2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Prepare for Warp Speed: Sub-millisecond Visual Place Recognition Using Event Cameras

Vignesh Ramanathan Michael Milford Tobias Fischer

***Abstract*— Visual Place Recognition (VPR) enables systems to identify previously visited locations within a map, a fundamental task for autonomous navigation. Prior works have developed VPR solutions using event cameras, which asynchronously measure per-pixel brightness changes with microsecond temporal resolution. However, these works rely on dense representations of the inherently sparse camera output and require tens to hundreds of milliseconds of event data to predict a place. Here, we break this paradigm with Flash, a lightweight VPR system that predicts places using sub-millisecond slices of event data. Our method is based on the observation that active pixel locations provide strong discriminative features for VPR. Flash encodes these active pixel locations using efficient binary frames and computes similarities via fast bitwise operations, which are then normalized based on the relative event activity in the query and reference frames. Flash improves Recall@1 for sub-millisecond VPR over existing baselines by $11.33\times$ on the indoor QCR-Event-Dataset and $5.92\times$ on the 8 km Brisbane-Event-VPR dataset. Moreover, our method reduces the duration for which the robot must operate without awareness of its position, as evidenced by a localization latency metric we term Time to Correct Match (TCM). To the best of our knowledge, this is the first work to demonstrate sub-millisecond event-based VPR.**

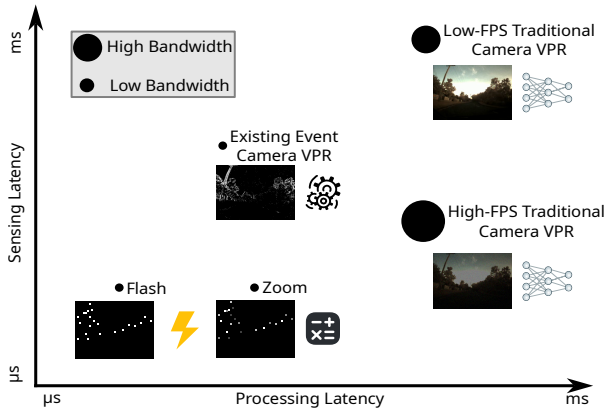

Fig. 1. Traditional cameras face a bandwidth–latency trade-off: higher frame rates demand more bandwidth, and processing dense frames is costly. Event cameras, by contrast, capture sparse scene changes at microsecond resolution. Existing methods, however, still accumulate tens to hundreds of milliseconds of events, squandering this speed. Flash exploits sub-millisecond slices of event data, using primarily bitwise operations to minimize both sensing and processing latencies.

I. Introduction

Visual place recognition (VPR) is a fundamental capability for autonomous systems, allowing them to localize and navigate by recognizing previously visited locations [1]–[5]. For high-speed robots, achieving low-latency place matching is crucial to reduce the time without a position estimate and improve safety in applications such as self-driving vehicles [6] and facilitating GPS-denied navigation [7]. The latency of a VPR system arises from both sensing and processing delays. To address processing latency, traditional camera-based methods have investigated lightweight architectures, bioinspired models, and specialized visual cues to increase VPR speed while preserving accuracy [8]–[10]. However, conventional cameras remain limited by bandwidth-latency trade-offs, constraining their usability in mobile robotic applications.

Event cameras are neuromorphic sensors that asynchronously detect per-pixel logarithmic changes in scene brightness. Unlike traditional cameras, event cameras sense the visual scene with microsecond resolution in a bandwidth-efficient manner [11], enabling sub-millisecond perception and localization, which are critical for high-speed robotic applications. While some vision tasks leverage these advantages [6], [12], [13], event camera-based VPR methods typically accumulate events over tens to hundreds of milliseconds to construct dense representations, thus failing to exploit the microsecond temporal resolution of event cameras [14]–[19]. In this work, we challenge this paradigm by demonstrating that the *spatial distribution of events alone*, even over sub-millisecond windows, contains sufficient discriminative information for place recognition.

In this work, we introduce an efficient, ultra-low-latency VPR approach, termed *Flash*. Our approach reduces sensing latency by relying on only sub-millisecond slices of event data. In the resulting event frames, fewer than 3% of pixels are active per frame in the indoor dataset and fewer than 9% in the outdoor dataset, with the average number of events per active pixel approaching one (Section VI-D). This makes a binary frame representation both a natural and a highly compact choice (Fig. 1).

While the binary representation exploits event sparsity, it also introduces an aliasing bias: reference frames with more active pixels tend to yield higher overlap scores, regardless of true similarity. To address this, we introduce a lightweight normalization that scales the similarity score by the ratio of active pixel counts between query and reference frames, mitigating the aliasing bias using only two floating-point operations per pair. Our design enables both sensing and

All authors are with the QUT Centre for Robotics, School of Electrical Engineering and Robotics, Queensland University of Technology, Brisbane, QLD 4000, Australia. This research was partially supported by funding from ARC Laureate Fellowship FL210100156 to MM and ARC DECRA Fellowship DE240100149 to TF. The authors acknowledge continued support from the Queensland University of Technology (QUT) through the Centre for Robotics. Corresponding author email: vignesh.ramanathan@hdr.qut.edu.au

processing to happen in a Flash.

Operating in the sub-millisecond regime poses a new challenge: the extremely short event windows produce a large number of reference frames, increasing both latency and computational cost. To address this, we propose uniformly sampling frames from the reference database to reduce its size while retaining sufficient map coverage. We further evaluate this strategy to characterize the trade-off between recognition accuracy and computational efficiency.

Our contributions can be summarized as follows:

1) We introduce the first event camera-based VPR system that recognizes places using less than a millisecond of event data.
2) We show that pixel locations responding to scene changes are sufficient for sub-millisecond place recognition with event cameras.
3) We reduce the storage and search times of the large databases created by sub-millisecond accumulation windows through subsampling and analyze the resulting accuracy–compute trade-offs.
4) We benchmark Flash against existing event camera-based VPR methods and conduct ablation studies to assess the sufficiency of active-pixel location encoding, the effectiveness of normalization in mitigating aliasing bias, and the distribution of time intervals during which the robot remains not localized.

