Event-frame object detection under dynamic background condition

Lu, Wenhao; Li, Zehao; Li, Junying; Lu, Yuncheng; Kim, Tony Tae-Hyoung

2024

Lu, W., Li, Z., Li, J., Lu, Y. & Kim, T. T. (2024). Event-frame object detection under dynamic background condition. Journal of Electronic Imaging, 33(4), 043028-. https://dx.doi.org/10.1117/1.JEI.33.4.043028

https://hdl.handle.net/10356/180590

https://doi.org/10.1117/1.JEI.33.4.043028

JEI Journal of Electronic Imaging

RESEARCH PAPER

Event-frame object detection under dynamic background condition

Wenhao Lu, Zehao Li, Junying Li, Yuncheng Lu, and Tony Tae-Hyoung Kim*
Nanyang Technological University, School of Electrical and Electronic Engineering, Singapore

ABSTRACT. Neuromorphic vision sensors (NVS) with the features of small data redundancy and transmission latency are widely implemented in Internet of Things applications. Previous studies have developed various object detection algorithms based on NVS's unique event data format. However, most of these methods are only adaptive for scenarios with stationary backgrounds. Under dynamic background conditions, NVS can also acquire the events of non-target objects due to its mechanism of detecting pixel intensity changes. As a result, the performance of existing detection methods is greatly degraded. To address this shortcoming, we introduce an extra refinement process to the conventional histogram-based (HIST) detection method. For the proposed regions from HIST, we apply a practical decision condition to categorize them as either object-dominant or background-dominant cases. Then, the object-dominant regions undergo a second-time HIST-based region proposal for precise localization, while background-dominant regions employ an upper outline determination strategy for target object identification. Finally, the refined results are tracked using a simplified Kalman filter approach. Evaluated in an outdoor drone surveillance with an event camera, the proposed scheme demonstrates superior performance in both intersection over union and *F*1 score metrics compared to other methods.

 [DOI: 10.1117/1.JEI.33.4.043028]

Keywords: neuromorphic vision sensor; event data; object detection; dynamic background

Paper 240378G received Apr. 13, 2024; revised Jun. 28, 2024; accepted Jul. 2, 2024; published Jul. 25, 2024.

1 Introduction

The Internet of Things (IoT)[1,2] is a network of interconnected physical devices equipped with sensors, software, and connectivity to exchange data. Inside the network, all the data can be transferred automatically, without human-to-human or human-to-computer interaction. IoT has various applications, e.g., smart cities,[3,4] industrial automation,[5,6] wearable technologies,[7,8] and surveillance tasks.[9,10]

As the "eyes" of the IoT, vision sensors play an important role in collecting video data and providing visual information about their surroundings. However, traditional vision sensors are not suitable for the IoT. They capture images at a constant sampling rate, irrespective of the dynamics of the scenario. This principle has high requirements for power consumption and transmission bandwidth. However, IoT usually has limited resources. In contrast to conventional sensors, the neuromorphic vision sensor (NVS) acquires event information based on the detection of changes in pixel intensity.[11,12] This characteristic greatly shrinks transmission latency and data redundancy. Hence, the NVS is widely used in IoT applications.

*Address all correspondence to Tony Tae-Hyoung Kim, thkim@ntu.edu.sg

1017-9909/2024/$28.00

In the last decade, some studies have focused on object detection based on the event data collected by NVS. In Refs. 13 and 14, the connected component labeling (CCL) method is applied for detection. It scans a constructed event-based image pixel by pixel to find and label the connected regions. During the scan, CCL considers that all the non-zero pixels within a neighboring adjacent area share the same label; otherwise, a new label is assigned. In this way, the target object is localized. Apart from CCL, Refs. 15 and 16 adopt a histogram-based (HIST) method. They build a histogram of an event-based image for each axis. The target object is in the areas where the associated histogram values are greater than a predefined threshold. These methods are friendly for hardware realization, with low memory and computation cost. However, the good performance of CCL and HIST is limited to stationary background cases, e.g., only target objects are moving in the scenario. For dynamic background, NVS can also acquire the events of non-target objects due to the mechanism of detecting pixel intensity changes. Since the above methods are not able to remove the influence of background, many false localization results are generated.

This paper proposes an object detection algorithm optimized for the data with event format. We focus on making the detection performance accurate and robust under dynamic background conditions. For the initial region proposal (RP) result obtained by the HIST method, an extra process is introduced to refine it. Based on a practical decision condition, the initial RP results are categorized into object-dominant and background-dominant cases. For the object-dominant case, the HIST method is used again to localize the target object more accurately. For the background-dominant case, an upper outline determination strategy helps find the target object position. Experiments verify the superior performance of our scheme in terms of intersection over union (IoU) and $F1$ score metrics.

The paper is structured as follows. Section 2 presents the proposed object detection background. In Sec. 3, we conduct experiments to validate our scheme. Finally, Sec. 4 gives concluding remarks to summarize the contributions.

2 Proposed Detection Algorithm

Figure 1 shows the general procedure for the proposed detection algorithm. First, an event-based frame is produced. Then, the noise events existing in the frame are mitigated. Afterward, the histogram-based RP method is used for the precleaned frames to find out the potential object area. Considering that the dynamic background may degrade the RP performance, we apply a decision condition to roughly check whether the proposed region is dominant by background or object. For the case that the proposed region is mainly object, the histogram-based RP is applied again to obtain a more accurate region. For the case that the proposed region is mainly background, we adopt "upper outline" determination and conditional judgment to identify the object area. Finally, the RP results from the above two cases are fed to trackers. The tracking procedure is with a simplified Kalman filter algorithm. This section presents each part in detail.

2.1 Frame Generation

For an NVS, it operates by detecting when and where the intensity change happens in the real scene. Every intensity change detected by a pixel in the NVS is called an event. The output of the NVS is the stream of events with address event representation (AER) format. Let us denote the j'th event as e_j. Each event consists of the associated pixel's coordinate (x_j, y_j) in which e_j has occurred, time stamp t_j which records when e_j has occurred, and the polarity p_j which indicates whether the intensity changes positively or not, i.e., $p_j \in \{-1,1\}$. Without loss of generality, the mathematical form of the j'th event is defined as $e_j = \{x_j, y_j, t_j, p_j\}$. Based on the fact that the NVS is always awake while the processor (e.g., FPGA or ASIC chip where the proposed algorithm is realized) goes to sleep and wakes up regularly, we manually set a time interrupt between NVS and processor to accumulate events, as shown in Fig. 2. In other words, the NVS performs as a memory. After a reasonable fixed time interval, the accumulated events form an event-based frame. Please be reminded that this frame is a binary image.

