Taming Event Cameras with Bio-Inspired Architecture and Algorithm: A Case for Drone Obstacle Avoidance

Jingao Xu[1], Danyang Li[1], Zheng Yang[1]✉, Yishujie Zhao[1], Hao Cao[1], Yunhao Liu[1], Longfei Shangguan[2]
[1]Tsinghua University [2]University of Pittsburgh
{xujingao13,lidanyang1919,hmilyyz,yunhaoliu}@gmail.com
cobs14@126.com,caoh19@mails.tsinghua.edu.cn,longfei@pitt.edu

ABSTRACT

Fast and accurate obstacle avoidance is crucial to drone safety. Yet existing on-board sensor modules such as frame cameras and radars are ill-suited for doing so due to their low temporal resolution or limited field of view. This paper presents **BioDrone**, a new design paradigm for drone obstacle avoidance using stereo event cameras. At the heart of BioDrone is two simple yet effective system design inspired by the mammalian visual system, namely, a chiasm-inspired signal processing pipeline for fast event filtering and obstacle detection, and a lateral geniculate nucleus (LGN)-inspired event matching algorithm for accurate obstacle localization. To make BioDrone a practical solution, we further take significant engineering efforts to deploy the software stack on FPGA through software and hardware co-design. The performance comparison with two state-of-the-art event-based obstacle avoidance systems shows BioDrone achieves a consistently high obstacle detection rate of 96.1%. The average localization error of BioDrone is 6.8cm with a 4.7ms latency, outperforming both baselines by over 40%.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile computing**.

✉ Zheng Yang is the corresponding author.
Jingao Xu and Danyang Li are co-primary authors.
Project repository: https://github.com/MobiSense/BioDrone.

ACM MobiCom '23, October 2–6, 2023, Madrid, Spain

ACM ISBN 978-1-4503-9990-6/23/10…$15.00
https://doi.org/10.1145/3570361.3613269

KEYWORDS

Mobile Computing; Drone-based Applications; Obstacle Avoidance; Event Camera; Bio-inspired Design

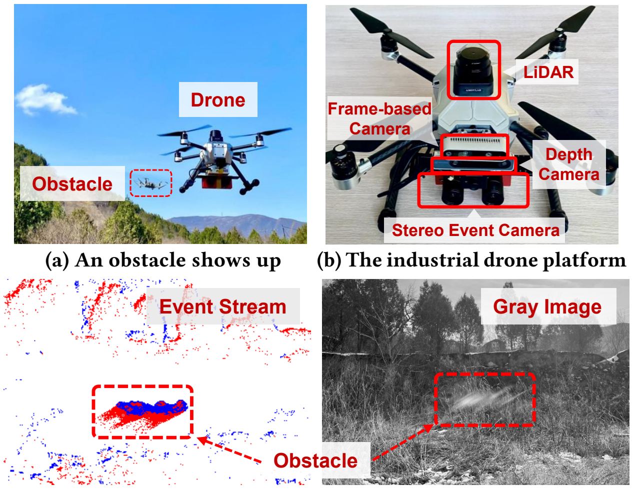
ACM Reference Format:
Jingao Xu, Danyang Li, Zheng Yang, Yishujie Zhao, Hao Cao, Yunhao Liu, Longfei Shangguan. 2023. Taming Event Cameras with Bio-Inspired Architecture and Algorithm: A Case for Drone Obstacle Avoidance. In *The 29th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '23), October 2–6, 2023, Madrid, Spain.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3570361.3613269

1 INTRODUCTION

Drones are among the most disruptive inventions in the past few years, spawning many novel applications including aerial imaging[7, 35, 37], last-mile delivery[47, 68, 71], sky networking[15, 61], and industrial inspection[1, 13, 75]. Despite their huge market value, safety remains a crucial challenge for drones, particularly for those high-speed drones in industrial and urban applications. For instance, DJI's industrial drones cruise at up to 25m/s[19], and the relative speed between two Amazon delivery drones can reach 30m/s[4]. Drone collisions with obstacles (e.g., birds[48], drones[16]) will not only cause financial loss but also threaten human safety[2, 27], which sets a strong barrier for drone adoption.

Fast and accurate obstacle detection and localization plays a key role in drone obstacle avoidance – the lower the detection latency, the more time the drone could take to react; and a higher localization accuracy increases the likelihood the drone can dodge them. Existing solutions primarily rely on frame-based cameras[22, 70] and radars[33, 45]. However, the low spatial-temporal sampling resolution of these onboard sensors makes it challenging for drones to perceive obstacles timely or localize them accurately.

For instance, the sampling interval of a typical frame-based camera varies from 20ms to 50ms, during which an obstacle can move up to 40cm (given a 20m/s relative speed). As a result, we are expected to see severe motion blurring on each image (Fig.1c). Such motion blurring will fail the vision



(a) An obstacle shows up (b) The industrial drone platform

(c) A comparison of captured event stream and ordinary gray image (without versus with obvious motion blur)

Figure 1: Snapshot of an obstacle avoidance maneuver.

algorithms, impairing both obstacle detection and localization accuracy. The radar-based solutions, on the other hand, suffer from high miss detection rates due to their limited field of view (FoV)[24, 76].

Drone obstacle avoidance with event cameras. Event cameras are novel bio-inspired sensors that report pixel-wise intensity changes asynchronously. Endowed with microsecond resolution, event cameras are able to capture high-speed motions without blurring (Fig.1c). Hence event cameras are envisioned to be an ideal solution to challenging vision tasks such as high-speed feature tracking[85], motion tracking[39], and simultaneous localization and mapping (SLAM)[26].

To better understand the potential of event cameras for obstacle avoidance, we reimplement seven state-of-the-art systems[10, 24, 29, 49, 52, 65, 84] (red circles in Fig.2, and [29, 65] require multi-modal fusion) with stereo event camera setup and evaluate their performance. Our benchmark studies (§2.3) reveal that they face fundamental challenges for high-speed drone adoption, as elaborated below.

• **Event burst impairs drone obstacle detection.** Event cameras are hyper-sensitive to environmental change. For instance, a slight change in lighting can lead to a remarkable change in pixel-wise intensity, resulting in hundreds of event reports. In practice, the scene in the camera's view changes rapidly due to drone movement; thus we will see an event burst where thousands of events are reported within a short time and those critical obstacle-triggered events are easily buried by massive numbers of environment-triggered events.

• **Event sticking delays drone obstacle localization.** Conventional vision algorithms are designed for frame-based cameras and cannot be directly applied to event streams for obstacle localization because the output of an event camera

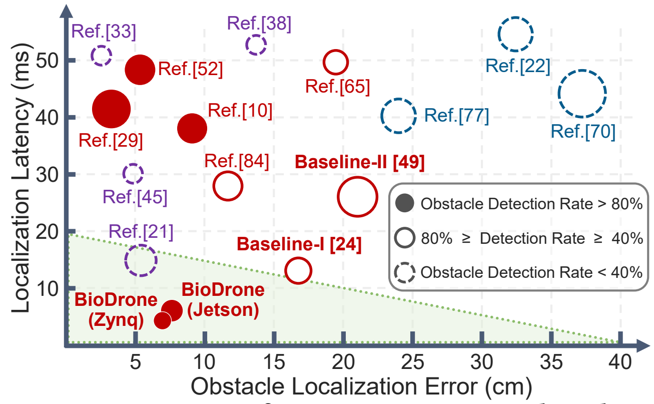


Figure 2: System performance comparison based on our field studies (§2). A smaller circle size indicates a more stable performance. The green area indicates a feasible performance range for high-speed (i.e., $20m/s$) drones (§2.2).

is not an image but a stream of asynchronous events. To address this issue, the current practice periodically sticks a lot of scattered events into a compact image and applies image-based algorithms (e.g., stereo triangulation[28] or deep neural networks[10, 29, 52, 65]) that are both computationally demanding. Repeating these operations would cause significant delays in obstacle localization.

Although existing solutions (e.g., Baseline-I[24]) achieve high obstacle *detection* accuracy by using a monocular event camera. The obstacle *localization* performance, however, drops significantly in high-speed scenarios (i.e., $20m/s$) due to the increasing task complexity and growing data volume.

Given that the event camera is a kind of bio-inspired vision sensor, we ask a question: Could we tackle the above challenges by studying how animals process binocular visual signals for efficient obstacle localization? To answer this question, we resort to bionics and take a comprehensive study (§3.1) on (i) how binocular visual signals are transmitted from the retina to the visual cortex in the mammalian visual system; and (ii) how they are rapidly filtered, matched, and spatio-temporal corrected through the visual pathway.

Our Work. In this paper, we leverage the **Bio**logical lessons learned from mammalian visual system and propose BioDrone, a **Drone**-oriented obstacle avoidance system. BioDrone features three key designs to fully unleash binocular event cameras' potential for obstacle localization and is implemented on FPGA with software-hardware co-design by embracing on-chip intelligence[6, 57], as elaborated below.

• On system architecture front, we imitate how mammal's *visual pathway* processes binocular visual signals and propose a visual-pathway-inspired signal processing pipeline for binocular event streams. Unlike the current practice where event streams are processed separately and not fused until the final triangulation stage, BioDrone fuses binocular event streams at an early stage, enabling the subsequent event filtering, matching, and localization modules to take full advantage of binocular information (§3.3).

• On system algorithm front, we first introduce a Chiasm-inspired Event Filtering (CEF) algorithm to quickly filter out environment-triggered events from the massive amount of events with a very low false positive rate (§4.1). We then propose a Lateral Geniculate Nucleus (LGN)-inspired Event Matching (LEM) algorithm to determine the spatial location of obstacles directly on binocular event streams thus eliminating cumbersome event sticking operations (§4.2).

• On system implementation front, we implement BioDrone on a commercial Xilinx Zynq-7020[73] chip. We design exclusive logic circuits, on FPGA, to parallelize the pixel-wise event processing, expediting the whole software stack (§5).

We deploy BioDrone on a drone testbed and further integrate it into ArduPilot[5], a widely-used open-source drone flight controller. We conduct extensive experiments with various types of obstacles and in different flying speed settings both indoors and outdoors. We compare the end-to-end obstacle localization accuracy and latency of BioDrone with two state-of-the-art (SOTA) event camera-based drone obstacle avoidance systems Baseline-I[24] (Science Robotics'20) and Baseline-II[49] (IROS'18). Evaluation results show that BioDrone achieves 96.1% obstacle detection rate, outperforming both baselines by >10%. BioDrone further achieves 6.8cm obstacle localization accuracy with 4.7ms latency, outperforming baselines by >40%.

In summary, this paper makes following contributions.
(1) We systematically study both the conventional sensor- and event camera-based drone obstacle avoidance systems, and reveal the fundamental limitations of these solutions.
(2) We demonstrate that bio-inspired designs can effectively improve drone obstacle detection and localization accuracy with a stereo event camera setup. As biology progresses and the future grasp of the visual system deepens, we believe this bio-inspired design will spawn new ideas in this domain.
(3) We fully implement BioDrone through software-hardware co-design and deploy it on an industrial drone, conducting a head-to-head comparison with two SOTA systems. The evaluation results show BioDrone's efficacy.

