

How Many Events Do You Need? Event-based Visual Place Recognition Using Sparse But Varying Pixels

Tobias Fischer and Michael Milford

Abstract—Event cameras continue to attract interest due to desirable characteristics such as high dynamic range, low latency, virtually no motion blur, and high energy efficiency. One of the potential applications that would benefit from these characteristics lies in visual place recognition for robot localization, i.e. matching a query observation to the corresponding reference place in the database. In this letter, we explore the *distinctiveness of event streams* from a small subset of pixels (in the tens or hundreds). We demonstrate that the absolute difference in the number of events at those pixel locations accumulated into event frames can be sufficient for the place recognition task, when pixels that display large variations in the reference set are used. Using such *sparse* (over image coordinates) but *varying* (variance over the number of events per pixel location) pixels enables frequent and computationally cheap updates of the location estimates. Furthermore, when event frames contain a constant number of events, our method takes full advantage of the event-driven nature of the sensory stream and displays promising robustness to changes in velocity. We evaluate our proposed approach on the Brisbane-Event-VPR dataset in an outdoor driving scenario, as well as the newly contributed indoor QCR-Event-VPR dataset that was captured with a DAVIS346 camera mounted on a mobile robotic platform. Our results show that our approach achieves competitive performance when compared to several baseline methods on those datasets, and is particularly well suited for compute- and energy-constrained platforms such as interplanetary rovers.

Index Terms—Localization, Data Sets for SLAM, Neuromorphic Sensing, Event-Based Vision

I. INTRODUCTION

VISUAL place recognition (VPR) is a crucial capability for autonomously navigating robots and vehicles [1]–[3]. In a nutshell, VPR is the task of retrieving the best matching reference observation out of a reference database, given a new observation at query time (Fig. 1). If the score associated to that match is sufficiently high, the match can then be used to e.g. propose a loop closure within a Simultaneous Localization

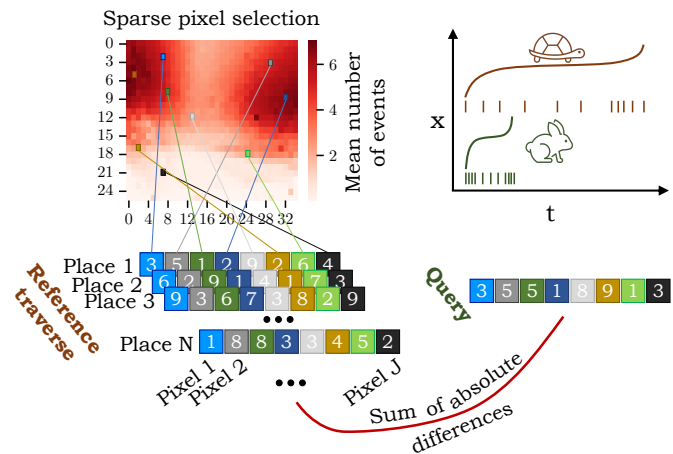


Fig. 1. **Event-based Place Recognition with Sparse Pixels.** Out of all available pixels, we select only a small subset (e.g. 0.028%) to describe a place – those that, on average, vary the most (top left). Each place descriptor contains the number of events that occurred at those sparse pixel locations (bottom left). Descriptor comparison is performed using the sum of absolute differences of these event counts (bottom right). If each feature descriptor contains a fixed number of events (summed over all sparse pixels), the feature descriptors demonstrate robustness to changes in velocity. This is due to the operating principle of event cameras, which captures the *changes* in the scene regardless of how quickly they occur – in an ideal scenario, the changes observed when going from place A to B are the same regardless of the velocity.

and Mapping (SLAM) system [4], or the VPR system can be used on its own to build a purely topological map [5].

Our recent research has demonstrated the use of event cameras in the VPR task [6]: their low power consumption and low bandwidth requirements make them well suited for on-board applications in robotic systems, and the high dynamic range and virtual absence of motion blur can be beneficial in low light conditions or high exposure to sunlight. At the same time, event cameras require new algorithms that can process the unconventional, and potentially very noisy, output data. They also have relatively low spatial resolution and are still comparatively expensive [7].

Each pixel of an event camera samples *independently* and is equi-spaced in *range* (i.e. an event is triggered when the change in the log intensity of a particular pixel exceeds a threshold), as opposed to conventional cameras where *all* pixels are sampled equi-spaced in *time* [7]. This property implies that the event count of an ideal event camera is proportional to the dynamic changes that occur in the scene – a static camera in a static scene would not output any events, while the total number of events is not impacted by how quickly the dynamic change occurs. Our first research question that we seek to answer is

Manuscript received: July 7, 2022; Revised September 20, 2022; Accepted October 12, 2022.

This paper was recommended for publication by Editor S. Behnke upon evaluation of the Associate Editor and Reviewers' comments. This work received funding from the Australian Government, via grant AUS-MURIB000001 associated with ONR MURI grant N00014-19-1-2571, Intel via grant RV3.248.Fischer, and by funding from ARC Laureate Fellowship FL210100156 to MM. The authors acknowledge continued support from the Queensland University of Technology (QUT) through the Centre for Robotics.

The authors are with the QUT Centre for Robotics, Queensland University of Technology, Brisbane, QLD 4000, Australia (e-mail: tobias.fischer@qut.edu.au, michael.milford@qut.edu.au).

Digital Object Identifier (DOI): see top of this page.

whether the dynamic change observed by traversing through a place remains relatively constant when the place is repeatedly traversed, potentially under changing appearance and velocities.

As single events carry minimal localization-relevant information in isolation, various feature extraction techniques to accumulate events across space and time have been proposed in the literature, so that these features can be consumed in downstream tasks [8]–[15]. An alternative approach is to use recurrent neural networks to reconstruct conventional images from the event stream, so that conventional computer vision techniques can be applied – often outperforming algorithms that operate “directly” on the event stream [16]–[18]. However, many of these feature extractors are workload or memory intensive, which offsets some of the computational and energy advantages of event cameras. Our second research question is thus whether we can describe a place in a storage and computationally friendly manner, while still achieving high performance and being robust to appearance variations and image misalignments.

