Low Latency Pedestrian Detection Based on Dynamic Vision Sensor and RGB Camera Fusion

Bingyu Huang[a], Gianni Allebosch[b], Peter Veelaert[c], Tim Willems[d], Wilfried Philips[e] and Jan Aelterman[f]

TELIN-IPI, Ghent University, Sint-Pietersnieuwstraat, Ghent, Belgium

{*bingyu.huang, gianni.allebosch, peter.veelaert, tim.willems, wilfried.philips, jan.aelterman*}*@ugent.be*

Keywords: Dynamic Vision Sensor, Motion Segmentation, Sensor Fusion, Autonomous Driving.

Abstract: Advanced driver assistance systems currently adopt RGB cameras as visual perception sensors, which rely primarily on static features and are limited in capturing dynamic changes due to fixed frame rates and motion blur. A very promising sensor alternative is the dynamic vision sensor(DVS) with microsecond temporal resolution that records an asynchronous stream of per-pixel brightness changes, also known as event stream. However, in autonomous driving scenarios, it's challenging to distinguish between events caused by the vehicle's motion and events caused by actual moving objects in the environment. To address this, we design a motion segmentation algorithm based on epipolar geometry and apply it to DVS data, effectively removing static background events and focusing solely on dynamic objects. Furthermore, we propose a system that fuses the dynamic information captured by event cameras and rich appearance details from RGB cameras. Experiments show that our proposed method can effectively improve detection performance while showing great potential in decision latency.

1 INTRODUCTION

Autonomous driving systems rely on robust and reliable perception mechanisms to ensure safety and efficiency in dynamic environments. Motion segmentation, the process of identifying and isolating moving objects within a scene, is a critical component of visual perception (Kulchandani and Dangarwala, 2015). Segmenting a moving foreground against a static background is a relatively easy task for event cameras thanks to its dynamic response characteristic and edge-like output (Schraml et al., 2010). Moving objects in autonomous driving scenarios is more challenging due to the more disturbing changing background. To counteract the interference of changing background pixels, approaches such as optical flow analysis (Kim and Kwon, 2015), motion clustering (Kim et al., 2010), and neural networks (Mane and Mangale, 2018) are introduced to characterize the difference between foreground and background. These methods all develop high-dimensional feature representations for backgrounds or targets and are robust to general scenarios. The disadvantage is that concentrated features rely on prior dataset training and reduce accuracy in the detection under occlusion or under/over exposure. Another limitation of traditional cameras is their inherent frame-based nature, which can lead to latency issues (Narasimhan and Nayar, 2003; Zhu et al., 2020; Chang et al., 2021), particularly in scenarios requiring rapid decision-making.

In contrast, Dynamic Vision Sensors (DVS), or event cameras, capturing per-pixel brightness changes asynchronously with microsecond temporal resolution (Brandli et al., 2014; Taverni et al., 2018; Gallego et al., 2020). This technology inherently provides a high temporal resolution that can significantly reduce latency and enhance responsiveness in time-critical scenarios. Event cameras are particularly advantageous in situations with low lighting conditions or rapid motion.

However, in autonomous driving scenarios, it is challenging to distinguish between events caused by the vehicle's ego-motion and events caused by actual moving objects in the environment, such as pedestri-

[a] https://orcid.org/0000-0003-0258-2684
[b] https://orcid.org/0000-0003-2502-3746
[c] https://orcid.org/0000-0003-4746-9087
[d] https://orcid.org/0000-0002-5264-919X
[e] https://orcid.org/0000-0003-4456-4353
[f] https://orcid.org/0000-0002-5543-2631

Huang, B., Allebosch, G., Veelaert, P., Willems, T., Philips, W. and Aelterman, J.
Low Latency Pedestrian Detection Based on Dynamic Vision Sensor and RGB Camera Fusion.
DOI: 10.5220/0013126600003912

In *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2025) - Volume 2: VISAPP*, pages 841-850
ISBN: 978-989-758-728-3; ISSN: 2184-4321

ans, cyclists, or other vehicles. This difficulty arises because the vehicle's motion induces a large number of events from static objects, like buildings, road signs, and trees, making it harder to identify and isolate meaningful events generated by truly dynamic elements in the scene. Properly filtering out these motion-induced events while retaining the relevant ones is crucial for ensuring reliable and accurate perception in fast-changing driving environments.

To address these challenges, we propose an epipolar geometry-based method to remove events triggered by static background objects. We further make use of the data from DVS to complement dynamic information for RGB detections and achieve a faster detection response. In summary, the contributions of this paper are:

- We introduce a novel motion segmentation algorithm for DVS employing the epipolar geometry principle.
- We propose a fusion method that makes use of motion information from DVS and appearance characteristics from RGB to accomplish low-latency detection.

