Chapter 6
Streaming PCA for Real-Time Structural Dynamics Tracking in Event-Based Vision

Moises Felipe Silva, Anthony Starfleaf, Andre Green, Nicholas Bruns, and David Mascarenas

Abstract Event-based neuromorphic imagers deliver asynchronous streams of pixel events encoding scene dynamics with microsecond resolution, but extracting precise object trajectories from such high-throughput data remains challenging. We introduce a set of PCA-based tracking methods that use the dominant direction of recent events to find and follow moving targets. Each method keeps a short history of event coordinates and updates its estimated principal direction of motion - either by recalculating from scratch or by fast incremental updates - so that the many event points collapse into a simple, low-dimensional view. In this view, each object appears as a clear feature, which we locate, refine to get precise positions, and then link over time to form continuous tracks. We demonstrate these approaches on scenes with several objects moving at different speeds and rotational rates, showing they offer reliable, scalable multi-object tracking with minimal computation. These PCA-driven techniques form a lightweight, flexible toolkit for real-time motion analysis in robotics, surveillance, structural monitoring, and other applications using event-based vision.

Keywords event-based cameras · machine learning · motion tracking · PCA · low-light sensing, tracking · object tracking

Introduction

Traditional cameras capture scenes as a sequence of complete frames, which imposes both latency and redundancy. In contrast, neuromorphic or event-based sensors operate by detecting and reporting changes in brightness at each pixel as they occur. This asynchronous design delivers temporal resolutions on the order of microseconds, effective rates well above 5,000 frames per second, and dynamic ranges exceeding 120 dB - greater than those available from conventional imagers [1, 2]. The output of such sensors resembles the operation of biological retinas, hence their frequent description as silicon retinas.

Despite these advantages, much of the current practice in event-based vision involves reassembling streams of events into artificial frame-like structures. While convenient, this approach diminishes the unique benefits of event data, especially its ability to encode motion directly at very fine time scales [3]. A more natural interpretation is to regard the event stream as a high-dimensional point cloud evolving in space and time. Within this representation, statistical tools such as principal component analysis (PCA) are well suited to expose dominant directions of motion and to reduce data complexity [1].

Here, we advance this perspective by formulating a tracking method that operates entirely in event space. The proposed algorithm maintains a continuously updated PCA representation, either on an event-by-event basis for minimal latency, or over short event buffers to improve throughput. These representations serve as the foundation for efficient object localization and, by extending the centroid trajectories into the frequency domain, for real-time spectral estimation as well. We evaluate the framework on synthetic and laboratory data of oscillating objects, demonstrating accurate recovery of both motion and oscillation characteristics at low computational cost.

Moises Felipe Silva (✉) · Anthony Starfleaf · Andre Green · Nicholas Bruns · David Mascarenas
Los Alamos National Laboratory, Los Alamos, NM 87545
e-mail: mfelipe@lanl.gov; aglor@lanl.gov; andre_green@lanl.gov; bruns@lanl.gov; dmascarenas@lanl.gov

Kristin B. Zimmerman, PhD (eds.), *Computer Vision and SLDV for Structural Dynamics, 2026, Vol. 6*, Conference Proceedings of the Society for Experimental Mechanics Series, https://doi.org/10.1007/978-8-743-81445-0_6

Related Works

Efforts to harness the capabilities of event-based sensors span both adaptations of existing vision pipelines and new algorithms designed for asynchronous data. One of the earliest strategies was to accumulate events into frame-like images or voxel grids, thereby enabling the reuse of conventional vision algorithms. While effective for compatibility, this practice discards much of the fine temporal structure that makes event data attractive in the first place [3].

Recent approaches emphasize processing events directly, with PCA standing out for such applications. Khairallah et al. [4] demonstrated that local principal components could be used to infer optical flow vectors for odometry at real-time rates, while Ramesh et al. [5] introduced PCA-RECT, which applied PCA to normalized event patches for efficient object detection on FPGA platforms. These works illustrate PCA's efficiency but are largely confined to single-object motion without addressing concurrent frequency recovery. Broader tracking methods have been explored by Hamann et al. [6], where a timestamp-driven scheme for following arbitrary features across the sensor has shown robustness under extreme speed and lighting conditions. On the other hand, deep learning approaches have exploited event sparsity using transformer and attention mechanisms [7], achieving impressive accuracy but often at high computational cost.

On a different note, another important marker of a subject's characterization is its oscillatory frequency content to be estimated from event streams. Pfrommer et al. [8] showed that flicker frequencies can be measured directly from asynchronous pixel activity, and later simulation studies investigated how different motion regimes affect frequency estimation [9]. Multi-window ensemble approaches have also been applied to location tracking, balancing the trade-off between latency and robustness [10]. These methods demonstrate the potential of using event data for spectral analysis, though they typically consider single or static sources rather than multiple moving targets.

Therefore, our work builds upon these foundations by integrating localization, trajectory recovery, and spectral estimation into a single PCA-based framework. Unlike earlier PCA applications, our methods maintain streaming updates, and operate effectively in multi-object scenarios. This positions the approach as a bridge between statistical PCA techniques and the broader goals of event-driven spectral and motion analysis.

