








Event-VPR: End-to-End Weakly Supervised Deep Network Architecture for Visual Place Recognition using Event-based Vision Sensor

Delei Kong , Zheng Fang , *Member, IEEE*, Kuanxu Hou , Haojia Li , Junjie Jiang ,
Sonya Coleman , *Member, IEEE*, and Dermot Kerr 

Abstract—Traditional visual place recognition (VPR) methods generally use frame-based cameras, which will easily fail due to rapid illumination changes or fast motion. To overcome this, we propose an end-to-end visual place recognition network using event cameras, which can achieve good recognition performance in challenging environments (e.g., large-scale driving scenes). The key idea of the proposed algorithm is firstly to characterize the event streams with the EST voxel grid representation, then extract features using a deep residual network, and finally aggregate features using an improved VLAD network to realize end-to-end visual place recognition using event streams. To verify the effectiveness of the proposed algorithm, on the event-based driving datasets (MVSEC, DDD17, Brisbane-Event-VPR) and the synthetic event datasets (Oxford RobotCar, CARLA), we analyze the performance of our proposed method on large-scale driving sequences including cross-weather, cross-season and illumination changing scenes, and then we compare the proposed method with state-of-the-art event-based VPR method (Ensemble-Event-VPR) to prove its advantages. Experimental results show that the performance of the proposed method is better than that of event-based ensemble scheme in challenging scenarios. To our knowledge, for visual place recognition task, this is the first end-to-end weakly supervised deep network architecture that directly processes event stream data.

Index Terms—Visual place recognition, event camera, event spike tensor, deep residual network, triplet ranking loss.

I. INTRODUCTION

VISUAL place recognition (VPR) [1] [2] aims to help a robot or a vision-based navigation system determine whether it locates in a previously visited place. It is one of the essential and challenging problems in the field of computer vision and mobile robotics. These fields have witnessed a surge

Manuscript created December 23, 2021. This work was supported by National Natural Science Foundation of China (62073066, U20A20197), Science and Technology on Near-Surface Detection Laboratory (6142414200208), the Fundamental Research Funds for the Central Universities (N182608003), and Major Special Science and Technology Project of Liaoning Province (No.2019JH1/10100026), and Aeronautical Science Foundation of China (No.201941050001). (*Corresponding author: Zheng Fang.*)

Delei Kong is with College of Information Science and Engineering, Northeastern University, Shenyang, China (e-mail: kong.delei.neu@gmail.com).

Zheng Fang, Kuanxu Hou and Junjie Jiang are with Faculty of Robot Science and Engineering, Northeastern University, Shenyang, China (e-mail: fangzheng@mail.neu.edu.cn, 2001995@stu.neu.edu.cn, 2001998@stu.neu.edu.cn).

Haojia Li is with Robotics Institute, The Hong Kong University of Science and Technology, Hong Kong (e-mail: hlied@connect.ust.hk).

Sonya Coleman and Dermot Kerr are with Faculty of Computing, Engineering and Built Environment, Ulster University, Northern Ireland, UK (e-mail: sa.coleman@ulster.ac.uk, d.kerr@ulster.ac.uk).

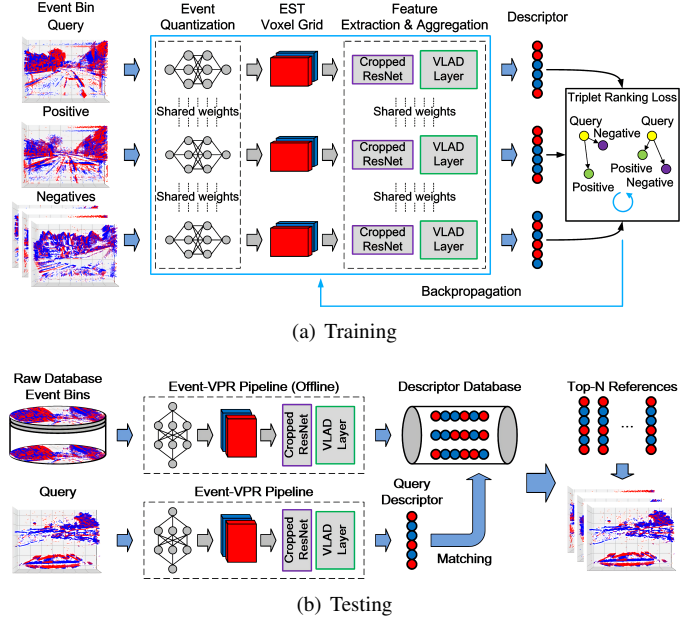


Fig. 1. Overview of the Proposed Event-VPR Architecture. From top to bottom: (a) In the training part, for a given event bin of query, we select the corresponding positive and negative event bins, and train the network model through the triplet ranking loss to learn the global descriptor vectors of the event bins. (b) In the testing part, for a given event bin of query, its descriptor vector is obtained through the trained network model, and several references are retrieved from the database of descriptor vectors calculated offline. (Note that the query, positive and negative samples in the figure are event bins denoting 3D event tensors, and the event frames in following sections are only for visualization.)

in the use of VPR for various applications in the last decade. In computer vision, VPR can be used to retrieve cross-time visual information and location information in a large-scale image database with geographic coordinates annotation, or be used in interactive 3D vision applications such as augmented reality (AR). In mobile robotics, the ability of robots to recognize visual places in GPS-denied environments is one of the key capabilities for autonomous localization and navigation. In simultaneous localization and mapping (SLAM), VPR is an important component of loop closure detection [3] [4], which can be used to detect candidate loop-closures and eliminate accumulated errors through global optimization for globally consistent pose estimation and mapping. In addition, visual place recognition can also perform precise visual localization

in a built environment map, and can be widely used in applications such as self-driving cars and service robots.

At present, there are many solutions to the VPR problem in large-scale environments. For those existing solutions, monocular, binocular, panorama cameras and other frame-based vision sensors are widely used when it comes to sensors. However, since frame-based vision sensors usually suffer from issues such as illumination change, motion blur and redundant information, this makes it difficult for traditional VPR methods to deal with recognition tasks in challenging environments. Additionally, in terms of algorithm principles, most existing methods are based on the appearance of scenes [5]. Due to various reasons, such as day-night, weather and seasonal changes, the appearance of the same place will change greatly at different times. And moreover, the appearance of different places at long distances may be very similar. These issues pose great challenges to the existing large-scale frame-based VPR methods.

In contrast to traditional frame-based VPR methods, we propose a novel VPR method using event cameras. Event cameras are neuromorphic visual sensors inspired by the biological retina, and work in a completely different way from frame-based cameras. They use an address-event representations (AER) and output pixel-level brightness changes (called events) with microsecond resolution to generate sparse asynchronous event streams [6] [7] [8] [9]. Event cameras have the advantages of low latency, high temporal resolution, low bandwidth, low power consumption and high dynamic range, which can effectively overcome the problems existing in typical frame-based cameras. Recently, Fischer [10] proposed a VPR method using event camera, however they used event streams to reconstruct image sequences, then the image sequences were used for VPR. In essence, it is still a VPR method using traditional images. **To achieve robust VPR directly using event streams, we propose a novel end-to-end event-based visual place recognition network architecture (Event-VPR).** The key idea is illustrated in Fig. 1, where an EST voxel grid representation generated by event streams are used as inputs in NetVLAD. To the best of our knowledge, this is the first end-to-end VPR method using event cameras. Experimental results on multiple datasets with different weather and scenes demonstrate that the proposed method is superior to the state-of-the-art event-based VPR method, and can effectively solve the challenges of large-scale scenes, high dynamic range and long-term adaptability in visual place recognition.

The main contributions of this paper are as follows:

- We propose a novel end-to-end weakly supervised network pipeline for visual place recognition (Event-VPR), which directly uses event streams from event camera as input. To our knowledge, this is the first end-to-end event stream-based VPR method ¹.
- We analyze and verify the effectiveness and robustness of the proposed method using multiple event camera-based driving datasets, including large-scale scene sequences

such as different weather, seasons, environments, and illumination conditions.

- Comprehensive comparisons between this method and event-based ensemble scheme [10] are carried out on the event camera-based driving dataset to evaluate the performance of both kind of VPR pipeline and prove the advantages of our method.
- Different event representations, network structures and loss functions of the proposed Event-VPR network are compared to show how they affect the overall performance.

The rest of this paper is organized as follows. Section II reviews related work on VPR using frame-based and event-based cameras. Section III describes the overall algorithm framework of our Event-VPR method, and introduces the representation learning of event-based data, feature extraction and description aggregation network, and network training process in detail. Next, the experimental results of Event-VPR on MVSEC, DDD17, Oxford RobotCar, Brisbane-Event-VPR and CARLA datasets are shown in Section IV. Finally, Section V concludes the paper.

II. RELATED WORK

A. Frame-based Methods

Visual sensors are the main sensor types for place recognition due to their low cost, low power consumption and abundant scene information. Nowadays, most popular VPR methods use frame-based visual sensors and appearance-based [5] approaches to realize large-scale place recognition. In this case, the VPR problem can generally be transformed into a large-scale geo-tagged image retrieval problem, and that can be solved by image matching. Extensive research on how to represent and match images has been carried out [1] [2]. Those approaches usually use traditional sparse feature extraction techniques (such as SIFT [11] and ORB [12]), and typical local descriptor aggregation techniques (such as BoW [3] [4] and VLAD [13] [14]) to establish a higher-order statistical model of image features. A typical work is DenseVLAD [15], which uses SIFT to extract intensive feature descriptors from images and uses VLAD for feature aggregation. With the rise of deep learning, off-the-shelf convolutional networks (such as Overfeat [16], VggNet and AlexNet [17]) are often used as trainable feature extractors. Modified versions of VLAD with a trainable pooling layer (such as NetVLAD [18]) were developed to obtain image descriptor vectors as compact image representations. In the retrieval and matching process, sequence-based matching is a widely recognized method. A well-known work is SeqSLAM [19], which searches highly similar short image sequences for VPR using the relative position information between consecutive frames. However, it has some issues in large-scale driving scenes such as low computational efficiency. Recently, researchers tried to further improve the recognition performance from different aspects. For example, some structure-based methods use structural information such as repeated edges and semi-dense maps of the scene [5], [20], [21] for place recognition. There are also some semantic-based works that mainly use semantic

¹Supplementary Material: An accompanying video for this work is available at <https://youtu.be/pcu118Wdc7g>.

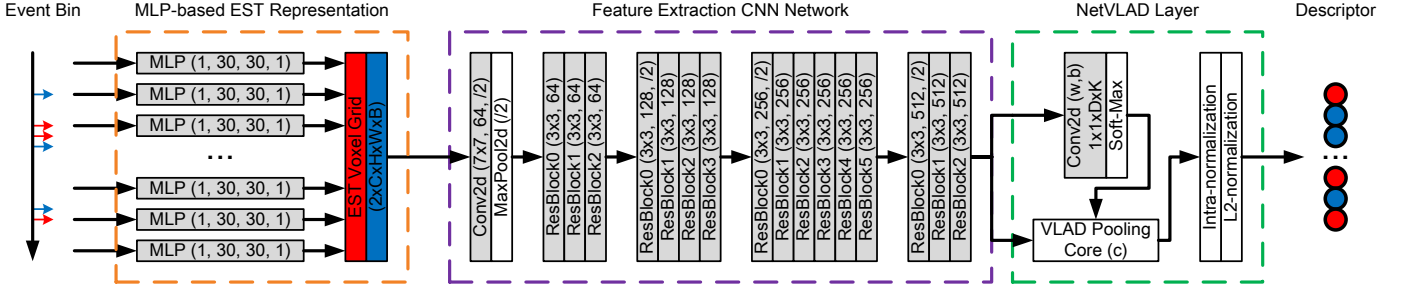


Fig. 2. Overview of the Proposed Pipeline. Firstly, the event bins are converted into EST voxel grids by a MLP-based kernel. Then, the cropped deep residual network ResNet34 is used to extract visual features of EST voxel grids. Next, VLAD-based local aggregated description layer is used for feature descriptor aggregation. Finally, the triplet ranking loss is used to train the network with weakly supervised training.

information such as landmarks, texts and objects in the scene to solve VPR problems. [22]–[24].

B. Event-based Methods

Though traditional frame-based VPR methods have been developed rapidly over the past decade, they still suffer from issues in challenging scenarios (such as illumination changes and motion blur) due to the inherent defects of frame-based cameras. Compared with standard frame-based cameras, event cameras have many advantages such as high dynamic range, high temporal resolution and low latency [8]. Due to these advantages, event cameras have drawn significant attention recently. However, to the best of our knowledge, there are still few works relating to event-based VPR. Among them, Milford et al. first tried to migrate SeqSLAM [19] to the event camera in 2018 and completed a relatively basic recognition experiment based on event frames [25]. More recently they proposed an event-based VPR scheme with ensembles of spatio-temporal windows (Ensemble-Event-VPR) [10]. This method uses event bins with different numbers of events and sizes of temporal windows to reconstruct a group of intensity frame sequences using E2Vid [26]. Then they compute the corresponding visual descriptors using the NetVLAD [18] pre-training model respectively, and perform approximate ensembles by using a strategy (such as averaging operation) on the distance matrix of the descriptors for place recognition. However, this method is not a direct event-based method but needs to convert the events into intensity frames, which is still a frame-based VPR method in essence. In addition, their ensemble scheme is computationally intensive and time-consuming due to the intensity reconstruction and ensemble using event bins with different lengths, which makes it difficult to achieve robust place recognition in large-scale scenes and deploy it on robots for real-time running. Different from the current event-based VPR approaches, we propose a novel end-to-end event-based visual place recognition network (Event-VPR) that directly uses event streams, which achieves excellent recognition results even in challenging environments.

