EventTracker: 3D Localization and Tracking of High-Speed Object with Event and Depth Fusion

Luo Xinyu
Shenzhen International Graduate School, Tsinghua University
ShenZhen, China
luo-xy23@tsinghua.edu.mail.cn

Haoyang Wang
Shenzhen International Graduate School, Tsinghua University
ShenZhen, China
haoyang-22@mails.tsinghua.edu.cn

Ciyu Ruan
Shenzhen International Graduate School, Tsinghua University
ShenZhen, China
rcy23@mails.tsinghua.edu.cn

Chenxin Liang
Shenzhen International Graduate School, Tsinghua University
ShenZhen, China
liangcx23@mails.tsinghua.edu.cn

Jingao Xu
School of Software, Tsinghua University
Beijing, China
xujingao13@gmail.com

Xinlei Chen*
Shenzhen International Graduate School, Tsinghua University
Pengcheng Laboratory
RISC-V International Open Source Laboratory
ShenZhen, China
chen.xinlei@sz.tsinghua.edu.cn

ABSTRACT

Accurately localizing high-speed dynamic objects in 3D space with low latency is crucial for various robotic applications. Current methods face challenges due to extended exposure times and limited sensor resolution, hindering precise object detection and localization. Event cameras, known for their high temporal resolution and asynchronous nature, offer a promising solution. To leverage the potential of the event camera, we propose *EventTracker*, a novel framework that integrates event and depth measurements for precise and low-latency 3D localization and tracking of the high-speed dynamic object. EventTracker incorporates a collaborative object detection and tracking algorithm optimized for both event and depth data, overcoming detection and registration challenges. Additionally, a graph-instructed optimization algorithm enhances accuracy by fusing heterogeneous sensor data effectively. Experimental evaluation in dynamic environments demonstrates significant improvements in localization performance compared to baseline methods.

CCS CONCEPTS

• **Computer systems organization → Sensors and actuators**; • **Computing methodologies → Optimization algorithms**.

KEYWORDS

Perception, Event Camera, Optimization, Object Tracking

ACM Reference Format:
Luo Xinyu, Haoyang Wang, Ciyu Ruan, Chenxin Liang, Jingao Xu, and Xinlei Chen. 2024. EventTracker: 3D Localization and Tracking of High-Speed Object with Event and Depth Fusion. In *International Workshop on Physics Embedded AI Solutions in Mobile Computing (PICASSO 24), November 18–22, 2024, Washington D.C., DC, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3636534.3694721

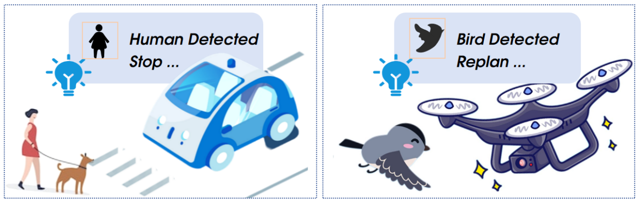


Figure 1: Illustration of the importance of high-speed perception.

1 INTRODUCTION

The ability of localizing dynamic objects at high speeds is crucial for various robots operations [1, 2] and even swarm control [3, 4]. Specifically, accurately estimating the location of high-speed objects in 3D space with low latency allows robots to have more reaction time and perform precise operations during interactions with objects [5, 6]. This capability is critically important in various robotic applications, such as enabling autonomous vehicles to avoid sudden pedestrians [7, 8], or helping drones evade unexpected obstacles [9, 10] (Fig.1). In these scenarios, even a slight delay or inaccuracy leads to significant failures, causing financial loss and threatening safety [11, 12].

Unfortunately, current methods are not able to offer feasible solutions for accurate and low-latency localization of the high-speed object, which can be divided into two categories:

Frame camera-based solutions. Using classical visual feature matching or learning-based techniques, these methods achieve object localization with frame camera [13, 14]. However, they suffer from lengthy localization latency due to extended frame exposure times (around 20ms) and additional image processing delays (another 10 ~ 20ms) [15]. Furthermore, motion blur in each frame confuses the algorithm, exacerbating the localization error [16].

LiDAR and radar-based solutions. Current practices employ

[*] Xinlei Chen is the corresponding author.

