

EvTTC: An Event Camera Dataset for Time-to-Collision Estimation

Kaizhen Sun^{1,*}, Jinghang Li^{1,*}, Kuan Dai¹, Bangyan Liao², Wei Xiong³ and Yi Zhou^{1,†}

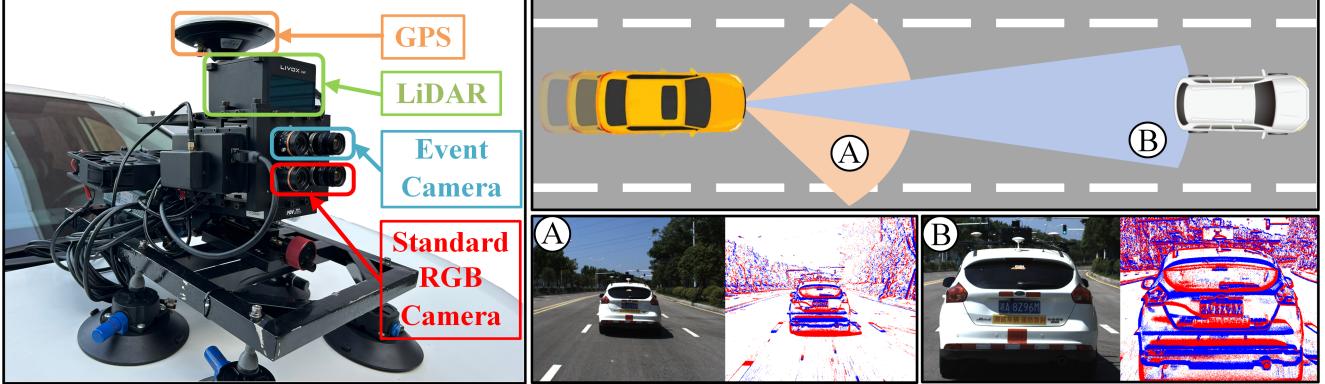


Fig. 1: Left: An overview of the data collection setup. Right: The top image shows a bird's-eye view of the vehicle on the highway, with fields of view and detection distances indicated by circled letters. The bottom row presents intensity images and accumulated event data captured by both the Medium Focal-Length and Telephoto-Lens camera pairs. The font in the figure should not be notably larger than that in the caption!

Abstract—Time-to-Collision (TTC) estimation lies in the core of the forward collision warning (FCW) functionality, which is key to all Automatic Emergency Braking (AEB) systems. Although the success of solutions using frame-based cameras (e.g., Mobileye’s solutions) has been witnessed in normal situations, some extreme cases, such as the sudden variation in the relative speed of leading vehicles and the sudden appearance of pedestrians, still pose significant risks that cannot be handled. This is due to the inherent imaging principles of frame-based cameras, where the time interval between adjacent exposures introduces considerable system latency to AEB. Event cameras, as a novel bio-inspired sensor, offer ultra-high temporal resolution and can asynchronously report brightness changes at the microsecond level. To explore the potential of event cameras in the above-mentioned challenging cases, we propose EvTTC, which is, to the best of our knowledge, the first multi-sensor dataset focusing on TTC tasks under high-relative-speed scenarios. EvTTC consists of data collected using standard cameras and event cameras, covering various potential collision scenarios in daily driving and involving multiple collision objects. Additionally, LiDAR and GNSS/INS measurements are provided for the calculation of ground truth TTC. Considering the high cost of testing TTC algorithms on full-scale mobile platforms, we also provide a small-scale TTC testbed for experimental validation and data augmentation. All the data and the design of the testbed are open sourced, and they can serve as a benchmark that will facilitate the development of vision-based TTC techniques.

Index Terms—Computer Vision for Transportation, Event-based Vision, Time to Collision, Collision Avoidance.

This work was supported by the National Key Research and Development Project of China under Grant 2023YFB4706600.

* Authors contributed equally.

† Corresponding author: Yi Zhou. Email: eeyzhou@hnu.edu.cn.

¹Neuromorphic Automation and Intelligence Lab (NAIL), School of Robotics, Hunan University, Changsha, China.

²School of Engineering, Westlake University, China.

³Xidi Zhijia (Hunan) Co., Ltd., Changsha, China.

I. INTRODUCTION

Time-to-Collision (TTC) refers to the time it takes for two objects to collide under their current speed. Automated Emergency Braking (AEB) systems, which typically include the Forward Collision Warning (FCW) functionality, are now standard features in automobiles. They apply the vehicle’s foundation brakes to prevent a front crash if the driver does not intervene. The key component of FCW is estimating the TTC, which refers to the time it takes for a potential collision under current speed. A large number of technical reports on AEB and FCW have demonstrated that these functionalities reduce front-to-rear crashes by 27%, rear-end collisions by 27%–50%, and rear-end injury crashes by 35%–56% [9], [10], [11], [12].

Compared to LiDAR or radar-based solutions [13], [14], [15], [16], traditional frame-based cameras are a more popular choice due to their low cost. There is extensive research on TTC estimation using traditional cameras [17], [18], [19], [20]. The most common method calculates TTC by analyzing two consecutive images of a monocular camera. The scale of the leading vehicle in the image changes due to its relative motion to the host vehicle, and this variation of scale is used to estimate the TTC. However, this approach is constrained by the frame rate of traditional cameras, which typically operate at 10 Hz in ADAS to balance cost, bandwidth, and power consumption. The resulting 100 ms interval between frames, even before accounting for computation time, can lead to significant latency in FCW and AEB, especially in high-speed scenarios.