III. METHODOLOGY

In this section, we will describe the network architecture and designing scheme of Event-VPR in detail, including the various module components of the proposed algorithm, the main steps and notes involved in network training.

A. Problem and Pipeline

Problem Definition. Define a database of events $D = \{P, E\}$ that contains n geo-location coordinates $P = \{P_1, \dots, P_n\}$ under a fixed reference system corresponding to n groups of event bins $E = \{E_1, \dots, E_n\}$, and each place coordinate P_i corresponds to several event bins $E_i = \{E_{i1}, \dots, E_{im}\}, i \in [1, n]$ (n, m is variable). Each event bin E_{ij} where $j \in [1, m]$ is collected by using an event camera in the area near the place coordinate P_i . The area of coverage (AOC) of all sub-areas is the same approximately, i.e. $S_{AOC}(P_1) \approx \dots \approx S_{AOC}(P_n)$. Hence, the problem of event-based VPR can be defined as follows. Given a query event bin denoted as E_q , the aim is to retrieve several event bins E_i which have most similarities to E_q from the database D thus obtaining its geographic location coordinates P_q according to P_i . To this end, we have designed a deep network to learn a function $f_{\text{Event-VPR}}(\cdot)$, which is used to map the query event bin E_q to a global descriptor vector $v_q = f(E_q)$ with fixed dimension such that $d(v_q, v_r) < d(v_q, v_s)$ if v_q is similar to v_r but different from v_s . Here $d(\cdot, \cdot)$ is the distance function (such as Euclidean distance). Then, the problem is simplified to find the place coordinates P_q of the sub-area, such that the global descriptor vector v_* from one of its event bins gives the minimum distance with the global descriptor vector v_q of the query, i.e. $d(v_q, v_*) < d(v_q, v_i), \forall i \neq *$. In practice, this can be done efficiently by a nearest neighbor search through a list of global descriptor vectors $\{v_i \mid i \in 1, 2, \dots, k\}$ that can be computed offline and stored in memory, while v_q is computed online.

The Proposed Pipeline. The key idea of the proposed Event-VPR algorithm is as follows. Firstly, we divide consecutive event streams into event bins and convert event bins into EST voxel grid representations using an MLP-based kernel. Then, a cropped deep residual network ResNet34 [27] is used to extract visual features from EST voxel grids. Next, a VLAD-based local aggregated description layer is used for feature descriptor aggregation. Finally, the triplet ranking loss is used to train the network with weakly supervised training. Corresponding to the aforementioned key idea, the proposed pipeline is divided into the following four parts: EST voxel grid representation, feature extraction convolution network, feature aggregated description layer and triplet ranking loss. The network architecture is illustrated in Fig. 2.

B. Events and EST Voxel Grid

Event Vision Sensor. Event camera [6] [7] [8] [9] is a bio-inspired neuromorphic vision sensor that works in a completely different way from standard cameras. The event camera does not output intensity frames at a fixed rate, but only outputs signals of local pixel-level brightness changes. When these pixel-level brightness changes (called events) exceed the set threshold, the event camera marks the timestamp with a microsecond resolution and outputs an asynchronous event stream. This event-based asynchronous data format is called address event representation (AER), which is used to simulate the transmission of neural signals in the biological vision system. In this way, information is continuously transmitted and processed, and the communication bandwidth is only occupied by the pixels that trigger the event.

Event-based Data. The pixel array of the event camera is capable of independently and logarithmically responding to pixel-level brightness changes (i.e. $L \doteq \log(I)$, where I is photocurrent) and triggering sparse asynchronous events $E = \{e_1, \dots, e_N \mid e_k \in \mathbb{N}^2 \times \mathbb{R}^+ \times [-1, 1], k \in [1, N]\}$. Without considering fixed pattern noise (FPN), the brightness change at the pixel $(x_k, y_k)^\top$ at time t_k is given by:

$$\begin{aligned} \Delta L(x_k, y_k, t_k) &= L(x_k, y_k, t_k) - L(x_k, y_k, t_k - \Delta t_k), \\ |\Delta L(x_k, y_k, t_k)| &\geq \vartheta, \end{aligned} \quad (1)$$

where Δt_k is the time interval between last triggered event and current triggered event of a pixel. When the brightness change of a pixel reaches the contrast threshold ϑ (here $\vartheta > 0$), the pixel triggers an event $e_k = (x_k, y_k, t_k, p_k)^\top$. Here, p_k is event polarity given by:

$$p_k = \frac{\Delta L(x_k, y_k, t_k)}{|\Delta L(x_k, y_k, t_k)|} = \{-1, 1\}. \quad (2)$$

In a real sensor, positive events (ON) and negative events (OFF) can be triggered according to different contrast thresholds ϑ . Furthermore, the events E can be summarized as an event measurement field with polarity defined on the 3D continuous spatio-temporal manifold:

$$S(x, y, t) = \sum_{e_k \in E} p_k \delta(x - x_k, y - y_k) \delta(t - t_k), \quad (3)$$

where (x, y, t) are the sampled grid space-time coordinates, $\delta(\cdot)$ denotes the Dirac pulse defined in the event domain and is used to replace each event.

EST Voxel Grid Representation. In order to use popular deep learning-based neural network techniques to extract visual features from sparse asynchronous event streams, we need to convert the event streams to a kind of representation that can be processed by convolutional networks. Typical representation methods for event streams include motion-compensated event frames (MCEF) [28], 4-channel event count and last-timestamp images (4CH) [29], and event voxel grid (EVG) [30]. In addition, the events can also be converted into traditional frame-based video (e.g. E2Vid) [26] [31] [32]. In this paper, we firstly divide event stream into event bins, then we use the voxel grid representation of event spike tensor (EST) to represent event bins. In order to obtain the

most meaningful visual feature information from the event measurement field, we convolve it with a trilinear voting kernel $k(x, y, t)$. Therefore, the convolution signal becomes:

$$\begin{aligned} (k * S)(x, y, t) &= \sum_{e_k \in E} p_k k(x - x_k, y - y_k, t - t_k), \\ k(x, y, t) &= \delta(x, y) \max\left(0, 1 - \left|\frac{t}{\Delta t}\right|\right). \end{aligned} \quad (4)$$

After the kernel convolution, the signal in Eq. (4) can be periodically sampled on the spatio-temporal coordinates (x, y, t) for voxel grid generation:

$$\begin{aligned} V_{\text{EST}}[x', y', t'] &= \sum_{e_k \in E} p_k \delta(x' - x_k, y' - y_k) \max\left(0, 1 - \left|\frac{t' - t_k}{\Delta t}\right|\right), \end{aligned} \quad (5)$$

where (x', y', t') are the sampled grid space-time coordinates, with $x' \in \{0, 1, \dots, W - 1\}$, $y' \in \{0, 1, \dots, H - 1\}$, $t' \in \{t_0, t_0 + \Delta t, \dots, t_0 + (C - 1)\Delta t\}$. Here, (W, H) is the spatial resolution of event camera, t_0 is the start timestamp, Δt is the size of time blocks, C is the number of time blocks (that is, the number of channels). In practice, we replace the manually designed kernel in Eq. (5) with a multi-layer perceptron (MLP) to generate an EST voxel grid as an end-to-end event representation, as illustrated in Fig. 3, where C is the number of channels and B is the batch size. The MLP receives the normalized timestamp of the event as input and has 2 hidden layers, each with 30 neurons. The value generated by the MLP for each event is put into the corresponding voxel grid coordinates. **Different from previous works on event stream representation, EST voxel grid can be learned and optimized according to specific tasks to maximise the performance of the whole network.**

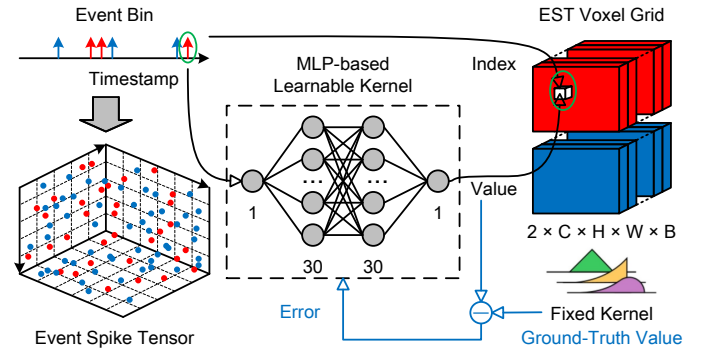


Fig. 3. Overview of Representation Learning for Asynchronous Event-based Data Using EST Voxel Grid. The value generated by the MLP-based kernel for each event is put into the corresponding coordinates, and kernel can use a pre-trained model or directly learned the representation through end-to-end training.

C. Feature Extraction and Aggregation

Feature Extraction Network. After converting the event bins into EST voxel grid representations, we need to extract features from them. To this end, we use the deep residual network ResNet34 which is designed for event-based handwritten number recognition tasks [27] as feature extraction network. In order to migrate it to our VPR task, the original network needs to be cropped. We modify the number of input channels of the first convolution layer of ResNet34 to make it

suitable for receiving a EST voxel grid V_{EST} as the network input. Meanwhile, in order to connect the feature description aggregation layers, we remove the last average pooling layer and the full connection layer at the end of the ResNet34. The network output $x = f_{ResNet34}(V_{EST})$ are the feature maps of the EST voxel grid. Here, x is the $w \times h \times D$ -dimensional feature tensor obtained by the feature extraction network.

Descriptor Aggregation Network. After getting feature maps of the EST voxel grid from the feature extraction network, we need to aggregate the features for descriptor matching. We use the vector of locally aggregated descriptor (VLAD) [13] [14] which is a trainable descriptor pooling method commonly used for place recognition and image retrieval. As illustrated in Fig. 4, we interpret the $w \times h \times D$ -dimensional feature maps x , output by the feature extraction network, as M D -dimensional local descriptors $\{x_1, \dots, x_M \mid x_i \in \mathbb{R}^D\}$ as input, and K D -dimensional cluster centers $\{c_1, \dots, c_K \mid c_k \in \mathbb{R}^D\}$ as VLAD parameters. The normalized output of the descriptor vector V_{VLAD} is a $D \times K$ -dimensional matrix, which is given by:

$$V_{VLAD,k}(x) = \sum_{i=1}^M \bar{a}_k(x_i)(x_i - c_k), \quad (6)$$

where $(x_i - c_k)$ is the residual vector of descriptor x_i to cluster center c_k , and $\bar{a}_k(x_i)$ denotes the soft assignment of descriptor x_i to cluster center c_k , which is given by:

$$\bar{a}_k(x_i) = \frac{e^{-\alpha \|x_i - c_k\|^2}}{\sum_{k'} e^{-\alpha \|x_i - c_{k'}\|^2}} = \frac{e^{w_k^\top x_i + b_k}}{\sum_{k'} e^{w_{k'}^\top x_i + b_{k'}}}. \quad (7)$$

It assigns the weight of descriptor x_i to cluster center c_k according to their proximity distance, where $w_k = 2\alpha c_k$, $b_k = -\alpha \|c_k\|^2$, $\alpha > 0$, and $\bar{a}_k(x_i) \in [0, 1]$. According to (7), soft assignment can be decomposed into a convolution layer and a soft-max layer, and the weight $\{w_k\}$ and bias $\{b_k\}$ of the convolution layer are taken as independent trainable parameters together with the clustering center $\{c_k\}$. Finally, the aggregated vector V_{VLAD} needs to be intra-normalized and L2-normalized to produce the final global descriptor vector $v \in \mathbb{R}^\Omega$, $\|v\|_2 = 1$, $\Omega = D \times K$ for event bins that can be used for efficient retrieval.

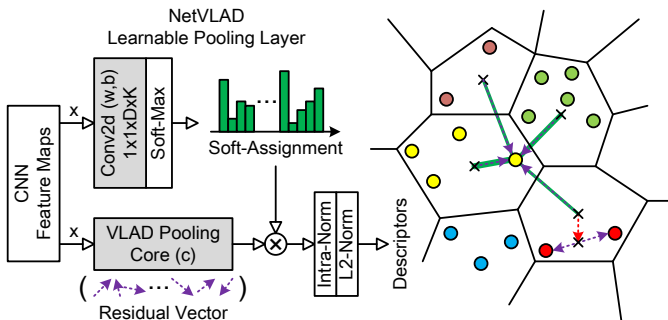


Fig. 4. Overview of VLAD-based Learnable Pooling Layer. Residual vector and soft assignment are statistics from descriptors to clusters, and the convolution layer and clusters are learnable through end-to-end training.

