Fully Autonomous Neuromorphic Navigation and Dynamic Obstacle Avoidance

Xiaochen Shang[1], Pengwei Luo[1], Xinning Wang[1], Jiayue Zhao[1], Huilin Ge[2], Bo Dong[3], Xin Yang[1,*]
[1]Key Laboratory of Social Computing and Cognitive Intelligence
(Dalian University of Technology), Minstry of Education,
[2]Jiangsu University of Science and Technology, [3]Cephia AI
xcshang0614@gmail.com
xinyang@dult.edu.cn

Abstract

Unmanned aerial vehicles could accurately accomplish complex navigation and obstacle avoidance tasks under external control. However, enabling unmanned aerial vehicles (UAVs) to rely solely on onboard computation and sensing for real-time navigation and dynamic obstacle avoidance remains a significant challenge due to stringent latency and energy constraints. Inspired by the efficiency of biological systems, we propose a fully neuromorphic framework achieving end-to-end obstacle avoidance during navigation with an overall latency of just 2.3 milliseconds. Specifically, our bio-inspired approach enables accurate moving object detection and avoidance without requiring target recognition or trajectory computation. Additionally, we introduce the first monocular event-based pose correction dataset with over 50,000 paired and labeled event streams. We validate our system on an autonomous quadrotor using only onboard resources, demonstrating reliable navigation and avoidance of diverse obstacles moving at speeds up to 10 m/s under different light conditions, with energy consumption reduced to 21% compared to traditional architecture.

1 Introduction

The utilization of UAVs across various applications expanded rapidly over the past decade [1]. Currently, most UAVs rely heavily on external aids such as positioning systems like Global Positioning System (GPS) [2] for localization and ground stations [3] for navigation and dynamic obstacle avoidance. However, such external aid is not feasible in all circumstances as it could be easily jammed [4] or interfered in multiple scenarios, including dense urban areas [5], caves, or even war zones [6]. Therefore, it is vital for UAVs to fully perform navigation and dynamic obstacle avoidance tasks using only sensors and computing resources onboard, without any dependence on external signals or infrastructure. Although applicable options are well researched, most solutions are designed for larger platforms but not for tiny UAV systems [7]. Ranging sensors like the Li-DAR system could provide accurate positioning information, but are too heavy and power-hungry to be deployed on tiny autonomous systems [8]. Vision-based approach may be an appropriate way for tiny UAVs since, firstly, visual sensors can be both lightweight and power-efficient [9, 10, 11]. Secondly, visual algorithms achieve state-of-the-art performance in multiple tasks. However, such high performance comes at excessive computational and memory costs. Mainstream approaches like simultaneous localization and mapping (SLAM) algorithms [12] and object recognition-based trajectory estimation methods [13, 14, 15] consume hundreds of megabytes to several gigabytes of memory and hundreds of gigaflops [16]. Such high consumption makes tiny UAV autonomy challenging.

Neuromorphic hardwares provide a solution to this problem since the asynchronous and sparse nature of their biomimetic data format could exceed the current standard of energy efficiency, computational consumption, and task accuracy, and thus represents a paradigm shift compared to the traditional

[*]Correspongding Author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

computer vision approach [17, 18, 19]. Ideally, such data structure could lead to a data processing method with higher processing speed and lower energy consumption, but contemporary methods treat event data similar to a traditional image frame [20, 21] using "event frames" [22, 23, 24, 25], and hence fails to fully leverage the inherent sparsity of event streams [26], resulting in a performance similar to traditional methods. Inspired by the efficiency of biological systems, we observed frogs can accurately localize fast-moving insects while exhibiting significant disregard for stationary objects. Further anatomical analysis of their visual neural pathways revealed a striking similarity between the frog's visual imaging mechanism and the working principles of event cameras; hence, by leveraging this similarity, a purely neuromorphic dynamic obstacle avoidance approach—mimicking frog visual neurons becomes feasible.

In this paper, we exhibit a fully neuromorphic pipeline. With only one monocular event camera and an inertial measurement unit (IMU), the autonomous UAV could accomplish the navigation task and the dynamic obstacle avoidance task simultaneously, purely with its onboard computing resources without any external aid. In the navigation module, the quadrotor uses IMU data to navigate long distances, and by coupling the visual-homing algorithm and event data, the quadrotor employs an SCNN network to mitigate the effects of error drift. To the best of our knowledge, we construct the first monocular event-based pose correction dataset with 50,234 paired event streams, each labeled with its ground truth extrinsic obtained by a motion capture system. In the obstacle avoidance module, by implementing our bio-inspired algorithm, the quadrotor could suppress the events produced by static objects. With only events generated by dynamic objects preserved, the algorithm bypasses target recognition and trajectory computation steps, directly outputting evasion maneuvers, and reduces the latency of obstacle avoidance to only 2.3 milliseconds. The significantly reduced latency provides UAVs with a longer time window for evasion maneuvers, substantially enhancing their performance when encountering high-speed moving objects. The comparative evaluation with other state-of-the-art dynamic obstacle approaches demonstrates the superior performance of our neuromorphic architecture and bio-inspired algorithm. Additionally, we have validated the effectiveness and robustness of our approach in real-world environments through physical flight experiments under different light conditions, with energy consumption reduced to 21% compared to the traditional structure.

In summary, our contributions to the community include:

- A fully neuromorphic framework enabling tiny UAVs to rely solely on onboard computation and sensing for real-time navigation and dynamic obstacle avoidance.
- A bio-inspired approach enabling tiny UAVs to accurately avoid dynamic obstacles at speeds up to 10 m/s with a latency of 2.3 milliseconds.
- An open-sourced monocular event-based pose correction dataset with over 50,234 paired and labeled event streams.

2 Related Work

2.1 Neuromorphic Control of Quadrotors

While some studies discuss the topic of neuromorphic control on objects like larger robots and robot arms [27, 28, 29], the neuromorphic control system on quadrotors remains an underexplored area in research. A Viale et al. [30] proposed the first example of a neuromorphic vision-based controller solving a high-speed UAV control task by using a spiking neuronal network with an Intel Loihi chip [31]. Dupeyroux et al. [32] accomplished the task of UAV landing with a 3-layer spiking neuronal network on Loihi, and recently, Paredes-Vallés et al. proposed the first fully neuromorphic vision and control pipeline for controlling an autonomous quadrotor and made the quadrotor successfully take off, fly along a given route, and then land [33]. The study of neuromorphic control of quadrotors is highly restricted by the hardware performance of embedded neuromorphic processing platforms [34, 35] in terms of the number of available neurons and synapses. The Intel Kapoho Bay with 2 Loihi chips [31] carries 262,100 neurons [30], and the SpiNNaker(SNN architecture) version [36] has 768,000 neurons. Though higher-neuron neuromorphic platforms expand computational capacity, they remain inadequate for tasks like optical flow estimation (requiring > 3.7M neurons [37]). In this work, we use Speck [38], a neuromorphic SoC (System on Chip) with 327,000 neurons [39] that could support at most 8 layers of SNNs.