Background

Problem Statement

Considering each asynchronous event as

$$\boldsymbol{e_i} = (\boldsymbol{x_i}, \boldsymbol{y_i}, \boldsymbol{t_i})\,,$$

where (x_i, y_i) is the pixel location and t_i the i-th timestamp. We group events into non-overlapping windows of duration T (per-buffer mode) or into a sliding window of duration T that advances by each new event (per-sample mode). Then, for the k-th window, we collect the N_k events into a data matrix

$$X^k = \begin{bmatrix} x_1 & x_2 & \cdots & x_{N_k} \\ y_1 & y_2 & \cdots & y_{N_k} \end{bmatrix} \in \mathbb{R}^{2\times N_k}\,.$$

We seek a rank-r subspace, spanned by orthonormal basis $W \in \mathbb{R}^{2\times r}$, that best reconstructs the centered data set X. That is analogous to the PCA minimization function via the eigen decomposition of the covariance matrix [11]:

$$C^k = \frac{1}{N_k}\left(X^k - \mu 1^\intercal\right)\left(X^k - \mu 1^\intercal\right)^\intercal,$$

where $\mu \in \mathbb{R}^2$ is the sample mean and 1 is a length-N_k vector of ones for padding. Solving this yields the top r eigenvectors of C^r, which become the columns of W. In per-sample mode, these statistics are updated with each new event, while in per-buffer mode, they are recomputed over the entire batch of N_k events at once. Then, for the i-th event, we define its running means as:

$$\mu_x = \frac{1}{N_k}\sum_{i=1}^{N_k} x_i, \qquad \mu_y = \frac{1}{N_k}\sum_{i=1}^{N_k} y_i,$$

and the centered covariances as

$$Cov\,(X,X) = \frac{1}{N_k - 1}\sum_{i=1}^{N_k}(x_i - \mu_x)^2, \qquad Cov\,(Y,Y) = \frac{1}{N_k - 1}\sum_{i=1}^{N_k}(y_i - \mu_y)^2,$$

where the full multivariate covariance matrix becomes

$$Cov\,(X,Y) = \frac{1}{N_k - 1}\sum_{i=1}^{N_k}(x_i - \mu_x)\,(y_i - \mu_y)\,,$$

deriving a symmetric and positive semidefinite matrix. Therefore, solving the eigenproblem $\det\,(C - \lambda I)\ = 0$ yields the characteristic equation

$$\lambda^2 - (Cov\,(X,\ X) + Cov\,(Y,\ Y))\,\lambda + \left[Cov\,(X,\ X) + Cov\,(Y,\ Y) - Cov(X,\ Y)^2\right] = 0,$$

whose roots

$$\lambda_{1,2} = \frac{Cov\,(X,\ X) + Cov\,(Y,\ Y) \pm \sqrt{(Cov\,(X,\ X) - Cov\,(Y,\ Y))^2 + 4\,Cov(X,Y)^2}}{2}$$

are the principal variances and the corresponding unnormalized eigenvectors admit the closed form

$$v_i \propto \begin{bmatrix} \lambda_i - Cov\,(Y,Y) \\ Cov\,(X,Y) \end{bmatrix},\ \ i = 1, 2.$$

After the variances have been normalized to unit length, projecting onto $[v_1, v_2]$ yields the best rank-2 reconstruction of the event cloud. To avoid $\mathcal{O}(N_k)$ recomputation on each event, when a new event (x_{new}, y_{new}) arrives and the oldest (x_{old}, y_{old}) is evicted, we update the means in constant time:

$$\mu_{x_{new}}\ =\ \mu_x\ + \frac{x_{new}\ -\ x_{old}}{N_k},\quad \mu_{y_{new}}\ =\ \mu_y\ + \frac{y_{new}\ -\ y_{old}}{N_k}.$$

In a similar fashion, each covariance entry is thus updated by removing the outgoing contribution and adding the incoming one:

$$Cov(X,Y)_{new}\ =\ Cov\,(X,Y) + \frac{[\,(x_{new}\ -\ \mu_{x_{new}})\,(y_{new}\ -\ \mu_{y_{new}}) -\ (x_{old}\ -\ \mu_x)\,(y_{old}\ -\ \mu_y)]}{(N_k - 1)}.$$

Finally, we recompute $\lambda_{1,2}$ and $v_{1,2}$ via the closed-form formulas above, achieving an exact sliding-window PCA in constant time per event. This whole process is also illustrated by Figure 1.

To estimate the positions of multiple moving objects, we project the events onto the principal axis identified by PCA. Each event is represented by a scalar coordinate along this axis, producing a one-dimensional distribution of activity. This distribution is discretized into equally spaced bins, forming a running histogram of event counts that evolves with the sliding buffer. The projection of each event onto the principal axis is given by

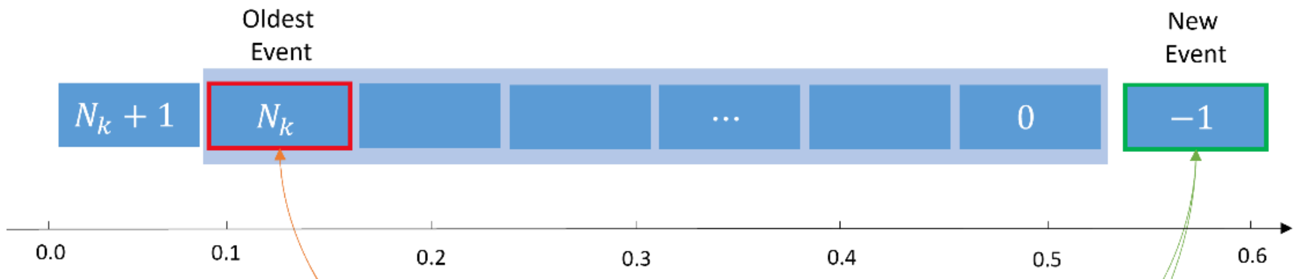
$$s_i = v_1^{\intercal}\begin{bmatrix} x_i \\ y_i \end{bmatrix}.$$

