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

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] Dept. of ECE, Hong Kong University of Science and Technology

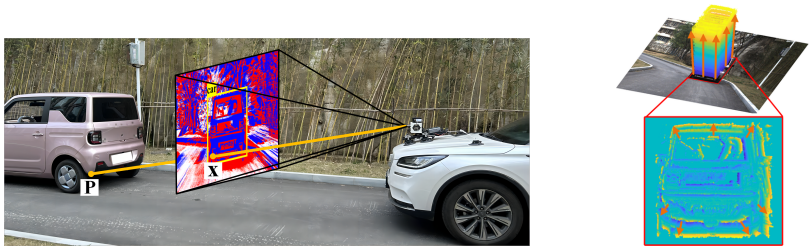
Abstract. Predicting a potential collision with leading vehicles is an essential functionality of any autonomous/assisted driving system. One bottleneck of existing vision-based solutions is that their updating rate is limited to the frame rate of standard cameras used. In this paper, we present a novel method that estimates the time to collision using a neuromorphic event-based camera, a biologically inspired visual sensor that can sense at exactly the same rate as scene dynamics. The core of the proposed algorithm consists of a two-step approach for efficient and accurate geometric model fitting on event data in a coarse-to-fine manner. The first step is a robust linear solver based on a novel geometric measurement that overcomes the partial observability of event-based normal flow. The second step further refines the resulting model via a spatio-temporal registration process formulated as a nonlinear optimization problem. Experiments on both synthetic and real data demonstrate the effectiveness of the proposed method, outperforming other alternative methods in terms of efficiency and accuracy. Dataset used in this paper can be found at `https://nail-hnu.github.io/EventAidedTTC/`.

Keywords: Event cameras · Time to collision · Autonomous driving

1 Introduction

The ability to identify objects that pose a conflicting threat to the host vehicle and make a brake decision is fundamentally important in Level-2 and beyond autonomous/assisted driving techniques. The time for such a potential collision between the host vehicle and the leading vehicle is coined by *time to collision* (TTC), which has been used for alerting drivers and autonomous driving systems for deceleration. The most straightforward way to calculate TTC requires knowing the absolute distance and relative speed between the two vehicles. Thus, LIDAR/Radar based solutions are typically applied [33,38]. However, it has been witnessed that experienced human drivers perform well on controlling the relative speed adaptively, indicating that visual information alone is enough for the task of TTC estimation.

⋆ equal contribution; ✉ corresponding author (eeyzhou@hnu.edu.cn).



(a) Geometry of the event-based TTC problem.

(b) Motion of contours and the corresponding events triggered.

Fig. 1: Illustration of the TTC estimation problem. (a) The leading vehicle is perceived by an following observer. (b) Events are triggered as the image size of the leading vehicle varies in the presence of relative speed. A motion flow field of outward expansion is visible in the close-up of the proposed image representation of events (see 3.3 for detail), indicating a decrease in the relative distance and an increase of collision risk.

There is a large body of literature on TTC estimation using standard cameras [5, 22–25, 28, 35, 36]. The most widely studied case uses a monocular standard camera and calculates the TTC using a pair of successive images. The leading car's scale (i.e., image size) varies in the presence of relative speed w.r.t the host vehicle, and the TTC can be recovered from such a scale variation. One bottleneck of these solutions is that their updating rate is limited to the frame rate of standard cameras. Standard cameras used in autonomous/assisted driving systems typically run around 10 Hz due to the consideration of cost, bandwidth, and energy consumption. The interval between two successive exposures (i.e., 100 ms), even not taking into account the computation time of the applied TTC algorithm, can be a notably big latency to a collision warning system, especially when the relative speed increases drastically.

Event-based cameras are biologically-inspired novel sensors that mimic the working principle of vision pathway of human beings. Different from standard cameras, an event-based camera reports only brightness changes at each pixel location asynchronously. This unique characteristic leads to much better performance in terms of temporal resolution and dynamic range. Therefore, event cameras are inherently well-qualified to deal with robotics applications involving aggressive motion [8, 29] and high dynamic range (HDR) scenarios [27, 30].

In this paper, we look into the problem of TTC estimation using an event-based camera. We focus on predicting a potential collision from event data given the presence of a leading vehicle as a prior. Our goal is to obtain accurate TTC estimates at an ultra framerate by leveraging the asynchronous characteristic and high temporal resolution of event cameras. As illustrated in Fig. 1, the image size of the leading vehicle varies when there is a relative speed between the leading vehicle and the observer. Such a scale variation induces an optical flow field of radial expansion (or contraction), and meanwhile a set of events are triggered as those contours expand (or contract). Ideally, the TTC can be derived from the divergence of this flow field [25]. However, recovering the full

flow field from event data is not trivial and even intractable because event data report only photometric variations along local gradient directions. To this end, we propose a geometric method that can recover the TTC from the profile of event-based normal flow. Specifically, the contribution of this paper consists of:

- A time-variant affine model, derived from the dynamics of contour points, that accurately guides the warping of events in the spatio-temporal domain.
- A robust linear solver based on the proposed geometric measurement that can overcome the partial observability of event-based normal flow.
- A nonlinear solver that further refines the resulting model via a spatio-temporal registration process, in which efficient and accurate data association is established by using an ingeniously modified representation of events, i.e., linear time surfaces.

Outline: The rest of the paper is organized as follows. Section 2 gives a literature review on the development of TTC techniques. Section 3 discloses the proposed event-based method for TTC estimation. Our method is evaluated in Sec. 4 and conclusions are drawn in Sec. 5.