2.2 Visual-homing Algorithm

Visual-homing comes from the idea that small insects such as ants and bees can navigate long distances despite their tiny brains. The mechanism behind such behavior can be categorized into two parts: path integration and drift error elimination. Cartwright and Collet [40] first proposed a snapshot model that describes the homing behavior of bees, and researchers in the field of robotics use this concept to develop efficient navigation algorithms for tiny robots [41, 42]. Subsequent researches focus on reducing the memory required for visual-homing, and has been made in two directions. The first is the reduction of the memory consumed by snapshots: Stürzl and Mallot [43] transformed the snapshot into the frequency domain and remembered only the lowest-frequency component, and reduced the size of the snapshot remarkably. The second direction is to increase the spacing between snapshots. Denuelle and Srinivasan [44] proposed a study that uses the homing vector as a position estimate relative to the snapshot, enabling the drone to navigate some distance toward the next snapshot area. Van Dijk et al. [45] combined two directions and successfully deployed visual-homing on a tiny 56-gram autonomous drone with one panoramic camera. For detailed biological concepts, please take a look at the supplementary note 3.

2.3 Frog-eye Receptive Field

In nature, frogs' visual systems exhibit high-fidelity motion detection for fast-moving objects with deliberate suppression of static background stimuli. The observed motion selectivity stems from specialized receptive field organization in the anuran retinotectal system [46, 47]. During the past decades, researchers conducted extensive research on such a mechanism and found that R3 ganglion cells respond to stimuli to ON-OFF brightness changes, create motion-sensitive detection zones[48, 49, 50]. In the standard model of such detection zones, ERF (excitatory receptive field) and IRF (inhibitory receptive field) generate symmetrical excitatory and inhibitory responses to ON-OFF stimuli. Extending these findings, Hoshino et al.[51] identified a functional asymmetry in the spatial organization of ERF and IRF. For detailed biological concepts, please take a look at the supplementary note 3.

2.4 Dynamic Obstacle Avoidance

The dynamic obstacle avoidance problem for unmanned aerial vehicles has been widely researched in recent years, but mainly in the aspect of quasi-static environment[52] and low-speed obstacles. Even though existing literature that relies on monocular vision[53, 54, 55, 14], stereo vision[56, 57]and depth camera [13, 58, 59] exhibits satisfactory performance on slow-moving objects like pedestrians [60], their performance dealing with high-speed dynamic obstacles like thrown balls, birds[61] and even other unmanned aerial vehicles cannot meet the requirement of real-time avoidance. However, despite Falanga et al.[26] displaying the concept of using event stream directly, many researchers still treat events in the form of "event frame" [23]. To the best of our knowledge, this is the first work that implements low-latency (2.3 milliseconds) dynamic obstacle avoidance when the quadrotor is executing a navigation task without the help of any external infrastructure.

3 Methods

In this section, we introduce our neuromorphic navigation and dynamic obstacle avoidance pipeline, which includes a neuromorphic control framework that allocates computing resources to minimize evasion latency and maximize navigation correctness, an event-visual-homing based end-to-end method and a bio-inspired dynamic obstacle avoidance algorithm that reduces the latency ofobstacle detection to 2.3 milliseconds and applicable for dodging multiple high-speed obstacles when the drone is navigating to its destination.

3.1 Overview of Neuromorphic Control

The neuromorphic control framework is implemented on the Speck Neuromorphic SoC [38] and deployed on a small quadrotor for navigation and dynamic obstacle avoidance. The schematic of the quadrotor is illustrated in Fig. 2. In this framework, we assume the quadrotor first performs an outbound flight towards a designated target, which could be under any control law, including manual control, and then performs an inbound navigation and avoids multiple dynamic obstacles during this navigation in a fully autonomous fashion. Since our focus is on the navigation and dynamic obstacle avoidance during the inbound flight, we assume the outbound flight is performed without

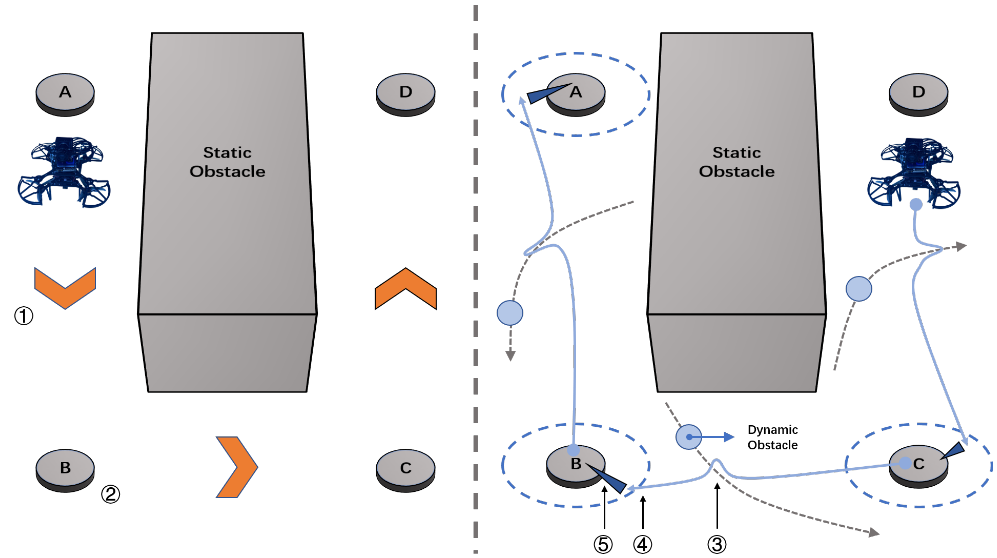


Figure 1: **Overview of the whole task.** During the outbound flight, which could be under any control law (1) and periodically records event stream (2). During the inbound flight, the quadrotor uses IMU information to travel to the location near next snapshot point and avoid any dynamic obstacles during its flight (3), and then records event stream continuously to recalibrate it's position (4) until the distance to the snapshot point is smaller than the threshold(5). These steps repeat until the quadrotor reaches its destination.

any collision, the environment is static (surroundings don't change), and no dynamic obstacles appear when the quadrotor is recalibrating the drift error of the IMU.

To minimize the obstacle avoidance latency under strict computational resource restriction [38] for longer response windows and higher success rates, we need to minimize the resources used for navigation. By introducing visual-homing, navigation during most of the flight is accomplished solely by odometry with negligible computational overhead, thereby reserving sufficient resources for obstacle avoidance. Moreover, both the calibration phase and obstacle avoidance module share the same monocular event camera, which not only reduces computational load but also significantly decreases the UAV's payload, ultimately enhancing its motion performance.

