Event-Aided Time-to-Collision Estimation for Autonomous Driving - Supplementary Materials -

Jinghang Li[1⋆], Bangyan Liao[2⋆], Xiuyuan LU[3], Peidong Liu[2], Shaojie Shen[3], and Yi Zhou[1✉]

[1] School of Robotics, Hunan University
[2] School of Engineering, Westlake University
[3] Department of ECE, Hong Kong University of Science and Technology

Overview

We provide more details about the submission in this document, which includes:

- A justification of key insights from a toy experiment. (Sec. I)
- An in-depth look at the three datasets we generate/collect. (Sec. II)
- More detailed results on real-world datasets. (Sec. III)
- An additional discussion on the performance of all involved methods for comparison, explaining the number reported in the paper according to an ablation study. (Sec. IV)

I Justification of Key Insights

As shown in [5], the success of geometric-model fitting to event data hinges on using an accurate parametric model. To this end, we develop a time-variant affine model that captures the true flow-field dynamics. To justify, we add a comparison of our proposed time-variant affine model against the widely used constant affine model, and also, the simplified affine model (assuming no horizontal motion) by *ECMD* [9]. The evaluation metric used is the contrast of the resulting image of warped events (IWE), and the goodness of fitting can also be assessed qualitatively from the IWE's sharpness. As seen in Fig. A, our method outperforms the others, justifying the key insight of our method.

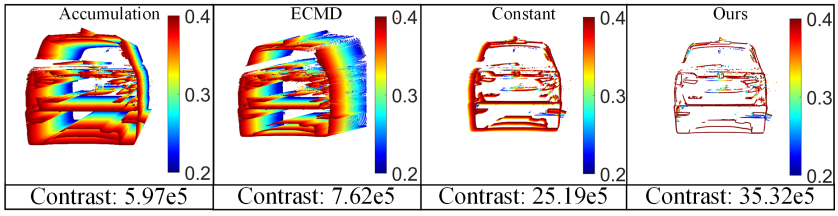


Fig. A: Comparison of using different affine models.

Fig. B: A snapshot of three datasets. From top to bottom: *Synthetic* dataset, *Slider* dataset, and *FCWD* dataset. From left to right: platform configuration, third-person view of the TTC scenarios, intensity images, and event data (represented with a naive accumulation of events).

II Dataset

We elaborate on the three datasets introduced in Section 4.2 of the paper. Our investigation reveals a significant lack of specific event datasets for the task of time-to-collision estimation. To this end, we create three platforms (1 virtual + 2 real) for data generation, and they consist of: 1) A customized virtual environment that synthesizes data in traffic scenes (II.A); 2) A small-scale test platform that mimics the discussed scenarios (II.B); and 3) A multi-sensor suite employed on a real car for data collection (II.C). Fig. B provides a snapshot of the platform and generated data for each dataset, and Tab. 1 details the configuration of the sensors used. We detail the data generation process for each dataset in the following.

II.A Synthetic Dataset

The simulation of forward-collision scenarios is built on top of CARLA [4], an open-source simulation platform. CARLA offers extensive APIs for customizing vehicle motions and scenes, supporting a wide array of sensors commonly used in robotics, including RGB cameras, LIDAR, IMU, and also, event-based cameras. Three subsets are created, and each one features distinct motion patterns and scenes. We set the synthesized event camera's parameters as follows. The eps value is set to 0.3, the refractory period is 1e−5 seconds, and the contrast

Table 1: Hardware specifications for our datasets.