Dataset	Frame Camera		Event Camera		Emergency Break	Groundtruth
	Resolution [pix]	Detection Range [m]	Resolution [pix]	Detection Range [m]		
DR (eye) VE [1]	1920×1080	–	–	–	✗	Driver Attention Map
DADA-2000 [2]	1584×660	–	–	–	✓	Driver Attention Map
Crash to Not Crash [3]	710×400	–	–	–	✓	2D Bounding Box
TSTTC [4]	1024×576	[16-96]	–	–	✗	TTC
MVSEC [5]	752×480	28	346×260	12	✗	Depth, GPS
DSEC [6]	1440×1080	80	640×480	35	✗	Depth, RTK GPS
ViViD++ [7]	1280×1024	53	640×480	50	✗	Depth, RTK GPS
M3ED [8]	1280×800	68	1280×720	65	✗	Depth, RTK GPS
FCWD (Ours)	1920×1200	[160-295]	1280×720	[99-197]	✓	TTC, Depth, RTK GPS

TABLE I: Comparison of different event camera datasets in driving scenarios. The symbol “–” indicates not available. Emergency Break indicates a sharp decrease in vehicle speed along with a small TTC value.

Event-based cameras, inspired by the biological mechanisms of the human visual system, operate differently from traditional frame-based cameras. Rather than capturing entire frames, event-based cameras asynchronously report changes in brightness at individual pixels. This approach offers spatio-temporal sparsity, microsecond-level temporal resolution, and high dynamic range, making event-based cameras well-suited for perception tasks involving rapid motion, such as robotics and autonomous driving [21], [22], [23], [24], [25].

Despite the potential of event-based cameras, research in the field of TTC estimation faces significant challenges, particularly due to the lack of suitable datasets focusing on high relative-speed scenarios. To fill this gap, we present a new dataset (see Fig. 1) that can serve as a benchmark featuring collision scenarios with high relative speed, multiple collision targets, and diverse scenes, facilitating the development and evaluation of TTC estimation methods. The benefits brought by the proposed dataset consist of:

- A diverse set of sequences featuring various targets, such as real vehicles, inflatable vehicles, and dummies, across a wide range of relative speeds, including both routine and challenging situations.
- A low-cost and small-scale TTC testbed that facilitates the generation of quasi-real data at different relative speeds. The design of the testbed is open-source.
- A specific benchmark for the TTC task that can serve as an evaluation platform for the community to test and compare different TTC estimation methods.

Outline: The rest of the paper is organized as follows. First, a literature review on datasets of event cameras and those for TTC tasks is provided in Sec. II. Then we provide details of our sensor setup and calibration in Sec. III, followed with descriptions of all the sequences (Sec. IV). Moreover, the design of the small-scale TTC testbed is disclosed in Sec. V. Finally, we establish the TTC benchmark by providing a comprehensive experiment results in Sec. VI, and make the conclusion in Sec. VII.

II. RELATED WORK

Over the past decade, numerous datasets have been developed to address specific tasks in autonomous driving.

Our literature review focuses on two categories: 1) datasets for predicting a traffic collision and 2) outdoor datasets collected using event cameras. Table I provides an overview of these datasets, briefly showcasing the different sensor configurations they use and their different focuses.

The first category focuses on collision risk prediction and TTC estimation. The DR(eye)VE Project [1] and DADA-2000 [2] are notable for their rich data and diverse scenarios. DR(eye)VE provides over 500k frames of normal driving scenes along with driver eye-tracking data, and DADA-2000 includes 2k emergency braking sequences and eye-tracking data from volunteers observing accident videos. These datasets facilitate predicting the risk of a collision by using drivers’ attention data as ground truth. Moreover, [3] provides manually labeled 2D bounding boxes and collision labels, featuring 122 emergency braking scenes and 100 normal driving scenes from YouTube. However, as shown in Tab. I, these datasets lack ground truth of depth and trajectory, limiting their usage in developing TTC estimation algorithms. Among these datasets, the TSTTC [4] dataset is closest to our focus, particularly collected for TTC estimation in driving scenarios. TSTTC is a large-scale dataset with over 200k frames captured from real-world highway and urban environments using monocular cameras with three different focal lengths, covering a detection range of 16-96 meters. It includes TTC ground truth and 2D/3D bounding boxes, which are crucial for TTC estimation. However, TSTTC mainly focuses on normal driving scenarios with low relative speed between vehicles, and lacks emergency braking under high-speed scenarios, which are essential for evaluating TTC algorithms in extreme conditions.

The second category is those collected using event cameras, which leverages the specialty of event cameras to handle high-speed motion and challenging lighting conditions. Among them, MVSEC [5] is the first stereo event camera dataset, where the stereo rig is handheld or mounted on a mobile platform, such as a drone, an automobile and a motorcycle. Afterwards, Gehrig *et al.* [6] present a large-scale stereo event dataset with higher spatial resolution, better sensor synchronization, and a wide baseline, leading to better depth estimation accuracy. To handle extreme lighting conditions, ViViD++ [7] integrates thermal sensors alongside

Devices	Models	Parameters
8-mm Lens Camera Pair	Prophesee EVKv4	1280×720 1/2.5" FoV: 41°×24°
	FLIR Blackfly S	1920×1200 1/2.3" FoV: 44°×29° Rate: 20Hz
16-mm Lens Camera pair	Prophesee EVKv4	1280×720 1/2.5" FoV: 22°×12°
	FLIR Blackfly S	1920×1200 1/2.3" FoV: 23°×14° Rate: 20Hz
LiDAR	Livox HAP (TX)	150m @ 10% NIST ±3cm range accuracy FoV: 120°×25° IMU: BMI088
GNSS/INS	CGI-610	Dual-Antenna RTK Rate: 100Hz ±3cm pos. accuracy
	UG005 X1	±2cm/s vel. accuracy

TABLE II: Sensors and characteristics.

event cameras, but hardware triggered synchronization is not witnessed. Limited by the immaturity of event cameras at that time, high definition (HD) data are not available in these datasets until the appearance of M3ED [8], which utilizes a Prophesee EVKv4 event camera with a spatial resolution of 1280×720 pixels. In general, these datasets lack specific scenarios and data for the development of TTC tasks.