2 Related Work

2.1 History of TTC Estimation

The earliest work in regards to TTC estimation can date back to the 1980s. D.N.Lee claimed that TTC was widely used in nature, such as birds' landing [15], and he also argued that TTC could provide enough information for a driver to adjust the velocity w.r.t obstacles [14]. The connection between the TTC and the 2D motion (optical flow) field is first disclosed in [25], which derives the TTC from the motion field's divergence and realizes obstacle avoidance. Successive work [22, 23] prove that the first-order visual models are sufficient to obtain TTC estimates, and [22] further demonstrates that normal flow is enough to recover the divergence as long as the displacement between the successive frames is small. Since then, a large number of similar ideas have been witnessed [28, 35, 36], including some variants that leverage the scale variation of features [5, 24]. More recently, learning approaches have been applied to solve the TTC problem. [20] proposes an end-to-end deep learning approach that directly estimates the time to collision using a sequence of images as input. [2] formulates the TTC problem (on multiple obstacles) as a pixel-level multiple binary classification task for different TTC thresholds. Although modern state-of-the-art solutions have pushed a remarkable step forward in terms of computational efficiency, the latency that originates from the sensing end (i.e., frame-based cameras) still exists. The only way to circumvent this bottleneck is to bring in sensors that have a higher temporal resolution to sense at the same rate as the scene dynamics.

2.2 Tryouts with Event-based Cameras

The research on the TTC problem, or more generally speaking, obstacle avoidance using event-based vision has received large attention from the community

of robotics [8, 31, 32, 37] and neuromorphic engineering [4]. Inspired by methods using standard cameras, most event-based solutions are built on top of optical flow from event data [1, 3, 19, 43, 44]. Among the flow-based methods, we find a major category [4, 6], which typically starts with localizing the Focus of Expansion (FOE) by increasingly shrinking the intersection area of multiple negative half-planes. These methods facilitate UAV heading and TTC estimation in the presence of marking lines. However, the generalizability of TTC estimation is hindered by the specific conditions it relies on (i.e., requiring marking lines), and its accuracy is notably influenced by the imprecisely estimated FOE. Besides, it's also worth mentioning that a parametric model of events' motion can be regressed from event data without explicitly solving the optical flow problem. A unifying contrast maximization framework presented in [10], also known as Contrast Maximization (CM), can fit a family of geometric models on event data without explicit data association, such as features and optical flow. Given the resulting geometric model (e.g., an affine model), the TTC can be easily recovered using the derivation in [22, 23, 25]. Such pipelines have been witnessed in [21,34]. Our method is similar to [21] in the sense that we also leverage the fact that TTC is the inverse of the radial flow field's divergence, but with differences: (*i*) Our geometric model is more general and can handle 2D planar translation, while [21] makes a simplification for competitive running time; (*ii*) Our model fitting method via spatio-temporal registration is more efficient than CM-based pipelines due to the least-squares nature; (*iii*) We provide a robust linear solver for initialization, which locates the convergence basin in a closed-form manner.

3 Methodology

In this section, we first state the specific problem solved in this work (Sec. 3.1). Second, we introduce a time-variant geometric model, derived from the real dynamics of contour points, that accurately guides the warping of events (Sec. 3.2). Third, we detail the method for geometric model fitting on event data via solving a spatio-temporal registration problem, in which a novel representation of events is proposed for efficient and accurate data association (Sec. 3.3). To initialize, we also propose a robust linear solver, which is based on a novel geometric measurement that overcomes the partial observability of event-based normal flow (Sec. 3.4). Finally, the design of a complete forward collision warning (FCW) system built on top of the proposed event-based method is discussed (Sec. 3.5).

3.1 Problem Statement

Given the identification result of a preceding vehicle as a prior, the goal of this work is to estimate the TTC during the blind period of the standard camera (*i.e.* the time interval between two successive exposures) using event data as input. The presence of relative speed between the proceeding vehicle and the observer leads to a size variation of the former on the observer's image plane, and such a zoom-in or zoom-out behavior of the car's contours will trigger a set of events.

The TTC information is encoded in the flow field of these contour points, which can be typically parametrized using a geometric model. Our method estimates the TTC by fitting such a parametric model on event data.

3.2 A Flow-Dynamics Consistent Geometric Model

Consider a proceeding vehicle followed by an observer running in the same lane. If the relative instantaneous angular velocity and linear velocity between them are denoted in the observer's body frame as ${}^{\mathtt{B}}\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^{\mathtt{T}}$ and ${}^{\mathtt{B}}\boldsymbol{\nu} = [\nu_x, \nu_y, \nu_z]^{\mathtt{T}}$, respectively, the scene flow of a 3D contour point ${}^{\mathtt{B}}\mathbf{P} = [X, Y, Z]^{\mathtt{T}}$ will be ${}^{\mathtt{B}}\dot{\mathbf{P}} = -{}^{\mathtt{B}}\boldsymbol{\omega} \times {}^{\mathtt{B}}\mathbf{P} - {}^{\mathtt{B}}\boldsymbol{\nu}$. Let $\mathbf{p} = [x, y]^{\mathtt{T}} = [X/Z, Y/Z]^{\mathtt{T}}$ be the image of $\mathbf{P}$ represented by the normalized coordinates.

Usually, the relative angular velocity can be omitted in our context and it's also reasonable to assume a constant relative linear velocity during a short time interval. Some existing methods [28,35,36] using as input the optical flow (or 2D feature correspondences) between two frames further assume a constant optical flow, and thus, the flow vector from start time t_0 to reference time $t_{\mathtt{ref}}$ becomes

$$\mathbf{p}(t_{\mathtt{ref}}) - \mathbf{p}(t_0) = \frac{1}{Z(t_0)} \begin{bmatrix} -\nu_x + x(t_0)\nu_z \\ -\nu_y + y(t_0)\nu_z \end{bmatrix} (t_{\mathtt{ref}} - t_0). \tag{1}$$

This gives rise to a constant parametric flow model, *e.g.* an affine transformation [36], and the TTC can be simply calculated from the model parameters [12] (p. 383). However, such a strong assumption may violate the time-variant nature of the flow field, especially when the relative distance varies notably within a short period of time. The key to the problem of geometric model fitting on event data is applying an accurate model. Therefore, we propose a time-variant parametric model derived from the real dynamics of the flow field.