2 BACKGROUND AND MOTIVATION

2.1 Drone Obstacle Avoidance Primer

As illustrated in Fig.3a, obstacle avoidance consists of *localization* and *action* two phases. During the *localization* phase, suppose an obstacle shows up abruptly at t_0 and is perceived by a drone at t_1 after a perception delay Δt_p. Upon detecting the obstacle, the drone takes Δt_c to localize the obstacle. Noting that the drone still follows its planned trajectory to move before localizing the obstacle at t_2. Afterward, the on-board flight controller changes the drone's trajectory to dodge the obstacle (i.e., keep a safe distance from it) in the *action* phase.

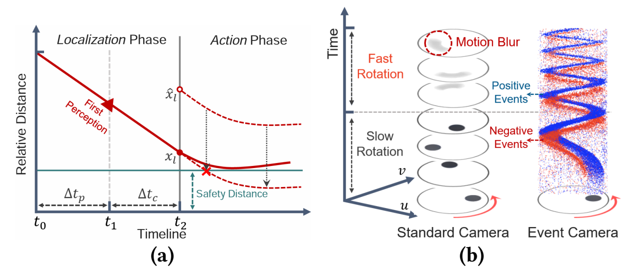


Figure 3: (a) Relative distance changes in an avoidance maneuver. (b) Illustration of the event stream.

Both the localization delay ($\Delta t_l = \Delta t_p + \Delta t_c$) and localization error ($\Delta x = \hat{x_l} - x_l$) are crucial to drone obstacle avoidance. A long delay Δt_l leaves the drone very short time to react, and a large localization error Δx misleads the flight controller to execute a wrong obstacle avoidance maneuver.

2.2 Limitations of the Current Practice

Existing radar-based (e.g., LiDAR[31, 33, 43], mmWave[21, 46, 76], or ultrasound[80]) and camera-based solutions (e.g., monocular[30, 70], stereo[22, 77], or depth[60]) are ill-suited for drone obstacle avoidance due to their high perception delay (high Δt_p) or low localization accuracy (high Δx). Take DJI Matrice 200 v2, the most advanced industrial drone, as an example. Its cruising speed can reach $v_c = 20m/s$ in a wide range of applications such as urgent package delivery, emergency services, and rapid terrain mapping. Based on the manufacturer's safety guideline[20], the safety distance between the drone and obstacle should be no less than $l_s = 60cm$, with an emergency braking distance at $l_b = 50cm$.

What is a feasible solution? Let l be the distance between the drone and the obstacle. To ensure the drone can successfully evade the obstacle, the localization delay Δt_l and accuracy Δx should satisfy the following equation:

$$v_c \times \Delta t_l + l_b + \Delta x \leq l - l_s. \quad (1)$$

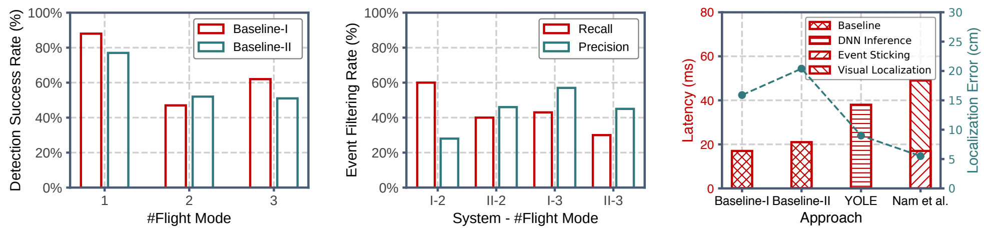
Now assuming an obstacle shows up 1.5m (i.e., $l = 1.5m$) away from the drone. Based on Eq.1, we can draw a feasible area (the triangle area in green) in Fig.2 where each point within this area represents an acceptable localization latency and accuracy. Any point outside this feasible area would lead to a failure in drone obstacle avoidance maneuvers.

Examining the current practice. We build a drone testbed by integrating five types of sensor units, including a LiDAR, a mmWave radar, a depth radar, a monocular camera, and a stereo camera, into an industrial drone (Fig.1b). We then re-implement seven obstacle avoidance solutions [21, 22, 33, 43, 45, 70, 77] and evaluate their performance in high-speed scenarios by conducting over 1,500 obstacle avoidance tests (§6). Fig.2 depicts their obstacle detection rate, average

Flight Mode	Event Generating Speed (e/ms)	
	Background	Obstacle
#1 (Normal Speed) $2m/s \le v_t \le 4m/s$ $5^\circ/s \le \omega \le 15^\circ/s$	103 (34.5%)	195 (**65.5%**)
#2 (Rapid Translation) $18m/s \le v_t \le 25m/s$ $5^\circ/s \le \omega \le 15^\circ/s$	675 (**71.4%**)	270 (28.6%)
#3 (Rapid Rotation) $2m/s \le v_t \le 4m/s$ $75^\circ/s \le \omega \le 95^\circ/s$	1091 (**75.9%**)	346 (24.1%)

Event Generating Speed Comparison

(a) Event Generating Speed



(b) Obstacle Detection Rate **(c) Event Filtering Rate** **(d) Localization Accuracy-Latency**

Figure 4: Performance of existing event-based solutions at different translation and rotation speed settings.

localization accuracy (Δx) and delay (Δt_l). As seen, frame camera-based solutions (i.e., blue circles) suffer from long localization delays due to their long frame exposure time (around 20ms) and excessive image processing delay (adding another 10-20ms). Worse still, the motion blurring shown in each frame would confuse the obstacle detection and localization algorithm, exacerbating both misdetection rate and localization errors. On the other hand, LiDAR and mmWave radar-based solutions (i.e., purple circles) suffer from low detection rate (i.e., ≤ 40%) due to their limited field of view.

2.3 Event Camera for Obstacle Avoidance: Opportunities and Challenges

Event cameras are bio-inspired sensors that work differently from frame-based cameras. Instead of capturing images at a fixed rate, an event camera measures per-pixel brightness changes asynchronously, resulting in a stream of events at *microsecond* resolution[26]. Specifically, an event camera has smart pixels (similar to the photoreceptor cells on retinas) that trigger events independently of each other: as shown in Fig.3b, once a pixel detects a change of intensity in the scene, it will instantly output an event $e_k = (\boldsymbol{x}, t_k, p_k)$, encoding the occurrence time t_k (at microsecond resolution), pixel location $\boldsymbol{x} = (u, v)$, and polarity p_k (blue for +1 *vs*. red for -1 in Fig.3b) of the intensity changes (i.e., brighter or darker).

The high temporal resolution empowers the event camera to detect environment changes promptly. However, leveraging event cameras to detect obstacles for high-speed drones is facing two fundamental challenges, as elaborated below.

• **C1: Event burst impairs drone obstacle detection.** Events captured by an event camera can be classified into two categories: environment-triggered and obstacle-triggered events. The former is generated due to the ego-motion of the event camera, while the latter is caused by the appearance of obstacles. To detect and further localize an obstacle, a system needs to identify obstacle events from massive events. To this end, existing solutions apply IMU-based ego-motion compensation algorithms to filter out environment-triggered events[24, 26, 49]. However, as shown in Fig.4a, the number of events generated per millisecond surged from around 300 to 1,500 and the new additions are mainly environment events. Such a burst of environmental events overwhelm obstacle events and degrade existing algorithms' performance.

To validate the above analysis, we conduct an obstacle detection experiment under different flight modes. As shown in Fig.4b, the detection rate of two SOTA solutions, Baseline-I[24] and -II[49], drops to < 60%. To better understand the reasons for failure cases, we further examine the event filtering performance of these two baselines in two high-speed flight modes (i.e., mode 2 and mode 3). We observe that both systems achieve a very low event filtering rate (recall and precision < 60% in Fig.4c), which confirms our analysis.

• **C2: Event sticking delays drone obstacle localization.** Once the obstacle is detected, the drone has to localize it in 3D space. Typically, localization is more time-consuming than detection due to the additional operations involved. For instance, Baseline-I requires binocular parallax optimization, matching, and triangulation after detection, which is more computationally intensive as outlined in [24].

Moreover, conventional vision algorithms (e.g., stereo triangulation[28]) or DNNs cannot be directly applied as the output of an event camera is not fix-rate frames but a stream of asynchronous events. To solve this issue, the current practice proposes to (i) stick all generated events within a time window (e.g., < 1ms) into an image and then apply image-based algorithms (Fig.5c); or (ii) design event data-oriented DNNs (e.g., spiking neural networks[63]) for object localization. However, as depicted in Fig.4d, although the localization accuracy is boosted, either the sticking operations, the stereo visual algorithms, or DNN inference introduces significant delays, leaving the drone no time to react.

In summary, although event cameras hold great potential for delay-sensitive tasks such as drone obstacle avoidance, there still lack effective algorithms and system support to fully unleash their potential, especially with a stereo setup for localization tasks.

3 BIO-INSPIRED ARCHITECTURE

Our system architecture and algorithms are inspired by the *biological visual pathway*. In this section, we first introduce the biological visual pathway in mammalian visual system and describe how visual information is filtered, processed,

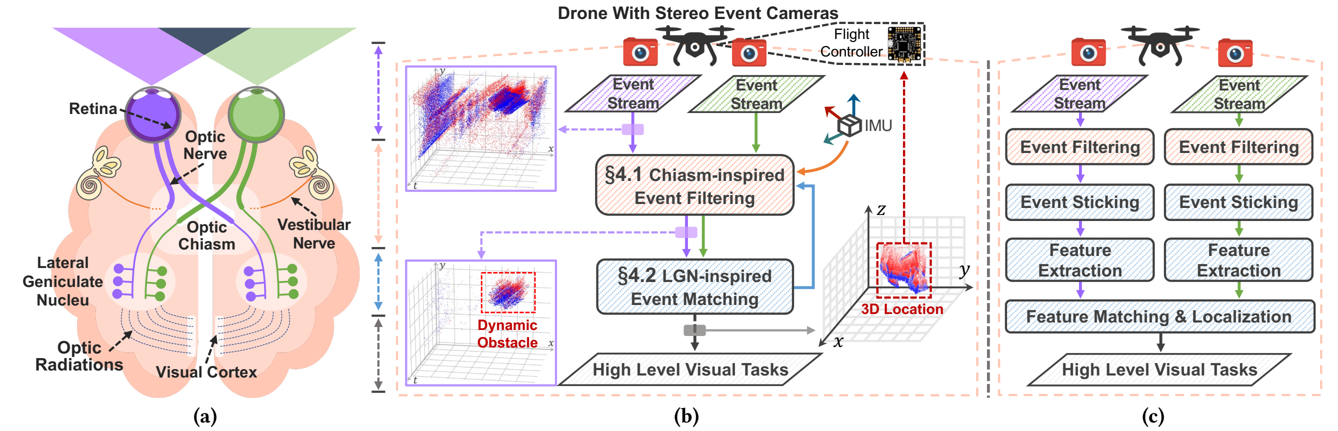


Figure 5: System architecture comparison. (a) Human binocular visual pathway. (b) BioDrone's architecture inspired by (a). (c) System architecture of conventional event-based systems[24, 49, 50], where binocular event streams are processed separately and follow traditional visual localization workflow (i.e., from feature extraction to matching and then stereo triangulation).

and transmitted from retina to brain through the pathway. We then present the lessons learned and explain how we leverage these insights to design BioDrone.

