Visual Place Recognition under Substantial Appearance Changes using Event-based Data

BMVC 2020 Submission # 404

Abstract

An event camera is a biologically inspired vision sensor which independently and asynchronously records changes of brightness as they occur in each pixel. With its high measurement rate, low latency, and high dynamic range properties, the sensor has a great potential to overcome some limitations of conventional vision sensors. We focus on its applicability in visual place recognition, which is one of the fundamental problems in computer vision and robot applications. In this work, we propose a framework which uses a convolutional neural network (CNN) architecture to train event-based images for place recognition tasks in challenging weather and illumination conditions. We also exploit data association for better image sequence matching. While showing insignificant effect when applied to conventional images, the matching algorithm contributes to noticeable performance improvement using event-based images. We evaluate the performance of the proposed method using the synthetic and real-world dataset compared with the state-of-the-art frameworks.

1 Introduction

Event cameras, such as DVS240 [10], are visual sensors that respond to the local brightness changes. Unlike conventional cameras, which synchronously update all pixel values at a constant frame rate, event cameras record pixel intensity changes asynchronously. The result can be represented by the set of events, and each event includes time, location, and polarity information of the pixel brightness changes.

An event camera has its advantages over conventional cameras. Due to its high temporal resolution, an event camera records much clearer image data in dynamic environments while traditional cameras may suffer from motion blur. The sensor's high dynamic range (HDR) captures scene features more robustly under challenging illumination conditions. Many researchers are of interest to apply this newly-developed sensor to enhance the performance of existing computer vision algorithms.

Robust visual place recognition under different illumination and weather conditions is an active research area for mobile robots and autonomous driving vehicles. In recent years, various approaches have been presented for effective visual place recognition under different circumstances [5, 15, 16, 22, 23].

In this work, we propose a convolutional neural network (CNN) based framework using event-based data for visual place recognition under substantial appearance changes due to illumination and weather conditions. The proposed framework is built on top of a CNN-based

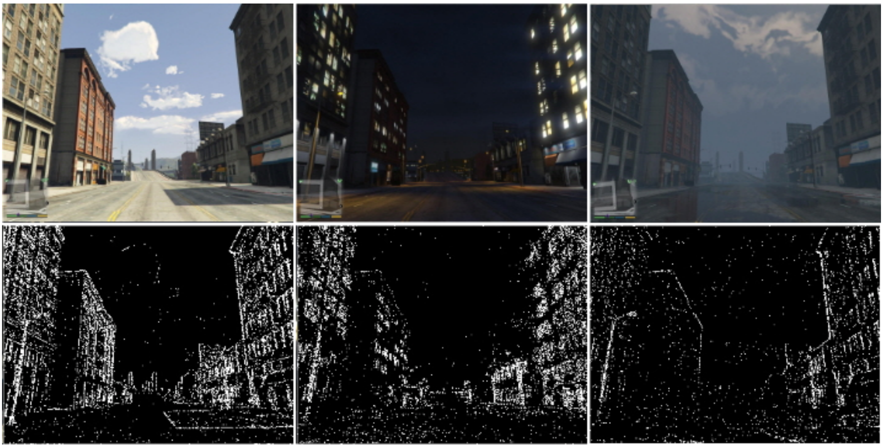

Figure 1: Examples of conventional images (top) and the event-based images (bottom) of virtual street-views under different illumination and weather conditions in a computer game (GTA5 [1]). While conventional images show substantial appearance changes, event-based images capture edges and corners robustly.

searching method [7] using a descriptor image obtained by a Light Detection and Ranging (LiDAR) sensor. Instead of using LiDAR data, we trained event-based images to the network. Since there is no proper event-based dataset that visits the same place multiple times under different illumination or weather conditions, we created event-based data for place recognition task using an open event camera simulator, ESIM [19]. To make the most of the event camera's high frame rate characteristics, we created synthetic event-based dataset from virtual street-views of a popular computer game Grand Theft Auto 5 (GTA5) [1] under different illumination and weather conditions. We created and tested another event-based dataset simulated from the Oxford Robot Car [12] dataset, which is a well-known real-world dataset for place recognition.