Flash's sub-millisecond recognition capability enables new possibilities for time-critical localization, where traditional VPR methods fail to meet latency requirements. The code will be released publicly to support reproducibility.

To foster future research, we make the code and dataset available: https://github.com/Vignesh-Ramanathan/flash

II. Related Works

This section reviews literature relevant to low-latency visual place recognition (VPR). We first discuss key works in traditional camera-based VPR that introduce concepts related to our approach, particularly visual saliency and efficient computing (Section II-A). We then review existing event camera-based VPR methods, showing how none have exploited the microsecond temporal resolution despite this being a key advantage of the sensor (Section II-B). Finally, we examine recent successes in sub-millisecond perception using event cameras in other vision tasks, demonstrating that ultra-low-latency processing is achievable but has not yet been realized for VPR (Section II-C).

A. Traditional Camera-based VPR

1) Visual Saliency for VPR: Visual saliency, the perceptual quality that distinguishes certain scene regions, has been widely used in VPR to identify cues relevant for place recognition. Zaffar et al. [10] used entropy maps to select salient regions for Histogram-of-Gradients-based matching, while Wang et al. [20] performed saliency detection in the frequency domain for efficiency. Keetha et al. [21] reduced perceptual aliasing by using VLAD clusters to distinguish environment-specific from place-specific cues. Peng et al. [22] proposed a multi-level attention framework combining semantic guidance with attentional pyramid pooling, and Nie et al. [23] integrated saliency cues with semantic embeddings through pooling operations.

While effective, these methods require explicit saliency computation. In contrast, the ego-motion of an event camera naturally captures brightness changes, and the resulting event locations serve as strong discriminative features for place recognition.

2) Efficient VPR: Recent works have pursued computational efficiency to minimize VPR processing latency. Ferrarini et al. [24] binarized CNNs to a single-bit precision. They later addressed the full-precision input bottleneck with depthwise separable convolutions [8]. Grainge et al. [25] analyzed how network design choices impact recall-latency trade-offs.

Bio-inspired approaches have also shown promise: Arcanjo et al. [26], [27] developed DorsoNet based on fruit fly circuits with voting mechanisms, while Hussaini et al. [28] and Hines et al. [9] employed SNNs with spike-timing-dependent plasticity and time-to-first-spike encoding to enable efficient place recognition.

Despite these advances, traditional camera-based methods remain fundamentally limited by frame capture rates. Event cameras overcome this limitation through microsecond-precision asynchronous sensing, enabling the ultra-low-latency recognition demonstrated in this work.

B. Event Camera-Based VPR

Existing event camera-based place recognition methods have adapted events for use with NetVLAD [29] through various dense representations. Fischer et al. [14] binned events with multiple window sizes before reconstructing frames, Lee et al. [15] generated illumination-invariant edge images, and Lee et al. [16] used SNNs for edge reconstruction. Kong et al. [17] fused Event Spike Tensors [30] with NetVLAD. Although effective, these approaches process events as dense representations over tens- to hundreds-of-milliseconds windows, sacrificing the sparsity and microsecond temporal resolution inherent to event cameras. Processing the dense representations using NetVLAD introduces significant processing latencies, further bottlenecking the microsecond resolution of event cameras.

Other works have explicitly incorporated the unique sensing characteristics of event cameras into their designs. Fischer et al. [31] exploited the sparse event distribution by sampling pixel coordinates with high variability, improving computational efficiency. Hines et al. [19] employed an SNN running on a SPECK^{TM} neuromorphic processor and demonstrates that combining neuromorphic sensing, algorithms, and hardware enables real-time, energy-efficient localization on a small, resource-constrained hexapod robot.

To date, no prior work has fully exploited the high temporal resolution of event cameras. Their change-driven, asynchronous events occur with microsecond precision, enabling the low-latency, bandwidth-efficient place recognition needed for high-speed robotics.

C. Low-latency applications of event camera

Prior works have leveraged the event camera's microsecond temporal resolution and spatiotemporally sparse output for low-latency perception. Gehrig et al. [32] developed a recurrent vision architecture for object detection, achieving detection latencies below 10 ms on a commercial GPU. In a subsequent study, Gehrig et al. [6] fused 20 FPS traditional camera images with event camera data using a Graph Neural Network to achieve low latency without compromising detection accuracy in a bandwidth-efficient manner.

Many works have used hardware accelerators and neuromorphic chips to process event camera data to achieve sub-millisecond latency perception. Chiavazza et al. [33] implemented a depth-from-motion algorithm on Intel's Loihi 2 neuromorphic chip, estimating optical flow directly from events via time-to-travel measurements and combining it with camera velocity to infer depth, achieving latencies below 0.5 ms. Dampfhoffer et al. [13] proposed a hybrid optical flow architecture with an asynchronous event branch for microsecond-scale predictions and a periodic branch for large-scale temporal context, breaking the accuracy-latency trade-off and achieving latencies of tens of microseconds on specialized hardware.

While other vision tasks have achieved sub-millisecond latencies with event cameras, VPR systems have yet to explore this capability. Flash fills this gap as the first event camera-based VPR system capable of recognizing places using less than a millisecond of event data, matching temporal capabilities demonstrated in other perception tasks.

III. Preliminaries

This section establishes the foundational concepts underlying our proposed method, Flash, providing the necessary background for understanding how it exploits event camera characteristics to achieve ultra-low-latency place recognition. We first formalize visual place recognition and its standard feature similarity matching formulation (Section III-A). We then review event camera principles, focusing on their asynchronous and change-driven sensing model (Section III-B).

A. Visual Place Recognition (VPR)

In Visual Place Recognition (VPR), the goal is to identify, for a given query frame I_Q, the reference frame $I_R \in \mathcal{D}$ that corresponds to the same physical location [1]. Each frame is represented by a feature vector $\mathbf{f} \in \mathbb{R}^d$, extracted using a learned or handcrafted method.