3.1 Biological Visual Pathway

As illustrated in Fig.5a, light entering eyes is refracted by the cornea and lens and then simulates photoreceptor cells on the *retina* to produce visual signals. The *optic nerves* carrying those visual signals from both eyes cross at the *optic chiasm*, which localizes at the base of the hypothalamus of the brain[36]. Additionally, the *vestibular nerves* that transmit human motion information, interact with the optic nerves at the optic chiasm and select which necessary visual signals will be further carried forward to the thalamus for subsequent processing[17]. Afterwards, the filtered visual signals enter the *lateral geniculate nucleus* (LGN) are re-organized and spatio-temporally correlated to achieve a 3D representation of environment[62]. Finally, the correlated visual representations reach the *visual cortex* via *optic radiations*, and complex visual perception tasks will be accomplished here.

3.2 Bio-lessons

We have learned two biological lessons from visual pathway:
• **L1: Early integration of binocular visual signals.** Binocular visual signals are integrated at an early stage (i.e., at optic chiasm instead of brain). This allows visual signal filtering and matching to take full advantage of the binocular information. In contrast, current practice[24, 49] process, filter, and extract visual features from each event stream independently, as illustrated in Fig.5c.
• **L2: Fast processing of low-level visual tasks.** The low-level visual tasks (e.g., object detection and localization) are rapidly accomplished along with the signal transmission through the visual pathway (i.e., the binocular visual signals are filtered at optical chiasm and then matched at LGN). The visual cortex, on the contrary, focuses on high-level visual tasks (e.g., object recognition or segmentation).

While the mechanisms of the optic chiasm and LGN handle visual signals are still being explored, the above early integration and collaborative signal-processing architecture inspires our design for handling binocular event streams. Our work aims to provide initial proof that bio-inspired designs can heighten the performance of corresponding bio-inspired sensors. Future designs, which rely on the latest research, are subject to evolution as new discoveries emerge.

3.3 Overview of BioDrone

BioDrone shares a similar architecture with the biological visual pathway to unleash the potential of event cameras, as shown in Fig.5b. We explain the functional units below.
• From the architecture perspective, following lesson-L1, BioDrone features a visual-pathway-inspired signal processing pipeline, fusing binocular event streams at an early stage, which allows the obstacle detection and localization tasks to combine and fully leverage the binocular event information.
• From the algorithm perspective, following lesson-L2, BioDrone attempts to mimic how *optical chiasm* and *LGN* process binocular visual signals and designs two heuristic algorithms. Specifically, BioDrone proposes a *Chiasm-inspired Event Filtering* (CEF) mechanism for event filtering and obstacle detection, and an *LGN-inspired Event Matching* (LEM) module to localize obstacles from the integrated event stream.

Finally, the obstacle localization results from LEM will be used to guide the flight controller to execute desirable evasive operations. By optimizing the algorithms to parallelize event processing, we accelerate the entire software stack on FPGA.

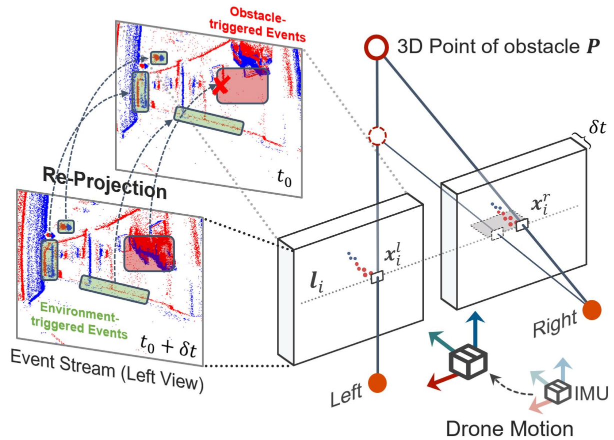
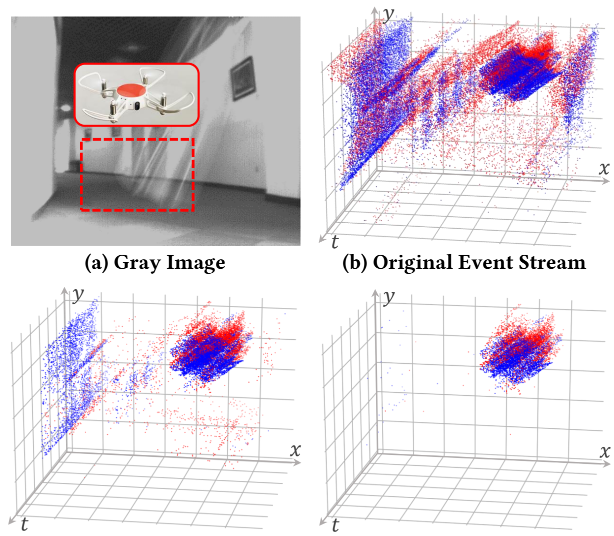


Figure 6: Illustration of the chiasm-inspired event filtering scheme. Left: the distinction between environment- and obstacle-events under ego-motion instruction; Right: the binocular constraint that an obstacle event should satisfy.



(a) Gray Image **(b) Original Event Stream**

(c) Filtering w/ Ego-motion **(d) Ultimate Filter Performance**

Figure 7: Step-by-step event filtering performance.

4 BIO-INSPIRED ALGORITHM DESIGN

In this section, we describe two bio-inspired algorithms for event filtering (§4.1) and obstacle localization (§4.2).

4.1 Chiasm-Inspired Event Filtering

Optic nerves carrying visual signals from eyes and vestibular nerves carrying motion signals from cochleas cross at optic chiasm. Like a busy intersection, optic chiasm is the rendezvous point where binocular visual information gets fused and filtered under the guidance of proprioceptive motion information. Typically, about 1,200,000 photoreceptors on human retina generate visual signals per second, yet merely around 1,700 of them would pass through optic chiasm[23].

Motivated by the chiasm's ultra-efficient signal filtering performance, we design a chiasm-inspired event filter that leverages the drone's IMU perception data (simulating the vestibular motion signals) to pick up obstacle events in binocular event streams. The insight behind this mechanism lies in two-fold: (i) IMU could be leveraged to infer the ego-motion of event cameras, which provides a priori knowledge to cull environment-triggered events. Just like in our daily life, when your head is turning right, your righter visual field becomes clear while the lefter blurs, and vice-versa; and (ii) the collaborative use of binocular event streams would further improve the filtering performance as the spatial relationship (i.e., pose transformation) between the stereo event cameras provides an additional constraint. For instance, a single eye is less sensitive to the depth change of a moving object compared to two eyes.

4.1.1 *Event Filtering Based On Ego-motion Instruction*.

We first explain how we filter events based on IMU sensor readings. As illustrated in both Fig.6 and Fig.7b, suppose we collect a batch of events $\mathcal{E}$ and IMU data $\mathcal{I}$ within a short time window $[t_0, t_0 + \delta t]$, and each dot in these figures represents an event and the color of this dot shows its polarity (red for -1 while blue for +1). The high-end IMU sensors on the drone can provide motion tracking results[24, 58]. Given an arbitrary event shows up in this window, say, $e_i = (\boldsymbol{x_i}, t_i, p_i)$ where $\boldsymbol{x_i}, p_i$ represents the event's pixel-wise location and polarity at timestamp t_i, we can estimate the event's past location $\hat{\boldsymbol{x_0}}$ at t_0 based on drone motions. There are two cases:
(i) If e_i is an environment-triggered event, its estimated location $\hat{\boldsymbol{x_0}}$ at t_0 should match its real location $\boldsymbol{x_0}$ captured at timestamp t_0 because the environment change is only caused by the drone movement.
(ii) Conversely, if e_i is an obstacle-triggered event, its estimated location $\hat{\boldsymbol{x_0}}$ at t_0 should not match $\boldsymbol{x_0}$ since the obstacle moves yet our event location prediction does not take into account the obstacle movement.

Based on the above analysis, we project all events captured within the time window $[t_0, t_0 + \delta t]$ to the timestamp t_0 and obtain a 2D map, as shown in Fig.6. The environment-triggered events (denoted by green rectangles) are aligned when being projected to this 2D map, whereas the obstacle-triggered events (denoted by red rectangles) are scattered due to obstacle movement. BioDrone explores this opportunity to filter out the former. We model this process in §A.1 and present the algorithm in §4.1.3. We further design circuits on FPGA to parallelize the projection process, as the operations for each event are independent (§5).

4.1.2 *Event Filtering Based On Binocular Consistency*.

Over 70% of environmental events are successfully filtered out with ego-motion instruction (Fig.7c). However, there are still stubborn residues since: (i) event cameras also suffer from sampling noise, and the firing time of those noisy events

Algorithm 1: Chiasm-Inspired Event Filtering

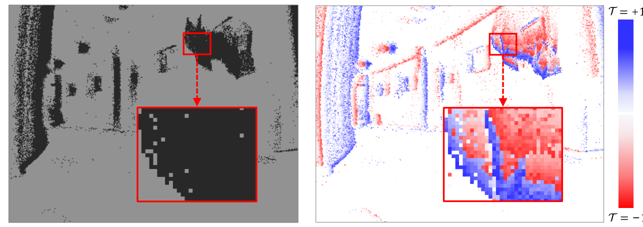
Input: Raw Event Stream: $\mathcal{E}^l = \{e_i^l\}, \mathcal{E}^r = \{e_i^r\}$
Output: Obstacle-triggered events: $\mathcal{E}^l_{obstacle}, \mathcal{E}^r_{obstacle}$

1 **for** each $\mathcal{E}$ in $\{\mathcal{E}^l, \mathcal{E}^r\}$ **do**
2   **for** each e_i in $\mathcal{E}$ **do**
3         $\boldsymbol{x} = \texttt{EventWarp}(e_i)$;
4         $D[\boldsymbol{x}]$.append(e_i);
5   **end**
6   **for** each $\boldsymbol{x}$ in D.key() **do**
7         $\rho_{imu}(\boldsymbol{x}) = \texttt{ObstacleCheck}(D[\boldsymbol{x}], \mathcal{E})$;
8   **if** $\rho_{imu}(\boldsymbol{x}) > \tau_{threshold}$ **then**
9             $\mathcal{E}_{IMU}$.extend$(D[\boldsymbol{x}])$;
10   **end**
11   **end**
12 **end**
13 $d = \texttt{DepthEstimate}(\mathcal{E}^l_{IMU}, \mathcal{E}^r_{IMU})$;
14 **for** each e_i^l in $\mathcal{E}^l$ **do**
15     $e_i^r = \texttt{EventDiscover}(e_i^l, d)$;
16   **if** $e_i^r \neq \texttt{Null}$ **then**
17         $\mathcal{E}^l_{obstacle}$.append(e_i^l); $\mathcal{E}^r_{obstacle}$.append(e_i^r);
18   **end**
19 **end**

is random; and (ii) due to the bias of projection view, different events may be accidentally mapped to the same pixel areas. Accordingly, it's challenging to find an optimal threshold to distinguish the different events in ego-motion instruction. This is also the reason why the current practice[24, 49] fails to achieve reliable event filtering performance.