PICASSO 24, November 18–22, 2024, Washington D.C., DC, USA

ACM ISBN 979-8-4007-0489-5/24/11.
https://doi.org/10.1145/3636534.3694721

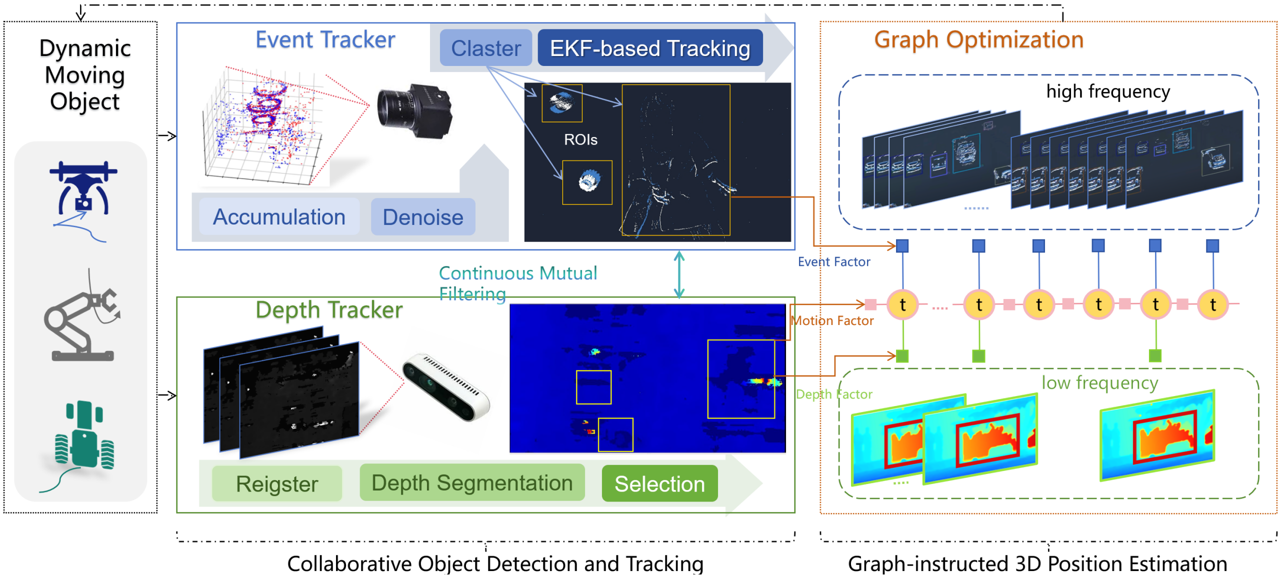


Figure 2: Overview of EventTracker.

filter and learning-based techniques for object tracking with LiDAR and radar sensors [17, 18]. These sensors operate by emitting light or frequency-modulated continuous waves in a specific spectrum, and these methods calculate the distance and angle to the object based on the reflected signals [19]. However, these methods suffer from cumulative drift due to limited spatial resolution and have a restricted field of view, resulting in a low object detection rate [20].

Remark. In summary, the lack of low-latency sensors and efficient algorithms makes 3D localization and tracking of the high-speed object challenging.

3D object localization with event and depth. Event cameras, characterized by their asynchronous and motion-activated nature and microsecond-level temporal resolution, are increasingly used in high-frequency detection [21, 22]. This trend motivates our exploration of leveraging these asynchronous measurements to enhance 3D tracking of high-speed object. Although event cameras excel at capturing fast-moving objects within their field of view then reacting fast[23, 24], they face challenges with scale uncertainty, which impedes accurate localization. In this work, we incorporate a depth camera to overcome this limitation and recover the scale in monocular sensing [25].

Albeit inspiring, translating this intuition into a practical 3D object localization system is non-trivial and two technical challenges have to be solved:

C1: Event burst hinders object detection. Event cameras are highly sensitive to light changes and often produce numerous environment-triggered events, leading to event burst [26]. The event burst makes rapid and accurate object detection difficult and poses challenges for registering event camera detection result with depth camera data.

C2: Heterogeneous data impedes sensor fusion. The event camera is sensitive to light changes caused by object motion, generating asynchronous events, while the depth camera provides depth information for each pixel. The data from these sensors pose different precision and density. This spatial heterogeneity presents challenges for data fusion. Additionally, the sensing delays between event and depth cameras vary. This temporal heterogeneity add complexity.

Our work. To tackle the above challenges, we design and implement *EventTracker*, the first framework to effectively fuse event and depth measurements for precise, low-latency 3D localization and tracking of the high-speed dynamic object. EventTracker can be integrated into autonomous vehicles and delivery drones, enabling them to achieve 3D tracking and avoidance of pedestrians and obstacles.