The best-matching reference frame I_R^* is determined by maximizing a similarity function $S(\cdot,\cdot)$:

$$I_R^* = \arg\max_{I_R \in \mathcal{D}} S(\mathbf{f}_{I_Q}, \mathbf{f}_{I_R}). \tag{1}$$

Here, $\mathbf{f}_{I_Q}$ and $\mathbf{f}_{I_R}$ denote the feature vectors of the query and a reference frame, respectively. The function $S(\cdot,\cdot)$ typically represents a similarity measure such as cosine similarity.

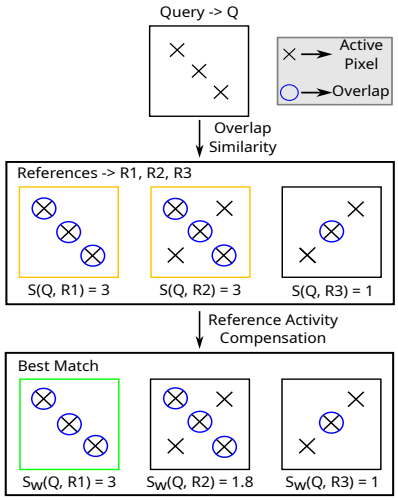

Fig. 2. **Overview of Flash:** We encode the active pixel locations as binary frames, and measure the query–reference similarity as their overlap using a bitwise operators. To prevent bias toward reference frames with high event activity, Reference Activity Compensation normalizes these similarity scores, filtering out incorrect matches.

B. Event Cameras

Event cameras asynchronously detect per-pixel brightness changes, resulting in bandwidth-efficient, high-temporal resolution perception of the environment [34], [35]. Each detected change produces an event $e_k = (x_k, y_k, t_k, p_k)$, where (x_k, y_k) are the spatial pixel coordinates, t_k is the timestamp of the event, and $p_k \in \{+1, -1\}$ indicates the polarity of the intensity change (increase or decrease).

Given an event stream $\mathcal{E} = \{e_k\}_{k=1}^N$, we partition it into disjoint temporal windows $\mathcal{W}_i$ of duration δ:

$$\mathcal{W}_i = \{e_k \in \mathcal{E} \mid t_i < t_k \leq t_i + \delta\}. \tag{2}$$

From each window $\mathcal{W}_i$, an event frame $\mathbf{I}_i \in \mathbb{R}^{H \times W}$ can be constructed, where (H, W) are the frame height and width. This binning process aggregates events into a frame-like representation (see Eq. (3)).

IV. Methodology: Flash

We present Flash, a binary-frame-based visual place recognition system that achieves sub-millisecond latency by exploiting the sparsity of event data. Unlike existing methods that accumulate events over long time windows to form dense representations, Flash operates directly on the binary pattern of active pixels within very small window sizes.

Flash, shown in Fig. 2, consists of three key components: (1) constructing binary frames that encode only pixel activity presence (Section IV-A); (2) computing query-reference similarity through efficient bitwise operations on active pixels of query (Section IV-B); and (3) applying Reference Activity Compensation (RAC) to prevent bias toward high-activity frames (Section IV-C). This design enables Flash to perform reliable place recognition efficiently, using less than 1 ms of event data per query, with meaningful place recognition possible using as low as $\sim 15\mu s$ of event data.

A. Binary Frame Construction

Event streams are partitioned into disjoint temporal windows $\mathcal{W}_i$ of duration δ (see Eq. (2)). From each window, we construct a binary event frame $B_i \in \{0,1\}^{H\times W}$:

$$B_i(x,y) = \begin{cases} 1, & \text{if } \exists e_k \in \mathcal{W}_i \text{ such that } (x_k, y_k) = (x,y) \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

This representation emphasizes the spatial distribution of the event activity while discarding intensity and frequency information, significantly reducing computational burden.

B. Overlap Similarity

Let $\mathcal{B}_\mathcal{Q}$ and $\mathcal{B}_\mathcal{R}$ denote the binary query and reference frames, respectively. The set of active pixels in the query is defined as

$$\Omega_{\mathcal{B}_\mathcal{Q}} = \{(x,y) \mid \mathcal{B}_\mathcal{Q}(x,y) = 1\}. \tag{4}$$

The similarity between $\mathcal{B}_\mathcal{Q}$ and $\mathcal{B}_\mathcal{R}$ is defined as the number of overlapping active pixels, restricted to the query's active set:

$$S(\mathcal{B}_\mathcal{Q}, \mathcal{B}_\mathcal{R}) = \sum_{(x,y)\in\Omega_{\mathcal{B}_\mathcal{Q}}} \big(\mathcal{B}_\mathcal{Q}(x,y) \,\&\, \mathcal{B}_\mathcal{R}(x,y)\big), \tag{5}$$

where $\&$ denotes the bitwise AND operator. This formulation ensures that only query-relevant locations contribute to the similarity score, with computational complexity $O(|\Omega_{\mathcal{B}_\mathcal{Q}}|)$ rather than $O(H \cdot W)$, providing significant speedup when $|\Omega_{\mathcal{B}_\mathcal{Q}}| \ll H \cdot W$. For the small time windows considered in Flash, $|\Omega_{\mathcal{B}_\mathcal{Q}}|$ is typically in the tens, whereas $H \cdot W$ ranges from one to tens of thousands (see Section VI-D).