For the histogram, we define a set of bin edges with spacing γ, covering the full range of projected coordinates. Each incoming event is assigned to one of these bins according to its projected value:

$$[s_{min}\,(j-1)\,\gamma, s_{min} + j\,\gamma)\,,\ \ j = 1, \ldots,\ \gamma.$$

As the buffer advances, updates are handled incrementally. When an old event leaves the buffer, the count in its corresponding bin is decreased, and when a new event enters, the count of its bin is increased. This allows the histogram to be maintained efficiently without recomputation. From this histogram, coarse positions of objects are obtained by identifying peaks with the highest event counts. To improve accuracy, we refine each peak location by computing an average-smoothed centroid over a neighborhood of bins within a half-width γ. This refinement produces a sub-bin resolution estimate of each object's location.

Once positions have been estimated over time, the resulting trajectories are transformed into the frequency domain using a discrete Fourier transform. The dominant peak in the frequency spectrum corresponds to the oscillatory behavior of the tracked object, and by mapping this peak to the physical sampling interval, we recover its oscillation frequency. The histogram-based object's localization is summarized in Figure 2. These steps altogether allow to maintain a high-resolution spatial histogram of events along the principal PCA axis, enabling efficient position estimation at event-rate.



First,

$$\mu_{x_{new}} = \mu_x + \frac{x_{new} - x_{old}}{N_k}, \qquad \mu_{y_{new}} = \mu_y + \frac{y_{new} - y_{old}}{N_k}.$$

$$Cov(X,Y)_{new} = Cov(X,Y)_{old} + \frac{\left[(x_{new} - \mu_{x_{new}})(y_{new} - \mu_{y_{new}}) - (x_{old} - \mu_x)(y_{old} - \mu_y)\right]}{(N_k - 1)}.$$

Then update,

$$\lambda_{1,2} = \frac{Cov(X,X) + Cov(Y,Y) \pm \sqrt{\left(Cov(X,X) - Cov(Y,Y)\right)^2 + 4\,Cov(X,Y)^2}}{2}$$

$$v_i \propto \begin{bmatrix} \lambda_i - Cov(Y,Y) \\ Cov(X,Y) \end{bmatrix}, \qquad i = 1,2.$$

Fig. 1 Sliding-window PCA update. The buffer (blue) shifts forward by dropping the oldest event (red) and adding the newest (green). Running means and covariances are updated incrementally, enabling constant-time PCA without full computation.

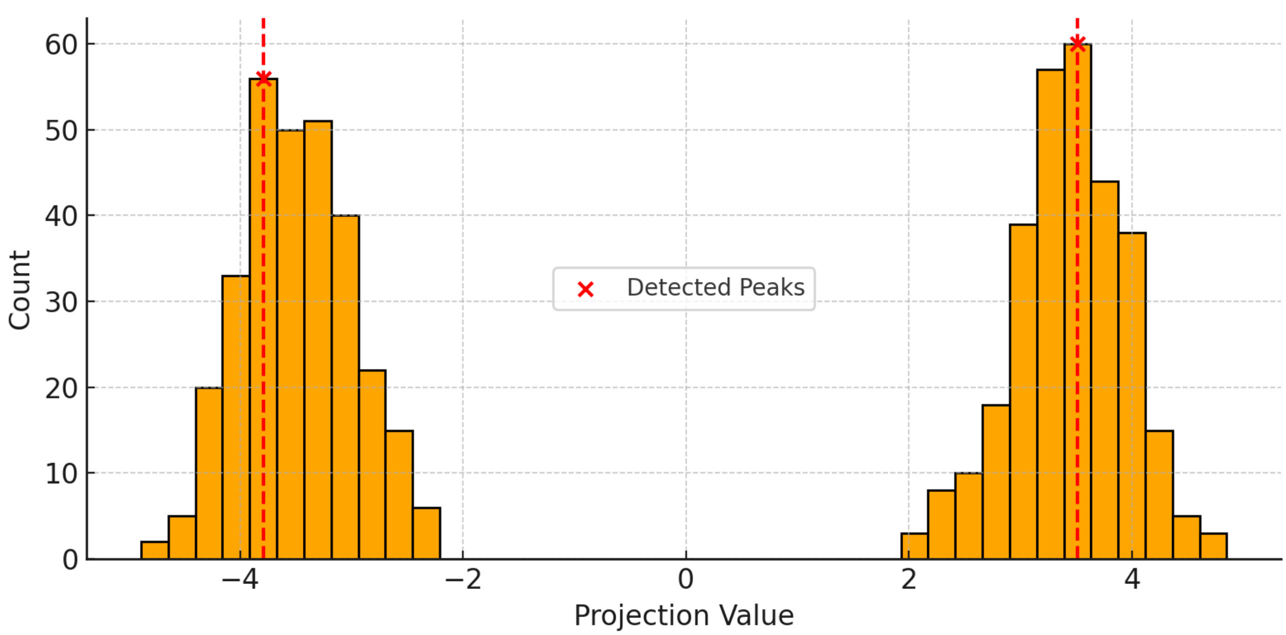


Fig. 2 Object localization along the PCA axis. Events are projected onto the dominant direction, binned into histograms (orange), and the peak bin is refined with a local centroid estimate (red) for precise position recovery.

