



KF-PEV: a causal Kalman filter-based particle event velocimetry

Osama AISattam^{1,3} · Michael Mongin² · Mitchell Grose¹ · Sidaard Gunasekaran² · Keigo Hirakawa¹

Received: 22 April 2024 / Revised: 13 July 2024 / Accepted: 23 August 2024
© The Author(s) 2024

Abstract

Event-based pixel sensors asynchronously report changes in log-intensity in microsecond-order resolution. Its exceptional speed, cost effectiveness, and sparse event stream make it an attractive imaging modality for particle tracking velocimetry. In this work, we propose a causal Kalman filter-based particle event velocimetry (KF-PEV). Using the Kalman filter model to track the events generated by the particles seeded in the flow medium, KF-PEV yields the linear least squares estimate of the particle track velocities corresponding to the flow vector field. KF-PEV processes events in a computationally efficient and streaming manner (i.e., causal and iteratively updating). Our simulation-based benchmarking study with synthetic particle event data confirms that the proposed KF-PEV outperforms the conventional frame-based particle image/tracking velocimetry as well as the state-of-the-art event-based particle velocimetry methods. In a real-world water tunnel event-based sensor data experiment performed on what we believe to be the widest field view ever reported, KF-PEV accurately predicted the expected flow field of the SD7003 wing, including details such as the lower velocity in the wake and the flow separation around the underside of an angled wing.

1 Introduction

Understanding the characteristics of the flow around objects plays an important role during the development of energy efficient and structured robust aerodynamic designs. Owing to recent technological advancements, state-of-the-art Particle Image Velocimetry (PIV), Particle Tracking Velocimetry

(PTV), and Optical Flow (OF) systems have achieved exceptional capabilities to estimate the flow velocity using complex setups and cutting-edge hardware. The classical two-dimensional two-component (2D-2C) setups remain a powerful and popular tool for analyzing two-dimensional complex flow phenomena due to its robust results, experimental simplicity, and validated algorithms (Scharnowski and Kähler 2020; Geisler and Schr 2014) (Fig. 1).

To estimate a flow velocity using the double exposure 2D-2C Frame-Based (FB) setup, a single high-speed camera is used to observe the seeded particles moving within the flow medium, such as a gas or liquid, while crossing the illuminated field of view (FOV). By capturing the movement of the seeding particles and estimate its velocity, the flow velocity can be estimated relatively using non-intrusive approach. After recording two consecutive frames, the PIV is performed by estimating the average movement of the particles within a spatial window (sub-interrogation areas) using cross-correlation to find the best spatial window match across the two frames (Thielicke and Sonntag 2021). Alternatively, the PTV estimates the velocity of the individual particles by tracking the trajectory of a particle between two frames using particle matching techniques (Janke et al. 2020). OF is another approach that leverages photometric gradient values to infer the underlying flow field without explicit tracking (Liu and Shen 2008; Liu et al. 2015).

✉ Osama AISattam
aasattam@ju.edu.sa

Michael Mongin
monginm1@dayton.edu

Mitchell Grose
grosem1@dayton.edu

Sidaard Gunasekaran
gunasekarans1@dayton.edu

Keigo Hirakawa
khirakawa1@dayton.edu

¹ Department of Electrical and Computer Engineering, University of Dayton, 300 College Park, Dayton, OH 45469, USA

² Department of Mechanical and Aerospace Engineering, University of Dayton, 300 College Park, Dayton, OH 45469, USA

³ Department of Electrical Engineering, Jouf University, Airport Rd, Sakaka, Jouf 72388, Saudi Arabia

Even though the PIV is more commonly used for its robustness against the noise and require less fancier hardware, the drawbacks for all methods include difficulty in analyzing slow and fast motions simultaneously, limits on frame rate/temporal resolution, high data throughput, high computational complexity, and high camera hardware costs associated with high-speed imaging (Scharnowski and Kähler 2020; Qureshi et al. 2020). In addition, the experimental hardware cost and complexity increase as the desired accuracy increases, as well as the computational and storage costs stemming from large data volume.

Thanks to the state-of-the-art event-based cameras boasting microsecond-order resolution, it possible to observe a fast-moving phenomenon with a significant reduction in the data throughput without sacrificing the temporal resolution. The reduction comes from the fact that the event-based (EB) sensor generates events only when a change of intensity occur for each pixel individually, unlike the FB systems that capture full resolution frames at a given frame rate, as illustrated in Fig. 2. Several event-based particle velocimetry feasibility studies have been conducted in recent years (Drazen et al. 2011; Wang et al. 2020; Willert and Klinner 2022). In our most recent work (AlSattam et al. 2024), we proposed the Particle Event Velocimetry (PEV), a noise-robust particle-level event-based estimate of the flow field based on robust flow field analysis. PEV performs the analysis through pre-processing filters, robust tracking, line fitting, and parametric velocity vector estimation. Even though the performance analysis showed that PEV in AlSattam et al. (2024) achieves state-of-the-art accuracy compared to EB (Drazen et al. 2011; Wang et al. 2020; Willert and Klinner 2022) and FB (Thielicke and Sonntag 2021; Janke et al. 2020) methods, however, it is unsuitable for real-time applications due to its non-causal design.

The goal of this present work is to develop a causal event-based particle tracking and velocity estimation in streaming format leveraging the Kalman filter operating on an event streams from event-based camera sensor. Specifically, the proposed Kalman filter-based particle event velocimetry (KF-PEV) efficiently solves linear least squares estimate of particle velocity, thanks to the iterative streaming format processing (i.e., update velocity estimates as new event is observed) designed to update the state variables for each incoming event. In our benchmarking, Kalman Filter-based PEV is shown to outperform state-of-the-art FB and EB particle velocimetry methods using the synthetic event data generated by the event-based camera simulator¹ for particle velocimetry that we developed. We also demonstrate the

KF-PEV operating in real-world water tunnel experiments involving actual event-based sensor data.

2 Background and related work

Image sensor technology has improved significantly since the time dynamic vision sensing (event-based sensing) was introduced to mimic the function of the human retina Mahowald (1992). Year 2008 marked first commercially available event-based cameras by Inivation (Lichtsteiner and Posch 2008), and Prophesee and Sony recently co-developed backside illuminated megapixel sensor in 2021. Thanks to its low power, high dynamic range, and high temporal resolution, event-based image sensors have the potential to positively impact many applications (Gallego et al. 2020; Finateu et al. 2020; Ni et al. 2012; Benosman et al. 2012; Rueckauer and Delbruck 2016; Brosch et al. 2015; Rebecq et al. 2018).

Unlike the conventional framing cameras, each pixel on the event camera operates independently and asynchronously. Instead of synchronously capturing the accumulation of photons to yield intensity frames, each pixel reports an *event* only if there is a change in intensity with time resolution of microseconds. It produces a positive event if the intensity increased within a certain threshold and a negative one if it was otherwise. Thus, the event camera will generate asynchronous data that include the spatial location of the event ($e.x, e.y$), the time stamp $e.t$, and the polarity of the event $e.p$. Unlike a framing camera, the event camera theoretically will not produce any event unless there is change in intensity that exceeds the threshold. Thus it is particularly advantageous for observing sparse high speed phenomenon (such as particle velocimetry), with savings in data throughput and computational complexity. See illustration in Fig. 2.

Events generated from asynchronous pixels require a different kind of processing than the cross-correlation and photometric approaches used in the conventional FB PIV/PTV/OF setup. Although some techniques can be borrowed with some modifications (like tracking algorithms), we develop below event-based processing techniques that makes full considerations for the asynchronous nature of the event streams.

The utilization of event-based sensors in particle velocimetry application is a developing field of research. The first proof of concept for event-based particle velocity was provided in Drazen et al. (2011). This concept was generalized to the stereo event-based particle tracking velocimetry based on 2D optical flow Wang et al. (2020). To improve robustness and overcome the uncertainty of optical flow-based tracking and matching, the output of this system is an average velocity per space-time volume using two event cameras. Tracking in Drazen and Wang can be implemented

¹ Initially reported in AlSattam et al. (2024).

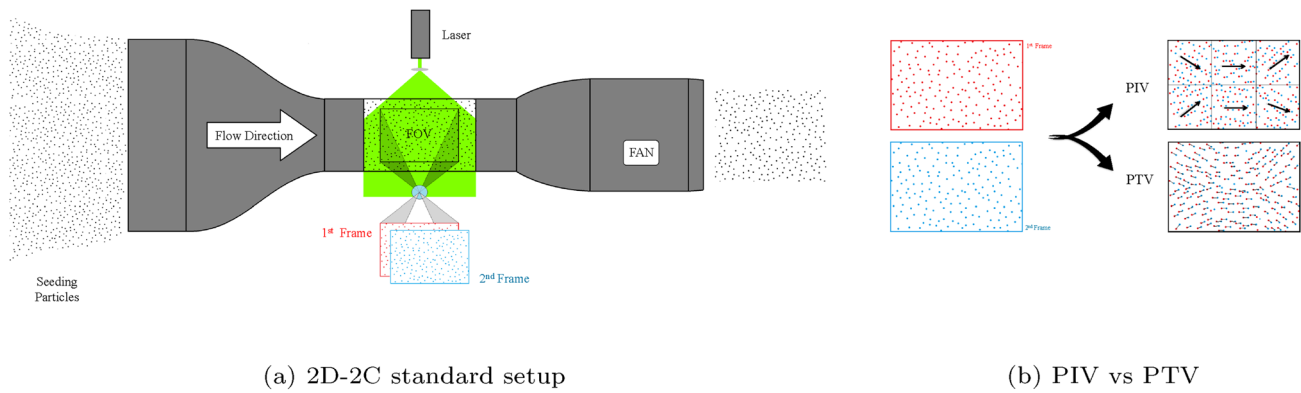


Fig. 1 Classical frame-based PIV/PTV

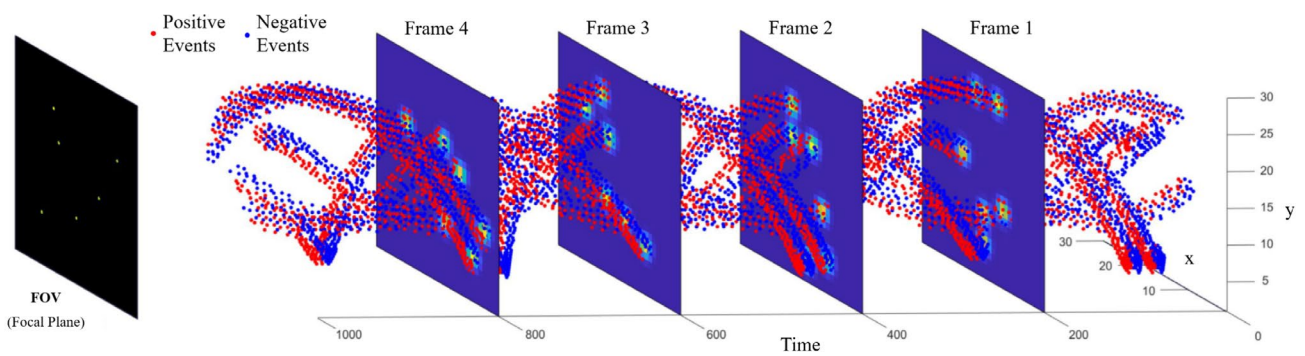


Fig. 2 Illustration of frame-based and event-based cameras. Particles are moving within the field of view (FOV) of the focal plane. The scatter plot represents the reported events caused by change in inten-