C. Reference Activity Compensation (RAC)

Direct overlap similarity (Eq. (5)) favors reference frames with high activity, which naturally produce more overlapping pixels regardless of actual scene correspondence (see Fig. 2). To address this bias, we introduce Reference Activity Compensation (RAC), a lightweight normalization scheme.

$$\tilde{w}(\mathcal{B}_\mathcal{Q}, \mathcal{B}_\mathcal{R}) = \begin{cases} \frac{|\Omega_{\mathcal{B}_\mathcal{Q}}|}{|\Omega_{\mathcal{B}_\mathcal{R}}|}, & \text{if } |\Omega_{\mathcal{B}_\mathcal{R}}| > |\Omega_{\mathcal{B}_\mathcal{Q}}| \\ 1, & \text{otherwise,} \end{cases} \tag{6}$$

Here, $|\Omega_{\mathcal{B}_\mathcal{R}}|$ denotes the number of active pixels in reference frame $\mathcal{R}$. Reference frames with large $|\Omega_{\mathcal{B}_\mathcal{R}}|$ are more likely to alias with a query frame $\mathcal{B}_\mathcal{Q}$, and the above normalization mitigates this effect when the query frame does not exhibit comparable event activity. When $|\Omega_{\mathcal{B}_\mathcal{R}}| \le |\Omega_{\mathcal{B}_\mathcal{Q}}|$, no adjustment is applied, as RAC targets aliasing from overly active reference frames; applying the ratio in this case could bias the query toward sparser frames.

The normalized similarity is then computed as:

$$S_w(\mathcal{B}_\mathcal{Q}, \mathcal{B}_\mathcal{R}) = \tilde{w}(\mathcal{B}_\mathcal{Q}, \mathcal{B}_\mathcal{R}) \cdot S(\mathcal{B}_\mathcal{Q}, \mathcal{B}_\mathcal{R}), \tag{7}$$

where $S(\mathcal{B}_\mathcal{Q}, \mathcal{B}_\mathcal{R})$ is the raw overlap similarity from Eq. (5). This adjustment penalizes overly active reference frames, ensuring a balanced comparison with only two floating-point operations per query-reference pair.

Finally, the best-matching reference frame is selected by maximizing the weighted similarity (following Eq. (1)):

$$B_R^* = \underset{\mathcal{B}_\mathcal{R}\in\mathcal{D}}{\operatorname{argmax}}\ S_w(\mathcal{B}_\mathcal{Q}, \mathcal{B}_\mathcal{R}). \tag{8}$$

D. System Efficiency

By restricting computations to active query pixels and using lightweight operations, Flash delivers a fast and efficient solution for sub-millisecond event-based VPR. The binary-frame design ensures scalability, bitwise operations for similarity guarantee low processing times, and overlap normalization improves robustness under varying levels of scene activity.

Flash's design enables highly efficient implementation through several key optimizations. The sparse binary arrays of the reference database can be stored efficiently by keeping only the coordinates of the active pixels, which are few for small window sizes. The similarity computation can leverage hardware-accelerated bitwise AND operations and population count (popcount) instructions available on modern CPUs for fast and cheap computations. RAC normalization requires only two floating-point operations: one for the normalization factor and one for multiplying the overlap similarity. Together, these implementation strategies can massively reduce the processing latencies over existing approaches and allow the high sensing capabilities of event cameras to be fully utilized.

V. Experimental setup

This section describes the datasets used to evaluate our approach (Section V-A) and defines the performance metrics (Section V-B). including a new metric, Time to Correct Match (TCM), which captures the temporal dynamics of place recognition. We benchmark our proposed approach against existing methods and introduce a strong baseline, termed *Zoom*, to assess both performance and efficiency (Section V-C).

A. Datasets

We evaluate on two event-based VPR datasets recorded using the DAVIS346 camera. The Brisbane-Event-VPR dataset (henceforth *Brisbane*) [14] spans an 8 km urban route traversed six times at $\sim$15 m/s, with natural stops and varying illumination and weather (excluding night-time). The QCR-Event-VPR dataset (henceforth *QCR*) [31] covers a 160 m indoor route traversed 16 times at $\sim$1 m/s with different speeds and camera orientations. Together, Brisbane and QCR enable evaluation across large-scale outdoor driving and structured indoor robotics.

B. Evaluation Metrics

1) Recall@1: To evaluate matching performance, we use Recall@1, which measures the fraction of query frames for which the top-ranked reference frame is a correct match. Formally, Recall@1 is defined as

$$\text{Recall@1} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{1}\big(I_R^* \in \mathcal{M}(I_Q^n)\big), \tag{9}$$

where N is the number of query frames, I_R^* denotes the top-ranked reference frame for query I_Q^n, $\mathcal{M}(I_Q^n)$ is the set of ground-truth matches for query I_Q^n, and $\mathbf{1}(\cdot)$ is the indicator function.

2) Time between Correct Match (TCM): We introduce TCM as a metric to evaluate the temporal performance of a VPR system. In environments where the robot encounters no unseen places, TCM quantifies the duration between consecutive correct place recognitions during a traverse.

For a sequence of queries $\{I_Q^1, \ldots, I_Q^N\}$, let

$$\delta_n = \begin{cases} 1, & \text{if the system correctly matches } I_Q^n \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

and define the time to the i-th correct match as

$$\text{TCM}_i = t_i - t_{i-1}, \quad t_i = \min\{n > t_{i-1} \mid \delta_n = 1\}, t_0 = 0. \tag{11}$$

Using the set of TCM values from a traverse, we can construct a TCM distribution that measures the number of correct matches for the given TCM time over the complete traverse:

$$P(\text{TCM} = \tau) = \frac{\#\text{ of correct matches with TCM } \tau}{\text{total number of places to match}}. \tag{12}$$

This distribution can be converted to a Cumulative Distribution Function (CDF) and it provides a measure of the time durations during which the robot is unable to obtain a correct position estimate. The higher the CDF value for a given TCM, the shorter the period of position "blindness". Unlike Recall@1, which measures average performance over the entire traverse, the CDF provides a more granular measure of VPR performance. This new measure is a more useful indicator of utility in latency-sensitive scenarios when the correct position is required within a specific time window, like fast moving robotic platforms, vehicles and drones.

C. Baselines

We benchmark our method against existing event camera-based VPR approaches, which are not explicitly designed for sub-millisecond timescales. We also introduce a strong new baseline, Zoom, which leverages event count information along with active pixel locations for sub-millisecond VPR.

