

EventEye: Towards High-Frequency Perception Enhancement for Autonomous Vehicles Using Infrastructure-Mounted Event Cameras

Jingfei Xia
The Chinese University of Hong Kong
Hong Kong, China
xj022@ie.cuhk.edu.hk

Yuze He
Carnegie Mellon University
Pittsburgh, USA
yuzeh@cs.cmu.edu

Chen Bian
The Chinese University of Hong Kong
Hong Kong, China
1155230455@link.cuhk.edu.hk

Zhenyu Yan
The Chinese University of Hong Kong
Hong Kong, China
zyyan@cuhk.edu.hk

Guoliang Xing*
The Chinese University of Hong Kong
Hong Kong, China
glxing@cuhk.edu.hk

Abstract

Infrastructure-assisted autonomous driving is an emerging paradigm expected to greatly enhance the safety of autonomous vehicles. Current systems primarily use LiDAR or RGB cameras as the infrastructure sensors, which suffer from high end-to-end latency and environmental susceptibility, respectively. In this work, we propose EventEye, the first system that integrates event cameras into infrastructure to enhance a vehicle's real-time perception by seamlessly fusing high-frequency event data with the vehicle's LiDAR data. The core concept of EventEye is to leverage the inherent sparsity of event camera data on the infrastructure to estimate Bird's-Eye-View (BEV) occupancy maps with grid flows recurrently. The generated occupancy map is then merged in real time with vehicle LiDAR-generated maps. We implement EventEye on a real-world testbed and evaluate the performance of EventEye across various scenarios. EventEye achieves an error of around 0.3 m with a processing latency of less than 25 ms. Compared to previous systems, this enables 51% earlier detection of incoming objects, providing autonomous vehicles with crucial additional reaction distance.

CCS Concepts

• Computer systems organization → Sensor networks.

Keywords

Event Camera, Heterogeneous Sensor Fusion, Infrastructure-assisted Autonomous Driving, Vehicle-Infrastructure Information Fusion

ACM Reference Format:

Jingfei Xia, Yuze He, Chen Bian, Zhenyu Yan, and Guoliang Xing. 2026. EventEye: Towards High-Frequency Perception Enhancement for Autonomous Vehicles Using Infrastructure-Mounted Event Cameras. In *ACM/IEEE International Conference on Embedded Artificial Intelligence and*

*Corresponding author.



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

SenSys '26, Saint Malo, France

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2309-4/26/05

<https://doi.org/10.1145/3774906.3800471>

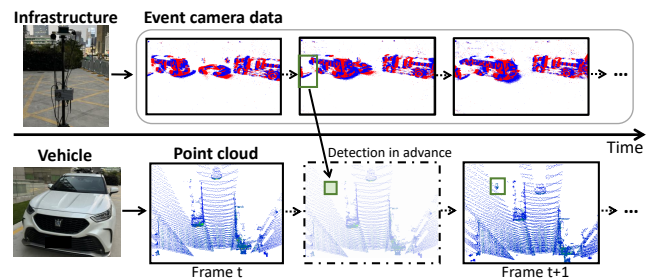


Figure 1: Enabling high-frequency 3D perception of vehicles using infrastructure-mounted event cameras.

Sensing Systems (SenSys '26), May 11–14, 2026, Saint Malo, France. ACM, New York, NY, USA, 15 pages. <https://doi.org/10.1145/3774906.3800471>

1 Introduction

Autonomous driving has the potential to revolutionize urban mobility and road safety in smart cities. Despite its promise, recent accidents have underscored significant public concerns regarding the reliability and safety of autonomous vehicle systems [5, 57]. Infrastructure-assisted autonomous driving offers a promising solution to these challenges. This approach leverages intelligent roadside infrastructure [43, 44, 46, 60] equipped with sensors to provide vehicles with enhanced environmental awareness through broader, unobstructed viewpoints. Such capability enables vehicles to extend their perception beyond direct line-of-sight limitations, substantially improving navigational safety and operational reliability.

Currently, many roadside infrastructure systems [23–25, 32, 54] employ LiDAR and RGB cameras for perception. However, these conventional sensors present significant limitations. Commercial off-the-shelf LiDAR systems, for example, operate at a 10Hz frame rate, introducing a variable latency of up to 100 milliseconds before an event is captured by the infrastructure's LiDAR. Also, the large data volumes produced by LiDAR point clouds necessitate significant processing time. As a result, there is a cumulative delay in acquiring, processing, and transmitting LiDAR data, which can result in outdated information by the time it reaches the vehicle, thus reducing its utility and presenting potential safety hazards. The significance of this latency issue is highlighted by previous work [41] showing that reducing latency by around 50 ms can decrease accidents by 47%. These limitations underscore the need for

alternative sensing solutions in infrastructure-assisted autonomous driving.

Recently, event cameras have emerged as a promising sensor technology in the fields of computer vision [15, 17, 78] and autonomous driving [16, 65, 77]. During the past few years, the cost of event cameras has decreased by 10× to approximately 500 USD [51], making them increasingly viable for real-world applications. In this work, we propose integrating event cameras with roadside infrastructure to enhance the real-time perception of vehicles equipped with LiDAR. Unlike traditional RGB cameras, event cameras capture changes in per-pixel brightness through asynchronous events with microsecond precision. This unique capability presents several key advantages when paired with infrastructure: 1) Event cameras offer exceptional temporal resolution up to 10^6 Hz, drastically reducing data acquisition times compared to commonly used LiDAR and RGB cameras. Additionally, their acquisition frequency can be adjusted through event data accumulation time, enabling flexible synchronization between infrastructure and vehicles. 2) When mounted on infrastructure, event cameras are stationary, and hence capture dynamic objects only. This naturally filters out less relevant background information right at the data acquisition stage and eliminates the need for time-consuming object extraction. 3) Given that dynamic objects typically represent only a small fraction of the entire scene, the resulting volume of event data is significantly reduced, thus minimizing data transmission time. 4) In addition, the high dynamic range and minimal motion blur of event cameras make them particularly effective for use in low-light conditions.

Despite their advantages, event cameras present a significant challenge in sensor fusion: bridging the gap between 2D event data from infrastructure-mounted event cameras and 3D point clouds from vehicle-mounted LiDAR systems. The fundamental differences in data representation - sparse, asynchronous events versus dense, structured point clouds - make it difficult to establish direct correspondences and perform real-time alignment. Additionally, the continuous motion of vehicles relative to stationary infrastructure creates a dynamic perspective gap, varying viewpoints and occlusion patterns, further complicating the perception fusion process.

In this paper, we propose EventEye, a novel event camera-based infrastructure system that delivers enhanced high-frequency perception to the vehicles and aligns the event camera perception with LiDAR perception on the vehicles. Our system strategically positions event cameras on the infrastructure to provide timely information to vehicles with LiDAR. To bridge the heterogeneous sensor gap, we utilize Bird's Eye View (BEV) occupancy grid maps as an intermediate representation. This approach discretizes the world into 2D grids indicating traffic participant occupancy, enabling unified representation by converting both 2D event data and 3D LiDAR point clouds into a common BEV format. This effectively addresses the challenges of different dimensionalities and data structures. To fuse the BEV occupancy map from both the infrastructure and the vehicle, which lacks rich semantic and geometry information on the stationary environment and exhibits significant perception disparities, we estimate the flow of occupied grids and jointly leverage the motion and position relationships of the occupied grids to achieve accurate alignment.

EventEye offers several key advantages. 1) The system requires only a single event camera on infrastructure and works with existing vehicle sensors, without demanding specific camera poses or precise infrastructure locations for perception fusion, making it suitable for widespread real-world deployment. 2) By leveraging the inherent sparsity of event camera data, EventEye minimizes computational overhead on resource-constrained infrastructure, thereby reducing end-to-end latency. 3) The generated BEV occupancy map and flow align the data formats of heterogeneous sensors, which can be integrated into downstream autonomous driving tasks such as navigation and planning [9, 12, 31], lowering the barrier of adoption. This representation is also lightweight, further reducing the infrastructure-vehicle communication overhead. 4) The system's efficient processing and the event camera's high temporal resolution allow the vehicles to detect objects earlier, providing crucial safety margins, especially for sudden obstacles and blind spots.

To evaluate our system, we implemented EventEye on a real-world testbed, utilizing a real vehicle paired with a portable roadside unit. We conducted extensive testing across five distinct road sections in an urban environment, collecting data from 27 infrastructure-vehicle pairs. Additionally, we collected another dataset in the autonomous driving simulator Carla [11] with 60 pairs of data. Experimental results demonstrate that EventEye achieves approximately 0.3 m trajectory error, which is sufficient for L4 autonomous driving according to Baidu Apollo's white paper [61]. The system achieves processing latency under 25 ms—a 26% reduction compared to existing infrastructure systems. Moreover, it enables autonomous vehicles to detect incoming objects more than 60ms earlier (51% reduction) compared to previous systems, yielding 2-3m additional reaction distance. The results show that EventEye can substantially enhance traffic safety through its superior latency reduction and increased reaction distance.

2 Related Work

Event Camera in Autonomous Driving. As an emerging sensor, the event camera has attracted substantial attention for its exceptional features, including high temporal resolution, low latency, low power consumption, and high dynamic range. It has been widely used for feature tracking [34, 45], image deblurring [47, 71], and obstacle avoidance [53, 63] in robotics and computer vision. Its unique ability to capture asynchronous, high-frequency changes in a scene makes it particularly advantageous for scenarios requiring real-time processing and quick decision-making, which makes it particularly suitable for autonomous driving. Previous work [6] has identified promising applications including sensor fusion, active vision systems, and sim-to-real learning. To support research in event-based vision for autonomous driving, several datasets have been collected, including MVSEC [77] and DSEC [16]. Based on the datasets, researchers have made significant progress in various tasks, including semantic segmentation [58, 69], optical flow estimation [17, 56, 78], and depth estimation [15, 20, 26]. These works typically assume event cameras are mounted on autonomous vehicles. However, adopting new sensors like event cameras on vehicles poses several challenges due to added costs, safety regulations, and the need to redesign data processing pipelines. A few recent studies [1, 62] have begun to explore their use on the

infrastructure, focusing on roadside perception tasks like motion detection and object tracking. However, these studies overlook the potential for cooperative perception between vehicles and roadside infrastructure equipped with event cameras, which could enable more robust and comprehensive perception systems.

Infrastructure-assisted Vehicle Perception. To address the limitation of a restricted perception range faced by individual vehicles, infrastructure-assisted autonomous driving offers a promising and effective solution. Some systems focus on providing application-level results to the vehicles, such as planning [27, 67] and mapping [7, 23]. For the perception system, based on the fusion strategy, they can be categorized into different types. Early-fusion systems are designed to transmit raw sensor data from infrastructure to vehicles and achieve accurate fusion results. However, VI-Eye [25] requires the transmission of raw point clouds and performs point cloud segmentation for alignment, resulting in significant computation and communication overhead. VRF [32] reduces this overhead but relies on a pre-built 3D environment map for the vehicles, which makes it vulnerable to environmental changes and not always available for the vehicles. To reduce the communication overhead, late-fusion systems [28, 52, 54] process the data on the infrastructure and only transmit the perception results to the vehicles. In particular, these methods focus primarily on aligning identical sensors, such as LiDAR and RGB cameras, on both sides. Mid-fusion methods [39, 42, 80] fuse heterogeneous sensors by aligning transmitted feature representations using neural networks. However, these approaches require joint training of models and incur high communication overhead due to feature transmission. Different from these methods, EventEye is the first to introduce event cameras to the infrastructure and efficiently fuse their data with vehicle-mounted LiDAR.