Our third research question asks whether we can use just a few pixels of event camera footage to obtain such a high performing system, and whether we can find *discriminative* image regions (*e.g.* the side of the road), opposed to less informative regions like the dashboard of a car.

Our contributions can be summarized as follows:

- 1) We introduce a novel high-performing, computationally lightweight, training-free event-based pipeline for visual place recognition that is robust to moderate velocity changes (see Fig. 1). The pipeline operates on a small subset of the stream of events, specifically on pixels that exhibit a high variation in the event counts during mapping time, combined with “surround suppression” to avoid selection of neighboring pixels, which exhibit a high correlation and thus add little information.
- 2) We compare our method to several baselines [6], [19] and provide extensive ablation studies that evaluate the impact of the number of pixels that are selected, the location of those pixels in the image space, the time duration of the event frames, and the sequence length.
- 3) We collect the QCR-Event-VPR dataset, which contains 16 traverses of a 160m long route traveled by a Clearpath Jackal robot with a DAVIS346 camera mounted forward-, side-, or downward-facing.

To foster future research, we make the code and QCR-Event-VPR dataset available: <https://github.com/Tobias-Fischer/sparse-event-vpr>.

II. RELATED WORKS

We review works on representing events in Section II-A before providing an overview of related works on visual place recognition in Section II-B, and detailing prior works on place recognition using event cameras in Section II-C.

A. Event Representations

A range of feature representations have been proposed for event data, many of them focusing on the object classification

task. Event frames [8] are the simplest representation, where a conventional single-channel image is formed, with each pixel containing the number of events that occurred at that location over either a fixed time period, a fixed number of events across the image, or using adaptive times slices with variable duration [20]. If positive and negative polarities are considered separately, a two-channel image is formed [9].

Lagorce *et al.* [11] developed the Hierarchy Of Time Surfaces (HOTS) that describe the recent activity within a local neighboring window, whereby less recent events exponentially decay in value. Sironi *et al.* proposed Histograms of Averaged Time Surfaces (HATS) [12] which is less sensitive to noise and contrast when compared to HOTS. FEAST [13] is an extension of HOTS with adaptive thresholds, which allows for a simpler network architecture and integrated noise filtering.

Gehrig *et al.* [14] presented a unifying framework that learns the event representation end-to-end in a deep network. Most recently Baldwin *et al.* [15] introduced Time-Ordered Recent Event (TORE) volumes to compactly store raw spike timing information. Detailed comparisons of different event representations are provided in [15], [21]; and [22] compares event representations in the SLAM context.

To the best of our knowledge, we are not aware of any other works in the event camera literature that make use of just a tiny fraction of the available pixels. The closest related works in that regard are [23], [24] which focus on estimating the region that an object encompasses to recognize [23] and track [24] the object using attention mechanisms. In addition, a variety of works explore using as few events as possible to *e.g.* update event-based graph networks [25], [26] or spiking neural networks [27] *in the temporal domain*, but in contrast to our work, still use all available pixels *spatially*. Note that the sparse event representation in our work is task-specific and not likely to be applicable in other tasks without modifications.

B. Visual Place Recognition

Visual place recognition (VPR) is a thriving area, with applications including autonomous driving, robot navigation and augmented reality [1]–[3]. One particular application of VPR is in the proposal of loop closure candidates within Simultaneous Localization and Mapping (SLAM) [4], where VPR is used to recognize previously visited locations to build a globally accurate map. Challenges for VPR include changes in the appearance of the scene due to the time of day, season, and weather; structural changes due to occlusions and new constructions; and viewpoint changes [3]. In this article, we focus mainly on appearance changes, which are predominant in road-based scenarios where both translation and rotation changes are limited [1].

A reoccurring theme in VPR research is the use of sequences to improve performance. The underlying motivation is that perceptual aliasing – the phenomenon that two distinct places look nearly identical – is less likely to occur when considering multiple observations over time. In the SeqSLAM [28], a sequence score is obtained as the sum of the difference scores within a time window, and two sequences are matched if the lowest difference score is significantly lower than the second

lowest difference score (ratio test). A large body of work has since focused on sequence matching [19], [29]–[31], and most recently sequence-based methods have been adapted to better work with deep-learned features [32].

The most closely related work in the conventional VPR literature is [33] that investigates the amount of visual information that is required for place recognition in conventional cameras by varying the number of pixels, their pixel depth, and the sequence length. They were able to successfully localize in a path-like office environment with just two pixels measuring light intensity, however requiring a relatively large sequence length of 200 frames in that case. Our paper goes beyond [33] by using sparse pixels, rather than down-sampling image regions, and explores unique properties of event cameras, which are not covered in [33].

C. Event-based Visual Place Recognition

Before introducing our proposed approach, we will review the related works in the VPR domain that use event cameras. In the initial approach by Milford *et al.* [34], temporally binned windows were formed into event frames [8], which were then patch-normalized and compared using sequence matching. Rather than using the whole image, here we show that the informativeness of certain areas in the image is non-uniformly distributed.

In our prior work [6], we first reconstructed conventional frames from the event stream via [16], [17], extracted features from these frames using NetVLAD [35], and then found the best match by comparing the query and reference features using the cosine distance. The key novelty was in using an ensemble of features which emerged by using varying temporal windows to reconstruct the frames.

Kong *et al.* [36] took a similar approach named Event-VPR where the NetVLAD layer is trained directly on a voxel grid representation [14]. Another recent interesting approach is EventVLAD [37] which estimates edge images from the event stream and subsequently trains the NetVLAD layer using these edge images. Rather than going the roundabout way via reconstructing conventional frames or applying computationally expensive deep learning methods, in this research we directly use the event stream to recognize places.

Works in related areas that could inform event-based visual place recognition include [38], [39], which tackle event-based visual odometry, and [40], [41], which introduced event-based SLAM algorithms, however lack a loop closure component.