To the best of our knowledge, this is the first approach to apply event-based data to train a CNN for visual place recognition. Since the features such as edges and corners of the scene appears more robustly in the event-based images as shown in Figure 1, our network trained by event-based data shows superior place recognition performance under substantial appearance changes than the same network trained by images taken by conventional cameras. We added a data association method that efficiently localize the query sequence in the database sequence to improve the performance.

2 Related work

In the past decade, various approaches have been introduced to tackle the problem of visual place recognition task. Galvez-Lopez *et al.* [4] used Bag of Words (BoW) method for efficient loop-detection in place recognition. Paul and Newman [16] introduced FAB-MAP which used a recursive Bayesian model for localization in large-scale maps. Cummins *et al.* [2] improved the robustness against perceptual aliasing and newly visited place detection by demonstrating the system on extreme datasets with FAB-MAP 2.0. Milford and Wyeth [15] proposed SeqSLAM, which uses a local navigation sequence and matching strat-

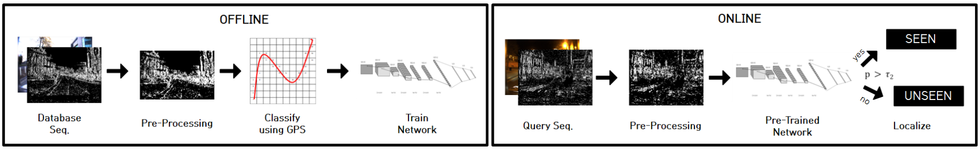

Figure 2: Pipeline of the proposed method. The network is trained offline and tested using event-based images and color images for evaluation.

egy of neighboring local images instead of global matching. Pepperell *et al.* [17] improved matching performance under various conditions by using image conversion with additional odometry information for matching. Vysotska and Stachniss [22] used a graph structure to represent comparisons and transitions between the query and database images in multi-sequence maps under various illumination conditions [23]. Hansen and Browning [6] proposed a Hidden Markov Model (HMM) framework to solve the limitation of the SeqSLAM in nonlinear trajectory.

Recently, many studies are devoting for the robust matching in dynamic environments using other sensors. Maddern and Vidas [11] used visible spectrum images and long-wave infrared images for place recognition through a day-night-cycle. Stone *et al.* [20] used sky-line images obtained by an ultraviolet sensor for localization using SeqSLAM. Kim *et al.* [7] presented descriptor images converted from LiDAR data.

Kim *et al.* [8] proposed a real-time 3D reconstruction and 6-DoF tracking method using a event camera. While this method required a GPU to perform 3d reconstruction, Rebecq *et al.* [18] improved the method to work using only a CPU. Milford *et al.* [14] introduced a place recognition on event data using SeqSLAM, which matched event-based images and detected loops at various velocity conditions at indoors.

3 Methodology

We convert event data to event-based images so that it can be trained and tested in a convolutional neural network. We divide the entire map into grid-based regions and assign each cell as a class for network training. Then we exploit data association for better image sequence matching. The pipeline of the proposed framework is shown in Figure 2.

3.1 Event data

Event cameras triggers events asynchronously by pixel when the log intensity of the pixel change exceeds the threshold as Equation 1.

$$|L(\mathbf{x},t) - L(\mathbf{x},t-\Delta t)| \geq C \tag{1}$$

Where L is the log intensity, $\mathbf{x} = [x,y]^T$ is the pixel location, and C is the contrast threshold. The set of N events can be represented as Equation 2.

$$E = \{e_k\}_{k=1}^{N} = \{(t_k, \mathbf{x}_k, sign(\frac{dI(\mathbf{x}_k)}{dt}))\}_{k=1}^{N} \tag{2}$$

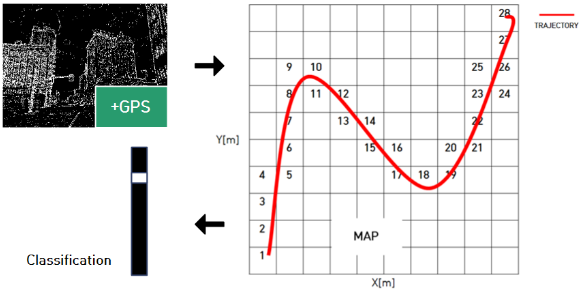

Figure 3: Illustration of grid-based region classification. Each cell is represented as a class.