In EventTracker, we first reduce environmental noise's impact on event and depth-based object detection by devising a Collaborative Object Detection and Tracking module. Concurrently, the object's detection by the event camera guides segmentation on the depth map, effectively addressing registration-related challenges. Second, to achieve high-accuracy 3D tracking of the high-speed object, we introduce a graph-instructed localization algorithm. This algorithm harmonizes heterogeneous observations from both cameras through joint optimization in a tightly coupled manner. Also, extensive experiments in indoor environments are conducted to comprehensively evaluate performance of EventTracker, which show that the localization performance of our method improves 38.5% on average compared with the baseline.

The contributions of this paper are as follows:

- We propose *EventTracker*, the first framework that effectively fuses event and depth measurements for precise, low-latency 3D tracking of the high-speed dynamic object.
- We introduce a collaborative object detection and tracking algorithm that fully leverages depth and event measurements for object detection.
- We design a novel graph-instructed optimization algorithm that fuses heterogeneous data with different spatial and temporal resolutions, achieving continuous 3D location estimation for moving objects.
- We implement EventTracker and conduct extensive quantitative and qualitative experiments in dynamic scenarios and noisy environments to validate our method.

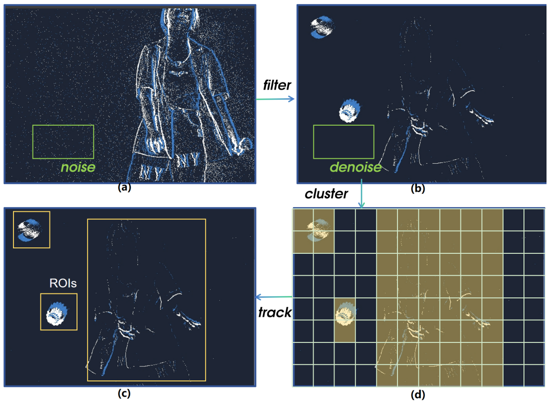


Figure 3: Illustration of event tracker.

The rest of this paper is organized as follows: Section 2 provides an overview of EventTracker. Section 3 details our system design and methodology. In Section 4, we present experimental results for performance evaluation. Finally, Section 5 concludes the paper.

2 OVERVIEW

Our framework is illustrated in Fig.2. This framework consists of two parts: collaborative object detection and tracking, and graph-instructed 3D position estimation.

Firstly, we handle cross-model data separately with the event tracker and depth tracker. The event tracker accumulates events, filters out noise, and employs a grid-based clustering algorithm to quickly locate the moving object, followed by EKF-based tracking. The depth tracker then receives several region of interest, and use its depth information to target the wanted region while providing its position on the third axis. Also, a continuous mutual check will be attached to the system for denoise and target association.

Secondly, graph optimization integrates data from event, depth, and motion factors. By leveraging residuals and our designed cost functions, we fuse these heterogeneous observations to optimize 3D position estimation.

3 SYSTEM DESIGN

In this section, we detail the system overview and the integrated workflow for robust object tracking through event and depth data processing. Section 3.1 details the design of the collaborative object detection and tracking component. Section 3.2 discusses the jointly graph-instructed estimation optimizer proposed in our system.

3.1 Collaborative Object Detection and Tracking

In this section, we transform image and event input into positional values. The event tracker and depth tracker operate independently to compute positions in three axes and perform continuous mutual filtering.

3.1.1 ***Event Tracker.*** EventTracker is designed to continuously monitor and track the position coordinates of moving objects within rapidly generated event streams. The raw generated event is susceptible to noise (Fig.3a).

Filter. We initially employ threshold filtering on the time image to preliminarily eliminate some noise. In time image, each pixel represents the average timestamp of all events occurring at that pixel. It allows for the event data to be dimensionally reduced to a more manageable 2D image while retaining its temporal factor.