III. PROPOSED APPROACH

A. Preliminaries and Notation

We closely follow the notation that was used in our prior work [6]. The i -th event is denoted as $\mathbf{e}_i = (\mathbf{u}_i, t_i, p_i)$, where $\mathbf{u}_i = (u_i, v_i)^T$ refers to the pixel location where the event occurred, t_i is the timestamp associated to the event, and $p_i \in \{-1, 1\}$ is the event's polarity that indicates positive or negative brightness change. As reviewed in Section II-A, prior work [8] formed event frames \mathbf{I} by accumulating events over a specified time window, so that:

$$\mathbf{I}_{u,v}(t_k) := \{|\mathbf{e}_i| \mid u_i = u \wedge v_i = v \wedge t_i \in [t_k, t_k + \tau)\}, \quad (1)$$

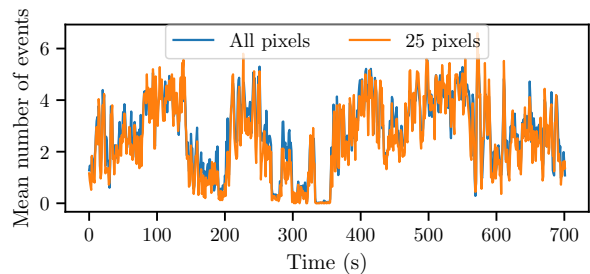


Fig. 2. **Event profile is maintained when drastically sub-sampling pixels.** This figure shows the number of events that occurred within a one second window, averaged either across all pixels (blue line) or when considering just 0.028% of the pixels (25 out of 89960 overall; orange line), on the Brisbane-Event-VPR dataset. One can observe a close correlation, indicating that the subset is representative of all pixels.

where $t_{k+1} = t_k + \tau$. Alternatively, events can also be accumulated so that each event frame contains a fixed number of events N (and event frames k and $k+1$ are non-overlapping):

$$\sum_{u,v} \mathbf{I}_{u,v}(k) = N. \quad (2)$$

In both cases, by considering the polarity p , two channels \mathbf{I}^+ (containing only events with $p = 1$) and \mathbf{I}^- ($p = -1$) can be created [9]. Unless otherwise stated, in the remainder of this paper, we accumulate events over a specified time window as in Eq. (1) and do not consider the polarity p .

B. Sparse Pixel Selection

In this work, rather than considering all pixels $\mathbf{u} \in (\{1, \dots, W\} \times \{1, \dots, H\})$, where W and H indicate the width and height of the image respectively, we instead select a small subset $\mathbf{S} \subsetneq \mathbf{I}$ of pixels. The motivation behind this choice is that we observed that small subsets \mathbf{S} are representative of the whole image, as illustrated in Fig. 2. We define $S_{u,v}$ as the variance in the number of events observed at pixel location (u, v) in the reference traverse¹:

$$S_{u,v} = \frac{1}{K} \sum_{k=1}^K (\mathbf{I}_{u,v}(t_k) - \mu)^2, \quad (3)$$

where μ is the mean number of events and K is the number of event frames.

To select the subset of pixels, we draw a sequence of samples based on the following probability mass function $p(u, v)$:

$$p(u, v) = P_{XY}(X = u, Y = v) = \frac{S_{u,v}}{A}, \text{ where} \quad (4)$$

$A = \sum_{u=1}^W \sum_{v=1}^H S_{u,v}$ is a normalization constant.

One could now sample from this probability mass function without replacement. However, to avoid selection of neighboring pixels, whose event rate is highly correlated, we employ surround suppression modeled by the inverse of a Gaussian probability density function. The methodology is detailed in the remainder of this subsection; and code implementing the sparse pixel selection is provided in the associated repository.

¹We note that using simply the mean number of events results in very similar behavior and performance, as mean and variance in this particular case are highly correlated.

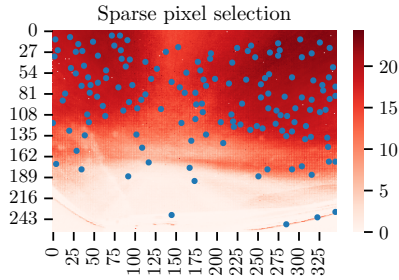


Fig. 3. **Event occurrences are unevenly distributed, and pixels with higher variance are selected.** To create this figure, we first accumulated events in $\tau = 1s$ windows across the whole “Sunset 1” reference traverse in the Brisbane-Event-VPR dataset into event windows. This figure shows the variance in the number of events per pixel across these event windows (see color bar). An example of the subset of pixels that were selected according to the probabilistic scheme described in Section III-B are shown with blue dots. The dark red area indicates that most selected pixels are in regions with a high variance (darker red indicates higher variance), while they are also spatially separated due to the surround inhibition. The light red areas at the bottom and center of the event frame correspond to the bonnet and street surface respectively, which (on average) have a much lower variance than the areas on the side where substantial vegetation and buildings are present.

Specifically, we first define the discrete bivariate Gaussian

$$\tilde{T}_{XY}(u, v, \sigma) = T_X(u, \sigma) T_Y(v, \sigma), \quad (5)$$

where $T_X(u, \sigma)$ and $T_Y(v, \sigma)$ are independent univariate Gaussian kernels along the u and v coordinates, respectively. The standard deviation σ of the Gaussian kernels is chosen to be the same across the u and v dimensions². The joint probability mass function $\hat{p}_{j+1}(u_{j+1}, v_{j+1})$ to select the $(j + 1)$ -th pixel in a recursive manner is then:

$$\hat{p}_{j+1}(u_{j+1}, v_{j+1}) = \tilde{A}_{j+1} \cdot \tilde{G}_{XY}(u_j, v_j, \sigma) \cdot \hat{p}_j(u_j, v_j), \quad (6)$$

where \tilde{G}_{XY} is the inverse distribution of \tilde{T}_{XY} , (u_j, v_j) denotes the coordinates of the j -th selected pixel, $\hat{p}_0(u_0, v_0) = p(u, v)$, and \tilde{A}_{j+1} is a normalization constant so that $\hat{p}_{j+1}(u_{j+1}, v_{j+1})$ is a valid probability mass function. We repeat this recursive process until we find a subset of pixels \mathbf{S} , where $\text{card}(\mathbf{S}) = J$. Fig. 3 shows an example of the selected sparse pixels.