2 RELATED WORK

2.1 Motion Segmentation on DVS

Event-based vision, particularly utilizing Dynamic Vision Sensors, has emerged as a transformative approach in computer vision, offering high temporal resolution and low latency, which are crucial for time-critical scenarios, such as autonomous driving, robotics, and industrial automation. Research in this domain can be broadly divided into two main methodologies: ego-motion compensation and neural network-based approaches. Ego-motion compensation focuses on deriving motion trajectories, optical flow, and other geometric properties directly from the event streams. By estimating the ego-motion, it's possible to predict the expected change in brightness at each pixel due to the vehicle's motion and subtract this from the DVS data (Stoffregen et al., 2019; Zhou et al., 2021; Parameshwara et al., 2020; Mishra et al., 2017). In contrast, neural network-based approaches harness the capabilities of deep learning to process complex, sparse, and asynchronous data from DVS. These techniques enable the extraction and segmentation of motion directly from raw event streams using various types of neural networks, including spiking neural networks (Parameshwara et al., 2021), graph neural networks (Mitrokhin et al., 2020; Alkendi et al., 2024), and recurrent neural networks (Zhang et al., 2023).

2.2 Fusion of RGB and DVS for Object Detection

Fusion of RGB images with Dynamic Vision Sensor (DVS) data has become a prominent approach for enhancing object detection, particularly in challenging environments such as low-light conditions (Liu et al., 2023) or scenes with rapid motion (Gehrig and Scaramuzza, 2024). Since event data and RGB data are fundamentally different in nature, most of the methods adopt deep learning models to learn features from multi-modal data. A common approach involves using separate convolutional neural networks (CNNs) to extract features and combine them in deeper layers (Zhou et al., 2023; Tomy et al., 2022). Some researchers introduce novel architectures to improve performance, such as the attention mechanism (Cao et al., 2021; Cho and Yoon, 2022), temporal and recurrent networks (Wan et al., 2023; Hamaguchi et al., 2023), and spiking neural network(SNN) (Cao et al., 2024).

3 METHODOLOGY

Our method is inspired by a geometric-based technique (Allebosch et al., 2023) designed to distinguish static backgrounds from true motion, using a linear array of RGB cameras on a moving vehicle. We generalize the technique to asynchronous streams from DVS, enabling lower-latency detection. In the following sections, we outline the fundamental principles of DVS, discuss the motion segmentation challenges we aim to address and introduce our proposed solution.

3.1 Background Removal on DVS

3.1.1 Working Principle of DVS

DVS operates by asynchronously detecting changes in brightness at individual pixels. Let $I(t,\mathbf{x})$ denote the intensity of light at time t, where $\mathbf{x} = (x,y)$ is the pixel location. DVS responds to changes in the logarithm of the intensity,

$$L(t,\mathbf{x}) = \log I(t,\mathbf{x}). \quad (1)$$

Each pixel sensor continuously monitors the change in $L(t,\mathbf{x})$ over time. An event is triggered when the change in logarithmic intensity exceeds a

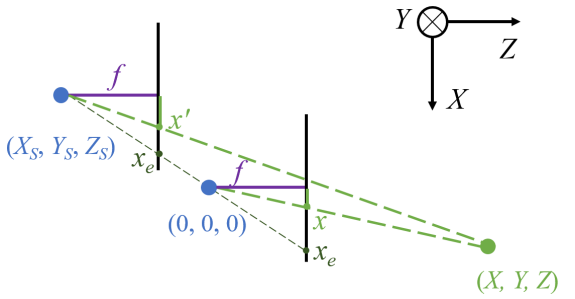


Figure 1: Camera configuration of our method. A pair of cameras observes a static object located at (X,Y,Z). The reference camera at current timestamp and the side camera at past timestamp are at positions (0,0,0), (X_S,Y_S,Z_S). f is the focal length.

predefined threshold C. Specifically, an event is generated at pixel $\mathbf{x}$ and time t_k if

$$|L(t_k,\mathbf{x}) - L(t_k - \Delta t_k,\mathbf{x})| \geq C. \tag{2}$$

When an event is triggered at pixel location $\mathbf{x}_k = (x_k,y_k)$, it is represented as

$$e_k = (\mathbf{x}_k,t_k,p_k), \tag{3}$$

where t_k is the timestamp of the event, and p_k is the polarity of the event, indicating whether the brightness increased or decreased,

$$p_k = \begin{cases} +1 & \text{if } L(t_k,\mathbf{x}_k) - L(t_k-\Delta t_k,\mathbf{x}_k) \geq C \\ -1 & \text{if } L(t_k,\mathbf{x}_k) - L(t_k-\Delta t_k,\mathbf{x}_k) \leq -C. \end{cases} \tag{4}$$

Hence a static DVS is inherently sensitive to motion in the scene. It ignores static objects or backgrounds because they do not cause any change in brightness. This property is particularly useful in applications where detecting motion is the primary goal, such as surveillance (Bolten et al., 2019), object tracking (Borer et al., 2017), or motion analysis (Xu et al., 2020), as it reduces the amount of data that needs to be processed by focusing only on areas with motion.

When a DVS is mounted on a moving vehicle, the situation becomes more complex. The ego-motion of the vehicle will cause the entire field of view to shift, and this will generate a large number of events across the sensor. A key challenge in this scenario is to differentiate between events caused by the motion of the vehicle (global motion) and those caused by independent motion within the scene (e.g., a pedestrian crossing the street).

3.1.2 Epipolar Geometry Principle

We will now show how it is possible to identify DVS events that are triggered by static background through epipolar geometry principles. Given the assumption that the vehicle is driving in a straight line at a constant speed during the period, with no vertical displacement of the vehicle, the overview of the geometric calculation model is shown in Figure 1.

