Blurring for Clarity: Passive Computation for Defocus-Driven Parsimonious Navigation using a Monocular Event Camera

Hrishikesh Pawar * Deepak Singh * Nitin J Sanket

Abstract

Navigation in cluttered, unstructured scenes is crucial for deploying aerial robots in humanitarian applications. To enhance efficiency and extend operational times, we propose a biologically inspired method called passive computation. By utilizing wave physics, we extract depth cues from defocus instead of relying on costly explicit depth computation. We demonstrate this approach by using a large aperture lens to get a shallow depth of field on a monocular event camera, enabling robust and parsimonious navigation through depth ordinality. The key idea is to optically "blur out" regions of disinterest, minimizing computational demands. In simulation experiments, our method achieved a success rate of 70% with over 62× computation savings compared to state-of-the-art techniques. Preliminary results on a real setup also show promise, highlighting the potential of defocus in enhancing event-based navigation for aerial robots.

1. Introduction

Tiny aerial robots, known for their agility, gap navigation, cost efficiency, and scalability in swarms, remain underutilized in critical applications such as search and rescue, reconnaissance, and anti-poaching. This is primarily due to limitations in onboard sensing and computation and reliance on external infrastructure, which restricts their deployment in unstructured environments. In contrast, nature's expert flyers, like insects and small birds, navigate dense, cluttered spaces such as forests using parsimonious strategies to extract depth cues, bypassing the need for resource-intensive 3D reconstructions commonly employed in robotics.

Inspired by nature, we coin the term *passive computation*, which leverages custom optical elements to exploit wave physics and induce depth-based cues by modifying the incoming visual signal without requiring electrical power. Specifically, we employ an event camera paired with a large aperture lens to mimic the vision of a single ommatidium (albeit with higher resolution) with a fixed focus. Our approach investigates the feasibility of such a minimalistic method for estimating depth cues to aid aerial navigation.

Figure 1. Illustration of the proposed approach in a real-world scene with a hand-held camera setup. The yellow frame (right) captures the tree positioned near the camera's focus distance, and the white frame (left) shows the tree positioned farther from the focus distance. For each frame, the corresponding events and the calculated sharpness maps are shown on each side's top and bottom corners. Notice that the difference between the two regions' sharpness maps (more white/sharp regions in yellow frame) shows how depth cues can be perceived using the proposed approach.

Optical defocus and Point Spread Functions (PSFs) have attracted growing interest within the event-camera community over the past decade but remain underexplored due to the coupling of events with ego-motion. Early work by [1] introduced a Spiking Neural Network (SNN) model for depth estimation in static scenes, leveraging defocus using varifocal liquid lenses. Subsequent studies, such as [2], utilized event rates as focus metrics for autofocus mechanisms, while [3] integrated events across focal planes with deep learning for sparse depth estimation. More recently, [4] proposed a theoretical framework for 3D localization and tracking using custom-aperture event cameras.

Traditionally, event cameras are paired with small-aperture lenses that produce sharp All-In-Focus (AIF) im-

*Equal Contribution. Authors are with the Perception and Autonomous Robotics (PeAR) Group, Robotics Engineering, Worcester Polytechnic Institute, Worcester, MA 01609 USA (Email: hpawar@wpi.edu; dsingh1@wpi.edu; nitin@wpi.edu).

ages, adhering to the pinhole camera model. While effective, this configuration generates large volumes of events, including those from irrelevant regions. In contrast, we couple a large-aperture lens (f/1.6) with a single monocular event camera at a fixed focus, intentionally introducing optical defocus to "blur out" background regions. Similar to the portrait blur or *bokeh* effect in photography, this design reduces events from defocused areas, decreasing computational load and latency while maintaining performance comparable to explicit depth estimation methods. Figure 1 illustrates our approach of blurring background regions in a real-world scene.

For navigation, our method leverages these "optically out-of-focus" regions as a guiding signal, encouraging movement towards them. This approach contrasts with conventional depth-based navigation, where a full scene-depth map is constructed, followed by motion planning to avoid obstacles. Our method is purposive and parsimonious, tailored specifically for aerial navigation. To the best of our knowledge, this is the first work to formalize the concept of *passive computation* and apply it to model optical defocus for event cameras in mobile robotic applications.