sity, which is continuously sampled in the time domain. The intensity frames are sampled synchronously, resulting in the loss of motion details between the frames

efficiently in streaming format. A three dimensional PTV method in Rusch and Roesgen (2021); Rusch and Roesgen (2023) used a 2D event-based particle detection on multiple cameras. The multi-camera track correspondence, along with triangulation enables three dimensional flow field reconstruction.

In Willert and Klinner (2022), the velocity of the fluid in 2D was estimated by finding the average velocity per voxel using maximum contrast as an alternative to the conventional cross-correlation-based PIV. Although averaging of vector velocity was shown to improve robustness for a dataset with very large integration time (10ms), the quality would deteriorate in presence of strong gradients in flow field.

We recently proposed the Particle Event Velocimetry (PEV) (AlSattam et al. 2024), aimed at improving the event-based velocity estimation in terms of noise robustness, the spatial-temporal particle trajectory, and the tolerance to very high particle density. This is accomplished by exploiting

specific characteristics of event-based sensors to overcome uncertainties in the event stream, integrating inceptive event filtering (Baldwin et al. 2019), and by improving the particle trajectory by nonlinear function fitting over the estimated tracks. However, this method is non-causal, and therefore cannot be implemented in real-time manner.

Our work shares some similarities with event-based PTV method by Borer et al. (2017) which uses baseline Kalman filtering to track the helium-filled soap bubbles. However, the seeding density used in this study is low (10 bubbles/sec) compared to the high seeding rate of standard PIV (8-10 particles per 32×32 interrogation window). The experimental setup is even more challenging for wind or water tunnels where the camera operates under low light conditions to track laser-illuminated sub-millimeter particles. Thus further modifications are needed to improve its performance in these demanding experiments. The increased computational cost of Kalman filtering may also be challenging for real-time operation (Rusch and Roesgen 2023).

3 Kalman filter-based particle event velocimetry (KF-PEV)

To implement a causal event-based particle velocimetry system, it is essential to use a streaming format processing that provide an optimal solution for the velocity estimation of each track in the presence of uncertainty. The Kalman filter (Kalman 1960) operates iteratively by adjusting its state variable estimates based on the evolving information. The most common application of Kalman filtering is in robust real-time tracking, where the recursion yields linear least square positional estimates by minimizing the effects of measurement noise and errors (Grewal and Andrews 2014; Brookner 1998). Although it is possible to differentiate the estimated Kalman tracks to recover the target velocity (e.g. to reconstruct the flow field in particle velocimetry), numerical differentiation has high risk of amplifying errors, unfortunately.

Instead, we propose to use the Kalman filter to estimate object velocity *directly* (as opposed to differentiating positional estimates) as a linear least square estimate of the states, admitting optimal high-quality flow field reconstruction in particle velocimetry. The proposed Kalman filter-based particle event velocimetry (KF-PEV) is summarized in the flow chart in Fig. 3. We detail the steps in KF-PEV below, such as the modifications made to the Kalman filter to handle asynchronous temporal sampling of event-based sensor and multiple tracks with unknown number of particles from unlabeled events. We initially develop a single particle event-based Kalman filter model, then generalize this model to handle multiple tracks simultaneously.

3.1 Single particle event-based Kalman filter model

To formulate a Kalman filter model suitable for the EB particle velocimetry system, it is essential to consider the nature

of the observation parameters provided by the sensor. For our proposed task, the event camera generates data includes $(e.x, e.y)$ pixel coordinate in addition to the time stamp $e.t$ and polarity $e.p$. The goal, however, is to estimate the individual particle location, velocity, and acceleration over time. Thus we define the Kalman state vector $\mathbf{X} \in \mathbb{R}^6$ with covariance matrix $\mathbf{P} \in \mathbb{R}^{6 \times 6}$ as follows:

$$\mathbf{X} = \left[x \quad \frac{dx}{dt} \quad \frac{d^2x}{dt^2} \quad y \quad \frac{dy}{dt} \quad \frac{d^2y}{dt^2} \right]^T = \left[x \quad \dot{x} \quad \ddot{x} \quad y \quad \dot{y} \quad \ddot{y} \right]^T, \quad (1)$$

where (x, y) are the current location of the track, (\dot{x}, \dot{y}) is the track velocity in x and y directions, and (\ddot{x}, \ddot{y}) represent the track acceleration. Consider the following state space model:

$$\mathbf{E}_k = \mathbf{H}\mathbf{X}_k + \mathbf{v} \quad (2)$$

$$\mathbf{X}_{k+1} = \mathbf{F}_k\mathbf{X}_k + \mathbf{w}. \quad (3)$$

Here, (2) is known as the measurement process yielding noisy positional observation $\mathbf{E}_k = (e_k.x, e_k.y)^T \in \mathbb{R}^2$, where $\mathbf{H} \in \mathbb{R}^{2 \times 6}$ and $\mathbf{v} \in \mathbb{R}^2$ with covariance matrix $\mathbf{R} \in \mathbb{R}^{2 \times 2}$ are observation matrix and the measurement noise receptively:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{v} \sim \mathcal{N}(0, \mathbf{R}). \quad (4)$$

On the other hand, the linear stochastic difference equation in (3) is regulated by the state transition matrix $\mathbf{F} \in \mathbb{R}^{6 \times 6}$ and the process noise $\mathbf{w} \in \mathbb{R}^6$ with with covariance matrix $\mathbf{Q} \in \mathbb{R}^{6 \times 6}$:

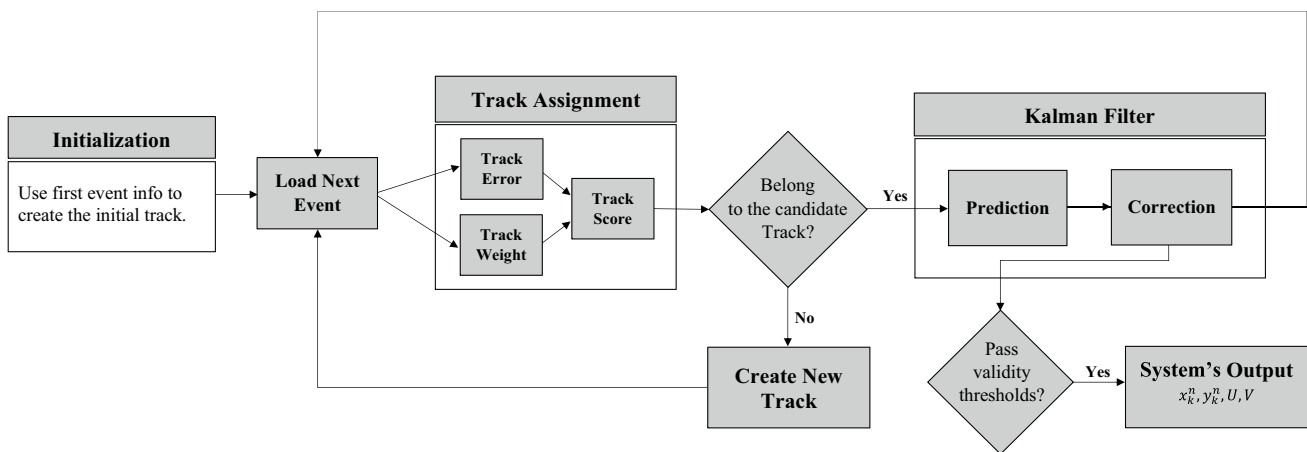


Fig. 3 Overview flowchart of the proposed KF-PEV algorithm

$$F_k = \begin{bmatrix} 1 & \Delta_k & \frac{\Delta_k^2}{2} & 0 & 0 & 0 \\ 0 & 1 & \Delta_k & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta_k & \frac{\Delta_k^2}{2} \\ 0 & 0 & 0 & 0 & 1 & \Delta_k \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, w \sim \mathcal{N}(0, Q). \tag{5}$$

In this work, we regard observation E_k as the pixel coordinate of the k th event location. Due to the asynchronous nature of event-based sensor, the events are not reported in an uniform temporal sampling rate. Therefore, the state transition matrix F_k must be adjusted every time based on the time interval $\Delta_k = e_{k,t} - e_{k-1,t}$ between the new event and the last reported one. Thus state space model in (2) and (3) is *nonstationary*—contrast this to the conventional *stationary* state space model with constant state transition matrix F .