Written in a continuous-time form, the optical flow $\mathbf{u}$ can be defined as the ordinary differential equation (ODE) of $\mathbf{p}$,

$$\mathbf{u}(t) \doteq \dot{\mathbf{p}}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} = \frac{1}{Z(t)} \begin{bmatrix} -\nu_x + x(t)\nu_z \\ -\nu_y + y(t)\nu_z \end{bmatrix}, \tag{2}$$

based on which the position of $\mathbf{p}$ at $t_{\mathtt{ref}}$ can be accurately obtained by the following integral: $\mathbf{p}(t_{\mathtt{ref}}) = \int_{t_0}^{t_{\mathtt{ref}}} \mathbf{u}(t)dt + \mathbf{p}(t_0)$. By solving the above ODE and substituting the boundary condition at $t_{\mathtt{ref}}$ into its general solution, we obtain

$$\mathbf{p}(t_{\mathtt{ref}}) = \underbrace{\frac{1}{Z(t_{\mathtt{ref}})} \begin{bmatrix} -\nu_x + x(t_0)\nu_z \\ -\nu_y + y(t_0)\nu_z \end{bmatrix}}_{\mathcal{A}(\mathbf{p}(t_0);\ \boldsymbol{\nu}/Z(t_{\mathtt{ref}}))} (t_{\mathtt{ref}} - t_0) + \mathbf{p}(t_0), \tag{3}$$

where $\mathcal{A}(\mathbf{p}(t_0); \boldsymbol{\nu}/Z(t_{\mathtt{ref}}))$ consists of a time-variant affine model and can be regarded as the average flow during the time interval. Note that the only difference between Eq. 3 and Eq. 1 is replacing $Z(t_0)$ with $Z(t_{ref})$. This small change leads to a more accurate geometric model, which is crucial to the following spatio-temporal registration. A justification of this can be found in the supplementary material.

3.3 Model Fitting via Spatio-Temporal Registration

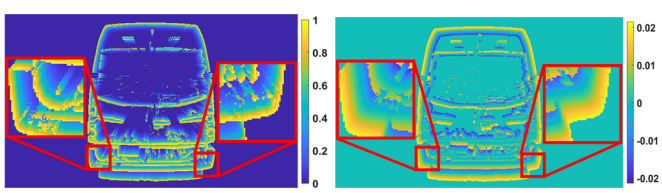
Our goal is to recover the above mentioned geometric model through inferring the event data association, namely determining implicitly which of these events are triggered by the same contour point. Such an operation is typically referred to as spatio-temporal registration [1], which has been widely applied to event-based model fitting tasks, such as 3D rotation estimation [18], 6D pose estimation [41] and motion segmentation [40].

We borrow the general idea of spatio-temporal registration to the problem of fitting an affine model. Instead of using Time Surfaces (TS), an image-like representation of event data in [17, 41], we propose a linear counterpart called Linear Time Surface (LTS). Different from the ordinary TS, an LTS stores the time difference between reference time t_{ref} and the timestamp of the event triggered closest to t_{ref}. As is known, each event $\mathbf{e}_k = (\mathbf{x}_k, t_k, p_k)$ consists of the space-time coordinates at which a certain amount of intensity change occurred and the sign (polarity $p_k \in \{+1, -1\}$) of the change. If the set of events triggered at pixel coordinate $\mathbf{x}$ is denoted by $\mathcal{E}_{\mathbf{x}}$, the LTS rendered at an arbitrarily given time, *e.g.*, $t_{\texttt{ref}}$, is defined by

$$\mathcal{T}(\mathbf{x}, t_{\text{ref}}) = \begin{cases} t_k - t_{\text{ref}}, k = \underset{i \text{ for } \mathbf{e}_i \in \mathcal{E}_{\mathbf{x}}}{\operatorname{argmin}} |t_i - t_{\text{ref}}| & \text{if } \mathcal{E}_{\mathbf{x}} \neq \emptyset \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Note that $\mathbf{x}$ denotes the raw pixel coordinate, and it can be easily transferred to the normalized coordinate system (*i.e.* $\mathbf{p}$ used across the paper) with intrinsic parameters.

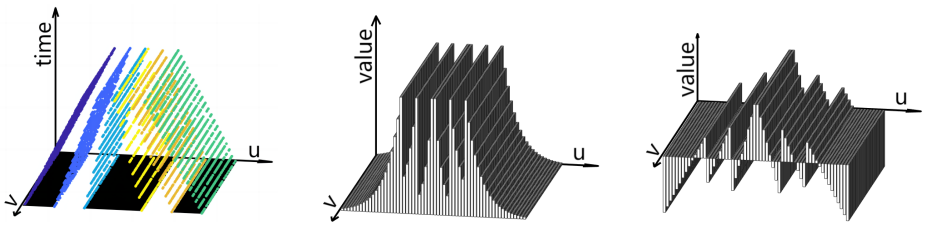
As illustrated in Fig. 2, the proposed LTS is, on one hand, similar to TS in the sense that it also comes with the property of distance field, which enables us to establish event-contour association efficiently. On the other hand, as shown in Fig. 3, the proposed LTS enhances the continuity of the distance field's gradient in a different way. Unlike [41] that circumvents the unilateral truncation of TS's gradient with a smoothing kernel, LTS simply sets reference time $t_{\texttt{ref}}$ as the



(a) Time Surface [40, 41]. **(b)** Linear Time Surface.

Fig. 2: Illustration of two image-like representations of event data. (a) The Time Surface with exponential decay used in [40, 41]; (b) The proposed Linear Time Surface. The applied color system is explained by the adjacent color bar.

[1] Generally speaking, any approach that establishes data association by leveraging information from the spatio-temporal domain falls into this category of methods.



(a) Raw event observations. **(b)** Time Surface. **(c)** Linear Time Surface.