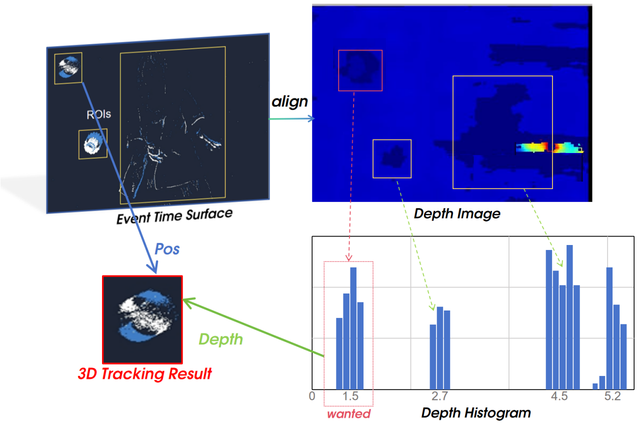


Figure 4: Illustration of depth tracker.

Specifically, for each pixel (i, j), the value T_{ij} in its time image T is calculated. This value is the average of the timestamps of all events on that pixel, i.e.,

$$T_{ij} = \frac{1}{I_{ij}} \sum_{t:t\in\xi_{ij}} t$$

,where I_{ij} represents the number of events mapped to pixel (i, j), and ξ_{ij} represents the set of events on the pixel.

Since T is generated based on the time information of the event stream, an increase in pixel value typically indicates the presence of more moving objects or activity at that location. By applying threshold filtering, we can roughly filter the noise out. Fig.3b shows the denoised event plane.

Cluster. Then we segregate and cluster the moving objects from the denoised event. We leverage a grid-based clustering approach, partitioning the field of view (FOV) of the camera into fundamental cells using a regular grid, as shown in Fig.3c. Within a specific time slice, a cell that hosts a variety of events surpassing a predefined threshold can be designated as an active cell. The adjacent active cells will be aggregated into one cluster, a box. With a structured grid in partitioning, the approach reduces algorithmic complexity and conserves resources, laying the foundation for tracking and segmentation algorithms.

EKF-based Tracking. After clustering, each cluster is associated with the nearest tracker box in Fig.3d based on distance calculations. A simple motion model predicts the initial position of the tracking target for preliminary location estimation. This approximation provides an initial forecast, enhanced later by the Extended Kalman Filter (EKF) for more precise predictions and updates over subsequent time steps. The EKF, integrated with motion and observatory data, continually refines the estimated target state, utilizing the Kalman gain to effectively handle system noise and improve accuracy in target state estimation.

3.1.2 ***Depth Tracker.*** Through the aforementioned algorithm, we obtain the positional values of moving objects from event. The detection results are output in the form of rectangular boxes. The center of the box is set as the center point of moving object. However, the detection results still contain a substantial number of moving objects owing to their exposure to cacophonous environments and the impact from multiple moving items. Depth information is subsequently introduced for further filtration and depth perception.

As depicted in Fig.4, boxes from the event slice are projected onto the depth map plane to generate ROIs, whose depth information are then harnessed for filtration and precise tracking of specific object. Theoretically, it is possible to simultaneously track multiple moving objects. The focus of our implementation is the process of associating a box with depth segmentation.

Firstly, the depth image is registered to the event plane according to the intrinsic and extrinsic matrices. Next, we project the ROI boxes onto the depth image and utilize the depth histogram to locate the object at the wanted depth.Once the target box is determined, the event tracker will categorize the others as noise, concentrating exclusively on tracking the selected box.

Hence, the 2D position of the object and its depth have been collaboratively estimated. Throughout this process, data from event and depth undergo mutual filtering to achieve the objective of tracking a specific object.

3.1.3 *Vision-based Position Estimation*.

We assume the event camera follows a conventional pinhole camera model, neglecting distortion errors. The projection function $\pi : \mathbb{R}^3 \longrightarrow \Omega$ converts a 3D point X_E in camera reference coordinate E into a 2D pixel x in the image plane, where $x \in \Omega \subset \mathbb{R}^2$. Specifically,

$$\pi(X_E) = \begin{bmatrix} f_x \frac{X_E}{Z_E} + c_x \\ f_y \frac{Y_E}{Z_E} + c_y \end{bmatrix}, \quad X_E = \begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix}$$

where f_x and f_y represent the camera focal lengths, and c_x and c_y are the principal points. The center of boxes detected in collaborative tracker is our observed point. We can estimate the object's preliminary location under E at time t_i with the center point of bounding box proposal as

$$x^{t_i} = \pi\left(X_{\mathrm{E}}^{t_i}\right) + v^{t_i} = \pi\left(X_0^{t_i} + t_{\mathrm{EO}}^{t_i}\right) + v^{t_i}$$

where $X_0^{t_i}$ represents the corresponding 3D point of center point x^{t_i} in the object reference, and v^{t_i} denotes the random noise of center point.