The Kalman filter is a sequential processing yielding a linear least squares estimate of the state vector X_k based on the previous state estimate X_{k-1} and a new event E_k . We define three steps: initialization, prediction, and correction. In our work, the initialization begins by recording the pixel location for $k=0$:

1. Initialization: Model Parameters H, R, Q, P_0
 Initial Estimates $X_0 = [e_{0,x} \ 0 \ 0 \ e_{0,y} \ 0 \ 0]^T$
 Track Timestamp Update $k = k + 1$. (6)

Before processing the subsequent event $E_k (k = 1, 2, \dots)$, we predict the state vector and covariance as follows:

2. Prediction: Compute State Transition Matrix F_k where $\Delta_k = e_{k,t} - e_{k-1,t}$
 Predict State Vector $\hat{X}_k = F_k X_{k-1}$
 Predict Covariance $\hat{P}_k = F_k P_{k-1} F_k^T + Q$. (7)

where X_{k-1} and P_{k-1} are the previous state vector and covariance estimates, respectively. Then, we update the predicted state vector \hat{X}_k and covariance \hat{P}_k using the new event E_k in the following correction step:

3. Correction: Optimal Gain $K_k = \hat{P}_k H^T (H \hat{P}_k H^T + R)^{-1}$
 Correct State Vector $X_k = \hat{X}_k + K_k (E_k - H \hat{X}_k)$
 Correct Covariance $P_k = (1 - K_k H) \hat{P}_k$
 Track Timestamp Update $k = k + 1$. (8)

Repeating the steps two and three for each generated event E_k (i.e., process in streaming format), the state vector estimates X_k updates not only the current position of the tracked particle (x, y) , but also the corresponding velocity (\dot{x}, \dot{y}) . Note that the accuracy of the measurements will increase as the number of the analyzed data increases Brookner (1998).

3.2 Track assignment for multiple particles

We generalize the event-based Kalman filtering developed in Sect. 3.1 to handle multiple particles with multiple tracks. Denote by the superscript n the state space model for the n th track (e.g., X^n, F^n, P^n , etc.). Then each new event $E_k = (e_{k,x}, e_{k,y})^T$ must be assigned to a particle track it belongs to, prior updating the Kalman filter parameters. This *track assignment* task is performed by a heuristical *track score* aimed at balancing the quality of the track fit and the reliability of the existing tracks.

Define the track error as the Euclidean distance between the track center (x^n, y^n) of the n th particle and the new event pixel coordinate $E_k = (e_{k,x}, e_{k,y})^T$:

$$\text{TrackError}\{n, E_k\} = \sqrt{(x^n - e_{k,x})^2 + (y^n - e_{k,y})^2}. \tag{9}$$

TrackError assesses the quality of the track assignment. Owing to the microsecond-order resolution of event-based sensor, we expect acceptable track assignment to result in a very small TrackError, even for a very fast particle.

As noted above, the accuracy of the Kalman filter improves as the number of the observations increases. The TrackError in (9) computed with track centers (x^n, y^n) with very few observations is less reliable and sensitive to noise. This increases the risk of an erroneous track assignment whenever TrackError has a low value because of inaccurate track center location. To overcome this issue, the notion of TrackWeight is introduced to penalize the TrackError when the number of events in a track is low:

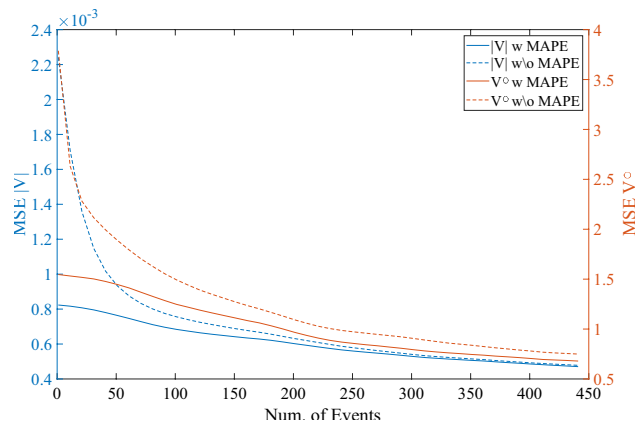


Fig. 4 MSE of velocity magnitude $|V|$ and angle V^o , as a function of the number of observed events in a track. MAPE thresholding in (13) significantly improves the accuracy of short tracks with low number of events. By setting a low rate of change threshold, the accuracy of the KF-PEV measurements improved at lower number of events. In addition, the longer tracks that were already stable had low MAPE score and therefore were unaffected

$$\text{TrackWeight}\{n\} = 1 + \frac{\min\left(\{\#\text{ of events in the } n\text{th track}\}, \lambda\right)}{\lambda}, \tag{10}$$

where λ is a user parameter specifying the upper limit to number of events. Notice that this weight is restricted to the range between 1 and 2, where the higher weights indicate more reliable TrackError.

We combine the TrackError and the TrackWeight to arrive at the final TrackScore, as follows:

$$\begin{aligned} \text{TrackScore}\{n, E_k\} \\ = \text{TrackWeight}\{n\} \cdot \left(1 - \frac{(\text{TrackError}\{n, E_k\})^2}{r^2}\right), \end{aligned} \tag{11}$$

where r is a user-specified radius threshold indicating the acceptable TrackError for accepting the track assignment. Intuitively, this score is designed to yield a negative value for $\text{TrackError} < r$, and smaller TrackError yields smaller TrackScore. The TrackWeight corresponding to a track with many observations increases the sensitivity to the TrackError, reflecting the increased confidence in the TrackError score. The track assignment for the event E_k is made by maximizing the TrackScore:

$$\hat{n} = \arg \max_n \text{TrackScore}\{n, E_k\}. \tag{12}$$

3.3 Rejection criteria

While the Kalman filter model is known for its effectiveness in handling noisy measurements, we implemented multiple rejection criteria to ensure that only accurate track assignments are accepted. First, the TrackScore in (11) is positive only if the TrackError is smaller than r . If the maximum score in (12) is positive, then the track assignment is accepted. The event E_k will be used to predict and update the assigned track according to the Kalman filter rules described in (7) and (8). On the other hand, suppose the maximum score in (12) is negative. The new event is deemed not sufficiently close to existing tracks (i.e., TrackError exceeds r), and therefore the track assignment is rejected. Instead a new track is formed, initialized by the event E_k according to the Kalman filter initialization described in (6).

Second, a moving particle enters and exits the sheet of laser while being measured. For this reason, the particle track is only valid for a short duration of the time. If a track has not seen a new event assignment for more than τ seconds, then it will be labeled as *inactive* track and will be excluded in the track assignment considerations for the subsequent events.

Finally, the Kalman filter estimates of velocity vector (\dot{x}, \dot{y}) improves with the number of observed events. Consequently, only tracks that surpass the specified length threshold are deemed acceptable. In addition, when two or more tracks intersect, errors can be introduced to the track

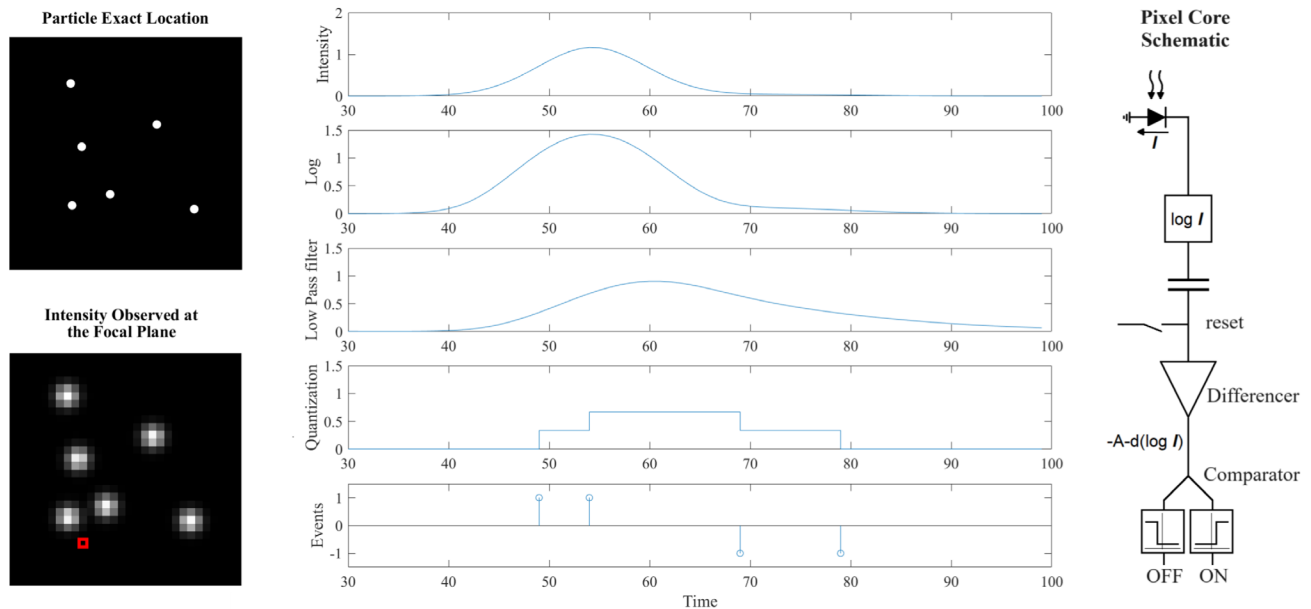


Fig. 5 Steps to generating synthetic event-based data for moving Gaussian particles, mimicking the analog signal path of the pixel readout circuitry. Although the steps taken are similar to the other event-based sensor simulators, the source image (moving particles)

are modeled in continuous time domain instead of frames. Plots showing temporal evolution of the pixel under investigation (red box). We also generate the corresponding intensity frames to test on frame-based particle velocimetry

assignment such that the state estimates would experience rapid changes. If the convergence of the state estimates suddenly changed, it can be flagged and labeled as unstable states. Therefore, only measurements from tracks that exceeds certain length and state stability are valid and can be used for velocity estimation. We use *mean absolute percentage error* (MAPE) to monitor the stability of state estimates, which is defined as follow:

$$MAPE_k^n = \frac{1}{6} \sum_{i=1}^6 \frac{|X_k^n(i) - X_{k-1}^n(i)|}{|X_k^n(i)|}, \tag{13}$$

where i indexes all 6 parameters within the Kalman state parameter $X_k^n \in \mathbb{R}^6$ in (1). By setting a low rate of change threshold, the accuracy of the KF-PEV measurements improved at lower number of events, as shown in Fig. 4. The longer tracks that were already stable had low MAPE score and therefore were unaffected.

4 Benchmarking study

We benchmark the accuracy of the proposed KF-PEV against the prior art in FB and EB methods objectively and subjectively. For objective benchmarking, we developed an event simulator for event-based PIV/PTV applications. The handful of event-based camera simulators available today (Rebecq et al. 2018; Shah et al. 2017) are frame based and not well suited for unresolved targets that resembles moving point sources—generated events that are inconsistent with the physical movement of particles. We instead model a continuous movement of the point source-like features of seed particles explicitly, along with the optical transfer function and circuit bandwidth in the event-based sensor to yield event streams in a continuous time space based on the flow field. Because the ground-truth flow field is known in simulation, the flow field reconstruction error can be objectively quantified.