Fig. 3: Comparison on characteristics of TS and LTS. (a) Events are triggered in the spatio-temporal domain as six straight lines traversing at different speeds. (b) The generated TS is an unsigned distance transform, and the gradient at the current position of contours is truncated unilaterally. (c) The generated LTS is a signed distance transform, and the gradient at the current position of contours is continuous by nature.

median timestamp of all involved events. Such a design brings two benefits: 1) The true location of contour points at reference time is not shifted, and thus, no bias is witnessed in the registration result; 2) The resulting distance transform becomes a signed function that leads to more accurate registration results [42].

The spatio-temporal registration problem is ultimately formulated as follows. Let $\mathcal{E}$ be the involved events[2]. Assuming the LTS at $t_{\texttt{ref}}$ is available, denoted by $\mathcal{T}_{\texttt{ref}}$, the goal is to find the optimal affine model such that all involved events would be warped properly to the zero-value locations of $\mathcal{T}_{\texttt{ref}}$. The overall objective function of the spatio-temporal registration is

$$\mathbf{a}^{\star} = \arg\min_{\mathbf{a}} \sum_{\mathbf{e}_k \in \mathcal{E}} \mathcal{T}_{\texttt{ref}}(W(\mathbf{p}_k, t_k, t_{\texttt{ref}}; \mathbf{a}))^2, \tag{5}$$

where $\mathbf{a} \doteq \frac{\boldsymbol{\nu}}{Z(t_{\texttt{ref}})} = [a_x, a_y, a_z]^{\mathsf{T}}$, and the warping function

$$W(\mathbf{p}_k, t_k, t_{\texttt{ref}}; \mathbf{a}) \doteq \mathcal{A}(\mathbf{p}_k; \mathbf{a})(t_{\texttt{ref}} - t_k) \tag{6}$$

transfers events to the common reference time $t_{\texttt{ref}}$. To enhance the smoothness of the object function, we smooth the LTS using a bilateral filter with a Gaussian kernel. Besides, an efficient bilinear-interpolation operation is performed to deal with the sub-pixel coordinates in the warping computation of each residual term. To guarantee the success of Eq. 5, we propose an initialization method that effectively provides an starting point within the convergence basin.

3.4 Initialization

The motion flow equation (Eq. 1) provides a possibility to conduct efficient geometric model fitting using optical flow as input measurements. However, event-based optical flow estimation is an ill-posed problem that is more serve than its standard-vision counterpart. This is due to the partial observability of event

[2]Involved events are those extracted from the spatio-temporal volume of interest.

cameras, where events can only be stimulated when the photometric gradient direction is not orthogonal to the epipolar line. Thus, the recovered motion flow estimate from event data, typically referred to as normal flow, is just the full flow's partial component along the local gradient direction. Consequently, the linear system (Eq. 1) established using raw normal flows as alternative measurements does not guarantee a correct fitting result. Actually, as will be seen in the following analysis, the more the normal flows differ from the full flows, the less accurate the result of model fitting.

To circumvent such a limitation of normal flow, we develop a more effective geometric measurement based on the following fact: The inner product of the full flow vector and the normal flow vector equals to the squared norm of the normal flow vector. Using as an example the event stimulated at pixel location $\mathbf{p}_k$ and by time t_k, the full flow vector could be calculated with Eq. 1, and the above fact is written as

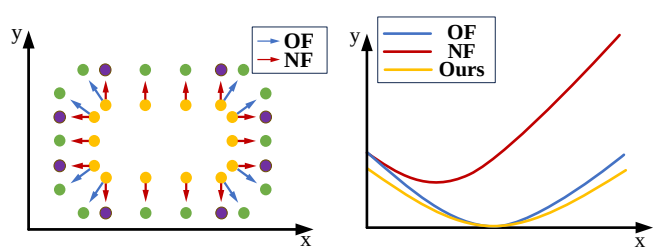
$$\frac{1}{Z_k}\begin{bmatrix}-\nu_x + x_k\nu_z \\ -\nu_y + y_k\nu_z\end{bmatrix}^{\mathtt{T}}\mathbf{n}_k = \mathbf{n}_k^{\mathtt{T}}\mathbf{n}_k, \tag{7}$$

where $\mathbf{n} = [n_x, n_y]^{\mathtt{T}}$ denotes the normal flow. By simply replacing $\frac{1}{Z(t_k)}$ with $\frac{Z(t_{\mathtt{ref}})}{Z(t_{\mathtt{ref}})Z(t_k)}$ in Eq. 7 and with some straightforward derivations, we can build the linear system with state vector $\mathbf{a}$ as,

$$\begin{bmatrix} n_{k,x} \\ n_{k,y} \\ (\delta t_k \mathbf{n}_k - \mathbf{p}_k)^{\mathtt{T}}\mathbf{n}_k \end{bmatrix}^{\mathtt{T}} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = -\mathbf{n}_k^{\mathtt{T}}\mathbf{n}_k, \tag{8}$$

where $\delta t_k \doteq t_{\mathtt{ref}} - t_k$, and $n_{k,\{\cdot\}}$ denotes the $\{\cdot\}$ component of $\mathbf{n}_k$. A minimal solver of Eq. 8 requires three measurements, and we use RANSAC [9] for robust estimation.

We show the benefit of using the proposed geometric measurement with a toy example. As shown in Fig. 4a, we simulate a number of contour points



(a) A toy example of radially expanding contour points.

(b) Energy curves of model fitting using different measurements.

Fig. 4: An analysis on performing spatio-temporal registration using different measurements, including (*i*) optical flow (OF), (*ii*) normal flow (NF), and (*iii*) our proposed geometric measurement, respectively. Note that only one dimension of the estimated parametric model is visualized in (b) for simplicity.

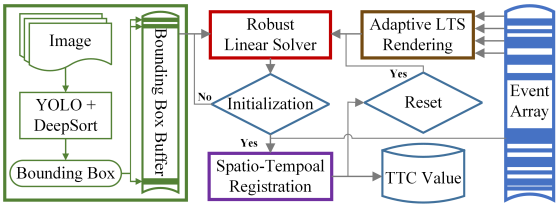


Fig. 5: Flowchart of the proposed FCW system. Key modules of the system consist of the vehicle detection module (green), the LTS rendering module (brown), the robust linear solver (red), and the spatio-temporal registration (purple).