Results

Simulated Experiments

To validate the proposed framework, we conducted a series of simulated experiments. We modeled moving objects as a sparse set of Gaussian discrete points that both oscillate at a prescribed frequency and may or may not move following a circular trajectory. The sensor field is defined by horizontal and vertical bounds of 240×180 px, simulating conventional event-based imagers. Therefore, the center of motion is taken as the midpoint of those ranges. In the case of N moving targets, their orbit radii are chosen by linearly spacing values between one-eighth and one-third of the field width, and we pick rotation speeds evenly between 0.1 and 0.2 revolutions per second. Each moving object also receives a random phase offset in $[0,\ 2\pi)$ to avoid synchronous trajectories.

At each time step, for each object, we define an oscillation (blinking) cycle to have it outputting “on” or “off” events. In that case, the first half of the cycle the object emits “on” events and output “off” events for the other half of the cycle. When “on”, it generates a set number of points spread around its current position in a soft, cloud-like pattern defined by a Gaussian distribution centered at $(c_x,\ c_y)$, where

$$c_x = x + r\cos(\omega t + \phi)\ ,\ c_y = y + r\sin(\omega t + \phi)\ .$$

To emulate background and sensor noise, we inject a Poisson-like cloud of uniformly distributed noise events across the full field of view of the simulated sensor. Figure 3 shows four different simulated scenarios: in (a), three stationary objects blink at distinct frequencies; in (b–d), the same objects move along different trajectories while blinking at varying rates.