C. Place Recognition Using a Small Subset of Pixels

We follow the typical place recognition setup where $\mathbf{S}^r(t_k)$ denotes the number of events observed at the selected pixels \mathbf{S} at the k -th place in the reference traverse (we use superscripts r and q to indicate reference and query traverses respectively). The aim is that given $\mathbf{S}^q(t_j)$ at query time, we want to find the best matching place k in \mathbf{S}^r .

The data association is performed using the sum-of-absolute-differences, as widely used for conventional camera images [28], [31], [42], but comparing the number of events rather than the intensity value, and only considering the subset of pixels:

$$\mathbf{D}(j, k) = \sum_{(u, v) \in \mathbf{S}} |\mathbf{S}^r(t_k) - \mathbf{S}^q(t_j)|, \quad (7)$$

where $\mathbf{D}(j, k)$ refers to the sum-of-absolute-differences between reference location j and query location k .

²Note that the variance $S_{u,v}$ in Eq. (3) and σ in Eq. (5) are not related. $S_{u,v}$ describes how much the event count varies over time, while σ describes the width of the Gaussian kernels that are used for surround suppression.



Fig. 4. **Recording setup for the QCR-Event-VPR dataset.** A DAVIS346 event camera (in red) was mounted on a Clearpath Jackal mobile robot. The camera can be installed forward-, side-, or downward-facing.

D. Using Spatio-Temporal Sequences

As the observed number of events within a few selected pixels is relatively low, and thus the expressiveness of a single difference value is limited, we rely on sequence matching [28] to combine the temporal information contained in L query event frames. Specifically, we use the computationally efficient 2D convolution-based implementation proposed in [32]:

$$\mathbf{D}_{seq}(j, k) = \mathbb{1}_L * \mathbf{D}_{seq}(j, k) / L, \quad (8)$$

where L describes the sequence length and $\mathbb{1}_L$ denotes the identity matrix of dimension $L \times L$.

In the experimental results, we show an ablation study that investigates the trade-off between using event frames that accumulate over longer time windows τ (or equivalently event frames with a bigger number of fixed events N) and using sequence matching that accumulates individual event frames over an equivalent time.

IV. EXPERIMENTAL SETUP

In this section, we first describe the datasets that we use in our experiments, namely the newly collected QCR-Event-VPR dataset and our previously introduced Brisbane-Event-VPR dataset (Section IV-A). We then set out the evaluation metrics in Section IV-B, and describe the baseline methods and pre-processing steps in Sections IV-C and IV-D respectively. Finally, we present implementation details in Section IV-E.

A. Datasets

We previously introduced the Brisbane-Event-VPR dataset [6] which was recorded in the suburb of Brookfield in Brisbane, Australia. The recording setup was such that a DAVIS346 camera was mounted forward-facing behind the windshield of a Honda Civic. The same 8km route was traversed six times at different times of the day and under different weather conditions. Following [6], we do not evaluate on the night-time traverse, as the camera parameters were not adjusted for these difficult lighting conditions. We extend Brisbane-Event-VPR to use the provided GPS information, instead of the manual labeling that was used in [6].

In this letter, we contribute the QCR-Event-VPR dataset, which was captured indoors in the QUT Centre for Robotics. The DAVIS346 was mounted on a Clearpath Jackal robot using

a flexible mount that enabled us to have the camera forward-, side-, or downward-facing. The same 160m route was traversed 16 times at different speeds and times of the day. We collected and evaluate on this dataset to demonstrate that our method generalizes to different recording setups, as well as to easily and safely adjust the velocities in which the route was traversed in (ranging from 154s to 262s, i.e. 70% difference in velocities). Fig. 4 shows the recording setup.

B. Evaluation Metrics

We use the standard information retrieval-based metrics that are commonly used in place recognition [1], [6]: precision $P = tp/(tp + fp)$ and recall $R = tp/(tp + fn)$. To create precision-recall curves, we sweep over the distance values in Eq. 7 for the system to decide whether to accept the match.

We report the precision at 100% recall (P@100R, i.e. what is the percentage of correct matches when the system is forced to match all query images to a database image) and recall at 99% precision (i.e. at the virtual absence of false data associations, how many query images were matched to database images). Following [6], [36], we allow a tolerance of 70 meters around the ground truth location as correct matches on the Brisbane-Event-VPR dataset. For the QCR-Event-VPR dataset, the tolerance is 3 meters.

C. Baseline Methods

We provide comparisons to several baseline methods: Firstly, the sum-of-absolute-differences of event counts when using all pixels (similar to [34] but without using patch normalization). For fair comparison, we use the same temporal binning as in our method (Eqs. 1 and 2). Secondly, we compare to two versions of our previous method, Ensemble-Event-VPR [6], that relies on the reconstruction of conventional images [43] and extracting NetVLAD features [35] from these reconstructed images. In one version, we use an ensemble that combines temporal window sizes of varying length, and in another we use a more computationally efficient method that uses a single temporal window size of 30ms to reconstruct the images. We note that the reconstructed images are of lower spatial resolution (346x260 pixels) and quality (see [6]) compared to conventional images taken with e.g. a smartphone camera, which impacts NetVLAD’s performance. Finally, we compare to Event-VPR by Kong *et al.* [36]³ and EventVLAD by Lee and Kim [37] (see Section II-C for a description of these methods).

D. Pre-processing

We follow the same pipeline as in [6] to remove bursts that are erroneously triggered for the majority of pixels within a very short time when sunlight hits the DAVIS346 bias generator, and to remove hot pixels. Such processing is needed due to the relative immaturity of event cameras, and is common in the event camera literature [6], [44].