(yellow) undergoing a 2D radial expansion. The true correspondences (green) can be obtained by searching along the full optical flow vectors (blue), while the purple points representing potential correspondences determined by the normal flow vectors (red). Note that some of the normal flows overlap with the full ones, and such a case can only be witnessed when the local gradient direction is in parallel to the full flow vector. We compare the results of model fitting using different measurements, including (*i*) full optical flow, (*ii*) normal flow, and (*iii*) the proposed geometric measurement. As shown in Fig. 4b, (*i*) and (*iii*) return identical model fitting results, while a bias is seen in the result of (*ii*).

3.5 Event-Aided Forward Collision Warning System

We build an event-aided FCW system on top of the proposed algorithms and a vehicle detection method. The established FCW system takes as input raw events and intensity images[3] from either a DAVIS [16] sensor or a hybrid camera system made up of a frame-based camera, an event camera, and a beamsplitter [11]. The FCW system manages to output TTC estimates at an ultra framerate. An overview of the FCW system is given in Fig. 5, in which the key modules are highlighted with rectangles in different colors. Each module takes at least one independent thread and the implementation details are provided in Sec. 4.1.

4 Experiments

The implementation detail of each core module of the FCW system is first discussed (Sec. 4.1). Second, we introduce the datasets used for evaluation (Sec. 4.2). Finally, both qualitative and quantitative evaluations are performed (Sec. 4.3), followed with an analysis on the computational performance (Sec. 4.4).

4.1 Implementation Details

Identification of Leading Vehicles. Although identifying leading vehicles is not the focus of this paper, we do need approaches to detect and keep tracking

[3]Note that intensity images are used only for identifying the leading vehicle.

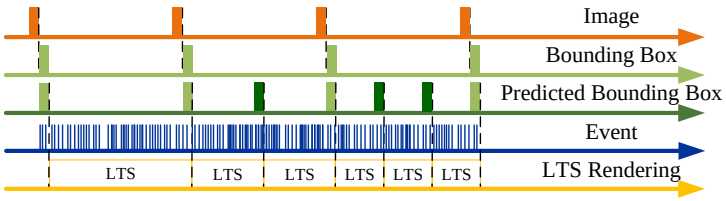


Fig. 6: Illustration of the adaptive LTS rendering.

the leading vehicle in the image plane. To this end, we utilize YOLO v5 [13] plus DeepSort [39] that constantly detect and locate the leading vehicle with a bounding box. The information of bounding box is not only used for indicating the presence of a leading vehicle but also restricting the size of the spatio-temporal volume from which the involved events are extracted.

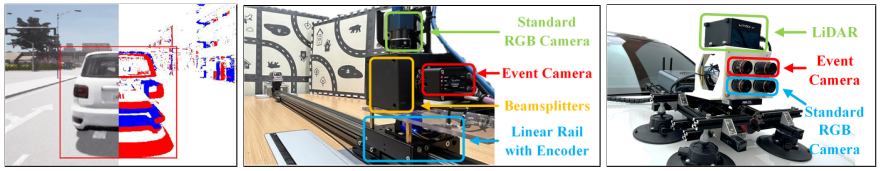
Rendering of LTS. To rationally allocate computational resources while preserving the success of spatio-temporal registration, we adopt an asynchronous strategy for rendering the LTS. Specifically, the LTS is rendered adaptively according to the expansion rate of the bounding box. To this end, we propose a simple and effective way to predict the bounding box (i.e., size and location) at anytime by utilizing the latest bounding box and affine model. Consequently, as an example shown in Fig. 6, the frequency of LTS rendering increases adaptively when the two vehicles get closer with no deceleration.

Robust Sampling. Although most of the background events are excluded by the bounding box, we still need a sampling strategy that further resists noises and outliers. To this end, we calculate the first-order and second-order image gradients of the LTS. Events will be reserved on the condition that the first-order gradient's magnitude is larger than 1×10^{-5}, and the second-order one's magnitude is lower than 1×10^{-3}.

Hyperparameters. We implement our robust linear solver on top of a standard RANSAC framework. Specifically, the maximum number of iterations is set to 300, and the inlier ratio is 0.9. As for the nonlinear spatio-temporal registration, we apply the Levenberg-Marquardt (LM) algorithm to iteratively solve Eq. 5. Once initialized, 10 iterations are typically enough for convergence.

4.2 Datasets

Three datasets are used to evaluate our method, and all these datasets capture scenarios featuring a leading vehicle followed by the host vehicle driving in the same lane. The first one is a fully synthetic dataset created using CARLA [7], an open-source simulator for urban and suburban driving (see Fig. 7a). We synthesize a hetero system that consists of a standard RGB camera and an event-based camera, which share identical intrinsic and extrinsic parameters (w.r.t the body coordinate of the host vehicle). The synthetic dataset consists of three subsets featuring different motion patterns, scenes, and different types of leading



(a) Synthetic dataset. **(b)** Small-scale test platform. **(c)** Multi-sensor suite.

Fig. 7: Illustration of our datasets and devices for data collection. (a) A selected snapshot of the synthetic dataset, on each side of which the intensity information and the events (with a naive accumulation) are illustrated, respectively. (b) The configuration of the small-scale test platform. (c) The multi-sensor suite mounted on a car.