3 A Motivational Study

In this section, we briefly introduce event cameras and provide a comprehensive comparison with commonly used sensors, such as LiDAR and RGB cameras, highlighting their limitations to demonstrate the benefits of using event cameras in infrastructure-assisted autonomous driving systems.

3.1 Event Camera

Unlike traditional frame-based sensors, event cameras are bio-inspired devices that operate asynchronously, detecting per-pixel brightness changes with microsecond-level precision. Rather than capturing full frames at fixed intervals, they generate a stream of events when pixel brightness changes occur. Each event is encoded as (t, x, y, p) , where t is the timestamp, (x, y) are pixel coordinates, and p indicates the brightness change polarity (increase or decrease). A key advantage of event cameras is their ability to operate independently at the pixel level with a high dynamic range, enabling them to provide timely and reliable information under varying lighting conditions. This makes event cameras particularly well-suited for real-time applications and low-light environments. When mounted on static infrastructure, event cameras activate pixels only for moving objects while ignoring static backgrounds. This data sparsity reduces computational and bandwidth requirements, making event cameras a promising solution for enhancing

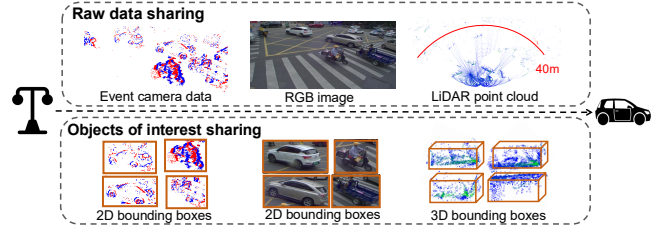


Figure 2: Comparison of three sensors in two typical vehicle-infrastructure cooperative patterns.

	LiDAR	RGB camera	Event camera
Refresh rate	10 Hz	30 Hz	up to 10^6 Hz
Raw data size (60s)	523.8 MB	230.1 MB	116.9 MB
Effective range	40-50 m	40 m	40 m
E2E Latency (Raw)	164.3 ms	104.9 ms	14.6 ms
E2E Latency (Objects)	115.9 ms	42.1 ms	11.8 ms

Table 1: Performance comparison of LiDAR, RGB camera, and event camera as infrastructure sensors.

the efficiency and effectiveness of infrastructure in autonomous driving systems.

3.2 Sensor Comparison

We deployed a portable platform on a real-world city road, equipped with side-by-side sensors of comparable specifications, including similar prices (around 1500 USD) and comparable fields of view (both vertical and horizontal). The setup includes a LIVOX AVIA LiDAR, a Basler RGB camera, and a Prophesee event camera to evaluate their performance in an infrastructure-based system. We use WiFi for data transmission between the vehicle and the infrastructure. The comparative results of these sensors are presented in Table 1. Specifically, we evaluate the overall latency of the system with various infrastructure sensors following the two typical cooperative patterns prevalent in current infrastructure systems [23–25, 54] as depicted in Fig. 2: either direct transmission of raw data or selective transmission of objects of interest such as vehicles and pedestrians. To ensure a fair comparison, we extract 2D bounding boxes from event camera data [62] to represent objects of interest and transmit them to the vehicle. We define the end-to-end latency as $t_{e2e} = t_{acquire} + t_{compute} + t_{transmit}$, where $t_{acquire}$, $t_{compute}$ and $t_{transmit}$ indicate data acquisition, computation and transmission time, respectively.

LiDAR. LiDAR is widely used in autonomous driving due to its ability to provide precise and long-range depth information. However, the point cloud becomes exceedingly sparse beyond 40-50 meters as shown in Fig. 2, making reliable perception difficult. However, the point cloud data tends to become exceedingly sparse beyond 40-50 meters as shown in Fig. 2, making it hard to generate reliable perception results. LiDAR operates at a 10 Hz fixed frame rate, generating frames every 100 ms, with each frame containing 48,000 points, accumulating to 523.8 MB within 60 seconds. Direct transmission of this data incurs significant communication overhead and an end-to-end latency exceeding 160 ms (100 ms capture + 64.3 ms transmission). Reducing data size by extracting objects of interest adds computational tasks but still results in over 110 ms of

latency (100 ms capture, 13.4 ms processing, 2.5 ms transmission) as demonstrated in Table 1. In a scenario where a vehicle travels at 20 m/s (72 km/h), it could move 2-3 meters by the time it receives updated information from the infrastructure. This delay represents a significant safety hazard in complex traffic environments.

RGB Camera. RGB cameras are also widely used in the infrastructure system due to their availability and low cost. However, RGB cameras also experience high latency due to large data sizes and low frame rates—104.9 ms for direct transmission and 42.1 ms for object transmission. More importantly, a significant drawback of RGB cameras is that they are highly susceptible to lighting conditions, particularly during nighttime or in direct sunlight. Although RGB cameras can mitigate this problem by increasing the exposure time, they inadvertently introduce motion blur for fast-moving objects and elevate the end-to-end latency. Together, these factors degrade the perception quality of the RGB camera, making it unsuitable for use as the infrastructure sensor. Furthermore, installing RGB cameras on the infrastructure will cause privacy issues and raise public concerns.

Event Camera. Event cameras can significantly reduce data size due to their unique operating principle. As the data size of the event camera varies depending on traffic conditions, Table 1 calculates raw data sizes over a 60-second interval to ensure fair comparison. The event camera generates only one-fifth of LiDAR's data volume and half of an RGB camera's output. Table 1 shows that the event camera reduces raw data transmission time to 14.6 ms. Additionally, it significantly lowers the end-to-end latency for transmitting objects of interest to 11.8 ms, which is 9.8× faster than LiDAR and 3.6× faster than the RGB camera. Furthermore, the event camera's high dynamic range ensures robust performance under various lighting conditions, giving it a clear advantage over the RGB camera [14].

4 System Design

In this section, we introduce EventEye, an event camera-based infrastructure system that delivers refined, near-instantaneous information to autonomous vehicles, facilitating rapid vehicle responses to complex road conditions. Our design objective is to leverage the high temporal resolution and sparsity of event camera data to capture dynamic and real-time traffic dynamics while minimizing infrastructure overhead. Moreover, the raw event camera data must be processed into a compatible format to seamlessly align with the LiDAR point cloud from the vehicle. The integrated perception results from EventEye can support autonomous driving systems in various downstream tasks, such as motion planning, behavior prediction, and motion estimation.

However, incorporating event cameras into infrastructure systems poses significant challenges due to heterogeneous sensor fusion complexities and substantial perception gaps. To the best of our knowledge, no existing methods have explored the fusion of event camera data with LiDAR data under scenarios with large perception gaps. To address this challenge, EventEye transforms event camera data and LiDAR point clouds into Bird's Eye View (BEV) space, creating a unified and compact BEV occupancy map that aligns data from these heterogeneous sensors. Unlike LiDAR, event cameras capture data without depth information, making direct Bird's Eye View (BEV) transformation challenging. Since no

existing methods handle BEV estimation using asynchronous event camera data, we propose a novel approach that selectively processes event data alongside historical BEV features, achieving accurate reconstruction using only monocular event cameras. Each occupancy map only contains the dynamic traffic participants like vehicles and pedestrians, excluding static and complicated environmental features. The perception gap between infrastructure and vehicles results in different occlusion patterns, making map fusion challenging. No existing methods effectively merge BEV maps with such large perception gaps. To achieve robust map fusion, our system considers both grid occupancy positions and their motion states, which we term "grid flow". Our key insight is that corresponding grids should maintain consistent flow patterns across infrastructure and vehicle perspectives after proper transformation. This joint utilization of positional and flow information enables precise transformation parameter estimation without relying on complex geometric features. As a result, our system effectively merges BEV maps from diverse viewpoints, producing unified representations that serve downstream applications such as navigation and obstacle avoidance.

As shown in Fig. 3, the system pipeline consists of three parts. First, the infrastructure leverages event camera sparsity and historical estimations to recurrently predict the BEV occupancy grid map and grid flow. Second, the vehicle detects and tracks objects, projecting results into a BEV grid map. Third, the infrastructure transmits its BEV map to the vehicle, where it is aligned with the vehicle-generated map by leveraging grid position and flow information to produce a comprehensive BEV occupancy map within the vehicle's view. The fused map integrates directly into the vehicle's existing processing pipeline without modifications, and the lightweight transformation and transmission process ensures near-instantaneous traffic updates.

4.1 Event-based BEV Estimation

To align heterogeneous sensor data, specifically 2D event camera data from infrastructure and 3D point clouds, a common approach is to perform a 3D transformation from the infrastructure view to the vehicle view. However, since event cameras provide only 2D information, a conventional method involves estimating depth from event data [26, 79] to project 2D pixels into 3D points. Unfortunately, this approach can introduce significant errors (ranging from 1.5 to 4.5 meters) in outdoor environments [13], which are unacceptable for autonomous vehicle applications.

Our idea is to fuse these two types of data in the bird's eye view (BEV) space. Our approach fuses 2D event data and 3D point clouds by converting both into bird's eye view (BEV) occupancy maps. A BEV occupancy map represents the world as a 2D matrix from a top-down perspective, where each grid cell is marked as occupied (1) or unoccupied (0), with each cell covering a small road region (40 cm, following [35, 74]). The binary nature of BEV occupancy ensures robustness by consistently marking grids as occupied regardless of object count or overlap, and circumvents the need for explicit depth estimation from event data. However, converting event camera data to a BEV occupancy map is challenging and underexplored. Existing BEV estimation methods [36, 48] are developed for RGB cameras and are incompatible with event cameras due to their

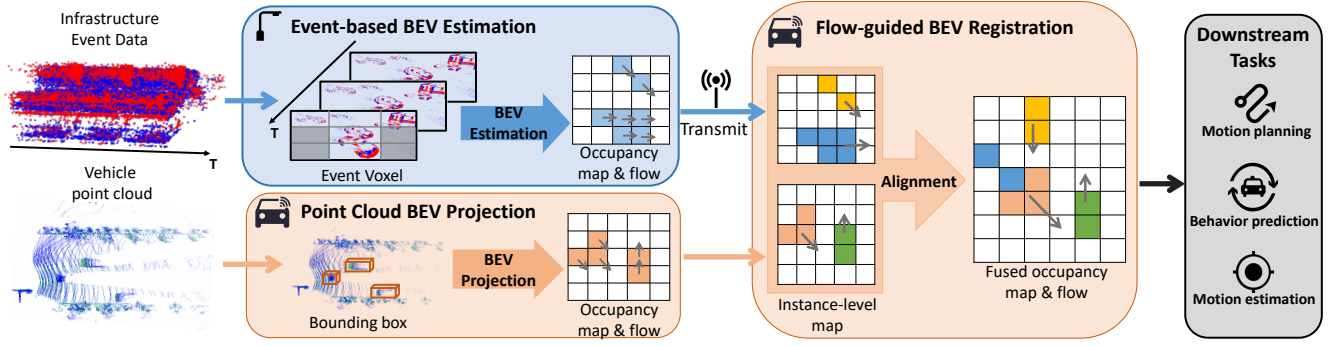


Figure 3: System architecture of EventEye. The blue box the operations on the infrastructure, and the orange boxes denote the operations on the vehicles.

fundamentally different imaging mechanisms and data structures. While recent event-based BEV mapping methods [26, 78] exist, they focus solely on grid occupancy and overlook flow information – the dynamic motion state of each grid cell, which is crucial for effective map fusion. Moreover, these methods process all pixels indiscriminately, failing to handle event camera noise, and do not exploit the inherent sparsity of event data in infrastructure settings, resulting in inefficient estimation.