D. Network Training

Training Triplet Building. In order to enable the deep network to have robust place recognition capability, it is necessary to obtain similar descriptor vectors for the query and the samples from the same place in the database (positives), and obtain different descriptor vectors for the samples from different places in the database (negatives). Therefore, we use metric learning [33] [34] [35] to train Event-VPR in an end-to-end weakly supervised manner to learn the function $f_{Event-VPR}(\cdot)$ that maps the input of event bins E into global descriptor vectors $v \in \mathbb{R}^\Omega$, as illustrated in Fig. 5. In the process of network training, the training query event E_q and its geographical location P_q is given. It is necessary to select suitable best positive and hard negatives from the database $D = \{P, E\}$. The geographic distance $d_{geo}(\cdot)$ and $d_{des}(\cdot)$ between 2 samples (E and E') are defined as follows:

$$\begin{aligned} d_{geo}(P, P') &= \|P - P'\|_2, \\ d_{des}(v, v') &= 1 - \frac{v \cdot v'}{\|v\|_2 \|v'\|_2}, \end{aligned} \quad (8)$$

where the Euclidean distance is used for geographic distance, and cosine distance is used for descriptor distance. In order to improve the search efficiency, firstly, all samples within the range of geographical distance λ are selected as potential positives $\{E_{pos,i}\}$ according to the location P_q of the query:

$$d_{geo}(P_q, P_{pos,i}) \leq \lambda, \forall E_{pos,i} \in E. \quad (9)$$

Then, among these potential positives, the positive with the smallest descriptor vector distance is selected as the best positive, and its descriptor vector is:

$$v_{best-pos} = \arg \min_i d_{des}(v_q, v_{pos,i}). \quad (10)$$

When selecting negatives, firstly, all samples outside the range of geographical distance δ are selected according to the location P_q of the query, and then n_{sample} of them are selected as randomly sampled negatives $\{E_{neg,i}\}$:

$$d_{geo}(P_q, P_{neg,i}) \geq \delta, \forall E_{neg,i} \in E. \quad (11)$$

Among these randomly sampled negatives, the samples that violate the margin condition are selected as the candidate hard negatives $\{E_{hard-neg,i}\}$:

$$d_{des}(v_q, v_{hard-neg,i}) \leq d_{des}(v_q, v_{pos,j}) + \epsilon, \forall E_{hard-neg,i} \in E, \quad (12)$$

where ϵ is a constant parameter, representing the margin between $d_{des}(v_q, v_{pos,i})$ and $d_{des}(v_q, v_{neg,j})$. In order to improve the training efficiency, we select the n_{neg} samples with the smallest descriptor vector distance as the hard negatives for training, where $n_{neg} \in [0, N_{neg}]$, $n_{neg} \ll n_{sample}$. With the increase of training iterations and the convergence of the network, the number of hard negatives will gradually decrease or even cannot be found.

Loss Function. Based on the above method, we use the data from the event camera datasets to obtain a set of training triplets from the training set, where each triplet is represented as $\xi = (E_q, E_{best-pos}, \{E_{hard-neg,j}\})$. Here, E_q is the query, $E_{best-pos}$ is the best positive, and $\{E_{hard-neg,j}\}$ is a group of hard negatives. If their global descriptor vector is

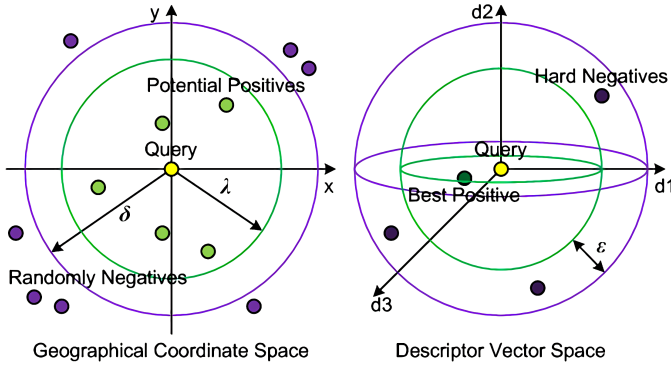


Fig. 5. Overview of Training Triplet Building. Here λ is the potential positive distance threshold, δ is the randomly negative distance threshold, and ϵ is the margin. (Note that the geographical coordinate space is a 2D space, and the descriptor vector space is a higher dimensional space, 3D space is only for visualization.)

$\gamma = (\mathbf{v}_q, \mathbf{v}_{\text{best-pos}}, \{\mathbf{v}_{\text{hard-neg},j}\})$, the descriptor vector distances of query \mathbf{E}_q to the best positive $\mathbf{E}_{\text{best-pos}}$ and the hard negatives $\{\mathbf{E}_{\text{hard-neg},j}\}$ are defined as $d_{\text{des}}(\mathbf{v}_q, \mathbf{v}_{\text{best-pos}})$ and $d_{\text{des}}(\mathbf{v}_q, \mathbf{v}_{\text{hard-neg},j})$ respectively. The loss function is designed to minimize the global descriptor distance between the query and the best positive, and to maximize the distance between the query and the hard negatives. We use weakly supervised triplet ranking loss which is defined as follows:

$$L_{\text{triplet}}(\gamma) = \sum_j [d_{\text{des}}(\mathbf{v}_q, \mathbf{v}_{\text{best-pos}}) - d_{\text{des}}(\mathbf{v}_q, \mathbf{v}_{\text{hard-neg},j}) + \epsilon]_+, \quad (13)$$

where ϵ denotes the margin, and $[\cdot]_+ = \max(\cdot, 0)$ means that the loss takes a positive number, i.e. $L_{\text{triplet}}(\gamma) \geq 0$.

In order to reduce the computation while ensuring the performance of the model, for each training triplet ξ and its corresponding global descriptor vector γ , we modify it to maximize the descriptor vector distance between the query \mathbf{E}_q and the nearest negative sample $\mathbf{E}_{\text{hard-neg}^*}$ in hard negatives $\{\mathbf{E}_{\text{hard-neg},j}\}$. Then the lazy triplet ranking loss is acquired, which is defined as follows:

$$\begin{aligned} L_{\text{lazy-triplet}}(\gamma) &= \max_j [d_{\text{des}}(\mathbf{v}_q, \mathbf{v}_{\text{best-pos}}) - d_{\text{des}}(\mathbf{v}_q, \mathbf{v}_{\text{hard-neg},j}) + \epsilon]_+ \\ &= [d_{\text{des}}(\mathbf{v}_q, \mathbf{v}_{\text{best-pos}}) - d_{\text{des}}(\mathbf{v}_q, \mathbf{v}_{\text{hard-neg}^*}) + \epsilon]_+ \end{aligned} \quad (14)$$

where the $\max(\cdot)$ operator is defined to select the global descriptor vector of the hardest negative sample $\mathbf{v}_{\text{hard-neg}^*}$ from the global descriptor vectors of hard negatives $\{\mathbf{v}_{\text{hard-neg},j}\}$.

However, the distance between $\mathbf{v}_{\text{hard-neg}^*}$ and the descriptor vector $\mathbf{v}_{\text{neg} \times}$ of other negative sample will be decreased when maximizing the descriptor distance between the query \mathbf{E}_q and the hardest negative sample $\mathbf{E}_{\text{hard-neg}^*}$, but the corresponding negative samples $\mathbf{E}_{\text{hard-neg}^*}$ and $\mathbf{v}_{\text{neg} \times}$ are from different places. To alleviate this issue, the extra distance $d(\mathbf{v}_{\text{hard-neg}^*}, \mathbf{v}_{\text{neg} \times})$ needs to be maximized. Here $\mathbf{E}_{\text{neg} \times}$ is a negative sample that selected randomly. Thus, the quadruplet ranking loss is defined as follows:

$$\begin{aligned} L_{\text{quadruplet}}(\gamma, \mathbf{E}_{\text{neg} \times}) &= L_{\text{triplet}}(\gamma) + [d_{\text{des}}(\mathbf{v}_q, \mathbf{v}_{\text{best-pos}}) - d_{\text{des}}(\mathbf{v}_q, \mathbf{v}_{\text{neg} \times}) + \epsilon']_+ \end{aligned} \quad (15)$$

and the corresponding lazy quadruplet ranking loss is defined as follows:

$$\begin{aligned} L_{\text{lazy-quadruplet}}(\gamma, \mathbf{E}_{\text{neg} \times}) &= L_{\text{lazy-triplet}}(\gamma) + [d_{\text{des}}(\mathbf{v}_q, \mathbf{v}_{\text{best-pos}}) - d_{\text{des}}(\mathbf{v}_q, \mathbf{v}_{\text{neg} \times}) + \epsilon']_+ \end{aligned} \quad (16)$$

where ϵ' denotes another interval margin. All above 3 ranking loss functions can be considered as variants of triplet ranking loss, which have been shown to be very effective in large-scale image retrieval tasks such as face recognition [36] and pedestrian re-identification [37].

Caching. In the process of network training, it is necessary to retrieve the descriptor vector of samples to calculate the descriptor distance. In order to improve the training efficiency, we build a cache for the descriptor vectors of the entire database and use the cached descriptor vectors to select the best positive and hard negatives. As the model parameters will be continuously adjusted during the training process and the descriptor vectors of the network output will also be continuously changed, we will update the cache regularly during the network training. Recalculating the cache every 500-1000 training queries can achieve a balance between the duration of the training epoch, the convergence speed of the network and the quality of the model.

Training Data Loading. Different from traditional synchronous image frames, events are triggered asynchronously, which are related to the specific scene and movement. Sparse event bins have the issue that the amount of data is not fixed in the process of data loading, which makes it difficult to distinguish each sample during batch training. When we build a batch process, we add a column of indexes to each event bin, arrange the query, the best positive and the hard negatives in turn, then merge them into a batch process and send them to the network for weakly supervised training. After that, the descriptor vectors are divided according to the indexes. The processing of event bins when loading a batch of training data $\mathbf{E}_{\text{input}}$ is given as follows:

$$\mathbf{E}_{\text{input}} = \sum_{i=0}^{B-1} \mathbf{E}_q^i \oplus \sum_{i=0}^{B-1} \mathbf{E}_{\text{best-pos}}^i \oplus \sum_{i=0}^{B-1} \sum_{x=1}^{n_{\text{neg}}} \mathbf{E}_{\text{hard-neg},x}^i, \quad (17)$$

where B is the batch size, $n_{\text{neg}} = \{n_0, \dots, n_{B-1}\} \in [0, N_{\text{neg}}]$ is the number of negatives in each batch, and \oplus denotes splicing each event bins.

Training Data Augmentation. We use the method similar to EventDrop [38] to enhance the original event stream data. We increase the diversity of training data by dropping events selected with various strategies (e.g. random drop, drop by timestamp, and drop by pixel area) to improve the generalization performance of our Event-VPR network model.

IV. EXPERIMENTS

In this section, for performance evaluation, we conducted experiments on multiple datasets such as MVSEC [39], DDD17 [40], Oxford RobotCar [41] Brisbane-Event-VPR [10] and CARLA [42], to verify the effectiveness of the proposed method through quantitative and qualitative experiments. We conducted three experiments to evaluate our proposed method. Firstly, we evaluated the performance of the Event-VPR in

TABLE I
THE SEQUENCES OF THE MVSEC [39], DDD17 [40], OXFORD ROBOTCAR [41], BRISBANE-EVENT-VPR [10] AND CARLA [42] DATASETS USED IN OUR EXPERIMENTS.

| Datasets | Scenarios | Sequences |
|-------------------------|--|---|
| MVSEC [39] | day night | outdoor_day1 / outdoor_day2 outdoor_night1 / outdoor_night2 / outdoor_night3 |
| DDD17 [40] | city town freeway (day) freeway (night) | rec1487839456 / rec1487842276 / rec1487844247 rec1487846842 / rec1487849151 / rec1487849663 rec1487417411 / rec1487419513 / rec1487430438 rec1487350455 / rec1487608147 / rec1487609463 |
| Oxford RobotCar [41] | sun/cloud overcast rain snow night | 2014-11-18-13-20-12 / 2015-03-10-14-18-10 / 2015-07-29-13-09-26 / 2015-09-02-10-37-32 2015-02-13-09-16-26 / 2015-05-19-14-06-38 2014-11-25-09-18-32 / 2014-12-05-11-09-10 2015-02-03-08-45-10 2014-11-14-16-34-33 / 2014-12-10-18-10-50 |
| Brisbane-Event-VPR [10] | sunrise (sr) morning (mn) daytime (dt) sunset (ss) | 2020-04-29-06-20-23 2020-04-28-09-14-11 2020-04-24-15-12-03 2020-04-22-17-24-21 |
| CARLA [42] | clearnoon (clnno) clearsunrise (clrsr) clearsunset (clrss) cloudynoon (cldnn) | N/A N/A N/A N/A |

different driving scenarios in MVSEC, DDD17 and Oxford RobotCar datasets and verified its long-term robustness. Then, we quantitatively compared and analyzed the performance of Event-VPR and the ensemble scheme of spatio-temporal windows (Ensemble-Event-VPR [10]) in detail on Brisbane-Event-VPR and CARLA datasets. Finally, we performed the network ablation study on the proposed Event-VPR from 3 aspects (event representation, feature extraction network and loss function), and the experimental results proved the advantages of each module in the proposed approach.