To address this issue, we first cluster events and obtain the obstacle's counter, as shown in the top-right corner of Fig.7c. The events outside the contour can be safely filtered out. However, since the contour estimated by clustering is less accurate, we are expected to see plenty of environment-triggered events around the obstacle. To filter out these events, we are inspired by binocular consistency. As illustrated in the bottom right of Fig.6, let e_l and e_r be an event reported by the left event camera and right event camera, respectively. We first estimate a rough depth of the obstacle by triangulating the center of contour[24]. If e_l is triggered by an obstacle, then according to the Epipolar constraint[82], e_r should show up on a specific area which can be estimated by the pixel location of e_l and the depth of the obstacle (estimated by triangulation). Otherwise, if e_r is not found in this specific area, both e_r and e_l should be environment-triggered event. We model this process in §A.2.

Finally, we remove the salt-and-pepper noise using morphological algorithm[24]. Compared to the original event



(a) Event Sticking **(b) Polarity Time-surface**

Figure 8: Event representation method comparison.

stream (Fig.7b), the obstacle is well separated after these two steps filtering, presenting a sharp and clear contour (Fig.7d).

4.1.3 Putting Them Together. Algorithm 1 shows how we practice these two steps filtering. Line 1–12 and Line 13–19 represents event filtering based on ego-motion instruction (first step) and binocular consistency (second step), respectively. Specifically, given input event streams $\mathcal{E}_l$ and $\mathcal{E}_r$ within the time window $[t_0, t_0 + \delta t]$, function `EventWarp` first predicts each event's past location at timestamp t_0. The algorithm then runs function `ObstacleCheck` on each pixel to decide whether the events are triggered by environment or an obstacle. Only those obstacle-triggered events (termed as $\mathcal{E}_{IMU}$) are fed into the second-step where `EventDiscover` function leverages the binocular consistency to eventually obtain obstacle-triggered events $\mathcal{E}_{Obstacle}$.

4.2 LGN-Inspired Event Matching

Visual signals passing through the optic chiasm are spatiotemporally correlated at LGN in order to obtain a 3D representation of the object. As shown in Fig.9, the architecture of LGN is characterized by six distinctive layers. The inner two layers are magnocellular layers that are responsible for detecting object motion and size (i.e., coarse feature), while the outer four layers are parvocellular layers for detecting the object's color and contour (i.e., fine details)[62]. Such a six-layer folding architecture supports a plethora of anatomical calculations without involving those computationally intensive spatial and temporal correlations.

Inspired by this elegant structure, we propose a neural-enhanced event matching algorithm, as elaborated below.

• First, we propose a novel event stream representation, namely polarity time-surface (§4.2.1), that maps the 3D event stream to the 2D space without sacrificing the valuable event features. Such a design can expedite feature matching without hurting the matching accuracy.

• Second, similar to LGN, we propose a six-layer hierarchical event feature extraction and matching algorithm (§4.2.2) that can localize the obstacle based on the binocular polarity time-surface timely and accurately.

4.2.1 ***Spatio-Temporal Representation of Events.*** To localize the obstacle, we have to extract reliable event features

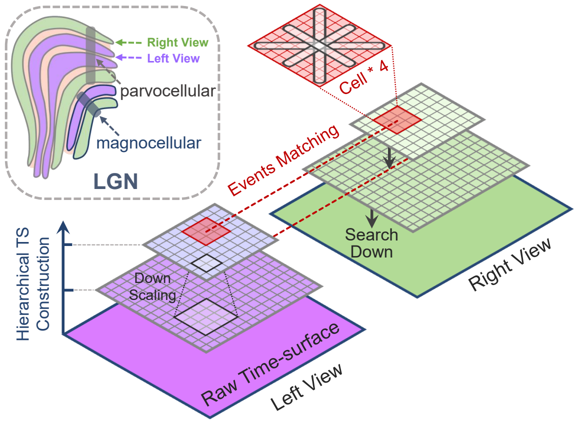


Figure 9: Illustration of our proposed LEM algorithm.

of the obstacle from the binaural event streams. Although it is viable to exact event features from the 3D meta event streams, this solution is time-consuming since there are hundreds of events stacking together within a short period.

In BioDrone, we propose a lightweight representation of event streams, namely, Polarity Time-Surface (P-TS), that can well retain rich spatio-temporal information. P-TS is a 2D map where each pixel value represents both the polarity and timestamp of the event. For instance, as shown in Fig.8b, the red and blue color indicates two different polarities of the event while the darkness of the color shows the time this event being captured. We leverage an exponential decay kernel[40] to prioritize the latest event over past events, emulating how LGN prioritizes the latest visual signals over the old ones[62]. For each pixel $\boldsymbol{x} = (u, v)^T$, its polarity time-surface presentation is formally defined as:

$$\mathcal{T}(\boldsymbol{x}, t) = \rho_{\text{last}}(x) \cdot \exp\left(-\frac{t - t_{\text{last}}(\boldsymbol{x})}{\eta}\right), \tag{2}$$

where $t_{\text{last}}(\boldsymbol{x})$ and $\rho_{\text{last}}(x)$ are the timestamp and polarity of the event showing up at pixel $\boldsymbol{x}$; η is the decay rate. The parameter $\rho_{\text{last}}(x)$ provides an additional polarity constraint for event matching. Compared to the sticked event image (Fig.8a), the proposed P-TS presentation retains the fine-grained texture of the obstacle, making it easily distinguishable. Generating a P-TS for each event stream takes $O(n)$ time since only a one-shot traversal of events is needed. Additionally, the process will be further accelerated on FPGA (§5) as each pixel could infer Eq.2 in parallel, superior to those event sticking-based solutions[52, 65] where periodically intensive memory access is unavoidable.

4.2.2 *Fast Event Feature Matching*.

Next, we run a feature extraction and matching on the binocular P-TS maps for obstacle localization. However, sweeping the entire P-TS maps for feature extraction and matching would consume significant amount of time. We thus resort to the lessons learned from LGN and design a pyramidal P-TS hierarchy to expedite feature extraction and matching.

On a high level, we build two 3-layer P-TS pyramids based on the P-TS map obtained from the left event stream and right event stream, respectively, as shown in Fig.9. In the P-TS pyramid, the bottom layer $\mathcal{T}_0$ (i.e., the original P-TS map) is subsampled by a factor of k to obtain the next pyramid level $\mathcal{T}_1$. $\mathcal{T}_1$ is then filtered in the same way and subsampled to obtain $\mathcal{T}_2$. We are expected to see a sequence of reduced resolution P-TS map on the pyramidal P-TS hierarchy, with a growing reception field. Each pixel on the top layer $\mathcal{T}_2$ corresponds to a reception field of $k^2 \times k^2$. By sweeping the $\mathcal{T}_2$ map, we can essentially reduce the searching space by k^4.

Pyramidal P-TS hierarchy generation. Let $\mathcal{T}_0^l$ and $\mathcal{T}_0^r$ be the left and right P-TS map, respectively. Without losing generality, we take the left P-TS map for algorithm description. We down-sample $\mathcal{T}_0$ by a factor of k:

$$\mathcal{T}_{i+1}(\boldsymbol{x}, t) = \mathcal{T}_i(k\boldsymbol{x} - \frac{k}{2}[1, 1]^T, t), \tag{3}$$

where $\mathcal{T}_{i+1}(i \in \{0, 1\})$ is the down-sampled P-TS. Each pixel in $\mathcal{T}_{i+1}$ represents a $k \times k$ region in previous P-TS. The pixel value is defined as that of the center pixel in the region.

Feature matching. We run feature matching on $\mathcal{T}_2$. For instance, given an event e^l in the left view, we leverage the following constraints to find its matched event e^r:

$$\sigma_e = |(\boldsymbol{x}^l)^T \boldsymbol{F} \boldsymbol{x}^r| < \epsilon_e, \tag{4}$$

$$\sigma_p = |\mathcal{T}_2^r(\boldsymbol{x}^r, t) - \mathcal{T}_2^l(\boldsymbol{x}^l, t)| < \epsilon_p, \tag{5}$$

where Eq.4 is the epipolar constraint (i.e., similar to §4.1.2, the matched events should on the same epipolar line[41]) and $\boldsymbol{F}$ is the fundamental matrix obtained from camera calibration. Eq.5 indicates the timestamp and polarity of these two events should be the same (i.e., difference within threshold ϵ_p). If no events on $\mathcal{T}_2^r$ satisfy Eq.4-5, we will down search the $k \times k$ region on $\mathcal{T}_1^r$, associated to the event with minimal σ_e and σ_p, to find the matched event; and further repeat the procedure on $\mathcal{T}_0^r$ if required. Once we find the matched e^l and e^r, the spatial location of the event with depth d is:

$$d = \frac{f * b}{||\boldsymbol{x}^l - \boldsymbol{x}^r||}, \tag{6}$$

where f is the focal length and b is the baseline distance between two optical centers.

We further propose a cell-matching operator to enhance Eq.4-5. As shown in Fig.9, each cell is a segment of pixels centered at the candidate pixel with a certain direction. We apply a horizontal, a vertical, and two diagonal cells, as LGN does, for aggregating scores (i.e., σ_e and σ_p). That is, the matching relationship of two events is not only determined by themselves, but also by their neighboring events. The operator makes the algorithm more sensitive to moving edges,

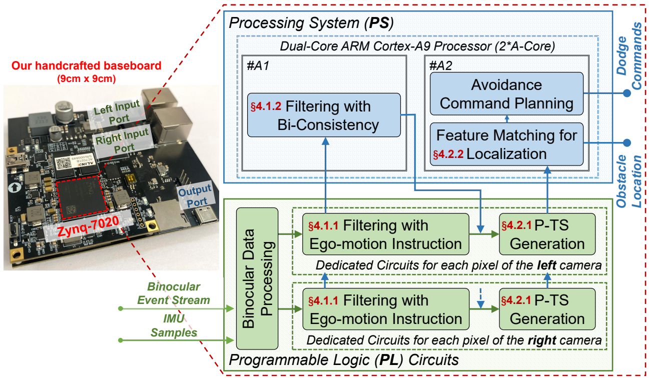


Figure 10: Implementation of BioDrone on a Zynq chip.

which fits the nature of event camera. Finally, the average 3D location of all events on the obstacle's contour is the result.

5 IMPLEMENTATION

We fully implement BioDrone on a Xilinx Zynq-7020 (the highlighted chip in Fig.10) through software-hardware co-design. It consists of a processing system (PS) and a programmable logic (PL) two modules. The PS features a dual-core ARM Cortex-A9 processor (i.e., #A1 and #A2), while PL is for hardware acceleration through FPGA. We also manufacture a baseboard for data input/output and voltage adaption.

• **PL:** We design exclusive logic circuits on FPGA to accelerate those event operations suitable for parallel and pipeline execution, i.e., data denoising, ego-motion-based filtering (§4.1.1), and P-TS generation (§4.2.1), on PL.