3.2 Events-to-image conversion

Since the event data is obtained asynchronously, unlike the conventional camera that synchronizes the image frame by frame, many studies have been conducted to convert the event data into a type of image format for processing by applying the existing image processing algorithm. Maqueda *et al.* [13] used a constant temporal window and count the events appear to predict the vehicle's steering angle. Zhu *et al.* [24] preserved the temporal information when the events spike using a grid that includes the polarity and spiking time. Also they converted the events in a spatiotemporal voxel-grid to a image [25]. Our conversion method $E \rightarrow I$ as shown in Equation 3 and 4 is similar with Zhu's method in that it removes polarity information and creates an image with events within a temporal period. Since it was determined that polarity information was not very important for place recognition, an image with removed polarity information was used.

$$I_n(x_i, y_j) = min(1, \sum_{e_k} \delta(x_i - x_k)\delta(y_j - y_k)) \tag{3}$$

$$e_k \in E^{n\Delta t}_{(n-1)\Delta t} = \{e_x | (n-1)\Delta t < t_x \leq n\Delta t\} \tag{4}$$

Where $\delta(x)$ is the dirac-delta function. Additionally, We used 3×3 sized Gaussian-Blur to the event images to make the distribution of events appearing at the edge of the structure constant. Also we removed the bottom quarter since it was judged that there was no special information for place recognition and minimize the affect of pedestrians and vehicles. As a result, the network was trained and validated using the generated images and the specific parameters are described at Section 4.

3.3 Network training

Our network was created by imitating existing general networks like LeNet [9] as shown in Table 1. When training with an event camera, the training dataset images have one channel and it was resized to $R_x \times R_y$. Similarly, when using a color image, the resolution was equally $R_x \times R_y$ and the number of channels C was 3 of red, green, and blue. In Table 1, MP is Max Pooling and the pooling area size is 2. ReLu is the activation function, C is the number of channels, and N is the total number of classes. The kernel size was 5, and was padded with size 2 for all Conv layers. Also the Fully Connected layers were all linear. The learning rate was set as 0.001, the training batch size was 64, and was trained for 1000 iterations.

Table 1: The overall design of the network.

Layer	Value
Input Image	$R_x \times R_y \times C$
MP(ReLu(Conv1))	input_channel=C, output_channel=6
Batch_Norm	channel=6, eps=1e-05, momentum=0.1
MP(ReLu(Conv2))	input_channel=6, output_channel=16
Batch_Norm	channel=16, eps=1e-05, momentum=0.1
MP(ReLu(Conv3))	input_channel=16, output_channel=32
ReLu(FullyConnected1)	in_features=7040, out_features=2×N
ReLu(FullyConnected2)	in_features=2×N, out_features=1.5×N
ReLu(FullyConnected3)	in_features=1.5×N, out_features=N

3.4 Localization and labelling

Let's denote train dataset T and the validation dataset V as Equation 5

$$T = \{(I_i^T, \mathbf{G}_i^T)\}_{i=0}^{N_T}, V = \{(I_V^i, \mathbf{G}_i^V)\}_{i=0}^{N_V} \tag{5}$$

We labeled each image I using GPS information $\mathbf{G} = [G_x, G_y]^T$ at the location where the image was taken. First, the latitude and longitude data was transformed into $[x, y]^T$, the Universal Transverse Mercator Coordinate (UTM) information appearing on flat ground. Subsequently, the ground was divided by the resolution of the interval ρ, and a class was sequentially assigned to each square grid through which the camera passes as Figure 3. As a result, the size N of the output vector of the network becomes the total number of grids that the camera of the train dataset passes. We hypothesized that the grid adjacent to the ground truth class in 8 directions also classified correctly. And we adopted the same degree of acceptance for the comparing traditional methods for clarity.