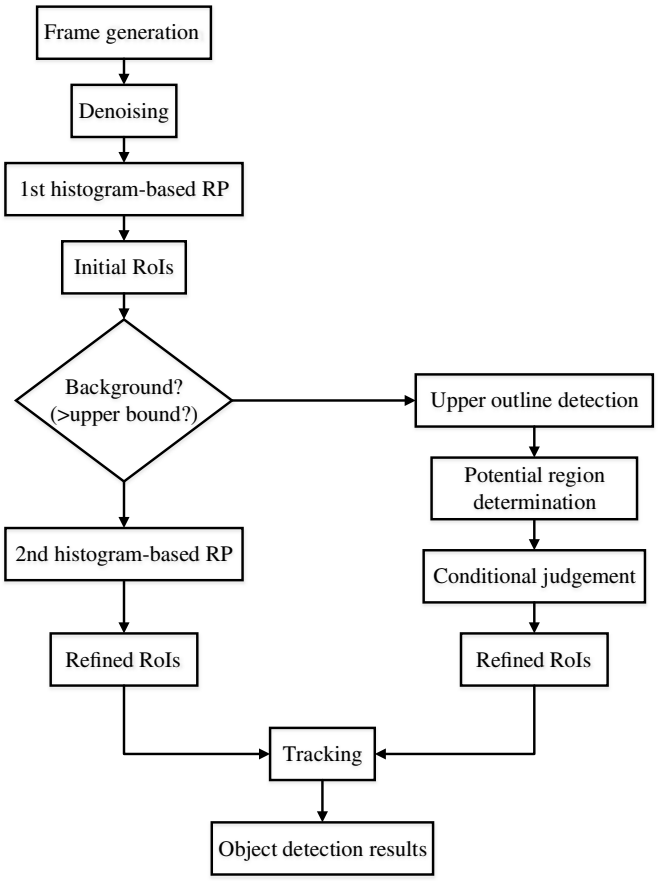

Fig. 1 The flow chart of the proposed detection algorithm.

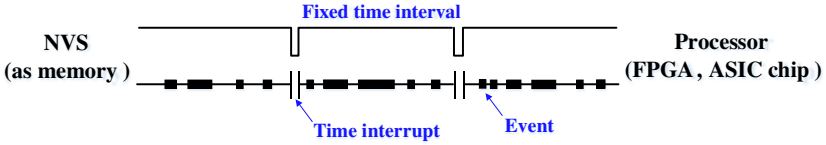

Fig. 2 Frame generation based on the time interrupt between NVS and processor.

2.2 Noise Removal

As shown in Fig. 3(a), a large number of noise events usually go along with the actual signals in the output of NVS. It is because the electronic noise, e.g., thermal noise, shot noise, and 1/f noise, can trigger false signals, which are interpreted as changes in light intensity by the sensor's pixels, even though no real change happens in the light environment. Since these hardware noises may affect the performance of the detection algorithm, we must filter the noise. That means the event-based frame should be precleaned before detection. A median filter is applied to remove the noise events in the frame. As an order-statistic filter, the median filter exhibits excellent performance for salt and pepper noise. It replaces the center pixel of a 3×3 neighborhood with the median value of the corresponding window. Consequently, the noise impact is greatly suppressed. For example, as shown in Fig. 3(b), most noise events are removed in the precleaned frame, while the actual signal is retained well.

2.3 Region Proposal

After noise removal, the precleaned frame is used for RP. It aims to find out some potential areas for the target object. These areas are also called regions of interest (RoIs). Due to the low memory and computation cost for hardware, the HIST-based RP method has been widely used in many

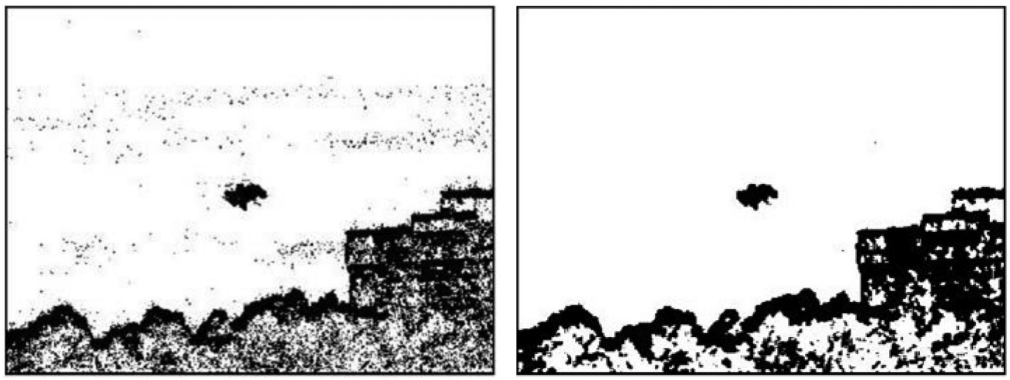

Fig. 3 Event-based frames in drone surveillance tasks. (a) Without denoising operation. (b) With denoising operation.

resource-constraint IoT applications. We also apply this method to get the initial RoIs in our work. Figure 4(a) shows its concept. Given an event-based frame, we first calculate the histograms along X-axis and Y-axis, respectively. They are denoted as H_x and H_y, respectively. Then, for each entry of H_x (H_y), we compare it with a threshold T_x (T_y) in order to find consecutive entries where all the associated H_x (H_y) values are greater than T_x (T_y). Finally, the RoIs for the target object are the intersections of the selected X and Y regions. It can be seen that under a