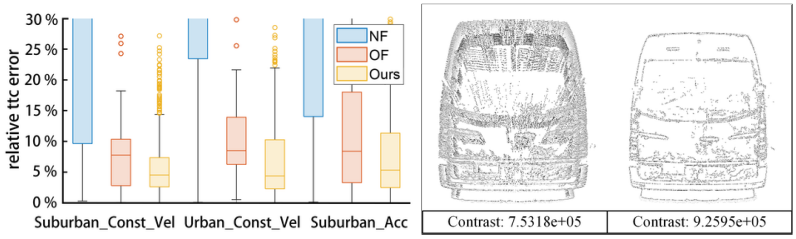
vehicles (*e.g.* Sedan, SUV, and Van, etc.). The second dataset is obtained by using a small-scale test platform that simulates the configuration of the TTC task (see Fig. 7b). The platform contains a hybrid optic system and a motorized slider. The hybrid optical system, mounted on the slider, consists of a standard camera, an event camera, and a beamsplitter. By spliting the incoming ray into two directions, the beamsplitter ensures both cameras share the same field of view. The position and the velocity of the slider can be controlled to emulate the relative motion between the leading vehicle (a model car) and the observer, and thus, the groundtruth TTC can be easily obtained. The third dataset, denoted by FCWD, consists of real data collected using a multi-sensor suite (including a pair of standard cameras, a pair of event cameras, and a LiDAR) as shown in Fig. 7c. The multi-sensor suite is mounted on the host vehicle that approaches the leading one at different speed. The groundtruth TTC can be obtained by differentiating the LiDAR data between the two cars. The characteristics of each dataset can be found in Table. 1, and are detailed in the supplementary material.

4.3 Quantitative and Qualitative Evaluation

To prove the effectiveness of the proposed geometric measurement, we first compare the performance of our initialization method using as measurements the normal flow (NF), the optical flow (OF), and the proposed geometric measurement (Ours), respectively. Note that the normal flows are estimated from event

Table 1: Characteristics of our datasets.

Dataset Name	Platform	Terrain	Sensor Motion	Event Cameras	RGB Cameras	Total Sequences
Synthetic	CARLA	Urban Suburban	Const. Vel Acceleration	CARLA DVS 640×480	CARLA RGB 640×480	21
Slider	Rail & Slider	Indoor	Const. Vel	inivation DVXplorer 640×480	DAHENG MER2 1440×1080	3
FCWD	Car	Urban	Normal Driving	Prophesee EVKv4 $\times 2$ 1280×720	FLIR Blackfly S $\times 2$ 1920×1200	3



(a) Statistics of TTC results. **(b)** Event alignment result.

Fig. 8: Comparison of our initialization method using different measurements. (a): Statistics of the relative TTC error from different methods illustrated with a box plot. The boxes and whiskers of NF's results are not shown completely for compactness. (b): Results of event alignment (in the format of images of warped events) with models fitted using NF (left) and the proposed geometric measurement (right), respectively.

Table 2: Quantitative evaluation on synthetic data.

Method	*Suburban_Const_Vel*		*Urban_Const_Vel*		*Suburban_Acc*	
	e_{TTC}	Runtime (s)	e_{TTC}	Runtime (s)	e_{TTC}	Runtime (s)
CMax [10]	3.76%	1.37	6.80%	0.94	5.06%	1.32
Our Init + CMax [10]	3.25%	0.82	5.87%	0.54	4.03%	0.68
FAITH [6]	88.6%	0.208	60.3%	0.176	87.2%	0.155
Image's FoE [36]	7.80%	0.037	8.46%	0.039	8.59%	0.037
ETTCM [26]	7.88%	0.031	8.43%	0.034	7.90%	0.029
Ours	4.29%	0.009	4.78%	0.008	3.58%	0.01

data using the method in [3], and the optical flows are approximated by using feature correspondences on intensity images[4]. We run the three initializers on all sequences of the synthetic data and analyze the statistics of the results. For quantitative evaluation, we use as metric the relative TTC error, defined as $e_{\text{TTC}} = |\frac{t_{gt}-t_{est}}{t_{gt}}| \times 100\%$, where t_{gt} denotes the GT TTC value and t_{est} the estimated TTC value. As shown in Fig. 8a, our initializer is on par with the OF-based one, and both outperform remarkably the NF-based one. Besides, we compare the resulting geometric models by illustrating the spatially-and-temporally aligned events. As seen in Fig. 8b, the model fitted using our geometric measurement leads to better alignment of events and a higher contrast.

We also compare our method against several alternative solutions in terms of TTC accuracy and computational efficiency. Methods used for comparison consist of the *CMax* method [10], an improved version of *CMax* initialized using our initializer, an event-based method (*FAITH* [6]) based on Focus of Expansion (FoE) estimation, an image-based method (*Image's FoE* [36]) that also uses FoE estimation, and an open-source event-based method called *ETTCM* [26].

We report the average value of e_{TTC} and runtime of each method. As shown in Table. 2 and Table. 3, *CMax* and its initialized version generally achieve the

[4]Note that intensity images are used only for providing full optical flow in the comparison, and they are not available in our method for geometric model fitting.

Table 3: Quantitative evaluation on real data.

Method	*Slider_500*		*Slider_750*		*Slider_1000*	
	e_{TTC}	Runtime (s)	e_{TTC}	Runtime (s)	e_{TTC}	Runtime (s)
CMax [10]	4.18%	0.90	4.16%	0.85	2.74%	0.93
Our Init + CMax [10]	4.18%	0.63	4.15%	0.76	2.74%	0.57
FAITH [6]	37.58%	0.151	34.45%	0.186	47.94%	0.233
Image's FoE [36]	12.37%	0.013	11.35%	0.013	7.55%	0.012
ETTCM [26]	16.12%	0.405	18.99%	0.498	15.20%	0.191
Ours	10.02%	0.017	8.95%	0.015	12.43%	0.016
Method	*FCWD 1*		*FCWD 2*		*FCWD 3*	
	e_{TTC}	Runtime (s)	e_{TTC}	Runtime (s)	e_{TTC}	Runtime (s)
CMax [10]	2.42%	2.53	2.39%	3.12	3.04%	3.34
Our Init + CMax [10]	2.33%	1.94	2.26%	2.38	2.97%	2.59
FAITH [6]	25.49%	0.214	27.91%	0.203	39.15%	0.210
Image's FoE [36]	5.39%	0.017	5.70%	0.018	6.02%	0.017
ETTCM [26]	15.52%	0.307	18.45%	0.282	19.08%	0.295
Ours	9.84%	0.017	11.55%	0.023	14.09%	0.025