3.2 Event Visual-homing

IMU During the outbound flight, the quadrotor records all IMU information it produces without any correction. Generally, the IMU drift error stacks over time and will gradually become too large

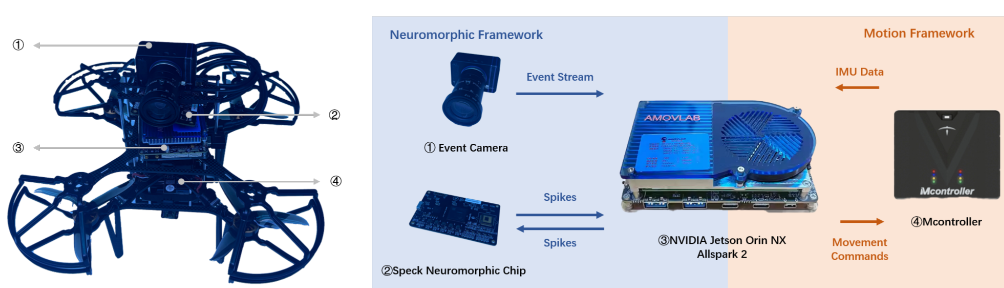


Figure 2: **Schematic of the neuromorphic quadrotor system** The left part is the quadrotor used in this work, total weight of 856 g; tip-to-tip diameter of 240 mm, with the numbers indicating the components in the right part. The right part is the hardware overview with the display of data flow, with components divided into two frameworks: the neuromorphic framework and the motion framework. One for processing neuromorphic data and the other one processing the movement control.

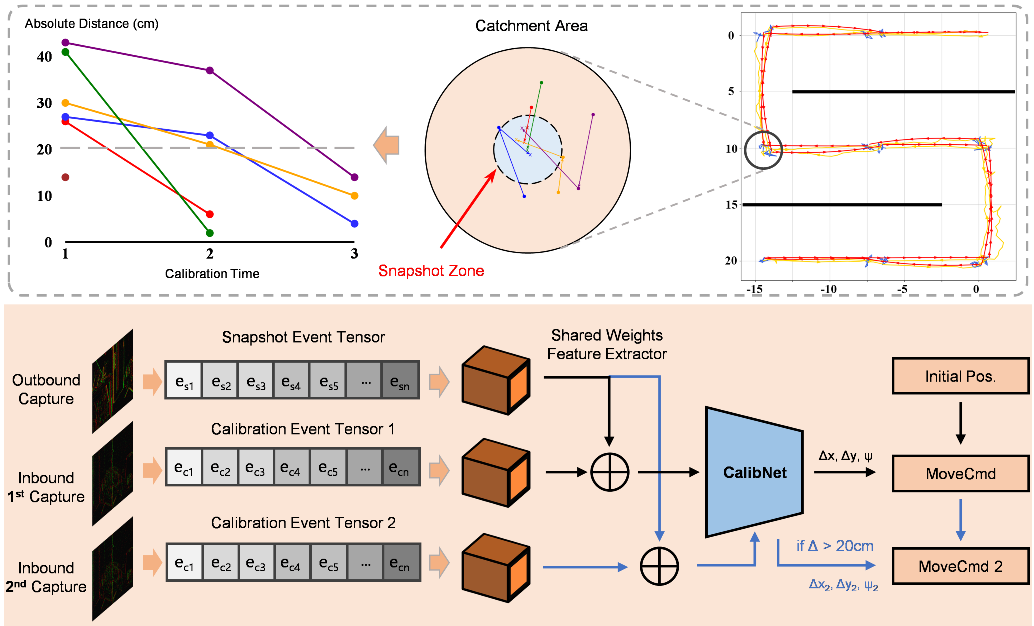


Figure 3: **Illustration of Visual-homing and CalibNet.**The quadrotor continuously calibrates itself in the catchment area until it reaches the snapshot zone. The catchment area is defined as a circular region with a 60cm radius, within which the quadrotor must remain positioned when initiating the calibration process. The snapshot zone constitutes a smaller 20cm-radius circular area centered within the catchment area; calibration terminates once the quadrotor enters this central zone.

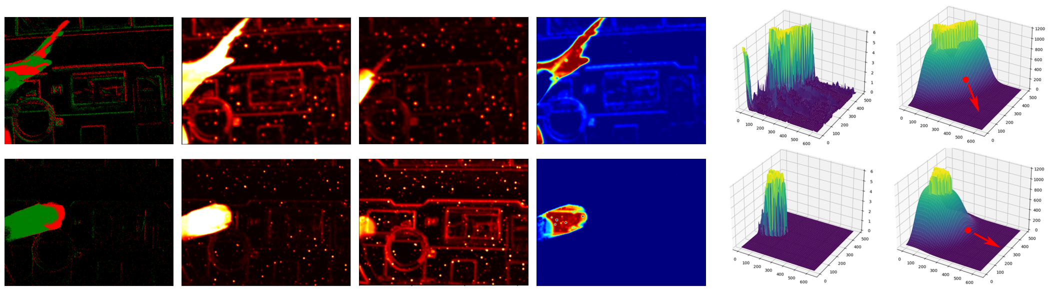
to provide applicable navigation information for the quadrotor[62]. A simplified version of the stack-over-time error can be defined as follows:

$$\delta r_N = \delta r_{N,0} + \delta v_{N,0,t} + \frac{1}{2}(g \cdot \delta\Theta_0 + b_{\alpha N})t^2 + \frac{1}{6}(g \cdot b_{gE})t^3 \tag{1}$$

where $\delta r_{N,0}$ is the initial position error, remains the same for all time; $\delta v_{N,0,t}$ is the initial velocity error with linear amplification; $g \cdot \delta\Theta_0$ denotes initial attitude angle error, $b_{\alpha N}$ denotes accelerometer error, and the third term exhibits quadratic divergence; $g \cdot b_{gE}$ denotes angular velocity error, and the last term exhibits cubic divergence. As shown in the formula, the IMU error exhibits approximately cubic drift over time. We let the quadrotor use a visual-homing algorithm to periodically return to snapshot positions, and recalibrate IMU drift error before the error aggregates too high. After the homing, the only error is the homing error during the recalibration process.

Event-based Drift Error Recalibration Network As we have shown in formula 1, we can periodically recalibrate the error caused by drift before it becomes too large, thereby keeping it at a relatively small level consistently. By quantitatively analyzing IMU error propagation, as shown in supplementary material 8, we can estimate the maximum potential position drift of the quadrotor, and set the calibration interval to a value that guarantees the quadrotor will remain within the catchment area. To fully utilize the advantage of event data and a neuromorphic framework, we use an SCNN (spiking convolutional neural network) with a Siamese structure [63]. The feature extractor of the network extracts features from two continuous event streams with a temporal time window of 50 ms. The first event stream is filmed at snapshot position during the outbound flight, and the second event stream is filmed near snapshot position during the inbound flight. Two feature tensors are then concatenated and passed to the calibration module. Finally, the network outputs a vector containing the relative x, y coordinate differences and the yaw angle difference between two captured points, as illustrated in Figure. 3.

To solve the scale issue which makes the network impossible to determine the absolute scale of object brought by the monocular event camera, we train the network using data obtained in a similar-scale environment and design a cyclic correction method where the UAV continues capturing event stream from the corrected position and performs repeated correction until the position error output by the network falls below a specific threshold. Training details are shown in supplementary note 6.