Dataset Name	Sensor Type	Rate	Specifications	Hardware-level Sync.
Synthetic	Carla DVS	N/A	640 × 480 pixels FoV: 52°H / 40°V	-
	Carla RGB	30Hz	640 × 480 pixels FoV: 52°H / 40°V	-
	Carla Traffic Manager	1000Hz	Report global location of all vehicle actors	-
Slider	Inivation DVXplorer	N/A	640 × 480 pixels FoV: 20°H / 15°V	✔
	DAHENG MER2	25Hz	1440 × 1080 pixels FoV: 17°H / 13°V color with global shutter	✔
	Encoder of slider motor	100KHz	Report the position and velocity of the hybrid optical system on the slider.	✔
FCWD	2×Prophesee EVKv4 baseline: 7.5cm	N/A	1280 × 720 pixels FoV: 22°H / 12°V	✔
	2×FLIR Blackfly S baseline: 7.5cm	20Hz	1920 × 1200 pixels FoV: 23°H / 15°V color with global shutter	✔
	LiDAR: Livox HAP	point rate: 452,000 points/s frame rate: 10Hz	range 150 m @ 10% reflectivity FoV: 120°H / 25°V ±3cm range precision @ 20 m	✔

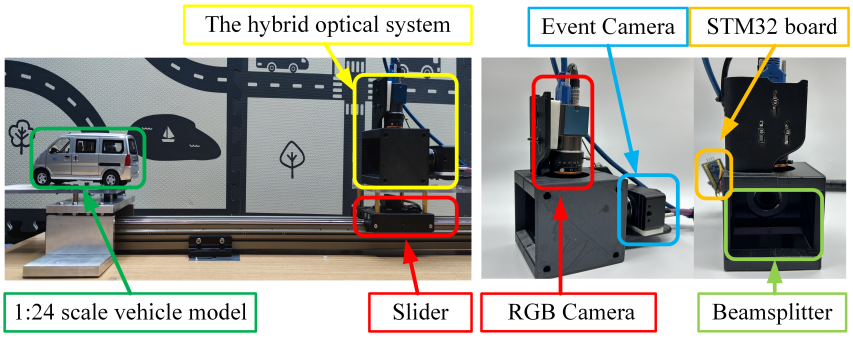
threshold is 0.15. The calculation of the groundtruth TTC is based on the absolute distance and relative speed between the host vehicle and the preceding one.

II.B Slider Dataset

To narrow the gap between simulated data and real-world ones, we design a small-scale test platform using a miniature replica of real vehicles to simulate car crash scenarios. As shown in Fig. C, this test platform is composed of a motorized slider, a hybrid optical system based on an open-source design in [7], and a 1:24 scale vehicle model. The hybrid optical system consists of an inivation DVXplorer event camera of $640\times480px$ resolution, an RGB camera with a spatial resolution of $1440 \times 1080px$, and a beamsplitter that divides incoming light into two paths, ensuring a unified field of view for both cameras. For precisely identifying event points within the bounding box of the leading vehicle, it is important to establish a pixel-to-pixel correspondence map, and furthermore, synchronize the time clocks of the two heterogeneous sensors. This pixel-to-pixel mapping between the two cameras can be established through an offline calibration scheme. To temporally synchronize these cameras, we use an STM32 development board, which sends a 25-Hz clock signal that triggers both cameras, ensuring a precise synchronization. The hybrid optical system, mounted on a slider, simulates collisions at three different speeds (*i.e.*, 500 mm/s, 750 mm/s, and 1000 mm/s, respectively). The groundtruth TTC is derived from the slider's position along the rail and its speed measured by the motor encoder.

II.C Forward Collision Warning Dataset (FCWD)

Publicly available datasets for autonomous driving using event cameras fall short of addressing the specific requirements of TTC estimation research. As listed in



(a) The proposed small-scale test platform. **(b)** The composition of the hybrid optical system.

Fig. C: Illustration of our small-scale test platform and hybrid optical system.

Table 2: Comparison of Different Event-Centric Datasets.