³We note that Event-VPR [36] is unfairly disadvantaged as it does not use sequences. However, [36] has shown that it performs similar to Ensemble-Event-VPR [6], and thus the performance of [6] (where sequence matching is used for fair comparisons) can be used as a proxy for [36].

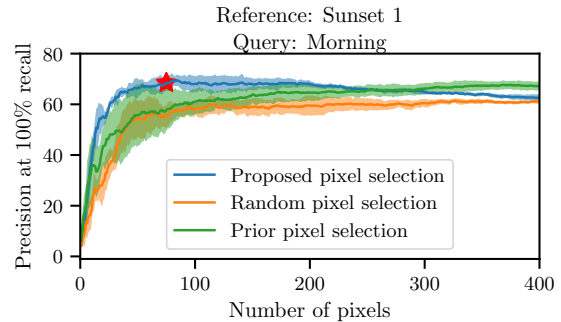


Fig. 5. **Comparing random with highly varying, discriminative pixels.** This figure simultaneously investigates the performance increase with an increasing number of pixels that are used to describe a place, and whether highly varying pixels (blue) perform better than random pixels (orange) or discarding the bottom third of the event frames a priori (green). The red star indicates the highest P@100R performance overall.

E. Implementation

We use Python alongside the NumPy, PyTorch and Tonic [45] libraries to implement our method. Unless otherwise noted, we use a time window length of $\tau = 1s$. We select $J = 150$ pixels based on Fig. 5 and the equivalent experiments for the other reference-query traverse pairs, which show that this is a high performing region. We use a sequence length of $L = 5$. The results are averaged over 5 trials; the selected pixels differ in each of the trials according to the random sampling strategy introduced in Section III-B. All experiments are conducted on a MacBook Pro M1 without using a GPU⁴.

V. RESULTS

In this section, we present results that answer our three research questions that we set out in Section I. Specifically, we first demonstrate that an intelligent selection of sparse pixels as introduced in Section III-B outperforms a random pixel selection (Section V-A). We then demonstrate that our approach is robust to moderate pixel image misalignments (Section V-B) and velocity variations (Sections V-C). We also compare the performance of our approach with two baseline methods, SAD and NetVLAD, and show that our approach performs similarly in terms of recall performance (Section V-D) but being computationally more lightweight (Section V-E). Finally, we briefly discuss whether our method is also applicable to conventional images (Section V-F).

A. Comparing Selection Strategies

Here we set out to compare our selection of highly varying pixels (Section III-B) with a random selection of pixels. We repeat each experiment five times, and report results on the Brisbane-Event-VPR dataset as the traverses are significantly longer than in the QCR-Event-VPR dataset. We select a sequence length of 5 (noting that we observed similar performance differences for other sequence lengths that are not shown due to space constraints).

⁴We do not make use of a GPU as our aim is a generically applicable method that can run on robotic systems that are power constrained.

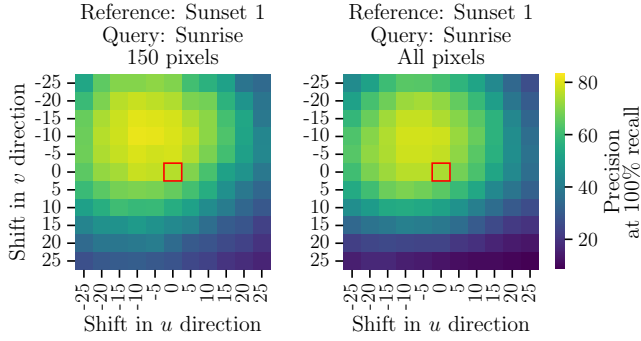


Fig. 6. **Robustness to pixel image misalignments.** This figure shows that the performance (as measured by P@100R) of our method (left plot) gracefully decays as the pixel shift increases. The performance decay is comparable to that of standard sum-of-absolute-differences where all pixels are used (right plot). Interestingly, the performance decay is not symmetric around the center (shown with a red border); the highest performance is observed at pixel offset $(-10, -15)$, indicating that there was a shift in camera position between reference and query times.

Figure 5 provides an example comparison on one reference (sunset 1) query (morning) combination. There are three interesting findings. Firstly, selecting highly varying pixels generally outperforms the random pixel selection, as well as a random pixel selection where with a priori knowledge no pixels are selected in the bottom third of the event frames (see Fig. 3). This is more pronounced the fewer pixels are being used. Secondly, adding more and more pixels does not necessarily result in a performance increase – in fact, in this particular example the best performance (highlighted by a red star) is obtained using just 83 pixels. Thirdly, combining the first two findings, the performance advantage of using highly varying pixels vanishes when using a large number of pixels. Averaged over the four different query traverses, the R@99P increased on average by 11.1%, and the P@100R by 5.3%.

B. Robustness to Pixel Image Misalignment

In our approach, we essentially compare the number of events that have occurred at particular pixel locations between the query and reference traverses. But what if the camera location changes between the traverses, so that pixel (u, v) at reference time does not correspond to the exact same pixel location at query time? We investigate this by artificially shifting all pixels (u, v) in the query traverse by $(\Delta u, \Delta v)$ so that $u^* = u + \Delta u$ and $v^* = v + \Delta v$, while leaving the reference traverse untouched.

Before presenting the results of this investigation, we want to note that 1) implicitly the robustness is shown in the Brisbane-Event-VPR results, as the camera mount was slightly moved between each recording session, and 2) the resulting pixel image misalignments are relatively minor, as opposed to major shifts in viewpoint which are still an open research problem in the visual place recognition field [1].

The results for this investigation is summarized in Fig. 6. There are two key take-aways: Firstly, the performance degrades gracefully, and secondly, the robustness to pixel image misalignments of our sparse method that uses just 150 pixels is roughly the same as when using all 89960 pixels.

