1469

Ester M, Kriegel H-P, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. 2nd International Conference on Knowledge Discovery and Data Mining. pp 226–231

Franceschelli L, Willert CE, Raiola M, Discetti S (2025) An assessment of event-based imaging velocimetry for efficient estimation of low-dimensional coordinates in turbulent flows. Exp Thermal Fluid Sci 164:111425

Fre P, Sorin AS (2015) Classification of arnold-beltrami flows and their hidden symmetries. Phys Part Nuclei 46:497–632

Gehrig D, Scaramuzza D (2024) Low-latency automotive vision with event cameras. Nature 629(8014):1034–1040

Gehrig M, Scaramuzza D (2023) Recurrent Vision Transformers for Object Detection with Event Cameras. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

Godbersen P, Schanz D, Schröder A (2024) Peak-cnn: improved particle image localization using single-stage cnns. Exp Fluids 65(10):153

Gunasekaran S, Mongin M, AlSattam OA, Hirakawa K (2025) Comparison of event-based alogrithms for experimental two-dimensional, two-component velocimetry. Journal of Aircraft 0, 1–15

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

Howell J, Hammarton TC, Altmann Y, Jimenez M (2020) High-speed particle detection and tracking in microfluidic devices using event-based sensing. Lab Chip 20(16):3024–3035

Julier SJ, Uhlmann JK (1997) A new extension of the kalman filter to nonlinear systems. Signal Processing Sensor Fusion and Target Recognition VI 3068:182–193

Kalman RE (1960) A new approach to linear filtering and prediction problems. J Basic Eng 82(1):35–45

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

Kähler CJ, Scharnowski S, Cierpka C (2012) On the resolution limit of digital particle image velocimetry. Exp Fluids 52(6):1629–1639

König J, Chen M, Rösing W, Boho D, Mäder P, Cierpka C (2020) On the use of a cascaded convolutional neural network for three-dimensional flow measurements using astigmatic PTV. Meas Sci Technol 31(7):074015

Lienhard JH (1966) Synopsis of Lift, Drag, and Vortex Frequency Data for Rigid Circular Cylinders. Pullman, Washington

Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft COCO: Common Objects in Context. 13th European Conf. on Computer Vision (ECCV). Zurich,

Litzenberger M, Posch C, Bauer D, Belbachir AN, Schon P, Kohn B, Garn H (2006) Embedded vision system for real-time object tracking using an asynchronous transient vision sensor. In: 2006 IEEE 12th Digital Signal Processing Workshop & 4th IEEE Signal Processing Education Workshop, pp. 173–178. IEEE, ??? (24.09.2006 - 24.09.2006)

Mahowald MA, Mead C (1991) The silicon retina. Sci Am 264(5):76–83

Maneewongvatana S, Mount DM (1999) Analysis of approximate nearest neighbor searching with clustered point sets. arXiv. p 9901013
Maya L, Fan L, Durocher A, Savard B, Vena P (2025) An evaluation of event-based cameras for particle image velocimetry. Exp Fluids 66(11):1–22
Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66
Ratz M, Sachs S, König J, Cierpka C (2023) A deep neural network architecture for reliable 3D position and size determination for Lagrangian particle tracking using a single camera. Meas Sci Technol 34(10):105203
Rebecq H, Gehrig D, Scaramuzza D (2018) ESIM: an Open Event Camera Simulator. Conference on Robotics Learning (CoRL). Zürich,
Reed AW, Hashemi C, Melamed D, Menon N, Hirakawa K, McCloskey S (2025) EBS-EKF: Accurate and High Frequency Event-based Star Tracking. arXiv, 2503–20101
Rusch A, Rösgen T (2023) Trackaer: real-time event-based quantitative flow visualization. Exp Fluids 64(8):136
Sachs S, Ratz M, Mäder P, König J, Cierpka C (2023) Particle detection and size recognition based on defocused particle images: a comparison of a deterministic algorithm and a deep neural network. Exp Fluids 64(21):21
Schröder A, Schanz D (2023) 3d lagrangian particle tracking in fluid mechanics. Annu Rev Fluid Mech 55(1):511–540
Schuman CD, Kulkarni SR, Parsa M, Mitchell JP, Date P, Kay B (2022) Opportunities for neuromorphic computing algorithms and applications. Nature Comput Sci 2(1):10–19
Su Q, Chou Y, Hu Y, Li J, Mei S, Zhang Z, Li G (2023) Deep Directly-Trained Spiking Neural Networks for Object Detection (21.07.2023). http://arxiv.org/pdf/2307.11411v3
Tayarani-Najaran M-H, Schmuker M (2021) Event-based sensing and signal processing in the visual, auditory, and olfactory domain: A review. Frontiers in neural circuits 15:610446
Wang Y, Idoughi R, Heidrich W (2020) Stereo Event-Based Particle Tracking Velocimetry for 3D Fluid Flow Reconstruction. In: Vedaldi A, Bischof H, Brox T, Frahm J-M (eds) Computer Vision - ECCV 2020. Lecture Notes in Computer Science. Springer International Publishing, Cham
Wang S, Wang Z, Li C, Qi X, So HK-H (2025) Spikemot: Event-based multi-object tracking with sparse motion features. IEEE Access 13:214–230
Wang H, Nie Y, Li Y, Liu H, Liu M, Cheng W, Wang Y (2024) Research, Applications and Prospects of Event-Based Pedestrian Detection: A Survey (05.07.2024). http://arxiv.org/pdf/2407.04277v1
Wang X, Jin Y, Wu W, Zhang W, Zhu L, Jiang B, Tian Y (2024) Object Detection using Event Camera: A MoE Heat Conduction based Detector and A New Benchmark Dataset. arXiv. pp 2412–06647
Willert CE (2023) Event-based imaging velocimetry using pulsed illumination. Exp Fluids 64(5):98
Willert C (2025) Event-based particle image velocimetry for high-speed flows. Meas Sci Technol 36(7):075302
Willert C, Franceschelli L, Amico E, Raiola M, Cafiero G, Discetti S (2026) Real-time event-based particle image velocimetry for active flow control. J Phys: Conf Ser 3173(1):012001
Willert CE, Klinner J (2022) Event-based imaging velocimetry: an assessment of event-based cameras for the measurement of fluid flows. Exp Fluids 63(6):101
Yu L, Chen H, Wang Z, Zhan S, Shao J, Liu Q, Xu S (2025) Spiking-ViT: A Multiscale Spiking Vision Transformer Model for Event-Based Object Detection. IEEE Transact Cognitive Develop Syst 17(1):130–146
Zeng X, Cao J, Lyu Z, He C, Cai W, Liu Y (2025) A temporally adaptive particle tracking velocimetry using continuous-wave illumination for fused event- and frame-based cameras. J Visualization 28:463
Zeng X, Lyu Z, Cao J, He C, Liu Y (2025b) Super-time-resolution particle tracking velocimetry via the fusion of event- and frame-based cameras. Meas Sci Technol 36(6):065302
Zubic N, Gehrig D, Gehrig M, Scaramuzza D (2023) From chaos comes order: Ordering event representations for object recognition and detection. International Conference on Computer Vision (ICCV)