While event cameras have been extensively used for dynamic obstacle avoidance [5, 6], their application to static scenes remains underexplored. This is largely due to the dominance of motion-induced events, making it challenging to discern obstacles from free-space regions. The closest work to ours, [7], employs a learning-based framework for quadrotor navigation in static environments by predicting dense depth maps from events. However, this method is computationally intensive for small robots. To overcome these challenges, we propose a lightweight approach for navigating dense, cluttered forests using a large-aperture lens and simple mathematical processing on event data (see §2). Our method achieves a **70%** success rate with a runtime of **16ms** in simulation on an Intel®i7 10th-generation CPU. Our key contributions are highlighted next:

- *Passive Computation for Navigation:* We define and apply *passive computation* to aerial navigation, using custom optical elements to induce depth-based cues without electrical power or intensive processing.
- *Efficient Depth Cues via Optical Defocus:* By pairing a large-aperture lens (f/1.6) with an event camera, we minimize irrelevant events from out-of-focus regions, significantly reducing computation while maintaining navigation performance.
- *Lightweight Navigation in Dense Environments:* Our method bypasses traditional depth-mapping, achieving a **70%** success rate with **62**$\times$ computational savings compared to state-of-the-art methods, offering a scalable solution for small aerial robots navigating cluttered environments.

2. Passive Computation For Navigation

We define *passive computation* as a method that leverages passive components (requiring no electrical power) with a sensor to induce signal characteristics through wave physics.

Our approach uses a large-aperture lens with an event camera to introduce depth-dependent optical blur, modulating event rates based on scene depth. The lens is focused at a predefined distance Z_f, chosen based on the robot's size, dynamic constraints, and safety margin. The large aperture creates a shallow Depth of Field (DoF), restricting the in-focus region to a specific depth range, enabling the robot to prioritize immediate obstacles while disregarding the background. Foreground objects, within $Z_f \pm 0.5\,\text{DoF}$, are segmented using focus-based depth cues, combining ordinal depth (foreground closer than background) with metric depth (foreground within a quantifiable range). This segmentation forms the basis of our parsimonious navigation strategy, ensuring efficient obstacle avoidance in cluttered environments. The DoF of a camera focused at Z_f is given by

$$DoF = \left(2NZ_f^2 c\right)/f^2 \tag{1}$$

where f is the focal length, N is the aperture number, and c is the circle of confusion [8, 9]. Further,

$$c \propto \left(|Z - Z_f|\right)/Z_f \tag{2}$$

We will now discuss a mathematical approach to perform foreground-background segmentation.

2.1. Defocus in Events

Event cameras operate asynchronously, with each pixel independently generating events when the change in logarithmic intensity exceeds a predefined threshold τ [10].

$$||\log(I(\mathbf{x}, t+\delta t))) - \log(I((\mathbf{x}, t))||_1 \geq \tau \tag{3}$$

where $I(\mathbf{x}, t)$ is the intensity at pixel coordinate ($\mathbf{x} = \begin{bmatrix} x & y \end{bmatrix}^\top$) at time t, δt is the time since the previous event at the same location.

Many events $\mathbf{e}_i$ are generated by the event camera in the time window $[T, T + \Delta T]$. All the events generated in a spatio-temporal window $\mathcal{E}$ is given by

$$\mathcal{E} = \{\mathbf{e}_k\} = \{(\mathbf{x}, t, p)_k\} \tag{4}$$

Here, k indexes through the events and p is event the polarity. Further we can obtain a subset $\mathcal{E}_{\mathcal{X},\mathcal{Y},\mathcal{T}} \subseteq \mathcal{E}$ by indexing between ranges of valid $x \in \mathcal{X}$, $y \in \mathcal{Y}$ and $t \in \mathcal{T}$.

Since, we are concerned about finding the foreground regions that are sharp, we leverage the large body of work on focal measure of event volumes [2, 3]. Inspired from EV-DodgeNet [5] and [11], we construct event frames $\mathcal{E}$ in a

spatio temporal neighborhood $[\mathcal{X}, \mathcal{Y}, \mathcal{T}]$ with the following difference. We treat both polarities equally (by taking absolute value) resulting in a single 2D event frame $\mathcal{E}$.