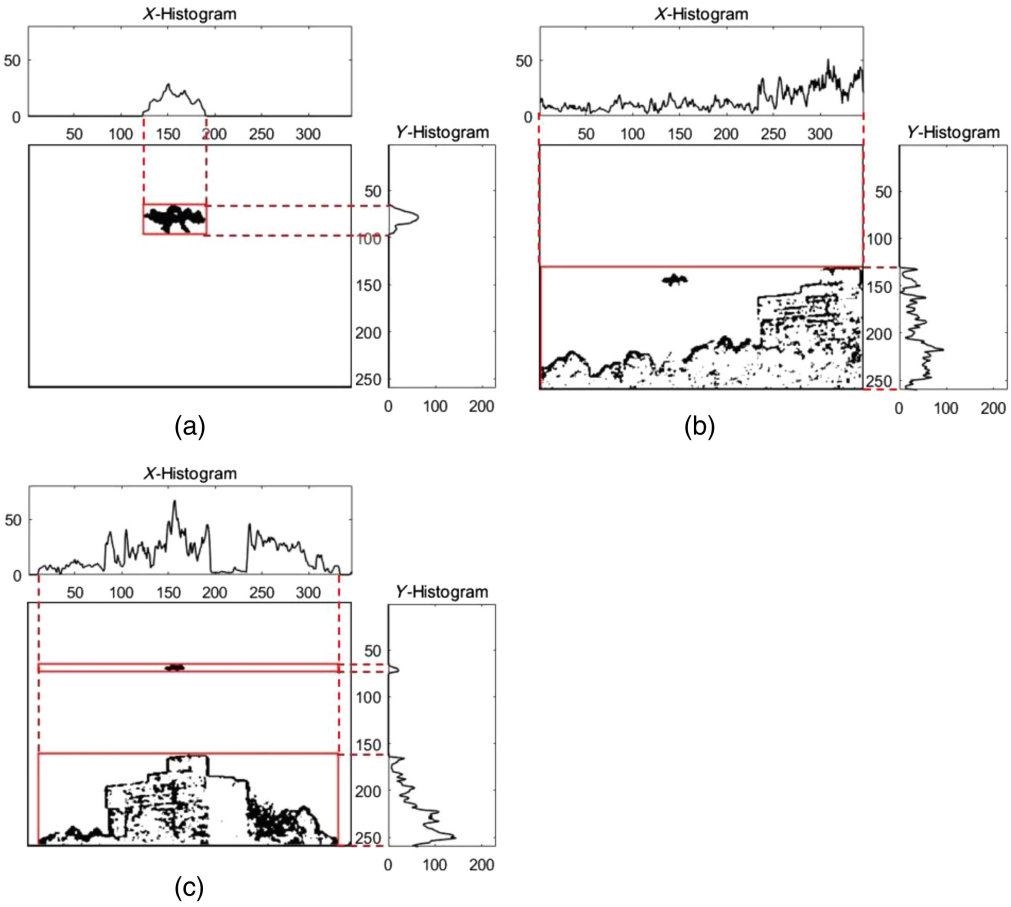

Fig. 4 Results of histogram-based RP in drone surveillance tasks. (a) Under stationary background. (b) Under dynamic background and the initial RoI is background-dominant. (c) Under dynamic background and the initial RoI is object-dominant in the top bounding box.

stationary background, the target object is located accurately using the above method. However, as shown in Figs. 4(b) and 4(c), when the background is dynamic, i.e., non-target objects (trees and buildings) also exist in the frame, the histograms H_x and H_y are highly distorted. After intersecting the selected X and Y regions, we obtain the bounding boxes that cover the entire X-axis. Clearly, the performance of the HIST-based RP method degrades a lot.

The above discussion tells us that under dynamic background, it is hard to differentiate between target and non-target objects by simply measuring histograms. Hence, we need to mitigate the effect of non-target objects in the background. This work designs an additional refinement process to improve the histogram-based RP results. A practical condition is used to roughly identify whether the initial RoIs are dominant by background or target object. This condition is derived from a relationship between the actual target object size and its size in an event-based frame, given as

$$\omega = \frac{\psi \times \mu}{\rho \times \eta}, \tag{1}$$

where ω is the width of the target object in a frame, ψ is the focal length, μ is the actual width of the target object, ρ is the distance between the target object and focal lens, and η is the actual width of a pixel in the frames generated by an event camera. The upper bound on ω is given as

$$\omega \leq \frac{\max(\psi) \times \max(\mu)}{\min(\rho) \times \eta}. \tag{2}$$

If the size of an initial RoI exceeds $\max(\omega)^2$, we consider that the background is dominant in this RoI, denoted as a "background-dominant" case. Otherwise, we consider that the target object is dominant, denoted as an "object-dominant" case. Next, the background-dominant case and object-dominant cases are discussed.

2.3.1 *Background-dominant case*

Although the background is dominant in the initial RoI, the target object may still exist. It is because if the histogram of the target object is included in that of the background along both the X-axis and Y-axis, this object can be mis-considered as part of the background, as shown in Fig. 4(b). Note that an event camera is designed to record pixel-level changes in luminance. It only captures the pixels where intensity changes. When the target object overlaps with a dynamic background, e.g., a drone flies within a closed or semi-closed structure, the pixel intensity changes of the target object interweave with the pixel intensity changes of dynamic background (other moving objects or light variations) in both time and space. As a result, it is hard to identify whether the generated event data are caused by the target object or other objects in the background. Hence, we exclude the situation that the target object overlaps the background in event camera-based applications. Besides, as shown in Figs. 3 and 4, the target object (drone) usually exists above the ground and other background objects. With these considerations, we assume that the target object is above the background. The refinement procedure is as follows:

1. With the assumption, we first find out the upper outline of the initial RoI. Figure 5(a) shows the detected upper outline of the initial RoI in Fig. 4(b).
2. Since the target object is above the background, for the obtained upper outline, the position of the target object's start point should be higher than that of the neighboring former outline point. Meanwhile, the distance between these two points must exceed a threshold T_r. Similarly, the target object's endpoint should be higher than the position of the neighboring latter outline point. Also, the distance between these two points must be greater than T_r. With these concepts, the potential range of the target object along the X-axis is determined, as shown in Fig. 5(b).
3. The right and left sides of the target object have been fixed from previous steps. This step is to identify the top side. Within the determined right and left boundaries, we scan the area commencing from the first row and proceeding to locate the foremost row where at least a