Different from above-mentioned datasets, the proposed one (called EvTTC) focuses on providing pre-collision data across various speeds and scenarios. The data are collected using a hardware-synchronized sensor suite, which includes two pairs of high-resolution event and RGB cameras with different focal length for covering a larger and wider field of view. Besides, the sensor suite also consists of a LiDAR, an IMU, and a GNSS/INS for providing high-frequency ground-truth information. The pre-collision data involve vehicles and pedestrians, offering comprehensive ground truth of depth, vehicle's pose, TTC, and 2D bounding boxes of front obstacles. As a result, it will fill the gap in high-quality data for TTC tasks under challenging driving conditions.

III. HARDWARE CONFIGURATION

The setup of the hardware is discussed in this section. Specifically, we disclose the sensors and devices used for recording raw data and ground truth (Sec. III-A), followed by the details of time synchronization across different devices (Sec. III-B) and multi-sensor calibration (Sec. III-C).

A. Sensors

The raw data used for estimating TTC contain RGB images and corresponding event streams. To cover a relatively

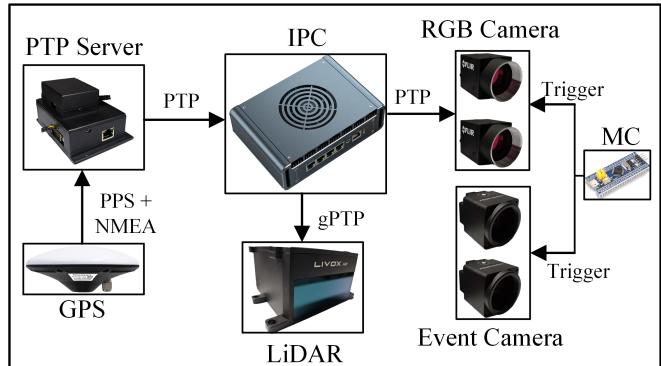


Fig. 3: First, the GPS sends PPS and NMEA messages to the PTP server for time synchronization. The PTP server then synchronizes the time with the industrial personal computer (IPC) using the PTP protocol. The IPC synchronizes time with the RGB camera using the PTP protocol and uses the gPTP protocol to synchronize time with the LiDAR. Finally, the micro-controller(MC) generates four pulses to synchronize the RGB and event cameras.

complete sensing range, we employ two groups of RGB-Event camera pair equipped with lens of different focal length. The camera pair using 8-mm lens consists of an RGB camera and an event camera, and it covers the close sensing range (see (A) in Fig. 1). As a complementary, the camera pair using 16-mm lens is used to cover the distant sensing range (see (B) in Fig. 1). In each pair, the event camera and the RGB camera are rigidly attached with a narrow baseline of 4 cm, and thus, the extrinsic parameters (w.r.t the coordinate system of the sensor suite) can be approximately shared, simplifying data fusion across the two sensor modalities. The employed event camera is Prophesee EVK-v4 (sensor size 1/2.5") that has a spatial resolution of 1280 × 720 pixels. To ensure a similar field of view (FOV), an FLIR Blackfly-S RGB camera (sensor size 1/2.3") with a 1920×1200 resolution is used and equipped with an identical lens. Detailed parameters of each device can be found in Tab. II.

To provide ground truth for the dataset, an automotive-grade solid-state LiDAR and two high-performance GNSS/INS units were prepared, with relevant details provided in Table. II. The LiDAR was rigidly mounted together with the two camera pairs and used to provide ground truth depth. UG005 X1 was rigidly mounted with the cameras to collect high-frequency vehicle pose and velocity data. In scenarios involving a moving lead vehicle, CGI-610 was installed on the lead vehicle to obtain high-frequency dynamic information. Professional vehicle safety assessment organizations such as Euro NCAP require dynamic data, including the relative position and speed between the test vehicle and the front obstacle, to be collected at a frequency of at least 100 Hz. To meet these requirements, both GNSS/INS units were configured to provide vehicle pose and velocity data at 100 Hz.

Car-to-Car Rear scenarios			Car-to-Pedestrian scenarios		
CCRs	CCRs-side	CCRm	CPLA	CPNA	CPNAO

Fig. 4: The top-view schematic of the dataset scenarios. The lateral shadowing of the DCV in the CCRs and CCRm scenarios indicates that data were collected across different lane positions.