Previous works aim to generate All-In-Focus (AIF) event frames; however, longer integration times led to artificial motion blur, as highlighted in EVDodgeNet. This issue was addressed in [5] using the EVDeblurNet network on the event frame $\mathcal{E}$. Another common approach is motion compensation on the event volume $\mathcal{E}$, as explored in [12, 13].However, these approaches are computationally intensive and unsuitable for small robots. In our lightweight approach, we focus solely on the relative sharpness between the foreground and background, which depends on factors such as integration time ΔT, camera optics, robot movement $[V, \Omega]^T$, and the sensor noise floor η. Here, ΔT and optics are design parameters constrained by practical considerations, which ultimately place an upper bound on the robot's velocity, as mentioned in **Remarks 2** and **3**.

Events are generated according to log intensity change described in Eq.3. Here intensity I is never directly observed but is a latent variable that the sensor perceives.

In a scene with consistent lighting, events are generated only due to camera motion. This means the intensity at any coordinate in the event frame changes with space and time. The rate of change of intensity with time is given by

$$\frac{dI}{dt} = \frac{\partial I}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial I}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial I}{\partial t} = \dot{x}\frac{\partial I}{\partial x} + \dot{y}\frac{\partial I}{\partial y} \tag{5}$$

where $\frac{\partial I}{\partial t} = 0$, since we assume that the scene lighting is constant. Here, $\dot{x}$ and $\dot{y}$ are pixel velocities in corresponding directions. This equation can also be represented as

$$\frac{dI}{dt} = \mathbf{v} \cdot \nabla_{\mathbf{x}} I \tag{6}$$

where $\nabla_{\mathbf{x}} I = [\partial I/\partial x, \partial I/\partial y]^T$ represents the spatial image gradient, quantifying the rate of intensity change in the x and y directions. The term $\mathbf{v} = (\dot{x}, \dot{y})$ denotes the pixel velocity or optical flow, describing the apparent motion of pixels on the image plane [14]. Events are generated when $|dI/dt| \geq \tau$ (from Eq.3). Hence, from Eq.6, the event rate depends on pixel velocity (which is related to the 3D velocity of the sensor/robot [14]) and spatial intensity gradients. The following four remarks highlight the effects of design parameters on foreground-background segmentation for static scene navigation.

Remark 1 *Smoothing caused by any Point Spread Function (PSF) attenuates high-frequency components, leading to a reduction in contrast and, consequently, a decrease in the number of events generated from the optically blurred regions.*

Remark 2 *The event integration time (ΔT) must decrease with increasing drone velocity (V) to preserve a high Signal-to-Noise Ratio (SNR), ensuring effective segmentation of sharp foreground objects from the blurred background. Here, signal refers to the sharp foreground, and noise is the blurred background.*

Remark 3 *There exists a lower limit to the event integration time ΔT, below which the information obtained from the event camera becomes insufficient and is dominated by noise.*

Remark 4 *Ratio of sharpness of foreground to background regions reduces as Z_f increases. Increasing the focal length f in proportion to Z_f can mitigate this problem, enabling high fidelity foreground-background segmentation.*

2.2. Foreground Segmentation Implementation Details

As mentioned in § 2.1, we utilize event frames $\mathcal{E}$ constructed using an integration time of $\Delta T = 1ms$ for segmentation. Inspired from [12], the sharpness/focus measure F used is given by spatial variance in a window size of 32×32 in $\mathcal{E}$. F maps event frame $\mathcal{E}$ to focal frame $\mathcal{F}$, i.e., $F : \mathcal{E} \rightarrow \mathcal{F}$.

2.3. Navigation Policy

Inspired from Ajna [15], we find free space in the image plane by adaptively thresholding $\mathcal{F}$. The high sharpness areas correspond to foreground pixels and low sharpness areas (blurry regions) correspond to background. Here, foreground pixels are the obstacles we need to dodge. A simple control policy in velocity space to move in a direction to align the robot with the free space center in the image plane is used. A simple Proportional-Integral-Derivative (PID) controller is used to track this desired velocity.

3. Experiments

We validate our approach using a set of extensive simulation experiments and preliminary results in the real-world which are explained next.

3.1. Evaluation Metrics

We benchmark our method against state-of-the-art navigation approaches using varying input modalities, including metric depth, optical flow (relative depth), and ordinal depth with boundary constraints, using the same navigation policy across all methods for fairness. Robots with richer information can avoid obstacles more effectively and from closer distances, while lower-information setups represent smaller robots with limited sensors and compute. We use the depth map from Blender as ground truth depth, and the trajectory obtained using this depth map, as the ground truth trajectory. Evaluation metrics, adapted from Ajna [15] and EdgeFlowNet [16], include Success Rate (SR), Average Path Length Increase (PLI), and Run Time (ms), with PLI calculated relative to the ground truth trajectory.