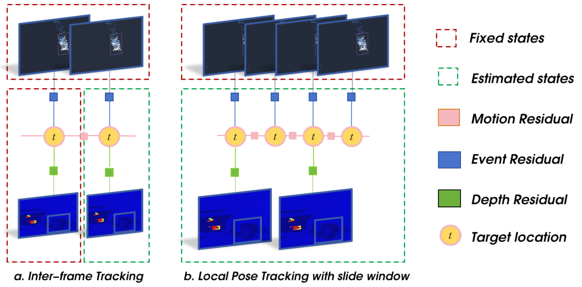
3.2 Graph-Instructed 3D Position Estimation

Due to the disparate frequencies of input sources from the event camera and the depth camera, a straightforward superimposition is not feasible. Therefore, we propose a graph-instructed 3d position estimation optimization framework.

As illustrated in Fig.5, we have developed inter-frame tracking and long-term local pose tracking systems to achieve continuous and highly precise position estimation. Inter-frame tracking integrates residuals from multiple modalities to estimate the next position, while local pose tracking employs a sliding window for joint optimization across multiple poses, ensuring a smooth trajectory.

3.2.1 *Inter-frame tracking*.

The optimization problem can be formulated as follows:



a. Inter-frame Tracking

b. Local Pose Tracking with slide window

Figure 5: Illustration of graph optimization.

$$\chi^* = \arg\min_{\chi} \sum_i \left(E_{\text{depth}}^{ti} + E_{\text{proj}}^{ti}\right)$$

where E_{depth}^{ti} and E_{depth}^{ti} represent the event and depth errors.

Event residual The event residual arises from the projection error when the estimated 3D position is converted to the 2D plane of the event camera. In the Fast Tracking section, we acquire observation coordinates x^{t_i} of detected moving objects at time t_i.

Now, we need to perform a coordinate system transformation to project the estimated three-dimensional position onto imaging plane of the event camera.

Let X_O^{ti} be the three-dimensional position of the moving object at time t_i. And the event camera projection error term can be given as:

$$X_{\mathrm{E}}^{t_i} = R_{\mathrm{EO}}\left(X_O^{t_i}(k)\right) + \boldsymbol{t}_{\mathrm{EO}}$$

$$\boldsymbol{E}_{\text{proj}}^{t_i} = \rho\left(\left\|x^{t_i} - \pi\left(X_{\mathrm{E}}^{t_i}\right)\right\|_{\Omega_E}^2\right)$$

Note that the $\rho()$ indicates the Huber loss function used here to increase its resilience to outliers.

Depth residual Acquiring the depth residual is more straightforward. It is the distance error between the estimated position and the depth observation. The depth camera distance error can be given as:

$$\begin{aligned} X_{\mathrm{D}}^{t_i} &= [x_{\mathrm{D}}^{t_i}, y_{\mathrm{D}}^{t_i}, z_{\mathrm{D}}^{t_i}] \\ &= R_{\mathrm{DE}}^{t_i}\left(X_O^{t_i}(k) + \boldsymbol{t}_{\mathrm{EO}}\right) + \boldsymbol{t}_{\mathrm{DE}} \end{aligned}$$

$$\boldsymbol{E}_{\text{depth}}^{t_i} = \rho\left(\left\|d^{t_i} - z_{\mathrm{D}}^{t_i}\right\|_{\Omega_D}\right)$$

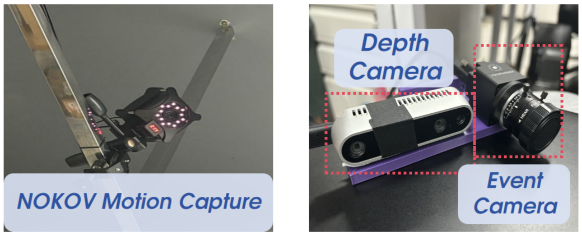
Motion model Similar to various modern SLAM systems, we utilize a constant velocity model to enforce constraints between the current object position and the previous position.

$$\bar{t}_{EO}^{ti} - \bar{t}_{EO}^{ti-1} = \bar{t}_{EO}^{ti-1} - \bar{t}_{EO}^{ti-2}.$$