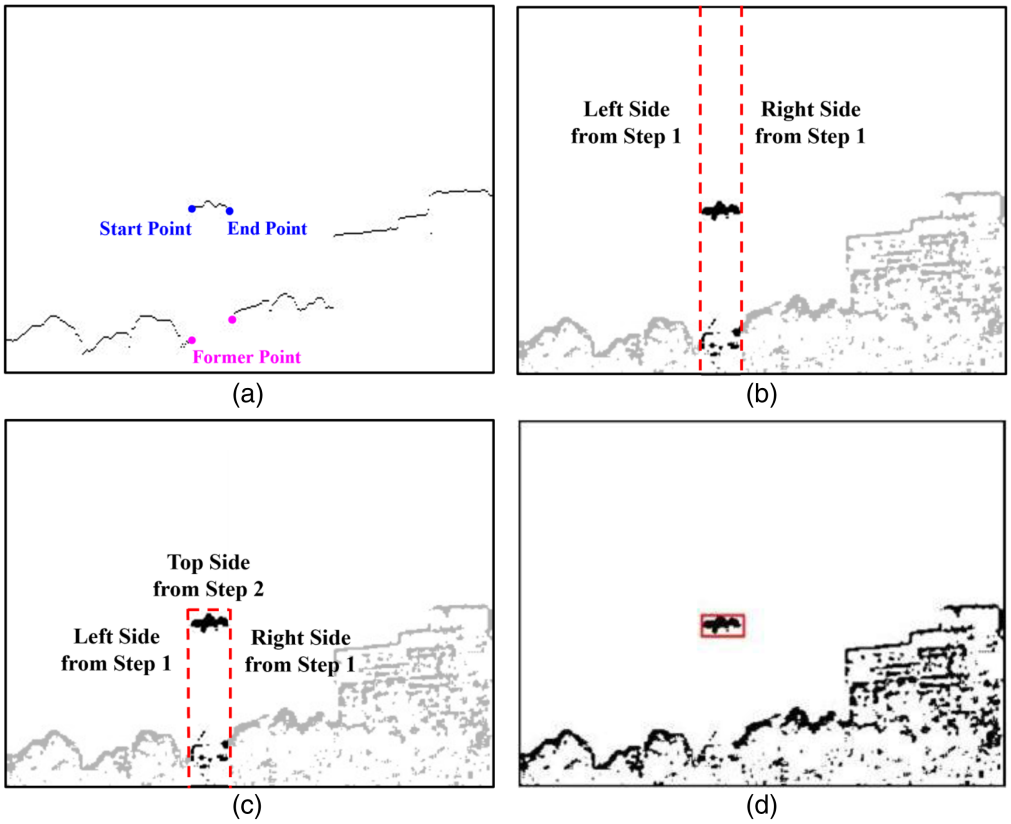

Fig. 5 The refinement procedure for the background-dominant case. (a) The upper outline is detected. (b) The left and right sides are determined. (c) The top sides are determined. (d) The target object is bounded.

single pixel exhibits a value of zero. The index of this particular row is considered as the top side of target object, as shown in Fig. 5(c).

4. For the region between the fixed right and left side, from the top side of the target object obtained by step 3, we find the foremost row where all the pixels are equal to one. This particular row is considered as the bottom side of the target object. Consequently, the target object is bounded, as shown in Fig. 5(d).
5. We may get several target object candidates from the above four steps. For each candidate, the following criteria (necessary conditions) help us to check if it is a real target object: (i) based on the fact that the background events are dominant among the entire events in the initial ROI, we measure the relative frequency $\xi =$ (the number of events in candidate)/(the number of entire events in the initial RoI). If ξ is less than or equal to a threshold T_ξ, this candidate may belong to a target object; (ii) we measure the density of the potential area, denoted as Λ. If Λ is greater than or equal to a threshold T_Λ, this candidate may belong to a target object; and (iii) if the potential area belongs to a target object, its aspect ratio (or called length-width ratio) R_{ap} should be in a reasonable range.

2.3.2 *Object-dominant case*

Although the initial RoI does not locate the target object tightly, it removes most of the background information. For example, as shown in the top bounding box in Fig. 6(a), the detected object width is as long as that of the background. However, all the non-target objects are excluded. Therefore, we can get a better RP result by using the HIST-based RP algorithm again for the initial ROI. Similar to the background-dominant case, the density Λ and the height-width ratio R_{wh} of the refined area are also measured to check if it is a real target object. As shown in Fig. 6(b), a second HIST-based RP can accurately locate the drone position.

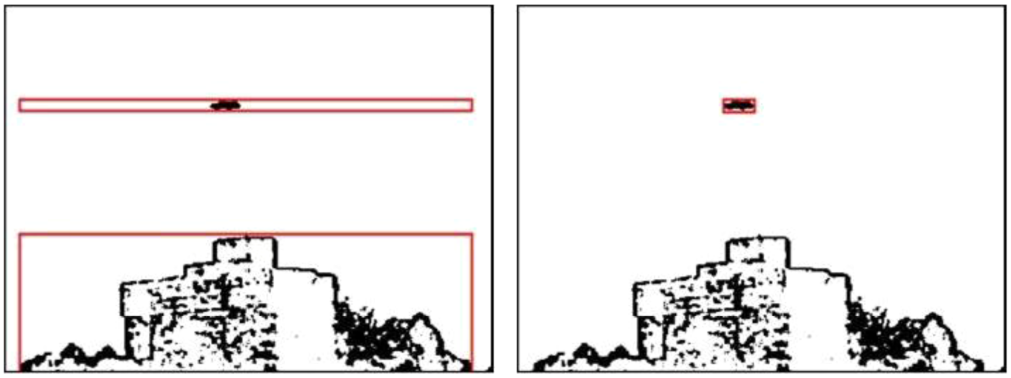

Fig. 6 The proposed areas within the top bounding box is object-dominant. (a) Single histogram-based RP result. (b) Double histogram-based RP result. Note that the bottom bounding box belongs to the background-dominant case.