3.2. Simulation results

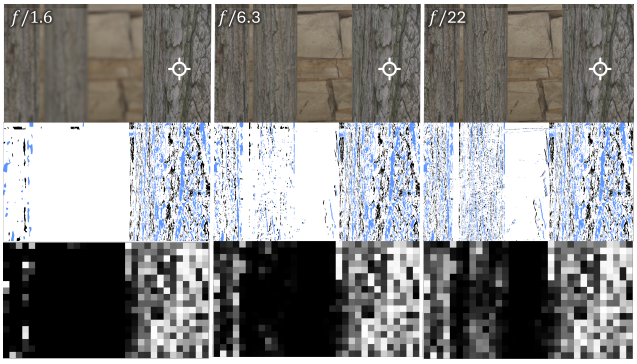

Figure 2. Variation of sharpness maps with aperture number (N). *Rows Top to Bottom*: (1) Rendered images from the Blender environment, (2) Generated events from the frames, and (3) Calculated sharpness maps from the events. *Columns Left to Right* represent increasing N, with the highlighted point indicating the focus point. *Notice higher sharpness with going from left to right and how the foregeound and background trees look similar in the third column.*

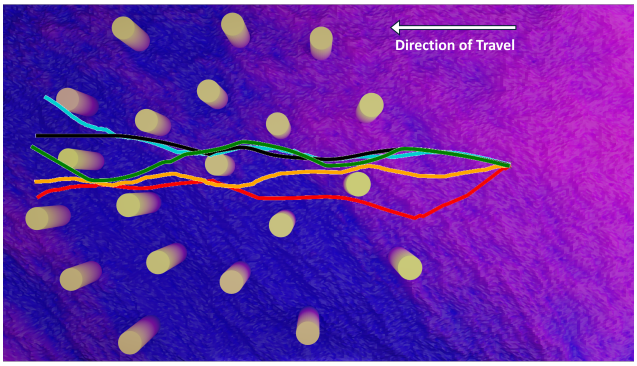

Figure 3. Comparison of various navigation methods: **Ground truth depth**, **Depth Pro**, **MiDaS-v2.1Small**, **RAFT**, ***Ours***.

We evaluated our method in a custom Blender® simulator inspired by [5, 15, 16], featuring a room-like structure with a bumpy floor and randomized cylindrical obstacles textured as tree trunks for realism (see Fig.2, top row). Event streams were generated using [17] from rendered frames with a $35mm$ lens (54.4° field of view) and a $36mm$ sensor at 640×480 px resolution, rendered via Blender's EEVEE engine. Camera focus (Z_f) was set to 1.5 metres for our method. Depth of Field (DoF) blur was applied using Blender's built-in function, with an aperture of $f/1.6$ for our method and $f/22$ for others simulating pinhole cameras. We benchmarked against Depth Pro [18], MiDaS-v2.1-small [19], and RAFT [20], running simulations on an Intel®i7 CPU (10th gen, 64 GB RAM) and NVIDIA RTX 3090 Ti GPU, with our method executed on the CPU and others on the GPU. Table 1 reports performance over 10 trials with randomized start and goal coordinates for robust evaluation. Figure 3 illustrates sample top-down trajectories for all methods from one evaluation starting point.

Method	SR (%) ↑	PLI (%)↓	Run time (ms)
Depth-Pro*	**90**	1.5	1001.12
MiDaS-v2.1S*	50	2.5	350.125
RAFT*	80	2.8	96.57
Ours†	70	**0.6**	**16.894**

Table 1. Quantitative evaluation for simulation experiments. * run on GPU, † run on CPU

Table 1 shows our approach achieves a 70% success rate with the lowest runtime, even on a CPU, outperforming GPU-based methods with a speedup of at least $62\times$ over Depth Pro. This highlights the computational efficiency and effectiveness of our method.

The performance of our approach is influenced by aperture size, which affects the depth of field and the sharpness contrast between foreground and background objects. As shown in Fig.2, the sharpness map varies with the aperture number (N), with larger apertures providing clearer segmentation of sharp foreground objects. Hence, we selected an $f/1.6$ aperture for both simulations and preliminary hardware experiments.