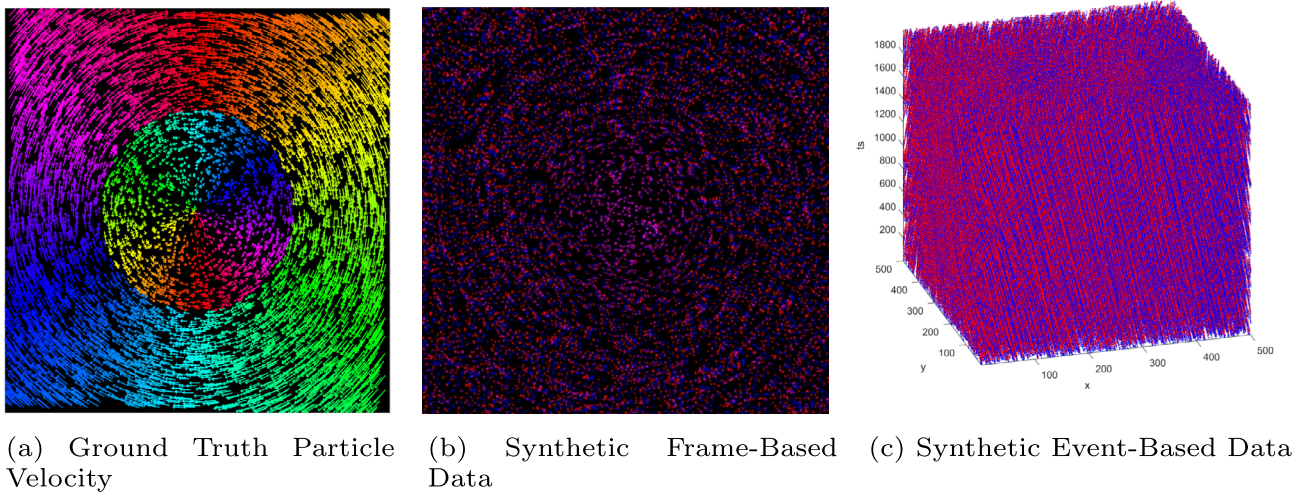


Fig. 6 Ground truth velocity vectors of the synthesized data. The velocity direction is encoded as the color hue

Table 1 Technical details of the synthetic particle velocimetry modeling

Model specs	Description	
Movement Type	Forced vortex with opposite rotation	
Duration	2	(mSec)
Particle Density	0.03	(PPP)
Linear Velocity Range	0 to 50K	(Pixel/Sec)
Angular Velocity	0.012	(Degree/uSec)
Particle Size	2 to 4	(Pixel)
Model Sensor Size	501 × 501	(Pixel)
Intensity Based output	2000 Frame	(1 frame/uSec)
Event Based Output	2 mSec of streaming event	(x, y, ts, p)

4.1 Simulation experiment setup

Our synthetic model is designed to produce continuous time event streams as well as FB images to be benchmarked against conventional PTV and PIV methods. We derive the changes of intensity of traversing particles at the pixel level in analytical form, and combine it with the pixel simulator modeling the optical transfer function and circuit bandwidth in the event-based sensor in Lichtsteiner and Posch (2008) from observing the intensity changes at the photodiode until reporting the event when occurred.

Specifically, we model each particle as a small two dimensional Gaussian function. Temporally, the intensity profile at a particular pixel of a particle traveling across the detector is also Gaussian function, whose width and height are determined by the proximity of the pixel to the particle path. The event based pixel observing this would apply a log function followed by a one pole lowpass filter to model the internal analog signal path. The events are then generated at level crossings (contrast thresholds). See Fig. 5.

Since the Gaussian function and one pole lowpass filters are relatively simple, the overall signal path inside a camera can be derived analytically. Thus the analytically simulated Gaussian-lowpass signal can be used to determine the exact timing of the level crossings with arbitrary timing accuracy. Unlike the “video-to-event” event simulators that are inherently frame-based simulation of events (Rebecq et al. 2018; Shah et al. 2017), analytical generation of events do not suffer from interpolation artifacts, erroneous optical flow calculations, etc. that are particularly severe for unresolved targets like seed particles.

Same technique can be used to generate a synthetic frame. Because the intensity profile of a particle at a particular pixel is modeled as Gaussian, we can calculate the exact integration over the exposure time without any temporal interpolation that would fail for unresolved targets such as the seed particles. Examples are shown in Fig. 5. Due to the analytical nature of the simulation, there is no theoretical limit to the frame rate or the exposure time (other than the computational resource required to render the simulated events/frames).

The user can specify the simulation experiment setup, such as particle density, the particle size, and flow field; as well as camera parameters such as contrast threshold and one pole lowpass filter in event-based cameras and exposure time and frame rate for frame-based cameras. In our simulation experiment, we chose the particle density and the exposure time to be similar to what is currently used in conventional FB PIV. The forced vortex with two direction model is chosen as the flow field, since it provides a wide range of velocities with a boundary where two opposite flow direction. The direction of movement is set to be clockwise for all particles located in the center and less than 250-pixel radius while the outer particles are moving in the counter clock manner as demonstrated in Fig. 6. Table 1 shows the technical details of the simulation model used in our experiment.

For benchmarking, the performance of particle velocimetry is evaluated objectively by computing the mean square error (MSE) of the estimated velocity components $\hat{V} = (\hat{x}, \hat{y})$ relative to $V = (x, y)$ (both component-wise and magnitude), as well as the angle error computed using the Spectral Angel Mapper (SAM) as described in eq 14:

$$\text{Angle}(\hat{V}, V) = \cos^{-1} \frac{\hat{V}^T V}{\|\hat{V}\| \|V\|}. \quad (14)$$

4.2 Benchmarking

We compare the proposed KF-PEV to the state-of-the-art frame-based (FB) and event-based (EB) PIT/PTV methods. For the FB-PIV implementation, we used the latest version of PIVlab (particle image velocimetry (PIV) tool with GUI Version 2.62.0.0) by Thielicke and Sonntag (2021). The FB-PTV benchmarking used Part2Track (A MATLAB package for double frame and time resolved Particle Tracking Velocimetry), which is an open-source system published in 2020 (Janke et al. 2020). The parameters in FB-PIV and FB-PTV play an important role to achieve the best measurements. For example, larger window size contributes to low spatial resolution while reducing the window size increases

Table 2 Result of the parameter analysis in Appendix, showing the ranges of parameters we tested and the optimal configuration that yielded best performance for our simulation

Method	Parameter	Tested range	Best performance
Event Based	Drazen et al. (2011) Mass weight	0 to 80	.3
	Wang et al. (2020) Propagation weight	0 to .9	0.4
	Willert and Klinner (2022) Window size	8-80 pixel	24 pixel
	Interrogation time	100 to 2000 μSec	1000 μSec
Frame Based	Thielicke and Sonntag (2021) Window size	16-24-32 pixel	16 pixel
	Time between frames	1 to 150 μSec	100 μSec
	Janke et al. (2020) Time between frames	1 to 250 μSec	50 μSec

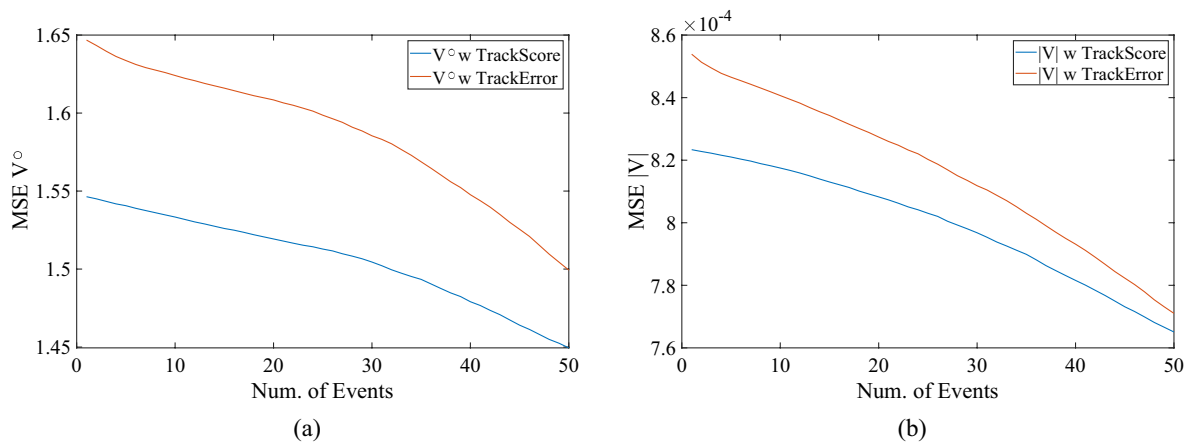


Fig. 7 Ablation study comparing tracking performance using TrackError in (9) and TrackScore in (11)

the cross-correlation error; and the interrogation time must be tuned to achieve the highest temporal accuracy with the minimum relative uncertainty (Adrian 1997; Raffel et al. 2018). In the benchmarking study, we iterated over ranges of each parameters to find the best performing parameter choices. (See Appendix for the parameter analysis.) Thus what we report for the competing algorithms is practically the peak performance of the competing algorithms (Fig. 14).

Similarly, we compare the proposed KF-PEV to state-of-the-art EB-PIV (Willert and Klinner 2022) and EB-PTV methods (Drazen et al. 2011; AlSattam et al. 2024; Wang et al. 2020). All reported parameters were used (e.g., propagation weight for Wang et al. (2020), mass ratio for Drazen et al. (2011), and the voxel sizes (time and space for Willert and Klinner (2022)). Other parameters were optimized experimentally by parameter sweeps, as reported in Table 2. (See Appendix for the parameter analysis.)