(A) Raw Event (B) ERF (C) IRF (D) Event RF (E) Raw PF (F) Diffused PF

Figure 4: **Illustration of the workflow of the Event RF Model and the potential field movement command generator.** The UAV avoids dynamic obstacles' high-energy zones via the gradient descent method. The red dot is the representation of the quadrotor and the length and direction of the red arrow represent the moving direction and speed of the quadrotor.

Dataset To address the absence of benchmark data for monocular event-based pose correction, we constructed a novel dataset containing 50,234 event stream pairs, each precisely annotated with 4-DoF relative pose $(\Delta x, \Delta y, \Delta z, \Delta\phi)$ ground truth. There are four distinct indoor scenarios contained in the dataset, and maximum object-camera proximity is constrained to a 10-meter range. The camera was mounted on a DJI Ronin SC gimbal ($\pm 0.02°$ stabilization accuracy) during shooting, which eliminated the influence of pitch and roll angles while simulating the stabilized attitude of a drone equipped with flight controllers. The ground truth of camera's shooting position is obtained by a motion capture system with 12 Vicon Vero 2.2 motion capture camera, each featuring with a resolution of 2048 x 1088 and a max frame rate of 330 Hz.

3.3 Bio-inspired Dynamic Obstacle Avoidance

Event Receptive Field Model The brightness-sensitive biological mechanism behind anuran ganglion cells exhibits isomorphic correspondence with event-based vision sensing. By leveraging the ERF-IRF spatial asymmetry, we proposed an Event RF (receptive field) model, used for suppressing the event stream produced by static objects and background, and enhancing the event stream produced by dynamic obstacles:

$$F(x, y, e_n, t) = min(A_1 K(t, \tau_e) G(x, y, e_n), E_{th}) - min(A_2 K(t - \Delta t, \tau_i) G(x, y, e_n), I_{th}) \quad (2)$$

where A_1 is ERF parameter, A_2 is IRF parameter, τ is energy decay parameter, Δt is IRF delay parameter, and $K(t, \tau)$ is time kernel function:

$$K(t, \tau) = \begin{cases} \mathrm{e}^{-t/\tau} & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (3)$$

and the first term $min(A_1 K(t, \tau_e) G(x, y, e_n), E_{th})$ is the ERF energy level, while E_{th} is ERF energy threshold, the second term $min(A_2 K(t - \Delta t, \tau_i) G(x, y, e_n), I_{th})$ is the IRF suppression level, while I_{th} is the IRF suppression threshold.

e_n is the event passed to the model, with its coordinates and timestamp as (x_n, y_n, t_n). $G(x, y, e_n)$ is the Gaussian kernel function:

$$G(x, y, e_n) = \frac{1}{2\pi\sigma_x\sigma_y} exp(-\frac{(x - x_n)^2}{2\sigma_x^2} - \frac{(y - y_n)^2}{2\sigma_y^2}) \quad (4)$$

where σ_x and σ_y are standard deviations along the major and minor axes of the 2D elliptical Gaussian function.In this model, the event stream from static objects is quickly suppressed by the IRF, which drives the energy level close to zero. In contrast, the event stream produced by moving objects resists suppression, allowing it to maintain a high energy level persistently, as shown in Fig. 4A–4D. A complete mathematical proof is provided in supplementary note 2.

Potential Field Based Movement Command Generator We proposed a potential field-based method to generate movement commands from the processed event stream obtained from the Event RF Model. By converting the energy map directly to the activation map, we can consider the event camera's field of view as a 2-dimensional plane and construct potential on this plane based on the energy level of the event stream, as shown in Fig. 4E. After removing the points with excessively low potential in this potential field (here we set the threshold as half of the maximum potential), we can consider that the potential field fully represents the moving obstacles within the event camera's field of view, as shown in Fig. 4F.

Since we are using a monocular event camera as the only sensor to capture the dynamic obstacle, the depth information of the dynamic obstacle cannot be obtained, which means obstacles far from the quadrotor can also be considered as dangerous objects that need to be avoided. To solve this, we use the Two-Pass Algorithm to make a connected component detection. Neglecting the potential level of points, we consider points with potential as 1 and points without potential as 0, and convert the map to a binary image I:

$$I = \{I(i,j)|I(i,j) \in \{0,1\}, 1 \leq i \leq H, 1 \leq j \leq W\} \tag{5}$$

By making the connected component detection, we can assess the danger level of dynamic obstacles based on the proportion of their potential field regions occupying the entire field of view. For those dangerous potential field clusters, we define a dilation function:

$$(I \oplus B)(x,y) = max \quad I(x+i)(y+j) \quad for \quad (i,j) \in B \tag{6}$$

where $\oplus$ is the symbol of dilation operation, (x, y) is the coordinate of the point in the plane surface, (i, j) is the offset in the structuring element B.

After the dilation process, since we can consider the position of the quadrotor in the center of the potential field map, we can now determine the motion direction and motion intensity of the quadrotor using gradient descent in the artificial potential field.

4 Experiments

4.1 Simulation Experiments

Before combining visual navigation and dynamic obstacle avoidance into a single neuromorphic system, we conducted separate experiments to verify the effectiveness of each part. The whole simulation experiment is in the Gazebo simulation environment.

Visual-homing Navigation We trained the Siamese Network using the dataset we constructed, and we test the navigation process in three different customized maps, with difficulty from low to high. The flight distance is 40 meters for the easiest map and 130 meters for the other two maps.

Fig. 3 shows the resulting trajectories for the proposed method. The quadrotor successfully and steadily followed the route of outbound flight and reached the starting point. Based on the calculated drift error propagation, we set the snapshot interval at 7.5 meters, with each snapshot occupying 240 KB of storage space. We conducted repeated experiments to analyze the propagation degrees of X-axis translations, Y-axis translations, and yaw angle errors. By obtaining these data, we ensured that under our setting of calibration interval, the quadrotor's positional drift remains strictly bounded within the designated 60cm-radius catchment area. Details on IMU error analysis can be found in supplementary note 8. We test the navigation procedure 10 times in each map, and in every single test, the quadrotor successfully reaches the destination.

Dynamic Obstacle Avoidance We use ESIM [51], an event camera simulator in Gazebo, to simulate event camera imaging effects for the quadrotor and conducted 300 dynamic obstacle avoidance tests. The dynamic obstacles were categorized into three groups based on size: coin-sized, tennis ball-sized, and basketball-sized. Each group was tested 25 times at four different distance ranges: within 0.2 - 0.5m, 0.5-1m, 1-2m, and beyond 2m. We set the closest starting distance of obstacles at 0.2m since firstly, if the obstacle is too close to the quadrotor, the entire field of view will be occupied by the dynamic obstacles and the algorithm cannot make effective obstacle avoidance commands, and secondly, in real-world scenarios, it is generally impossible for dynamic to abruptly appear within the drone's immediate proximity.

For each obstacle detection, we also marked its centroid in the image frame and compared it with the centroid of the algorithm-processed event stream to validate the position error in dynamic obstacle detection, as shown in Table 1. Details about the calculation are provided in supplementary note 4.