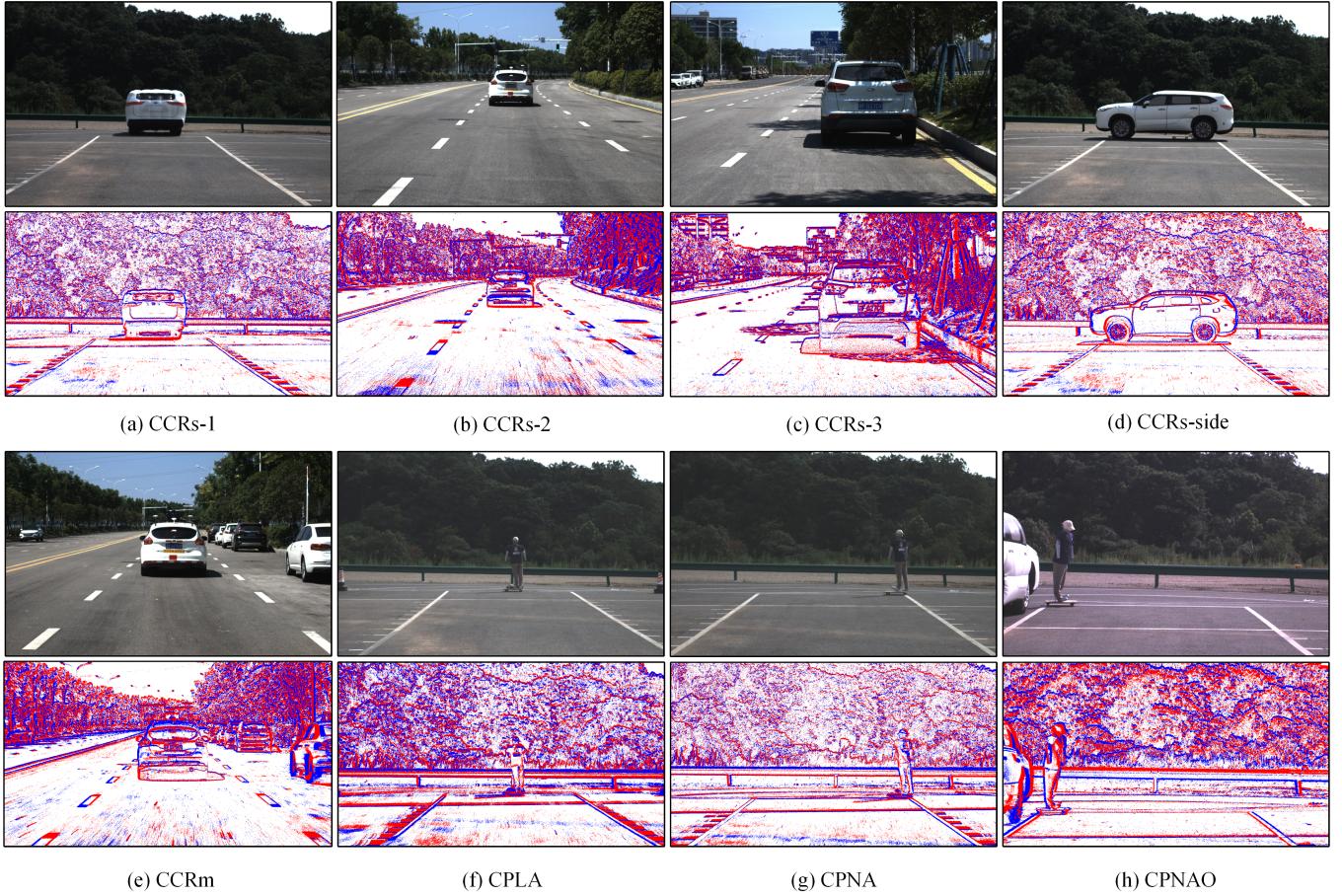


Fig. 5: An overview of our dataset scenarios. CCRs-1, CCRs-2, and CCRs-3 represent three different types of GVT stationary scenarios on the road. CCRs-side indicates a scenario with the GVT positioned stationary in a lateral orientation. CCRm represents a low-speed GVT moving down the road as the DCV accelerates towards it. CPLA depicts a stationary APT positioned on the road, while CPNA represents an APT crossing the road. CPNAO indicates a scenario where the APT suddenly crosses the road from behind an obstacle.

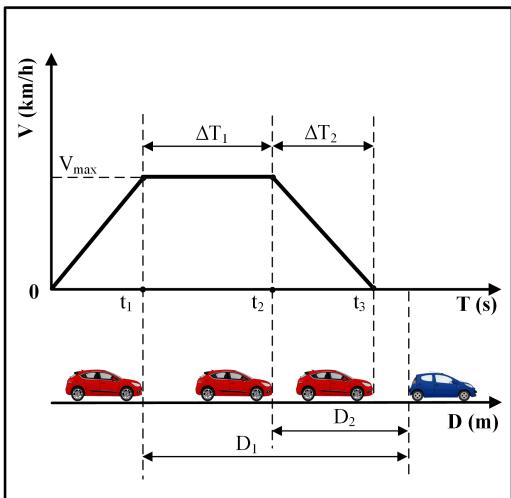


Fig. 6: Definitions of parameters used to describe TTC scenarios in the sequence. ΔT_1 : total time at constant speed, ΔT_2 : total braking time, D_1 : distance to the GVT when the DCV reaches maximum speed, D_2 : distance to the GVT at the onset of braking.

Seq. Name	V_{rel}	Tier	$\ V_{rel}\ _{max}(km/h)$	$D_1(m)$	$D_2(m)$	$\Delta T_1(s)$	$\Delta T_2(s)$	Total
CCRs-1		low	22.7	50	7	7.4	1.3	3
		medium	33.5	32	9	2.5	1.5	3
		high	70.4	109	31	4.2	3.5	3
CCRs-2		low	14.2	35	12	4.0	3.0	1
		medium	40.3	58	24	3.0	4.0	1
		high	57.6	115	38	4.8	5.5	1
CCRs-3		low	24.1	31	14	2.6	3.5	1
		medium	41.4	42	25	1.5	3.8	1
CCRs-side		low	21.2	46	6	7.0	1.1	1
		medium	35.3	42	12	3.0	2.0	1
		high	61.9	76	40	2.1	4.5	1
CCRm		low	25.2	57	20	5.5	3.5	3
		medium	37.4	34	19	1.5	3.0	3
CPLA		low	24.5	88	6	13.1	1.3	1
		medium	41.4	89	10	7.1	1.7	1
		high	61.6	106	26	4.9	3.5	1
CPNA		low	23.8	92	11	13.0	1.9	1
		medium	38.9	108	12	9.1	1.6	1
		high	61.7	108	23	5.2	2.5	1
CPNAO		low	23.7	94	8	14.2	1.9	1
		medium	39.7	104	18	8.2	2.5	1
		high	55.0	87	42	2.9	4.7	1

TABLE III: The relevant parameters for each TTC scenario.

