Dynamic Object Avoidance using Event-Data for a Quadruped Robot

Shifan Zhu, Nisal Perera, Shangqun Yu, Hochul Hwang, and Donghyun Kim*

***Abstract*—As robots increase in agility and encounter fast-moving objects, dynamic object detection and avoidance become notably challenging. Traditional RGB cameras, burdened by motion blur and high latency, often act as the bottleneck. Event cameras have recently emerged as a promising solution for the challenges related to rapid movement. In this paper, we introduce a dynamic object avoidance framework that integrates both event and RGBD cameras. Specifically, this framework first estimates and compensates for the event's motion to detect dynamic objects. Subsequently, depth data is combined to derive a 3D trajectory. When initiating from a static state, the robot adjusts its height based on the predicted collision point to avoid the dynamic obstacle. Through real-world experiments with the Mini-Cheetah, our approach successfully circumvents dynamic objects at speeds up to $5\,\mathrm{m/s}$, achieving an 83% success rate.**

Supplemental video: https://youtu.be/wEPvynkVlLA

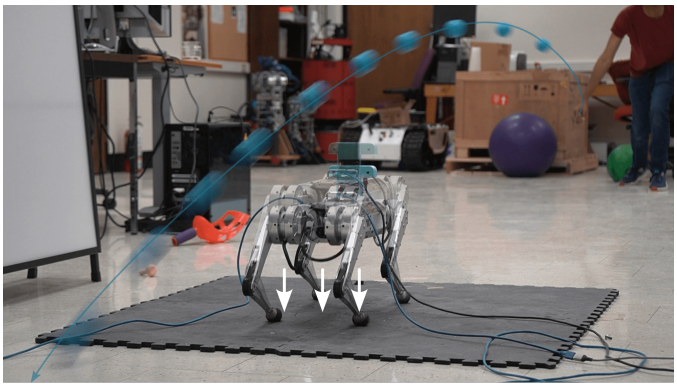

Fig. 1. Event and depth sensor-equipped robot avoiding a dynamic object.

I. Introduction

Recent progress in artificial intelligence, robot control, computer vision, and computing power advanced the perception-based collision avoidance capability of robotic systems [1]–[6]. Nonetheless, recognizing and avoiding highly dynamic objects (e.g., a high-speed car, kicked ball) remains an unsolved challenge because of the inherent limitations of conventional cameras. For example, RGB cameras' low temporal resolution and dynamic range make it difficult to capture rapid objects unless the bright and consistent light condition is secured. Such constraints hinder the robot's ability to swiftly respond to rapidly moving objects, thus posing significant obstacles to safe navigation.

Event cameras, referred to as neuromorphic cameras, have been recently introduced as a promising solution to address aforementioned challenges [7] [8]. In contrast to conventional shutter-based cameras that capture entire images at a regular frame rate, event cameras asynchronously detect local brightness changes at the pixel level. By capturing distinct local events, the event camera offers benefits like high dynamic range, minimal latency, energy efficiency, and superior motion blur mitigation [9]. In this work, we leverage the capabilities of event-based neuromorphic cameras to differentiate a dynamic object from a static scene. Furthermore, we capitalize on the event camera's high-temporal resolution attribute to avoid a fast-moving object with our quadruped system.

While event cameras offer numerous advantages, their data often exhibits higher noise levels and is captured at a relatively low resolution compared to the dense pixel information provided by frame-based cameras. Another aspect to note is that event data is motion-dependent; edges parallel with motion usually do not generate events, as these events arise solely from brightness changes caused by the camera's relative motion. Moreover, event cameras solely capture changes in brightness, omitting visual details such as color and texture. Given the fundamental differences in sensing mechanisms between RGB and event cameras, directly applying RGB image-based object avoidance algorithms to event data is not feasible.

The methodology of leveraging motion to refine event data and compute pixel-specific metrics has been explored by numerous researchers. Mitrokhin et al. [10] incorporated x-shift, y-shift, expansion, and 2D rotation into their motion compensation. Falanga et al. [11] managed to reduce the latency to $3.5\,\mathrm{ms}$ by leveraging the rotational data from an IMU, though they didn't account for translational motion. He et al. [12] filled this gap, compensating for both rotational and translational movements and deploying an adaptive threshold driven by angular and linear velocities. However, these approaches share the same drawback: their detection results tend to be either noisy or incomplete. This challenge arises from relying on a single threshold: while a higher value curtails noise, it may omit crucial detection areas, and a lower value can introduce excessive noise (Fig. 3). Thus, the introduction of a dual-threshold detection method stands out as a promising solution for refining results.