Dataset	Event Resolution	HFOV	Urgent Brake	Event Stereo	RGB Stereo	LiDAR	Sync.
MVSEC [13]	346 × 260	65°	✖	✔	✔	✔	Partially
DSEC [6]	640 × 480	60°	✖	✔	✔	✔	Fully
ViViD++ [8]	640 × 480	44°	✖	✖	✖	✔	Partially
M3ED [2]	1280 × 720	63°	✖	✔	✔	✔	Fully
Ours(FCWD)	1280 × 720	22°	✔	✔	✔	✔	Fully

Table 2, these datasets either exhibit low spatial resolution (*e.g.*, 346x260 or 640x480) or lack comprehensive temporal synchronization among sensors. Additionally, these datasets are predominantly designed for tasks such as Simultaneous Localization and Mapping (SLAM) and object detection, rather than for TTC estimation. To achieve a wider Horizontal Field of View (HFOV), all datasets utilize short focal length lenses for both event cameras and RGB cameras. Moreover, the host vehicle, on which these sensors are mounted, typically decelerates in advance to keep a safe distance from the leading vehicle.

To this end, we develop a multi-sensor suite incorporating a stereo event camera, a stereo RGB camera, and a LiDAR. Specifically, we employ a HD-resolution (720p) event camera Prophesee EVKv4 to record event data. The event camera is synchronized with other sensors to millisecond accuracy via a triggering signal from an STM-32 development board. The RGB camera and the LiDAR are synchronized to sub-millisecond accuracy using the Precision Time Protocol (PTP). Additionally, we equip event cameras and RGB cameras with telephoto lenses, suitable for data collection in forward-collision scenarios. Note that we only use one event camera and one RGB camera in this work.

Three sequences were recorded using our multi-sensor suite to evaluate our algorithm in real-world settings. The multi-sensor system is mounted on the host vehicle's engine cover with suction cups. The vehicle is driven towards a station-

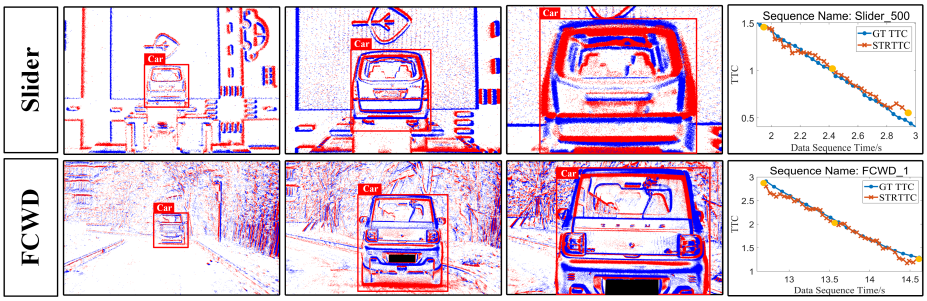


Fig. D: Illustration of the input (event data and the bounding box) and a continuous estimation results of TTC. From left to right: Three selected views of input data in a chronological order and the TTC estimation results through the whole process. Note that the first three columns correspond to the positions highlighted with yellow dots in the right-most plots.

ary one ahead, breaking only at the minimum safety distance, as depicted in the multimedia content. Groundtruth TTC values are obtained from the distance to the leading vehicle, measured by the LiDAR, and the vehicle's speed, determined by the LiDAR-inertial odometry (Fast-lio [12]).

III Detailed Real-World Experiments of TTC Estimation

We provide more detailed results on our real data (*Slider* and *FCWD*). As shown in Fig. D, our results are always consistent with the ground truth. We observe that the TTC results become increasingly accurate as the distance between the host vehicle and the leading vehicle decreases. This happens due to the fact that the contour of the leading vehicle on the image plane enlarges, and it will generate more event data, leading to linear time surfaces (LTS) with more spatio-temporal information. We contend that improvements are feasible with further engineering efforts. Given the current runtime statistics for a single computation, an updating rate of 200 Hz, as claimed in the paper, can be achieved.

IV Discussion on Accuracy and Efficiency

This section elaborates on detailed configuration of each method we compare against in Section 4.3 of the paper. The configuration and parameter selection for each method significantly affect the accuracy of TTC calculations and runtime. For a fair comparison, Table. 3 shows the result of each method under various parameter settings on our FCWD dataset. The runtime reported in the table represents the average computation time to get one TTC result on the corresponding sequence.