Therefore, we propose a novel method that not only estimates BEV occupancy maps with flow information for infrastructure-mounted event cameras but also minimizes computational overhead by leveraging data sparsity and focusing on the very few activated pixels. Specifically, we first extract the geometric features from the event camera data and then estimate the height distribution of each pixel in the event data. Then, using the camera pose parameters and the estimated height distribution, we project the event features into the BEV space to determine which areas of the top-down BEV map are occupied. The framework is shown in Figure 4.

4.1.1 Event Representation. To begin with, we divide the flow into T overlapping fixed time windows and transform the event flow data into a compact matrix that is compatible with neural network input. Note that the time window refers to the accumulation period for processing event data. Unlike frame-based RGB cameras, event cameras are not restricted by fixed frame rates and can process data in overlapping windows (e.g., 0–20 ms, 10–30 ms), enabling higher temporal resolution and faster updates. Different from previous methods [26] that divide each time window into several bins and assign events to specific bins, our approach acknowledges that each event influences the entire time window, and its impact diminishes over time. Initially, we divide each time window into K temporal bins, where K is a hyperparameter. Within each window, we normalize event timestamps to span from 0 to K , mapping each event’s occurrence time to this normalized range: $t^* = \frac{t - t_{min}}{t_{max} - t_{min}}K$, where t^* is the normalized timestamp. t_{max} and t_{min} are the maximum and minimum timestamps within the time window. For temporal encoding, we employ K Gaussian functions whose mean values are uniformly spaced from 0 to $K-1$ to compute feature maps for each temporal bin. Each event contributes to these feature maps based on its normalized timestamp, with its influence determined by the Gaussian functions, resulting in a smooth temporal distribution across the bins. For the k -th bin, the corresponding feature map

can be computed as:

$$E_k = \sum_{e_i} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(t_i^* - k)^2}{2\sigma^2}\right) \delta((x, y) - (x_i, y_i)) p_i$$

where $e_i = (t_i, x_i, y_i, p_i)$ is the single event data in the time window and $\delta(\cdot)$ is the delta function indicating the position of the event. Since we have K functions, we can obtain K feature maps. We concatenate them to obtain an event time voxel $E_{voxel} \in \mathbb{R}^{K \times H \times W}$, where H and W are the camera resolution.

4.1.2 Sparse Feature Extraction. After processing event data representations, we input the event voxel into a ResNet [22] and an FPN encoder [73] to extract multi-scale event feature map F_e . Event cameras mounted on infrastructure primarily detect moving traffic participants, resulting in sparse event data with predominantly zero-valued pixels. Moreover, road conditions exhibit temporal variations, with low-traffic periods generating even sparser event representations. Consequently, processing such sparse data using traditional frame-based pipelines would incur unnecessary computational overhead. To optimize computational efficiency, we adopt a patch-based approach: dividing the event voxels into fixed-size patches and selectively processing only those containing events. The extracted features from these active patches are then reassembled into a comprehensive feature map according to their original spatial positions.

$$P_{active} = \{p_{i,j} | p_{i,j} \in \text{patch}(E_{voxel}), \sum p_{i,j} \neq 0\}$$

$$F_e = \bigcup \phi(p_{i,j}), \quad \forall p_{i,j} \in P_{active}$$

$\phi(\cdot)$ is the feature extraction function. To preserve spatial relationships during sparse patch processing, we employ learnable positional embeddings. We first restore the original event voxel size and define a positional embedding matrix with dimensions matching the number of patches. Each patch receives a unique positional encoding that captures its spatial location in the original feature map. A simple CNN then merges this positional information with the patch features, enabling the model to maintain accurate spatial awareness throughout processing.

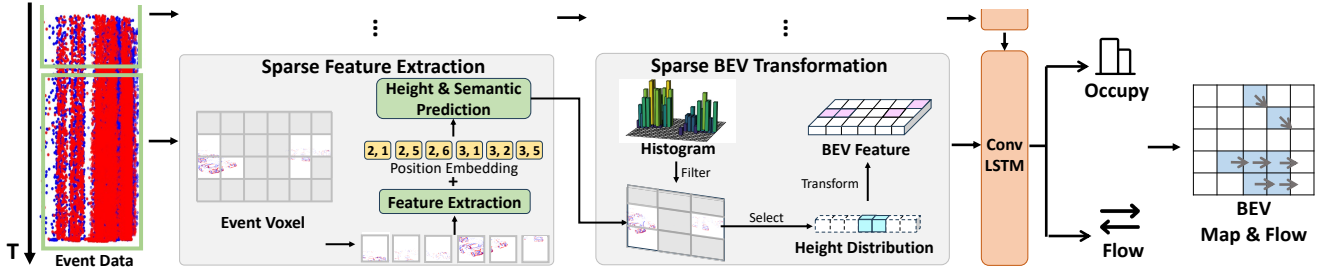


Figure 4: Illustration of estimating BEV occupancy grid map and grid flow from the event camera.

4.1.3 Selective BEV Transformation. We employ two ResNet blocks to estimate the height distribution $F_h \in \mathbb{R}^{C_h \times \frac{H}{r} \times \frac{W}{r}}$ and semantic features $F_s \in \mathbb{R}^{C_s \times \frac{H}{r} \times \frac{W}{r}}$ from event features, where r denotes the network downsampling rate. We merge them to obtain $F_{\text{fuse}} \in \mathbb{R}^{C_h \times C_s \times \frac{H}{r} \times \frac{W}{r}}$. Based on F_h , we derive a transformation matrix to project the fused features into BEV space, yielding BEV features F_{bev} . Mapping all height bins to BEV grids with their corresponding semantic features becomes computationally intensive, particularly at higher camera resolutions. To address this, we leverage the sparse nature of event cameras and introduce a dual-selective transformation strategy. First, we selectively transform height distributions: since event cameras capture traffic participants (vehicles and pedestrians) with relatively fixed heights, we only transform bins whose probability exceeds a threshold, discarding unlikely estimates. Second, we selectively transform semantic features based on likely traffic participant locations by constructing an event histogram from pixel-wise event counts and downsampling it via max pooling to form a pixel mask $H \in \mathbb{R}^{\frac{H}{r} \times \frac{W}{r}}$. Since higher histogram values typically correspond to traffic participants while lower values indicate environmental noise, we only transform regions where histogram values exceed a threshold into BEV features. This dual-selective approach reduces computational overhead by processing fewer pixels while also helping the model focus on meaningful objects, enabling efficient BEV conversion without sacrificing perception performance.

4.1.4 Recurrent BEV Estimation. After obtaining the BEV features F_{bev} , we proceed to predict the occupancy grid map and flow. To better utilize temporal relationships, we enhance BEV prediction by incorporating previously predicted BEV features as prior information, rather than relying on a single time window. Specifically, we obtain BEV features from multiple time windows and sequentially feed L BEV features into a ConvLSTM network [55] for recurrent prediction. This temporal integration approach significantly enhances the accuracy and robustness of our predictions, particularly for flow estimation, which characterizes the motion features of each grid. The motion patterns captured across consecutive time windows provide rich temporal context for understanding dynamic object behaviors. Finally, the ConvLSTM output is processed through a CNN layer to generate the final occupancy grid map and flow predictions, effectively combining spatial and temporal information for comprehensive scene understanding.

4.1.5 Model Training. The model outputs position and flow information for each occupied grid in a BEV map of dimensions X and Y . For occupancy prediction, the model outputs $M_{\text{occ}} \in \mathbb{R}^{2 \times X \times Y}$, supervised by a combination of focal and dice losses L_{grid} to address class imbalance from grid sparsity. For flow prediction, we decompose the task into flow orientation $M_{\text{ori}} \in \mathbb{R}^{1 \times X \times Y}$, supervised by sin loss L_{ori} , and direction classification $M_{\text{dir}} \in \mathbb{R}^{3 \times X \times Y}$, supervised by cross-entropy loss L_{dir} . The direction classifier categorizes flow into forward ($[0, \pi]$), backward ($[\pi, 2\pi]$), or unoccupied, resolving the π ambiguity in the sin loss. Additionally, we supervise height estimation by discretizing ground truth heights into C_h bins and applying cross-entropy loss L_{height} on the upsampled height distribution F_h , with ground truth heights collected from LiDAR. The total training loss is:

$$L = \alpha_0 L_{\text{height}} + \alpha_1 L_{\text{grid}} + \alpha_2 L_{\text{ori}} + \alpha_3 L_{\text{dir}}$$

where L_{ori} and L_{dir} are computed only for occupied grids. After prediction, only occupied grid indices and their corresponding flow values are transmitted to the vehicle, minimizing communication overhead.

4.2 Vehicle BEV Projection

Meanwhile, on the vehicle side, we employ a lightweight approach to generate BEV occupancy grid maps with flow information. We process LiDAR point clouds using PointPillars [33], a computationally efficient 3D detection model. The detected 3D bounding boxes are then projected onto BEV space to construct the occupancy grid map. For flow estimation, we utilize the orientation of detected objects as the grid flow for their corresponding occupied grids. However, due to LiDAR's lower sampling rate (10 fps) compared to the event cameras, there can be a non-negligible time discrepancy between the vehicle's map and the infrastructure's map. To address this, we employ a 3D multi-object tracking method [68] with a linear motion model between consecutive LiDAR frames to estimate object positions. Upon receiving high-temporal resolution data from the infrastructure, vehicles align the rectified BEV map with the infrastructure BEV map to update their local maps with the latest information. The infrastructure continuously sends high-frequency data to vehicles, with LiDAR synchronized to event camera timestamps to ensure proper alignment with vehicle data. This eliminates the need for vehicles to wait for LiDAR bounding box detection, allowing immediate acknowledgment of critical events from infrastructure, thus enabling prompt perception responses for autonomous vehicles.

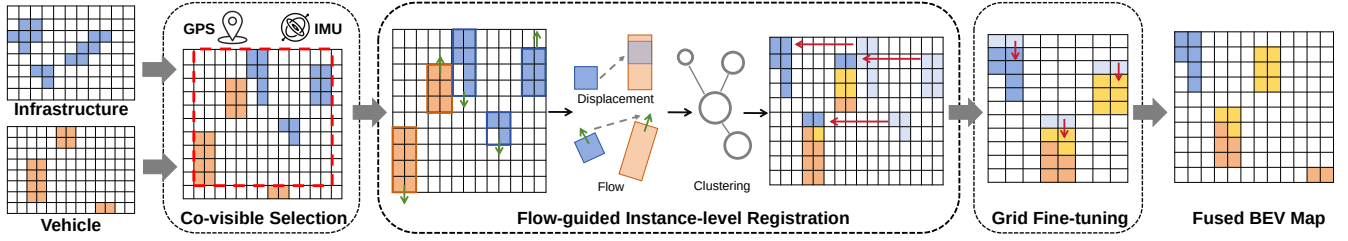


Figure 5: Illustration of flow-guided BEV occupancy grid map alignment.