The projection (x,y) on image plane of reference camera and side camera are denoted as follows,

$$(x,y) = \frac{f}{Z}(X,Y), \tag{5}$$

$$(x',y') = \frac{f}{Z-Z_S}(X-X_S,Y-Y_S). \tag{6}$$

For a static object locates at (X,Y,Z), the disparity between projection x on the reference camera and the projection x' on the side camera is

$$(\Delta_{static,x},\Delta_{static,y}) = \frac{f}{Z-Z_S}(X_S - X\frac{Z_S}{Z}, Y_S - Y\frac{Z_S}{Z}). \tag{7}$$

Except for depth Z, the variables in disparity vector $(\Delta_{static,x},\Delta_{static,y})$ can be obtained through camera configuration. X_S and Y_S are the horizontal and vertical setup distances between two cameras. Note that $Y_S = 0$ because the cameras are set on the same level. Z_S is the driving distance. We assume the camera pair is mounted on a moving vehicle driving at speed v, and the driving distance is

$$Z_S = v\Delta t, \tag{8}$$

where Δt is the time interval between the reference camera at the current timestamp and the side camera at the past timestamp. We note that the driving speed doesn't have to be constant, only the vehicle displacement needs to be known.

Object depth can be obtained by several potential approaches, such as inferring object depth from DVS data by neural network model (Hidalgo-Carrió et al., 2020), or speculating from stereo vision (Ghosh and Gallego, 2022). In this paper, we use separate depth sensors, which are fast and suitable for real-time processing. It is shown in (Allebosch et al., 2023) that even sparse depth information can already provide tight uncertainty bounds for disparity analysis. We also refer to this work for an in-depth description of the relation between depth accuracy and disparity.

When we review expression 7, substituting Z_s for $v\Delta t$ (eq 8) , the only value that is still unknown for the right-hand side of the equation is Δt. Therefore, we can determine the specific time interval for which the disparity along the x-direction is 0. Let $\Delta_{static,x} = 0$ and we get the desired time interval,

$$\Delta t = -\frac{fX_S}{xv}. \tag{9}$$

The projections of two cameras that satisfy the above condition are located at the intersection of the common line of sight and image plane. We denote these special locations as epipoles x_e in Figure 1.

The epipolar geometry principle reveals that a static object observed by current reference camera is also observed by the side camera at Δt before. If not, it means the visual information is triggered by moving objects. We can therefore distinguish moving pedestrians with static backgrounds in the event stream.

3.1.3 Event Frame Representation

In this paper, we employ the event frame (Perot et al., 2020; Gehrig et al., 2019) as event representation instead of a single event. Theoretically, when we accept an event from the reference camera, we can search for matching point in the side camera by the epipolar geometry principle and judge if it is triggered by a moving object. The challenge is that with single events, slight deviations in time and location can cause significant errors in the computation of matching epipoles, leading to mismatches in feature correspondence, since DVS is highly sensitive to small changes in lighting, noise, and other environmental factors. Another disturbance is that even if there is no change in light intensity, there will be event output due to thermal noise and junction leakage current (Feng et al., 2020).

Another consideration is real-time performance. In real-time applications like autonomous driving, drones, or augmented reality, processing individual events can lead to significant computational overhead and delay. By using event frames, we can balance the need for temporal resolution and real-time performance because event frames can be processed more efficiently than streams of single events.

As shown in Figure 2a, we visually compare different temporal widths that can be used to create the event frame representation. We zoom in on the left bottom area (left leg of the pedestrian) and compare the distribution of the events in each temporal window. When the window size is 1 ms, the event distributions on the image plane are not consistent in two consecutive temporal windows. Since epipolar geometry relies on matching points across views, inconsistency of triggered events and transient noise could mislead the estimation of corresponding points between cameras. When we expand the window width to 2ms, the event distribution on the image plane is more stable and better for epipolar geometry analysis. Figure 2b shows more visualizations of different temporal window sizes. As the window width extends, the contour of the pedestrian is more clear. Smaller windows provide faster response times but may capture fewer details, while larger windows enhance feature clarity by accumulating more events, though at the cost of slower responses. On the other hand, large wide window sizes may cause overcrowding of events and overlap between edges. In the 12ms temporal window visualization, the pedestrian's ankle begins to blur due to its faster motion compared to other body areas, causing a higher event density in that region. The temporal width is a hyperparameter that can be tuned according to the driving speed. In this paper, we set the temporal window size as 8ms.