• **PS:** Before loading specific tasks, we first exploit a core-isolation strategy to isolate the computing resources of #A1 from PS, reducing the impact of CPU scheduling on task execution to better match the PL pipelines. We realize it by building a Linux OS with boot parameter `isolcpus=<cpu #A1>`. We execute the binocular-based filtering (§4.1.2) which requires frequent memory access and cannot be easily implemented through FPGA, on #A1. The final obstacle localization and command planning tasks are executed on #A2.

• **Data flow in-between:** We further leverage the physical-level direct memory access (DMA) technique[72] to transmit intermediate data among PL, #A1, and #A2. Compared with network-level solutions such as PL-PS ethernet interface[34] and OpenAMP[55], DMA ensures data interaction processes would not be interrupted by CPU scheduling.

We integrate BioDrone into an ArduPliot APM flight controller and deploy it on an AMOVLAB P450-NX drone. Implementation details can be found at the project repository.

6 EVALUATION

6.1 Experimental Methodology

Field studies. We conduct field studies both indoors and outdoors as shown in Fig.11. The performance is evaluated

Table 1: Different Drone Flight Mode Configurations

Flight Mode* (Trajectory)	Speed		Event Generating Speed (e/ms)
	Translation (m/s)	Rotation (°/s)	
1 ($A \rightarrow B \rightarrow C_1 \rightarrow D_1$)	2.0–6.0	0–10.0	100–400
2 ($A \rightrightarrows B \rightarrow C_1 \rightrightarrows D_1$)	15.0–26.5	0–10.0	350–1200
3 ($A \rightarrow B \looparrowright C_2 \rightarrow D_2$)	2.0–6.0	20–100	900–2100

* →: acceleration; →: uniformity; →: deceleration.

in three flight modes, as defined in Table 1. The drone follows planned trajectories to move; Four volunteers throw six different types of obstacles toward the drone during the drone's movement.

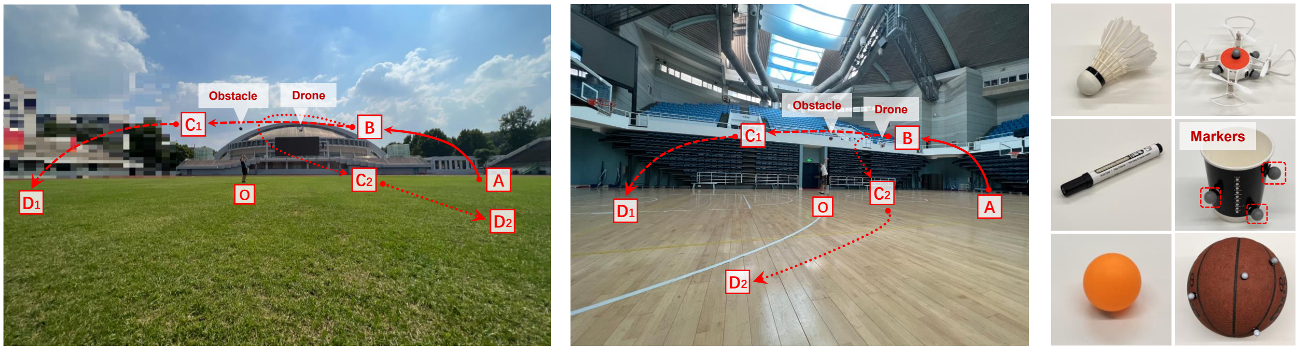
Repeatability. Before conducting experiments under each flight mode, we program a series of pre-determined flight commands into the on-board APM flight controller, enabling the drone to follow the planned trajectory, speed, and acceleration, to make our experiments repeatable.

Metrics and Ground truth. The drone logs its localization results with timestamps. We download these logs and evaluate end-to-end (E2E) localization latency Δt_l and error Δx (defined in §2). Indoors, an OptiTrack motion capture system could provide $<1mm$ localization ground truth. Outdoors, since we cannot deploy OptiTrack to obtain ground truth, we collect event streams and run an advanced yet heavy event-based object localization and segmentation neural network[3] offline. The results are taken as ground truth. We also log event classification results reported by it to examine BioDrone's event filtering performance.

Baselines. We compare the accuracy and latency of BioDrone with Baseline-I[24] and -II[49]. We also compare the LEM module in BioDrone with ESVO[84]. As these baselines are not implemented on FPGA, we thus implement BioDrone on the drone's onboard Nvidia Jetson TX2 for a fair comparison with them.

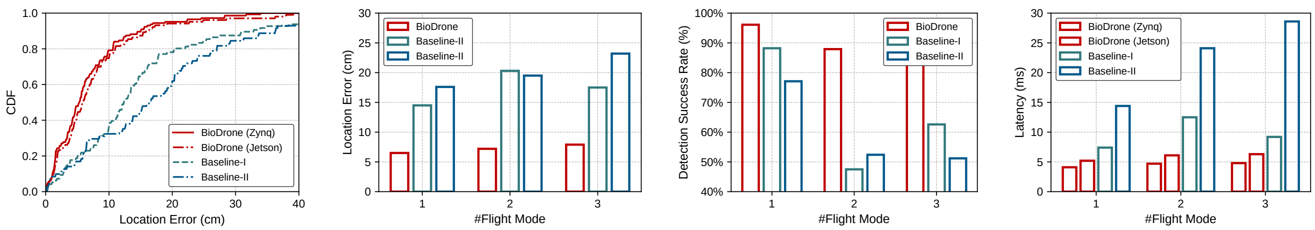
6.2 Overall Performance

Obstacle detection and localization. Fig.12a depicts the localization performance of BioDrone and two comparative systems. The average E2E location error of BioDrone is $7.5cm$, outperforming Baseline-I and Baseline-II, of which the average location errors are $15.9cm$ and $20.4cm$, respectively. We then compare the three systems under different flight modes. Fig.12b shows that BioDrone achieves the lowest average location error across all three flight modes, outperforming both baselines by >60% in all flight modes. Moreover, as shown in Fig.12c, the obstacle detection rate of BioDrone under three flight modes are 96.1%, 87.9% and 92.6%, outperforming baselines by over 10% in low-speed (mode 1) and over 40% in high-speed scenarios (flight mode 2 & 3), respectively. Unlike the two related works that depend solely on ego-motion instruction for event filtering, BioDrone incorporates a binocular constraint to upgrade the framework, improving obstacle



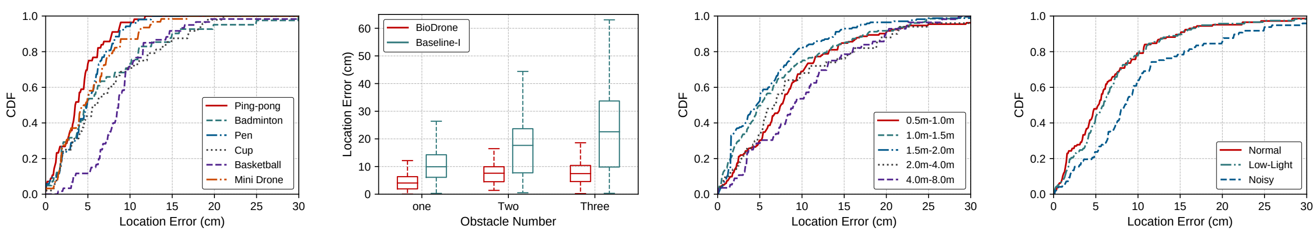
(a) Outdoor experiments (b) Indoor experiments (c) Different Obstacles

Figure 11: Experimental scenarios of BioDrone. The red lines show the drone's movement trajectory.



(a) Localization Error (b) Localization Error (c) Objection Detection Rate (d) Localization Latency

Figure 12: Overall Performance Comparison



(a) Impact of Obstacle Type (b) Impact of Obstacle Number (c) Impact of Obstacle Distance (d) Impact of Scene Dynamic

Figure 13: System Robustness Evaluation

detection performance. Additionally, BioDrone's LEM module optimizes stereo event matching, resulting in increased spatial localization accuracy.

End-to-end latency. We further evaluate the E2E latency, including the obstacle detection and localization two phases. As depicted in Fig.12d, the E2E latency of BioDrone is 5.2ms in flight mode 1, outperforming both baselines (6.1ms and 6.3ms, respectively). As the flight speed grows (i.e., mode 2&3), the E2E latency of Baseline-I and -II grows rapidly, making these two solutions fail in obstacle avoidance. In contrast, BioDrone maintains the lowest E2E latency (i.e., within 6ms) in all scenarios. Typically, related works encounter an inherent increase in system latency due to the simultaneous processing of binocular event streams. The subsequent steps – binocular parallax optimization, matching, and triangulation – for obstacle localization post-detection further delay the system. In contrast, BioDrone introduces the LEM module, which directly localizes objects from event streams, thereby reducing end-to-end latency.

FPGA acceleration. As shown in Fig.12a and Fig.12d, compared to a pure software implementation on Jetson, the software-hardware co-design of BioDrone on Zynq further reduces the localization bias by 9.3% (6.8cm vs. 7.5cm) and latency by 22.9% (4.7ms vs. 6.1ms). Our hardware design on FPGA (i) realizes the parallel and pipeline execution among operations for different events, reducing the E2E latency; and (ii) enables the system to select a shorter time window Δt (§4.1), which improves the event filtering performance for better localization accuracy.

6.3 System Robustness Evaluation

Impact of obstacle type. We evaluate the impact of different types of obstacles (Fig.11c, in terms of form factor and miscellaneous texture). The results are shown in Fig.13a. As seen, textured obstacles like badminton, pen and cup have lower average localization errors of 7.2cm, 5.0cm and 7.5cm, respectively. In contrast, obstacles with relatively large size and weak texture (e.g., basketball) lead to larger localization errors. Further scrutiny reveals that the event cameras are likely to generate more events for larger obstacles, and these events are on the vicinity of the obstacle and thus are likely to distort the obstacle localization. Nevertheless, the localization error is still within the acceptable range, which makes BioDrone a feasible solution for drone obstacle avoidance.

Impact of obstacle quantity. In this experiment, two volunteers are asked to throw multiple (2-3) obstacles toward the drone; we examine the detection and localization results for each obstacle individually. As depicted in Fig.14b, BioDrone outperforms Baseline-I by > 40% in all settings. As the number of obstacles grows, we observe a slight increase (around 3*cm*) in BioDrone's localization error. In contrast, the localization error of Baseline-I grows dramatically to 24.68 *cm*. The results demonstrate that the LEM module could effectively extract spatio-temporal features of different obstacles and thus distinguish them from each other. On the contrary, Baseline-I simply clusters events for triangulation, making it difficult to separate obstacles close to each other.

Impact of obstacle distance As shown in Fig.13c, when the obstacle appears at around 1.0-2.0*m*, BioDrone achieves the highest localization accuracy where the average location error is 6.58*cm*, and the average location error will slightly increase (within 9.5*cm* though) as the distance increases. Generally, a longer distance fails to generate sufficient events, making the feature matching more challenging.