B. Time Synchronization

We synchronize all sensors in the sensor suite at the hardware level. To this end, we utilize Precision Timing Protocol (PTP) [26] and the generalized Precision Time Protocol (gPTP) [27], which can provide sub-microsecond synchronization accuracy in the Ethernet.

The synchronization process of the sensor devices is shown in Fig. 3. First, we use a PTP server to synchronize GPS’s clock with the system clock of the industrial personal computer (IPC). The IPC is then designated as the master clock, using the PTP protocol to synchronize with the RGB cameras. Since the Livox HAP only supports the gPTP protocol, we use gPTP to synchronize between the IPC and the LiDAR. To synchronize the event cameras with the system time, we utilize a micro-controller to generate four 20 Hz synchronization pulses. These pulses are used to simultaneously trigger the two pairs of RGB cameras and event cameras. The RGB and event cameras were configured in external trigger mode. When the RGB camera received a synchronization pulse, it began capturing an image and recorded its timestamp. Simultaneously, the event camera also received the synchronization pulse, immediately generating a trigger data and recording its timestamp. By calculating the time offset between the image and trigger data timestamps, hardware-level time synchronization was achieved. The timestamps of the data from the integrated navigation systems UG005 X1 and CGI-610 are synchronized using their built-in GPS clocks.

C. Calibration

1) *Intrinsic and Extrinsic Calibration of Cameras*: We used the Kalibr toolbox [28] to calibrate the two camera pairs individually. The clocks of the cameras were synchronized using external trigger signals. We employed the simple image recon library to reconstruct the event stream at the midpoint of the RGB image exposure. This library utilizes the methods described in Frequency Cam [29].

2) *Extrinsic Calibration of Camera and IMU*: We also used the Kalibr toolbox to calibrate the extrinsic parameters of the camera and IMU. In front of the AprilTag grid, we excited the sensor device along all IMU axes (both rotational and translational) to complete the external calibration.

3) *Extrinsic Calibration of Cameras and LiDAR*: To calibrate the extrinsic parameters between the RGB camera and the LiDAR, we first stabilize the sensor setup and accumulate several frames of LiDAR point clouds along with the corresponding RGB images captured at the same time. We then use the CAD models of both sensors to provide an initial transformation and carefully fine-tune the alignment using the Manual Calibration Tool in the Sensors Calibration toolbox [30], maximizing the overlap between the projected point clouds and the images.

IV. SEQUENCES OVERVIEW

Our dataset primarily focuses on Car-to-Car emergency collision scenarios. Therefore, we have collected multiple sequences involving various vehicle targets and a range of Car-to-Car motion states. Since pedestrians are also a

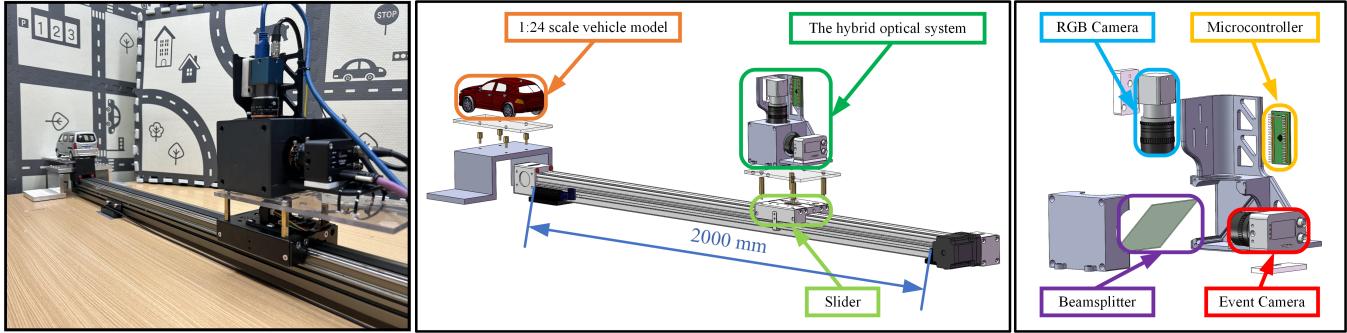


Fig. 7: From left to right are the actual setup of our small-scale TTC testbed, followed by the assembly diagram, and the assembly layout of the hybrid optical system.

significant subject of study in ADAS, we have additionally collected emergency collision sequences involving Car-to-Pedestrian scenarios.

As summarized in Fig. 4, our dataset includes two types of scenarios: Car-to-Car Rear scenarios and Car-to-Pedestrian scenarios. Fig. 6 illustrates the complete motion process of the Data Collection Vehicle (DCV). Table. III provides a detailed overview of parameters such as maximum relative speed, braking distance, and braking time for different scenarios.

A. Car-to-Car Rear scenarios

In the Car-to-Car Rear stationary (CCRs) scenarios outlined in Fig. 4, we selected two real vehicles and one inflatable target vehicle as collision targets. To obtain more realistic emergency collision data, we included an inflatable target vehicle with the same size and similar reflective characteristics as a real vehicle (detectable by LiDAR). As illustrated in Fig. 4, the relative positions and motion states of the vehicles in the following three scenarios are depicted. When the Global Vehicle Target (GVT) is a real vehicle, the DCV approaches the GVT at approximately 20, 40, and 60 km/h. When the GVT is an inflatable target vehicle, it is positioned both longitudinally and laterally. And the DCV approaches the GVT at approximately 20, 40, and 60 km/h.