A. Experiment Setup

Dataset Selection. Since there is no dataset for place recognition of the event camera at present, we selected and modified several currently open datasets of driving scenes for our experiments. The scenes of the experimental datasets are shown in Fig. 6. Among them, the MVSEC [39] and DDD17 [40] datasets are the existing event-based driving datasets recorded in the real environment. We selected intensity images and event bins of 5 outdoor driving sequences from the MVSEC dataset (including day-and-night scenes, using left-eye DAVIS camera) and 12 outdoor driving sequences from the DDD17 dataset (including urban, town and freeway scenes), which include rapid illumination changes and scene structure changes (Fig. 6(a) and 6(b)). In addition, the Oxford RobotCar [41] dataset is a standard dataset commonly used in VPR. We use the event synthesizer V2E [43] [44] to convert the image sequences, recorded by the center camera of the three-eye vision sensor (Bumblebee XB3), into corresponding event streams. As far as possible, we selected sequences with the same trajectory under different weather and season conditions, including sunny, cloudy, rainy, snowy, dusky and night scenes throughout the year (Fig. 6(c)). In the work of Tobias et

al. [10], the Brisbane-Event-VPR dataset they provide is an event-based VPR dataset for outdoor driving scenes, including sequences with the same trajectory under different illumination conditions such as sunrise, morning, daytime, sunset and night (Fig. 6(d)). And we used CARLA [42] simulator to record several event-based driving sequences with the same trajectory under different weather and illumination conditions for further evaluation (Fig. 6(e)). The illumination and appearance of the above scenarios are quite different. We consider such scenarios to be more challenging for testing the performance of VPR methods, which can better verify the robustness of our algorithm.

Dataset Configuration. We randomly divide the sequences of the same trajectory into geographically non-overlapping training and test sets (detailed results are shown in Table I and Table II). In the MVSEC dataset, we select approximately 40k training samples and 10k test samples from 5 sequences. In the DDD17 dataset, we select approximately 240k test samples from 12 sequences. From the Oxford RobotCar dataset, we also select approximately 50k training samples and 12k test samples from 11 sequences. In the Brisbane-Event-VPR dataset, we selected approximately 24k training samples and 3k test samples from 5 sequences. Since the sequences of the MVSEC and DDD17 datasets come from different trajectories, we randomly sample and divide sequences into database sets and query sets. The database set accounts for 70% of the total number of sequence samples and the query set accounts for 30%. In the Oxford RobotCar dataset, all sequences are from the same trajectory. In order to verify the performance of the model across seasons and weather, we put together multiple sequences of the same season / weather, and then use random sampling to divide database sets and query sets. In the Brisbane-Event-VPR dataset and the CARLA dataset,

TABLE II
SEQUENCE DIVISION AND PARAMETER CHOICES OF THE MVSEC [39], DDD17 [40], OXFORD ROBOTCAR [41], BRISBANE-EVENT-VPR [10] AND CARLA [42] DATASETS USED IN OUR EXPERIMENTS.

| Datasets | Train sequences (database / query) | Test sequences (database / query) | Train database / query $N_{db,train} / N_{q,train}$ | Test database / query $N_{db,test} / N_{q,test}$ | Random negatives $N_{n,r}$ | Hard negatives $N_{n,h}$ |
|--------------------------------|---------------------------------------|--------------------------------------|--|---|-------------------------------|-----------------------------|
| MVSEC [39] | day2 | day1 | 20008 / 8575 | 8355 / 3582 | 300 | 10 |
| | day1 | day2 | 8355 / 3582 | 20008 / 8575 | 500 | 10 |
| | night2 & night3 | night1 | 4701 / 2016 | 1892 / 812 | 100 | 10 |
| | night1 & night3 | night2 | 3890 / 1668 | 2704 / 1159 | 100 | 10 |
| | night1 & night2 | night3 | 4596 / 1971 | 1997 / 857 | 100 | 10 |
| DDD17 [40] | sequences in Oxford RobotCar | city1 | 24658 / 10568 | 2849 / 1221 | 500 | 10 |
| | | city2 | | 3863 / 1656 | 100 | 10 |
| | | city3 | | 3661 / 1570 | 100 | 10 |
| | | town1 | | 12593 / 5398 | 300 | 10 |
| | | town2 | | 3006 / 1289 | 100 | 10 |
| | | town3 | | 20041 / 8590 | 500 | 10 |
| | | freeway1 | | 14664 / 6285 | 500 | 10 |
| | | freeway2 | | 20892 / 8954 | 500 | 10 |
| | | freeway3 | | 21935 / 9401 | 500 | 10 |
| | | freeway4 | | 9816 / 4207 | 300 | 10 |
| | | freeway5 | | 8450 / 3622 | 300 | 10 |
| | | freeway6 | | 10199 / 4372 | 300 | 10 |
| Oxford RobotCar [41] | all winter | all spring | 19933 / 8514 | 12602 / 5401 | 500 | 10 |
| | all winter | all summer | 19933 / 8514 | 8401 / 3601 | 500 | 10 |
| | all winter | all autumn | 19933 / 8514 | 8401 / 3601 | 500 | 10 |
| | sun & cloud | cloud | 12277 / 5261 | 4200 / 1801 | 500 | 10 |
| | sun & cloud | overcast | 16477 / 7062 | 8401 / 3601 | 300 | 10 |
| | sun & cloud | rain | 16477 / 7062 | 7636 / 3273 | 300 | 10 |
| | sun & cloud | snow | 16477 / 7062 | 4200 / 1801 | 300 | 10 |
| | sun & cloud | night | 16477 / 7062 | 8401 / 3601 | 300 | 10 |
| | night | cloud | 8401 / 3601 | 4200 / 1801 | 300 | 10 |
| | night | overcast | 8401 / 3601 | 8401 / 3601 | 300 | 10 |
| | night | rain | 8401 / 3601 | 7636 / 3273 | 300 | 10 |
| | night | snow | 8401 / 3601 | 4200 / 1801 | 300 | 10 |
| | night | night | 4200 / 1801 | 4200 / 1801 | 300 | 10 |
| | night | night | 4200 / 1801 | 4200 / 1801 | 300 | 10 |
| Brisbane- Event-VPR [10] | (dt & mn) / sr | ss1 / ss2 | 9659 / 5365 | 578 / 566 | 300 | 10 |
| | (ss2 & mn) / sr | ss1 / dt | 8716 / 5365 | 578 / 557 | 300 | 10 |
| | (ss2 & dt) / sr | ss1 / mn | 8201 / 5365 | 578 / 570 | 300 | 10 |
| | (ss2 & dt) / mn | ss1 / sr | 8201 / 5087 | 578 / 574 | 300 | 10 |
| CARLA [42] | cldnn / clrss | clrnn / clrsr | 3345 / 3291 | 3217 / 3184 | 300 | 10 |
| | cldnn / clrsr | clrnn / clrss | 3345 / 3184 | 3217 / 3291 | 300 | 10 |
| | clrsr / clrss | clrnn / cldnn | 3184 / 3291 | 3217 / 3345 | 300 | 10 |
| | clrss / cldnn | clrsr / clrnn | 3291 / 3345 | 3184 / 3217 | 300 | 10 |
| | clrnn / cldnn | clrsr / clrss | 3217 / 3345 | 3184 / 3291 | 300 | 10 |
| | clrss / clrnn | clrsr / cldnn | 3291 / 3217 | 3184 / 3345 | 300 | 10 |
| | clrsr / cldnn | clrss / clrnn | 3184 / 3345 | 3291 / 3217 | 300 | 10 |
| | cldnn / clrnn | clrss / clrsr | 3345 / 3217 | 3291 / 3184 | 300 | 10 |
| | clrsr / clrnn | clrss / cldnn | 3184 / 3217 | 3291 / 3345 | 300 | 10 |
| | clrss / clrsr | cldnn / clrnn | 3291 / 3184 | 3345 / 3217 | 300 | 10 |
| | clrnn / clrss | cldnn / clrsr | 3217 / 3291 | 3345 / 3184 | 300 | 10 |
| | clrnn / clrsr | cldnn / clrss | 3217 / 3184 | 3345 / 3291 | 300 | 10 |
| | clrnn / clrsr | cldnn / clrss | 3217 / 3184 | 3345 / 3291 | 300 | 10 |

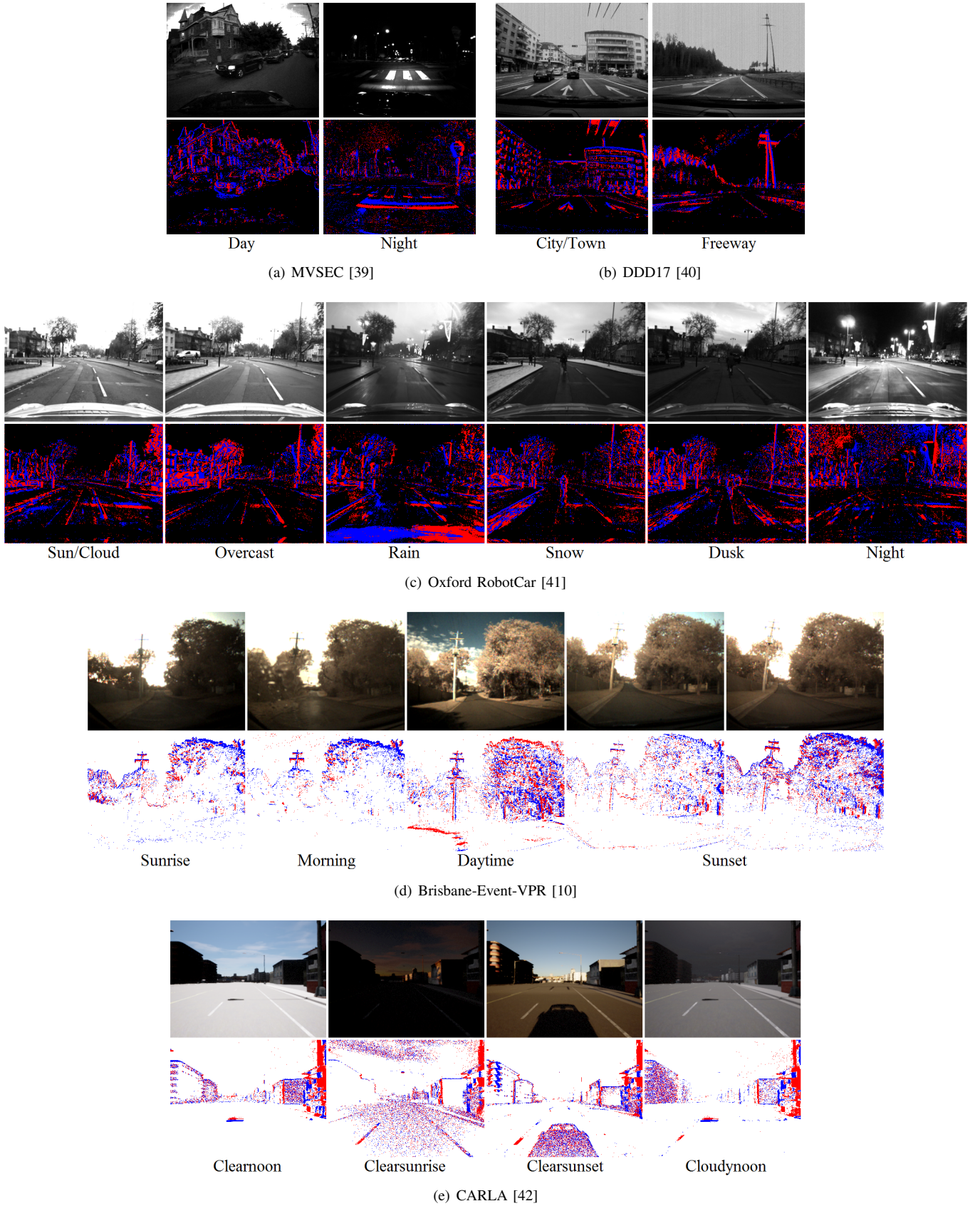


Fig. 6. The Scenarios of MVSEC [39], DDD17 [40], Oxford RobotCar [41], Brisbane-Event-VPR [10] and CARLA [42] Datasets in Our Experiments. From top to bottom: (a) The day and night scenes for the MVSEC Dataset. (b) The city/town and freeway scenes for the DDD17 Dataset. (c) Synthetic event bins for the Oxford RobotCar Dataset using V2E [43] [44], we selected sequences with the same trajectory as far as possible, covering scenes in different weather and seasons. (d) The scenes for the Brisbane-Event-VPR dataset including scenes under different illumination conditions. (e) The scenes for the CARLA dataset including scenes under different weather and illumination conditions. (Note that the samples we used in the figure are event bins denoting 3D event tensors, and the intensity / event frames are only for visualization.)