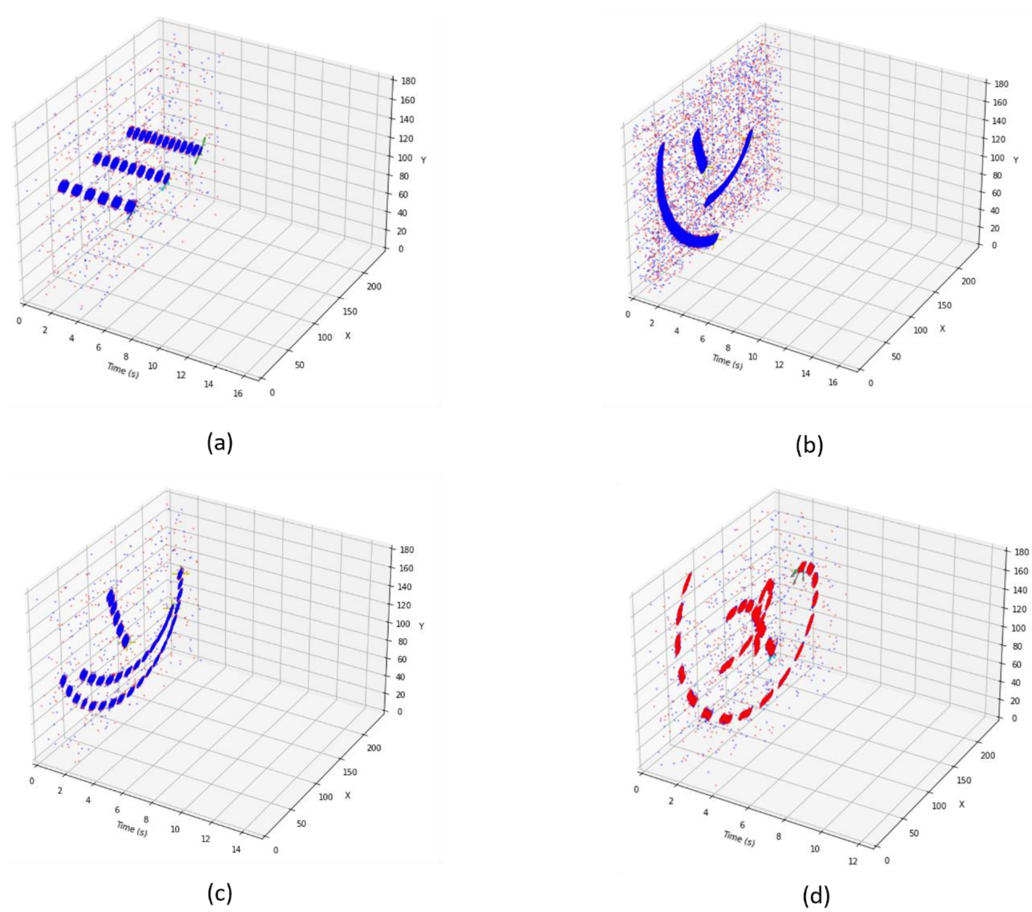
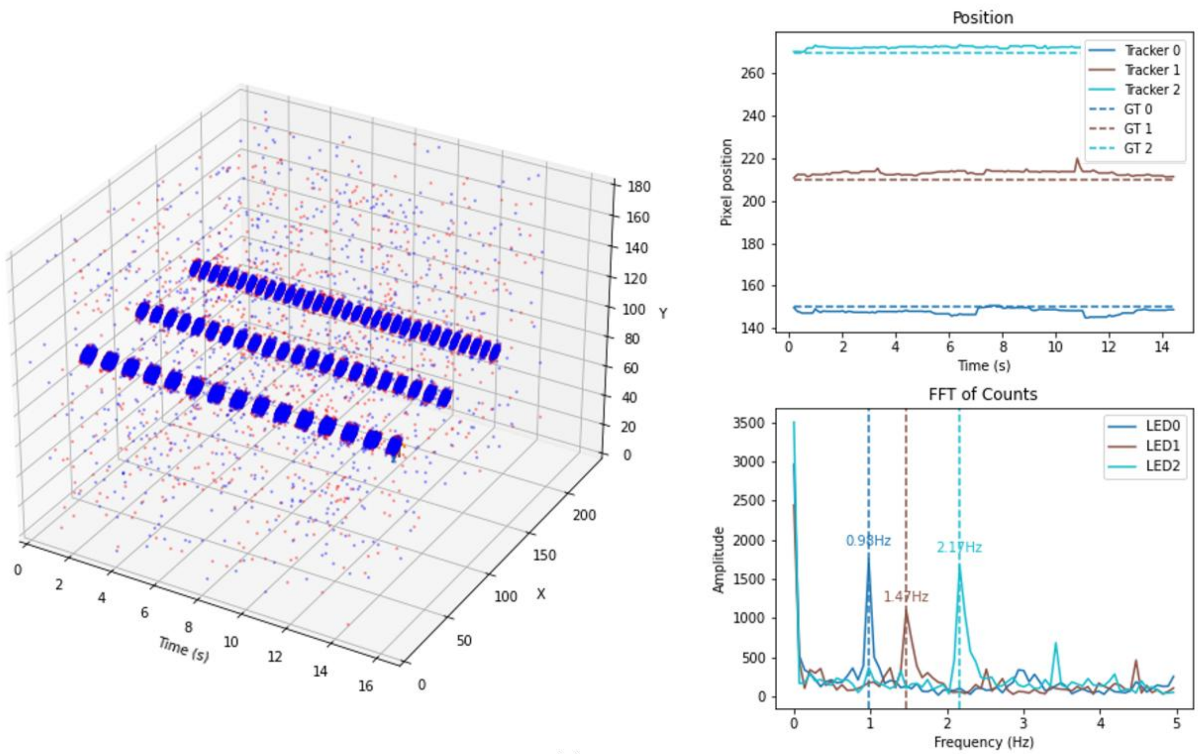
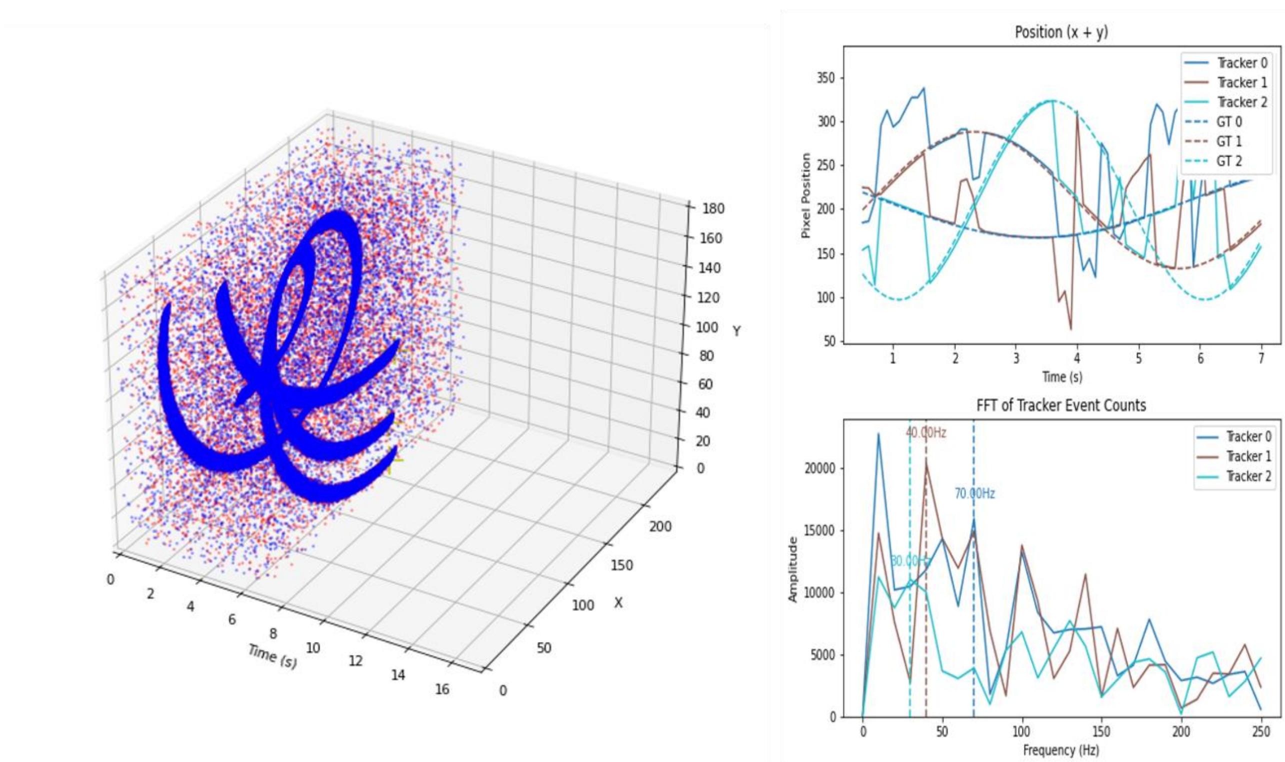