In the Car-to-Car Rear moving (CCRm) scenarios described in Fig. 4, we selected a real vehicle as the collision target. The GVT travels at a constant speed of 20 km/h, while the DCV approaches the GVT at speeds of approximately 40 and 60 km/h. As shown in Table. III, data were collected for the CCRs-1 and CCRm scenarios with different lane positions, including the DCV directly aligned with the GVT, partially offset, and fully offset.

B. Car-to-Pedestrian scenarios

In the ar-to-Pedestrian (CP) scenario outlined in Fig. 4, we selected an inflatable adult as the collision target. There are three types of scenarios in the CP category: Car-to-Pedestrian Longitudinal Adult (CPLA), Car-to-Pedestrian Nearside Adult (CPNA), and Car-to-Pedestrian Nearside Adult Obstructed (CPNAO). In the CPLA scenario, the DCV approaches the APT at speeds of 20, 40, and 60 km/h,

with the APT positioned in the middle of the road, facing away from the vehicle. In the CPNA scenario, the DCV also approaches the APT at speeds of 20, 40, and 60 km/h, with the APT crossing the road at a constant speed. In the CPNAO scenario, the DCV approaches the APT at 20, 40, and 60 km/h, but with a parked vehicle obstructing the DCV driver's view of the APT as it crosses the road at a constant speed.

C. Ground Truth TTC

We define the ground-truth TTC as the ratio of the relative distance to the relative velocity along the z axis in the coordinate system of the telephoto camera pair. All subsequent data are transformed into the camera coordinate system using the extrinsic parameters between the sensors.

For real-world road scenario data, the relative distance and velocity between the vehicles were obtained using GNSS/INS and LiDAR sensors mounted on the vehicles. First, the position and velocity of the DCV in the camera coordinate system were obtained through the GNSS/INS on the DCV. Then, during GVT motion scenarios, the position and velocity of the GVT in the camera coordinate system were obtained via its GNSS/INS. In static GVT scenarios, the distance between the DCV and GVT, after the DCV comes to a stop, was measured using LiDAR to determine the GVT's position in the camera coordinate system. We used the positions and velocities of the DCV and GVT in the camera coordinate system to calculate their relative distance and velocity, deriving the ground truth TTC at 100 Hz.

We also provide ground truth pose and depth for EvTTC. First, poses and velocity-compensated LiDAR scans were obtained using the LiDAR odometry FAST-LIO [31]. Subsequently, multiple frames of point clouds were accumulated, and occlusions were removed from the point clouds, as described in M3ED [8].

V. SMALL-SCALE TESTBED

Evaluating algorithm performance in real-world online scenarios is highly valuable. However, the cost of using a full-size platform is high, and the differences between simulation and real-world data are considerable. To address

Devices	Models	Parameters
Hybird optical system	Inivation DVXplorer	FoV: 640×480 1/3.5" FoV: 20°×15° 1440×1080
	DAHENG MER2	FoV: 17°×13° 1/2.9" Rate: 25Hz

TABLE IV: Sensor Parameters of the hybrid optical system.

these limitations, a small-scale TTC testbed was designed to simulate emergency vehicle collision scenarios. The platform primarily consists of a 1:24 scale model vehicle, a linear rail, and an open-source hybrid optical system [36]. Figure 7 illustrates the installation details, and Table IV provides the hardware specifications of the platform.

The hybrid optical system comprises an inivation DVXplorer event camera, an RGB camera, and a beam splitter. The beam splitter divides the incoming light into two paths, ensuring that both cameras share a unified field of view, facilitating pixel-level correspondence between the event and RGB cameras. To achieve precise time synchronization, a microcontroller is used to generate two synchronized 25 Hz pulse signals, which are employed to simultaneously trigger both cameras. The linear rail has an effective travel distance of 2 meters, driven by a servomotor, driver, and an STM32 development board.

A 1:24 scale model vehicle was selected as the collision target and positioned directly in front of the camera. The hybrid optical system was fixed on the slider and propelled towards the target at speeds of 500, 750, and 1000 mm/s. For the small-scale TTC testbed, the ground truth TTC is calculated based on the velocity measured by the motor encoder and the displacement obtained through integration.

VI. EXPERIMENTS

To validate and quantitatively evaluate the performance of different algorithms, we use the relative TTC error, defined as follows:

$$e_{\text{TTC}} = \left| \frac{t_{gt} - t_{est}}{t_{gt}} \right| \times 100\% \quad (1)$$

Here, t_{gt} represents the ground truth TTC value, and t_{est} represents the estimated TTC value.

TTC Benchmark: We employed various advanced TTC estimation methods to evaluate the validity of the collected data. These TTC estimation methods include STRTTC [24], CMax [32], ETTCM [33], FAITH [34], AEB-Tracker [35] and Image's FOE [20]. Multiple sequences from the dataset were selected for testing. The accuracy and computational efficiency of these algorithms on our dataset are presented in Table V.

The STRTTC method demonstrated robust performance under various conditions due to its resilient sampling strategy and efficient initialization scheme. The CMax method achieves higher accuracy because it utilizes all events within the bounding box for computation, resulting in a better signal-to-noise ratio. However, its nonlinear least-squares

optimization increases computational complexity (e.g., processing 3e5 events typically takes around 3 seconds). In our experiments, the ETTCM method was configured with a scaling model and a neighboring size of 3 to compute the TTC. The FAITH method, which replaces optical flow with normal flow for TTC estimation, resulted in reduced accuracy. The AEB-Tracker method achieved exceptionally high computational efficiency, up to 10 kHz. However, in the presence of significant background noise, it relies on a low-frequency feature detector (e.g., 20 Hz) to filter out the noise. Finally, the Image's FOE method achieved high accuracy and efficiency across all sequences. However, its performance is constrained by the limitations of standard cameras, such as data latency, and motion blur at high speeds.