Impact of environmental dynamic. We further evaluate BioDrone's effectiveness in low-light (i.e., dark) and noisy environments (i.e., with high background dynamics), respectively. The results are shown in Fig.13d. As seen, even in low-light conditions, the average localization accuracy remains consistent (a minor decrease within 10%), due to the HDR of event camera that still manages to capture sufficient events in dark environments. However, in noisy environments, there is a noticeable 33% decrease in average localization accuracy because BioDrone sometimes interprets dynamic background objects as foreground obstacles. A potential solution to this problem could be the integration of depth-based filtering algorithms, which is left as a future work.

6.4 Ablation Study

Contributions of each module. We examine how CEF and LEM contribute to BioDrone. We gradually embed CEF and LEM into the baseline system (i.e., Baseline-I) and repeatedly examine the localization accuracy and end-to-end latency. As shown in Fig.15a, without applying these two modules, Baseline-I achieves a localization error and latency of 15.9*cm* and 10.4*ms*. As we integrate CEF module into Baseline-I, we observe the localization error declines to 12.1*cm* and the latency drops to 6.3*ms*. We then integrate LEM into Baseline-I and observe that the localization error drops remarkably to 8.8*cm*. However, the delay grows due to the lack of an efficient event-filtering mechanism. Finally, we integrate both CEF and LEM into Baseline-I. As expected, both the localization error and latency reach the minimum.

Performance of CEF. We compare CEF with the filtering module of Baseline-I in high-speed mode (flight modes 2 and 3). We denote CEF in mode 2 and mode 3 as B-2 and B-3, respectively. Likewise, the filtering module of Baseline-I as I-2 and I-3, respectively. In Fig.14b, a higher recall means more obstacle-triggered events are preserved while a higher precision indicates more background events are removed. As seen, the recall of CEF is $\geq$ 81% and its precision is $\geq$ 82%. In contrast, the filter module of Baseline-I achieves an inferior recall of 43% under flight mode 3. Even worse, the precision further drops to 28% under flight mode 2. This result demonstrates the efficacy of CEF in event filtering.

Performance of LEM. We also evaluate the performance of LEM by comparing it with the localization module in ESVO[84] and Baseline-I. As shown in Fig.14c, LEM reduces localization error by 23.8% compared to ESVO where event features are matched using naive block-matching operations. Besides, compared with Baseline-I, which exploits event clustering and triangulates the spatial location of an object at cluster-level, LEM reduces the localization error by 55.1%.

6.5 System Efficiency Study

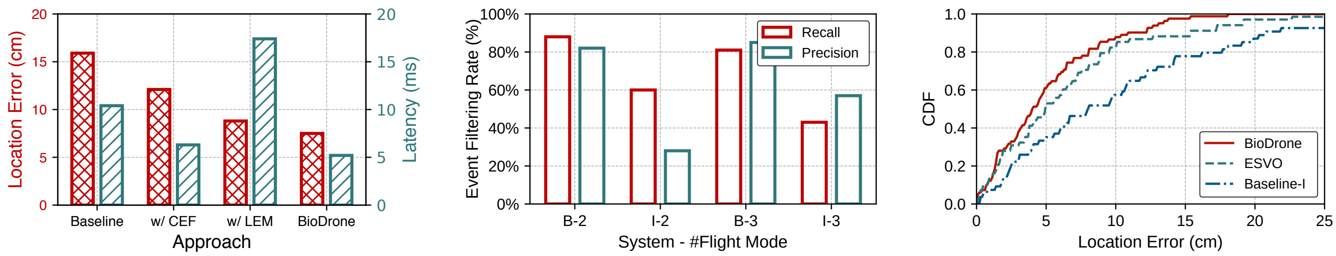
We select a typical 120*ms* obstacle avoidance example and log the system latency, CPU workload, and memory usage in Fig.15. The drone perceives the obstacle at 24*ms*, followed instantly by performing an avoidance maneuver. After 110*ms*, the obstacle disappears from the drone's view.

- At the beginning 24*ms*, LEM is not triggered, and the latency of CEF stays <0.07*ms* with CPU workload <9%.
- Then, during the 24-110*ms*, the drone detects an obstacle and adjusts its own trajectory. CEF introduces a relatively higher computing latency and more CPU workload due to the larger number of events. However, the computing latency of CEF maintains at a low level (i.e., 2.73*ms*) with occupying the CPU workload below 14%. Meanwhile, LEM continuously localizes the obstacle, introducing an additional 2.45*ms* computing latency and around 27% CPU workload.
- Finally, after 110*ms* when the drone has successfully flied away from the obstacle, states of the two modules return to the way it was before 24*ms* in terms of latency and CPU workload. Throughout the obstacle avoidance process, the memory resources occupied by the CEF and LEM are within 5.8 MB and 4.9 MB, respectively.

During the whole avoidance procedure, BioDrone reserves >60% CPU computational resources for upper-layer tasks.

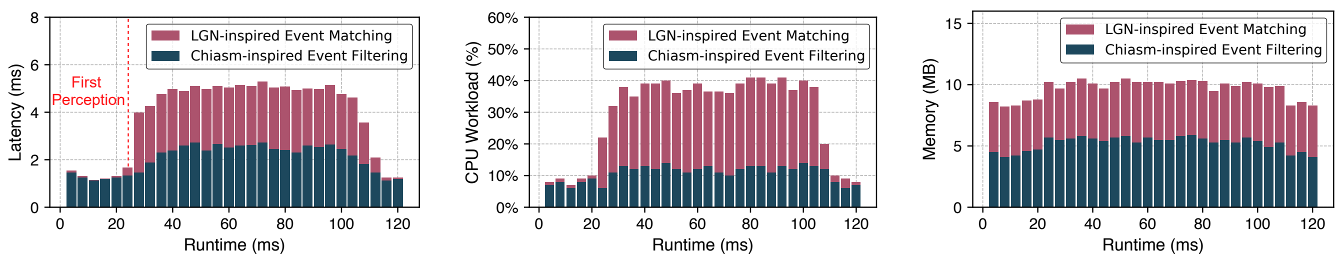
7 RELATED WORK

Object/Obstacle localization. Object localization and tracking attract broad interest within the mobile computing community, with various systems designed around diverse sensors like WiFi [42, 79], CSI [14], acoustic signals [44], RFID [54,



(a) Impact of Different Module (b) Chiasm-inspired Events Filtering (c) LGN-inspired Events Matching

Figure 14: Ablation Study



(a) System Latency (b) CPU Workload (c) Memory Usage

Figure 15: System Efficiency on the Drone

67], IMU [11, 74], camera [75], radar [43], etc.. For the drone obstacle localization task, camera- and radar-based solutions are often preferred in both academia and industry due to their accuracy and widespread implementation[25, 32].

The existing literature relies on frame-based cameras (e.g., monocular [83], stereo [38]), depth cameras [66], millimeter-wave radar [81], LiDAR [8], or VIO [64, 78]. However, these works require the obstacles to be either static or quasistatic. According to our measurements (§2.2), existing solutions are not competent for high-speed drones (i.e., relative speed >$20m/s$). The limitation comes from the sensors' physical nature and cannot be easily solved with algorithms.

Event-based algorithms and systems. Event cameras offer numerous potential advantages over conventional frame-based cameras, including high temporal resolution, low latency, and high dynamic range[10, 26, 29, 52, 65]. Recent systems use them for scene reconstruction [84], SLAM[59], object tracking[24, 49], and HDR image reconstruction [51].

Among these systems, Baseline-I[24] presents a notable obstacle avoidance solution for drones. It utilizes IMU readings to filter out background events, further assisting in obstacle detection, and is most closely related to our work. However, Baseline-I is primarily oriented towards obstacle detection tasks using a monocular event camera setup, whereas our system, BioDrone, aims at obstacle localization leveraging a binocular configuration. This shift in task focus, from detection to localization, combined with the change in hardware configuration, introduces new challenges to fully harness event camera's potential for drone obstacle avoidance, as elaborated in §2.3. In this work, we introduce a novel binocular signal processing pipeline, inspired by the visual pathway, featuring two new modules, CEF and LEM. Our design aims to enhance the accuracy and efficiency of event-based systems, with particular benefits for high-speed drones.

Bio-inspired design for event-based vision. Biological principles drive the design of event camera pixels and some event processing algorithms, such as spiking neural networks (SNN[63]), spatiotemporal oriented filters (STOF[56]), and spike-timing dependent plasticity (STDP[12]). In general, current innovations mainly mimic the working principles of the human visual cortex and design sophisticated algorithms for high-level object recognition[53], segmentation[50], and understanding[9]. Albeit inspiring, these bio-inspired systems are not the optimal solution for obstacle avoidance-related tasks due to the large computational overhead. In BioDrone, we find those delay-sensitive tasks are not executed at the visual cortex but exactly at the earlier *binocular visual pathway*. We take the bio-lessons learned from it and design BioDrone for fast obstacle detection and localization.

8 LIMITATION AND FUTURE WORK

• **Bio-inspired design.** In this study, we leverage biological concepts from the visual pathway to architect a unique signal-processing pipeline for binocular event cameras. We extrapolate the mechanisms of Chiasm and LGN to devise corresponding event filtering and localization algorithms. Note that the understanding of physiological visual perception, including stereoscopic vision, is an ongoing issue, with future findings potentially reshaping our current knowledge. Future designs will evolve based on new research progress.

• **Software-hardware co-design.** In this work, we deploy BioDrone on a commercial Zynq computing platform. By jointly scheduling computational resources of both PL and PS, we manage to boost system accuracy and real-time performance as related works[57, 69]. However, due to implementation complexities, operations related to binocular matching – which require frequent memory access – are still processed in software on the PS, and cannot yet be accelerated by the FPGA. This leaves room for future enhancements.
• **Event camera upgrade.** Event cameras, while promising, are still in their early stages of development and have certain limitations. For instance, their resolution is lower compared to standard cameras (QVGA *vs*. 1080p), restricting the usable FoV and effective observation distance for obstacle avoidance. Also, current event cameras do not support dynamic adjustment of aperture and focus. This limitation necessitates manual calibration during transitions between various work environments to avoid event sampling failure from overexposure or insufficient lighting. However, we expect these issues will be mitigated as event cameras continue to evolve and become integrated into next-generation devices.

9 CONCLUSIONS

We have presented the design and implementation of BioDrone, a software solution to support fast and accurate drone obstacle detection and localization using event cameras. BioDrone exploits biological knowledge behind human visual systems and designs a visual pathway-inspired architecture, a chiasm-inspired event filtering module, and an LGN-inspired event matching mechanism to unleash the full potential of event cameras. Extensive evaluations conducted on an industrial drone demonstrate its superior performance. Through BioDrone, we present that the bio-inspired design paradigm produces simple yet effective solutions to potentially replace heavy-weight ones, adding a new solution dimension for sensing problems with strict restrictions on accuracy, time delay, computation, and energy.

10 ACKNOWLEDGMENTS

We sincerely thank the MobiSense Group, the anonymous reviewers, and our shepherd for their constructive comments and feedback in improving this work. This work is supported in part by the National Key Research Plan under grant No. 2021YFB2900100 and the NSFC under grant No. 61832010.