1) Existing Methods: In the following, we outline the set of baselines used for comparison before introducing our proposed variant. We first consider the sum of absolute differences (SAD) computed over all pixels in the frame, a widely used approach in prior works [18], [36]. Since we base our approach on sampling query pixels, we further evaluate two pixel sampling strategies: first, we sample a fixed set of random pixels across query-reference pairs. Distances are computed using SAD (Rand-Pix+SAD), following the baseline approach in [31]. In the second, we use the variance-based pixel selection strategy proposed in [31], which selects pixels with high variance across the reference set and computes SAD over these locations (Sparse-Event-VPR). This approach, closely related to our proposed method, focuses on the most distinctive pixels while discarding less informative ones. Sparse-Event-VPR outperforms certain learning-based event VPR methods, including Event-VPR [17] and EventVLAD [15]. We sample 150 pixels for Rand-Pix+SAD and Sparse-Event-VPR as done in [31].

We note that reconstruction-based pipelines such as E2VID [37], a key component of Ensemble-Event-VPR [14], cannot generate stable outputs at the fine temporal resolutions considered here and are therefore excluded from our quantitative evaluation.

2) Zoom: As a baseline closely aligned with the proposed method, we use event count frames instead of binary frames and compute a masked sum of absolute differences (SAD) over the query's active pixels:

$$D(I_Q, I_R) = \sum_{(x,y)\in\Omega_{I_Q}} |I_Q(x,y) - I_R(x,y)|. \tag{13}$$

We normalize the distances based on the relative activity of the query and reference frames:

$$w(I_Q, I_R) = \begin{cases} \dfrac{|\Omega_{I_R}|}{|\Omega_{I_Q}|} & |\Omega_{I_R}| \geq |\Omega_{I_Q}| \\ 1 & \text{otherwise} \end{cases}. \tag{14}$$

Using full event counts, Zoom provides a richer but less efficient representation, highlighting the Flash's speed and computational efficiency, while serving as a baseline.

VI. Results

A. Comparison to State-of-the-Art

We first evaluate the capabilities of existing event camera-based VPR approaches at sub-millisecond timescales, and then assess the performance of the proposed method against the baselines described in Section V-C.

a) Recall@1: Fig. 3 presents the Recall@1 performance of Flash and Zoom compared to existing approaches across varying window sizes. Flash achieves the highest performance, improving average Recall@1 over all shown window sizes by $5.42\times$ i.e. $\sim 542\%$ (QCR) and $2.67\times$ (Brisbane) over the best baseline (Sparse-Event-VPR), with even larger gains of $11.33\times$ (QCR) and $5.92\times$ (Brisbane) at sub-millisecond timescales. Zoom also delivers consistent improvements, with overall gains of $4.69\times$ (QCR) and $2.69\times$ (Brisbane). At sub-millisecond scales, Zoom achieves $10.02\times$ (QCR) and $5.89\times$ (Brisbane) gains. Between the two proposed methods, Flash outperforms Zoom on QCR by 0.13× overall and 0.12× at sub-millisecond scales. In contrast, Zoom overall slightly outperforms Flash on Brisbane ($0.01\times$) due to its stronger performance on the sunset1–sunset2 pair with no significant difference in performance at sub-millisecond timescales. These results indicate that the locations of active pixels provide more discriminative information than other pixel sampling strategies. Interestingly, Zoom's performance drops at 5 ms on the Brisbane dataset due to RAC's inability to fully mitigate aliasing from high-activity reference frames.

We also evaluate the importance of RAC in our proposed methods. On QCR, Flash and Zoom outperform their variants without RAC by $1.39\times$ and $0.11\times$, respectively. On Brisbane