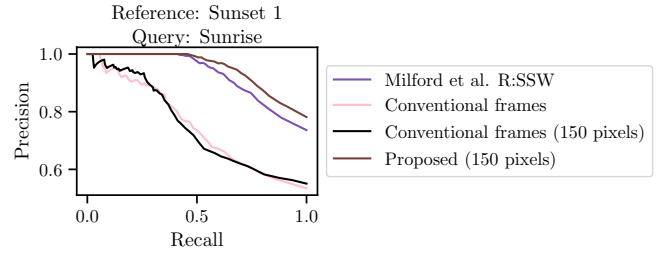


Fig. 7. **Application of sparse pixel selection to conventional images.** This figure investigates the performance of sparse pixel selection on conventional images. Similar performance trends as in the event-based case can be observed, with all pixels (pink) performing similarly to the sparse pixel selection (black). The performance on conventional images is significantly lower compared to the event stream, however noting that the DAVIS346 native RGB frames are of low quality.

C. Robustness to Velocity Changes

In our first research question, we asked ourselves whether the amount of dynamic change remains relatively constant when a place is repeatedly traversed. As defined in Section III-A, we accumulate events either over a given time or by splitting event frames such that each frame contains a fixed number of events N . It is often argued that the latter way preserves the data-driven nature of event cameras [46]. Indeed, in traverses that have significant velocity differences within the QCR-Event-VPR dataset, one can observe that the total number of events across the whole traverse remains roughly the same ($\pm 10\%$), while the duration changes significantly (70% change between reference and query traverse). This also implies that the dynamic change within each event frame remains similar between the reference and query traverses despite the velocity differences.

Indeed using event frames with a fixed N results in much higher performance, i.e. a P@100R of $79.0 \pm 1.4\%$ (averaged over 5 trials) compared to $20.6 \pm 1.3\%$ when using fixed τ . For fair comparison, we 1) used a temporal window size that is as long as the effective average window size when keeping N constant, 2) kept all other hyper-parameters the same, and 3) ensured that the same subset of pixels was used. In summary, our method displays promising robustness to change in velocity, which will be relevant in high-speed applications as further discussed in Section VI.

D. Comparison to State-of-the-Art

Precision-recall curves for comparisons with the state-of-the-art are shown in Fig. 8. Generally speaking, our method outperforms SAD [34] and EventVLAD [37] on all traverses. It also outperforms Event-VPR [36], however as noted in Footnote 3, Event-VPR is disadvantaged as sequence matching is not used. Our proposed method shows very similar performance when compared to a single model of Ensemble-Event-VPR [6], i.e. first reconstructing conventional images and applying NetVLAD on those images. The ensemble of 9 models as proposed in [6] outperforms our method, but is not applicable in practical scenarios due to the high computational demands as discussed next. These observations hold on both the Brisbane-Event-VPR dataset and the QCR-Event-VPR dataset.

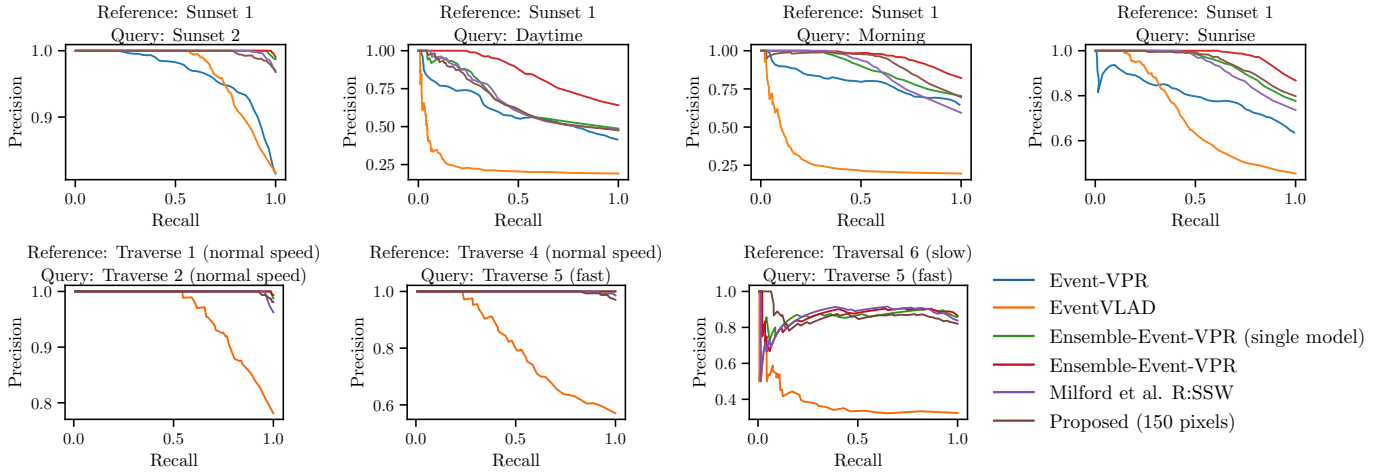


Fig. 8. **Precision-recall curves.** Top row: Results for the Brisbane-Event-VPR dataset. Following [6], the Sunset 1 traverse was used as reference traverse, and the remaining four traverses as query traverses. Our proposed method using 150 selected pixels (brown) generally outperforms using all pixels [34] (purple) and EventVLAD [37] (orange). Our method performs roughly on par with the much more computationally expensive Ensemble-Event-VPR [6] when one of their models is used individually (green), but is outperformed by their even more expensive ensemble variant [6] (red). Bottom row: Results on the QCR-Event-VPR dataset. The general trend of these experiments follows those observed on the Brisbane-Event-VPR dataset: Our method performs competitively while being significantly faster at inference time. Specifically, the left plot shows a traverse pair where the robot traveled at the same speed, the middle plot shows moderate speed variation, and in the right plot there is a large variation in speeds between the query and reference traverses.