In this paper, we divide the dynamic object detection algorithm into two scenarios to accommodate the event camera's distinct sensing mechanisms. (1) In situations of static robot motion, the event camera's motion-dependent mechanism exclusively captures event data associated with dynamic

*University of Massachusetts Amherst, 140 Governors Dr, U.S. donghyunkim@cs.umass.edu

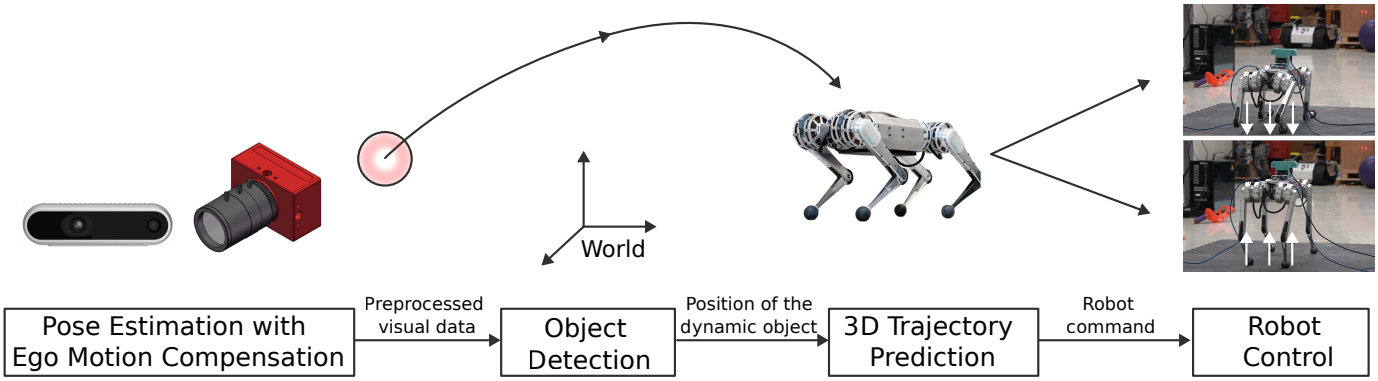

Fig. 2. Overall pipeline: Our proposed method consists of (a) Ego motion estimation and compensation, (b) dynamic object detection, (c) 3D trajectory prediction, and (d) robot control.

objects, thereby automatically filtering out the background. (2) Conversely, during dynamic robot motion, events are not just triggered by dynamic objects but also by the static background, necessitating motion compensation to accurately detect dynamic objects.

Our objective is to develop a system that not only detects objects but also predicts their trajectories and avoids them deftly. Given time constraints, we harness the capabilities of event cameras to showcase avoidance capability in the first scenario. We additionally present the detection result under camera motion and anticipate conducting more comprehensive experiments in the future.

Our contributions are outlined as follows: Firstly, we devised an object avoidance framework that combines the capabilities of event and depth cameras to accomplish swift and robust dynamic object avoidance. Secondly, we conducted evaluations on a quadruped robotic platform to validate the effectiveness of our proposed system. Our results highlight the system's proficiency in avoiding high-speed objects, representing a substantial advancement towards agile navigation in legged robots. Thirdly, we presented dynamic object detection results that surpass the performance of current leading detection algorithms.

II. Method

The framework comprises four key components illustrated in Fig. 2. Firstly, we calibrate intrinsic and extrinsic parameters and synchronize the cameras to guarantee optimal detection outcomes. Then, we utilize our previous work [13] to estimate ego-motion and compensate event data. Subsequently, a detection algorithm is applied to obtain 2D object positions within the pixel frame. By integrating the 2D positions with depth data, we can obtain 3D positions. Leveraging these 3D positions, the 3D trajectory prediction module calculates the object's trajectory and predicts its future collision point. This prediction facilitates the system in determining the optimal height for the robot. Finally, the robot control algorithm executes the appropriate high-level command by precisely controlling the positions of the robot's joints.

A. Dynamic Object Detection

Our detection methodology leverages the capabilities of event cameras while complementing them with depth data. This integration enhances the precision of 3D dynamic object detection, which is foundational for subsequent trajectory prediction and avoidance. Our detection process falls into two categories. In situations where the robot remains stationary, only moving objects stimulate event data. We then employ a filtering and clustering technique to fine-tune the 2D detection outcome. In scenarios where the robot is moving, motion compensation is required. These two categories can be distinguished by the motion of the event camera.