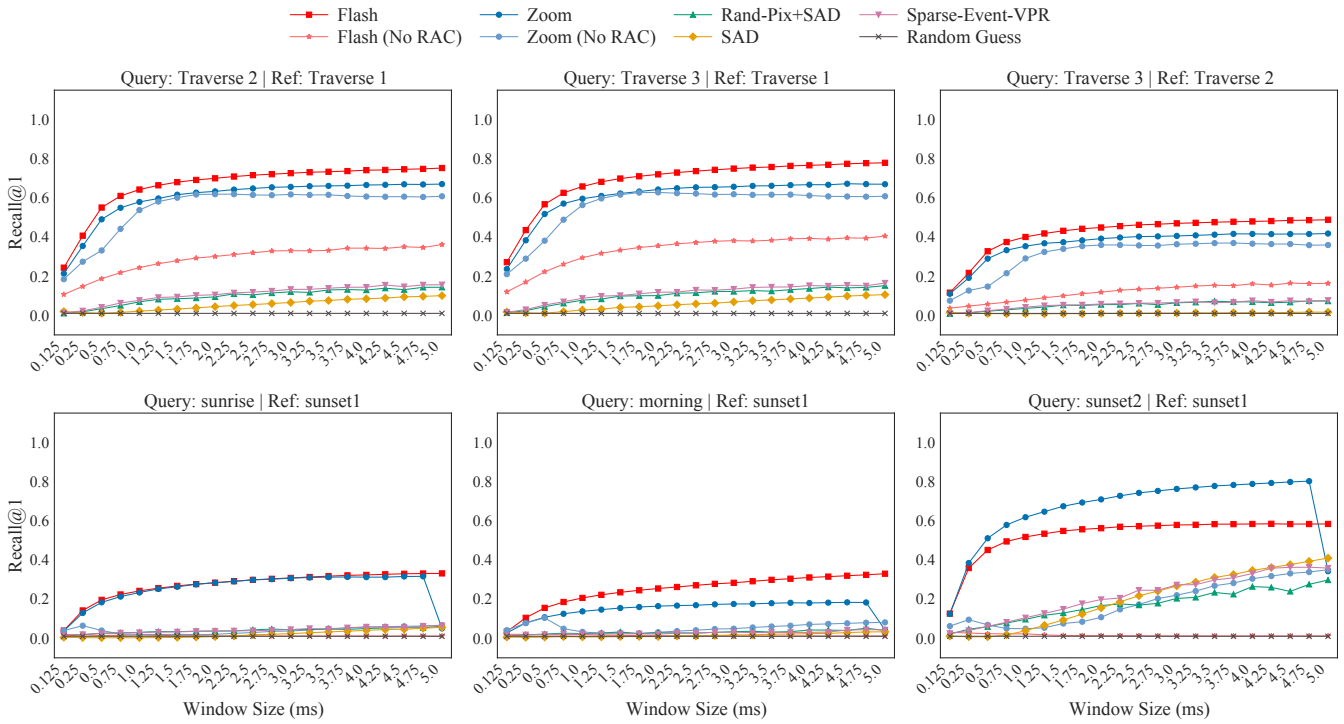

Fig. 3. Recall@1 across different event accumulation window sizes for the QCR-Event-VPR dataset (top row) and the Brisbane-Event-VPR dataset (bottom row). Flash and Zoom consistently achieve the highest recall. At sub-millisecond timescales, Flash improves matching performance by 11.33× (QCR) and 5.92× (Brisbane), while Zoom outperforms the baseline by 10.02× (QCR) and 5.89× (Brisbane).

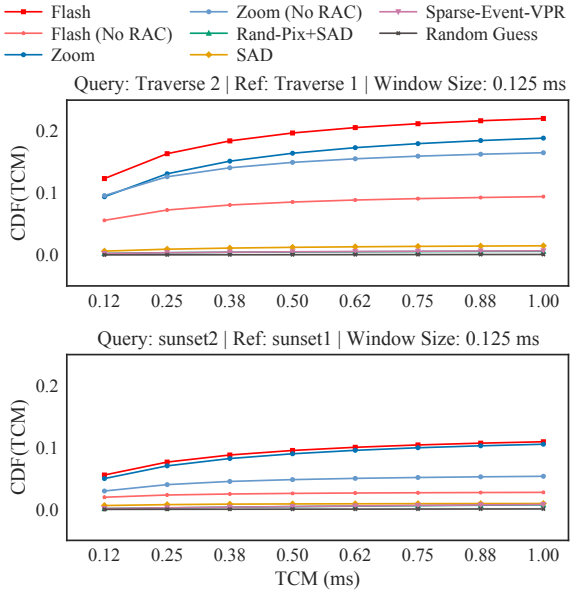

Fig. 4. VPR performance measured using the Time between Correct Match (TCM) metric, where higher CDF values indicate more frequent correct recognitions. Flash and Zoom achieve remarkably higher correct matches within short intervals, with Flash exceeding the best baseline by up to 13.91× on QCR and 10.46× on Brisbane for a 0.125 ms event window.

which includes outdoor lighting challenges such as sun glare, the improvements of using RAC are larger: 23.87× for Flash and 2.82× for Zoom. At sub-millisecond scales, Flash improves by 1.89× (QCR) and 9.25× (Brisbane), while Zoom improves by 0.34× (QCR) and 2.52× (Brisbane). These results show that normalizing similarity scores based on relative query–reference activity is both beneficial and crucial under challenging illumination conditions.

b) Time to Correct Match (TCM): Given our focus on high-speed localization, we evaluate position blind time using the TCM metric. Fig. 4 shows that Flash and Zoom achieve correct position estimates far more frequently than the baselines. For VPR predictions with a 0.125 ms event window, Flash predicts the correct position on or before 1 ms 13.91× more frequently on QCR and 10.46× more frequently on Brisbane compared to the best baseline (SAD). Zoom also achieves substantial gains, outperforming the baseline by 11.77× on QCR and 10.04× on Brisbane. These results demonstrate that our proposed methods substantially reduce the time instant for which the robot operates without awareness of its position in the map.

B. Temporal Lower Bound for Event Camera-based VPR

To evaluate the minimal temporal requirements for event-based VPR, we plot Recall@1 as a function of accumulation window sizes on the order of tens of μs (Fig. 5). Even at the smallest window of $\sim$15 μs, our system achieves performance above random chance, demonstrating that ultra-low-latency VPR is feasible, albeit with reduced Recall@1. At such timescales, only a few to tens of events are available for matching (see Section VI-D), and the robot has moved just a fraction of a millimeter in both QCR and Brisbane. However, due to the sensing dynamics of event cameras, we note that some scene information from earlier motion appears within this interval. These results establish a practical lower bound on the temporal resolution required for event-based VPR, highlighting the ability of our method to extract

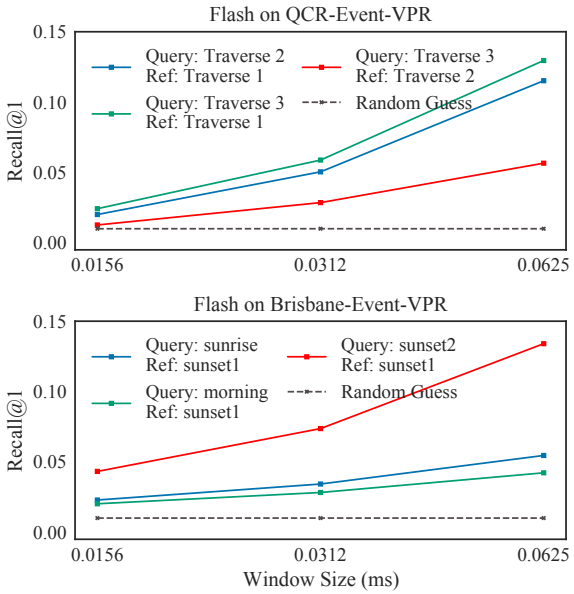

Fig. 5. Recall@1 performance of Flash across ultra-short temporal accumulation windows. Even at ∼15 μs, corresponding to only a few tens of events and sub-millimetre robot motion, Flash achieves accuracy above random chance, demonstrating the feasibility of ultra-low latency event-based VPR and establishing a practical lower bound on the temporal resolution required.