E. Computational Efficiency

Our method using sparsely selected pixels is 50 times faster compared to SAD [34], with the calculation of Eq. 7 taking $\approx 1\text{ms}$ compared to $\approx 50\text{ms}$ when using all pixels. The computational advantage is even more pronounced in the comparison to [6]: $\approx 1\text{ms}$ for our method compared to over $\approx 1000\text{ms}$ for a single model in [6] (image reconstruction: $\approx 10\text{ms}$, NetVLAD feature extraction $\approx 1000\text{ms}$, distance matrix calculation: $\approx 8\text{ms}$) and over $\approx 2000\text{ms}$ when considering an ensemble of 9 models as proposed in [6]. We note that both EventVLAD [37] and Event-VPR [36] have similar runtimes (i.e. $\approx 1000\text{ms}$) as the single model version of [6] due to their architectural similarity. In summary, our method achieves competitive performance and can easily be deployed on systems that do not have a GPU.

F. Application to Conventional Images

In this section, we apply our sparse pixel selection to conventional camera-based place recognition. Instead of using the variance in the event counts, we simply used the variance in the intensity of the native RGB frames that are provided by the DAVIS346 camera (the DAVIS346 can simultaneously record events and RGB frames) in the selection process.

Fig. 7 shows that using the native RGB frames of the DAVIS camera (pink line) led to lower performance compared to using the event-based approach (purple line). We also found that the sparse pixels can be used as place recognition descriptors, even in the case of RGB frames (black line). Indeed, just like in the event-based case (brown line), the pixel subsampling led to similar performance as using all pixels, while being computationally more efficient. We note that these findings match those in [6] where a similar experiment that compared conventional image data with event data for visual place recognition was conducted. We want to emphasize that the native RGB frames provided by the DAVIS346 are of low quality, which affects the results.

VI. DISCUSSION AND CONCLUSIONS

In this paper, we presented a novel training-free method for event-based visual place recognition. Our main goal was to introduce a computationally lightweight method that is applicable in mobile robots. Despite our method’s simplicity and not requiring any training, we have demonstrated competitive performance when compared to the state-of-the-art. To foster future research in this area, we make both the QCR-Event-VPR dataset and our code available to the community.

Throughout the paper we have answered our research questions: We found that the dynamic change when traversing through a place remains relatively constant even under moderate changes of appearance, velocity and viewpoint (research question 1). We have further demonstrated that the event count of a small subset of pixels is sufficient to succinctly describe a place – which has the practical benefit of very low demands with respect to storage and compute (research question 2). Finally, we have shown that a high performing visual place recognition system can be obtained when using discriminative image regions – i.e. those that have a large variance in the number of events across the reference traverse (research question 3).

There are several directions for future works. Firstly, we could use adaptive duration event frames [20] or motion-corrected edge-like images via contrast maximization [47], as opposed to the simple accumulation of events to form event frames. It would also be interesting to use overlapping event frames at query time, which would enable place matching on an event-by-event basis. Secondly, our method could benefit from a more rigorous information-theory driven approach to the pixel selection [48]. Thirdly, we are interested in taking the precise timing and order of the events into account [49]. Taken together with sophisticated odometry [5], this could pave the way to place recognition in ultrasonic applications. Finally, we could imagine to apply our methodology to other camera

types, including light-field cameras that can trade-off spatial and temporal resolution [50], [51] – or even design cheap sensors that only capture a sparse representation of the scene.