Image's FoE [11] takes as input the intensity images from an RGB camera running at 10 Hz. Within the bounding box, the SURF [1] algorithm is employed

Table 3: Quantitative Analysis of Each Method Under Different Parameters. Lower is better.

	Method	*FCWD_1*		*FCWD_2*		*FCWD_3*	
		e_{TTC}	Runtime (s)	e_{TTC}	Runtime (s)	e_{TTC}	Runtime (s)
	Image's FoE [11]	5.39%	0.017	5.70%	0.018	6.02%	0.017
	FAITH [3]	25.49%	0.214	27.91%	0.203	39.15%	0.210
Event Num $1e5$	CMax [5]	5.68%	1.73	5.65%	1.93	6.92%	1.79
Event Num $1e5$	Our Init + CMax [5]	5.85%	1.39	4.05%	1.61	4.52%	1.47
Event Num $2e5$	CMax [5]	2.42%	2.53	2.39%	3.12	3.04%	3.34
Event Num $2e5$	Our Init + CMax [5]	2.33%	1.94	2.26%	2.38	2.97%	2.59
Event Num $3e5$	CMax [5]	6.04%	2.93	4.37%	2.83	3.37%	2.83
Event Num $3e5$	Our Init + CMax [5]	2.10%	2.13	2.12%	1.99	2.57%	2.14
Neighboring Size $s=2$	ETTCM Scaling [10]	15.52%	0.307	18.45%	0.282	19.08%	0.295
Neighboring Size $s=2$	ETTCM Translation [10]	19.44%	0.839	17.30%	0.772	21.20%	0.806
Neighboring Size $s=2$	ETTCM 6-DOF [10]	28.00%	1.09	27.44%	0.997	36.78%	1.05
Neighboring Size $s=3$	ETTCM Scaling [10]	28.07%	0.526	32.04%	0.484	266.50%	0.458
Neighboring Size $s=3$	ETTCM Translation [10]	124.72%	1.53	117.06%	1.41	109.84%	1.44
Neighboring Size $s=3$	ETTCM 6-DOF [10]	25.23%	2.02	32.52%	1.82	36.43%	1.91
	Ours	9.84%	0.017	11.55%	0.023	14.09%	0.025

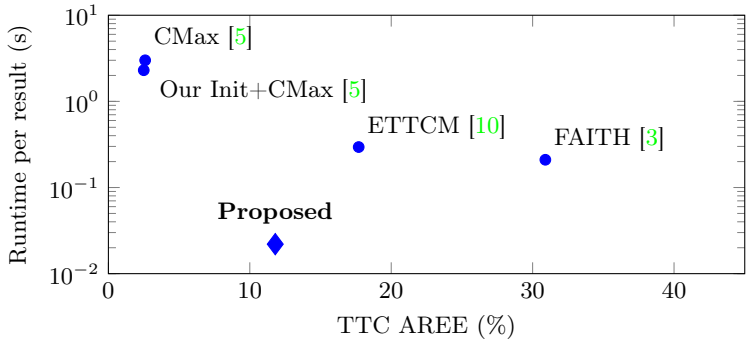


Fig. E: Runtime and Accuracy Comparison of event based TTC Estimation Methods: An Average Performance Evaluation on our FCWD.

to extract feature points on the lead vehicle in two consecutive frames. Based on the matched feature points, the affine motion model is estimated to calculate the TTC. The runtime includes feature extraction and matching between two consecutive frames, as well as the affine model fitting and the TTC calculation. Note that the runtime does not include the system latency caused by the time interval between two successive exposures.

In FAITH [3], we employ default parameters from its open-source code. Events triggered within the bounding box are used as input, and the runtime represents the average time for each result.