To quantify the computational cost of the model, we recorded multiple event streams of dynamic obstacles and processed these event streams with our algorithm to calculate the processing time to evaluate the delay of our model. Since the model relies on generating IRF fields from prior processed events and applying decay on both ERF and IRF fields, biased results inevitably arise when processing arbitrarily cropped sections of the raw event stream. We use our algorithm to process the whole event stream and compute the ratio between the processing time and the total length of the event stream to obtain the unbiased average latency of 2.3 ms. Detailed data are shown in supplementary note 11.

Table 1: **Centroid Difference Between RF Model and GT (m)**

Obstacle Type	Distance	Mean	Median	Std. Dev.	M.A.D	SR
Coin-sized	0.2m - 0.5m	0.0179	0.0164	0.0089	0.0015	94%
	0.5m - 1m	0.0177	0.0134	0.0094	0.0008	92%
	1m - 2m	0.0138	0.0126	0.0077	0.0202	90%
	2m+	0.0132	0.0123	0.0053	0.0042	86%
Tennis-sized	0.2m - 0.5m	0.0213	0.0228	0.0093	0.0052	92%
	0.5m - 1m	0.0180	0.0169	0.0072	0.0093	98%
	1m - 2m	0.0168	0.0146	0.0095	0.0002	100%
	2m+	0.0129	0.0102	0.0103	0.0057	96%
Basketball-sized	0.2m - 0.5m	0.0306	0.0297	0.0141	0.0017	84%
	0.5m - 1m	0.0271	0.0261	0.0135	0.0026	96%
	1m - 2m	0.0196	0.0196	0.0059	0.0081	100%
	2m+	0.0222	0.0169	0.0166	0.0109	100%

Combined Task Simulation With the core algorithms proven, we then demonstrated the complete pipeline by combining the complex tasks together. Using the 3 maps we created in Gazebo (mentioned in 4.1.1), we made the quadrotor traverse the outbound route using odometry without any global position information and added randomly throwing dynamic obstacles when the quadrotor is flying during its inbound journey. Among 50 tests on each map, the success rate is 100% for the first map, 98% for the second map, and 94% for the last map. Figure demonstrations and other details are shown in supplementary note 7.

4.2 Real-world Experiments

In this section, we conduct indoor experiments using our neuromorphic platform mentioned in section 3.1, and also conduct extra indoor experiments under different extreme conditions (flicker condition and darkish condition) to validate the algorithm's robustness.

Indoor Experiment As previously mentioned, the main goal of the indoor experiment is to verify the effectiveness of our neuromorphic framework in a real-world setup and test our neuromorphic structure's advantage in computational resource consumption, energy consumption, and verify the performance of the framework with such low consumption in a tiny autonomous quadrotor. The experiment is conducted in a 10m * 10m flight arena. Three experimenters stationed at designated locations threw dynamic obstacles at passing drones, and experimenters were instructed to remain stationary to prevent the quadrotor from misidentifying them as dynamic obstacles. During 10 repeated trials, the quadrotor successfully avoided all dynamic obstacles and reached the destination in every instance, as shown in Fig. 5. Details about obstacles are shown in supplementary note 5.

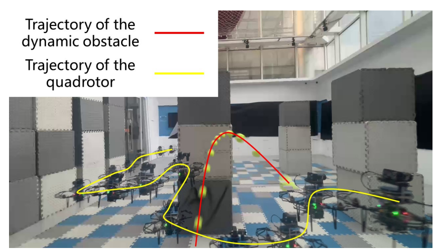


Figure 5: **Real-world Experiment** The quadrotor avoids the tennis ball thrown by experimenters during navigation.

Complex Environment Experiments We conducted additional experiments in both outdoor (a square with static boxes) and cluttered indoor environments (office corridors), testing three dynamic obstacles: thrown objects, sparse crowds, and dense crowds. Results show consistent performance across environments, with outdoor lighting/airflow variations causing no significant impact. The system maintained high navigation success rates for thrown objects and sparse crowds, with performance degrading only in extreme crowd densities.

Table 2: **Quantitative evaluation on Outdoor Environments and Cluttered Indoor Environments**

Scenario	Obstacle Type	SR
Office building corridor	Thrown objects	94.6%
	Sparse pedestrians	93.8%
	Dense pedestrians	62.1%
Outdorr square	Thrown objects	94.1%
	Sparse pedestrians	93.7%
	Dense pedestrians	54.8%

Reduced Energy Consumption on Neuromorphic Hardware We tested the energy consumption and run time between different setups, and a main observation is that the neuromorphic chip demonstrates a two orders of magnitude reduction in power consumption compared to conventional devices. Systems equipped with this neuromorphic chip achieve a 95% reduction in operational energy consumption (down to 5% of original levels) when executing identical tasks using the same algorithms. The total system energy consumption decreases to 21% of baseline values. Notably, in neuromorphic systems, the primary energy expenditure originates from three core processes: the onboard computer operations, data exchange between the neuromorphic chip and flight controller, and motion command execution. Details about the energy consumption of each architecture are displayed in supplementary note 10.

Robustness Under Extreme Light Condition Despite our main goal is to validate the advantage of our bio-inspired algorithm and fully neuromorphic framework, to better demonstrate the effectiveness of the unique event stream data modality, we need to test the framework under extreme light conditions to prove its robustness. We test the flight performance in the same arena under three different light conditions: light (10 - 100 lux), flicker (1 - 100 lux), dim (1 - 10 lux), and dark (0 - 1 lux). The result shows that the performance of the quadrotor is approximately the same under different light conditions, but it does not work in dark conditions. Experiment details are shown in supplementary note 9.

4.3 Comparison and Analysis

Comparison with the State-of-the-Arts Since this work is the first to implement a fully neuromorphic pipeline on complex navigation and dynamic obstacle avoidance tasks, to provide a reference level, we compare our Event RF model to some related traditional approaches based on object recognition and trajectory estimation[64, 13, 26, 65], as shown in Table 3. These approaches are tested under the same simulation environment using their official codes. Since Li-DAR can get precise position information of obstacles, there is no position error for the Li-DAR method. Among all the works, our system achieves the lowest latency, significantly lower than other types of sensors. The only approach with comparable latency is the method 3 [26], which also employs an event camera but lacks a navigation module, thus allocating all computational resources to obstacle avoidance. Since our method does not perform object recognition or trajectory estimation, we cannot compare prediction speed errors. However, in terms of positional error, our work also achieves the lowest. Regarding obstacle avoidance success rate, our performance is very close to the best, with only a 2% gap.

Table 3: **Quantitative evaluation on Dynamic Obstacle Avoidance Task** Only Method 4 utilized non-visual sensors, and experiments that did not employ onboard computational resources were specifically marked.