3.5 Matching algorithm

A matching algorithm using a output vector of the network was needed to efficiently match the query images to the database images. Since the output vector contains information about the probability belonging to each class, we used this information to determine whether the query image came from the previously experienced or the first time seen. We made a simple matching algorithm by referring to the algorithm proposed by Vysotska and Stachniss [22] and SeqSLAM by Milford *et al*. for maintain correlation between successive frames. using the output vector p, we propose the following algorithm shown as Algorithm 1. Two parameters, the global threshold τ_1 and the local threshold τ_2, are used for discriminating seen and unseen. In the case of τ_2, the best performance was shown at about 0.01. Therefore, we drew the precision-recall curve while sweeping τ_1 as shown in Section 5.2.

4 Experiments

The experiments were designed to show the followings. First, the higher performance and robustness in place recognition when using the event camera image in different environment conditions than when using an ordinary color camera. Second, the precision and recall

Algorithm 1: Matching algorithm

```
Result: Match_vector
Match_vector[0] = index(max(p[0])), i = 1;
while $i \leq N_q$ do
    previous = Match_vector[i-1];
    local_max = index(max(p[i][previous-K:previous+K]));
    if p[i][local_max] > $\tau_2$ then
        Match_vector[i] = index(local_max);
        i += 1;
    else
        global_max = index(max(p[i]));
        if p[i][global_max] > $\tau_1$ then
            Match_vector[i] = index(global_max);
            i += 1;
        else
            Match_vector[i] = -1;
            i+= 1;
        end
    end
end
```

performance improvement by applying the proposed matching algorithm. We conducted experiments using various datasets, event camera simulator [19], and real event cameras. We used an NVIDIA GTX 1080Ti for all experiments and network implementations.

4.1 Datasets

We conducted experiments using the virtual city dataset extracted from the Grand Theft Auto 5 (GTA5) [1] game, a popular role playing game in a virtual city. the Oxford Robot Car dataset [12], and our own campus dataset that taken directly by event camera. For the GTA dataset, we used the open source software called G2D [3], the image frame and the camera's six degrees of freedom for each frame were obtained and the event image was also obtained using ESIM [19]. By using ESIM, we wanted to show that even if we can't afford to use an event camera, we can increase the matching performance simply by converting it to an event image. The characteristic of this dataset is that the changes in the surrounding environment are very large, and the vehicle's speed and location over time match exactly. We traversed two paths shown as Figure 4 and overlaps were present. Also we used the Oxford Robot Car dataset [12] and our campus dataset which is obtained by a DVXplore Lite event camera [6]. In the Oxford dataset and our campus dataset, We select the database as a clean weather in daytime, and query as night time with large illuminance differences. And the viewpoint of the camera was significantly different also. All the datasets were resized to $R_x = 160, R_y = 90$ for training and we set $\Delta t = 1/60$, $1/16$, and $1/30$ second for each dataset respectively.

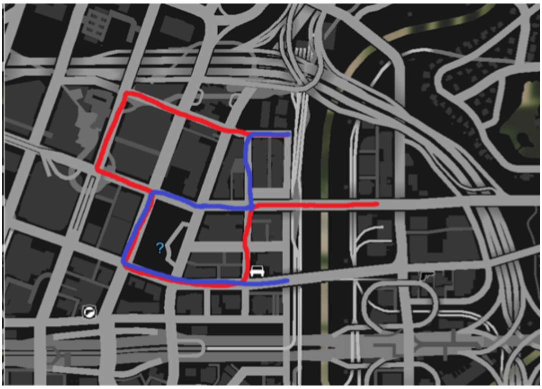

Figure 4: Two overlapping paths in the virtual city of GTA5 used in the experiments. The images in the first path in red are used for training and the images in the second path in blue are used for the evaluation. Parts of the paths are overlapped.