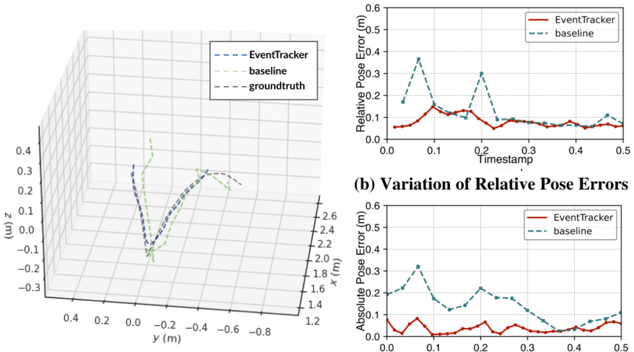
We primarily use a simple motion model to perform prior position estimation. After initial estimation, we employ the Levenberg-Marquardt algorithm to minimize the sum of residuals, thereby solving out the parameters x^{t_i}that fit best.

3.2.2 *Local Pose Tracking with slide window*.

As shown in Fig.5 , approximately every few seconds (typically, a keyframe is selected), the local pose tracking corrects accumulated errors. Using a sliding window approach, it processes all frames following the



(a) (b)



(a) Trajectory Comparison

(b) Variation of Relative Pose Errors

(c) Variation of Absolute Pose Errors

Figure 7: Overall performance of trajectory error.

most recent keyframe as input and optimizes pose jointly. Denote the set of frames as $\mathcal{F}$, the optimization problem is formulated as:

$$\chi^* = \arg\min_{\chi} \sum_{i\in\mathcal{F}} \left(E^i_{\text{depth}} + E^i_{\text{proj}}\right)$$

where E^i_{depth} and E^i_{depth} represent the event and depth errors.

4 EVALUATION

4.1 Experimental Setup

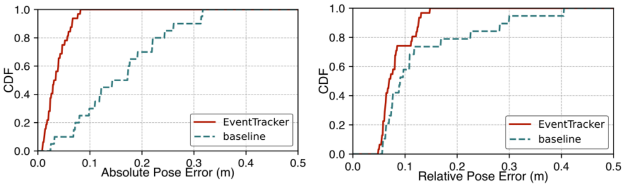
Setting To validate our proposed approach, we conducted experiments in a laboratory setting. We utilized a Intel Realsense D435i depth camera and a Prophesee EVK4 HD event camera for data collection (Fig.6a). Meanwhile, we employed a motion capture system with fourteen NOKOV cameras (Fig.6b) to collect ground truth at the frequency of 240hz. In our experiment, we aimed to validate our 3D tracking system by estimating the position and trajectory of dynamically moving objects such as balls and drones. These objects exhibited different and random movement patterns, providing a robust test for our tracking algorithm.

Evaluation Metrics Relative Pose Error (RPE) and Absolute Pose Error (APE) are employed as evaluation metrics to separately assess the local and global localization accuracy of the tracking algorithm.

Baseline We benchmark our method against another event-based fast-moving object detection approach[23]. It leverages temporal information from asynchronous event streams for object detection in the event plane and incorporates additional size information to estimate depth.

4.2 Overall Performance

Here, we showcase the overall performance in ball-throwing scenarios. The comparison of results highlights EventTracker's superiority over the baseline algorithm across all evaluation metrics. EventTracker achieves high accuracy consistently at high frequencies.



(a) Cumulative Distribution of APE (b) Cumulative Distribution of RPE

Figure 8: Overall performance in CDF.

From Fig.7, it is visually evident that our method more accurately fits the trajectory curve compared to the continuous positions estimated by the baseline method. Though, there is a slight loss of accuracy when the object changes its direction, our method closely tracks the ground truth for most of the time. Also, we plot the relative pose error and absolute pose error along with the timestamp in Fig.7b and Fig.7c . Our analysis reveals that our method achieves a mean APE of 0.037m and a mean RPE of 0.082m, which improves 38.5% on the average RPE compared with the baseline. Notably, all absolute pose errors for our estimations are less than 8cm, significantly outperforming the baseline, which has pose errors exceeding 20cm. This demonstrates that the long-term sliding windows for collaborative optimization in our design significantly enhance the precision and consistency of estimations over time.

The cumulative distribution function (CDF) graph plotted in Fig.8 of RPE illustrates the error distribution, revealing that approximately 70% of the estimated poses have errors less than 8cm. Compared to the baseline, the cumulative distribution of APE exhibits a sharper rise and a lower error range, indicating the superior performance of our method. Unlike the baseline method, which relies solely on event data, our approach effectively filters out noise and distractions, while reducing residuals from various sources, thereby achieving greater precision.