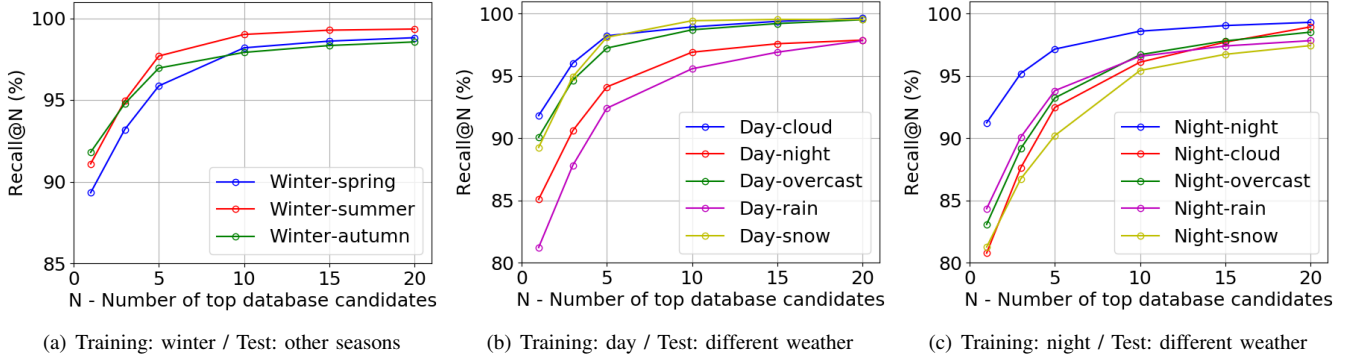


Fig. 7. Recall@N of Our Event-VPR on Different Season and Weather Sequences in the Oxford RobotCar Dataset [41]. From left to right: (a) Training: All winter sequences, Test: Spring, summer and autumn sequences, (b) Training: Day-time sequences (sun and cloud), Test: Different scene sequences (cloud, night, overcast, rain and snow), (c) Training: Night-time sequences, Test: Different scene sequences (night, cloud, overcast, rain and snow).

all sequences are also from the same trajectory. In order to facilitate the comparison with the ensemble scheme [10], we use different sequences as the database sets and query sets for cross training. For parameters, we choose potential positive distance threshold $\lambda = 10\text{m}$, randomly negative distance threshold $\delta = 25\text{m}$, and true positive geographical distance threshold $\varphi = 20\text{m}$ from the MVSEC and Oxford Robocar datasets. Moreover, in the DDD17 dataset, potential positive distance threshold $\lambda = 30\text{m}$, randomly negative distance threshold $\delta = 65\text{m}$, and true positive geographical distance threshold is $\varphi = 60\text{m}$. In addition, we choose potential positive distance threshold $\lambda = 35\text{m}$, randomly negative distance threshold $\delta = 75\text{m}$, and true positive geographical distance threshold $\varphi = 70\text{m}$ from the Brisbane-Event-VPR dataset and our recorded CARLA dataset.

Evaluation Indices. In our experiments, we use Recall@N and PR curve, two common performance evaluation indicators for VPR, to evaluate the performance of the proposed method. Specifically, when calculating Recall@N, for each query, the N-nearest neighbor database samples are retrieved as positive samples according to the descriptor distance. If at least one of the samples is less than φ away from the query according to their geographical distance, it is considered to be correctly identified. Then, we calculate the Recall@N with different N to correctly identify the query. For the query set, we traverse descriptor vectors for all queries to calculate Recall@N which is the percentage of correctly identified query descriptors. In the following experiments, we restrict the analysis to $N = \{1, 3, 5, 10, 15, 20\}$. When drawing PR curves, for each query sample, it is first necessary to determine whether the nearest neighbor database sample retrieved by the model is a positive sample or a negative sample according to the descriptor distance threshold ϑ . If the descriptor distance $d_{des}(v_q, v_{db}) \leq \vartheta$, it is considered as a positive sample, otherwise it is considered as a negative sample. Here, $\vartheta \in [\min d_{des}(v_q, v_{db}), \max d_{des}(v_q, v_{db})]$. Then, the positive samples and negative samples are judged respectively according to the geographical distance threshold φ , and true positive (TP), false positive (FP), true negative (TN) and false negative (FN) samples are obtained for recall and precision calculation. Finally, for different descriptor distance thresholds

$\vartheta_k, k \in [0, K]$ as follows:

$$\vartheta_k = \frac{k}{K} \cdot \max d_{des}(v_q, v_{db}) - \left(1 - \frac{k}{K}\right) \cdot \min d_{des}(v_q, v_{db}), \quad (18)$$

the recall and precision are calculated respectively, and PR curves are drawn.

B. Performance Analysis in Different Scenarios

In the first experiment, we validated the performance of the Event-VPR using different scenarios from the MVSEC, DDD17 and Oxford RobotCar datasets. We selected several challenging sequences for the experiments. For example, the night scene sequence in the MVSEC dataset has low illumination with little light. There are many similar scenes and high dynamic range scenes (e.g. clouds and tunnels) in the freeway scenes in the DDD17 dataset. There are also changing weather and seasonal conditions in the Oxford RobotCar dataset. For instance, there are light reflections from the surface due to water on a rainy day and the bright street lamps at night, which result in a high dynamic range scenario for visual cameras. Our experiment does not use the dusk sequence since the poor GPS values of those sequences are not good enough to be considered as ground truth data.

Results on MVSEC. As shown in Table III, the experimental results on the MVSEC dataset show the recognition performance (Recall@N) of our method in day-time and night-time scenes. On the MVSEC dataset, for 2 day-time sequences and 3 night-time sequences, we use cross-validation method to train our network models and test their performance on different sequences respectively. The results show that the Recall@1 of the model in night sequences can achieve 97.05% on average, which is almost the same as that in the day-time sequences. Except as described, since the five outdoor driving sequences on the MVSEC dataset are from different trajectories, it is not possible to compare the place recognition performance together across day and night. However, the experimental results in the following Oxford RobotCar dataset show this performance.

Results on Oxford RobotCar. Fig. 7 shows the recognition performance of Event-VPR network models under various

TABLE III
RECALL@N OF OUR EVENT-VPR ON SEVERAL SEQUENCES IN THE
MVSEC [39] DATASET

| Training | Test | Recall@1 | Recall@5 | Recall@10 |
|-----------------|--------|---------------|----------|-----------|
| day2 | day1 | 99.51% | 99.84% | 99.98% |
| day1 | day2 | 91.52% | 97.57% | 99.17% |
| night2 & night3 | night1 | 98.67% | 99.33% | 99.95% |
| night1 & night3 | night2 | 95.11% | 98.22% | 99.11% |
| night1 & night2 | night3 | 97.37% | 98.68% | 99.34% |

weather conditions and different seasons on the Oxford RobotCar dataset. First, the experimental results from the model trained on all winter sequences and tested on different weather sequences in three other seasons (spring, summer and autumn) are shown in Fig. 7(a). The results demonstrate that our model can achieve similar recognition performance in different seasons, and can realize long-term visual place recognition across seasons. Next, we trained network models separately using all day-time sequences and all night-time sequences, and tested with the unseen scene sequences (e.g. overcast, rain, and snow). The experimental results of the model trained on all day-time sequences (only including sun and cloud) and tested on different sequences (including cloud, night, overcast, rain and snow) respectively are shown in Fig. 7(b). We noticed that the test results on the overcast and snow sequences are almost the same as those on cloud sequences, and the difference between Recall@1 is only approximately 2.15%. The test results on the rain and night sequences are slightly poor, with Recall@1 being 8.66% lower on average. The experimental results from the model trained on all night-time sequences and tested on different sequences (including night, cloud, overcast, rain and snow) respectively are shown in Fig. 7(c). Due to the significant amount of noisy events in the night sequences, the trained model has slightly poor performance in all the day-time scene sequences.

Results on DDD17. In order to verify the generalization capability of the proposed Event-VPR network, we use the network model trained on synthesis event streams from the Oxford RobotCar dataset, and then we test the trained model on various scene sequences (e.g. city, town and freeway) from the DDD17 dataset recorded using a DAVIS event camera, and the test results of all sequences are shown in Table IV. Surprisingly, without any transfer learning or fine-tuning of the trained network, the results show that our method can achieve similar high performance on urban scene sequences of the DDD17 dataset as on urban scene sequences of the Oxford RobotCar dataset. Even on very challenging freeway scene sequences with a large number of similar appearances and high dynamic range scenes, the Recall@1 of test model can still achieve about 77.35% on average. It means that just by using artificial synthesis event streams to train the Event-VPR network model, the high performance can be obtained on real event streams data recorded by the event camera.

C. Performance Comparison with Ensemble-Event-VPR

In the second experiment, we compared our Event-VPR to the Ensemble-Event-VPR [10] scheme using the Brisbane-

TABLE IV
RECALL@N OF OUR EVENT-VPR MODEL TRAINED USING THE
SYNTHETIC EVENTS IN THE OXFORD ROBOTCAR [41] DATASET AND
TESTED ON THE CITY, TOWN AND FREEWAY SEQUENCES IN THE DDD17
[40] DATASET.

| Scenarios | Sequences | Recall@1 | Recall@5 | Recall@10 |
|--------------------|---------------|---------------|----------|-----------|
| city | rec1487839456 | 90.46% | 93.48% | 97.68% |
| | rec1487842276 | 91.73% | 95.54% | 98.14% |
| | rec1487844247 | 89.66% | 94.38% | 98.75% |
| town | rec1487846842 | 89.10% | 94.32% | 97.12% |
| | rec1487849151 | 88.49% | 93.58% | 97.42% |
| | rec1487849663 | 88.55% | 93.27% | 96.96% |
| freeway (day) | rec1487417411 | 76.80% | 90.43% | 93.65% |
| | rec1487594667 | 76.26% | 91.07% | 94.14% |
| | rec1487430438 | 78.06% | 91.10% | 94.27% |
| freeway (night) | rec1487350455 | 80.18% | 88.52% | 91.76% |
| | rec1487608147 | 79.50% | 92.22% | 95.57% |
| | rec1487609463 | 73.31% | 88.82% | 93.21% |

Event-VPR [10] dataset and our own recorded dataset in the CARLA [42] simulator to prove our advantages. **Basically, we show the comparison results of 2 evaluation indices, Recall@1 and PR curves. In addition, in order to better demonstrate the performance of the proposed Event-VPR, we added some intuitive experimental results to further verify the effectiveness of the proposed method. We draw retrieval success-rate maps and matching matrices for different sequences by referring to [45] and [10].**

Comparison on Brisbane-Event-VPR. In order to compare the performance of our event-VPR method with the Ensemble-Event-VPR [10] scheme proposed by Tobias et al., we compared the performance of VPR under different illumination conditions on their Brisbane-Event-VPR [10] dataset. The Brisbane-event-VPR dataset provides image frame and event stream sequences of the same trajectory under various illumination conditions in a day (such as sunrise, morning, daytime and sunset). In this experiment, we used query sets and database sets similar to those in the experiments of Ensemble-Event-VPR, to retrieve and match hundreds of GPS coordinate positions to verify the accuracy of the Event-VPR network model. It should be noted that the Ensemble-Event-VPR scheme does not require training, and directly uses the pre-trained model of NetVLAD as the feature extractor, while our models are obtained by cross-training on different sequences respectively. To make a fair comparison, we use different sequences as the query set and the database set both during training and testing. The experimental results of Recall@1 are shown in Table V, and the PR curves corresponding to our experiment are shown in Fig. 8. Since the Ensemble-Event-VPR scheme is to reconstruct event bins into intensity image frames, while our event-VPR method directly processes event bins, so it has better accuracy than the individual model and frame-based model in the paper of Tobias et al. [10], and even better than the ensemble model in their paper in some cases. However, the test performance of Event-VPR on very few sequences is slightly worse than their ensemble

TABLE V

COMPARISON OF OUR EVENT-VPR WITH ENSEMBLE-EVENT-VPR [10] PROPOSED BY TOBIAS ET AL. ON THEIR BRISBANE-EVENT-VPR [10] DATASET.

| Training (database / query) | Test (database / query) | Event-VPR (Ours) | Recall@1 Ensemble-Event-VPR [10] | | | |
|-------------------------------|-------------------------|---------------------|-------------------------------------|----------------|----------------|--------|
| | | | Ensemble | Individual (N) | Individual (t) | Frame |
| (daytime & morning) / sunrise | sunset1 / sunset2 | 81.20% | 86.72% | 77.38% | 75.26% | 83.53% |
| (sunset2 & morning) / sunrise | sunset1 / daytime | 41.77% | 37.76% | 30.39% | 28.05% | 35.07% |
| (sunset2 & daytime) / sunrise | sunset1 / morning | 63.55% | 55.08% | 38.59% | 39.47% | 50.70% |
| (sunset2 & daytime) / morning | sunset1 / sunrise | 63.69% | 53.48% | 38.85% | 38.50% | 41.36% |