2.4 Tracking

Once an RoI is proposed, it is continuously monitored by a tracker. By monitoring a RoI's presence and movement, we enable to further differentiate between target objects and non-target objects in the background, thereby minimizing the chances of mis-detection. Each tracker is characterized by a set of parameters: (1) the coordinates of the top-left location of the bounding box, denoted as $(\mathcal{X}, \mathcal{Y})$; (2) the width w and height h of the bounding box; (3) the velocities of the bounding box along X- and Y-axes, respectively; and (4) the tracker state, which can be "free," "tracking," or "locked." The "free" state indicates that the tracker is not actively tracking any specific region. Conversely, the "tracking" state signifies that the tracker has successfully aligned with a proposed region in one frame. Lastly, a tracker is labeled as "locked" when it consistently aligns with an RoI over two consecutive frames.

The tracking procedure is based on a simplified Kalman filter algorithm.[16] Denote the parameters of the m'th RoI in the j'th frame as $\mathcal{R}_j^m = \{\mathcal{X}_j^m, \mathcal{Y}_j^m, w_j^m, h_j^m\}$, where $m = 1, \cdots, M$. Also, denote the parameters of the n'th tracker in the j'th frame as $\mathcal{T}_j^n = \{\mathcal{X}_j^n, \mathcal{Y}_j^n, w_j^n, h_j^n\}$, where $n = 1, \cdots, N$. The assignment process is as follows. First, we measure the overlap area between the RoI and each tracker, given as

$$\begin{aligned} \mathrm{OV}_{\mathrm{area}} = &\max(0, \min(\mathcal{X}_{j-1}^n + w_{j-1}^n, \mathcal{X}_j^m + w_j^m) - \max(\mathcal{X}_{j-1}^n, \mathcal{X}_j^m)) \\ &\times \max(0, \min(\mathcal{Y}_{j-1}^n + h_{j-1}^n, \mathcal{Y}_j^m + h_j^m) - \max(\mathcal{Y}_{j-1}^n, \mathcal{Y}_j^m)). \end{aligned} \tag{3}$$

With $\mathrm{OV}_{\mathrm{area}}$, the overlap ratio between an RoI and a tracker is given as

$$\mathrm{OV}_{\mathrm{ratio}} = \frac{\mathrm{OV}_{\mathrm{area}}}{w_{j-1}^n \times h_{j-1}^n}. \tag{4}$$

If $\mathrm{OV}_{\mathrm{ratio}}$ is greater than a predefined threshold $\mathrm{OV}_{\mathrm{th}}$, this RoI is assigned to the tracker. Otherwise, we assign this RoI to a tracker in which the status is free. Then, the tracker status is renewed. For the case that the previous tracker state is "free," it will be changed to "tracking." For the case that the previous tracker state is "tracking," it will be changed to "locked." For the case that the status has already been "locked," it remains unchanged. In addition, this tracker's parameters are updated based on the weighted average between $\mathcal{R}_j$ and $\mathcal{T}_{j-1}$. Let β be the weighting degree coefficient. The update procedure is governed as

$$\mathcal{T}_j^n = (1-\beta)\mathcal{R}_j^m + \beta(\mathcal{T}_{j-1}^n + \zeta_{j-1}^n \times \Delta t),$$

where Δt is the difference between time stamps in the j'th frame and the $(j-1)$'th frame, and ζ_j^n is the velocity of the n'th tracker in the frame. Denote the velocity of the n'th tracker along X- and Y-axes as $\zeta_j^n(\mathcal{X})$ and $\zeta_j^n(\mathcal{Y})$, respectively. These two velocities are derived from

$$\zeta_j^n(\mathcal{X}) = (1-\beta) \times \frac{(\mathcal{X}_j^m - \mathcal{X}_{j-1}^n) + (w_j^m - w_{j-1}^n)}{\Delta t} + \beta \times \zeta_{j-1}^n(\mathcal{X}),$$
$$\zeta_j^n(\mathcal{Y}) = (1-\beta) \times \frac{(\mathcal{Y}_j^m - \mathcal{Y}_{j-1}^n) + (h_j^m - h_{j-1}^n)}{\Delta t} + \beta \times \zeta_{j-1}^n(\mathcal{Y}),$$

respectively. We also eliminate those trackers which lose RoIs. Suppose a tracker was marked as locked or tracking state in the previous frame. If this tracker does not match any RoI in the current frame, it is removed. For more details, refer to Ref. 16.

3 Experiment Results

A DAVIS346 camera with the resolution of 346×260 is used to monitor drone flying in an outdoor environment. It collects event data in AER format. Table 1 shows the details of the data collection settings. From Table 1, the focal length is 50 or 100 mm. The drone's width is less than or equal to 81 cm. The distance between the target object and the focal lens is greater than or equal to 100 m. To make the decision condition stated in Eq. (2) tolerant for more real scenarios, we set $\max(\psi)$ as 100 mm, $\max(\mu)$ as 1 m, and $\min(\rho)$ as 50 m. The collected data are converted into event-based frames. The time interval for each frame is 40 ms. In addition, the actual pixel width η in the frames generated by DAVIS346 is 18.5 μm. A manual annotation is implemented on these frames to produce the ground truth for the target objects in the associated scenarios. Four hundred event-based frames with dynamic backgrounds are selected for the test. Three factors are considered for the frame selection: the types of background objects, the degree of pixel intensity changes, and the distance between the drone and the camera during data collection. In the drone surveillance task, many background objects can also be captured, e.g., buildings, clouds, trees, and birds, as shown in the second row of Fig. 7. It is important to identify whether these background objects affect the performance of our detection algorithm or not. We also include frames that exhibit a wide range of intensity changes, as shown in the fourth row of Fig. 7. These frames are used to evaluate the robustness of the proposed algorithm across different levels of changes in pixel intensity. Besides, event-based frames obtained from various distances (100 to 400 m) between the drone and the camera are selected to enrich the test dataset, as shown in the sixth row of Fig. 7. Note that different flying distances result in variations in the apparent size and shape of the drones within the frames. It may challenge the detection algorithm's ability to generalize across different scenarios.