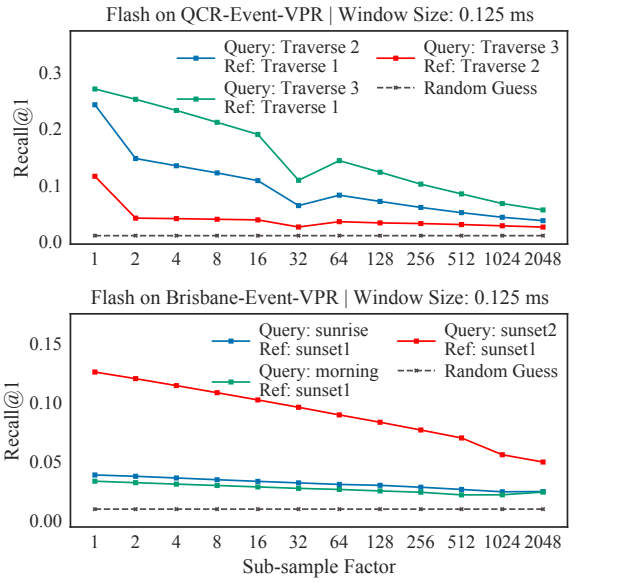

Fig. 6. Effect of sub-sampling the reference database on Recall@1 for QCR and Brisbane. Progressive halving of the reference frames maintains Recall@1 with minimal loss while reducing storage and search time, with the first halving reducing Recall@1 by ∼36.85% on QCR and ∼3.67% on Brisbane. Aggressive reductions degrade performance, with total drops of ∼61.40% and ∼23.56% for a 2048× reduction.

discriminative spatial features from extremely short temporal slices.

C. Sub-sampling the Reference Database

As our system operates at sub-millisecond latency, smaller temporal windows result in large reference databases. For 0.125 ms window size, this corresponds to ∼5.8M frames for Brisbane and ∼1.3M frames for QCR datasets. This leads to increased storage demands and longer search times, which are mitigated to a certain extent by our sparse and binary feature descriptors. To further reduce the compute, we progressively half the database size by retaining every second reference frame to tradeoff accuracy for lower latency. As shown in Fig. 6, the first halving reduces Recall@1 by ∼36.8% on QCR and only ∼3.6% on Brisbane, while cutting storage and search time by 50%. Across successive halvings, the average Recall@1 drop per halving is ∼11.6% for QCR and ∼4.8% for Brisbane, resulting in total reductions of ∼61.4% and ∼23.5%, respectively, at a 2048× database reduction. These results highlight that moderate sub-sampling provides substantial efficiency gains with minimal accuracy loss.

D. Event Data Statistics

The datasets used in our experiments were collected using the DAVIS346 event camera, which has a sensor resolution of 346×260 pixels. We downsample the event frames to 86×45 before using them for VPR. Table I reports the average number of events and active pixels for different temporal accumulation windows in the QCR and Brisbane datasets. At sub-millisecond windows, only a small fraction of pixels are active. At 1 ms, ∼173 pixels (3.1%) for QCR and ∼492 pixels (8.8%) for Brisbane are active on average per frame, whereas at the smallest window of ∼ 15μs, only ∼3 pixels (0.05%) for QCR and ∼12 pixels (0.2%) for Brisbane are active on average per frame, and these are sufficient for Flash to achieve above-chance performance. As the window size decreases, the number of events per active pixel tends toward 1. This supports our choice of binary frames for sub-millisecond VPR.

TABLE I

AVERAGE EVENTS AND ACTIVE PIXELS VS. ACCUMULATION WINDOW SIZE FOR QCR AND BRISBANE, SHOWING THAT SUB-MS VPR IS POSSIBLE FROM VERY SPARSE DATA.

Window Size (ms)	QCR-Event-VPR		Brisbane-Event-VPR	
	#Events	#Active Pixels	#Events	#Active Pixels
0.0156	3.50	3.48	12.65	12.44
0.0312	7.08	7.01	25.31	24.65
0.0625	14.16	13.87	50.62	48.45
0.1250	28.32	27.18	100.66	89.01
0.2500	56.64	52.34	202.48	163.38
0.5000	113.29	97.74	404.95	291.48
0.7500	169.93	137.80	607.43	399.66
1.0000	226.58	173.58	809.90	493.01

VII. DISCUSSION AND CONCLUSIONS

We presented *Flash*, a novel event-based visual place recognition method that departs from dense event representations and instead operates directly on active pixel locations for efficiency. By processing sub-millisecond slices of event data with lightweight operations, Flash enables ultra-low-latency localization. Our experiments show that Flash outperforms baselines by a large margin, demonstrating that accurate place recognition can be achieved directly from event locations at unprecedented latencies, opening new directions for time-critical robotic localization.

Despite these promising results, our evaluation remains

limited to datasets that do not fully capture the challenges of high-speed robotics. In particular, the performance of Flash under extreme motion dynamics and very high event rates is yet to be explored.

Future work includes deploying Flash on dedicated hardware to fully exploit its computational efficiency, integrating it into event-based visual-inertial odometry pipelines for high-speed SLAM, and evaluating it on high-speed datasets such as drone racing or agile manipulation. Such extensions would further establish Flash as a practical component of real-world event-based perception systems.

REFERENCES

[1] S. Schubert, P. Neubert, *et al.*, "Visual place recognition: A tutorial," *IEEE Robotics & Automation Magazine*, vol. 31, no. 3, pp. 139–153, 2024.

[2] S. Garg, T. Fischer, and M. Milford, "Where is your place, visual place recognition?" in *International Joint Conference on Artificial Intelligence*, 2021, pp. 4416–4425.

[3] P. Yin, J. Jiao, *et al.*, "General place recognition survey: Toward real-world autonomy," *IEEE Transactions on Robotics*, vol. 41, pp. 3019–3038, 2025.