4.3 Simulation results

We report the results of the simulation study in Table 3 and Fig. 8. The proposed KF-PEV yielded a very stable flow field reconstructions with highest accuracy among all methods compared. Flow field is dense, while the boundaries are very cleanly preserved. As illustrated in Fig. 4, the accuracy of the velocity estimates indeed improves with an increased number of observations per track, consistent with the principles of Kalman filter theory. Slow moving particles inherently yield fewer events, however, and therefore it is difficult to reject the tracks by thresholding the number of observations only. Instead, thresholding based on MAPE was more effective for improving the accuracy for shorter tracks. Thus we empirically found that the n th track should be rejected if the following conditions are satisfied:

$$\{MAPE_k^n > \mu\} \cap \{\# \text{ of events in the } n\text{th track} < \tau\}. \quad (15)$$

Other benchmarking results in Table 3 and Fig. 8 are produced with optimized parameters (described in Appendix). Objectively, the proposed KF-PEV has the lowest MSE in (V_x, V_y) and angle error scores among all methods compared in Table 3. The number of tracks was the highest among the event-based PTV, albeit fewer than FB-PTV method (Janke et al. 2020). The number of tracks can be increased by adjusting the rejection thresholds (μ, τ) in (15) to exceed method in Janke et al. (2020), but at the risk of including more outliers. FB methods (Janke et al. 2020; Thielicke and Sonntag 2021) in addition to EB-PIV (Willert and Klinner 2022) show moderate to high mean squared error (MSE) values. By comparison, EB-PTV methods (Wang et al. 2020; Drazen et al. 2011; AlSattam et al. 2024) improved accuracy. Among these, AlSattam et al. (2024) stand out with the lowest magnitude MSE. The performance of the proposed KF-PEV was best among EB methods in every performance metric except for the velocity magnitude, which was very close second to AlSattam et al. (2024). FB-PIV method in Thielicke and Sonntag (2021) outperformed FB-PTV method (Janke et al. 2020) in objective metrics, but the situation may reverse with lower particle density (which would be disadvantageous for PIV and advantageous for PTV).

Qualitatively, Fig. 8 shows that the proposed KF-PEV has the densest track estimation among all the EB methods. Particles exhibiting low velocities near the center are detected within the anticipated flow magnitude and direction, as are particles with higher velocities located around the edge. FB-PIV method (Thielicke and Sonntag 2021) and EB-PIV method (Willert and Klinner 2022) appear to produce smooth flow field (despite the coarse resolution yielding better performance; supported by parameter analysis in Appendix). EB-PTV method in Wang et al. (2020) did not yield enough tracks; methods (Drazen et al. 2011; AlSattam

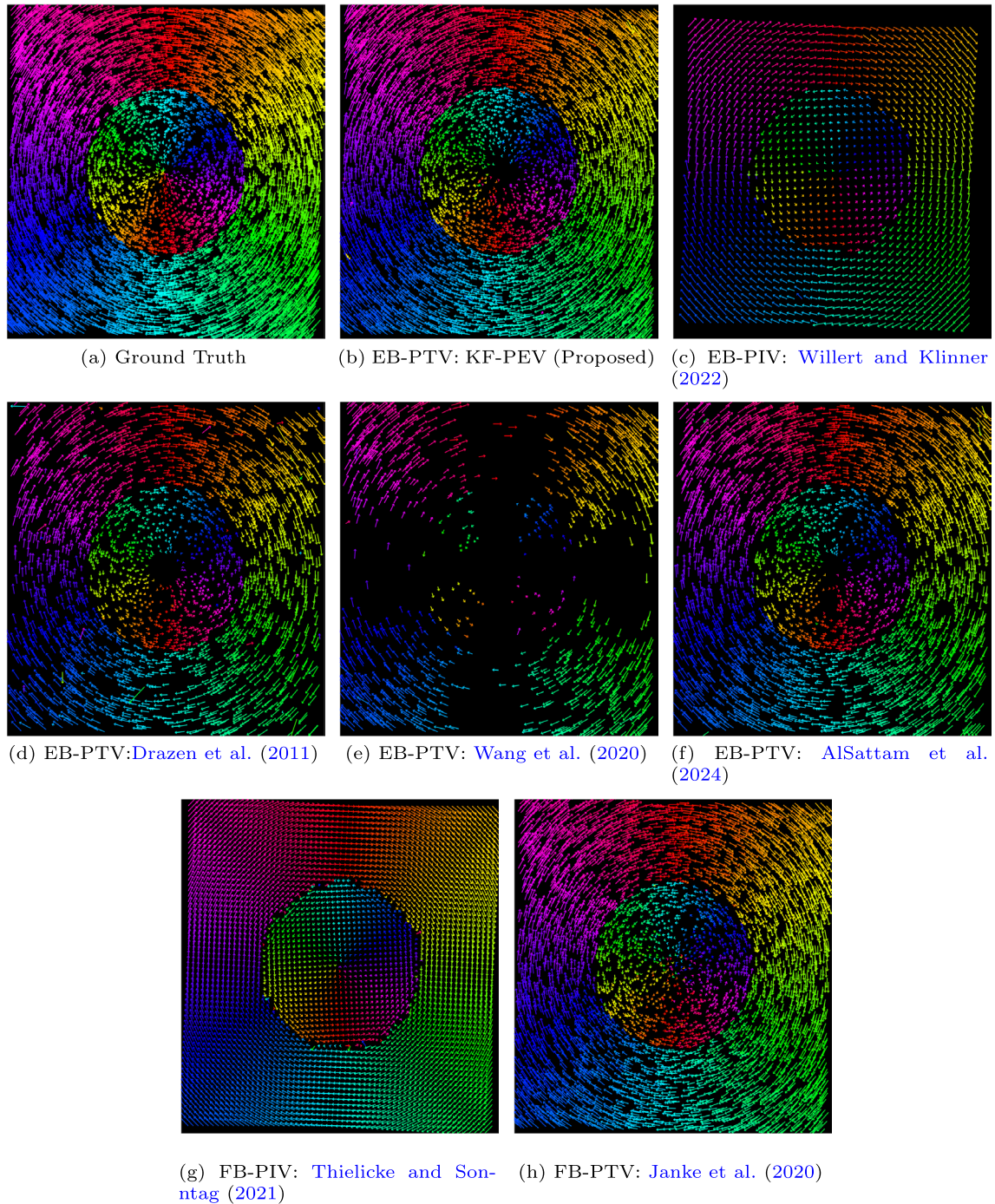


Fig. 8 Estimated particle velocity vectors of the synthetic particle data computed by EB and FB methods

et al. 2024) yielded satisfactory results, though not as dense as the proposed KF-PEV. On the other hand, the FB-PTV method (Janke et al. 2020) reconstructed the flow field with fewer outliers than the proposed KF-PEV. However, closer examination reveals that the velocity vector angles seemed to be *quantized* (as indicated by the non-smooth hues in Fig. 8(f); Table 3 also shows poor SAM angle performance).

Finally, the risk of incorrect track assignment is increased when the seeding density is high. We performed an ablation study comparing the tracking assignment using TrackError in (9) instead of TrackScore in (11). Figure 7 indeed confirms that Kalman filtering with TrackError alone is inferior, confirming that TrackWeight in (10) plays a key role in track assignment when the track event count is low.

Table 3 Benchmarking result using the simulated event-based pixel data, comparing the proposed KF-PEV with frame-based and event-based prior art. The "bold" formatting is used to emphasize the best performance in each column, while the "underlined>" formatting denotes the second-best performance

Method	PIV/PTV	MSE			SAM Angle	# of tracks	
		Vx	Vy	Magnitude			
Frame Based	Janke et al. (2020)	PTV	2.486E-03	2.449E-03	2.384E-03	5.24	4243
	Thielicke and Sonntag (2021)	PIV	1.490E-03	1.413E-03	1.278E-03	4.44	N/A
Event Based	Drazen et al. (2011)	PTV	9.885E-04	1.039E-03	8.906E-04	1.86	2540
	Wang et al. (2020)	PTV	9.335E-04	9.530E-04	6.069E-04	1.48	862
	Willert and Klinner (2022)	PIV	2.961E-03	3.015E-03	2.192E-03	8.72	N/A
	AlSattam et al. (2024)	PTV	<u>6.440E-04</u>	<u>6.778E-04</u>	5.451E-04	1.30	3107
KF-PEV(Proposed)	PTV	6.342E-04	6.173E-04	5.721E-04	1.05	3931	

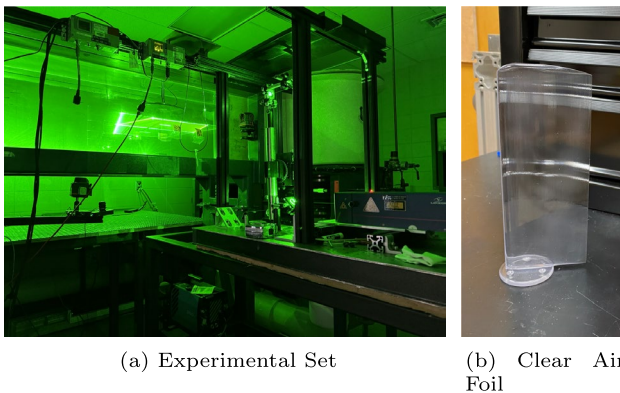


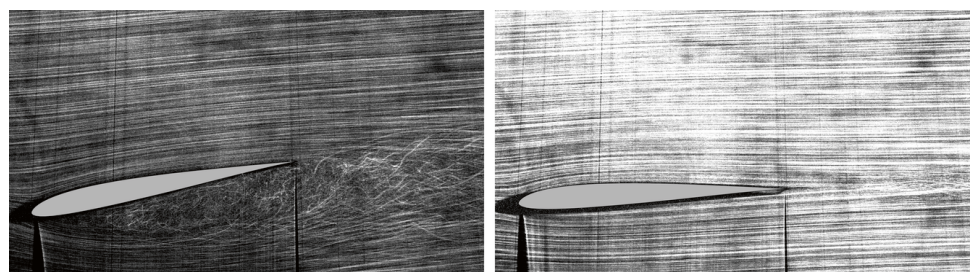
Fig. 9 Experimental setup. **a** University of Dayton Water Tunnel facility and **b** the SD7003 3D printed clear airfoil

5 Real event-based sensor data experiments

5.1 Real experiment setup

In addition to the simulation experiments, the proposed KF-PEV was tested in a real-world environment. The particle event velocimetry experiments with the event-based camera was conducted at the University of Dayton Water Tunnel, shown in Fig. 9(a). The facility is a horizontal free-surface water tunnel with a 6:1 contraction and a 38 cm wide and 46 cm high test section of 152 cm length. The free-surface speed range is 0.075 m/s to 0.38 m/s. For this study, the freestream velocity of the tunnel was set to 0.28 m/s. The tunnel is equipped with a one degree of freedom motion

Fig. 10 Visualization of events observed around the SD7003 wing fixed at **a** negative eight-degree and **b** zero-degree angles of attack. The event time is encoded by a gray scale (gray = past, black = recent)



stage consisting of a H2W Technologies linear motor driven by a Xenus XTL-230-18 motor driver from Copley Controls. This motor is outfitted with a rack and pinon gear to control the angle of attack of the airfoil position. An SD7003 wing, as shown in Fig. 9(b), was 3D printed using a transparent material that allows the laser sheet to pass through and illuminate both sides. The event camera was positioned below the test section focusing on a thin laser plane illuminating the trailing edge of a rotating cylinder with the trailing plate to capture the vortex locking in the wake.