4.3 Flow-guided BEV Alignment

Once we obtain BEV maps from both the infrastructure and the vehicle, we need to fuse them for collaborative perception. However, this presents significant challenges due to the lack of detailed shape and geometry information in the maps and the large perception gap between the two views. The generated BEV occupancy maps contain only basic information — the positions and motion status of traffic participants — lacking crucial details such as object shape and geometry, as well as static environmental features like buildings, traffic signs, and road lanes. Furthermore, differing occlusion patterns from the varying viewpoints of infrastructure and vehicles cause the same object to appear significantly different across maps. These limitations render conventional fusion approaches ineffective. Feature point-based registration methods [2, 3, 24, 25] and semantic BEV map matching methods [4, 59] rely on semantic feature extraction and consistent object representations, which cannot be reliably achieved with these simplified maps. Meanwhile, learning-based BEV fusion methods [36, 70], while successful for combining calibrated RGB camera views on a single vehicle, require joint training between infrastructure and autonomous vehicles, significantly limiting their practical utility in real-world deployments.

To overcome the challenge of our BEV map lacking static semantic landmarks for registration, our approach leverages two key insights: 1) Aim to maximize the number of aligned instances within the co-visible region to increase the overall intersection-over-union (IOU). 2) Ensure that for the same instance, the grid flow remains consistent across both the transformed infrastructure map and the vehicle map. The algorithm proceeds through three steps: 1) utilizing GPS and IMU data to select the co-visible region; 2) converting grid maps into discrete instances and using a clustering algorithm to estimate the most appropriate transformation parameters; 3) Adjusting the infrastructure map to the vehicle perspective and fine-tuning the alignment to further enhance the IOU. Our method solely uses the positional features and flow characteristics of the binary grid, achieving high accuracy and efficiency in the map alignment process.

Step1: Co-visible Selection. We first verify if there is sufficient overlap between the infrastructure and vehicle maps, as a lack of a co-visible region would render the infrastructure map ineffective when the vehicle is distant. This verification helps avoid unnecessary computational waste on the vehicle. Using GPS and IMU data, we determine the positions of vehicles and infrastructure to derive initial rotation and translation parameters. Points sampled from the infrastructure’s occupied grids are then converted into vehicle coordinates using these parameters, producing the transformed

occupancy grid. Similarly, grid flow is adjusted to match the vehicle perspective using the same transformation parameters. While this transformation process relies on GPS and IMU data, which can introduce significant errors in real-world applications due to their inherent inaccuracies, it nevertheless serves as a valuable starting point for our registration process.

Step2: Flow-guided Instance-level Registration. After determining the co-visible region, we initiate instance-level registration, where each instance corresponds to a cluster of grids that represent a moving object. For the infrastructure’s BEV map, we use the DBSCAN algorithm [21] to group occupied grids based on their positions. The instance flow for each group is determined by averaging the flows of all grids within that group. In the vehicle’s BEV map, we define instances more directly: groups of grids associated with the same bounding box are treated as individual instances, with the box’s orientation serving as the instance flow.

To find the matching instances, we calculate the displacement and flow differences between each instance pair. The position difference between each pair of instances is measured in three ways: 1) the displacement between the center grid of two instances; 2) the maximum displacement that two instances can be overlapped; 3) the minimum displacement that two instances can be overlapped. This design approach is adopted because the same instance may appear different in shape and size when viewed from the infrastructure and vehicle perspectives, primarily due to occlusion. By considering these three different position displacements, our method comprehensively captures various possible matching conditions between instances, ensuring robust matching even under significant appearance variations across different viewpoints. Given n infrastructure instances and m vehicle instances, we compute two matrices: $H_{pos} \in \mathbb{R}^{m \times n \times 3 \times 2}$ containing the displacement along x and y axis between infrastructure-vehicle instance pairs, and $H_{flow} \in \mathbb{R}^{m \times n}$ capturing their motion similarities.

To determine transformation parameters between two BEV maps, we first filter out pairs with large discrepancies in displacement and flow differences. This eliminates unlikely matches and reduces the search space to pairs with reasonable alignment potential. Next, we identify the most frequently occurring displacement and flow differences within the matrices H_{pos} and H_{flow} , respectively. The optimization problem can be formulated as:

$$(T^*, \theta^*) = \arg \max_{T, \theta} \sum_{i,j} \mathbb{1}(\|H_{pos}(i, j) - T\| < \epsilon_{pos}) \cdot \mathbb{1}(\|H_{flow}(i, j) - \theta\| < \epsilon_{flow})$$

where $\cdot$ is the indicator function and T include translation along x and y axis. The intuition behind this approach is that all instances are affected by the same GPS and IMU errors initially. Common patterns in position and flow differences across multiple instance pairs likely represent these systematic errors rather than random noise. By identifying and correcting these common patterns, we can adjust all instances coherently, leading to enhanced overall alignment accuracy. It naturally maximizes the number of well-aligned instance pairs within the co-visible region, effectively optimizing the Intersection over Union (IOU) in this area. Specifically, we apply DBSCAN algorithm to cluster the infrastructure-vehicle pair-wise flow and position differences in the H_{flow} and H_{pos} . Each resulting cluster represents a set of potential transformation parameters. We select the optimal parameters by identifying the cluster with the most instances. When multiple clusters tie for the largest size, we choose the one with the smallest internal variance, as it represents a more consistent and reliable pattern of differences. Using this selected cluster, we compute the final transformation parameters by averaging all differences within it. We then apply these parameters to transform the infrastructure's BEV map data accordingly.

Step3: Grid Fine-tuning. We further refine the transformation parameters through an iterative optimization process that aims to maximize the IOU (Intersection over Union) of occupied grids in the co-visible region. For each parameter, we set a step size. We then generate a new transformed occupancy map, calculate the IOU for this adjustment, and update the parameters based on IOU changes. This iterative process continues until optimal alignment is achieved. This iterative process continues until the IOU no longer shows any improvement, which typically occurs within five iterations. After optimization, the infrastructure occupancy map is successfully transformed to align with the vehicle perspective and is ready for downstream autonomous driving tasks.

5 Testbed and Dataset

Existing infrastructure-assisted autonomous driving datasets [64, 75, 76] do not deploy the event camera on the infrastructure, thus they cannot be used to evaluate our system. Therefore, to assess our system, we evaluate it using two newly collected datasets, one from the real world and the other from the CARLA simulator [11]. Our collected datasets involve multiple sensors on the infrastructure, including LiDAR, RGB camera, and event camera. The summary of the two datasets is demonstrated in Table 2. We will introduce our testbed and the datasets in detail as follows.

5.1 Testbeds

To implement our system, we modified a vehicle and built a portable sensor platform as the lamppost (see Fig. 6). The portable platform is equipped with a Prophesee EVK4 event camera, a LIVOX HAP LiDAR, and a Basler RGB camera. The event camera has a resolution of 720×1280 , and the RGB camera 1200×1920 . We utilize we utilize a wide-angle lens with a horizontal field of view (HFOV) of approximately 160° . The LiDAR features a FOV of $120^\circ \times 25^\circ$ and captures $\sim 45,000$ points per frame. The Lidar data is used to generate ground truth for BEV map estimation. We set a GPS and an IMU to record the location and pose of the sensors. Additionally, the height of the platform can be adjusted to 2-5 meters to adapt to different scenarios. On the vehicle, we installed a RoboSense LiDAR

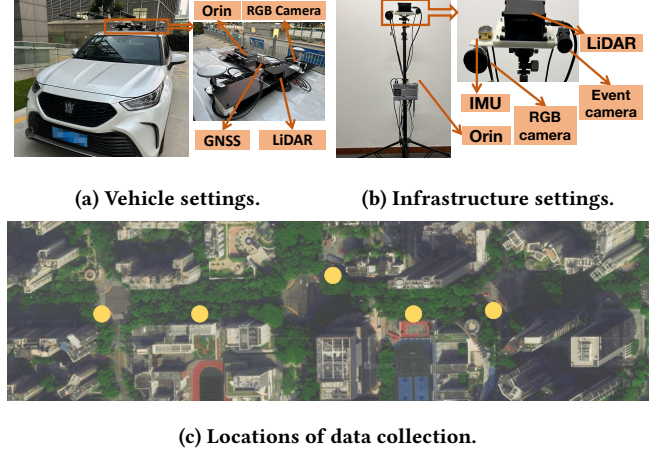


Figure 6: Testbed for the real-world data collection and evaluation.

and a Basler camera, approximately 1.9 meters above ground. The RoboSense LiDAR has a FOV of $120^\circ \times 25^\circ$ with 78,000 points per frame. An RTK-GNSS system records vehicle positions. For managing multiple sensors during data collection, all sensors are integrated with the Robot Operating System (ROS) for control. Besides, NVIDIA Jetson Orin is installed as the computation device on both sides, with an 802.11ac router for communication. Network Time Protocol (NTP) is used to synchronize the time clock of the lamppost and the vehicle.

5.2 Real-world Dataset

Our testbed collects data from city roads, focusing on five scenarios, such as straight roads and intersections. Smaller intersections can be adequately covered with a single event camera, whereas larger intersections with 6–8 traffic lanes typically require two event cameras to achieve full coverage. We gather 27 sequences of infrastructure and vehicle data, each with about 3 ~ 10 co-visible vehicles and pedestrians. We set lamppost under different poses and heights and calibrate the extrinsic parameters between sensors using the Sensors Calibration toolbox [72]. Since Lidar has a smaller FOV than the camera, we use an image mask to filter out the camera pixels and only make sure all the sensors has the same view. To obtain ground truth BEV occupancy map, the LiDAR point cloud and the manually annotated bounding boxes are first transformed into the camera's perspective, and then projected into the bird's-eye view. Due to LiDAR's sparsity, we employ an interpolation method [49] to densify the points. The projected points are divided into grids, and a grid is labeled as occupied only if object points fall within it. The grid flow is annotated with the bounding box orientation. Additionally, to reduce the white noise caused by the event camera, we utilize a time-space window filter. This filter counts the occurrences of events within a certain temporal-spatial window and removes event data if few surrounding events are detected within a specific time period. We manually align the infrastructure view and the vehicle view to obtain an accurate transformation matrix as the ground truth.

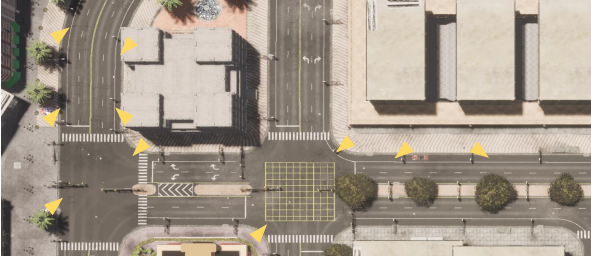


Figure 7: Locations of the infrastructures set in the CARLA. The yellow triangles show the orientations of event cameras.