Method	Latency	Pos. Err.	SR
Method 1 [64]	19.12 ms	0.11 m	**96.3**
Method 2 [13]	39.49 ms	0.11 m	89.1
Method 3 [26]	3.56 ms	0.09 m	86.7
Method 4 (GTX 4090) [65]	14 ms	LiDAR	95.75
Method 4 (onboard) [65]	27 ms	LiDAR	86.5
Ours	**2.34 ms**	**0.02 m**	94.5

Analysis of Event RF Model We delve into the parameter choosing for the Event RF model and conduct simulation experiments on multiple dynamic obstacles of different sizes to evaluate the effect of parameter selection on performance and the generalizability of the parameters, as shown in Table. 4. There are 3 pairs of parameter in Event RF model: $(A_1, A_2), (\tau_e, \tau_i), (E_{th}, I_{th})$, and 3 separate parameters: $\Delta t, \sigma_x, \sigma_y$, and the value of each parameter significantly affects the model's performance.

Table 4: **Model Performance with Fixed Parameters vs. Obstacles at Various Speeds** While the model exhibits general robustness across a range of velocities, task-specific parameter tuning can yield superior performance in dedicated scenarios.

Type	SR
Low-speed (2m/s)	100%
High-speed (8m/s)	100%
Ultra-high-speed (15m/s)	70%
Ultra-high-speed (specified parameters)	70%

σ_x, σ_y, as the standard deviation along the major and minor axes of the 2D elliptical Gaussian function, affects the size of the receptive field generated by each event. Under perfect motion compensation, $\sigma_x = \sigma_y = 1$ makes the IRF just sufficient to suppress the stimulation caused by the ERF. However, considering the limited computational resources, achieving perfect motion compensation is challenging, along with the inherent noise introduced by the event camera itself. Setting $\sigma_x = \sigma_y = 2$ could achieve a better result. Making σ_x and σ_y unequal could enable the model to exhibit anisotropy, reducing sensitivity to motion in specific directions, especially when setting different σ_x and σ_y for ERF and IRF individually.

Δt affects the delay of the IRF relative to the ERF. Higher Δt increases the size of high-energy regions for dynamic objects in the model, thereby increasing the distance between the centroid of the real obstacle and the centroid of the dynamic obstacle in the model, resulting in greater error. However, if Δt is too small, the ERF will be rapidly overridden by the IRF, thereby reducing the model's sensitivity to slow-moving objects. In experiments, we find $\Delta t = 5$ ms delivers optimal performance, and this value is suitable for the vast majority of dynamic obstacle avoidance scenarios.

Through mathematical derivation, we found that the model achieves optimal performance when $\frac{A_1}{A_2} = e^{-\Delta t/\tau_i}$, the model reaches optimal performance, and since $\tau_e < \tau_i$, we can determine the values of A_1, A_2 and τ_e based on the value of τ_i. In biological systems, the value of τ_i typically ranges from 25 to 50 ms. We conducted tests at 5-millisecond (ms) intervals and found the best value as 25 ms. Therefore, $\frac{A_1}{A_2}$ should be 1.22, and when $\tau_e = 5$ ms we get the best result. Under ideal conditions, setting $E_{th} = I_{th}$ would enable perfect static event cancellation. However, in practice, sensor noise and firing threshold fluctuations in biological neurons necessitate permitting minor deviations to prevent noise-induced false dynamic responses; here we choose $\frac{I_{th}}{E_{th}} = 1.2$ based on our experimental testing. Details on the analysis process can be found in supplementary note 12.

5 Conclusion

This paper presents a fully autonomous neuromorphic navigation and dynamic obstacle avoidance pipeline for tiny autonomous unmanned aerial vehicles. Its Event RF Model is the first bio-inspired algorithm that could make the quadrotor bypass the object recognition and trajectory estimation processes, thus avoiding dynamic obstacles in a real-time manner. By reducing the latency to 2.3 ms, the model gives a much longer reaction time window for the quadrotor when facing dynamic obstacles with speeds up to 10m/s.Comparative evaluations under identical experimental conditions prove our neuromorphic approach outperforms current state-of-the-art solutions for autonomous UAVs, delivering significantly lower latency in high-velocity dynamic obstacle avoidance while maintaining comparable success rates under stringent onboard computational constraints. The dataset presented in our work also establishes a solid foundation for further research on event-based pose calibration. Moreover, with reduced energy consumption and robustness under various light conditions, this work presents a substantial step toward neuromorphic sensing and controlling for UAVs, and exhibits the great potential of neuromorphic architecture on tiny autonomous robots, revealing the possibility of tiny autonomous robots to evolve to higher levels of operational capability and performance.

Broader Impacts and Safeguards While this work on autonomous drones aims to benefit applications like search and rescue in GPS-denied environments, we acknowledge its dual-use potential. To mitigate risks such as privacy invasion and malicious payload delivery, our approach integrates key safeguards. Primarily, the use of an event camera—which captures only illumination changes rather than identifiable imagery—provides an inherent layer of privacy protection by design. Furthermore, our open-source license and code documentation explicitly prohibit harmful misuse. These measures help ensure the technology's responsible development and deployment.

Limitation and Future Work The current work relies on IMU information for navigation, and the monocular event camera could not obtain depth information of the dynamic obstacle. Future work will further explore the Event RF Model's capabilities by leveraging its ability to distinguish between dynamic and static objects. A stereo vision setup will be used to explore the possibility of tiny autonomous neuromorphic quadrotors exploring and avoiding dynamic obstacles in completely unfamiliar environments without relying on any prior information.

Acknowledgments and Disclosure of Funding

The study has been supported by National Key Research and Development Program of China (No.2022ZD0210500).

References

[1] Dan Cireşan, Ueli Meier, Jonathan Masci, and Jürgen Schmidhuber. A committee of neural networks for traffic sign classification. In *The 2011 international joint conference on neural networks*, pages 1918–1921. IEEE, 2011.

[2] Kimon P Valavanis and George J Vachtsevanos. *Handbook of unmanned aerial vehicles*. Springer Publishing Company, Incorporated, 2014.

[3] Xiao Liang, Shirou Zhao, Guodong Chen, Guanglei Meng, and Yu Wang. Design and development of ground station for uav/ugv heterogeneous collaborative system. *Ain Shams Engineering Journal*, 12(4):3879–3889, 2021.

[4] Ali Krayani, Atm Shafiul Alam, Lucio Marcenaro, Arumugam Nallanathan, and Carlo Regazzoni. Automatic jamming signal classification in cognitive uav radios. *IEEE Transactions on Vehicular Technology*, 71(12):12972–12988, 2022.

[5] Nour El-Din Safwat, Roberto Sabatini, Alessandro Gardi, Ismail Mohamed Hafez, and Fatma Newagy. Urban air mobility communication performance considering cochannel interference. *IEEE Transactions on Aerospace and Electronic Systems*, 60(4):5089–5100, 2024.

[6] Roman Horbyk. "the war phone": mobile communication on the frontline in eastern ukraine. *Digital War*, 3(1):9–24, 2022.

[7] Sawyer B Fuller. Four wings: An insect-sized aerial robot with steering ability and payload capacity for autonomy. *IEEE Robotics and Automation Letters*, 4(2):570–577, 2019.