1) Motion Compensation: As the operation of event cameras is intricately tied to motion, our initial step involves the estimation of ego-motion to rectify and align the data generated by the event camera. This correction process is vital to ensure that subsequent detection accurately distinguishes the static and dynamic objects. In this work, 6 Degrees of Freedom (DoF) ego-motion is estimated based on [13]. Specifically, by integrating RGBD and event data, we formulate a direct method-based SLAM algorithm for 6 DoF pose estimation. Upon obtaining the pose, we project events from a $5\,\mathrm{ms}$ period to the current timestamp in preparation for the subsequent detection module. The motion compensation is done with the following equations:

$$e'_k = K\,T_k^t\,K^{-1} e_k \;:\; e_k \in E_{t-5ms,\;t} \tag{1}$$

where e_k is normalized event data within $5\,\mathrm{ms}$ time window, e'_k is the projected event data at timestamp t, K is the intrinsic parameters, and T_k^t is the interpolated transformation that takes event data's position from timestamp k to timestamp t.

2) Detection: Inspired by [11], we define the time-image as $\mathcal{T}$. Hence, $\mathcal{T}_{i,j}$ represents the timestamp information at pixel (i, j), which can be defined as the following equation:

$$\mathcal{T}_{i,j} = \frac{\sum(e_t - t_0)}{\mathcal{I}_{i,j} * (t_1 - t_0)} : e_t \in E_{i,j} \tag{2}$$

where $E_{i,j}$ represents all the events that are projected onto pixel (i, j). $\mathcal{I}_{i,j}$ is the number of events that are projected

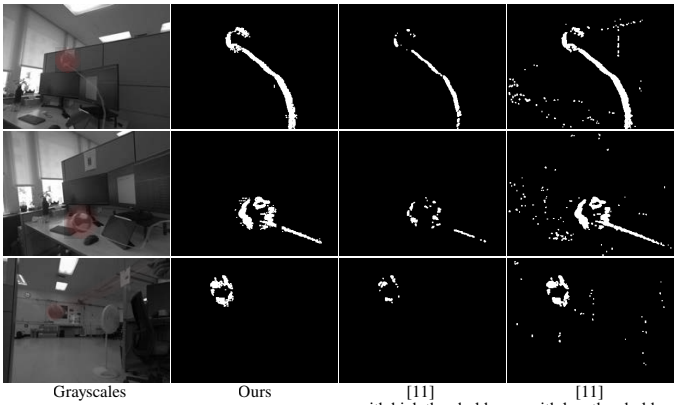

Fig. 3. Detection result comparison. The first column is the grayscale images from the RGBD camera, and dynamic objects are highlighted in red. The second column is the detection result from our method. The third and fourth columns are the detection results from [11] with a high and low threshold, respectively. Note that the first column of the RGB camera has a wider field of view than other columns from the event camera.

onto pixel (i, j). t_0 and t_1 are the begin and end timestamps of the event period E.

The given equation calculates pixel statistics within the range of 0 to 1. Dynamic objects typically approach 1. Selecting an appropriate threshold to differentiate pixels tied to dynamic objects can be intricate. A higher threshold will yield fewer detected pixels (third column in Fig. 3), while a lower threshold can introduce significant detection noise (fourth column in Fig. 3). To address this, we initiate with a mean filtering on $\mathcal{T}_{i,j}$ to minimize noise. Subsequently, we apply a comparatively high threshold to pinpoint potential regions for dynamic object detection (Eq. 3). We then adjust the threshold to be lower, ensuring enhanced detection results (Eq. 4). A comparative analysis between our method and the one from [11] is illustrated in Fig. 3.

$$\mathcal{R}_{i,j} = 1 : T_{i,j} > th_h \tag{3}$$

$$\begin{aligned} \mathcal{D}_{i+x,j+y} = 1 : T_{i+x,j+y} > th_l \\ i,j \in \mathcal{R}_{i,j} \quad x,y \in [-10,10] \end{aligned} \tag{4}$$

where th_h represents the higher threshold, th_l represents the lower threshold. $\mathcal{R}$ is the region of interest and $\mathcal{D}$ is the detection results. x, y are from the window around the regions of interest.