Dataset	#inf.-veh. pairs	#road sections	#point clouds	#events
Real-world	27	5	5888	784.3M
CARLA	60	10	17964	91.65M

Table 2: Summary of two new datasets.

5.3 CARLA Simulation Dataset

In addition to the real-world dataset, we generated a dataset using the CARLA simulation environment [11]. We set lampposts and ego vehicles on the Town 10 map. Each lamppost has an event camera, RGB camera, and depth camera, all with a resolution of 960×1280 and a FOV of 90° . The depth camera simulates flash LiDAR from prior systems [23, 54, 76] and provides ground truth for the BEV occupancy grid map. Each vehicle is equipped with a LiDAR featuring a 360° FOV. The sensors on the lampposts are positioned in various poses as depicted in Figure 7. A total of 10 lampposts are strategically placed around the map, covering diverse traffic scenarios such as two-way straight roads and intersections. At each position, 5-7 ego vehicles are recorded together with the infrastructure and there are 60 pairs in total. We simulate traffic flow with 120-150 vehicles across the map. To obtain the ground truth BEV occupancy map, we convert the depth images into point cloud frames and extract the bounding box and velocity of each object from CARLA. Then we select points within the bounding box and project them onto the BEV space. The BEV grids containing the projected points are marked as occupied. The grid flow is obtained by assigning the object velocity to the corresponding grids.

In both datasets, we include three different types of moving objects, namely vehicles, bicycles, and pedestrians, for evaluation. For vehicles, we cover a wide range of types, including family cars, buses, ambulances, etc.. Also, we include vehicles with varying speeds for evaluation. The vehicle speeds in the dataset can reach up to 20 m/s.

6 Evaluation

6.1 Experiments Setup

In EventEye, we accumulate event camera data over a 20ms time window for processing. For BEV map estimation, we slice this time window into 5 bins to form event representation. The height prediction for each event pixel is discretized into 90 bins. The infrastructure BEV occupancy grid map is configured as 100×200 cells, with

each cell covering a $0.4m \times 0.4m$ area - a scale consistent with previous planning algorithms [29, 37, 40]. This configuration enables a 40-meter perception range for the event camera. For vehicles, the aligned map dimensions are set to 400×400 . We train both the BEV estimation model and the 3D point cloud detection model [33] on the two datasets using an NVIDIA A6000 GPU. For BEV map alignment, we adopt an efficient DBSCAN implementation [66] that enables real-time system operation.

6.2 Evaluation Metrics

6.2.1 Absolute Trajectory Error, Reaction Distance. To evaluate the BEV map registration performance, we adopt the absolute trajectory error (ATE) from [54] to measure the discrepancy between estimated and ground truth trajectories. The ATE is calculated as the square root mean error between trajectories: $ATE = \sqrt{\frac{1}{n} \sum^n (T_e - T_g)}$, where n denotes the number of frames, T_e and T_g represent the estimated and ground truth trajectories, respectively. To illustrate the system benefit, we introduce reaction distance, defined as the spatial distance between the autonomous vehicle and objects at their first detection point. A higher reaction distance provides vehicles with more space and time to respond to potential hazards, thereby significantly reducing collision risks in real-world scenarios.

6.2.2 Recall, Precision, IOU. We evaluate the BEV occupancy grid map estimation using intersection-over-union (IOU), recall, and precision. Let S_p denote the set of predicted occupied grids and S_g denote the set of ground truth occupied grids. IOU measures the overlap between predicted and ground truth BEV maps, defined as $IOU = \frac{|S_p \cap S_g|}{|S_p \cup S_g|}$. We measure the precision and recall at both grid and instance level. Grid-level recall and precision are defined as $R = \frac{|S_p \cap S_g|}{|S_g|}$ and $P = \frac{|S_p \cap S_g|}{|S_p|}$, respectively. In addition to grid-level metrics, we also evaluate recall and precision at the instance level on the map. For instance-level evaluation, we consider an instance as a true positive if over 50% of its constituent grids are correctly predicted. The instance-level recall and precision are then calculated based on these true positive criteria.

6.2.3 AOS. To measure flow estimation error, we adopt the average orientation similarity (AOS) metric from the KITTI benchmark [18].

The AOS is defined as $AOS = \frac{1}{|S_p|} \sum_{i \in S_p} \frac{1 + \cos \Delta_\theta^{(i)}}{2}$, where Δ_θ represents the angle difference between predicted and ground truth orientation for a specific grid. This metric quantifies orientation similarity on a scale from 0 to 1, with 1 indicating perfect alignment. Higher AOS values indicate better performance. We also extend this evaluation to assess grid flow at the instance level.

6.3 End-to-end Evaluation

To evaluate end-to-end performance in the real world, we deploy the testbed described in Section 5.1 and drive the vehicle through a road segment around 350m long. The real-world road contains a crossing road (II) and three straight road segments (I, III, IV). The BEV occupancy map and flow from the portable sensor platform are transmitted to the vehicle and aligned with the vehicle's BEV map. Figure 9 demonstrates the system's latency and trajectory error throughout this road segment. Figure 9a shows the system's

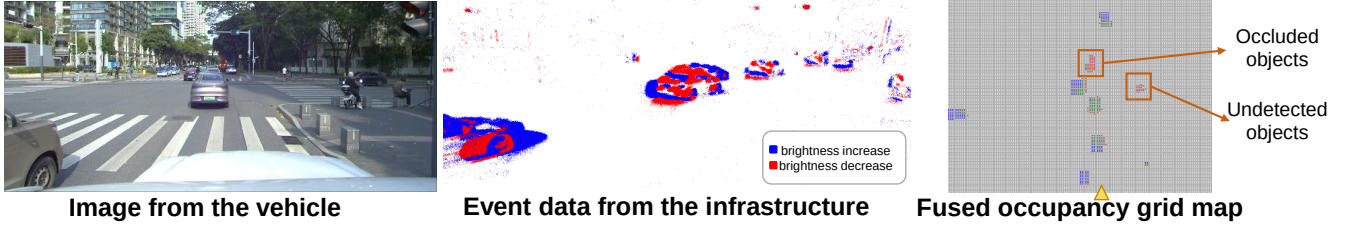


Figure 8: A typical scene in the real world. Blue and red grids indicate areas visible only from the vehicle and infrastructure, respectively. Green grids are areas visible to both. The yellow triangle is the ego vehicle position.

Modality	Methods	CARLA dataset							Real-world dataset						
		Grid-level				Instance-level			Grid-level				Instance-level		
		Recall	Precision	IOU	AOS	Recall	Precision	AOS	Recall	Precision	IOU	AOS	Recall	Precision	AOS
RGB	BEVHeight [74]	0.51	0.66	0.41	0.80	0.53	0.78	0.80	0.53	0.68	0.43	0.79	0.48	0.79	0.70
	BEVHeight [74]	0.63	0.78	0.55	0.89	0.67	0.85	0.90	0.54	0.70	0.44	0.83	0.50	0.80	0.83
	E2DEPTH [26]	0.29	0.41	0.21	-	0.24	0.44	-	0.26	0.18	0.12	-	0.21	0.09	-
Event	EventEye	0.71	0.79	0.60	0.92	0.75	0.87	0.93	0.65	0.74	0.54	0.91	0.64	0.86	0.92

Table 3: Comparison of BEV occupancy grid map and grid flow estimation.

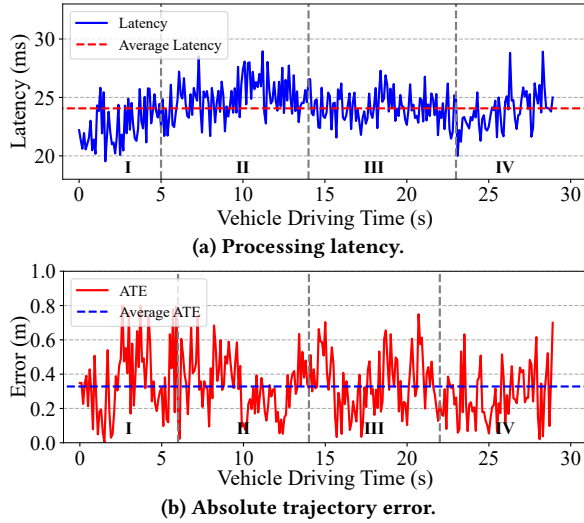


Figure 9: An End-to-end evaluation of EventEye on a road segment.

processing latency, demonstrating that EventEye requires approximately 24 ms on average for end-to-end processing, including data processing on both vehicle and infrastructure, communication, and registration. The latency fluctuations reflect the effectiveness of our sparse processing module: when fewer event points are active, fewer pixels require processing, resulting in reduced latency. Combined with the 20 ms window for event data accumulation, the total end-to-end latency remains under 50 ms, ensuring timely delivery of critical road information for real-time autonomous driving. Figure 9b illustrates EventEye's absolute trajectory error. The system achieves an average ATE of around 37 cm - less than the size of a single grid, which shows that EventEye can support decimeter-level downstream tasks effectively [19]. According to Baidu Apollo's white paper [61], decimeter-level localization accuracy is sufficient for L4 autonomous driving, enabling vehicles to perform capabilities found in state-of-the-art autonomous systems.

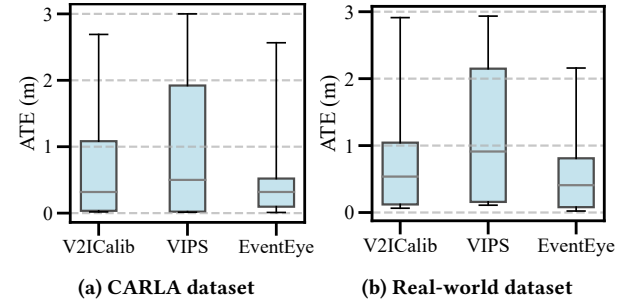


Figure 10: The ATE of EventEye and baseline methods. The gray line indicates the mean value of ATE.

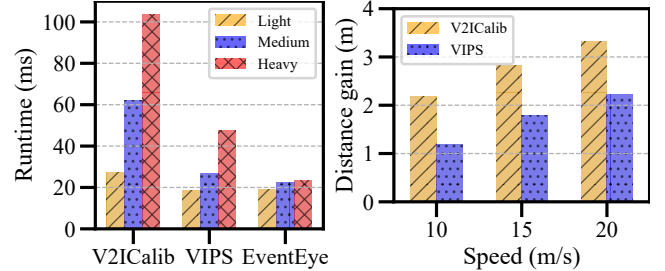


Figure 11: Processing latency Figure 12: Reaction distance under different traffic conditions by EventEye over baseline methods.

6.4 Performance of EventEye

To better evaluate our proposed event camera-based infrastructure system, we implement two conventional LiDAR-based systems, VIPS [54] and V2I-Calib [52], and conduct comprehensive comparisons on both CARLA and the real-world dataset.

6.4.1 Trajectory Error. Figure 10 illustrates the ATE distribution with error bars for the 5th and 95th percentiles across different systems. EventEye achieves ATE of approximately 0.31 m on the CARLA dataset and 0.38 m on the real-world dataset on average.