REFERENCES

- [1] S. Garg, T. Fischer, and M. Milford, “Where is your place, Visual Place Recognition?” in *IJCAI*, 2021, pp. 4416–4425.
- [2] C. Masone and B. Caputo, “A survey on deep visual place recognition,” *IEEE Access*, vol. 9, pp. 19 516–19 547, 2021.
- [3] S. Lowry *et al.*, “Visual place recognition: A survey,” *IEEE Trans. Robot.*, vol. 32, no. 1, pp. 1–19, 2015.
- [4] C. Cadena *et al.*, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [5] M. Xu, T. Fischer, N. Sündenhauf, and M. Milford, “Probabilistic appearance-invariant topometric localization with new place awareness,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 6985–6992, 2021.
- [6] T. Fischer and M. Milford, “Event-Based Visual Place Recognition With Ensembles of Temporal Windows,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6924–6931, 2020.
- [7] G. Gallego *et al.*, “Event-based vision: A survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, 2020.
- [8] H. Rebecq, T. Horstschäfer, G. Gallego, and D. Scaramuzza, “EVO: A geometric approach to event-based 6-dof parallel tracking and mapping in real time,” *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 593–600, 2017.
- [9] A. I. Maqueda, A. Loquercio, G. Gallego, N. Garcia, and D. Scaramuzza, “Event-based vision meets deep learning on steering prediction for self-driving cars,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 5419–5427.
- [10] G. Orchard *et al.*, “HFirst: A temporal approach to object recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 10, pp. 2028–2040, 2015.
- [11] X. Lagorce, G. Orchard, F. Galluppi, B. E. Shi, and R. B. Benosman, “HOTS: A Hierarchy Of event-based Time-Surfaces for pattern recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1346–1359, 2016.
- [12] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. Benosman, “HATS: Histograms of Averaged Time Surfaces for robust event-based object classification,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 1731–1740.
- [13] S. Afshar *et al.*, “Event-based feature extraction using adaptive selection thresholds,” *Sensors*, vol. 20, no. 6, p. 1600, 2020.
- [14] D. Gehrig, A. Loquercio, K. G. Derpanis, and D. Scaramuzza, “End-to-end learning of representations for asynchronous event-based data,” in *IEEE Int. Conf. Comput. Vis.*, 2019, pp. 5633–5643.
- [15] R. Baldwin, R. Liu, M. M. Almatrafi, V. K. Asari, and K. Hirakawa, “Time-ordered recent event (TORE) volumes for event cameras,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022, to appear.
- [16] H. Rebecq, R. Ranfil, V. Koltun, and D. Scaramuzza, “High speed and high dynamic range video with an event camera,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 6, pp. 1964–1980, 2021.
- [17] C. Scheerlinck, H. Rebecq, N. Barnes, R. E. Mahony, and D. Scaramuzza, “Fast Image Reconstruction with an Event Camera,” in *IEEE Winter Conf. Appl. Comput. Vis.*, 2020, pp. 156–163.
- [18] L. Pan *et al.*, “High frame rate video reconstruction based on an event camera,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2519–2533, 2022.
- [19] M. Milford *et al.*, “Sequence searching with deep-learned depth for condition- and viewpoint-invariant route-based place recognition,” in *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2015.
- [20] M. Liu and T. Delbruck, “Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors,” in *Brit. Mach. Vis. Conf.*, 2018.
- [21] F. Gu *et al.*, “MDOE: A spatiotemporal event representation considering the magnitude and density of events,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 7966–7973, 2022.
- [22] J. Jiao *et al.*, “Comparing representations in tracking for event camera-based SLAM,” in *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2021, pp. 1369–1376.
- [23] M. Cannici, M. Ciccone, A. Romanoni, and M. Matteucci, “Attention mechanisms for object recognition with event-based cameras,” in *IEEE Winter Conf. Appl. Comput. Vis.*, 2019, pp. 1127–1136.
- [24] A. Renner, M. Evanusa, and Y. Sandamirskaya, “Event-based attention and tracking on neuromorphic hardware,” in *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2019.
- [25] Y. Li *et al.*, “Graph-based asynchronous event processing for rapid object recognition,” in *IEEE Int. Conf. Comput. Vis.*, 2021, pp. 934–943.
- [26] S. Schaefer, D. Gehrig, and D. Scaramuzza, “AEGNN: Asynchronous event-based graph neural networks,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 12 371–12 381.
- [27] M. Gehrig, S. B. Shrestha, D. Mouritzen, and D. Scaramuzza, “Event-based angular velocity regression with spiking networks,” in *IEEE Int. Conf. Robot. Autom.*, 2020, pp. 4195–4202.
- [28] M. J. Milford and G. F. Wyeth, “SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights,” in *IEEE Int. Conf. Robot. Autom.*, 2012, pp. 1643–1649.
- [29] F. Han, H. Wang, G. Huang, and H. Zhang, “Sequence-based sparse optimization methods for long-term loop closure detection in visual SLAM,” *Autonomous Robots*, vol. 42, no. 7, pp. 1323–1335, 2018.
- [30] S. M. Siam and H. Zhang, “Fast-SeqSLAM: A fast appearance based place recognition algorithm,” in *IEEE Int. Conf. Robot. Autom.*, 2017, pp. 5702–5708.
- [31] P. Hansen and B. Browning, “Visual place recognition using HMM sequence matching,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2014, pp. 4549–4555.
- [32] S. Garg, M. Vankadari, and M. Milford, “SeqMatchNet: Contrastive learning with sequence matching for place recognition & relocation,” in *Conf. Robot. Learn.*, 2022, pp. 429–443.
- [33] M. Milford, “Vision-based place recognition: how low can you go?” *Int. J. Robot. Res.*, vol. 32, no. 7, pp. 766–789, 2013.
- [34] M. Milford, H. Kim, S. Leutenegger, and A. Davison, “Towards Visual SLAM with Event-based Cameras,” in *Robotics: Science and Systems Workshops*, 2015.
- [35] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “NetVLAD: CNN Architecture for Weakly Supervised Place Recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1437–1451, 2018.
- [36] D. Kong *et al.*, “Event-VPR: End-to-end weakly supervised deep network architecture for visual place recognition using event-based vision sensor,” *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–18, 2022.
- [37] A. J. Lee and A. Kim, “EventVLAD: Visual place recognition with reconstructed edges from event cameras,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2021, pp. 2247–2252.
- [38] E. Mueggler, G. Gallego, H. Rebecq, and D. Scaramuzza, “Continuous-time visual-inertial odometry for event cameras,” *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1425–1440, 2018.
- [39] D. Liu *et al.*, “Asynchronous optimisation for event-based visual odometry,” in *IEEE Int. Conf. Robot. Autom.*, 2022, pp. 9432–9438.
- [40] W. Chamorro, J. Solà, and J. Andrade-Cetto, “Event-based line SLAM in real-time,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 8146–8153, 2022.
- [41] H. Kim, S. Leutenegger, and A. J. Davison, “Real-time 3d reconstruction and 6-dof tracking with an event camera,” in *Eur. Conf. Comput. Vis.*, 2016, pp. 349–364.
- [42] S. Hausler, A. Jacobson, and M. Milford, “Multi-process fusion: Visual place recognition using multiple image processing methods,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1924–1931, 2019.
- [43] C. Scheerlinck *et al.*, “Fast image reconstruction with an event camera,” in *IEEE Winter Conf. Appl. Comput. Vis.*, 2020, pp. 156–163.
- [44] C. Scheerlinck, N. Barnes, and R. Mahony, “Continuous-time intensity estimation using event cameras,” in *Asian Conf. Comput. Vis.*, 2018, pp. 308–324.
- [45] G. Lenz *et al.*, “Tonic: event-based datasets and transformations,” 2021, Documentation available under <https://tonic.readthedocs.io>.
- [46] G. Gallego and D. Scaramuzza, “Accurate angular velocity estimation with an event camera,” *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 632–639, 2017.
- [47] G. Gallego, H. Rebecq, and D. Scaramuzza, “A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 3867–3876.
- [48] A. Fontan, J. Civera, and R. Triebel, “Information-driven direct RGB-D odometry,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 4929–4937.
- [49] J. Dauwels, F. Vialatte, T. Weber, and A. Cichocki, “On similarity measures for spike trains,” in *Advances in Neuro-Information Processing*, 2008, pp. 177–185.
- [50] M. Gupta, A. Agrawal, A. Veeraraghavan, and S. G. Narasimhan, “Flexible voxels for motion-aware videography,” in *Eur. Conf. Comput. Vis.*, 2010, pp. 100–114.
- [51] S. Tambe, A. Veeraraghavan, and A. Agrawal, “Towards motion-aware light field video for dynamic scenes,” in *IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1009–1016.