4.2 Comparison with color camera images

We compared the matching performance between the event camera image and the color camera image using the event-image conversion method and the CNN model presented in Section 3. The network was trained using the clear day images of the first trajectory in the GTA dataset, and verification process was performed using the same trajectory images taken on night and rainy days, for each event and color image datasets. In First, matching was performed by assuming that the element having the highest value (the most probable) among the N-dimensional vector, is its class which the corresponding image locates. In other words, the correlation between two consecutive frames was not considered (although it actually exists). Hereinafter, this matching method will be referred to as Global-Max. In addition, we experimented in the same way using the Sequence-matching method presented in 3, and showed the advantage of our Sequence-matching method. The parameters were as $N = 104$, $\rho = 20m$ and also $\tau_1 = 0.99, \tau_2 = 0.01$ were fixed for measure the accuracy. In addition, we performed matching using the first trajectory and the second trajectory where the paths do not match completely. The network was trained with images of the first GTA dataset on a clear day, and matching validation was performed with images of the second GTA dataset on a clear, night, and rainy conditions and the parameters were as the same. Finally, we verified the accuracy with OpenSeqSLAM [21] and Graph-Based [22], for compare with our method. All the parameters of the algorithms was set to default except the expansion rate α of the Graph-Based method. Since the accuracy extremly low when α was 0.7, the experiment was conducted with α set at 0.1.

4.3 Comparison with traditional methods

We compared the matching accuracy through drawing a precision-recall curve with SeqSLAM, Graph-Based method which are existing place recognition methods using the color camera sequence. In this case, we applied the matching algorithm described in Section 3 to facilitate comparison with existing methods for each datasets.

Table 2: The accuracy while using the event images compared to color images. The superscript 1 and 2 of the dataset means the trajectory type of the GTA dataset.

Global-Max				Sequence Matching			
Database	Query	Event	Color	Database	Query	Event	Color
clean^{1}	night^{1}	85.80	3.39	clean^{1}	night^{1}	93.20	3.79
clean^{1}	rain^{1}	75.42	31.36	clean^{1}	rain^{1}	81.34	38.22
clean^{1}	clean^{2}	55.58	34.07	clean^{1}	clean^{2}	91.28	55.35
clean^{1}	night^{2}	46.04	0.58	clean^{1}	night^{2}	77.44	21.27
clean^{1}	rain^{2}	33.26	19.77	clean^{1}	rain^{2}	56.74	46.16
SeqSLAM				Graph-Based			
Database	Query	Event	Color	Database	Query	Event	Color
clean^{1}	night^{1}	98.14	49.13	clean^{1}	night^{1}	85.49	24.77
clean^{1}	rain^{1}	98.14	93.21	clean^{1}	rain^{1}	88.55	34.62
clean^{1}	clean^{2}	32.33	76.98	clean^{1}	clean^{2}	10.93	10.93
clean^{1}	night^{2}	40.70	9.30	clean^{1}	night^{2}	25.35	6.05
clean^{1}	rain^{2}	36.74	50.47	clean^{1}	rain^{2}	37.91	6.28

5 Results

In this section, the effectiveness and accuracy of the event camera compared to the existing color camera was verified. When the same algorithm was used, the performance of the event camera and the color camera were significantly different, showing the potential of the event camera in robust location recognition. Also, compared with other state-of-the-art matching algorithms, we draw a precision-recall curve to show that our algorithm is sufficiently usable in dynamic environments and matching situations using CNN.

5.1 Performance comparison with color images

We evaluated the performance of place recognition when the event image and the color image were used respectively. Table 2 shows the results that the event image's ability to describe the features of the place was outperform than the color image. In particular, when matching using images taken at night, the matching accuracy of the event image was significantly higher than that of the color image. This is because there is a large difference in brightness between the color image taken during the day and the color image taken at night, but the event image seems to be because the difference between the image taken during the day and the night was not large because it detects a change in brightness rather than the absolute brightness. In addition, we can verify that SeqSLAM and Graph-Based matching algorithm suffer when the trajectory of database and query does not overlap and take a different path. The precision, recall and the accuracy measurements were same as [23].

In almost all cases, we can see that using an event image performs better than using a color image. SeqSLAM has a particularly good matching performance in matching between the first paths, which is a perfect linear match. However, in most situations, it does not move perfectly the same way as before. We can see that our method is the best in the more general situation such as matching the first path and the second path.