According to previous works [38, 52], this accuracy falls within acceptable ranges for real-world infrastructure-assisted autonomous driving, validating the practical feasibility of the system’s localization performance. In addition, EventEye demonstrates a more compact error distribution, with fewer significant error samples, achieving a 95th percentile error 20% lower than V2ICalib and 63% lower than VIPS in the real-world dataset. Specifically, EventEye’s errors remain predominantly below 0.8 meters, while V2ICalib and VIPS frequently produce outliers exceeding 1 meter and 1.5 meters, respectively. This distribution indicates that our method produces more stable results compared to the alternative approaches. The performance gap between EventEye and baseline methods stems from several key limitations in the baseline approaches. V2ICalib fails when there are few co-visible objects between the infrastructure and vehicles. VIPS struggles with uneven object distributions between views, where object count imbalances generate multiple possible matching patterns, leading to inaccurate registrations. Additionally, both methods are vulnerable to viewpoint-dependent variations in bounding box sizes caused by occlusions, resulting in object misalignment and erroneous results. Our system overcomes these challenges by using occupancy grids for alignment and absolute measures for clustering transformation parameters, enabling robust results across diverse scenarios.

6.4.2 Latency. Following the criterion in [54], we compare the end-to-end latency under three traffic conditions based on the number of vehicles: light (≤ 4 vehicles), medium ($4 - 8$ vehicles), and heavy (≥ 8 vehicles). As shown in Figure 11, while all three methods demonstrate similar latency under light traffic conditions, V2ICalib and VIPS experience significant performance degradation as traffic complexity increases. This stems from their need to compute complex matching scores between each pair of objects, with computational complexity growing quadratically with the number of objects. In light traffic, the matching patterns are straightforward and easily identified. However, in heavy traffic scenarios - where infrastructure support is most critical due to safety concerns and occlusions - the algorithms require more computational iterations to determine matching patterns, causing registration times to rise substantially, with V2ICalib exceeding 100 ms and VIPS reaching around 50 ms. In contrast, EventEye maintains relatively consistent performance by efficiently calculating displacement and flow differences between grid clusters to identify matching grids. It achieves an average latency of 24.2 ms, showing a 26% reduction compared to VIPS and a 66% reduction compared to V2ICalib. This demonstrates EventEye’s ability to efficiently handle real-time registration tasks across varying traffic conditions.

6.4.3 Reaction Distance. Leveraging the event camera’s microsecond temporal resolution, EventEye promptly detects incoming vehicles and pedestrians, enabling earlier reactions to potential hazards—particularly crucial for objects rushing from blind spots. We evaluate this detection capability using the CARLA simulator, focusing on vehicles approaching the autonomous vehicle or emerging from blind spots. The system’s performance is measured by calculating the distance between the autonomous vehicle and rushing vehicles at their first detection point. Figure 12 illustrates additional reaction distance gained by EventEye compared to V2ICalib and VIPS under different vehicle speeds. With EventEye’s efficient

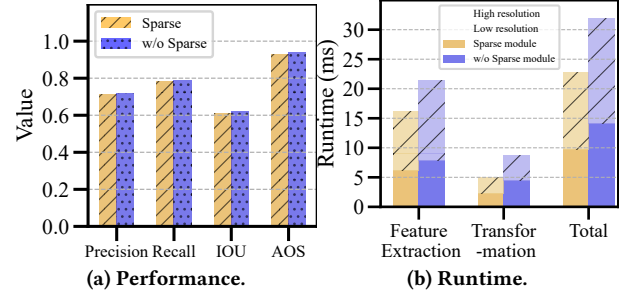


Figure 13: Performance and runtime comparison for our sparse module.

processing and the event camera’s high temporal resolution, our system detects objects 67 ms earlier than VIPS (51% reduction) and 112 ms earlier than V2ICalib (64% reduction) in latency. At high speeds (20 m/s), this temporal advantage translates to critical safety improvements, providing over 2 m and 3 m of additional reaction distance, respectively. The significance of these latency reductions is highlighted by previous research [41] showing that a mere 50 ms decrease in latency can reduce accidents by 47%. These results demonstrate that our system’s substantial latency improvements can provide extensive safety benefits for autonomous vehicles.

6.5 Micro Benchmark

6.5.1 BEV Map and Flow Estimation. We evaluate our method with three baseline methods: 1) E2DEPTH [26] which projects event points through depth estimation; 2) RGB-based BEV estimation method BEVHeight [74] for infrastructure; 3) BEVHeight with the event embedding from [26]. The comparisons are shown in Table 3. From the results, we observe that monocular depth estimation methods generate significant errors in autonomous driving scenarios, particularly with real-world datasets. In the CARLA simulator, we can create dense depth images as ground truth which is hard to satisfy in the real world, making it hard to apply this method effectively in real-world scenarios. Compared to the RGB-based method, our method achieves higher performance in BEV estimation across both datasets. This improvement is attributed to the fact that event pixels primarily concentrate on moving objects, and our design effectively directs the model’s focus to the locations where these events occur. Additionally, the use of an event camera enables more precise flow estimation due to the unique polar features inside the event camera. Such a dynamic feature can allow the model to effectively track and analyze the motion status of the object. Compared to direct BEV estimation from event representation, EventEye demonstrates enhanced recall improvements. Our model’s recurrent structure, which incorporates historical predictions as priors, enables it to identify objects that were visible in the past but have since become less distinct as they move. Moreover, EventEye demonstrates greater improvement in flow estimation on real-world datasets. Our system can effectively handle challenges such as white noise and streetlight interference, enabling it to manage complex traffic scenarios under actual conditions.

6.5.2 Sparse Module. We also evaluate the effect of our proposed sparse module within the BEV estimation framework. Here, we compare the inference runtime under different input image resolutions and the model performance with the model without the

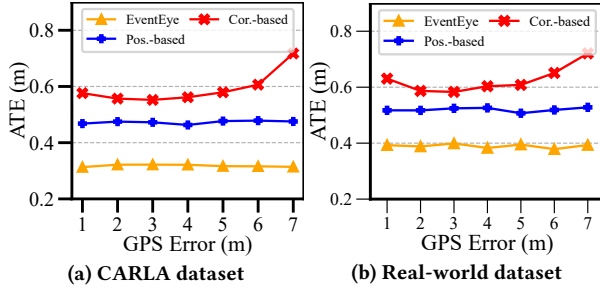


Figure 14: The ATE with respect to GPS on the CARLA and real-world dataset.

Data type	Data Size (KB)	Transmission time (ms)
Raw point cloud	1125.13	54.4
Raw event data	349.06	30.5
BEV map & flow (Ours)	5.31	4.2

Table 4: The average size of shared data and the transmission time measured on an 802.11ac network.

sparse module. Figure 13 presents a runtime and performance comparison to evaluate the impact of using the sparse module. The results indicate that the sparse module reduces the time required for feature extraction and transformation by an average of 21% and 43%, respectively. As a result, it could reduce the overall runtime by around 30%. Additionally, we observe that during inference, our sparse module reduces the data size of feature extraction input by 67% and BEV transformation input by 93% on average. Concurrently, the performance of the model experiences a minimal decline of only 1-2%. This demonstrates that our sparse module can significantly lessen the computational overhead while maintaining robust performance.

6.5.3 BEV Occupancy Map Registration. We evaluate the performance of the BEV occupancy map registration on both datasets. This module operates on the vehicle to fuse the BEV occupancy grid maps from both the infrastructure and the vehicle. Our registration module is compared with two alternative approaches: 1) position-based registration, which does not use flow features in our registration method and serves as an ablation baseline, and 2) correlation-based registration, which performs registration using feature points extracted from the BEV map. Figure 14 demonstrates registration error under varying GPS error conditions. The results show that EventEye consistently outperforms both alternative methods. The correlation-based method performs poorly due to the BEV map’s lack of detailed geometric features. In addition, the accuracy of this method depends on its search space size, with larger spaces providing robustness to greater GPS errors; however, expanding the search space significantly increases computational overhead. In addition, compared to position-based registration, our methods improves the performance by around 40%. The flow features provide essential information for registration and help better align the grids between the infrastructure and the vehicle. By leveraging both positional and flow features from BEV grids, EventEye maintains robust registration accuracy even under significant GPS errors.

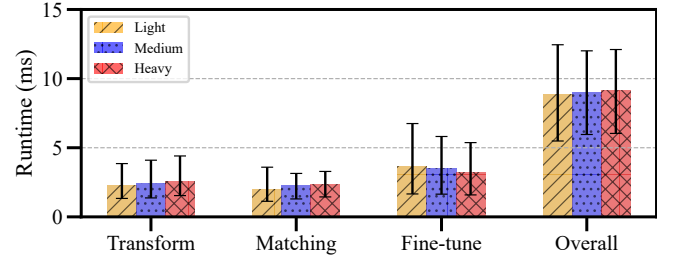


Figure 15: Registration runtime of each step in different traffic conditions.

6.6 System Overhead

In this section, we measure EventEye’s system overhead. First, we measure the runtime of flow-guided BEV map registration on the vehicle across different traffic conditions as described in section 6.4.2. Figure 15 shows runtime with error bars for the 5th and 95th percentiles. The average registration time is approximately 9 ms, with a maximum of 13 ms, meeting real-time autonomous driving requirements. The consistent performance across varying traffic conditions demonstrates how our grid-based approach, which relies solely on grid displacement and flow, effectively reduces computational complexity. Notably, fine-tuning requires more time under light traffic conditions. This occurs because matching fewer objects produces less reliable initial results, a limitation also observed in previous methods [52, 54]. Our approach addresses this challenge through additional iterations to ensure robust registration.

We also analyze the communication overhead of EventEye by comparing data size and transmission time with ways of sharing raw point clouds and the raw event data directly. These experiments are conducted over an 802.11ac Wi-Fi network, and the results are presented in Table 4. EventEye sends the BEV occupancy map and flows, with a total size of approximately 5.31KB, to the autonomous vehicle in just 4.2 ms. Compared to transmitting raw point clouds, our system achieves a 211.8× reduction in data size and a 12.9× reduction in transmission time. When compared to transmitting raw event data, the reductions are 65.7× for data size and 7.3× for transmission time.

6.7 System Robustness

6.7.1 Perception Distance. We evaluate the BEV occupancy grid and flow estimation performance at different perception distances under different resolution settings using the CARLA simulator. We train models for resolutions 480×640 (low) and 960×1280 (high), representing typical event camera resolutions. Grid-level IOU and AOS at each 10-meter interval are shown in Figure 16. Low-resolution cameras perform moderately within 20 meters but decline more rapidly with distance than high-resolution cameras. Low-resolution camera captures most events of nearby objects, while high resolution captures detail at greater distances. Increasing resolution enhances distant object perception. For low-resolution cameras, IOU remains above 0.5 at 30-40 meters, exceeding the threshold for correct object detection [18, 33, 73], showing that EventEye maintains reliable detection performance at long distances.

	Impact Factor	Recall	Precision	IOU	AOS
Road Type	Straight road	0.75	0.81	0.64	0.95
	Intersection	0.68	0.77	0.57	0.92
Lighting	Daytime	0.79	0.83	0.68	0.91
	Night	0.77	0.82	0.66	0.89
New Section	Event	0.61	0.75	0.51	0.85
	RGB	0.35	0.57	0.28	0.59

Table 5: Performance of BEV occupancy map estimation in different road sections and different illumination conditions.