VII. CONCLUSION

We present an event camera dataset, called EvTTC, for the task of time-to-collision estimation in autonomous driving. It consists of data captured using frame-based and event-based cameras, covering various scenarios of potential collision in daily driving. High-precision external reference signals are also provided for ground truth, and a benchmark of event-based TTC is established. Besides, a small-scale TTC testbed for experimental validation and data augmentation is presented. We wish the release of EvTTC can facilitate the development of forward collision warning techniques using event cameras.

REFERENCES

- [1] A. Palazzi, D. Abati, F. Solera, R. Cucchiara, *et al.*, “Predicting the driver’s focus of attention: the dr (eye) ve project,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 7, pp. 1720–1733, 2018.
- [2] J. Fang, D. Yan, J. Qiao, J. Xue, H. Wang, and S. Li, “Dada-2000: Can driving accident be predicted by driver attentionf analyzed by a benchmark,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 4303–4309.
- [3] H. Kim, K. Lee, G. Hwang, and C. Suh, “Crash to not crash: Learn to identify dangerous vehicles using a simulator,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 978–985.
- [4] Y. Shi, Z. Huang, Y. Yan, N. Wang, and X. Guo, “Tsttc: A large-scale dataset for time-to-contact estimation in driving scenarios,” *arXiv preprint arXiv:2309.01539*, 2023.
- [5] A. Z. Zhu, D. Thakur, T. Özaslan, B. Pfommer, V. Kumar, and K. Daniilidis, “The multivehicle stereo event camera dataset: An event camera dataset for 3d perception,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2032–2039, 2018.
- [6] M. Gehrig, W. Aarents, D. Gehrig, and D. Scaramuzza, “Dsec: A stereo event camera dataset for driving scenarios,” *IEEE Robotics and Automation Letters*, 2021.
- [7] A. J. Lee, Y. Cho, Y.-s. Shin, A. Kim, and H. Myung, “Vivid++: Vision for visibility dataset,” *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6282–6289, 2022.
- [8] K. Chaney, F. Cladera, Z. Wang, A. Bisulco, M. A. Hsieh, C. Korpela, V. Kumar, C. J. Taylor, and K. Daniilidis, “M3ed: Multi-robot, multi-sensor, multi-environment event dataset,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4015–4022.
- [9] J. B. Cicchino, “Effectiveness of forward collision warning and autonomous emergency braking systems in reducing front-to-rear crash rates,” *Accident Analysis & Prevention*, vol. 99, pp. 142–152, 2017.
- [10] B. Fildes, M. Keall, N. Bos, A. Lie, Y. Page, C. Pastor, L. Pennisi, M. Rizzi, P. Thomas, and C. Tingvall, “Effectiveness of low speed autonomous emergency braking in real-world rear-end crashes,” *Accident Analysis & Prevention*, vol. 81, pp. 24–29, 2015.

Method	CCRs1-low		CCRs1-medium		CCRs1-high		CCRs2-low		CCRs2-medium	
	e_{TTC} (%)	Runtime (s)								
STRITTC [24]	7.54	<u>0.031</u>	9.64	<u>0.026</u>	7.78	0.045	8.15	<u>0.024</u>	11.83	<u>0.024</u>
CMax [32]	2.56	<u>3.833</u>	3.44	<u>3.701</u>	<u>7.39</u>	3.228	5.04	<u>3.246</u>	<u>11.20</u>	<u>3.449</u>
ETTCM [33]	45.11	0.081	44.94	0.081	<u>43.03</u>	0.300	48.83	0.350	<u>52.84</u>	0.098
FAITH [34]	15.15	0.156	20.63	0.155	19.51	0.337	25.83	0.169	47.45	0.191
AEB-Tracker [35]	39.57	4.2×10^{-5}	42.41	3.9×10^{-5}	39.04	1.1×10^{-5}	37.09	9.4×10^{-6}	43.17	7.3×10^{-6}
Image's FoE [20]	5.37	0.036	<u>5.12</u>	0.031	1.86	<u>0.041</u>	3.76	0.031	3.85	0.031

Method	CCRs2-high		CCRM-low		CCRM-medium		Slider-750		Slider-1000	
	e_{TTC} (%)	Runtime(s)	e_{TTC} (%)	Runtime(s)	e_{TTC} (%)	Runtime(s)	e_{TTC} (%)	Runtime(s)	e_{TTC} (%)	Runtime(s)
STRITTC [24]	13.68	<u>0.028</u>	9.87	0.031	10.36	<u>0.023</u>	8.95	0.015	12.43	0.016
CMax [32]	<u>11.10</u>	<u>3.826</u>	<u>9.27</u>	3.658	<u>14.73</u>	<u>3.657</u>	4.16	0.85	2.74	0.93
ETTCM [33]	49.46	0.127	<u>71.05</u>	0.059	58.00	0.121	18.99	0.498	15.20	0.191
FAITH [34]	49.62	0.165	75.78	0.116	56.26	0.11	34.45	0.186	47.94	0.233
AEB-Tracker [35]	42.43	6×10^{-6}	26.95	1×10^{-5}	35.99	1.1×10^{-5}	157.88	1×10^{-5}	165.56	7.8×10^{-6}
Image's FoE [20]	2.92	0.031	5.60	<u>0.027</u>	3.86	0.026	11.35	<u>0.013</u>	<u>7.55</u>	<u>0.012</u>

TABLE V: Quantitative evaluation on our data. Best and second best results are **highlighted** and underlined.