[8] Thinal Raj, Fazida Hanim Hashim, Aqilah Baseri Huddin, Mohd Faisal Ibrahim, and Aini Hussain. A survey on lidar scanning mechanisms. *Electronics*, 9(5):741, 2020.

[9] Sivakumar Balasubramanian, Yogesh M Chukewad, Johannes M James, Geoffrey L Barrows, and Sawyer B Fuller. An insect-sized robot that uses a custom-built onboard camera and a neural network to classify and respond to visual input. In *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*, pages 1297–1302. IEEE, 2018.

[10] Stéphane Viollet, Stéphanie Godiot, Robert Leitel, Wolfgang Buss, Patrick Breugnon, Mohsine Menouni, Raphaël Juston, Fabien Expert, Fabien Colonnier, Géraud L'Eplattenier, et al. Hardware architecture and cutting-edge assembly process of a tiny curved compound eye. *Sensors*, 14(11):21702–21721, 2014.

[11] Jiqing Zhang, Malu Zhang, Yuanchen Wang, Qianhui Liu, Baocai Yin, Haizhou Li, and Xin Yang. Spiking neural networks with adaptive membrane time constant for event-based tracking. *IEEE Transactions on Image Processing*, 2025.

[12] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[13] Zhefan Xu, Xiaoyang Zhan, Baihan Chen, Yumeng Xiu, Chenhao Yang, and Kenji Shimada. A real-time dynamic obstacle tracking and mapping system for uav navigation and collision avoidance with an rgb-d camera. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10645–10651. IEEE, 2023.

[14] Minghao Lu, Han Chen, and Peng Lu. Perception and avoidance of multiple small fast moving objects for quadrotors with only low-cost rgbd camera. *IEEE Robotics and Automation Letters*, 7(4):11657–11664, 2022.

[15] Jiqing Zhang, Bo Dong, Yingkai Fu, Yuanchen Wang, Xiaopeng Wei, Baocai Yin, and Xin Yang. A universal event-based plug-in module for visual object tracking in degraded conditions. *International Journal of Computer Vision*, 132(5):1857–1879, 2024.

[16] Bruno Bodin, Harry Wagstaff, Sajad Saecdi, Luigi Nardi, Emanuele Vespa, John Mawer, Andy Nisbet, Mikel Luján, Steve Furber, Andrew J Davison, et al. Slambench2: Multi-objective head-to-head benchmarking for visual slam. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3637–3644. IEEE, 2018.

[17] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2022.

[18] Yang Wang, Bo Dong, Yuji Zhang, Yunduo Zhou, Haiyang Mei, Ziqi Wei, and Xin Yang. Event-enhanced multi-modal spiking neural network for dynamic obstacle avoidance. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 3138–3148, 2023.

[19] Jianchuan Ding, Bo Dong, Felix Heide, Yufei Ding, Yunduo Zhou, Baocai Yin, and Xin Yang. Biologically inspired dynamic thresholds for spiking neural networks. *Advances in neural information processing systems*, 35:6090–6103, 2022.

[20] Alex Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. In *Robotics: Science and Systems XIV*. Robotics: Science and Systems Foundation, June 2018.

[21] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 989–997, 2019.

[22] Minggui Teng, Chu Zhou, Hanyue Lou, and Boxin Shi. Nest: Neural event stack for event-based image enhancement. In *European Conference on Computer Vision (ECCV)*, pages 660–676. Springer, 2022.

[23] Botao He, Haojia Li, Siyuan Wu, Dong Wang, Zhiwei Zhang, Qianli Dong, Chao Xu, and Fei 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)*, pages 3071–3078. IEEE, 2021.

[24] Haiwei Zhang, Jiqing Zhang, Bo Dong, Pieter Peers, Wenwei Wu, Xiaopeng Wei, Felix Heide, and Xin Yang. In the blink of an eye: Event-based emotion recognition. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023.

[25] Jiqing Zhang, Yuanchen Wang, Wenxi Liu, Meng Li, Jinpeng Bai, Baocai Yin, and Xin Yang. Frame-event alignment and fusion network for high frame rate tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9781–9790, 2023.

[26] Davide Falanga, Kevin Kleber, and Davide Scaramuzza. Dynamic obstacle avoidance for quadrotors with event cameras. *Science Robotics*, 5(40):eaaz9712, 2020.

[27] Travis DeWolf, Kinjal Patel, Pawel Jaworski, Roxana Leontie, Joe Hays, and Chris Eliasmith. Neuromorphic control of a simulated 7-dof arm using loihi. *Neuromorphic Computing and Engineering*, 3(1):014007, 2023.

[28] Ioannis Polykretis, Lazar Supic, and Andreea Danielescu. Bioinspired smooth neuromorphic control for robotic arms. *Neuromorphic Computing and Engineering*, 3(1):014013, 2023.

[29] Abdulla Ayyad, Mohamad Halwani, Dewald Swart, Rajkumar Muthusamy, Fahad Almaskari, and Yahya Zweiri. Neuromorphic vision based control for the precise positioning of robotic drilling systems. *Robotics and Computer-Integrated Manufacturing*, 79:102419, 2023.

[30] Antonio Vitale, Alpha Renner, Celine Nauer, Davide Scaramuzza, and Yulia Sandamirskaya. Event-driven vision and control for uavs on a neuromorphic chip. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 103–109. IEEE, 2021.

[31] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018.

[32] Julien Dupeyroux, Jesse J Hagenaars, Federico Paredes-Vallés, and Guido CHE de Croon. Neuromorphic control for optic-flow-based landing of mavs using the loihi processor. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 96–102. IEEE, 2021.

[33] Federico Paredes-Vallés, Jesse J Hagenaars, Julien Dupeyroux, Stein Stroobants, Yingfu Xu, and Guido CHE de Croon. Fully neuromorphic vision and control for autonomous drone flight. *Science Robotics*, 9(90):eadi0591, 2024.

[34] Le Zhu, Michael Mangan, and Barbara Webb. Neuromorphic sequence learning with an event camera on routes through vegetation. *Science Robotics*, 8(82):eadg3679, 2023.

[35] Ning Qiao, Hesham Mostafa, Federico Corradi, Marc Osswald, Fabio Stefanini, Dora Sumislawska, and Giacomo Indiveri. A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses. *Frontiers in neuroscience*, 9:141, 2015.

[36] Francesco Galluppi, Christian Denk, Matthias C Meiner, Terrence C Stewart, Luis A Plana, Chris Eliasmith, Steve Furber, and Jörg Conradt. Event-based neural computing on an autonomous mobile platform. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 2862–2867. IEEE, 2014.

[37] Jesse Hagenaars, Federico Paredes-Vallés, and Guido De Croon. Self-supervised learning of event-based optical flow with spiking neural networks. *Advances in Neural Information Processing Systems*, 34:7167–7179, 2021.