best accuracy. This is because *CMax* uses all events within the spatio-temporal volume defined by the bounding box, enhancing accuracy with a better signal-to-noise ratio. However, the non-least-squares nature of *CMax* results in high computation complexity, making it unsuitable for real-time applications like TTC estimation (*e.g.*, it takes *CMax* 3 s to return an result from $2e5$ events occurred during an interval of 18 ms). *FAITH* shows a potential for real-time applications. However, its success heavily relies on the sense that the normal flow distributed uniformly around the FoE. Additionally, simply using normal flow as a replicate of optical flow leads to inaccurate TTC estimation results. The *Image's FoE* method performs normally good in terms of both metrics. However, it requires two successive images as input, and the runtime reported does not consider the time interval between two successive exposures ($30 \sim 100$ ms). This system latency induced by standard cameras can be a fatal delay to online control and decision making. As an event-based incremental methodology, *ETTCM* does not scale well with input data generated per unit time. Our method outperforms *ETTCM* in both metrics. Thanks to the robust sampling strategy and efficient initializer, our method achieves stable runtime in various scenarios. Since the input of each method is different, it is hard to set a criterion for counting runtime in an absolutely fair manner. Most methods for comparison are run with default parameters. Some of the event-based methods (*e.g.*, *CMax*) need a fine tune on the number of events used according to the spatial resolution of the event camera used. The mean runtime is reported for each sequence, and detailed configuration of each method can be found in the supplementary material. In conclusion, our method balances accuracy and efficiency.

In addition, we also conduct an experiment to investigate how sensitive our method is to the lateral offset of the preceding vehicle. We add a range of lateral offsets (*i.e.*, $[-4, 4]$ m) to the preceding vehicle in our synthetic dataset, and evaluate the performance of our TTC method. As seen in Fig. 9, the relative TTC error is always below 10% as long as the lateral offset is smaller than 3

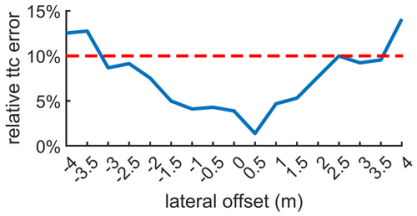


Fig. 9: Sensitivity analysis on the lateral offset of the preceding car.

Node (#Threads)	Function	Time (ms)
Vehicles Identification (GPU)	Yolov5	11
	DeepSort	6
Data Pre-processing (1)	LTS Rendering	1 - 3
Initialization & Reset (1)	Robust Sampling	5
	Linear Solver	10
Spatio-Temporal Registration (1)	Non-Linear Solver	5 - 10

Table 4: Computational performance.

m, indicating that our method has certain robustness against violation of the assumption on application scenarios.

4.4 Computational Performance

The proposed FCW system is implemented in C++ on ROS and runs in real-time on a laptop with an AMD Ryzen 5800H CPU and an NVIDIA RTX 3060 GPU. GPU is only used for Vehicles Identification node. The computational performance of each node is summarized in Table. 4. Note that these numbers reported are obtained when using an RGB camera (1440×1080 pixels) and an event camera (640×480 pixels). Every node takes one independent thread, and their runtime are listed in the right-most column. As observed in our experiments, the initialization is usually applied only once, and all subsequent updates can be done by constantly calling the spatio-temporal registration. Thus, our system can output TTC up to 200 Hz.

It is also worth mentioning that the advantage of our method is not only in terms of computation efficiency, but also in terms of system latency. If the exposure time of a standard camera is ignored, the latency of pure image-based methods consists of the time interval between two successive exposures (e.g., 33 ms for a 30-fps camera), and the computation time (e.g., 40 ms by [28, 36]). The overall system latency is 73 ms. For our method, the average latency of our algorithm is mainly caused by the computation time (5-10 ms once initialized), indicating a great potential for handling sudden variations in relative speed.

5 Conclusion

This paper proposes an event-based method for estimating the time to a potential collision with a preceding vehicle. To the best of our knowledge, this is the first event-based TTC solution for autonomous/assisted driving scenarios. The proposed two-step approach can estimate efficiently the TTC via fitting a geometric model to raw event data. Experiments demonstrate the effectiveness of the proposed method on both synthetic and real data, showing that our method has the best overall performance in terms of accuracy and speed. We will release the software together with the datasets to the community and wish that the work will serve as a benchmark for future research on event-based TTC estimation.

Acknowledgment

We thank Javier Hidalgo-Carrió and Davide Scaramuzza for releasing the design of the Beamsplitter [11], based on which we build our FCW system. We also thank Dr. Yi Yu for proof reading. This work was supported by the National Key Research and Development Project of China under Grant 2023YFB4706600.

References

1. Almatrafi, M., Baldwin, R., Aizawa, K., Hirakawa, K.: Distance surface for event-based optical flow. IEEE transactions on pattern analysis and machine intelligence **42**(7), 1547–1556 (2020)
2. Badki, A., Gallo, O., Kautz, J., Sen, P.: Binary ttc: A temporal geofence for autonomous navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12946–12955 (2021)
3. Benosman, R., Clercq, C., Lagorce, X., Ieng, S.H., Bartolozzi, C.: Event-based visual flow. IEEE transactions on neural networks and learning systems **25**(2), 407–417 (2013)
4. Clady, X., Clercq, C., Ieng, S.H., Houseini, F., Randazzo, M., Natale, L., Bartolozzi, C., Benosman, R.: Asynchronous visual event-based time-to-contact. Frontiers in neuroscience **8**, 9 (2014)
5. Dagan, E., Mano, O., Stein, G.P., Shashua, A.: Forward collision warning with a single camera. In: IEEE Intelligent Vehicles Symposium, 2004. pp. 37–42. IEEE (2004)
6. 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)
7. 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)
8. Falanga, D., Kleber, K., Scaramuzza, D.: Dynamic obstacle avoidance for quadrotors with event cameras. Science Robotics **5**(40), eaaz9712 (2020)
9. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
10. 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)
11. 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)
12. Jähne, B., Haussecker, H., Geissler, P.: Handbook of computer vision and applications, vol. 2. Citeseer (1999)
13. Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., Kwon, Y., Michael, K., Fang, J., Yifu, Z., Wong, C., Montes, D., et al.: Ultralytics/yolov5: v7. 0-yolov5 sota realtime instance segmentation. Zenodo (2022)
14. Lee, D.N.: A theory of visual control of braking based on information about time-to-collision. Perception **5**(4), 437–459 (1976)