To build an event frame, we aggregate the events at each pixel in the time interval centered around t, with a window width Δt_w. The set of events on the image plane in the time window is $E = \{e_k\}_{k=1}^{N_e}$, N_e is the number of events in the time window. We define the event frame as a 2D grid that stores the accumulation of the events at each pixel $\mathbf{x}$(with coordinates (x,y)) in the time window. Formally, the event accumulation at each pixel in the event frame can be expressed as

$$F(\mathbf{x},t) = \sum_{\left\{k|\mathbf{x}_k=\mathbf{x},t_k\in\left[t-\frac{\Delta t_w}{2},t+\frac{\Delta t_w}{2}\right]\right\}} p_k. \quad (10)$$

3.1.4 Background Removal

Once the event frame is constructed, we apply epipolar geometry to match corresponding points between different camera views. For each pixel location $\mathbf{x}=(x,y)$ where there is event accumulation $F_{ref}(t,\mathbf{x})$ in the current event frame of the reference camera, we retrieve the corresponding matching point $F_{side}(t+\Delta t,\mathbf{x})$ from the side camera at time $t+\Delta t$. The difference $D(t,\mathbf{x})$ is

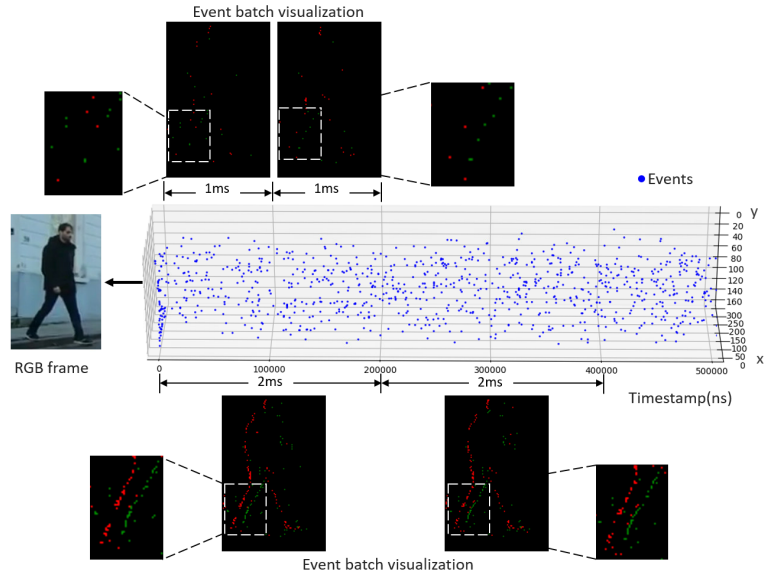
$$D(t,\mathbf{x}) = F_{ref}(t,\mathbf{x}) - F_{side}(t+\Delta t,\mathbf{x}). \quad (11)$$

To classify the event as either being caused by the background or by a moving object, we use a threshold τ to determine whether the event is from the background or a moving object. In this paper, we define pixels where preserved events accumulate as *active pixels*, the indicator function is

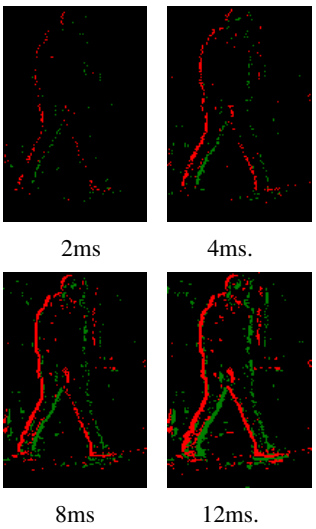
$$\mathcal{A}(t,\mathbf{x}) = \begin{cases} 1, & \text{if } |D(t,\mathbf{x})| \geq \tau \\ 0, & \text{if } |D(t,\mathbf{x})| < \tau. \end{cases} \quad (12)$$

The difference threshold τ can be tuned according to the event density. In this paper, we set $\tau = 1$.

Figure 3 shows the background removal result of examples with event frames. The part with limited disparity is removed, and the moving pedestrians are preserved. Due to camera jitter and measurement errors, there is still a small amount of residual background. In the next section, we will introduce an intuitive and effective fusion method to further make use of the amount of filtered versus non-filtered events.



(a) Visualization of events in temporal window.



(B) comparison of different temporal window size.

Figure 2: Visualization of event data in different temporal widths. The green and red pixels represent positive and negative polarity respectively.

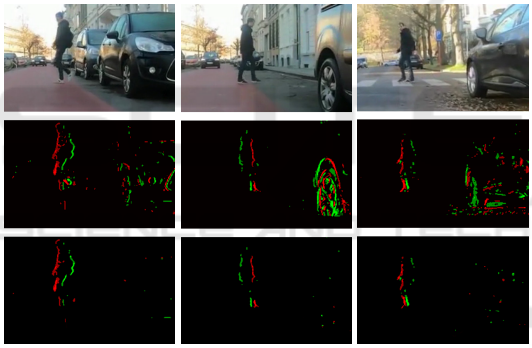


Figure 3: The first row is the image frame from the RGB camera; the second row is the corresponding event frame. The bottom row is the result of background removal by our proposed method.

3.2 Detection Fusion of RGB Frames and DVS Data

In the previous section, we explain how we removed static background on event frames and kept dynamic information, which is complementary to extracting information from RGB cameras. Detectors based on RGB image frames rely on features like texture and visible contrast structure, which are inherently static. Hence, an object looks roughly the same whether it is moving or not. Common object detection methods are divided into two major categories: traditional methods that combine feature extraction with classifiers (Viola and Jones, 2001; Dalal and Triggs, 2005; Felzenszwalb et al., 2009), and deep learning methods that use neural network models (Girshick et al., 2014; Redmon, 2016; Lin, 2017; Zhao et al., 2024). In this paper, we consider an example of a neural network model as the RGB detector, which generates bounding boxes with confidence scores as detection output. We define a bounding box B as a rectangular area bounded by a set of pixel coordinates, with an associated scalar confidence score C_b.