[38] Man Yao, Ole Richter, Guangshe Zhao, Ning Qiao, Yannan Xing, Dingheng Wang, Tianxiang Hu, Wei Fang, Tugba Demirci, Michele De Marchi, et al. Spike-based dynamic computing with asynchronous sensing-computing neuromorphic chip. *Nature Communications*, 15(1):4464, 2024.

[39] Ole Richter, Yannan Xing, Michele De Marchi, Carsten Nielsen, Merkourios Katsimpris, Roberto Cattaneo, Yudi Ren, Yalun Hu, Qian Liu, Sadique Sheik, et al. Speck: A smart event-based vision sensor with a low latency 327k neuron convolutional neuronal network processing pipeline. *arXiv preprint arXiv:2304.06793*, 2023.

[40] BA Cartwright and TS Collett. How honey bees use landmarks to guide their return to a food source. *Nature*, 295(5850):560–564, 1982.

[41] Aymeric Denuelle and Mandyam V Srinivasan. Bio-inspired visual guidance: From insect homing to uas navigation. In *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 326–332. IEEE, 2015.

[42] Julien Dupeyroux, Julien R Serres, and Stéphane Viollet. Antbot: A six-legged walking robot able to home like desert ants in outdoor environments. *Science Robotics*, 4(27):eaau0307, 2019.

[43] Wolfgang Stürzl and Hanspeter A Mallot. Efficient visual homing based on fourier transformed panoramic images. *Robotics and Autonomous Systems*, 54(4):300–313, 2006.

[44] Aymeric Denuelle and Mandyam V Srinivasan. A sparse snapshot-based navigation strategy for uas guidance in natural environments. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3455–3462. IEEE, 2016.

[45] Tom van Dijk, Christophe De Wagter, and Guido CHE de Croon. Visual route following for tiny autonomous robots. *Science Robotics*, 9(92):eadk0310, 2024.

[46] Frédéric Gaillard, Michael A Arbib, Fernando J Corbacho, and Hyun Bong Lee. Modeling the physiological responses of anuran r3 ganglion cells. *Vision research*, 38(17):2551–2568, 1998.

[47] Younginha Jung, Sungmoo Lee, Jun Kyu Rhee, Chae-Eun Lee, Bradley J Baker, and Yoon-Kyu Song. Imaging electrical activity of retinal ganglion cells with fluorescent voltage and calcium indicator proteins in retinal degenerative rd1 blind mice. *bioRxiv*, pages 2023–12, 2023.

[48] Jerome Y Lettvin, Humberto R Maturana, Warren S McCulloch, and Walter H Pitts. What the frog's eye tells the frog's brain. *Proceedings of the IRE*, 47(11):1940–1951, 2007.

[49] Horace B Barlow et al. Possible principles underlying the transformation of sensory messages. *Sensory communication*, 1(01):217–233, 1961.

[50] Kristian Donner and Carola AM Yovanovich. A frog's eye view: Foundational revelations and future promises. In *Seminars in Cell & Developmental Biology*, volume 106, pages 72–85. Elsevier, 2020.

[51] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *Conference on robot learning (CoRL)*, pages 969–982. PMLR, 2018.

[52] Xin Zhou, Zhepei Wang, Hongkai Ye, Chao Xu, and Fei Gao. Ego-planner: An esdf-free gradient-based local planner for quadrotors. *IEEE Robotics and Automation Letters*, 6(2):478–485, 2020.

[53] Omid Esrafilian and Hamid D Taghirad. Autonomous flight and obstacle avoidance of a quadrotor by monocular slam. In *2016 4th International Conference on Robotics and Mechatronics (ICROM)*, pages 240–245. IEEE, 2016.

[54] Yi Lin, Fei Gao, Tong Qin, Wenliang Gao, Tianbo Liu, William Wu, Zhenfei Yang, and Shaojie Shen. Autonomous aerial navigation using monocular visual-inertial fusion. *Journal of Field Robotics*, 35(1):23–51, 2018.

[55] Thomas Eppenberger, Gianluca Cesari, Marcin Dymczyk, Roland Siegwart, and Renaud Dubé. Leveraging stereo-camera data for real-time dynamic obstacle detection and tracking. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10528–10535. IEEE, 2020.

[56] Helen Oleynikova, Dominik Honegger, and Marc Pollefeys. Reactive avoidance using embedded stereo vision for mav flight. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 50–56. IEEE, 2015.

[57] Andrew J Barry, Peter R Florence, and Russ Tedrake. High-speed autonomous obstacle avoidance with pushbroom stereo. *Journal of Field Robotics*, 35(1):52–68, 2018.

[58] Sikang Liu, Michael Watterson, Sarah Tang, and Vijay Kumar. High speed navigation for quadrotors with limited onboard sensing. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 1484–1491. IEEE, 2016.

[59] Albert S Huang, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox, and Nicholas Roy. Visual odometry and mapping for autonomous flight using an rgb-d camera. In *Robotics Research: The 15th International Symposium ISRR*, pages 235–252. Springer, 2017.

[60] Zhefan Xu, Yumeng Xiu, Xiaoyang Zhan, Baihan Chen, and Kenji Shimada. Vision-aided uav navigation and dynamic obstacle avoidance using gradient-based b-spline trajectory optimization. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1214–1220. IEEE, 2023.

[61] Junqing Ning, Haotian Zhang, and Quan Quan. Dynamic obstacle avoidance of quadcopters with monocular camera based on image-based visual servo. In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 150–156. IEEE, 2022.

[62] John H Wall, David M Bevly, et al. Characterization of various imu error sources and the effect on navigation performance. In *Proceedings of the 18th international technical meeting of the satellite division of the institute of navigation (ION GNSS 2005)*, pages 967–978, 2005.

[63] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6, 1993.

[64] Zhefan Xu, Xiaoyang Zhan, Yumeng Xiu, Christopher Suzuki, and Kenji Shimada. Onboard dynamic-object detection and tracking for autonomous robot navigation with rgb-d camera. *IEEE Robotics and Automation Letters*, 9(1):651–658, 2023.

[65] Zhefan Xu, Xinming Han, Haoyu Shen, Hanyu Jin, and Kenji Shimada. Navrl: Learning safe flight in dynamic environments. *IEEE Robotics and Automation Letters*, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper's contributions are presented in the section 3 and 4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed at the end of the paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Proofs of all models and formulas used in this paper are provided in the supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.

- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**
Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?
Answer: [Yes]
Justification: All information needed to reproduced the result are provided in the section 3 of the paper.
Guidelines:
    - The answer NA means that the paper does not include experiments.
    - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
    - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
    - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
    - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
        (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
        (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
        (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
        (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**
Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?
Answer: [No]
Justification: We will release the code upon acceptance.
Guidelines:
    - The answer NA means that paper does not include experiments requiring code.

- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Training and testing details are provided in the supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Errors are provided in the table 1 and in the supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).

- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The devices we used and the latency of the system are provided in section 3. Training details are provided in the supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discussed the broader impacts in the conclusion.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We discuss the safeguards in the conclusion.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets used in this paper are properly credited, the license and terms of use are explicitly mentioned and properly respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The model proposed in this paper is clearly documented.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.