Fig. 3 3D visualizations of four simulated scenarios with three oscillating objects. (a) Stationary objects blinking at 1.0, 1.5, and 2.2 Hz; (b) circular motion with fast blinking at 30, 40, and 70 Hz; (c) moderate blink rates of 2.0, 4.5, and 7.0 Hz; (d) slower oscillations at 1.25, 2.0, and 3.5 Hz.



(a)



(b)

Fig. 4 *Continued*

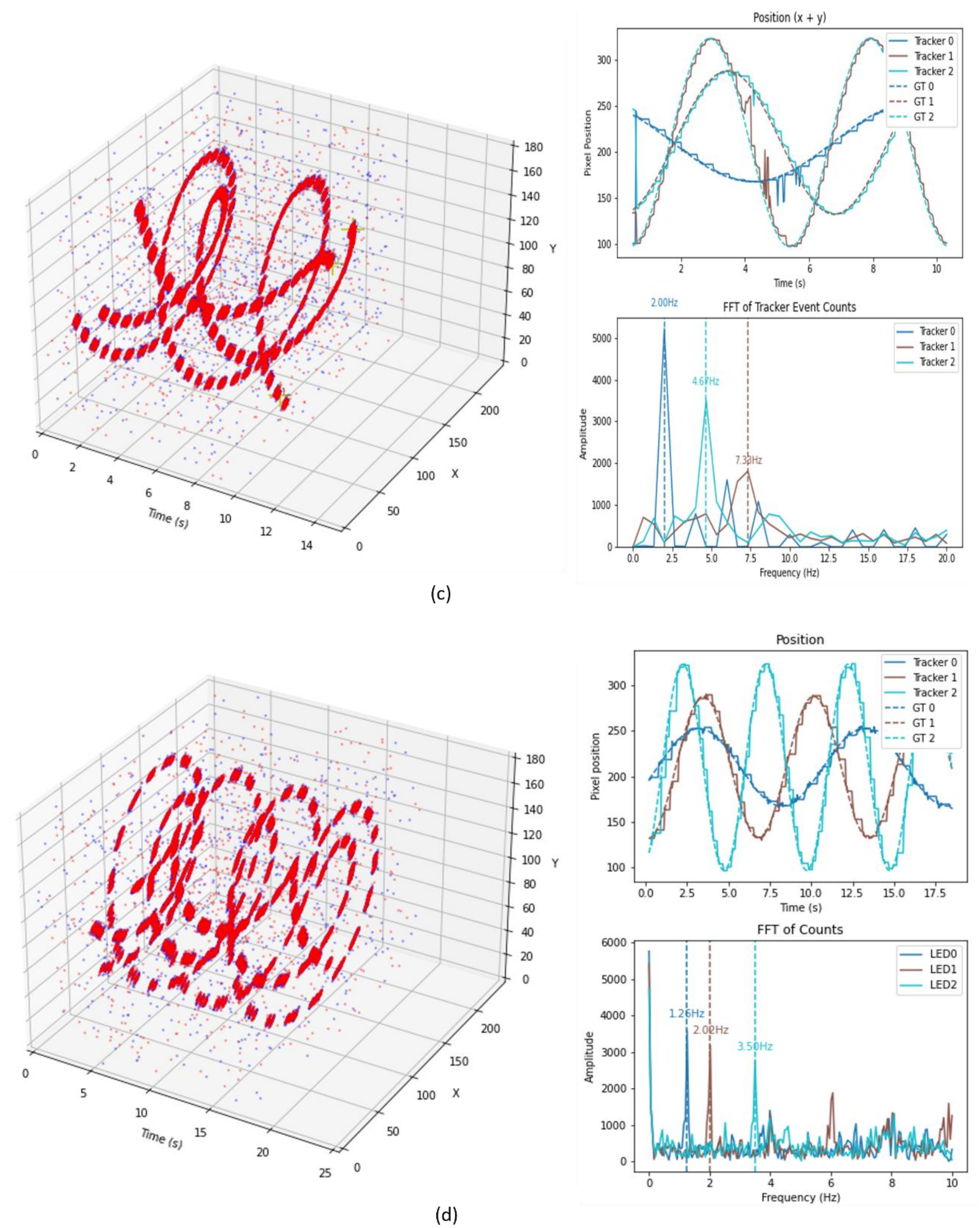


Fig. 4 Frequency estimation and spatial localization results for four scenarios: (a) static blinkers at 1.0, 1.5, 2.2 Hz ($\sim$0.02 Hz error); (b) rapid motion at 30, 40, 70 Hz ($\sim$0.5 Hz error with brief instabilities); (c) moderate motion at 2.0, 4.5, 7.0 Hz ($\sim$0.03 Hz error); (d) slow oscillators at 1.25, 2.0, 3.5 Hz ($\sim$0.015 Hz error).