- [11] I. Isaksson-Hellman and M. Lindman, “Evaluation of rear-end collision avoidance technologies based on real world crash data,” *Proceedings of the Future Active Safety Technology Towards zero traffic accidents (FASTzero), Gothenburg, Sweden*, pp. 9–11, 2015.
- [12] M. Rizzi, A. Kullgren, and C. Tingvall, “Injury crash reduction of low-speed autonomous emergency braking (aeb) on passenger cars,” in *International Research Council on the Biomechanics of Injury (IRCOBI 2014)*. International Research Council on the Biomechanics of Injury, 2014, pp. 656–665.
- [13] M. Gawande, P. Rajalakshmi, *et al.*, “Autonomous emergency breaking (aeb) evaluation for indian traffic scenarios using gps and lidar data,” in *2022 IEEE IAS Global Conference on Emerging Technologies (GlobConET)*. IEEE, 2022, pp. 655–660.
- [14] M. Bosnak and I. Skrjanc, “Efficient time-to-collision estimation for a braking supervision system with lidar,” in *2017 3rd IEEE International Conference on Cybernetics (CYBCONF)*. IEEE, 2017, pp. 1–6.
- [15] M. Kotur, N. Lukić, M. Krunic, and Ž. Lukač, “Camera and lidar sensor fusion for 3d object tracking in a collision avoidance system,” in *2021 Zooming Innovation in Consumer Technologies Conference (ZINC)*. IEEE, 2021, pp. 198–202.
- [16] K. Venkatesha, A. Vengadarajan, and U. Singh, “Detection mechanism for vehicle collision avoidance using mmwave radar,” in *2023 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*. IEEE, 2023, pp. 1–5.
- [17] E. Dagan, O. Mano, G. P. Stein, and A. Shashua, “Forward collision warning with a single camera,” in *IEEE Intelligent Vehicles Symposium, 2004*. IEEE, 2004, pp. 37–42.
- [18] F. Meyer and P. Bouthemy, “Estimation of time-to-collision maps from first order motion models and normal flows,” in *1992 11th IAPR International Conference on Pattern Recognition*, vol. 1. IEEE Computer Society, 1992, pp. 78–82.
- [19] A. Negre, C. Braillon, J. L. Crowley, and C. Laugier, “Real-time time-to-collision from variation of intrinsic scale,” in *Experimental Robotics: The 10th International Symposium on Experimental Robotics*. Springer, 2008, pp. 75–84.
- [20] S. Stabinger, A. Rodriguez-Sanchez, and J. Piater, “Monocular obstacle avoidance for blind people using probabilistic focus of expansion estimation,” in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2016, pp. 1–9.
- [21] D. Falanga, K. Kleber, and D. Scaramuzza, “Dynamic obstacle avoidance for quadrotors with event cameras,” *Science Robotics*, vol. 5, no. 40, p. eaaz9712, 2020.
- [22] H. Rebecq, T. Horstschafer, and D. Scaramuzza, “Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization,” 2017.
- [23] X. Lu, Y. Zhou, and S. Shen, “Event-based visual inertial velocimeter,” *arXiv preprint arXiv:2311.18189*, 2023.
- [24] J. Li, B. Liao, X. LU, P. Liu, S. Shen, and Y. Zhou, “Event-aided time-to-collision estimation for autonomous driving,” 2024.
- [25] D. Gehrig and D. Scaramuzza, “Low-latency automotive vision with event cameras,” *Nature*, vol. 629, no. 8014, pp. 1034–1040, 2024.
- [26] IEEE Std 1588-2019 (Revision of IEEE Std 1588-2008), “IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems,” pp. 1–499, 2020.
- [27] IEEE Std 802.1AS-2020 (Revision of IEEE Std 802.1AS-2011), “IEEE Standard for Local and Metropolitan Area Networks—Timing and Synchronization for Time-Sensitive Applications,” pp. 1–421, 2020.
- [28] P. Furgale, J. Rehder, and R. Siegwart, “Unified temporal and spatial calibration for multi-sensor systems,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1280–1286.
- [29] B. Pfommer, “Frequency cam: Imaging periodic signals in real-time,” *arXiv preprint arXiv:2211.00198*, 2022.
- [30] G. Yan, Z. Liu, C. Wang, C. Shi, P. Wei, X. Cai, T. Ma, Z. Liu, Z. Zhong, Y. Liu, M. Zhao, Z. Ma, and Y. Li, “Opencalib: A multi-sensor calibration toolbox for autonomous driving,” *arXiv preprint arXiv:2205.14087*, 2022.
- [31] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, “Fast-lio2: Fast direct lidar-inertial odometry,” *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.
- [32] G. Gallego, H. Rebecq, and D. Scaramuzza, “A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3867–3876.
- [33] U. M. Nunes, L. U. Perrinet, and S.-H. Ieng, “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*, 2023, pp. 23 653–23 663.
- [34] R. Dinaux, N. Wessendorp, J. Dupeyroux, and G. C. De Croon, “Faith: Fast iterative half-plane focus of expansion estimation using optic flow,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7627–7634, 2021.
- [35] Z. Wang, T. Molloy, P. van Goor, and R. Mahony, “Asynchronous blob tracker for event cameras,” *IEEE Transactions on Robotics*, 2024.
- [36] J. Hidalgo-Carrió, G. Gallego, and D. Scaramuzza, “Event-aided direct sparse odometry,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5781–5790.