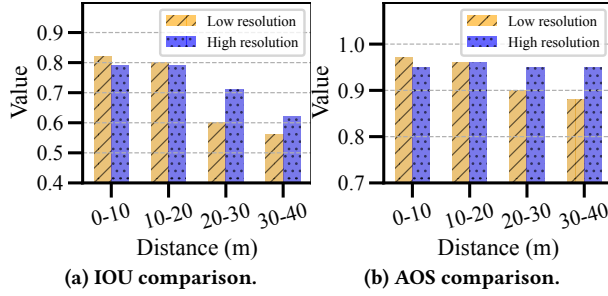


Figure 16: Performance of EventEye at different perception distance.

6.7.2 Road Types. In our experiments, we evaluate EventEye’s performance on different road types, like straight roads and intersections, as shown in Table 5. EventEye performs better on straight roads, where driving patterns are more consistent, while performance at intersections is slightly lower due to complexity. However, it is still comparable to other baselines in Table 3. This shows that our system adapts well to various traffic conditions, making it suitable for diverse city road environments.

6.7.3 Lighting Condition. Another significant advantage of the event camera is its high dynamic range, which enables it to operate effectively under diverse lighting conditions. We evaluate our system’s performance with daytime and mixed data (day and night). As shown in Table 5, nighttime data performance is only slightly reduced (1-2%) compared to daytime data. This slight decrease is due to interference from streetlights and building lights. Our sparse module effectively filters out this noise and minimizes the impact of such interfering light sources, maintaining robust performance across different lighting conditions.

6.7.4 New Road Section. We compare Event-based and RGB-based systems in a new environment. Specifically, we train models on data from various road sections and evaluate them on entirely different road sections with significant domain gaps, including different road types, camera poses, and mounting positions. As shown in Table 5, our method significantly outperforms the RGB model, which depends on specific environmental conditions. The event camera captures brightness changes, a low-level feature that generalizes well, allowing the event-based model to remain robust and adaptable in diverse traffic scenarios.

7 Discussion

Event Camera Limitations. Although event cameras are gaining popularity in the research community, their technology is not yet as mature as other sensing modalities. Compared to traditional

RGB cameras, current event cameras have several limitations. First, they lack semantic details, which limits their ability to provide comprehensive scene understanding. Integrating event cameras with complementary sensors, such as RGB cameras that provide rich semantic information, is therefore a promising direction for future research. Second, event cameras currently have limited resolution, which, as discussed in Section 6.7.1, reduces their effective perception range and constrains practical deployment. Although high-resolution event cameras (e.g., 720p) have recently emerged, they remain expensive. In addition, event cameras are susceptible to noise under adverse weather conditions such as rain and snow [8], which can degrade performance. Recent works [10, 30] have proposed effective event-noise reduction techniques, and incorporating such approaches into our feature extraction pipeline is a promising direction for improving robustness in challenging environments. Nevertheless, event camera technology has advanced rapidly in recent years, including integration into smartphone chips [50]. With growing demand and mass production, costs are expected to decrease while sensor quality continues to improve.

Broader Applications. Although EventEye utilizes LiDAR on the vehicle, our approach is applicable to other vehicle sensors as well. Data from common autonomous driving sensors, such as RGB cameras and radars, can be readily converted into BEV occupancy maps and flow, which can then be fused with the BEV map generated from infrastructure-mounted event cameras. Consequently, event camera-ready infrastructure can seamlessly interoperate with vehicles equipped with diverse sensing suites, enabling broad compatibility across different sensor configurations and platforms. Beyond autonomous driving, EventEye also has potential in a range of related domains, such as smart infrastructure management for traffic monitoring, congestion detection, and incident awareness to support adaptive traffic control, as well as cooperative robot navigation and autonomous delivery systems.

8 Conclusion

In this paper, we introduce EventEye, the first system to utilize an event camera on infrastructure to deliver enhanced timely information to vehicles in real time, which enables vehicles to quickly recognize road conditions and react promptly. We implement and evaluate EventEye on both simulated and real-world datasets. The experimental results demonstrate that EventEye outperforms state-of-the-art solutions in both accuracy and efficiency while providing substantial safety margins for complex road scenarios.

Acknowledgments

This work is partially supported by Research Grants Council of Hong Kong under grants SRFS2526-4S02, GRF 14214022 and GRF 14213525.

References

- [1] Manideep Reddy Aliminati, Bharatesh Chakravarthi, Aayush Atul Verma, Arpitsinh Vaghela, Hua Wei, Xuesong Zhou, and Yezhou Yang. 2024. SEVD: Synthetic Event-based Vision Dataset for Ego and Fixed Traffic Perception. *arXiv preprint arXiv:2404.10540* (2024).
- [2] P.J. Besl and Neil D. McKay. 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 2 (1992), 239–256. doi:10.1109/34.121791

- [3] P. Biber and W. Strasser. 2003. The normal distributions transform: a new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)* (Cat. No.03CH37453), Vol. 3. 2743–2748 vol.3. doi:10.1109/IROS.2003.1249285
- [4] A. Birk and S. Carpin. 2006. Merging Occupancy Grid Maps From Multiple Robots. *Proc. IEEE* 94, 7 (2006), 1384–1397. doi:10.1109/JPROC.2006.876965
- [5] Nicholas Carr. 2014. *The glass cage: Automation and us*. WW Norton & Company.
- [6] Guang Chen, Hu Cao, Jorg Conradt, Huajin Tang, Florian Rohrbein, and Alois Knoll. 2020. Event-Based Neuromorphic Vision for Autonomous Driving: A Paradigm Shift for Bio-Inspired Visual Sensing and Perception. *IEEE Signal Processing Magazine* 37, 4 (2020), 34–49. doi:10.1109/MSP.2020.2985815
- [7] Jiahe Cui, Shuyao Shi, Yuze He, Jianwei Niu, Guoliang Xing, and Zhenchao Ouyang. 2024. VILAM: Infrastructure-assisted 3D Visual Localization and Mapping for Autonomous Driving. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. USENIX Association, Santa Clara, CA, 1831–1845. <https://www.usenix.org/conference/nsdi24/presentation/cui>
- [8] Ethan Delaney, Tim Brophy, Enda Ward, Fiachra Collins, Edward Jones, Brian Deegan, and Martin Glavin. 2025. Evaluating Event-Based Vision Sensing in Rain and Fog. *IEEE Sensors Journal* 25, 16 (2025), 31545–31562. doi:10.1109/JSEN.2025.3584460
- [9] Vikrant Dewangan, Basant Sharma, Tushar Choudhary, Sarthak Sharma, Aakash Aanegola, Arun K Singh, and K Madhava Krishna. 2023. UAP-BEV: Uncertainty Aware Planning Using Bird's Eye View Generated From Surround Monocular Images. In *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*. IEEE, 1–8.
- [10] Saizhe Ding, Jinze Chen, Yang Wang, Yu Kang, Weiguo Song, Jie Cheng, and Yang Cao. 2023. E-MLB: Multilevel benchmark for event-based camera denoising. *IEEE Transactions on Multimedia* 26 (2023), 65–76.
- [11] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An open urban driving simulator. In *Conference on robot learning*. PMLR, 1–16.
- [12] Jingjing Fan, Lili Fan, Qinghua Ni, Junhao Wang, Yi Liu, Ren Li, Yutong Wang, and Sanjin Wang. 2024. Perception and Planning of Intelligent Vehicles Based on BEV in Extreme Off-road Scenarios. *IEEE Transactions on Intelligent Vehicles* (2024).
- [13] Justas Furmonas, John Liobe, and Vaidotas Barzdenas. 2022. Analytical review of event-based camera depth estimation methods and systems. *Sensors* 22, 3 (2022), 1201.
- [14] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. 2020. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence* 44, 1 (2020), 154–180.
- [15] Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. 2018. 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*. 3867–3876.
- [16] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. 2021. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters* 6, 3 (2021), 4947–4954.
- [17] Mathias Gehrig, Mario Millhäusler, Daniel Gehrig, and Davide Scaramuzza. 2021. E-raft: Dense optical flow from event cameras. In *2021 International Conference on 3D Vision (3DV)*. IEEE, 197–206.
- [18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 3354–3361. doi:10.1109/CVPR.2012.6248074
- [19] Roshan George, Joseph Clancy, Tim Brophy, Ganesh Sistu, William O'Grady, Sunil Chandra, Fiachra Collins, Darragh Mullins, Edward Jones, Brian Deegan, and Martin Glavin. 2025. Infrastructure Assisted Autonomous Driving: Research, Challenges, and Opportunities. *IEEE Open Journal of Vehicular Technology* 6 (2025), 662–716. doi:10.1109/OJVT.2025.3542213
- [20] Germain Haessig, Xavier Berthelon, Sio-Hoi Ieng, and Ryad Benosman. 2019. A spiking neural network model of depth from defocus for event-based neuromorphic vision. *Scientific reports* 9, 1 (2019), 3744.
- [21] Michael Hahsler, Matthew Piekenbrock, and Derek Doran. 2019. dbscan: Fast density-based clustering with R. *Journal of Statistical Software* 91 (2019), 1–30.
- [22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
- [23] Yuze He, Chen Bian, Jingfei Xia, Shuyao Shi, Zhenyu Yan, Qun Song, and Guoliang Xing. 2023. VI-Map: Infrastructure-Assisted Real-Time HD Mapping for Autonomous Driving. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking (Madrid, Spain) (ACM MobiCom '23)*. Association for Computing Machinery, New York, NY, USA, Article 66, 15 pages. doi:10.1145/3570361.3613280
- [24] Yuze He, Li Ma, Jiahe Cui, Zhenyu Yan, Guoliang Xing, Sen Wang, Qintao Hu, and Chen Pan. 2022. AutoMatch: Leveraging Traffic Camera to Improve Perception and Localization of Autonomous Vehicles. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*. 16–30.
- [25] Yuze He, Li Ma, Zhehao Jiang, Yi Tang, and Guoliang Xing. 2021. VI-eye: Semantic-based 3D point cloud registration for infrastructure-assisted autonomous driving. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. 573–586.
- [26] Javier Hidalgo-Carrió, Daniel Gehrig, and Davide Scaramuzza. 2020. Learning monocular dense depth from events. In *2020 International Conference on 3D Vision (3DV)*. IEEE, 534–542.
- [27] Mai Hirata, Manabu Tsukada, Keisuke Okumura, Yasumasa Tamura, Hideya Ochiai, and Xavier Défago. 2021. Roadside-Assisted Cooperative Planning using Future Path Sharing for Autonomous Driving. In *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*. 1–7. doi:10.1109/VTC2021-Fall52928.2021.9625324
- [28] Yue Hu, Juntong Peng, Sifei Liu, Junhao Ge, Si Liu, and Siheng Chen. 2024. Communication-Efficient Collaborative Perception via Information Filling with Codebook. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15481–15490.
- [29] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhao Wang, et al. 2023. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17853–17862.
- [30] Haiyan Jiang, Xiaoshuang Wang, Wei Tang, Qinghui Song, Qingjun Song, and Wenchao Hao. 2024. Event Stream Denoising Method Based on Spatio-Temporal Density and Time Sequence Analysis. *Sensors* 24, 20 (2024), 6527.
- [31] Suzanna Jiwan, Xiao Li, Sertac Karaman, and Daniela Rus. 2023. Risk-Aware Neural Navigation From BEV Input for Interactive Driving. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5659–5665.
- [32] Kaleem Nawaz Khan, Ali Khalid, Yash Turkar, Karthik Dantu, and Fawad Ahmad. 2024. VRF: Vehicle Road-side Point Cloud Fusion. In *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services (Minato-ku, Tokyo, Japan) (MOBISYS '24)*. Association for Computing Machinery, New York, NY, USA, 547–560. doi:10.1145/3643832.3661874
- [33] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. 2019. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12697–12705.
- [34] Siqi Li, Zhikuan Zhou, Zhou Xue, Yipeng Li, Shaoyi Du, and Yue Gao. 2024. 3D Feature Tracking via Event Camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18974–18983.
- [35] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. 2023. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 1477–1485.
- [36] Zhiqi Li, Wenhao Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. 2022. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European conference on computer vision*. Springer, 1–18.
- [37] Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahao Li, Jan Kautz, Tong Lu, and Jose M Alvarez. 2024. Is ego status all you need for open-loop end-to-end autonomous driving?. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14864–14873.
- [38] Ciyun Lin, Yuying Wang, Bowen Gong, Hui Liu, and Hongchao Liu. 2025. Roadside-Onboard Point Cloud Registration for Vehicle-Infrastructure Cooperation Perception in Traffic Collision Zones. *IEEE Sensors Journal* 25, 15 (2025), 29533–29544. doi:10.1109/JSEN.2025.3581277
- [39] Genjia Liu, Yue Hu, Chenxin Xu, Weibo Mao, Junhao Ge, Zhengxiang Huang, Yifan Lu, Yinda Xu, Junkai Xia, Yafei Wang, et al. 2024. Towards Collaborative Autonomous Driving: Simulation Platform and End-to-End System. *arXiv preprint arXiv:2404.09496* (2024).
- [40] Haochen Liu, Zhiyu Huang, and Chen Lv. 2023. Occupancy prediction-guided neural planner for autonomous driving. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 4859–4865.
- [41] Haolan Liu, Zixuan Wang, and Jishen Zhao. 2023. COLA: characterizing and optimizing the tail latency for safe level-4 autonomous vehicle systems. *arXiv preprint arXiv:2305.07147* (2023).
- [42] Yifan Lu, Yue Hu, Yiqi Zhong, Dequan Wang, Siheng Chen, and Yanfeng Wang. 2024. An extensible framework for open heterogeneous collaborative perception. *arXiv preprint arXiv:2401.13964* (2024).
- [43] Su Ma, Hong Zhang, and Xiaomin Xing. 2018. Scalability for smart infrastructure system in smart grid: a survey. *Wireless Personal Communications* 99 (2018), 161–184.
- [44] Rashid Mehmood, Simon See Iyad Katib, and Imrich Chlamtac. 2020. *Smart infrastructure and applications*. Springer.
- [45] Nico Messikommer, Carter Fang, Mathias Gehrig, and Davide Scaramuzza. 2023. Data-driven feature tracking for event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5642–5651.
- [46] Emmanuel Ndashimye, Sayan K Ray, Nurul I Sarkar, and Jairo A Gutiérrez. 2017. Vehicle-to-infrastructure communication over multi-tier heterogeneous networks: A survey. *Computer networks* 112 (2017), 144–166.