Analysis

As shown in Figure 4, our PCA-based tracking filter accurately recovers the dominant oscillation frequencies in all four simulated scenarios. In the static case (Figure 5-a), three objects blinking at 1.0, 1.5, and 2.2 Hz yield a mean absolute frequency error of approximately 0.02 Hz ($\approx$ 1% relative error). Under rapid motion, Figure 5-b (30, 40, and 70 Hz blinkers), the tracker experiences brief chaotic behavior when object trajectories intersect but stabilizes on its own, underscoring the framework's resilience. Despite these transient instabilities, frequency recovery remains largely unaffected (MAE $\approx$ 0.5 Hz, max error 1.1 Hz). We do observe that at higher frequencies the energy in harmonic peaks increases, which could occasionally lead to misidentifying a harmonic rather than the fundamental frequency. Moderate-motion scenarios (Figure 5-c at 2.0, 4.5, 7.0 Hz and Figure 5-d at 1.25, 2.0, 3.5 Hz) maintain high spectral fidelity, with MAE around 0.03 Hz and 0.015 Hz, respectively. In per-buffer mode, centroid localization remains robust, with an average spatial error of 1.2 px (σ = 0.3 px) across all scenarios. When Poisson-distributed background events are injected at 20% of the scene rate, both spectral and spatial errors increase by only about 10%, demonstrating the filter's robustness to realistic noise levels.

Importantly, the filter exhibits strong resilience to realistic sensor noise and partial overlap among event clouds. When Poisson-distributed background events are injected at 20% of the scene rate, both spatial and spectral metrics degrade by only about 10%, underscoring robustness to moderate noise levels. Together, these results demonstrate that our unified approach not only matches or exceeds prior single-object PCA methods in accuracy but also extends naturally to multiple, concurrent targets without sacrificing speed. By fusing histogram-based localization with sliding-window PCA updates, the tracker delivers high-precision spatial and spectral estimates in real-time, paving the way for rich, event-driven motion analyses in low-latency vision systems.

Laboratory Tests

To evaluate the tracker under real sensing conditions, we conducted a laboratory test using a single LED blinking at 1 Hz with a 50% duty cycle. The LED was fixed in front of the event camera and driven by a function generator. The LED was mounted in a fixed position directly in front of the event camera, providing a stable, high-contrast point source. This offers a clean, well-controlled source for validating frequency estimation.

The PCA-based tracker was implemented in a manner where incoming events were projected along the principal axis and accumulated in the histogram to refine the centroid location, as mentioned earlier. Figure 5 shows the histogram segment corresponding to approximately 4 seconds of recording, and it shows a clear, narrow peak at the LED position, demonstrating the method's ability to isolate a static point source under continuous blinking. An event snapshot created at the same time, shown in Figure 6, confirms this localization, with the detected centroid marked by a black cross.

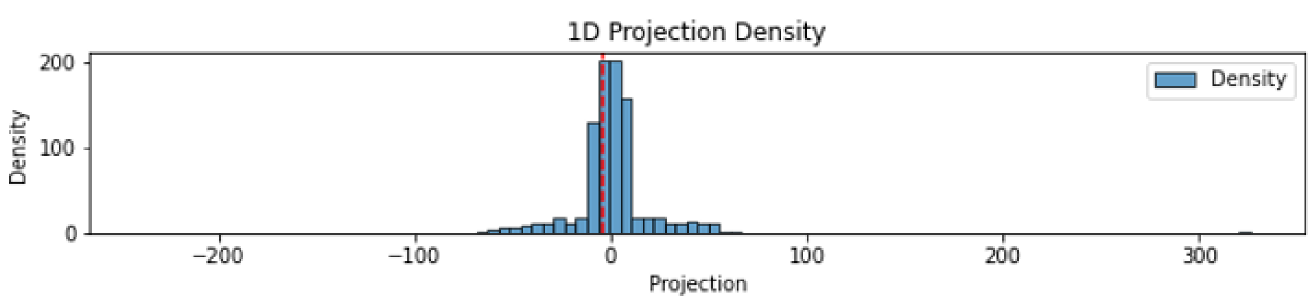


Fig. 5 Histogram of projected events over a 4-second recording of a static 1 Hz LED. A sharp, narrow peak indicates the detected LED position, demonstrating stable localization under periodic blinking.

We also examined the LED's detected pixel position over time (Figure 6). Although the source itself was fixed, small oscillations are visible in the estimated position trace. These appear as regular spikes and are attributed to the LED's on-off transitions. As the LED switches between on and off, subtle imbalances in event polarity can cause the centroid estimate to jitter slightly around its true position. Despite this artifact, the tracker remains stable. Finally, the frequency spectrum of the tracker's event counts reveals a dominant peak at 1 Hz, aligned with the programmed blink rate. These results confirm that the PCA tracker can not only localize a simple static source but also recover its periodic dynamics with high accuracy in real-time.