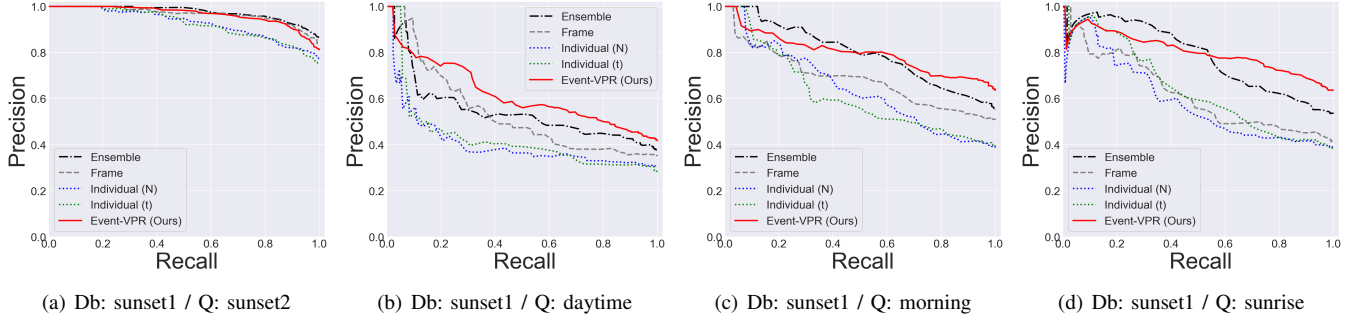


Fig. 8. PR Curves of Our Event-VPR and Ensemble-Event-VPR [10] on Different Illumination Sequences in Their Brisbane-Event-VPR [10] Dataset. From left to right: (a) Training: (daytime & morning) / sunrise, test: sunset1 / sunset2, (b) Training: (sunset2 & morning) / sunrise, test: sunset1 / daytime, (c) Training: (sunset2 & daytime) / sunrise, test: sunset1 / morning, (d) Training: (sunset2 & daytime) / morning, test: sunset1 / sunrise. (Note: Here 'Db' and 'Q' mean 'Database' and 'Query' respectively.)

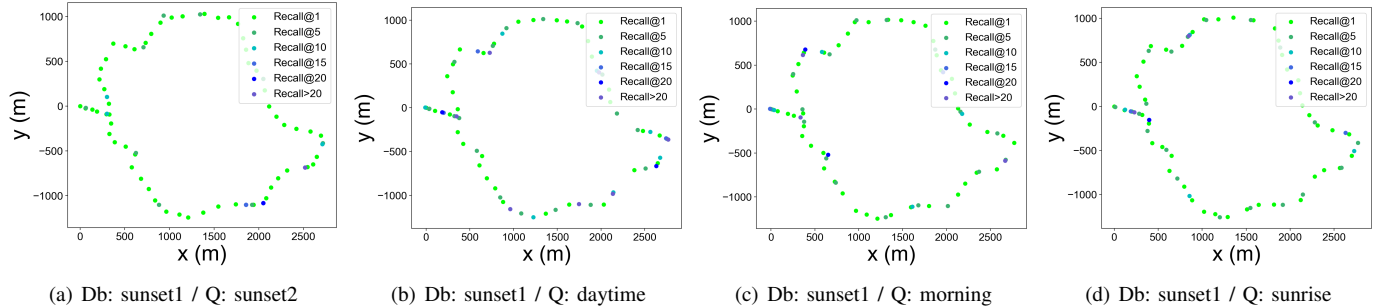


Fig. 9. Retrieval Success-Rate Maps between Databases and Queries of Our Event-VPR on Different Illumination Sequences in Their Brisbane-Event-VPR [10] Dataset. The order of subgraphs from left to right is consistent with Fig. 8. (Note: Here 'Db' and 'Q' mean 'Database' and 'Query' respectively.)

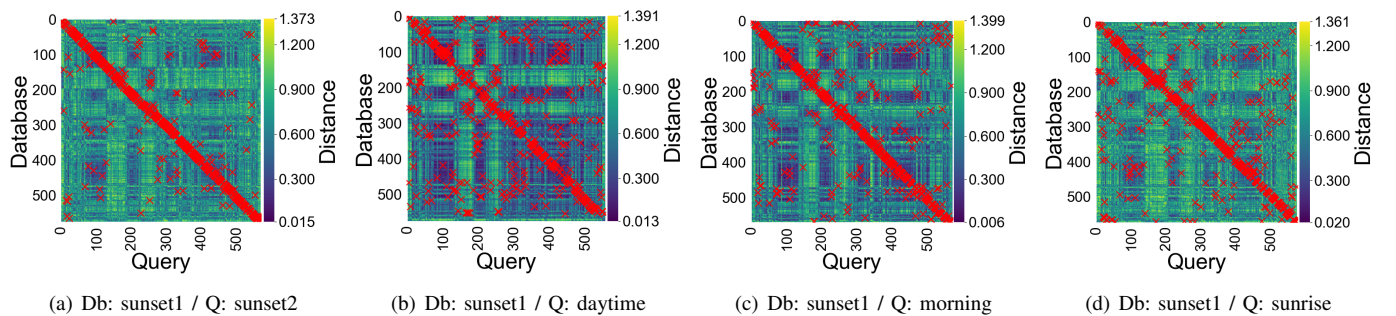


Fig. 10. Matching Matrices between Databases and Queries of Our Event-VPR on Different Illumination Sequences in Their Brisbane-Event-VPR [10] Dataset. The order of subgraphs from left to right is consistent with Fig. 8. The red cross represents the descriptor matching result of query samples and database samples. The more red cross is concentrated on the diagonal, the better the performance. The background shows the descriptor distance matrix between all query samples and all database samples. (Note: Here 'Db' and 'Q' mean 'Database' and 'Query' respectively.)

TABLE VI
COMPARISON OF OUR EVENT-VPR WITH ENSEMBLE-EVENT-VPR [10] PROPOSED BY TOBIAS ET AL. ON OUR CARLA [42] DATASET.

| Training (database / query) | Test (database / query) | Event-VPR (Ours) | Recall@1 Ensemble-Event-VPR [10] | | |
|-----------------------------|----------------------------|---------------------|-------------------------------------|----------------|----------------|
| | | | Ensemble | Individual (N) | Individual (t) |
| cloudynoon / clearsunset | clearnoon / clearsunrise | 66.10% | 54.16% | 43.15% | 43.18% |
| cloudynoon / clearsunrise | clearnoon / clearsunset | 89.90% | 77.89% | 63.63% | 62.61% |
| clearsunrise / clearsunset | clearnoon / cloudynoon | 81.75% | 87.47% | 77.83% | 76.28% |
| clearsunset / cloudynoon | clearsunrise / clearnoon | 61.62% | 51.26% | 41.62% | 39.56% |
| clearnoon / cloudynoon | clearsunrise / clearsunset | 58.45% | 70.38% | 63.66% | 61.95% |
| clearsunset / clearnoon | clearsunrise / cloudynoon | 45.61% | 40.18% | 31.69% | 27.96% |
| clearsunrise / cloudynoon | clearsunset / clearnoon | 87.80% | 71.04% | 60.08% | 60.05% |
| cloudynoon / clearnoon | clearsunset / clearsunrise | 51.26% | 58.76% | 58.08% | 51.93% |
| clearsunrise / clearnoon | clearsunset / cloudynoon | 62.23% | 43.04% | 30.08% | 29.88% |
| clearsunset / clearsunrise | cloudynoon / clearnoon | 79.73% | 76.19% | 67.31% | 63.38% |
| clearnoon / clearsunset | cloudynoon / clearsunrise | 52.34% | 37.44% | 29.15% | 27.67% |
| clearnoon / clearsunrise | cloudynoon / clearsunset | 66.42% | 43.25% | 36.31% | 35.18% |

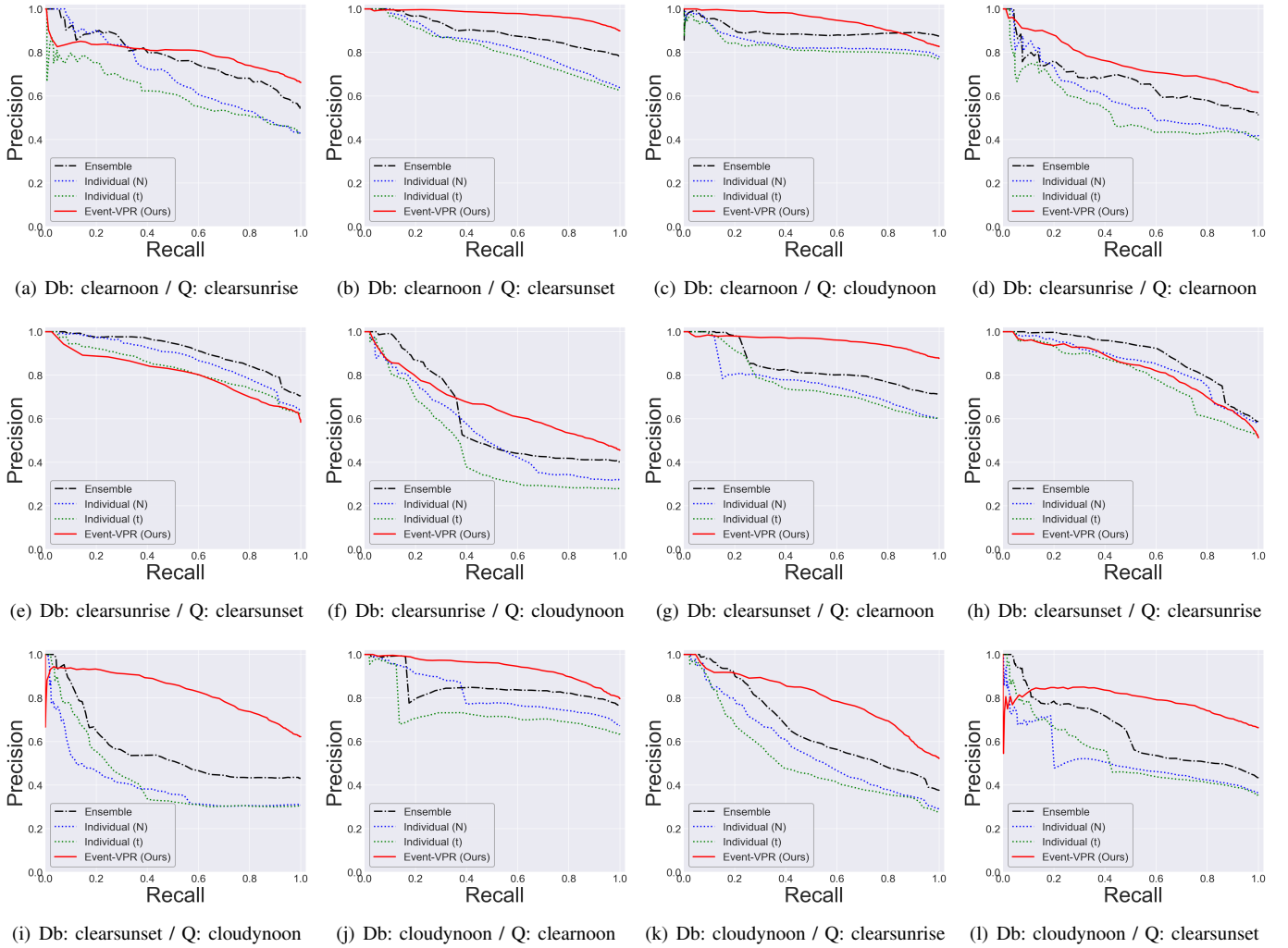


Fig. 11. PR Curves of Our Event-VPR and Ensemble-Event-VPR [10] on Different Weather and Illumination Sequences in Our CARLA [42] Dataset. From top left to bottom right: (a) Training: cloudynoon / clearsunset, test: clearnoon / clearsunrise, (b) Training: cloudynoon / clearsunrise, test: clearnoon / clearsunset, (c) Training: clearsunrise / clearsunset, test: clearnoon / cloudynoon, (d) Training: clearsunset / cloudynoon, test: clearsunrise / clearnoon, (e) Training: clearnoon / cloudynoon, test: clearsunrise / clearsunset, (f) Training: clearsunset / clearnoon, test: clearsunrise / cloudynoon, (g) Training: clearsunrise / cloudynoon, test: clearsunset / clearsunrise, (h) Training: cloudynoon / clearnoon, test: clearsunset / clearsunrise, (i) Training: clearsunrise / clearnoon, test: clearsunset / cloudynoon, (j) Training: clearsunset / clearsunrise, test: cloudynoon / clearnoon, (k) Training: clearnoon / clearsunset, test: cloudynoon / clearsunrise, (l) Training: clearnoon / clearsunrise, test: cloudynoon / clearsunset. (Note: Here 'Db' and 'Q' mean 'Database' and 'Query' respectively.)