There are three kinds of object detection schemes: the CCL method,[13] the histogram-based method,[16] and our proposed method. They are denoted as CCL, HIST, and PROPOSED, respectively. The threshold settings in our scheme are as follows. Recall that T_r is the threshold for the distance between the target object's start (end) point and the neighboring former (later) outline point. Let us denote the minimum apparent drone width in frames as $\min(\omega)$. We use $\min(\omega)$ to guide the determination of T_r. In practice, an event camera with a focal length ψ of 50 mm is able to capture drones at a distance ρ of up to 200 m. When ψ becomes 100 mm, ρ is up to 400 m. From Eq. (1), the minimum apparent drone width is given as

$$\min(\omega) = \frac{\psi \times \min(\mu)}{\max(\rho) \times \eta} = \frac{50 \text{ mm} \times 40 \text{ cm}}{200 \text{ m} \times 18.5 \ \mu\text{m}} = \frac{100 \text{ mm} \times 40 \text{ cm}}{400 \text{ m} \times 18.5 \ \mu\text{m}} \approx 5.$$

As the drones cannot fly too close to background objects, we set T_r as triple $\min(\omega)$, i.e., $T_r = 15$. T_ξ is the threshold for the relative frequency ξ. To determine the value of T_ξ, we

Table 1 Summary of data collection.

Time	10:00 am to 7:00 pm
Drone size	40×40 cm and 81×67 cm
Focal lens	50 and 100 mm
Distance	100 to 400 m

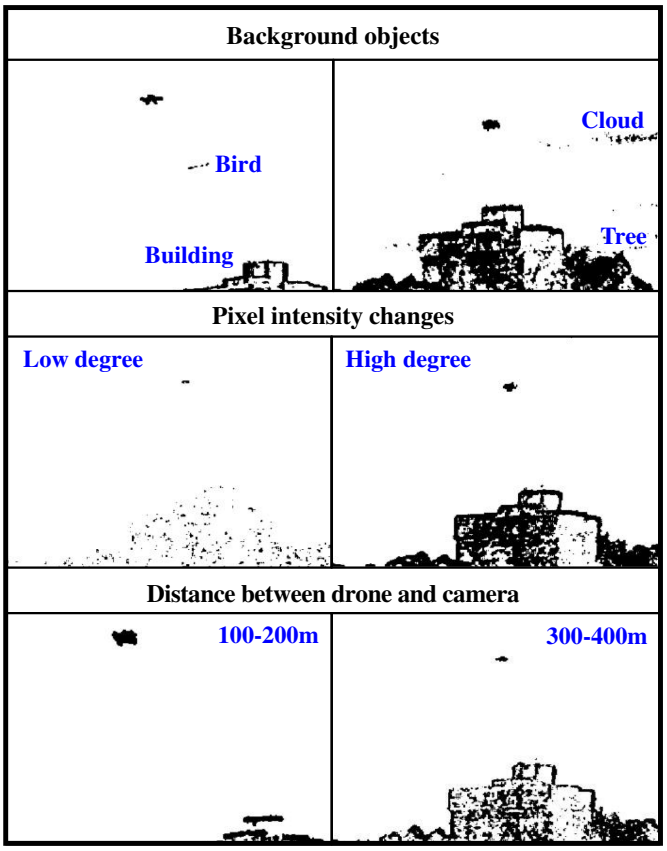

Fig. 7 Samples of event-based frame selection for the test dataset.

randomly choose 100 event-based frames in which the background dominates the initial RoI. For each frame, the relative frequency is calculated, which is given as

$$\frac{\text{the number of events within the annotated drone area}}{\text{the number of the entire events in the initial RoI}}.$$

After that, we obtain the average relative frequency value u_ξ and the associated standard deviation σ_ξ. Since three sigma limits are usually applied to build the boundary in statistical control, we set T_ξ (the upper bound on the relative frequency) as

$$u_\xi + 3 \times \sigma_\xi = 0.0373 + 3 \times 0.0182 \approx 0.1.$$

T_Λ is the threshold for the density of the potential area. Similar to T_ξ, 100 event-based frames are randomly selected from the collected data. Then, the density of the annotated drone area is calculated for each frame, given as

$$\frac{\text{the number of events within the annotated drone area}}{\text{the size of the annotated drone area}}.$$

Again, we derive the average density value u_Λ and the associated standard deviation σ_Λ. With three sigma limits, T_Λ (the lower bound on the density) is set as

$$u_\Lambda - 3 \times \sigma_\Lambda = 0.4602 - 3 \times 0.0231 \approx 0.4.$$

The range of aspect ratio R_{ap} is based on the actual drone size. For multirotor drones, their aspect ratio is close to 1:1. For fixed-wing drones, their wingspan is usually much greater than the fuselage, and the aspect ratio can reach 4:1. Hence, we set R_{ap} within 1 and 4, i.e., $1 \leq R_{\text{ap}} \leq 4$. OV_{th} is the threshold for the overlap ratio OV_{ratio} as stated in Eq. (4). Suppose the minimum drone speed during data collection is at a very low level, e.g., 10 km/h. Since the time interval for each frame is only 40 ms, the movement of the drone approximates horizontal/vertical translation. With Eq. (1), the overlap ratio OV_{ratio} can be expressed as

$$\begin{aligned}\mathrm{OV}_{\mathrm{ratio}} &= \frac{\frac{\psi\times(\text{drone speed}\times\text{each frame's time interval})}{\rho\times\eta}}{\frac{\psi\times\mu}{\rho\times\eta}} \\ &\geq \frac{\min(\text{drone speed})\times\text{each frame's time interval}}{\max(\mu)} \\ &= \frac{10\ \text{km/h}\times 40\ \text{ms}}{1\text{m}} \\ &\approx 0.1.\end{aligned}$$

From the above equation, we can easily get that the threshold for the overlap ratio is equal to 0.1, i.e., $\mathrm{OV}_{\mathrm{ratio}} = 0.1$.