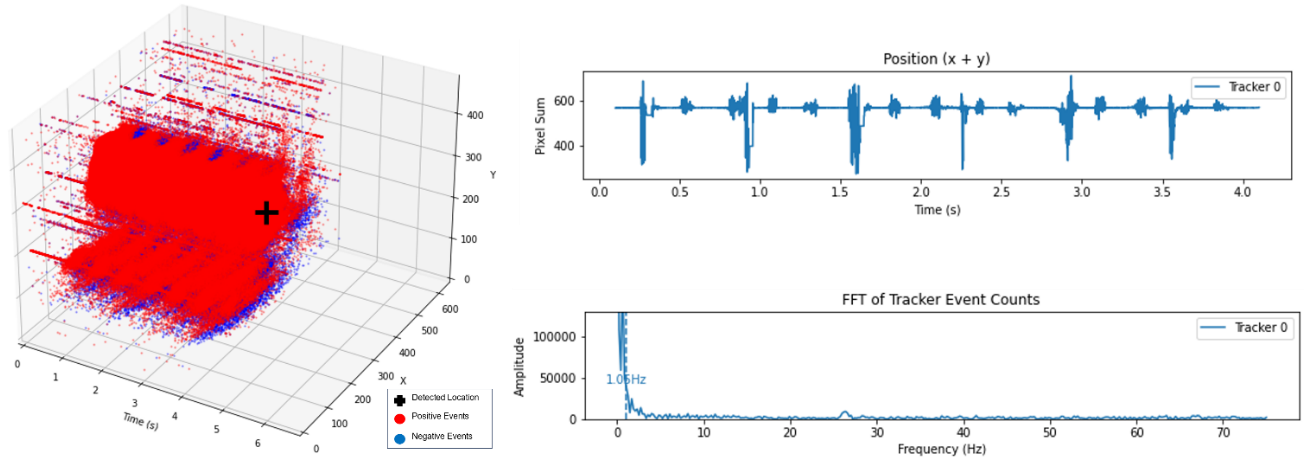


Fig. 6 Event snapshot and spectral analysis of the static 1 Hz LED. Left: detected centroid marked by a black cross on the event display. Right-top: estimated pixel position versus time, showing small oscillations linked to the LED's on-off transitions. Right-bottom: frequency spectrum of tracker event counts with a clear dominant peak at 1 Hz.

Conclusions

This paper presented a PCA-based framework for object tracking and frequency identification in event-driven sensing systems. In particular, the method was assessed through simulation studies and a laboratory experiment involving oscillating light sources, where it was shown to provide accurate estimates of spatial trajectories and dominant motion frequencies. The framework relied on continuous updates of principal directions combined with histogram refinement and did not require reconstructed frames or trained learning models. It was further observed that the approach handled overlapping motion paths and noise without significant degradation in performance. These findings indicate that PCA-based processing constitutes a reliable basis for low latency motion analysis in embedded applications.

Acknowledgments This paper was supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project number 20250620MFR, as well as Innovation Hub Sprint funding sources internal to Los Alamos National Laboratory. Los Alamos National Laboratory is operated by Triad National Security, LLC, for the National Nuclear Security Administration of US Department of Energy (Contract No. 89233218CNA000001). This work was also supported by the Department of Energy – Environmental Management Office with R&D Award: 277483. The DVS/DAVIS technology was developed by the Sensors group of the Institute of Neuroinformatics (University of Zurich and ETH Zurich), which was funded by the EU FP7 SeeBetter project (grant 270324).

References

1. G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based Vision: A Survey," IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 1, pp. 154–180 (2020)
2. Luming Wang, Hao Shi, Xiaoting Yin, Kailun Yang, Kaiwei Wang, and Jian Bai, "EgoEvGesture: Gesture Recognition Based on Egocentric Event Camera," CoRR, abs/2503.12419v2 (2025)
3. X. Zheng, Y. Liu, Y. Lu, T. Hua, T. Pan, W. Zhang, D. Tao, and L. Wang, "Deep Learning for Event-based Vision: A Comprehensive Survey and Benchmarks," CoRR, abs/2302.08890 (2023)
4. Mahmoud Z. Khairallah, Fabien Bonardi, David Roussel, and Samia Bouchafa, "PCA Event-Based Optical Flow for Visual Odometry," CoRR, 2021
5. B. Ramesh, A. Ussa, L. Della Vedova, H. Yang, and G. Orchard, "PCA-RECT: An Energy-efficient Object Detection Approach for Event Cameras," CoRR, abs/1904.12665v1
6. F. Hamann, D. Gehrig, F. Febryanto, K. Daniilidis, and G. Gallego, "ETAP: Event-based Tracking of Any Point," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2025
7. A. Sabater, L. Montesano, and A. C. Murillo, "Event Transformer+: A multi-purpose solution for efficient event data processing," CoRR, abs/2211.12222v2
8. B. Pfrommer, "Frequency Cam: Imaging Periodic Signals in Real-Time," CoRR, abs/2211.00198

9. C. J. Benson and M. Holzinger, “Simulation and Analysis of Event Camera Data for Non-resolved Objects,” Journal of the Astronautical Sciences, vol. 71, article 3
10. T. Fischer and M. Milford, “Event-based visual place recognition with ensembles of temporal windows,” IEEE Robotics and Automation Letters, vol. 5, no. 4, pp. 6924–6931 (2020)
11. I. T. Jolliffe, “Principal Component Analysis,” 2nd ed., Springer (2002)
12. M. Brand, “Fast Low-Rank Modifications of the Thin Singular Value Decomposition,” MERL Tech. Rep. TR2006-059 (2006)