Upon obtaining the 2D detection outcome, we advance to determine the 3D location of the dynamic object, leveraging the capabilities of depth data at timestamp t. This 3D data is subsequently transformed from the camera's frame to the robot's frame. Conclusively, the derived 3D location is published via Lightweight Communications and Marshalling (LCM), making it accessible for the subsequent trajectory prediction module to utilize.

B. Trajectory Prediction

In this paper, we assume the robot initiates from a static position and encounters a singular dynamic object within its field of view. To compute the trajectory, we collect two detection results and apply a naive collision check by projecting the two detection outcomes onto the x-y plane in the world frame. Within this plane, the trajectory estimated by the two detection results appears as a straight line, and the robot is represented as a circle. Leveraging the formula to calculate the minimum distance from the center of a circle to a line, we establish a collision criterion, as articulated in Eq. 5.

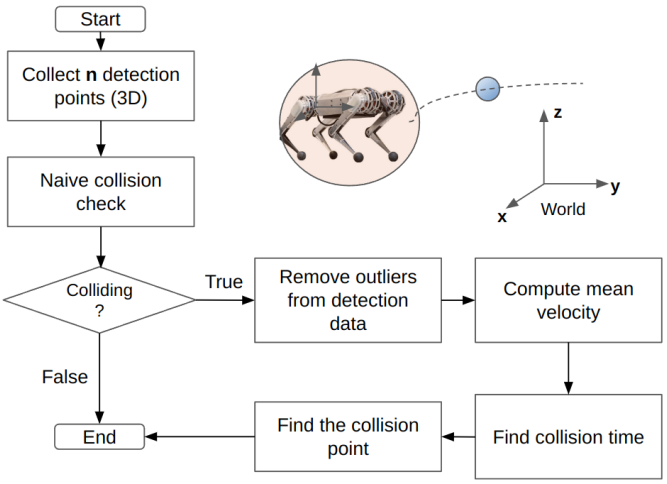

Fig. 4. Trajectory prediction and collision check.

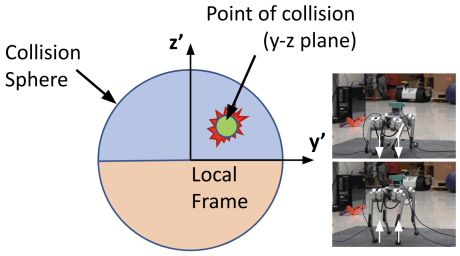

Fig. 5. Illustration of the coordinates frames and division of collision where to determine robot's motion direction.

$$R \geq \frac{ax + by + c}{\sqrt{a^2 + b^2}} \tag{5}$$

where R is the robot's radius and $ax + by + c = 0$ defines the object's trajectory in the x-y plane.

We gather a set of five detection results and compute the object's mean velocity using filtered data points if a potential collision is predicted. These filtered detection outcomes are then projected onto the y-z plane in the world frame. Utilizing a fourth-order polynomial equation derived from the 2D plane, we accurately pinpoint the potential collision point within the y-z plane. The point of the collision is illustrated in Fig. 5. Projecting the object's 3D trajectory onto the 2D plane serves as an efficient preliminary step for quickly eliminating non-collision scenarios. This strategy also allows us to forecast the dynamic object's position as it approaches the robot, offering valuable insights into the collision time and collision location.

TABLE I
DYNAMIC OBJECT AVOIDANCE EXPERIMENT RESULTS

	No. of Trial	*No. of Success*	*Success Rate*
Object from air	13	11	0.85
Object from ground	11	9	0.82
Total	24	20	0.83

C. Control

The control policy is based on the collision position within the collision circle. As depicted in Fig. 5, if the collision point falls in the circle's upper half, the robot decreases its height to evade the dynamic object. Conversely, if it's in the lower half, the robot raises its height to sidestep the obstacle. In this paper, we have prioritized height adjustment as the robot can change its height more rapidly than moving left or right. The decision is subsequently communicated to the low-level whole-body controller to adjust the robot's height.

III. EXPERIMENT

A. Setup

We evaluate the proposed algorithm on the Mini-Cheetah, assessing its capability to evade dynamic objects in both aerial and ground scenarios. Event and depth cameras are placed atop the robot. A visual representation is provided in Fig 5.