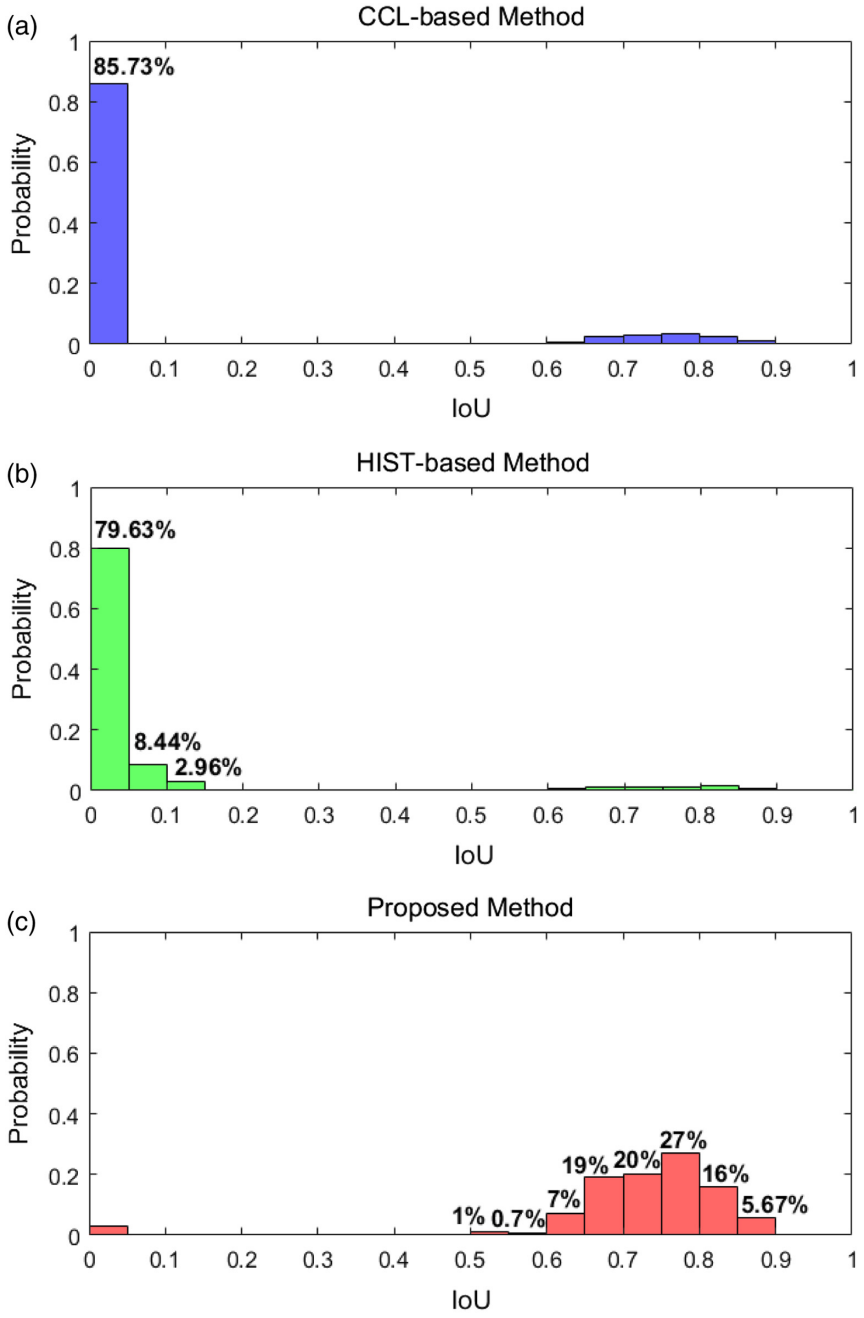

Fig. 8 The probability distribution of measured IoU values. (a) CCL-based method. (b) HIST-based method. (c) Proposed method.

The resultant algorithms are evaluated based on two metrics. One is called the IoU, given as

$$\text{IoU} = \frac{A_{\text{GT}} \cup A_{\text{PT}}}{A_{\text{GT}} \cap A_{\text{PT}}},$$

where A_{GT} is the area of a ground truth box enclosing the target object, and A_{PT} is the area of a tracker box enclosing a detected object. This metric determines how well the algorithm's detection result aligns with the actual target object in the image. A higher IoU score means better object detection performance. Suppose that a correct detection happens when the IoU is greater than a threshold. We also measure the $F1$ score under various IoU threshold. It is the harmonic mean of precision rate $\hat{P}$ and recall rate $\hat{R}$, given as

$$F1 = 2 \times \frac{\hat{P} + \hat{R}}{\hat{P} \times \hat{R}},$$

$$\hat{P} = \frac{\text{number of true positive boxes}}{\text{number of proposed boxes}},$$

$$\hat{R} = \frac{\text{number of true positive boxes}}{\text{number of ground truth boxes}}.$$

Since precision and recall rates are in a trade-off relationship, the $F1$ score provides a comprehensive way to evaluate the detection performance. Algorithms with high $F1$ scores often have a good balance between precision and recall rates. The results are summarized in Figs. 8 and 9.

Figure 8 shows the distribution of IoU results for the resultant methods. It can be seen that for the CCL-based method, the measured IoU values are mainly distributed between 0 and 0.05. That means only a minor overlap exists between the detected and ground truth bounding boxes. The HIST-based method also displays a similar distribution, with a majority of detections at low IoU values. Again, this method has limited detection capability under dynamic background. For the proposed scheme, about 96.37% of measured IoU results are greater than 0.5. It implies a more robust and accurate object detection approach.

Figure 9 depicts the $F1$ score results. It is observed that for the proposed detection scheme, its $F1$ score remains at a high level within a wide range of the IoU threshold. As a comparison, the performance of HIST is poor, never reaching an $F1$ score of 0.4. For example, when the IoU threshold is 0.6, the $F1$ score of HIST is only 0.1089. It is because the dynamic background leads to the incorrect histogram computation for target objects. The CCL-based method's performance is similar to that of HIST. At the same IoU threshold level, the $F1$ score of our scheme is 0.9401. With the refinement process stated in Sec. 2.3, our scheme mitigates the effect of non-target objects. Hence, it works well. The experiment results agree with our design.

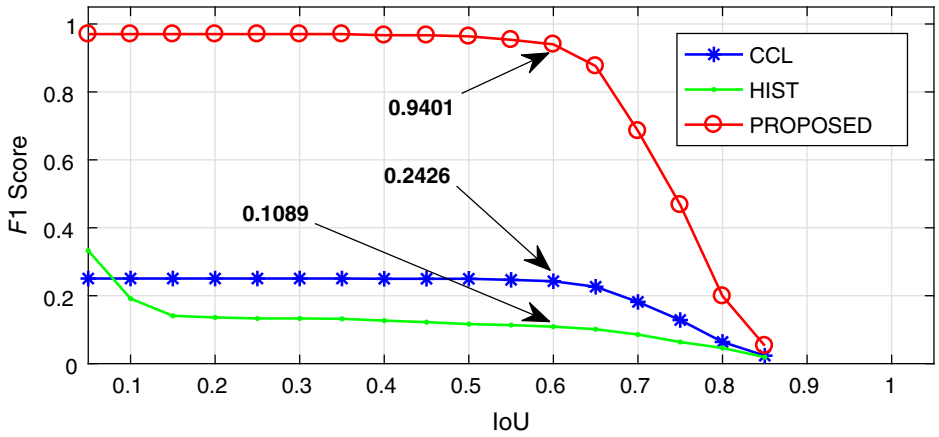

Fig. 9 Comparison of CCL-based, HIST-based, and proposed detection algorithm on test dataset in terms of $F1$ score for various IoU thresholds.