15. Lee, D.N.: The optic flow field: The foundation of vision. Philosophical Transactions of the Royal Society of London. B, Biological Sciences **290**(1038), 169–179 (1980)
16. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128x128 120dB 30mW asynchronous vision sensor that responds to relative intensity change. In: IEEE Intl. Solid-State Circuits Conf. (ISSCC). pp. 2060–2069 (2006)
17. Lin, S., Xu, F., Wang, X., Yang, W., Yu, L.: Efficient spatial-temporal normalization of sae representation for event camera. IEEE Robotics and Automation Letters **5**(3), 4265–4272 (2020)
18. Liu, D., Parra, A., Chin, T.J.: Spatiotemporal registration for event-based visual odometry. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4937–4946 (2021)
19. Liu, M., Delbruck, T.: Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors. BMVC (2018)
20. Manglik, A., Weng, X., Ohn-Bar, E., Kitanil, K.M.: Forecasting time-to-collision from monocular video: Feasibility, dataset, and challenges. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 8081–8088. IEEE (2019)
21. 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: European Conference on Computer Vision. pp. 3–20. Springer (2022)
22. Meyer, F., Bouthemy, P.: 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, pp. 78–82. IEEE Computer Society (1992)
23. Meyer, F.G.: Time-to-collision from first-order models of the motion field. IEEE transactions on robotics and automation **10**(6), 792–798 (1994)
24. Negre, A., Braillon, C., Crowley, J.L., Laugier, C.: Real-time time-to-collision from variation of intrinsic scale. In: Experimental Robotics: The 10th International Symposium on Experimental Robotics. pp. 75–84. Springer (2008)
25. Nelson, R.C., Aloimonos, J.: Obstacle avoidance using flow field divergence. IEEE Transactions on pattern analysis and machine intelligence **11**(10), 1102–1106 (1989)
26. 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)
27. Pan, L., Scheerlinck, C., Yu, X., Hartley, R., Liu, M., Dai, Y.: Bringing a blurry frame alive at high frame-rate with an event camera. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR) (2019)
28. Poiesi, F., Cavallaro, A.: Detection of fast incoming objects with a moving camera. In: British Mach. Vis. Conf. (BMVC) (2016)
29. Rebecq, H., Horstschaefer, T., Scaramuzza, D.: Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization. In: British Mach. Vis. Conf. (BMVC) (2017)
30. Rebecq, H., Ranftl, R., Koltun, V., Scaramuzza, D.: High speed and high dynamic range video with an event camera. IEEE Trans. Pattern Anal. Mach. Intell. (2019)
31. Rodríguez-Gómez, J.P., Tapia, R., Garcia, M.d.M.G., Martínez-de Dios, J.R., Ollero, A.: Free as a bird: Event-based dynamic sense-and-avoid for ornithopter robot flight. IEEE Robotics and Automation Letters **7**(2), 5413–5420 (2022)
32. Sanket, N.J., Parameshwara, C.M., Singh, C.D., Kuruttukulam, A.V., Fermüller, C., Scaramuzza, D., Aloimonos, Y.: EVDodgeNet: Deep dynamic obstacle dodging with event cameras. In: IEEE Int. Conf. Robot. Autom. (ICRA) (2020)

33. Shaw, D.C., Shaw, J.Z.: Vehicle collision avoidance system (Jun 25 1996), US Patent 5,529,138
34. Shiba, S., Aoki, Y., Gallego, G.: A fast geometric regularizer to mitigate event collapse in the contrast maximization framework. Advanced Intelligent Systems p. 2200251 (2022)
35. Souhila, K., Karim, A.: Optical flow based robot obstacle avoidance. International Journal of Advanced Robotic Systems **4**(1), 2 (2007)
36. 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)
37. Walters, C., Hadfield, S.: Evreflex: Dense time-to-impact prediction for event-based obstacle avoidance. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1304–1309. IEEE (2021)
38. Widman, G., Bauson, W.A., Alland, S.W.: Development of collision avoidance systems at delphi automotive systems. In: Proc. Int. Conf. Intelligent Vehicles. pp. 353–358. Citeseer (1998)
39. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: 2017 IEEE International Conference on Image Processing (ICIP). pp. 3645–3649. IEEE (2017)
40. Zhou, Y., Gallego, G., Lu, X., Liu, S., Shen, S.: Event-based motion segmentation with spatio-temporal graph cuts. IEEE Transactions on Neural Networks and Learning Systems (2021)
41. Zhou, Y., Gallego, G., Shen, S.: Event-based stereo visual odometry. IEEE Transactions on Robotics **37**(5), 1433–1450 (2021)
42. Zhou, Y., Li, H., Kneip, L.: Canny-vo: Visual odometry with rgb-d cameras based on geometric 3-d–2-d edge alignment. IEEE Transactions on Robotics **35**(1), 184–199 (2018)
43. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: EV-FlowNet: Self-supervised optical flow estimation for event-based cameras. In: Robotics: Science and Systems (RSS) (2018)
44. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: Unsupervised event-based learning of optical flow, depth, and egomotion. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR) (2019)