- [47] Liyuan Pan, Cedric Scheerlinck, Xin Yu, Richard Hartley, Miaomiao Liu, and Yuchao Dai. 2019. Bringing a blurry frame alive at high frame-rate with an event camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6820–6829.
- [48] Jonah Philion and Sanja Fidler. 2020. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer, 194–210.
- [49] Cristiano Premebida, Luis Garrote, Alireza Asvadi, A Pedro Ribeiro, and Urbano Nunes. 2016. High-resolution lidar-based depth mapping using bilateral filter. In *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)*. IEEE, 2469–2474.
- [50] Prophesee. 2024. Prophesee's Metavision Image Deblur Solution for Smartphones is Now Production-Ready. <https://www.prophesee.ai/2024/02/27/metavision-image-deblur-production-ready/>. Accessed: 2026-03-25.
- [51] Prophesee. 2026. Metavision AMD Kria Starter Kit. <https://www.prophesee.ai/event-based-metavision-amd-kria-starter-kit/>. Accessed: 2026-03-25.
- [52] Qianxin Qu, Yijin Xiong, Xinyu Zhang, Chen Xia, Qian Peng, Ziqiang Song, Kang Liu, Xin Wu, and Jun Li. 2024. V2I-Calib+: A Multi-terminal Spatial Calibration Approach in Urban Intersections for Collaborative Perception. *arXiv preprint arXiv:2410.11008* (2024).
- [53] Nitin J Sanket, Chethan M Parameshwara, Chahat Deep Singh, Ashwin V Kurutukulam, Cornelia Fermüller, Davide Scaramuzza, and Yiannis Aloimonos. 2020. Evdodgenet: Deep dynamic obstacle dodging with event cameras. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 10651–10657.
- [54] Shuyao Shi, Jiahe Cui, Zhehao Jiang, Zhenyu Yan, Guoliang Xing, Jianwei Niu, and Zhenchao Ouyang. 2022. VIPS: Real-time perception fusion for infrastructure-assisted autonomous driving. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*. 133–146.
- [55] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems* 28 (2015).
- [56] Shintaro Shiba, Yoshimitsu Aoki, and Guillermo Gallego. 2022. Secrets of event-based optical flow. In *European Conference on Computer Vision*. Springer, 628–645.
- [57] Jack Stilgoe. 2020. *Who Killed Elaine Herzberg?* Springer International Publishing, Cham, 1–6. doi:10.1007/978-3-030-32320-2_1
- [58] Zhaoning Sun, Nico Messikommer, Daniel Gehrig, and Davide Scaramuzza. 2022. Ess: Learning event-based semantic segmentation from still images. In *European Conference on Computer Vision*. Springer, 341–357.
- [59] Sooraj Sunil, Saeed Mozaffari, Rajmeet Singh, Behnam Shahrrava, and Shahpour Alirezaee. 2023. Feature-Based Occupancy Map-Merging for Collaborative SLAM. *Sensors* 23, 6 (2023). doi:10.3390/s23063114
- [60] Wang Tong, Azhar Hussain, Wang Xi Bo, and Sabita Maharjan. 2019. Artificial intelligence for vehicle-to-everything: A survey. *IEEE Access* 7 (2019), 10823–10843.
- [61] Tsinghua University and Baidu. 2022. Tsinghua & Baidu Released AIR White Paper 2.0: Major Breakthrough in Vehicle-Infrastructure Collaboration - AIR. <https://air.tsinghua.edu.cn/en/info/1007/1685.htm> Accessed: 2025-09-28.
- [62] Aayush Atul Verma, Bharatesh Chakravarthi, Arpitsinh Vaghela, Hua Wei, and Yezhou Yang. 2024. eTraM: Event-based Traffic Monitoring Dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 22637–22646.
- [63] Celyn Walters and Simon Hadfield. 2021. Evreflex: Dense time-to-impact prediction for event-based obstacle avoidance. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 1304–1309.
- [64] Hai Wang, Xin Yuan, Yingfeng Cai, Long Chen, and Yicheng Li. 2022. V2I-CARLA: A Novel Dataset and a Method for Vehicle Reidentification-Based V2I Environment. *IEEE Transactions on Instrumentation and Measurement* 71 (2022), 1–9. doi:10.1109/TIM.2022.3158998
- [65] Tsun-Hsuan Wang, Alexander Amini, Wilko Schwarting, Igor Gilitschenski, Sertac Karaman, and Daniela Rus. 2022. Learning Interactive Driving Policies via Data-driven Simulation. In *2022 International Conference on Robotics and Automation (ICRA)*. IEEE.
- [66] Yiqiu Wang, Yan Gu, and Julian Shun. 2020. Theoretically-Efficient and Practical Parallel DBSCAN. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (Portland, OR, USA) (SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA, 2555–2571. doi:10.1145/3318464.3380582
- [67] Junqing Wei, Jarrod M Snider, Tianyu Gu, John M Dolan, and Bakhtiar Litkouhi. 2014. A behavioral planning framework for autonomous driving. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 458–464.
- [68] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 2020. Ab3dmot: A baseline for 3d multi-object tracking and new evaluation metrics. *arXiv preprint arXiv:2008.08063* (2020).
- [69] Ruihao Xia, Chaoqiang Zhao, Meng Zheng, Ziyang Wu, Qiyu Sun, and Yang Tang. 2023. Cmda: Cross-modality domain adaptation for nighttime semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 21572–21581.
- [70] Enze Xie, Zhiding Yu, Daquan Zhou, Jonah Philion, Anima Anandkumar, Sanja Fidler, Ping Luo, and Jose M Alvarez. 2022. M² BEV: Multi-Camera Joint 3D Detection and Segmentation with Unified Birds-Eye View Representation. *arXiv preprint arXiv:2204.05088* (2022).
- [71] Fang Xu, Lei Yu, Bishan Wang, Wen Yang, Gui-Song Xia, Xu Jia, Zhendong Qiao, and Jianzhuang Liu. 2021. Motion deblurring with real events. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2583–2592.
- [72] Guohang Yan, Zhuochun Liu, Chengjie Wang, Chunlei Shi, Pengjin Wei, Xinyu Cai, Tao Ma, Zhizheng Liu, Zebin Zhong, Yuqian Liu, Ming Zhao, Zheng Ma, and Yikang Li. 2022. OpenCalib: A Multi-sensor Calibration Toolbox for Autonomous Driving. *arXiv preprint arXiv:2205.14087* (2022).
- [73] Yan Yan, Yuxing Mao, and Bo Li. 2018. Second: Sparsely embedded convolutional detection. *Sensors* 18, 10 (2018), 3337.
- [74] Lei Yang, Kaicheng Yu, Tao Tang, Jun Li, Kun Yuan, Li Wang, Xinyu Zhang, and Peng Chen. 2023. BEVHeight: A Robust Framework for Vision-based Roadside 3D Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 21611–21620.
- [75] Xiaoqing Ye, Mao Shu, Hanyu Li, Yifeng Shi, Yingying Li, Guangjie Wang, Xiao Tan, and Errui Ding. 2022. Rope3D: The Roadside Perception Dataset for Autonomous Driving and Monocular 3D Object Detection Task. *arXiv preprint arXiv:2203.13608* (2022).
- [76] Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, et al. 2022. Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 21361–21370.
- [77] Alex Zihao Zhu, Dinesh Thakur, Tolga Özaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. 2018. The multivehicle stereo event camera dataset: An event camera dataset for 3D perception. *IEEE Robotics and Automation Letters* 3, 3 (2018), 2032–2039.
- [78] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. 2019. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 989–997.
- [79] Junyu Zhu, Lina Liu, Bofeng Jiang, Feng Wen, Hongbo Zhang, Wanlong Li, and Yong Liu. 2023. Self-supervised Event-based Monocular Depth Estimation using Cross-modal Consistency. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 7704–7710.
- [80] Walter Zimmer, Gerhard Arya Wardana, Suren Sritharan, Xingcheng Zhou, Rui Song, and Alois C Knoll. 2024. Tumtraf v2x cooperative perception dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 22668–22677.