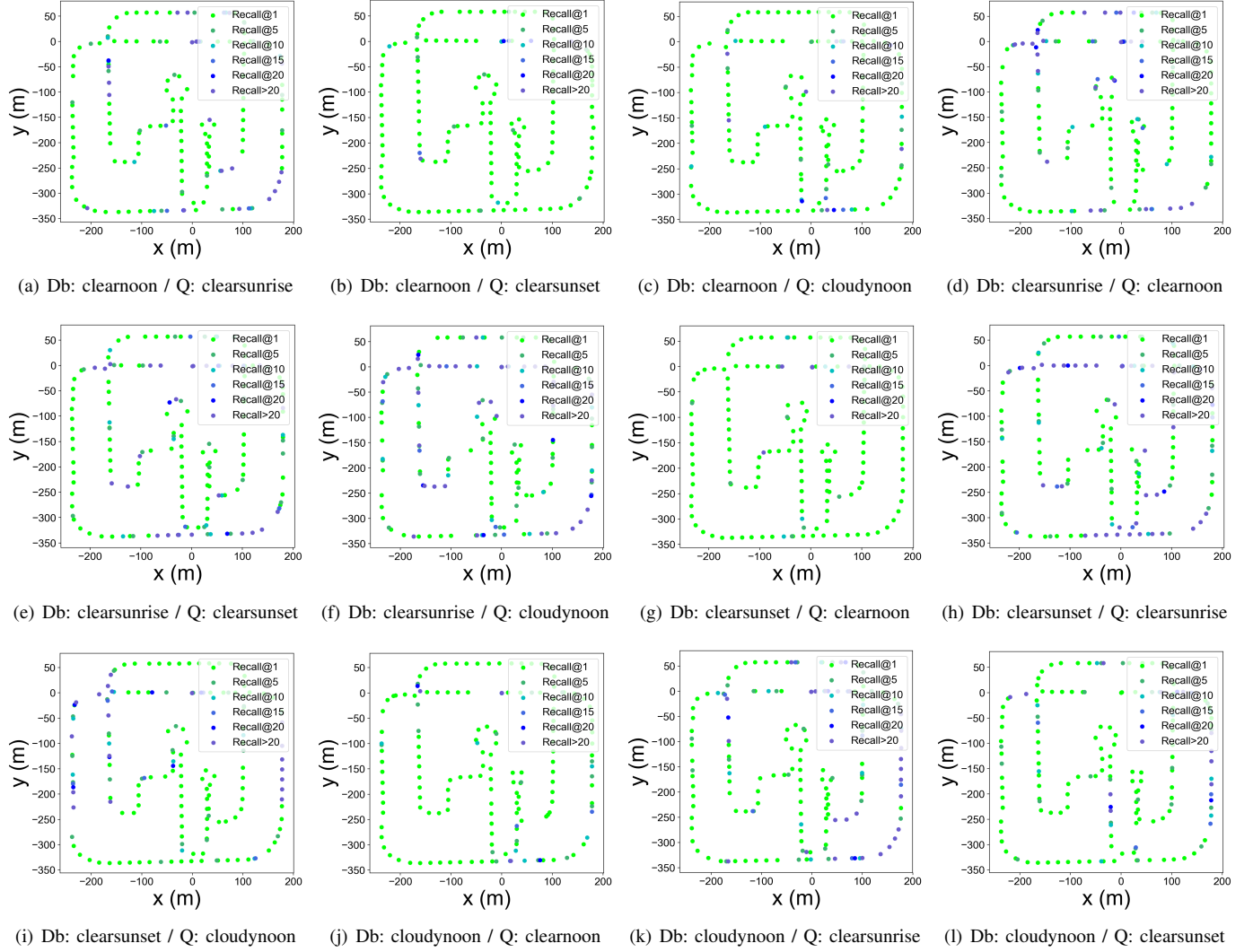


Fig. 12. Retrieval Success-Rate Maps between Databases and Queries of Our Event-VPR on Different Weather and Illumination Sequences in Our CARLA [42] Dataset. The order of subgraphs from top left to bottom right is consistent with Fig. 11. (Note: Here 'Db' and 'Q' mean 'Database' and 'Query' respectively.)

scheme. It shows that the scheme of window ensembles with different temporal lengths and number of events is indeed beneficial to the improvement of the overall performance, but it does not conflict with the end-to-end learnable advantage of our Event-VPR method. In summary, our Event-VPR can achieve similar or even better performance than their event-based ensemble scheme. As supplements, retrieval success-rate maps for different sequences are shown in Fig. 9, and matching matrices are shown in Fig. 10.

Comparison on CARLA. For further performance comparison, we compared the performance of VPR on several dataset sequences recorded by the CARLA [42] simulator. Our recorded CARLA dataset provides frames and event streams of driving sequences with the same route trajectory under multiple weather and multiple illumination conditions, including clear noon, clear sunrise, clear sunset, cloudy noon, etc. In addition, the driving route of our recorded dataset in CARLA [42] simulator is shown in the accompanying video. Whether it is training or testing, both the database and

the query come from different sequences. The experimental results of Recall@1 are shown in Table VI, and the PR curves corresponding to our experiment are shown in Fig. 11. It can be seen from the experiment results that whether Recall@1 or PR curve, the performance of our Event-VPR is better than Ensemble-Event-VPR in most cases. However, there are only a few poor results of the clearsunrise-vs-clearsunset scenario. For our analysis, since the sunrise and sunset scenes in the Brisbane-Event-VPR dataset are recorded under natural light, there is no significant difference between them. In the CARLA simulator, there is a huge difference in illumination between sunrise and sunset scenes, and there are obviously different shadows on the ground. In this case, the ensemble scheme using the reconstructed images is less affected. As supplements, retrieval success-rate maps for different sequences are shown in Fig. 12, and matching matrices are shown in Fig. 13.

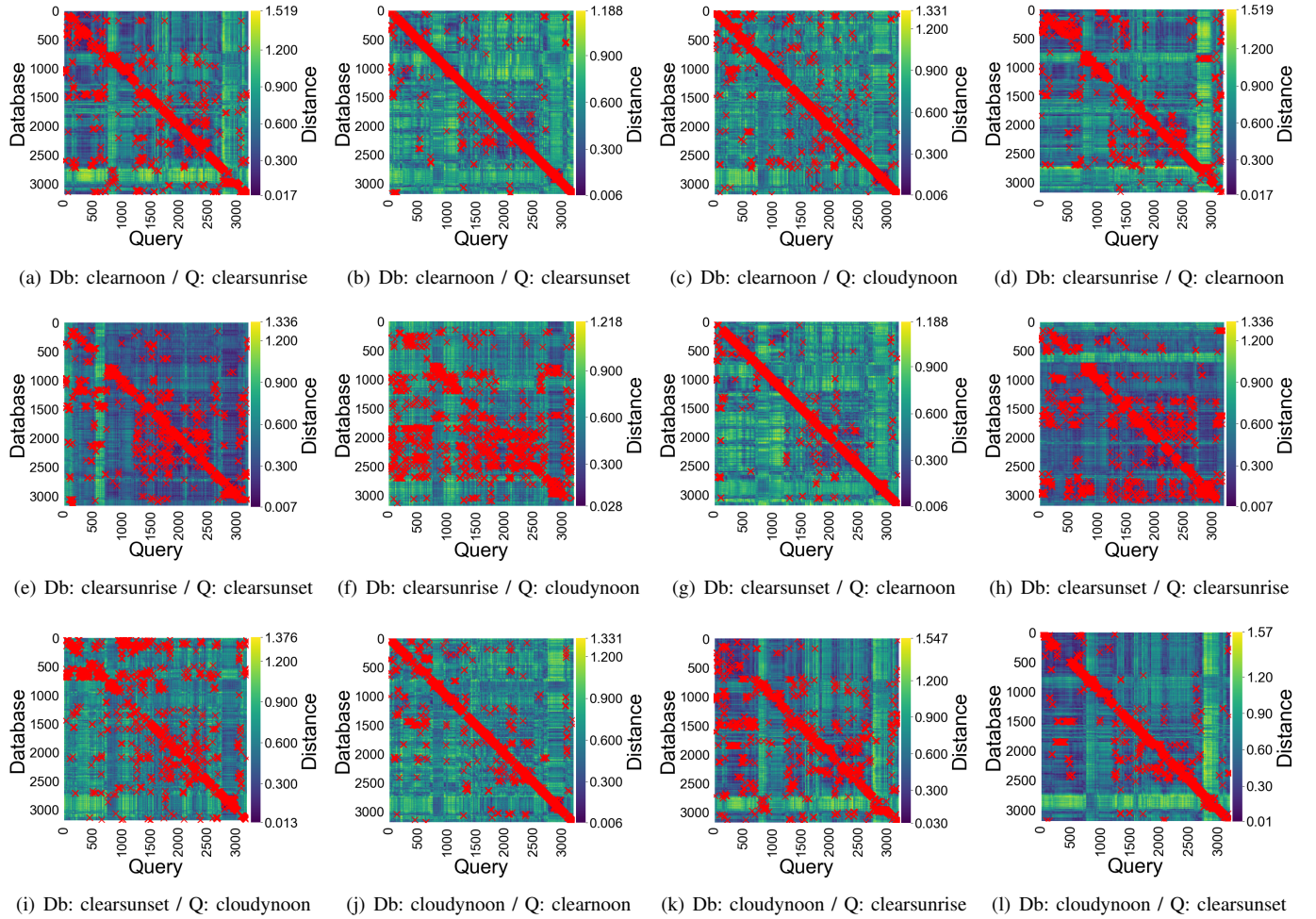


Fig. 13. Matching Matrices between Databases and Queries of Our Event-VPR on Different Weather and Illumination Sequences in Our CARLA [42] Dataset. The order of subgraphs from top left to bottom right is consistent with Fig. 11. The red cross represents the descriptor matching result of query samples and database samples. The more red cross is concentrated on the diagonal, the better the performance. The background shows the descriptor distance matrix between all query samples and all database samples. (Note: Here 'Db' and 'Q' mean 'Database' and 'Query' respectively.)

D. Network Ablation Study

Impact of Event Representations. In this experiment, we tried to explore the impact of different event representations on the performance of Event-VPR. In addition to the EST voxel grid described in our method, we have also tried several other event representations, including event frame (EF) [28], unipolar event voxel grid (EVG) [30], and 4-channel event count and last-timestamp image (4CH) [29]. In all cases we used ResNet34 as the feature extraction network. As shown in Fig. 14(a), the results show that EST voxel grid has significant advantages over these other representations in different datasets. Since EF discards timestamps and EVG discards event polarity, their experimental results are not as good as EST voxel grid. Due to 4CH discarding earlier timestamps, it performs poorer than EST voxel grid. EST voxel grid not only retains the local spatio-temporal neighborhood information of events, but also makes convolutional network extract more effective features through using a learnable kernel to suppress the noise events.

Impact of Network Structures. In this experiment, we explored the Event-VPR performance of different network

structures. In addition to the ResNet34 described in our method, we also trained VGG-16, AlexNet and other two deep residual networks with different network capacities. We used EST voxel grid as the event representation in all experiments. As shown in Fig. 14(b), experimental results demonstrate that different feature extraction networks have no significant impact on the network performance. We find that a larger network increases the computing resources without improving the accuracy when testing ResNet with different network capacities. We assume that as a larger network requires more parameters, then the network is more prone to overfitting. We finally choose ResNet34 as the feature extraction network for the proposed Event-VPR pipeline. However, as demonstrated, it can be replaced with other different feature extraction networks according to experimental needs and computing resources.

Impact of Loss Functions. In this experiment, different weakly supervised loss functions are compared for proposed Event-VPR, including triplet loss (TL), quadruplet loss (QL), lazy triplet loss (LTL) and lazy quadruplet loss (LQL). EST voxel grid was used as event representation and ResNet34

was used as feature extraction network in all experiments. As shown in Fig. 14(c), experimental results show that different weakly supervised loss functions only have a slight impact on model performance. Primitive triplet and quadruplet loss functions use the sum operator rather than the max operator in the simplified loss functions, so using primitive triplet and quadruplet loss functions tends to require longer training time. The simplified triplet and quadruplet loss functions can guarantee the performance and improve the efficiency of network training. Among them, the training results of quadruplet loss are slightly better than the corresponding triplet loss, and the trained model can obtain relatively better differentiability, so as to obtain more accurate retrieval results. Therefore, the above weakly supervised loss functions can be used alternately in training process, so as to obtain a high accuracy model in a shorter training time.