A common approach for selecting target bounding boxes is to set a confidence threshold and pick up boxes that meet or exceed this threshold. As shown in Figure 4, if a pedestrian appears suddenly behind cars, single confidence level based judgment would lose early detections due to low confidence scores in these scenarios. In this case, the background removal result from the DVS can provide complementary information for the bounding boxes, distinguishing early-appearing pedestrians from misdetections in low-confidence bounding boxes C_b.

Specifically, we define the *active pixel percentage* for each bounding box B as

$$P_{\mathrm{a}}(B) = \frac{\sum_{\mathbf{x}\in B} \mathcal{A}(t,\mathbf{x})}{\sum_{\mathbf{x}\in B} \mathcal{F}(t,\mathbf{x})}, \tag{13}$$

where $\mathcal{F}(t,\mathbf{x})$ is an indicator function that judges if events accumulate at the pixel $\mathbf{x}$,

$$\mathcal{F}(t,\mathbf{x}) = \begin{cases} 1, & \text{if } |F(t,\mathbf{x})| > 0 \\ 0, & \text{if } |F(t,\mathbf{x})| = 0. \end{cases} \tag{14}$$

By calculating the active pixel percentage of a bounding box, we can further determine whether the

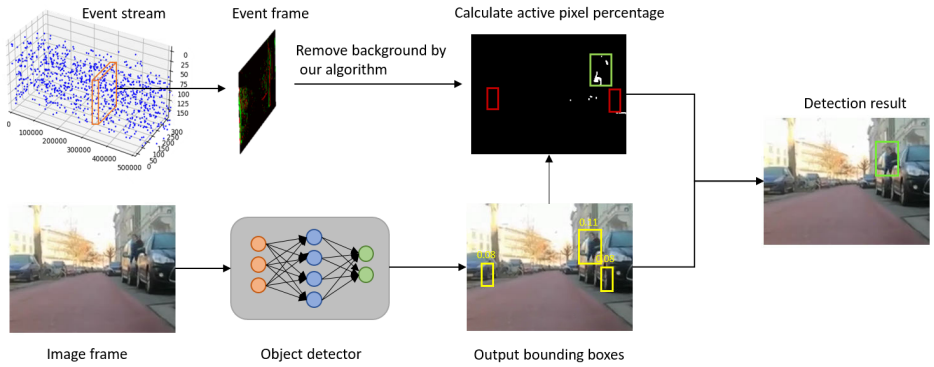


Figure 4: Illustration of proposed fusion method. Our algorithm designed for event stream to remove static background is introduced in Section 3.1.

bounding box comes from a static or dynamic object. We proposed a straightforward and efficient fusion strategy. We set a low confidence threshold $T_{c,l}$ and a high confidence threshold $T_{c,h}$ for the confidence score C_b. There are three possibilities for RGB detection results:

1) $C_b \geq T_{c,h}$: This signifies that RGB detectors assign a conclusive confidence score, suggesting a high probability of a person being present. We retain this candidate in the final set of detections, with minimal risk of false positives.

2) $C_b < T_{c,l}$: This indicates a very low detection score. We assume a negligible likelihood of a person being present and exclude the candidate from the final detections.

3) $T_{c,l} \leq C_b < T_{c,h}$: These bounding boxes have a medium detection score, signaling no clear preference to either keep or discard it as a detection. Therefore, this range benefits the most from our epipolar geometry based detection on event frame. If the active pixel percentage in the bounding box $P_a(B)$ is higher than a threshold T_a, it's more likely that there is a crossing person and we keep the candidate.

By making full use of candidate bounding boxes, our proposed fusion method is able to boost the early detection performance for moving pedestrian detection. In the next section, we evaluate both detection performance and latency on a custom dataset.

3.3 Experimental Evaluation

When a pedestrian first appears from behind an occlusion, only a small portion of their body is visible, making it difficult for detectors to recognize them. RGB detectors based on neural networks often assign a low confidence score in such situations. We set up an experiment to demonstrate that our fusion method can differentiate between bounding boxes coming from early-emerging pedestrians and those caused by background misdetections. Hence, our method can detect pedestrians earlier and provide more reaction time for the drivers to avoid collisions.

3.3.1 Custom Dataset

We designed a platform with three Go Pro Hero 7 cameras mounted on an electric cargo bicycle. The side cameras are set up beside the center camera perpendicular to the driving direction with an equal distance. In addition, radar is also equipped to record driving speed and obtain object depth. The vehicle was driving on a straight road in the city center. Two experimenters crossed the road in front of the vehicle individually or together, and the obtained videos were captured into 21 sequences. The recorded RGB sequences are converted into event data using DVS simulator ESIM (Rebecq et al., 2018) with default simulation settings.