[4] X. Zhang, L. Wang, and Y. Su, "Visual place recognition: A survey from deep learning perspective," *Pattern Recognition*, vol. 113, p. 107760, 2021.

[5] I. Moskalenko, A. Kornilova, and G. Ferrer, "Visual place recognition for aerial imagery: A survey," *Robotics and Autonomous Systems*, vol. 183, p. 104837, 2025.

[6] D. Gehrig and D. Scaramuzza, "Low-latency automotive vision with event cameras," *Nature*, vol. 629, no. 8014, pp. 1034–1040, 2024.

[7] F. Vanegas, K. J. Gaston, J. Roberts, and F. Gonzalez, "A framework for UAV navigation and exploration in gps-denied environments," in *IEEE Aerospace Conference*, 2019, pp. 1–6.

[8] B. Ferrarini, M. Milford, K. D. McDonald-Maier, and S. Ehsan, "Highly-efficient binary neural networks for visual place recognition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 5493–5500.

[9] A. D. Hines, P. G. Stratton, M. Milford, and T. Fischer, "VPRTempo: a fast temporally encoded spiking neural network for visual place recognition," in *IEEE International Conference on Robotics and Automation*, 2024, pp. 10 200–10 207.

[10] M. Zaffar, S. Ehsan, M. Milford, and K. McDonald-Maier, "Cohog: A light-weight, compute-efficient, and training-free visual place recognition technique for changing environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1835–1842, 2020.

[11] G. Gallego, T. Delbrück, *et al.*, "Event-based vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 154–180, 2022.

[12] B. Zhang, Y. Gao, J. Li, and H. K.-H. So, "Co-designing a submillisecond latency event-based eye tracking system with submanifold sparse cnn," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2024, pp. 5771–5779.

[13] M. Dampfhoffer, T. Mesquida, *et al.*, "Graph neural network combining event stream and periodic aggregation for low-latency event-based vision," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2025, pp. 6909–6918.

[14] T. Fischer and M. Milford, "Event-based visual place recognition with ensembles of temporal windows," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6924–6931, 2020.

[15] A. J. Lee and A. Kim, "Eventvlad: Visual place recognition with reconstructed edges from event cameras," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021, pp. 2247–2252.

[16] H. Lee and H. Hwang, "Ev-reconnet: Visual place recognition using event camera with spiking neural networks," *IEEE Sensors Journal*, vol. 23, no. 17, pp. 20 390–20 399, 2023.

[17] D. Kong, Z. Fang, *et al.*, "Event-vpr: End-to-end weakly supervised deep network architecture for visual place recognition using event-based vision sensor," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–18, 2022.

[18] G. B. Nair, M. Milford, and T. Fischer, "Enhancing visual place recognition via fast and slow adaptive biasing in event cameras," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024, pp. 3356–3363.

[19] A. D. Hines, M. Milford, and T. Fischer, "A compact neuromorphic system for ultra–energy-efficient, on-device robot localization," *Science Robotics*, vol. 10, no. 103, p. eads3968, 2025.

[20] H. Wang, C. Wang, and L. Xie, "Online visual place recognition via saliency re-identification," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 5030–5036.

[21] N. V. Keetha, M. Milford, and S. Garg, "A hierarchical dual model of environment- and place-specific utility for visual place recognition," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6969–6976, 2021.

[22] G. Peng, J. Zhang, H. Li, and D. Wang, "Attentional pyramid pooling of salient visual residuals for place recognition," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 885–894.

[23] J. Nie, D. Xue, *et al.*, "Efficient saliency encoding for visual place recognition: Introducing the lightweight pooling-centric saliency-aware vpr method," *IEEE Robotics and Automation Letters*, vol. 9, no. 7, pp. 6035–6042, 2024.

[24] B. Ferrarini, M. J. Milford, K. D. McDonald-Maier, and S. Ehsan, "Binary neural networks for memory-efficient and effective visual place recognition in changing environments," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2617–2631, 2022.

[25] O. Grainge, M. Milford, *et al.*, "Design space exploration of low-bit quantized neural networks for visual place recognition," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5070–5077, 2024.

[26] B. Arcanjo, B. Ferrarini, *et al.*, "An efficient and scalable collection of fly-inspired voting units for visual place recognition in changing environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2527–2534, 2022.

[27] B. Arcanjo, B. Ferrarini, *et al.*, "Aggregating multiple bio-inspired image region classifiers for effective and lightweight visual place recognition," *IEEE Robotics and Automation Letters*, vol. 9, no. 4, pp. 3315–3322, 2024.

[28] S. Hussaini, M. Milford, and T. Fischer, "Spiking neural networks for visual place recognition via weighted neuronal assignments," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4094–4101, 2022.

[29] R. Arandjelović, P. Gronat, *et al.*, "NetVLAD: CNN architecture for weakly supervised place recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437–1451, 2018.

[30] D. Gehrig, A. Loquercio, K. G. Derpanis, and D. Scaramuzza, "End-to-end learning of representations for asynchronous event-based data," in *IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5633–5643.

[31] T. Fischer and M. Milford, "How many events do you need? event-based visual place recognition using sparse but varying pixels," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12 275–12 282, 2022.

[32] M. Gehrig and D. Scaramuzza, "Recurrent vision transformers for object detection with event cameras," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 884–13 893.

[33] S. Chiavazza, S. M. Meyer, and Y. Sandamirskaya, "Low-latency monocular depth estimation using event timing on neuromorphic hardware," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2023, pp. 4071–4080.

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

[35] C. Brandli, R. Berner, *et al.*, "A 240 × 180 130 db 3 µs latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.

[36] M. Milford, H. Kim, S. Leutenegger, and A. Davison, "Towards visual slam with event-based cameras," in *The problem of mobile sensors workshop in conjunction with RSS*, 2015.

[37] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "High speed and high dynamic range video with an event camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.