For event-based camera, Prophesee EVK2 camera with 50 mm lens was used, yielding a field of view (FOV) of 18x10cm. To the best of authors' knowledge, this is the widest FOV used in EB PIV/PTV experiments reported in the literature thus far. The main difference between the standard 2C2D PIV setup with the conventional PEV system is that the event camera requires a continuous light laser source to keep track of the particle at 1 MHz resolution.

5.2 Real event-based sensor data experiment result

The SD7003 wing experiment was carried out with fixed 0° and -8° angles of attack as shown by the visualization in Fig. 10. Note that both ends of the airfoil have two apparent gaps at the tips, stemming from the laser refraction when passing through the transparent airfoil. Similarly, there are areas surrounding the SD7003 where the particles are absent due to the parallax effect of the lens. These are real-world constraints on the experiment configurations, and do not represent a limitation to the event-based particle velocimetry.

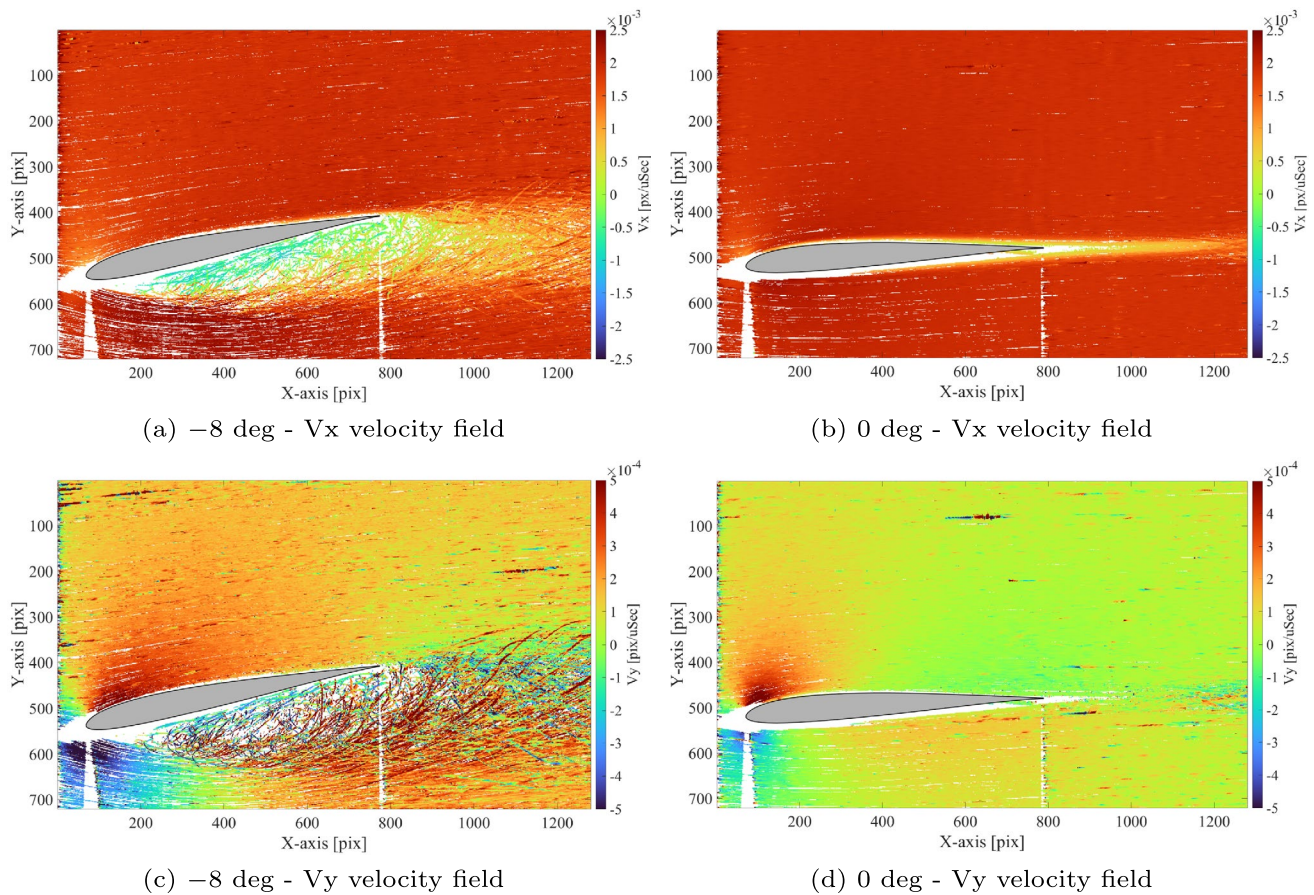


Fig. 11 Velocity vector field of the SD7003 wing estimated by KF-PEV. **a, b** Velocity V_x in X direction; **c, d** Velocity V_y in Y direction

Given the airfoil's stationary position throughout data collection, we accumulate tracks over time, resulting in a very dense flow field. The proposed KF-PEV resulted in a high-resolution, dense flow field estimates in Fig. 11, with flow vectors consistent with the expected flow dynamics from the SD7003 wing experiment. KF-PEV handled diverse velocity patterns well, representing a diversity of velocity magnitudes and angles. We furthermore smoothed the velocity field by averaging the Kalman velocity measurements within the 8×8 and 16×16 pixel windows, yielding velocity reconstructions in Fig. 12. Larger window yields a smoother, more stable results, while smaller window preserves the details of free flowing particles better. If no measurements exist within the pre-defined window, a larger spatial window was used to estimate the velocity.

Qualitatively, it can be seen that the trails of the events in Fig. 10 match the direction of the reconstructed flow field in Fig. 12. Additionally, the smooth variation of the velocity vectors' direction and magnitude yields what is expected. It can be seen that at 0 degrees, the particle trajectories follow the expected trends, where a higher V_x -velocity is observed on the upper surface and a lower velocity in the wake. The

V_y -velocity also shows the expected behavior. At an -8 degree angle of attack, flow separation on the lower surface can be observed. Even in the separated region of the flow, the particle trajectories in Fig. 12 were captured in great detail using the KF-PEV approach, as seen in both V_x and V_y velocities.

Finally, Kalman filter yields linear least squares error estimates of the state variables in (1), including the acceleration in the x and y directions from events. Although not the main parameter we seek to reconstruction in velocimetry, the estimated particle acceleration field shown Fig. 13 for 0 and -8 degree angles of attack are informative. At 0 degrees, the flow shows almost no acceleration variation except near the front of the airfoil. However, at -8 degrees, significant acceleration variations are observed under the wing where disturbances in the flow occur.

5.3 Discussions

While the proposed KF-PEV algorithm showed competitive performance in the quantitative analysis, it is not without limitations. The computational complexity of KF-PEV

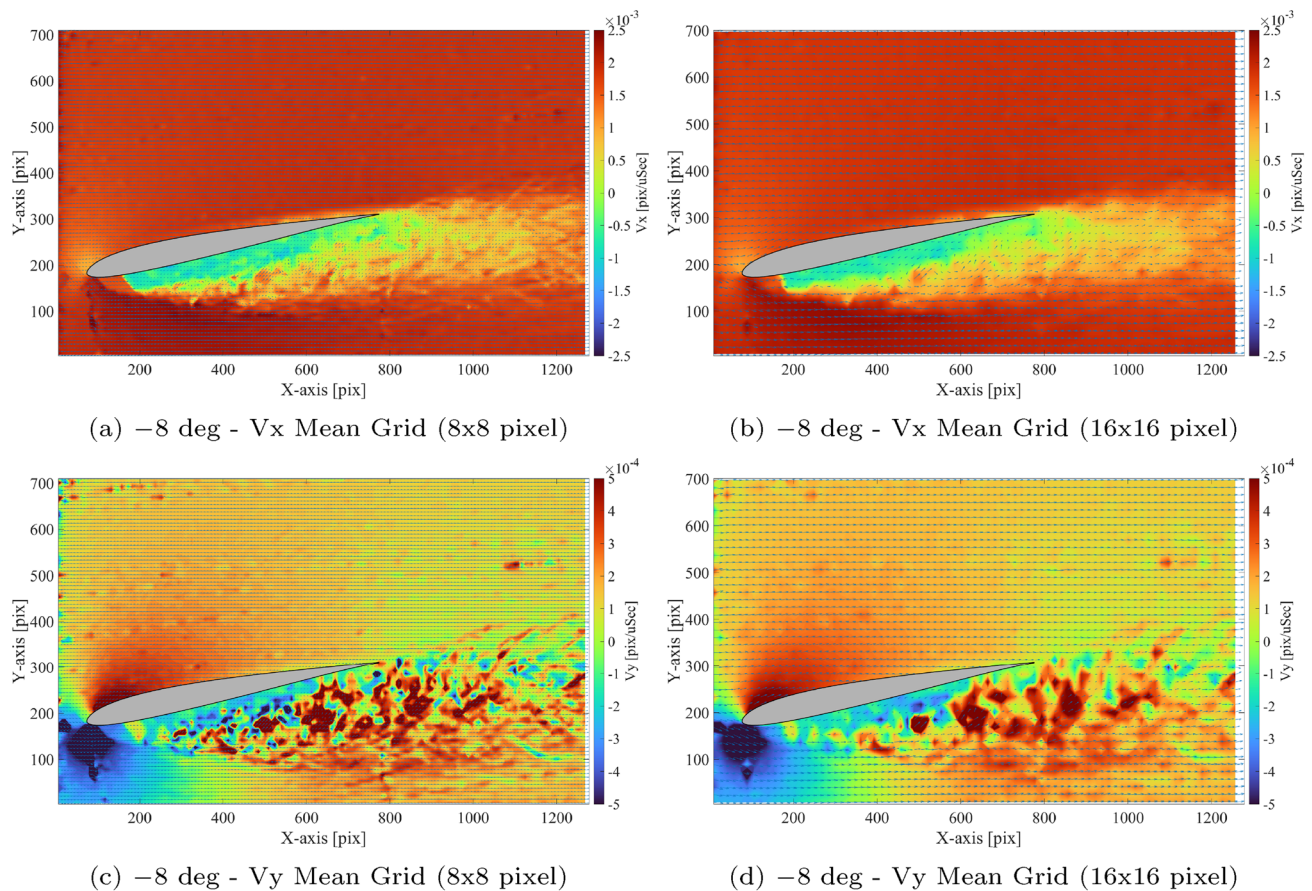


Fig. 12 Uniform velocity field distribution for -8 degree angle of attack, girded in (a, c) 8x8 pixel and (b, d) 16x16 pixel windows. (a, b) Velocity V_x in X direction; (c, d) Velocity V_y in Y direction