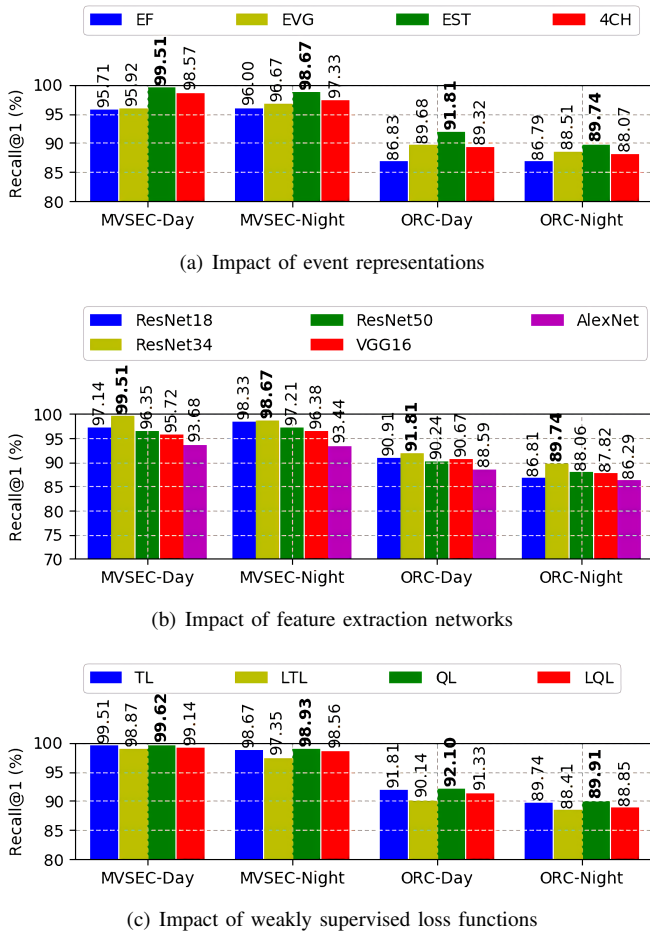


Fig. 14. Comparison of the Performance Impact about Event-VPR (MVSEC & Oxford RobotCar Datasets). From top to bottom: (a) Impact of event representations, (b) Impact of feature extraction networks, (c) Impact of weakly supervised loss functions.

V. CONCLUSIONS

We proposed an end-to-end weakly supervised network architecture and pipeline (Event-VPR) to solve the problem of large-scale place recognition using event cameras. The key idea is using the feature description aggregation layers based on VLAD for EST voxel grid representation generated

by event streams. The results showed the effectiveness and robustness of our Event-VPR in large-scale driving sequences under cross-weather, cross-season and illumination changing scenes. For the recognition performance, our Event-VPR is significantly better than the event-based ensemble VPR scheme. In addition, our ablation study showed that EST voxel grid representation has significant advantages over other representations in different datasets. It is important to note that event cameras have many advantages (such as low latency, low power consumption, high dynamic range) over conventional frame-based cameras. However, there are still some deficiencies compared to frame-based VPR methods due to the low spatial resolution of event cameras. In future work, we will try to combine standard cameras and event cameras to implement a hybrid network architecture for visual place recognition, or realize a visual place recognition architecture based on deep spiking convolutional network under the premise of ensuring algorithm performance, so that we can deploy it on autonomous vehicles or mini-UAVs to solve visual place recognition and visual loop detection of mobile robots.

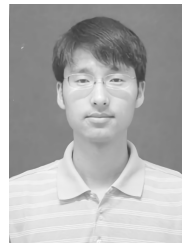
REFERENCES

- [1] S. Lowry, N. Sunderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual Place Recognition: A Survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016.
- [2] Z. Zeng, J. Zhang, X. Wang, Y. Chen, and C. Zhu, "Place recognition: An overview of vision perspective," *Applied Sciences (Switzerland)*, vol. 8, no. 11, p. 2257, 2018.
- [3] A. Angeli, D. Filliat, S. Doncieux, and J. A. Meyer, "Fast and incremental method for loop-closure detection using bags of visual words," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1027–1037, 2008.
- [4] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.
- [5] A. Oertel, T. Cieslewski, and D. Scaramuzza, "Augmenting visual place recognition with structural cues," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5534–5541, 2020.
- [6] T. Delbrück, B. Linares-Barranco, E. Culurciello, and C. Posch, "Activity-driven, event-based vision sensors," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. IEEE, 2010, pp. 2426–2429.
- [7] D.-i. D. Cho and T.-j. Lee, "A review of bioinspired vision sensors and their applications," *Sensors and Materials*, vol. 27, no. 6, pp. 447–463, 2015.
- [8] G. Gallego, T. Delbruck, G. M. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based Vision: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, apr 2020. [Online]. Available: <http://dx.doi.org/10.1109/TPAMI.2020.3008413>
- [9] G. Chen, H. Cao, J. Conradt, H. Tang, F. Rohrbein, and A. Knoll, "Event-based neuromorphic vision for autonomous driving: a paradigm shift for bio-inspired visual sensing and perception," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 34–49, 2020.
- [10] 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.
- [11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [12] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2011, pp. 2564–2571.
- [13] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 3304–3311.
- [14] R. Arandjelovic and A. Zisserman, "All about VLAD," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1578–1585.

- [15] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 Place Recognition by View Synthesis," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, 2018, pp. 257–271.
- [16] Z. Chen, O. Lam, A. Jacobson, and M. Milford, "Convolutional neural network-based place recognition," *Australasian Conference on Robotics and Automation*, ACRA, vol. 02-04-Dece, 2014.
- [17] M. Lopez-Antequera, R. Gomez-Ojeda, N. Petkov, and J. Gonzalez-Jimenez, "Appearance-invariant place recognition by discriminatively training a convolutional neural network," *Pattern Recognition Letters*, vol. 92, pp. 89–95, 2017.
- [18] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, 2018, pp. 1437–1451.
- [19] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *Proceedings - IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1643–1649.
- [20] A. Torii, J. Sivic, M. Okutomi, and T. Pajdla, "Visual Place Recognition with Repetitive Structures," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 11, 2015, pp. 2346–2359.
- [21] Y. Ye, T. Cieslewski, A. Loquercio, and D. Scaramuzza, "Place recognition in semi-dense maps: Geometric and learning-based approaches," in *British Machine Vision Conference 2017, BMVC 2017*. University of Zurich, 2017.
- [22] L. G. Camara and L. Preučil, "Spatio-Semantic ConvNet-Based Visual Place Recognition," in *arXiv*. IEEE, 2019, pp. 1–8.
- [23] Z. Hong, Y. Petillot, D. Lane, Y. Miao, and S. Wang, "TextPlace: Visual place recognition and topological localization through reading scene texts," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-October, 2019, pp. 2861–2870.
- [24] A. Benbihi, S. Arravechia, M. Geist, and C. Pradalier, "Image-Based Place Recognition on Bucolic Environment Across Seasons from Semantic Edge Description," in *Proceedings - IEEE International Conference on Robotics and Automation*. IEEE, 2020, pp. 3032–3038.
- [25] M. Milford, H. Kim, M. Mangan, S. Leutenegger, T. Stone, B. Webb, and A. Davison, "Place Recognition with Event-based Cameras and a Neural Implementation of SeqSLAM," *arXiv preprint arXiv:1505.04548*, 2015. [Online]. Available: <http://arxiv.org/abs/1505.04548>
- [26] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "Events-to-video: Bringing modern computer vision to event cameras," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, 2019, pp. 3852–3861.
- [27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, 2016, pp. 770–778.
- [28] 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 *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, dec 2018, pp. 3867–3876.
- [29] A. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras," *arXiv preprint arXiv:1802.06898*, 2018. [Online]. Available: <https://arxiv.org/abs/1802.06898>
- [30] D. Gehrig, A. Loquercio, K. Derpanis, and D. Scaramuzza, "End-to-end learning of representations for asynchronous event-based data," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-Octob, 2019, pp. 5632–5642.
- [31] C. Scheerlinck, H. Rebecq, D. Gehrig, N. Barnes, R. Mahony, and D. Scaramuzza, "Fast image reconstruction with an event camera," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 156–163.
- [32] 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.
- [33] B. Kulis, "Metric learning: A survey," *Foundations and Trends in Machine Learning*, vol. 5, no. 4, pp. 287–364, 2013.
- [34] L. Yang and R. Jin, "Distance metric learning: A comprehensive survey," *Michigan State University*, vol. 2, no. 2, p. 4, 2006.
- [35] M. Kaya and H. Ş. Bilge, "Deep metric learning: A survey," *Symmetry*, vol. 11, no. 9, p. 1066, 2019.
- [36] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.
- [37] W. Chen, X. Chen, J. Zhang, and K. Huang, "Beyond triplet loss: a deep quadruplet network for person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 403–412.
- [38] F. Gu, W. Sng, X. Hu, and F. Yu, "Eventdrop: data augmentation for event-based learning," *arXiv preprint arXiv:2106.05836*, 2021. [Online]. Available: <http://arxiv.org/abs/2106.05836>
- [39] A. Z. Zhu, D. Thakur, T. Özarslan, B. Pfrommer, V. Kumar, and K. Daniilidis, "The Multivehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2032–2039, 2018.
- [40] J. Binas, D. Neil, S.-C. Liu, and T. Delbruck, "DDD17: End-To-End DAVIS Driving Dataset," *arXiv preprint arXiv:1711.01458*, 2017. [Online]. Available: <http://arxiv.org/abs/1711.01458>
- [41] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The Oxford RobotCar dataset," *International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.
- [42] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, nov 2017, pp. 1–16.
- [43] Y. Hu, S.-C. Liu, and T. Delbruck, "V2e: From video frames to realistic dvs events," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1312–1321.
- [44] T. Delbruck, Y. Hu, and Z. He, "V2E: From video frames to realistic DVS event camera streams," *arXiv preprint arXiv:2006.07722*, 2020. [Online]. Available: <http://arxiv.org/abs/2006.07722>
- [45] M. A. Uy and G. H. Lee, "Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4470–4479.

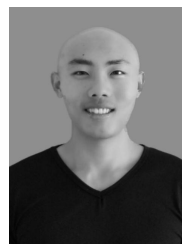


Delei Kong received the B.S. degree in automation from Henan Polytechnic University, China, in 2018, and the M.S. degree in control engineering from Northeastern University, China, in 2021. Since 2021, he has been a research assistant with Northeastern University, China. His research interests include event-based vision, robot visual navigation, and neuromorphic computing.



Zheng Fang (Member, IEEE) received the B.S. degree in automation and the Ph.D. degree in pattern recognition and intelligent systems from Northeastern University, China, in 2002 and 2006, respectively. He was a Postdoctoral Research Fellow of Carnegie Mellon University from 2013 to 2015. He is currently a Professor with the Faculty of Robot Science and Engineering, Northeastern University. His research interests include visual/laser SLAM, and perception and autonomous navigation of various mobile robots. He has published over 60 papers

in well-known journals or conferences in robotics and computer vision, including JFR, TPAMI, ICRA, IROS, BMVC, etc.



Kuanxu Hou received the B.S. degree in robot engineering from Northeastern University, China, in 2020. He is currently pursuing the M.S. degree in robot science and engineering with Northeastern University, China. His research interests include event-based vision, visual place recognition, and deep learning.



Haojia Li received the B.S. degree in robot engineering from Northeastern University, China, in 2021. He is currently pursuing the MPhil. degree with the Hong Kong University of Science and Technology, Hong Kong. His research interests include multi-sensor fusion, event-based vision, and construction robot.



Junjie Jiang received the B.S. degree in automation from Northeastern University at Qinhuangdao, China, in 2020. He is currently pursuing the M.S. degree in robot science and engineering with Northeastern University, China. His research interests include spiking neural network, robot visual navigation, and reinforcement learning.



Sonya Coleman (Member, IEEE) received the B.Sc. degree (Hons.) in mathematics, statistics, and computing, and the Ph.D. degree in mathematics from Ulster University, Londonderry, U.K., in 1999 and 2003, respectively. She is currently a Professor with the School of Computing and Intelligent System, Ulster University, and also a Cognitive Robotics Team Leader with the Intelligent Systems Research Centre. Her research has been supported by funding from various sources such as EPSRC, The Nuffield Foundation, The Leverhulme Trust, and the EU.

She was involved in the EU FP7 funded projects RUBICON, VISUALISE, and SLANDAIL. She has authored or coauthored over 150 publications in robotics, image processing, and computational neuroscience. Dr. Coleman was awarded the Distinguished Research Fellowship by Ulster University in recognition of her contribution research in 2009.



Dermot Kerr received the B.S. degree in computing science and the Ph.D. degree in computing and engineering from Ulster University, Londonderry, U.K., in 2005 and 2008, respectively. He is currently a Lecturer with the School of Computing, Engineering and Intelligent System, Ulster University. He was involved in the EU FP7 funded projects VISUALISE and SLANDAIL. His current research interests include computational intelligence, biologically inspired image processing, mathematical image processing, omnidirectional vision, and robotics. Dr.

Kerr is an Officer and a member of the Irish Pattern Recognition and Classification Society.