A APPENDIX

A.1 Modeling of Ego-motion Instruction

Generally, a camera's rotation produces a major number of events than translation in a very short time period δt[24]. Therefore, we focus on estimating the ego-rotation as IMU average angular velocities, $\overline{\omega} = \sum_{\delta t} \omega t/\delta t$. Event e_i could be warped into the frame coordinate at t_0 by $\boldsymbol{\phi} : \mathbb{R}^3 \to \mathbb{R}^3$:

$$\boldsymbol{\phi}(\boldsymbol{x}, t) = \boldsymbol{K}\boldsymbol{R}_t\boldsymbol{K}^{-1}\boldsymbol{x}, \boldsymbol{x} = [u, v]^T, \tag{7}$$

where $\boldsymbol{K}$ is the camera's intrinsic matrix and $\boldsymbol{R}_t$ is the rotation matrix corresponding to $\overline{\omega}t$ given by Rodrigues' rotation formula[18]. We further exploit a projection $\boldsymbol{\Pi} : \mathbb{R}^3 \to \mathbb{R}^2$ to project events onto image coordinates and define the set of wrapped events at t_0 as:

$$\mathcal{E}' = \{(\boldsymbol{\Pi}(\boldsymbol{\phi}(\boldsymbol{x}, t_i - t_0)), t_i, p_i) : (\boldsymbol{x}, t_i, p_i) \in \mathcal{E}\}. \tag{8}$$

Those corresponding events re-projected at pixel (u, v) are:

$$\mathcal{E}'_{uv} = \{(\boldsymbol{x}, t, p) \in \mathcal{E}' : \boldsymbol{x} = (u, v)\}. \tag{9}$$

We further examine the event distribution to distinguish different types of events using a time-based solution. Specifically, the more uniformly the generation time of all events in $\mathcal{E}'_{uv}$, the more likely these events are environment-triggered events, as only moving obstacles will bring an additional time-cluttered event burst on specific pixels. We calculate:

$$T_{uv} = \frac{1}{|\mathcal{E}'_{uv}|}\sum t, \quad (\boldsymbol{x}, t, p) \in \mathcal{E}'_{uv}\,, \tag{10}$$

To eventually separate environment and obstacle events, for each pixel $x = (u, v)$, we calculate a score for it as:

$$\rho_{imu}(\boldsymbol{x}) = \frac{T_{uv} - \overline{T}}{\delta t} \in [-1, 1], \tag{11}$$

where $\overline{T}$ is the average time of all events in $\mathcal{E}$. A higher score indicates a more misaligned generation time of $\mathcal{E}'_{uv}$, and hence the more likely associated events are from an obstacle.

A.2 Modeling of Binocular Constraint

Given an event $\boldsymbol{e}^l = (\boldsymbol{x}^l, t^l, p^l)$ from the left event stream, if it is triggered by the movement of $\boldsymbol{P}$ on the obstacle, there must be a twin event $\boldsymbol{e}^r$ in the right stream with $\boldsymbol{x}^r$:

$$\boldsymbol{x}^r = \boldsymbol{\pi}(\boldsymbol{T}_{rl}, \boldsymbol{P}), \quad \boldsymbol{P} = Z\boldsymbol{K}^{-1}\boldsymbol{x}^l, \tag{12}$$

where Z is the estimated depth of the obstacle, $\boldsymbol{T}_{rl}$ is the transformation from left to right camera (obtained by calibration), and $\boldsymbol{\pi}(\cdot, \cdot)$ is the 3D-2D projection model.

As the estimation of Z by clustering and triangulation is inaccurate, we have to examine a few more pixels centered at $\boldsymbol{x}^r$ and along the epipolar line[82]. The candidate region C (the gray dotted area in the right event stream of Fig.6) is:

$$C = \{(\boldsymbol{x}, t, p) : \boldsymbol{x} \in \boldsymbol{x}^r \pm [\delta x, 0]^T, t \in [t_0, t_0 + \delta t]\}. \tag{13}$$

where δx denotes is set as 5 pixels in practice. We examine all events in C and define the matching score of e^l as:

$$\rho_b(\boldsymbol{x}_l) = \min_{e^r \in C} |t^l - e^r(t)|, \tag{14}$$

where $e^r(t)$ is the timestamp of an event in the right stream, and $\rho_b(\boldsymbol{x}_l)$ is set as ∞ if C is empty. Therein, a small $\rho_b(\boldsymbol{x}_l)$ indicates we could find a matched event associated to $\boldsymbol{P}$ and thus e^l is more likely to be an obstacle-triggered event.

REFERENCES

[1] Ali J Ben Ali, Zakieh Sadat Hashemifar, and Karthik Dantu. 2020. Edge-SLAM: edge-assisted visual simultaneous localization and mapping. In *Proceedings of the ACM Mobisys*.

[2] Usman Ali, Hong Cai, Yasamin Mostofi, and Yorai Wardi. 2018. Motion-communication co-optimization with cooperative load transfer in mobile robotics: An optimal control perspective. *IEEE Transactions on Control of Network Systems* 6, 2 (2018), 621–632.

[3] Inigo Alonso and Ana C Murillo. 2019. EV-SegNet: Semantic segmentation for event-based cameras. In *Proceedings of the IEEE/CVF CVPR*.

[4] Amazon. 2021. Amazon Drones Swarm. https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011.

[5] Ardupilot. 2018. ArduPilot Mega. https://ardupilot.org/.

[6] Abu Bakar, Rishabh Goel, Jasper de Winkel, Jason Huang, Saad Ahmed, Bashima Islam, Przemysław Pawełczak, Kasım Sinan Yıldırım, and Josiah Hester. 2022. Protean: An Energy-Efficient and Heterogeneous Platform for Adaptive and Hardware-Accelerated Battery-Free Computing. In *Proceedings of the ACM SenSys*.

[7] Arjun Balasingam, Karthik Gopalakrishnan, Radhika Mittal, Mohammad Alizadeh, Hamsa Balakrishnan, and Hari Balakrishnan. 2021. Toward a Marketplace for Aerial Computing. In *Proceedings of the ACM DroNet*.

[8] Nikolaos Baras, Georgios Nantzios, Dimitris Ziouzios, and Minas Dasygenis. 2019. Autonomous obstacle avoidance vehicle using lidar and an embedded system. In *Proceedings of the ACM MOCAST*.

[9] Luis A Camuñas-Mesa, Teresa Serrano-Gotarredona, Sio-Hoi Ieng, Ryad Benosman, and Bernabé Linares-Barranco. 2017. Event-driven stereo visual tracking algorithm to solve object occlusion. *IEEE transactions on neural networks and learning systems* (2017).

[10] Marco Cannici, Marco Ciccone, Andrea Romanoni, and Matteo Matteucci. 2019. Asynchronous convolutional networks for object detection in neuromorphic cameras. In *Proceedings of the IEEE/CVF CVPR Workshops*.

[11] Yifeng Cao, Ashutosh Dhekne, and Mostafa Ammar. [n.d.]. ITrackU: tracking a pen-like instrument via UWB-IMU fusion. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*. 453–466.

[12] Natalia Caporale and Yang Dan. 2008. Spike timing–dependent plasticity: a Hebbian learning rule. *Annu. Rev. Neurosci.* (2008).

[13] Ying Chen, Hazer Inaltekin, and Maria Gorlatova. 2023. AdaptSLAM: Edge-Assisted Adaptive SLAM with Resource Constraints via Uncertainty Minimization. In *Proceedings of the IEEE INFOCOM*.

[14] Guoxuan Chi, Zheng Yang, Jingao Xu, Chenshu Wu, Jialin Zhang, Jianzhe Liang, and Yunhao Liu. 2022. Wi-drone: wi-fi-based 6-DoF tracking for indoor drone flight control. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*.

[15] Sandeep Chinchali, Apoorva Sharma, James Harrison, Amine Elhafsi, Daniel Kang, Evgenya Pergament, Eyal Cidon, Sachin Katti, and Marco Pavone. 2021. Network offloading policies for cloud robotics: a learning-based approach. *Autonomous Robots* 45, 7 (2021), 997–1012.

[16] CNet. 2016. Hawk attacks drone in a battle of claw versus machine. www.cnet.com/news/this-hawk-has-no-love-for-your-drone/.

[17] Kathleen E Cullen. 2012. The vestibular system: multimodal integration and encoding of self-motion for motor control. *Trends in neurosciences* 35, 3 (2012), 185–196.

[18] Jian S Dai. 2015. Euler–Rodrigues formula variations, quaternion conjugation and intrinsic connections. *Mechanism and Machine Theory* (2015).

[19] DJI. 2020. DJI Industrial Drones. https://www.dji.com/products/industrial.

[20] DJI. 2022. DJI Matrice 200 v2. https://www.dji.com/matrice-200-series-v2.

[21] Sedat Dogru and Lino Marques. 2020. Pursuing drones with drones using millimeter wave radar. *IEEE Robotics and Automation Letters* 5, 3 (2020), 4156–4163.

[22] Thomas Eppenberger, Gianluca Cesari, Marcin Dymczyk, Roland Siegwart, and Renaud Dubé. 2020. Leveraging stereo-camera data for real-time dynamic obstacle detection and tracking. In *Proceedings of the IEEE/RSJ IROS*.

[23] Lynda Erskine and Eloisa Herrera. 2007. The retinal ganglion cell axon's journey: insights into molecular mechanisms of axon guidance. *Developmental biology* 308, 1 (2007), 1–14.

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

[25] Paula Fraga-Lamas, Lucía Ramos, Víctor Mondéjar-Guerra, and Tiago M Fernández-Caramés. 2019. A review on IoT deep learning UAV systems for autonomous obstacle detection and collision avoidance. *Remote Sensing* (2019).

[26] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. 2020. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence* 44, 1 (2020), 154–180.

[27] Nakul Garg and Nirupam Roy. 2020. Enabling self-defense in small drones. In *Proceedings of the ACM HotMobile Workshop*. 15–20.

[28] Richard I Hartley and Peter Sturm. 1997. Triangulation. *Computer vision and image understanding* 68, 2 (1997), 146–157.

[29] Botao He, Haojia Li, Siyuan Wu, Dong Wang, Zhiwei Zhang, Qianli Dong, Chao Xu, and Fei Gao. 2021. FAST-Dynamic-Vision: Detection and Tracking Dynamic Objects with Event and Depth Sensing. In *Proceedings of the IEEE/RSJ IROS*.

[30] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE ICCV*.

[31] Yuze He, Li Ma, Zhehao Jiang, Yi Tang, and Guoliang Xing. 2021. VI-eye: semantic-based 3D point cloud registration for infrastructure-assisted autonomous driving. In *Proceedings of the ACM MobiCom*.

[32] Jin-wen Hu, Bo-yin Zheng, Ce Wang, Chun-hui Zhao, Xiao-lei Hou, Quan Pan, and Zhao Xu. 2020. A survey on multi-sensor fusion based obstacle detection for intelligent ground vehicles in off-road environments. *Frontiers of Information Technology & Electronic Engineering* (2020).

[33] Dony Hutabarat, Muhammad Rivai, Djoko Purwanto, and Harjuno Hutomo. 2019. Lidar-based obstacle avoidance for the autonomous mobile robot. In *Proceedings of the IEEE ICTS*.