scales linearly (roughly) with the particle density. KF-PEV code is written in Matlab (single thread), and flow field reconstruction was conducted a Lenovo Legion Pro 7i 16IRX8H Laptop, equipped with a 2.2 GHz Intel Core i9 HX 24-core CPU and 32GB of DDR5 memory. Table 4 reports the time required to process one million events by KF-PEV as well as alternative methods. Despite all three event sequences comprising 1 million events each, variations in processing time were observed due to differences in event density, noise levels, and positive to negative events ratio within each sample. The analysis revealed that Drazen was faster than Wang in very dense synthetic data, but Wang was faster for less dense real-world data. By contrast, KF-PEV was slower, as expected due to increased computational overhead of Kalman filtering, but the increase was not computationally prohibitive. While we have not achieved real-time performance with the unoptimized Matlab code, we believe it is possible to narrow gap considerably by further optimizing the code, multi-threading (e.g., 24 cores in i9 HX processor), and reduced particle density (a reasonable trade off for applications needing real-time inference of dynamic

flow field). As a future research direction, we plan to develop real-time KF-PEV implementation.

On the other hand, the proposed KF-PEV algorithm incorporates multiple parameters, requiring fine-tuning for some experiments. Some are Kalman filter-specific, such as the covariance matrix for the process and measurement noise; additional parameters such as spatial and temporal thresholds used in track assignment have direction relations to the particle size and particle density. Fortunately, although the tuning was carried out by trial-and-error, we empirically found that the performance of KF-PEV was relatively insensitive or stable to these parameters, and even consistent between the synthetic and real-world experiments. Furthermore, the empirical distribution of the event positions generated by the observed particles is assumed to be sufficiently similar to the normal distribution that the Kalman filter is designed for Benson and Holzinger (2023), such that the KF-PEV still maintains a satisfactory performance. It is difficult to quantify the exact loss stemming from making this assumption, however.

Finally, there are also limitations to the event-based particle velocimetry itself (not specific to KF-PEV). In

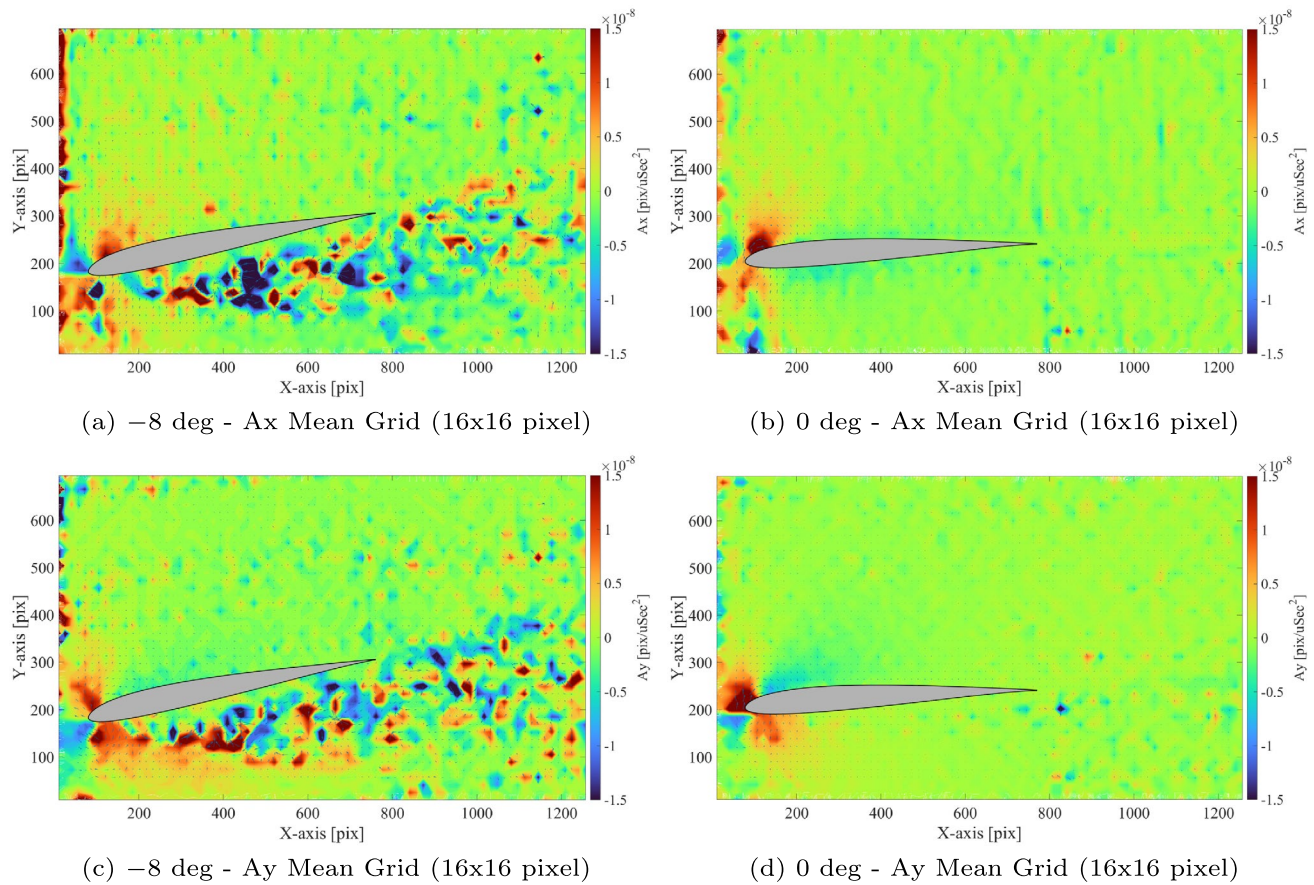


Fig. 13 Kalman state variable also yields particle acceleration. Uniform acceleration field distribution, girded in 16×16 pixel windows, for –8 degree angle of attack (**a, c**) and 0 degree angle of attack (**b, d**). (**a, b**) Acceleration A_X in X direction; (**c, d**) Acceleration A_Y in Y direction

Table 4 Execution time comparison: examining the processing time for 1 million events by Drazen et al. (2011); Wang et al. (2020), and KF-PEV for both synthetic and real-world data

Method	Execution time for 1 million events		
	Synthetic data (≈ 0.5 mSec)	Sample 1 (\approx 172 mSec)	Sample 2 (≈ 102 mSec)
Drazen et al. (2011)	57.19 Sec	29.59 Sec	5.32 Sec
Wang et al. (2020)	80.14 Sec	6.97 Sec	1.91 Sec
KF-PEV (proposed)	457.31 Sec	65.35 Sec	14.83 Sec

our real-world experiments, the number of distinctive particles that the event-based cameras detected seemed to be less than the number of particles recorded by a high-speed intensity camera observing the same seeded flow field. This discrepancy could potentially be attributed to the relatively low pixel density of the commercially available event-based sensors. Like any emerging technology, sensor sensitivity issues are likely to become irrelevant within a few years as the device capabilities advance.

6 Conclusion

In this article, we proposed a causal event-based particle velocimetry technique called Kalman filter-based particle event velocimetry (KF-PEV). Leveraging a Kalman filter redesigned to handle asynchronous event-based pixels and multiple particles, KF-PEV is robust to noise and is in a streaming format. The Kalman filtering also admits a linear least squares estimate of the particle velocity directly, rather than differentiating the positional estimates at the risks of amplifying noise and estimation error.

Additionally, we developed a simulation framework for objectively benchmarking event-based particle velocimetry. This overcomes the limitations of frame-based event simulation methods by generating event streams of moving unresolved targets in continuous time using analytical models of the underlying signal. Evaluation by objective performance metrics indicate that KF-PEV's reconstruction of the flow field represents a substantial improvement over the state-of-the-art frame-based (Thielicke and Sonntag 2021; Janke et al. 2020) and event-based (Wang et al. 2020; Willert and Klinner 2022; AlSattam et al. 2024)

PIV/PTV techniques in terms of velocimetry magnitude and flow direction.

Real event-based sensor data experiments conducted in the water tunnel show robustness to noise and errors in the widest field-of-view reported for event-based particle velocimetry experiment. The proposed KF-PEV successfully tracked particles in dynamic flow field, and reconstructed a dense high-resolution flow field of airfoil consistent with its expected behavior.

Appendix: Benchmark optimization

For the benchmarking methods in Sect. 4.2, we tuned the parameters listed in Table 2 to optimize for the performance. Each parameter is evaluated over its respective dynamic range in terms of MSE of angle and velocity magnitude, as illustrated in Fig. 14. The parameter minimizing the MSE scores are reported in Table 2, and subsequently used to produce Table 3. For this reason, what we report for the competing algorithms are their theoretical peak performance scores.