B. Result

As illustrated in Fig. 3, our detection strategy adopts a high threshold to identify regions of interest and a lower threshold to capture detailed detections. This approach not only surpasses the detection results from [11] with a high threshold (third column in Fig. 3) but also ensures minimal noise, unlike the results in the fourth column. It is worth highlighting that a thin string is tethered to the ball. In the first and second cases, our method adeptly identifies this string yet struggles in the third row. This detection challenge for the third case arises since the string's color closely matches the background, which will not trigger event data.

We conduct a series of ball-throwing experiments with the robot in a stationary position. Our primary aim is to validate the proficiency of our detection, trajectory prediction, and control modules within an indoor dynamic object avoidance setting. Notably, there is no preliminary information about the dynamic objects. These dynamic objects move at speeds of up to 5 meters and are launched from a distance of 8 meters from the robot. As shown in Table I, in the airborne trials, the robot successfully avoided the incoming ball in 11 of the 13 attempts, achieving an 85% success rate. In contrast, for the ground-based scenarios, the robot managed to successfully evade in 9 of the 11 tests, attaining an 82% success rate. Aggregating the results, the robot demonstrated an overall success rate of 83%. These results underscore the robustness of our algorithm in equipping the robot to proficiently avoid dynamic objects across two situations.

IV. CONCLUSION AND DISCUSSION

Our research introduces a comprehensive strategy leveraging event and depth data from a legged system to detect and avoid dynamic objects. Our approach includes an ego-motion compensation module for dynamic object detection and a 3D trajectory predictor, resulting in a success rate of over 80% in avoiding both airborne and ground dynamic objects, as demonstrated on the MIT Mini-cheetah. Our ongoing efforts will focus on refining the detection system to better compensate for more complex ego-motion. In addition, we will develop a more responsive controller that can adaptively avoid multi-dynamic obstacles. The adaptive behavior will enable the robot to adjust its height and orientation as needed.

ACKNOWLEDGMENT

We express our gratitude to Naver Labs and MIT Biomimetic Robotics Lab for providing the Mini-cheetah robot as a research platform for conducting dynamic motion studies on legged robots.

REFERENCES

[1] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation," *CoRR*, vol. abs/1702.03920, 2017. [Online]. Available: http://arxiv.org/abs/1702.03920

[2] M. Sorokin, J. Tan, C. K. Liu, and S. Ha, "Learning to navigate sidewalks in outdoor environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3906–3913, 2022.

[3] M. Seo, R. Gupta, Y. Zhu, A. Skoutnev, L. Sentis, and Y. Zhu, "Learning to walk by steering: Perceptive quadrupedal locomotion in dynamic environments," *arXiv preprint arXiv:2209.09233*, 2022.

[4] K. S. Sikand, S. Rabiee, A. Uccello, X. Xiao, G. Warnell, and J. Biswas, "Visual representation learning for preference-aware path planning," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 11 303–11 309.

[5] J. Carpentier and P.-B. Wieber, "Recent progress in legged robots locomotion control," *Current Robotics Reports*, vol. 2, no. 3, pp. 231–238, 2021.

[6] D. Kim, D. Carballo, J. Di Carlo, B. Katz, G. Bledt, B. Lim, and S. Kim, "Vision aided dynamic exploration of unstructured terrain with a small-scale quadruped robot," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2464–2470.

[7] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[8] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis *et al.*, "Event-based vision: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 1, pp. 154–180, 2020.

[9] A. Mondal, S. R, J. H. Giraldo, T. Bouwmans, and A. S. Chowdhury, "Moving object detection for event-based vision using graph spectral clustering," *CoRR*, vol. abs/2109.14979, 2021. [Online]. Available: https://arxiv.org/abs/2109.14979

[10] A. Mitrokhin, C. Fermüller, C. Parameshwara, and Y. Aloimonos, "Event-based moving object detection and tracking," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[11] D. Falanga, K. Kleber, and D. Scaramuzza, "Dynamic obstacle avoidance for quadrotors with event cameras," *Science Robotics*, vol. 5, no. 40, p. eaaz9712, 2020.

[12] B. He, H. Li, S. Wu, D. Wang, Z. Zhang, Q. Dong, C. Xu, and F. Gao, "Fast-dynamic-vision: Detection and tracking dynamic objects with event and depth sensing," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 3071–3078.

[13] S. Zhu, Z. Tang, M. Yang, E. Learned-Miller, and D. Kim, "Event camera-based visual odometry for dynamic motion tracking of a legged robot using adaptive time surface," *arXiv preprint arXiv:2305.08962*, 2023.