5 CONCLUSION

In this paper, we propose a Collaborative 3D Object Tracking System framework that leverages event and depth measurements for tracking moving objects. We design both an event tracker and a depth tracker to detect and track objects collaboratively. To fuse the heterogeneous data, we develop a graph-based 3D position estimation technique. Experimental results demonstrate that our method achieves superior high-speed and high-accuracy tracking performance. Such high-precision object pose estimation can also be applied to swarm control, where many algorithms rely on UAVs with strong perception capabilities [27, 28]. The integration of event cameras offers significant advancements in this area, which is one of our future research directions.

ACKNOWLEDGMENTS

This paper was supported by the National Key R&D program of China (2022YFC3300703), the Natural Science Foundation of China under Grant 62371269, Guangdong Innovative and Entrepreneurial Research Team Program (2021ZT09L197) and Meituan

REFERENCES

[1] Yuxuan Liu, Haoyang Wang, Fanhang Man, Jingao Xu, Fan Dang, Yunhao Liu, Xiao-Ping Zhang, and Xinlei Chen. Mobiair: Unleashing sensor mobility for city-scale and fine-grained air-quality monitoring with airbert. In *Proceedings of the 22nd ACM MobiSys*, pages 223–236, 2024.

[2] Guoguang Du, Kai Wang, Shiguo Lian, and Kaiyong Zhao. Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. *Artificial Intelligence Review*, 54:1677 – 1734, 2020.

[3] Zuxin Li, Fanhang Man, Xuecheng Chen, Baining Zhao, Chenye Wu, and Xinlei Chen. Tract: Towards large-scale crowdsensing with high-efficiency swarm path planning. In *Adjunct Proceedings of the 2022 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2022 ACM International Symposium on Wearable Computers*, UbiComp/ISWC '22 Adjunct, page 409–414, New York, NY, USA, 2023. Association for Computing Machinery.

[4] Zuxin Li, Fanhang Man, Xuecheng Chen, Susu Xu, Fan Dang, Xiao-Ping Zhang, and Xinlei Chen. Quest: Quality-informed multi-agent dispatching system for optimal mobile crowdsensing. In *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications*, pages 1811–1820, 2024.

[5] ZiYun Wang, Fernando Cladera Ojeda, Anthony Bisulco, Daewon Lee, Camillo Jose Taylor, Kostas Daniilidis, M. A. Hsieh, Daniel D. Lee, and Volkan Isler. Ev-catcher: High-speed object catching using low-latency event-based neural networks. *IEEE Robotics and Automation Letters*, 7:8737–8744, 2022.

[6] Zhuozhu Jian, Zejia Liu, Haoyu Shao, Xueqian Wang, Xinlei Chen, and Bin Liang. Path generation for wheeled robots autonomous navigation on vegetated terrain. *IEEE Robotics and Automation Letters*, 9(2):1764–1771, 2024.

[7] Xinlei Chen, Aveek Purohit, Carlos Ruiz Dominguez, Stefano Carpin, and Pei Zhang. Drunkwalk: Collaborative and adaptive planning for navigation of micro-aerial sensor swarms. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, SenSys '15, page 295–308, New York, NY, USA, 2015. Association for Computing Machinery.

[8] Jingao Xu, Danyang Li, Zheng Yang, Yishujie Zhao, Hao Cao, Yunhao Liu, and Longfei Shangguan. Taming event cameras with bio-inspired architecture and algorithm: A case for drone obstacle avoidance. *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, 2023.

[9] Xinlei Chen, Carlos Ruiz, Sihan Zeng, Liyao Gao, Aveek Purohit, Stefano Carpin, and Pei Zhang. H-drunkwalk: Collaborative and adaptive navigation for heterogeneous mav swarm. *ACM Trans. Sen. Netw.*, 16(2), apr 2020.

[10] Xuecheng Chen, Zijian Xiao, Yuhan Cheng, ChenChun Hsia, Haoyang Wang, Jingao Xu, Susu Xu, Fan Dang, Xiao-Ping Zhang, Yunhao Liu, et al. Soscheduler: Toward proactive and adaptive wildfire suppression via multi-uav collaborative scheduling. *IEEE Internet of Things Journal*, 2024.