4 Conclusion

Considering the dynamic background, this paper presented an object detection algorithm optimized for the data with event format. For the proposed scheme, we first generated an event-based frame and used a median filter to remove noise. Then, with an HIST-based RP method, we got the initial RoIs within the precleaned frame. To mitigate the effect of non-target objects in the background, an extra process was introduced to refine the initial RoIs. With a practical decision condition, the initial RoIs was categorized into object-dominant and background-dominant cases. For the object-dominant case, we applied the histogram-based RP again to locate the target object accurately. For the background-dominant case, the upper outline determination strategy was used to find the target object position. Finally, the refined RoIs were tracked based on a simplified Kalman filter approach. The experiment results demonstrated the superior performance of our scheme in terms of the IoU and $F1$ score metrics.

Disclosures

No potential conflict of interest was reported by the authors.

Code and Data Availability

The archived version of the code described in this paper can be freely accessed through Github at https://github.com/dvsdetect/dvsdetect

Acknowledgments

This work was supported by ST Engineering Advanced Networks and Sensors Pte. Ltd.

References

1. P. Singh, B. Acharya, and R. K. Chaurasiya, "Low-area and high-speed hardware architectures of LBlock cipher for Internet of Things image encryption," *J. Electron. Imaging* **31**(3), 033012 (2022).
2. Y. B. Zikria et al., "Next-generation Internet of Things (IoT): opportunities, challenges, and solutions," *Sensors* **21**(4), 1174 (2021).
3. F. Cirillo et al., "Smart city IoT services creation through large-scale collaboration," *IEEE Internet Things J.* **7**(6), 5267–5275 (2020).
4. S. Jung and J. Kim, "Adaptive and stabilized real-time super-resolution control for UAV-assisted smart harbor surveillance platforms," *J. Real-Time Image Process.* **18**(5), 1815–1825 (2021).
5. K. Karunanithy and B. Velusamy, "Cluster-tree based energy efficient data gathering protocol for industrial automation using WSNs and IoT," *J. Ind. Inf. Integr.* **19**, 100156 (2020).
6. H. R. Chi et al., "A survey of network automation for industrial internet-of-things toward industry 5.0," *IEEE Trans. Ind. Inf.* **19**(2), 2065–2077 (2022).
7. J.-Y. Wu et al., "IoT-based wearable health monitoring device and its validation for potential critical and emergency applications," *Front. Public Health* **11**, 1188304 (2023).
8. S. D. Mamdiwar et al., "Recent advances on IoT-assisted wearable sensor systems for healthcare monitoring," *Biosensors* **11**(10), 372 (2021).
9. X. Zhou et al., "Deep-learning-enhanced multitarget detection for end–edge–cloud surveillance in smart IoT," *IEEE Internet Things J.* **8**(16), 12588–12596 (2021).
10. M. Abbas Fadhil Al-Husainy and B. Al-Shargabi, "Secure and lightweight encryption model for IoT surveillance camera," *Int. J. Adv. Tr. Comput. Sci. Eng.* **9**(2), 1840–1847 (2020).
11. F. Liao, F. Zhou, and Y. Chai, "Neuromorphic vision sensors: principle, progress and perspectives," *J. Semicond.* **42**(1), 013105 (2021).
12. P. Lichtensteiner, C. Posch, and T. Delbruck, "A 128x128 120dB 15μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits* **43**(2), 566–576 (2008).
13. V. Mohan et al., "Ebbinnot: a hardware-efficient hybrid event-frame tracker for stationary dynamic vision sensors," *IEEE Internet Things J.* **9**(21), 20902–20917 (2022).
14. D. Singla et al., "HyNNA: improved performance for neuromorphic vision sensor based surveillance using hybrid neural network architecture," in *IEEE Int. Symp. Circuits and Syst. (ISCAS)*, IEEE, pp. 1–5 (2020).
15. J. Acharya et al., "EBBIOT: a low-complexity tracking algorithm for surveillance in IoVT using stationary neuromorphic vision sensors," in *32nd IEEE Int. System-on-Chip Conf. (SOCC)*, IEEE, pp. 318–323 (2019).
16. A. Ussa et al., "A hybrid neuromorphic object tracking and classification framework for real-time systems," *IEEE Trans. Neural Netw. Learn. Syst.* 1–10 (2023).

Wenhao Lu received his PhD in electrical engineering from City University of Hong Kong, in 2022. He is currently a research fellow in the School of Electrical and Electronic Engineering, Nanyang Technological University. His research interests include neural networks and machine learning.

Zehao Li received his BE degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2020, where he is currently pursuing his PhD under the supervision of Prof. Kim Tae-Hyoung.

Junying Li received her BE degree in communication engineering from Beijing Jiaotong University, China, in 2021, and her MS degree in electronics engineering from Nanyang Technological University, Singapore, in 2023, where she is currently pursuing her PhD. In 2023, she joined the Center for Integrated Circuits and Systems, Nanyang Technological University, as a research associate.

Yuncheng Lu received his BE degree in electronic science and engineering from Harbin Institute of Technology, Weihai, China, in 2017, and his MS degree in electronics from Nanyang Technological University, Singapore, in 2018. He is currently pursuing his PhD at Nanyang Technological University.

Tony Tae-Hyoung Kim received his PhD in electrical and computer engineering from the University of Minnesota, Minneapolis, Minnesota, USA, in 2009. From 2001 to 2005, he worked for Samsung Electronics. In November 2009, he joined Nanyang Technological University where he is currently an associate professor. His current research interests include computing-in-memory for machine learning and energy-efficient circuits and systems for edge computing.