[34] Xilinx Inc. 2022. *MPSoC PS and PL Ethernet Example Projects*. https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/478937213/MPSoC+PS+and+PL+Ethernet+Example+Projects

[35] Aditya Jain, Zerina Kapetanovic, Akshit Kumar, Vasuki Narasimha Swamy, Rohit Patil, Deepak Vasisht, Rahul Sharma, Manohar Swaminathan, Ranveer Chandra, Anirudh Badam, et al. 2019. Low-cost aerial imaging for small holder farmers. In *Proceedings of the ACM Compass*.

[36] Glen Jeffery. 2001. Architecture of the optic chiasm and the mechanisms that sculpt its development. *Physiological Reviews* 81, 4 (2001), 1393–1414.

[37] Sagar Jha, Youjie Li, Shadi Noghabi, Vaishnavi Ranganathan, Peeyush Kumar, Andrew Nelson, Michael Toelle, Sudipta Sinha, Ranveer Chandra, and Anirudh Badam. 2021. Visage: enabling timely analytics for drone imagery. In *Proceedings of the ACM MobiCom*.

[38] Vicky S Kalogeiton, Konstantinos Ioannidis, G Ch Sirakoulis, and Elias B Kosmatopoulos. 2019. Real-time active SLAM and obstacle avoidance for an autonomous robot based on stereo vision. *Cybernetics*

and Systems (2019).

[39] Hanme Kim, Stefan Leutenegger, and Andrew J Davison. 2016. Real-time 3D reconstruction and 6-DoF tracking with an event camera. In *Proceedings of the Springer ECCV*.

[40] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E Shi, and Ryad B Benosman. 2016. Hots: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE transactions on pattern analysis and machine intelligence* (2016).

[41] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. 2009. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision* 81, 2 (2009), 155.

[42] Danyang Li, Jingao Xu, Zheng Yang, Yumeng Lu, Qian Zhang, and Xinglin Zhang. 2021. Train once, locate anytime for anyone: Adversarial learning based wireless localization. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*.

[43] Danyang Li, Jingao Xu, Zheng Yang, Qian Zhang, Qiang Ma, Li Zhang, and Pengpeng Chen. 2022. Motion Inspires Notion: Self-supervised Visual-LiDAR Fusion for Environment Depth Estimation. In *Proceedings of the ACM MobiSys*.

[44] Kaikai Liu, Xinxin Liu, and Xiaolin Li. 2013. Guoguo: Enabling fine-grained indoor localization via smartphone. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*.

[45] Sikang Liu, Michael Watterson, Sarah Tang, and Vijay Kumar. 2016. High speed navigation for quadrotors with limited onboard sensing. In *Proceedings of the IEEE ICRA*.

[46] Chris Xiaoxuan Lu, Stefano Rosa, Peijun Zhao, Bing Wang, Changhao Chen, John A Stankovic, Niki Trigoni, and Andrew Markham. 2020. See through smoke: robust indoor mapping with low-cost mmWave radar.. In *Proceedings of the ACM MobiSys*.

[47] Yunfei Ma, Nicholas Selby, and Fadel Adib. 2017. Drone relays for battery-free networks. In *Proceedings of the ACM Sigcomm*.

[48] Daily Mail. 2016. When eagles attack! drone camera mistaken for rival. www.dailymail.co.uk/video/news/video-1154408/Golden-Eagle-attacks-drone-cameramistaking-rival.html.

[49] Anton Mitrokhin, Cornelia Fermüller, Chethan Parameshwara, and Yiannis Aloimonos. 2018. Event-based moving object detection and tracking. In *Proceedings of the IEEE IROS*.

[50] Anton Mitrokhin, Zhiyuan Hua, Cornelia Fermuller, and Yiannis Aloimonos. 2020. Learning visual motion segmentation using event surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14414–14423.

[51] Mohammad Mostafavi, Lin Wang, and Kuk-Jin Yoon. 2021. Learning to reconstruct hdr images from events, with applications to depth and flow prediction. *International Journal of Computer Vision* (2021).

[52] Yeongwoo Nam, Mohammad Mostafavi, Kuk-Jin Yoon, and Jonghyun Choi. 2022. Stereo Depth from Events Cameras: Concentrate and Focus on the Future. In *Proceedings of the IEEE/CVF CVPR*. 6114–6123.

[53] Ying Nan, Rong Xiao, Shaobing Gao, and Rui Yan. 2019. An event-based hierarchy model for object recognition. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2342–2347.

[54] Isura Nirmal, Abdelwahed Khamis, Mahbub Hassan, Wen Hu, and Xiaoqing Zhu. 2021. Deep learning for radio-based human sensing: Recent advances and future directions. *IEEE Communications Surveys & Tutorials* (2021).

[55] OpoenAMP. 2022. *OpenAMP Project*. https://www.openampproject.org/

[56] Garrick Orchard, Ryad Benosman, Ralph Etienne-Cummings, and Nitish V Thakor. 2013. A spiking neural network architecture for visual motion estimation. In *2013 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 298–301.

[57] Nhat Pham, Hong Jia, Minh Tran, Tuan Dinh, Nam Bui, Young Kwon, Dong Ma, Phuc Nguyen, Cecilia Mascolo, and Tam Vu. 2022. PROS: an efficient pattern-driven compressive sensing framework for low-power biopotential-based wearables with on-chip intelligence. In *Proceedings of the ACM MobiCom*.

[58] Tong Qin, Peiliang Li, and Shaojie Shen. 2018. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *Proceedings of the IEEE Transactions on Robotics* (2018).

[59] Henri Rebecq, Timo Horstschäfer, Guillermo Gallego, and Davide Scaramuzza. 2016. Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time. *IEEE Robotics and Automation Letters* 2, 2 (2016), 593–600.

[60] Bernard Schmidt and Lihui Wang. 2014. Depth camera based collision avoidance via active robot control. *Journal of manufacturing systems* 33, 4 (2014), 711–718.

[61] Ramanujan K Sheshadri, Eugene Chai, Karthikeyan Sundaresan, and Sampath Rangarajan. 2021. SkyHAUL: A Self-Organizing Gigabit Network In The Sky. In *Proceedings of the ACM MobiHoc*.

[62] Chris Tailby, Soon Keen Cheong, Alexander N Pietersen, Samuel G Solomon, and Paul R Martin. 2012. Colour and pattern selectivity of receptive fields in superior colliculus of marmoset monkeys. *The Journal of Physiology* 590, 16 (2012), 4061–4077.

[63] Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothée Masquelier, and Anthony Maida. 2019. Deep learning in spiking neural networks. *Neural networks* (2019).

[64] Catherine Tong, Jinchen Ge, and Nicholas D Lane. 2021. Zero-shot learning for imu-based activity recognition using video embeddings. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 4 (2021), 1–23.

[65] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. 2018. Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios. *IEEE Robotics and Automation Letters* 3, 2 (2018), 994–1001.

[66] Dashuai Wang, Wei Li, Xiaoguang Liu, Nan Li, and Chunlong Zhang. 2020. UAV environmental perception and autonomous obstacle avoidance: A deep learning and depth camera combined solution. *Computers and Electronics in Agriculture* (2020).

[67] Jue Wang and Dina Katabi. 2013. Dude, where's my card? RFID positioning that works with multipath and non-line of sight. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*.

[68] Weiguo Wang, Luca Mottola, Yuan He, Jinming Li, Yimiao Sun, Shuai Li, Hua Jing, and Yulei Wang. 2022. MicNest: Long-Range Instant Acoustic Localization of Drones in Precise Landing. (2022).

[69] Zeyu Wang, Jingao Xu, Xu Wang, Xiangwen Zhuge, Xiaowu He, and Zheng Yang. 2023. Industrial Knee-jerk: In-Network Simultaneous Planning and Control on a TSN Switch. In *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*. 356–369.

[70] Felix Wimbauer, Nan Yang, Lukas Von Stumberg, Niclas Zeller, and Daniel Cremers. 2021. MonoRec: Semi-supervised dense reconstruction in dynamic environments from a single moving camera. In *Proceedings of the IEEE/CVF CVPR*.

[71] Kwame-Lante Wright, Ashiwan Sivakumar, Peter Steenkiste, Bo Yu, and Fan Bai. 2020. Cloudslam: Edge offloading of stateful vehicular applications. In *Proceedings of the IEEE/ACM SEC*.

[72] Xilinx. 2022. Direct Memory Access. https://www.xilinx.com/products/intellectual-property/axi_dma.html.

[73] Xilinx. 2022. Xilinx Zynq-7000 SoC. https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.

[74] Huatao Xu, Pengfei Zhou, Rui Tan, Mo Li, and Guobin Shen. 2021. Limu-bert: Unleashing the potential of unlabeled data for imu sensing applications. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*.

[75] Jingao Xu, Hao Cao, Zheng Yang, Longfei Shangguan, Jialin Zhang, Xiaowu He, and Yunhao Liu. 2022. SwarmMap: Scaling Up Real-time Collaborative Visual SLAM at the Edge. In *Proceedings of the USENIX NSDI*.

[76] Jingao Xu, Guoxuan Chi, Zheng Yang, Danyang Li, Qian Zhang, Qiang Ma, and Xin Miao. 2021. FollowUpAR: Enabling Follow-up Effects in Mobile AR Applications. In *Proceedings of the ACM MobiSys*.

[77] Zhenbo Xu, Wei Zhang, Xiaoqing Ye, Xiao Tan, Wei Yang, Shilei Wen, Errui Ding, Ajin Meng, and Liusheng Huang. 2020. Zoomnet: Part-aware adaptive zooming neural network for 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 12557–12564.

[78] Mingyu Yang, Yu Chen, and Hun-Seok Kim. 2022. Efficient deep visual and inertial odometry with adaptive visual modality selection. In *Proceedings of the Springer ECCV*.

[79] Zheng Yang, Chenshu Wu, and Yunhao Liu. 2012. Locating in Fingerprint Space: Wireless Indoor Localization with Little Human Intervention. In *Proceedings of the ACM MobiCom*.

[80] Jawad N Yasin, Sherif AS Mohamed, Mohammad-Hashem Haghbayan, Jukka Heikkonen, Hannu Tenhunen, and Juha Plosila. 2021. Low-cost ultrasonic based object detection and collision avoidance method for autonomous robots. *International Journal of Information Technology* 13, 1 (2021), 97–107.

[81] Hang Yu, Fan Zhang, Panfeng Huang, Chen Wang, and Li Yuanhao. 2020. Autonomous obstacle avoidance for uav based on fusion of radar and monocular camera. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[82] Zhengyou Zhang. 1998. Determining the epipolar geometry and its uncertainty: A review. *International journal of computer vision* (1998).

[83] Zhouyu Zhang, Yunfeng Cao, Meng Ding, Likui Zhuang, and Jiang Tao. 2020. Monocular vision based obstacle avoidance trajectory planning for Unmanned Aerial Vehicle. *Aerospace Science and Technology* (2020).

[84] Yi Zhou, Guillermo Gallego, and Shaojie Shen. 2021. Event-based stereo visual odometry. *IEEE Transactions on Robotics* 37, 5 (2021), 1433–1450.

[85] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. 2017. Event-based feature tracking with probabilistic data association. In *Proceedings of the IEEE ICRA*.