3.3. Preliminary Real-world results

We tested our approach on a real system using a 35mm $f/1.6$ Arducam C-mount lens paired with an Inivation DVXplorer event camera (640×480 px). To address its limited horizontal field of view, we added a Vivitar $0.43\times$ focal length reducer, achieving an effective field of view of 12.1°. The camera was focused at 1.1m during experiments. As shown in Fig.1, the sharpness map effectively distinguishes foreground from background when the tree is near the focus distance (yellow frame). These results are promising, setting the stage for future exploration of navigation strategies on aerial robots.

4. Conclusion

We proposed a passive computation approach for parsimonious navigation with event cameras, leveraging defocus as a depth cue. Pairing a large-aperture lens with a simple mathematical formulation for sharpness in event space, we effectively segmented immediate foreground obstacles from distant ones. This segmentation, combined with a basic navigation policy, achieved a 70% success rate in static scene simulations with a runtime of just 16 ms on a CPU. Preliminary hardware results using an event camera with a compound lens setup further validate our method. Future work will focus on improving robustness in real-world environments and further reducing runtime for greater efficiency.

References

[1] Germain Haessig, Xavier Berthelon, Sio-Hoi Ieng, and Ryad Benosman. A spiking neural network model of depth from defocus for event-based neuromorphic vision. *Scientific Reports*, 9(1):3744, 2019. 1

[2] Shijie Lin, Yinqiang Zhang, Lei Yu, Bin Zhou, Xiaowei Luo, and Jia Pan. Autofocus for event cameras. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16323–16332, 2022. 1, 2

[3] Shiming Wang, Jian Yu, and Xiaoping Chen. Learning monocular depth from focus with event focal stack. *arXiv preprint arXiv:2405.06944*, 2023. 1, 2

[4] Sachin Shah, Matthew A. Chan, Haoming Cai, Jingxi Chen, Sakshum Kulshrestha, Chahat Deep Singh, Yiannis Aloimonos, and Christopher A. Metzler. Codedevents: Optimal point-spread-function engineering for 3d-tracking with event cameras. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25265–25275, 2024. 1

[5] Nitin J. Sanket, Chethan M. Parameshwara, Chahat Deep Singh, Ashwin V. Kuruttukulam, Cornelia Fermüller, Davide Scaramuzza, and Yiannis Aloimonos. Evdodgenet: Deep dynamic obstacle dodging with event cameras. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10651–10657, 2020. 2, 3, 4

[6] Davide Falanga, Kevin Kleber, and Davide Scaramuzza. Dynamic obstacle avoidance for quadrotors with event cameras. *Science Robotics*, 5(40):eaaz9712, 2020. 2

[7] Anish Bhattacharya, Marco Cannici, Nishanth Rao, Yuezhan Tao, Vijay Kumar, Nikolai Matni, and Davide Scaramuzza. Monocular event-based vision for obstacle avoidance with a quadrotor, 2024. 2

[8] Joseph W. Goodman. *Introduction to Fourier Optics*. W.H. Freeman, 3rd edition, 2005. 2

[9] Alfred A. Blaker. *Applied Depth of Field*. Focal Press, 1985. 2

[10] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008. 2

[11] 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)*, June 2019. 2

[12] Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3867–3876, 2018. 3

[13] Guillermo Gallego, Mathias Gehrig, and Davide Scaramuzza. Focus Is All You Need: Loss Functions for Event-Based Vision . In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12272–12281, Los Alamitos, CA, USA, June 2019. IEEE Computer Society. 3

[14] A. Verri and T. Poggio. Motion field and optical flow: qualitative properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):490–498, 1989. 3

[15] Nitin J. Sanket, Chahat Deep Singh, Cornelia Fermüller, and Yiannis Aloimonos. Ajna: Generalized deep uncertainty for minimal perception on parsimonious robots. *Science Robotics*, 8(81):eadd5139, 2023. 3, 4

[16] Sai Ramana Kiran Pinnama Raju, Rishabh Singh, Manoj Velmurugan, and Nitin J. Sanket. Edgeflownet: 100fps@1w dense optical flow for tiny mobile robots. *IEEE Robotics and Automation Letters*, 10(1):128–135, 2025. 3, 4

[17] Daniel Gehrig, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Video to events: Recycling video datasets for event cameras. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, June 2020. 4

[18] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R. Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second, 2024. 4

[19] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R. Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second, 2024. 4

[20] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 402–419, Cham, 2020. Springer International Publishing. 4