To measure the reaction time (latency) between the first visible instance and the first detection, we manually initialize the first visible annotation of pedestrians, followed by semi-automatic interval tracking using SSD (Liu et al., 2016)and DSST (Danelljan et al., 2016), which would be manually corrected when necessary. Our evaluation comprises 18,982 annotation boxes and 33,123 frames. We choose Yolo v4 (Wang et al., 2021) as the detector for RGB cameras and use the original structure and pre-trained weights. In this paper we focus only on detecting objects in the 'person' class.

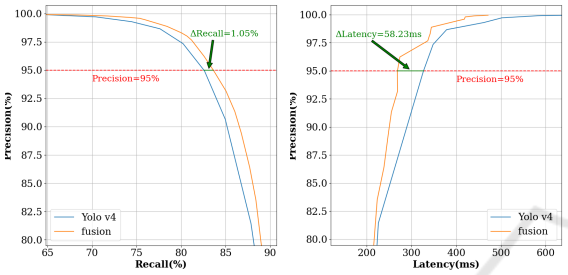
In section 3.2, we introduce the fusion method, which uses three parameters: $T_{c,h}$, $T_{c,l}$ and T_a. We define a range of values for each parameter and evaluate all possible combinations: $T_{c,h} \in \{0.5, 0.7, 0.8, 0.9\}$, $T_{c,l} \in \{0.1, 0.2, 0.3, 0.5\}$, $T_a \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. For each combination, we evaluate the performance by three metrics, precision, recall, and latency. We select the best points from the Pareto front (Ngatchou et al.,

2005), optimized using the F1 score. The Precision-Recall curve and Precision-Latency curve to compare the detection performance and latency are shown in Figure 5.

3.3.2 Result

Table 1: Performance of Yolov4 detector and fusion method.

Method	$F1_{max}$	MAP	average latency(ms)
Yolo v4	88.35	90.49	375.65
Fusion	89.28	91.59	328.09



(a) Precision-Recall curve. (b) Precision-Latency curve.

Figure 5: Comparative analysis of the Yolo v4 detector and our proposed fusion method.

Latency. We define the detection latency as the reaction time between the first annotation and the first detection. Figure 6 shows how the fusion method reduces latency by leveraging motion information from event data. By analyzing the active pixel percentage in bounding boxes with low confidence scores, we can distinguish the early appearance of moving pedestrians from false detections. The Precision-Latency curve Figure 5b demonstrates that the fusion method maintains high precision while reducing detection latency by an average of 76.72 milliseconds. Figure 7 shows several examples of gain in reaction time and distance by our proposed fusion method.

Detection Performance. The Precision-Recall curve Figure 5a illustrates that the fusion method achieves an average increase of 2.05% in recall at equivalent precision levels. It means that, for the same number of false positive detections, our fusion method can detect actual pedestrians more often. Setting the precision threshold at 95%, the fusion method shows a 1.05% higher recall and an average latency gain of 58.23 ms. These results suggest that the fusion method is more efficient, providing accurate detections with reduced response times. Table 1 shows that our proposed fusion method improves MAP by 1.10% while the response is 47.56 ms faster than Yolo v4 detector on average.

3.4 Discussion

The results demonstrate that our proposed fusion approach of using DVS data, filtered by epipolar geometry, combined with RGB-based detections significantly improves pedestrian detection in autonomous driving scenarios. By eliminating static background events, our method enables more accurate and focused detection of dynamic objects, such as pedestrians, even in challenging environments with complex backgrounds.

One key advantage of our approach lies in the use of epipolar geometry to filter out irrelevant events generated by static objects, allowing the system to focus on truly dynamic elements in the scene. The fusion with RGB data further complements the system by leveraging the rich spatial and appearance information from the RGB frames, ensuring that both static and dynamic visual cues are effectively utilized.

Compared to previous methods that rely solely on RGB or DVS data, our approach offers significant improvements in terms of detection latency and accuracy. The ability to detect pedestrians earlier, even when partially occluded or moving quickly, makes our method particularly suitable for real-time applications in autonomous driving. Nevertheless, real-world testing and further refinement are needed to fully validate the system's robustness across different driving conditions.

4 CONCLUSION AND FUTURE WORK

In this paper, we presented a novel fusion algorithm that integrates DVS data with RGB camera detections for improved pedestrian detection in autonomous driving scenarios. By applying epipolar geometry to remove static background events from the DVS data, we significantly enabled the system to focus on dynamic objects. The proposed fusion of the high temporal resolution of DVS with the rich spatial detail of the RGB cameras brings faster and more accurate detection, particularly in low-light and fast-moving environments.

Our method demonstrates strong potential for reducing detection latency and improving performance in real-time perception systems, especially in safety-critical applications such as autonomous vehicles. By combining the complementary strengths of both sensor modalities, our system addresses the limitations faced by traditional camera-based detection methods.