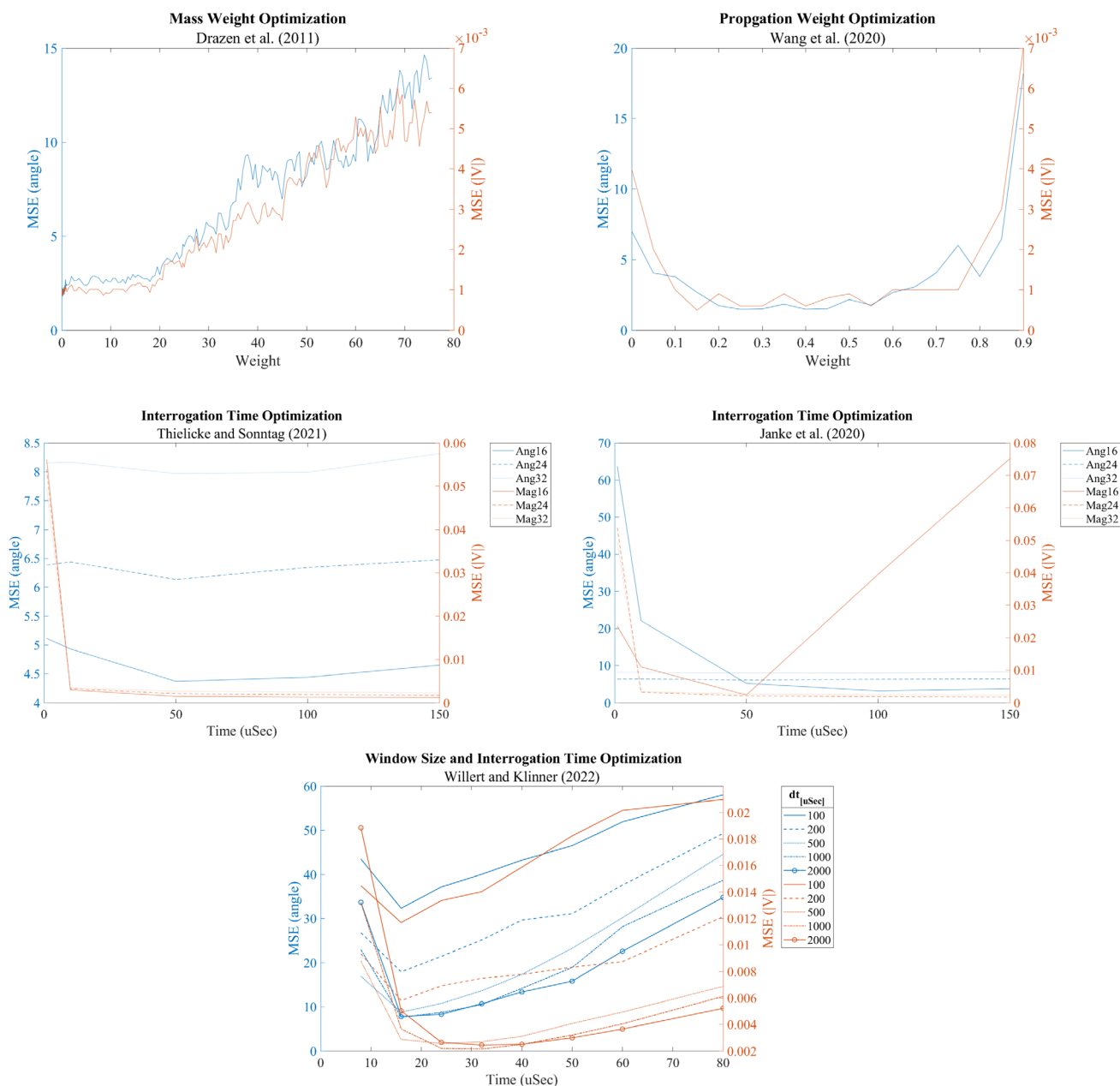


Fig. 14 Parameter performance study for benchmark algorithms

Author Contributions Proposed KF-PEV was developed by OA, MG, and KH, with inputs from MM and SG. Simulation was developed by OA and KH. Real-world experiment was conducted by OA, MM, and SG. Data processing was conducted by OA and KH. Manuscript was drafted by OA, KH, and SG, with inputs from MM and MG.

Funding SG is grateful for the support from AFOSR monitored by Dr. Gregg Abate for extending the current efforts in the experimental evaluation of event cameras for PIV (Grant no: FA9550-23-1-0707).

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

References

- Adrian RJ (1997dec) Dynamic ranges of velocity and spatial resolution of particle image velocimetry Dynamic ranges of velocity and spatial resolution of particle image velocimetry. *Measure Sci Technol*, pp 8121393–1398. <https://doi.org/10.1088/0957-0233/8/12/003>
- AlSattam OA, Mongin MP, Killian A, Gunasekaran S, Hirakawa K (2024) Toward event-based noise-robust high density particle velocimetry toward event-based noise-robust high density particle velocimetry. *AIAA SCITECH 2024 Forum Aiaa scitech 2024 forum* (p 2663)
- Baldwin RW, Almatrafi M, Kaufman JR, Asari V, Hirakawa K (2019) Inceptive event time-surfaces for object classification using neuromorphic cameras inceptive event time-surfaces for object classification using neuromorphic cameras. In: *Image analysis and recognition: 16th international conference, ICIAR 2019, Waterloo, ON, Canada, August 27–29, 2019, Proceedings, Part II 16 image analysis and recognition: 16th international conference, iciar 2019, Waterloo, on, Canada, August 27–29, 2019, proceedings, part ii 16* (pp 395–403)
- Benosman R, Ieng S-H, Clercq C, Bartolozzi C, Srinivasan M (2012) Asynchronous frameless event-based optical flow Asynchronous frameless event-based optical flow. *Neural Netw*, pp 2732–37
- Benson C, Holzinger M (2023) Simulation and analysis of event camera data for non-resolved objects. *J Astron Sci* 71(1):3
- Borer D, Delbruck T, Rösgen T (2017) Three-dimensional particle tracking velocimetry using dynamic vision sensors. *Exp Fluids* 58(12):1–7
- Brookner E (1998) Tracking and kalman filtering made easy
- Brosch T, Tschechne S, Neumann H (2015) On event-based optical flow detection. *Front Neurosci* 9:137
- Drazen D, Lichtsteiner P, Häflicher P, Delbrück T, Jensen A (2011) Toward realtime particle tracking using an event-based dynamic vision sensor. *Exp Fluids* 51(5):1465–1469
- Finateu T, Niwa A, Matolin D, Tsuchimoto K, Mascheroni A, Reynaud E (2020) 5.10 a 1280 × 720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86 μm pixels, 1.066 geps readout. In: *Programmable event-rate controller and compressive data-formatting pipeline. 2020 ieee international solid-state circuits conference-(isscc)* (pp 112–114)
- Gallego G, Delbrück T, Orchard G, Bartolozzi C, Taba B, Censi A et al (2020) Event-based vision: a survey. *IEEE Trans Pattern Anal Mach Intell* 44(1):154–180
- Geisler R, Schr A (2014) Investigation of unsteady flow fields for flow control research by means of particle image velocimetry. *J Flow Control Measure Visual*
- Grewal MS, Andrews AP (2014) *Kalman filtering: theory and practice with matlab*. John Wiley & Sons
- Janke T, Schwarze R, Bauer K (2020) Part2track: a matlab package for double frame and time resolved particle tracking velocimetry. *SoftwareX* 11:100413
- Kalman RE (1960) A new approach to linear filtering and prediction problems
- Lichtsteiner P, Posch C (2008) C. and t. delbruck, “an 12 × 128 120 db 15 μs latency temporal contrast vision sensor. *IEEE J Solid State Circuits* 43:566–576
- Liu T, Merat A, Makhmalbaf M, Fajardo C, Merati P (2015) Comparison between optical flow and cross-correlation methods for extraction of velocity fields from particle images. *Exp Fluids* 56(1):23
- Liu T, Shen L (2008) Fluid flow and optical flow. *J Fluid Mech* 614(253):291
- Mahowald M (1992) Vlsi analogs of neuronal visual processing: a synthesis of form and function
- Ni Z, Pacoret C, Benosman R, Ieng S, REGNIER, S. (2012) Asynchronous event-based high speed vision for microparticle tracking. *J Microsc* 245(3):236–244
- Qureshi MH, Tien W-H, Lin Y-JP (2020) Performance comparison of particle tracking velocimetry (ptv) and particle image velocimetry (piv) with long-exposure particle streaks. *Measure Sci Technol* 32(2):024008
- Raffel M, Willert C, Scarano F, Kähler C, Wereley S, Kompenhans J (2018) *Particle image velocimetry: a practical guide*. Springer. Retrieved from <https://books.google.com/books?id=00q2zQEACAAJ>
- Rebecq H, Gallego G, Mueggler E, Scaramuzza D (2018) Emvs: Eventbased multi-view stereo. 3D reconstruction with an event camera in real-time. *Int J Comput Vis* 126(12):1394–1414
- Rebecq H, Gehrig D, Scaramuzza D (2018, October) ESIM: an open event camera simulator. In: *Conference on robotics learning (CoRL)*
- Rueckauer B, Delbruck T (2016) Evaluation of event-based algorithms for optical flow with ground-truth from inertial measurement sensor. *Front Neurosci* 10:176
- Rusch A, Roesgen T (2021) Trackaer: real-time event-based particle tracking. In: *14th International symposium on particle image velocimetry (vol 1)*
- Rusch A, Rösgen T (2023) Trackaer: real-time event-based quantitative flow visualization. *Exp Fluids* 64(8):136
- Scharnowski S, Kähler CJ (2020) Particle image velocimetry-classical operating rules from today's perspective. *Opt Lasers Eng* 135:106185
- Shah S, Dey D, Lovett C, Kapoor A (2017) Airsim: High-fidelity visual and physical simulation for autonomous vehicles. *Field and service robotics*. Retrieved from <https://arxiv.org/abs/1705.05065>
- Thielicke W, & Sonntag R (2021) Particle image velocimetry for matlab: accuracy and enhanced algorithms in pivlab. *J Open Res Softw*, 9(1)
- Wang Y, Idoughi R, Heidrich W (2020) Stereo event-based particle tracking velocimetry for 3d fluid flow reconstruction. In: *European conference on computer vision* (pp 36–53)
- 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):1–20

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