[11] A. N. Wilson, Abhinav Kumar, Ajit Jha, and Linga Reddy Cenkeramaddi. Embedded sensors, communication technologies, computing platforms and machine learning for uavs: A review. *IEEE Sensors Journal*, 22:1807–1826, 2022.

[12] Xuecheng Chen, Haoyang Wang, Zuxin Li, Wenbo Ding, Fan Dang, Chengye Wu, and Xinlei Chen. Deliversense: Efficient delivery drone scheduling for crowdsensing with deep reinforcement learning. In *Proceedings of the 2022 ACM UbiComp*, pages 403–408, 2022.

[13] Yang Hong, Kaixuan Wei, Linwei Chen, and Ying Fu. Crafting object detection in very low light. In *BMVC*, volume 1, pages 1–15, 2021.

[14] Shoujie Li, Zihan Wang, Changsheng Wu, Xiang Li, Shan Luo, Bin Fang, Fuchun Sun, Xiao-Ping Zhang, and Wenbo Ding. When vision meets touch: A contemporary review for visuotactile sensors from the signal processing perspective. *IEEE J-STSP*, 2024.

[15] Safouane El Ghazouali, Youssef Mhirit, Ali Oukhrid, Umberto Michelucci, and Hichem Nouira. Fusionvision: A comprehensive approach of 3d object reconstruction and segmentation from rgb-d cameras using yolo and fast segment anything. *Sensors (Basel, Switzerland)*, 24, 2024.

[16] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios. *IEEE Robotics and Automation Letters*, 3(2):994–1001, 2018.

[17] Tiantian Liu, Chao Wang, Zhengxiong Li, Ming-Chun Huang, Wenyao Xu, and Feng Lin. Wavoice: An mmwave-assisted noise-resistant speech recognition system. *ACM Transactions on Sensor Networks*, 20(4):1–29, 2024.

[18] Yi Zhu, Chenglin Miao, Hongfei Xue, Zhengxiong Li, Yunnan Yu, Wenyao Xu, Lu Su, and Chunming Qiao. Tilemask: A passive-reflection-based attack against mmwave radar object detection in autonomous driving. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 1317–1331, 2023.

[19] Jinrui Zhang, Huan Yang, Ju Ren, Deyu Zhang, Bangwen He, Ting Cao, Yuanchun Li, Yaoxue Zhang, and Yunxin Liu. Mobidepth: Real-time depth estimation using on-device dual cameras. In *Proceedings of the 28th ACM MobiCom*, pages 528–541, 2022.

[20] Mircea Paul Muresan and Sergiu Nedevschi. Multimodal sparse lidar object tracking in clutter. *2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 215–221, 2018.

[21] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbrück, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. *The International Journal of Robotics Research*, 36:142 – 149, 2016.

[22] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020.

[23] Davide Falanga, Kevin Kleber, and Davide Scaramuzza. Dynamic obstacle avoidance for quadrotors with event cameras. *Science Robotics*, 5(40):eaaz9712, 2020.

[24] Haoyang Wang, Xinyu Luo, Ciyu Ruan, Xuecheng Chen, Wenhua Ding, Yuxuan Liu, and Xinlei Chen. Poster: Fusing event and depth sensing for dynamic objects localization and tracking. In *Proceedings of the 25th International Workshop on Mobile Computing Systems and Applications*, HOTMOBILE '24, page 141, New York, NY, USA, 2024. Association for Computing Machinery.

[25] Yihao Zuo, Jiaqi Yang, Jiaben Chen, Xia Wang, Yifu Wang, and Laurent Kneip. Devo: Depth-event camera visual odometry in challenging conditions. *2022 International Conference on Robotics and Automation (ICRA)*, pages 2179–2185, 2022.

[26] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:1964–1980, 2019.

[27] Haoyang Wang, Jingao Xu, Chenyu Zhao, Zihong Lu, Yuhan Cheng, Xuecheng Chen, Xiao-Ping Zhang, Yunhao Liu, and Xinlei Chen. Transformloc: Transforming mavs into mobile localization infrastructures in heterogeneous swarms, 2024.

[28] Haoyang Wang, Xuecheng Chen, Yuhan Cheng, Chenye Wu, Fan Dang, and Xinlei Chen. H-swarmloc: efficient scheduling for localization of heterogeneous mav swarm with deep reinforcement learning. In *Proceedings of the 20th ACM SenSys*, pages 1148–1154, 2022.