Looking forward, several areas require further exploration and refinement. First, we plan to extend our

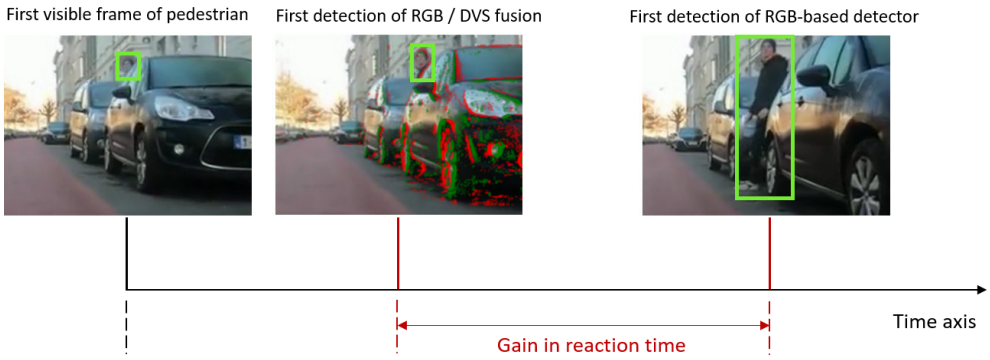


Figure 6: Detection latency of proposed fusion method and Yolo v4. Traditional RGB-based detector tend to give a low confidence score on suddenly appearing pedestrian at the early moment until the pedestrian is sufficiently visible. The fusion method is able to detect a pedestrian ealier while Yolo takes much time to recognize an occluded pedestrian. In the shown scenario, the vehicle is driving at the speed of 2.80 m/s and our proposed fusion method detects the pedestrian 258.59 ms earlier than Yolov4, referring to a gain in distance of 0.72m for driver to avoid the collision.

Figure 7: Detection result of low-latency fusion. The first column is the first detection of the fusion method and the last column is the first detection of YOLO v4 detector. The second and third columns are the raw event frame and the background removal result by our proposed method. The confidence score is shown above bboxes. The confidence score threshold for Yolo v4 to accept a detection is 0.8.

research by testing the system on real-world event data, addressing potential challenges such as sensor noise and environmental variability. Our future work will also focus on optimizing the algorithm for edge processing devices, enabling it to be deployed in resource-constrained environments such as embedded systems in autonomous vehicles.

By continuing to refine and validate the proposed method in real-world conditions, we aim to develop a robust, efficient, and reliable solution for pedestrian detection and other object detection tasks in dynamic environments. In future work, we will study more diverse and complex traffic scenarios.

REFERENCES

Alkendi, Y., Azzam, R., Javed, S., Seneviratne, L., and Zweiri, Y. (2024). Neuromorphic vision-based motion

segmentation with graph transformer neural network. *arXiv preprint arXiv:2404.10940*.

Allebosch, G., Van Hamme, D., Veelaert, P., and Philips, W. (2023). Efficient detection of crossing pedestrians from a moving vehicle with an array of cameras. *Optical Engineering*, 62(3):031210–031210.

Bolten, T., Pohle-Fröhlich, R., and Tönnies, K. D. (2019). Application of hierarchical clustering for object tracking with a dynamic vision sensor. In *Computational Science–ICCS 2019: 19th International Conference, Faro, Portugal, June 12–14, 2019, Proceedings, Part V 19*, pages 164–176. Springer.

Borer, D., Delbruck, T., and Rösgen, T. (2017). Three-dimensional particle tracking velocimetry using dynamic vision sensors. *Experiments in Fluids*, 58:1–7.

Brandli, C., Berner, R., Yang, M., Liu, S.-C., and Delbruck, T. (2014). A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341.

Cao, H., Chen, G., Xia, J., Zhuang, G., and Knoll, A. (2021). Fusion-based feature attention gate component for vehicle detection based on event camera. *IEEE Sensors Journal*, 21(21):24540–24548.

Cao, J., Zheng, X., Lyu, Y., Wang, J., Xu, R., and Wang, L. (2024). Chasing day and night: Towards robust and efficient all-day object detection guided by an event camera. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9026–9032. IEEE.

Chang, M., Feng, H., Xu, Z., and Li, Q. (2021). Low-light image restoration with short-and long-exposure raw pairs. *IEEE Transactions on Multimedia*, 24:702–714.

Cho, H. and Yoon, K.-J. (2022). Event-image fusion stereo using cross-modality feature propagation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 454–462.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee.

Danelljan, M., Häger, G., Khan, F. S., and Felsberg, M. (2016). Discriminative scale space tracking. *IEEE transactions on pattern analysis and machine intelligence*, 39(8):1561–1575.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2009). Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645.

Feng, Y., Lv, H., Liu, H., Zhang, Y., Xiao, Y., and Han, C. (2020). Event density based denoising method for dynamic vision sensor. *Applied Sciences*, 10(6):2024.

Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A. J., Conradt, J., Daniilidis, K., et al. (2020). Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180.

Gehrig, D., Loquercio, A., Derpanis, K. G., and Scaramuzza, D. (2019). End-to-end learning of representations for asynchronous event-based data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5633–5643.

Gehrig, D. and Scaramuzza, D. (2024). Low-latency automotive vision with event cameras. *Nature*, 629(8014):1034–1040.

Ghosh, S. and Gallego, G. (2022). Multi-event-camera depth estimation and outlier rejection by refocused events fusion. *Advanced Intelligent Systems*, 4(12):2200221.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

Hamaguchi, R., Furukawa, Y., Onishi, M., and Sakurada, K. (2023). Hierarchical neural memory network for low latency event processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22867–22876.