For the methods of CMax [5] and Our Init + CMax, the main factor affecting the accuracy of TTC estimation and computation time is the number of events

involved. There are two main strategies: using a fixed temporal window or a constant number of events. In the forward-collision scenario, the number of events within the bounding box fluctuates significantly over time with a fixed time interval on our high-resolution event camera. At the beginning of each sequence, the camera is far from the leading vehicle and only a small number of events are triggered, resulting in large estimation error. As the leading vehicle gets closer, an increasingly larger number of events are generated in a short period of time, resulting in long computation time or even an abortion of the algorithm. Therefore, we choose the second strategy, *i.e.*, using a constant event number of events. Table. 3 shows the estimation error under different event numbers. To seek the balance between the computation time and estimation accuracy, we report the result of $2e^5$ in the experiment result.

The ETTCM [10] method supports different motion models and neighboring sizes. We follow evaluate its performance under different configurations. Our evaluation tries three motion models with a neighboring sizes of 2 and 3, respectively, looking for a comprehensive performance in terms of accuracy and efficiency. The combination of a scaling model and a neighbouring size of 2 is selected in the report of our paper. The ETTCM method estimates the TTC and reports computation time on a per-event basis. To simplify comparisons, we define the computation time of the ETTCM method as the multiplication of the time required by a single computation and the total number of events processed by our method every time. This offers a standardized metric for comparing the runtime of our approach against that of the ETTCM method.

Figure E presents a comparison of runtime and accuracy for all event-based TTC estimation methods, indicating our method achieving a state-of-the-art performance.

References

1. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. Comput. Vis. Image. Und. **110**(3), 346–359 (2008)
2. Chaney, K., Cladera, F., Wang, Z., Bisulco, A., Hsieh, M.A., Korpela, C., Kumar, V., Taylor, C.J., Daniilidis, K.: M3ed: Multi-robot, multi-sensor, multi-environment event dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 4015–4022 (June 2023)
3. Dinaux, R., Wessendorp, N., Dupeyroux, J., De Croon, G.C.: Faith: Fast iterative half-plane focus of expansion estimation using event-based optic flow. IEEE Robotics and Automation Letters **6**(4), 7627–7634 (2021)
4. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. In: Conference on robot learning. pp. 1–16. PMLR (2017)
5. Gallego, G., Rebecq, H., Scaramuzza, D.: A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 3867–3876 (2018)
6. Gehrig, M., Aarents, W., Gehrig, D., Scaramuzza, D.: Dsec: A stereo event camera dataset for driving scenarios. IEEE Robotics and Automation Letters **6**(3), 4947–4954 (2021)

7. Hidalgo-Carrió, J., Gallego, G., Scaramuzza, D.: Event-aided direct sparse odometry. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5781–5790 (2022)
8. Lee, A.J., Cho, Y., Shin, Y.s., Kim, A., Myung, H.: Vivid++: Vision for visibility dataset. IEEE Robotics and Automation Letters **7**(3), 6282–6289 (2022)
9. McLeod, S., Meoni, G., Izzo, D., Mergy, A., Liu, D., Latif, Y., Reid, I., Chin, T.J.: Globally optimal event-based divergence estimation for ventral landing. In: Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I. pp. 3–20. Springer (2023)
10. Nunes, U.M., Perrinet, L.U., Ieng, S.H.: Time-to-contact map by joint estimation of up-to-scale inverse depth and global motion using a single event camera. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 23653–23663 (2023)
11. Stabinger, S., Rodriguez-Sanchez, A., Piater, J.: Monocular obstacle avoidance for blind people using probabilistic focus of expansion estimation. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1–9. IEEE (2016)
12. Xu, W., Zhang, F.: Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. IEEE Robotics and Automation Letters **6**(2), 3317–3324 (2021)
13. Zhu, A.Z., Thakur, D., Özaslan, T., Pfrommer, B., Kumar, V., Daniilidis, K.: The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. IEEE Robotics and Automation Letters **3**(3), 2032–2039 (2018)