Hidalgo-Carrió, J., Gehrig, D., and Scaramuzza, D. (2020). Learning monocular dense depth from events. In *2020 International Conference on 3D Vision (3DV)*, pages 534–542. IEEE.

Kim, D.-S. and Kwon, J. (2015). Moving object detection on a vehicle mounted back-up camera. *Sensors*, 16(1):23.

Kim, J., Ye, G., and Kim, D. (2010). Moving object detection under free-moving camera. In *2010 IEEE International Conference on Image Processing*, pages 4669–4672. IEEE.

Kulchandani, J. S. and Dangarwala, K. J. (2015). Moving object detection: Review of recent research trends. In *2015 International conference on pervasive computing (ICPC)*, pages 1–5. IEEE.

Lin, T. (2017). Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer.

Liu, Z., Yang, N., Wang, Y., Li, Y., Zhao, X., and Wang, F.-Y. (2023). Enhancing traffic object detection in variable illumination with rgb-event fusion. *arXiv preprint arXiv:2311.00436*.

Mane, S. and Mangale, S. (2018). Moving object detection and tracking using convolutional neural networks. In *2018 second international conference on intelligent computing and control systems (ICICCS)*, pages 1809–1813. IEEE.

Mishra, A., Ghosh, R., Principe, J. C., Thakor, N. V., and Kukreja, S. L. (2017). A saccade based framework for real-time motion segmentation using event based vision sensors. *Frontiers in neuroscience*, 11:83.

Mitrokhin, A., Hua, Z., Fermuller, C., and Aloimonos, Y. (2020). Learning visual motion segmentation using event surfaces. In *Proceedings of the IEEE/CVF Con-*

ference on Computer Vision and Pattern Recognition, pages 14414–14423.

Narasimhan, S. G. and Nayar, S. K. (2003). Contrast restoration of weather degraded images. *IEEE transactions on pattern analysis and machine intelligence*, 25(6):713–724.

Ngatchou, P., Zarei, A., and El-Sharkawi, A. (2005). Pareto multi objective optimization. In *Proceedings of the 13th international conference on, intelligent systems application to power systems*, pages 84–91. IEEE.

Parameshwara, C. M., Li, S., Fermüller, C., Sanket, N. J., Evanusa, M. S., and Aloimonos, Y. (2021). Spikems: Deep spiking neural network for motion segmentation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3414–3420. IEEE.

Parameshwara, C. M., Sanket, N. J., Gupta, A., Fermuller, C., and Aloimonos, Y. (2020). Moms with events: Multi-object motion segmentation with monocular event cameras. *arXiv preprint arXiv:2006.06158*, 2(3):5.

Perot, E., De Tournemire, P., Nitti, D., Masci, J., and Sironi, A. (2020). Learning to detect objects with a 1 megapixel event camera. *Advances in Neural Information Processing Systems*, 33:16639–16652.

Rebecq, H., Gehrig, D., and Scaramuzza, D. (2018). Esim: an open event camera simulator. In *Conference on robot learning*, pages 969–982. PMLR.

Redmon, J. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Schraml, S., Belbachir, A. N., Milosevic, N., and Schön, P. (2010). Dynamic stereo vision system for real-time tracking. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 1409–1412. IEEE.

Stoffregen, T., Gallego, G., Drummond, T., Kleeman, L., and Scaramuzza, D. (2019). Event-based motion segmentation by motion compensation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7244–7253.

Taverni, G., Moeys, D. P., Li, C., Cavaco, C., Motsnyi, V., Bello, D. S. S., and Delbruck, T. (2018). Front and back illuminated dynamic and active pixel vision sensors comparison. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 65(5):677–681.

Tomy, A., Paigwar, A., Mann, K. S., Renzaglia, A., and Laugier, C. (2022). Fusing event-based and rgb camera for robust object detection in adverse conditions. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 933–939. IEEE.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. Ieee.

Wan, Z., Mao, Y., Zhang, J., and Dai, Y. (2023). Rpeflow: Multimodal fusion of rgb-pointcloud-event for joint optical flow and scene flow estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10030–10040.

Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2021). Scaled-YOLOv4: Scaling cross stage partial network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13029–13038.

Xu, L., Xu, W., Golyanik, V., Habermann, M., Fang, L., and Theobalt, C. (2020). Eventcap: Monocular 3d capture of high-speed human motions using an event camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, S., Sun, L., and Wang, K. (2023). A multi-scale recurrent framework for motion segmentation with event camera. *IEEE Access*.

Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., and Chen, J. (2024). Detrs beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16965–16974.

Zhou, Y., Gallego, G., Lu, X., Liu, S., and Shen, S. (2021). Event-based motion segmentation with spatio-temporal graph cuts. *IEEE transactions on neural networks and learning systems*, 34(8):4868–4880.

Zhou, Z., Wu, Z., Boutteau, R., Yang, F., Demonceaux, C., and Ginhac, D. (2023). Rgb-event fusion for moving object detection in autonomous driving. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7808–7815. IEEE.

Zhu, Z., Wei, H., Hu, G., Li, Y., Qi, G., and Mazur, N. (2020). A novel fast single image dehazing algorithm based on artificial multiexposure image fusion. *IEEE Transactions on Instrumentation and Measurement*, 70:1–23.