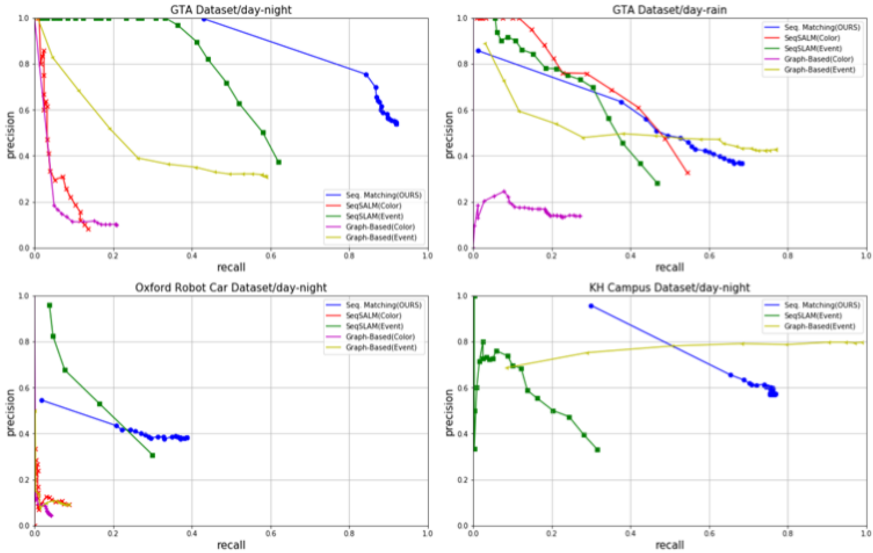

Figure 5: The precision-recall curves for four different datasets.

5.2 Precision-recall curve

We compared the matching performance with existing methods using various datasets. The existing methods to be compared with our proposed method Local Sequence Match, were SeqSLAM and Graph-Based method. Since both methods use a color image sequence for mathcing, they were performed using a color image sequence first. In addition, the results were obtained by operating the existing methods using an event image for comparison. Our method, SeqSLAM and Graph-Based method were τ_1, uniqueness factor μ and the non match cost m as sweep parameters respectively. When using a dataset taken at night as a query image, we can see that the performance of our proposed method outperforms than the existing method. In Figure 5, we can see that our method outperforms or similar with other methods.

6 Conclusion

We proposed a method for recognizing a place using an event camera. Convolutional Neural Network was applied to the image made from event data, and an algorithm was proposed to perform localization more efficiently by using the probability vector output therefrom. At first, we showed that the event images are more robust to changes in the surrounding environment due to weather and time changes than the conventional color image. Also, we draw the precision-recall curve and compare the performance difference with other existing methods, and show that the matching performance in the extreme dynamic environment such as day-night condition is better than the existing methods.

In the future work, we will conduct research on how the network may not be affected by changes in the viewpoint of the camera, and make more efficient matching using the characteristics of the event camera.

References

[1] Grand theft auto 5. https://www.rockstargames.com/V/. Accessed: 2020-04-01.

[2] M. Cummins and P. Newman. Appearance-only slam at large scale with fab-map 2.0. *I. J. Robotic Res.*, 30:1100–1123, 08 2011. doi: 10.1177/0278364910385483.

[3] A.-D. Doan, A.M. Jawaid, T.-T. Do, and T.-J. Chin. G2D: from GTA to Data. *arXiv preprint arXiv:1806.07381*, pages 1–9, 2018.

[4] D. Galvez-López and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.

[5] P. Hansen and B. Browning. Visual place recognition using hmm sequence matching. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4549–4555, 2014.

[6] Inivation. Dvxplorer. In *2020 International CES*, 2020. URL https://www.ces.tech/Articles/2020/DVxplorer.aspx.

[7] G. Kim, B. Park, and A. Kim. 1-day learning, 1-year localization: Long-term lidar localization using scan context image. *IEEE Robotics and Automation Letters*, 4(2): 1948–1955, 2019.

[8] H. Kim, S. Leutenegger, and A. Davison. Real-time 3d reconstruction and 6-dof tracking with an event camera. volume 9910, pages 349–364, 10 2016. ISBN 978-3-319-46465-7. doi: 10.1007/978-3-319-46466-4_21.

[9] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[10] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128$\times$ 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2): 566–576, 2008.

[11] W. Maddern and S. Vidas. Towards robust night and day place recognition using visible and thermal imaging. 07 2012.

[12] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017. doi: 10.1177/0278364916679498. URL http://dx.doi.org/10.1177/0278364916679498.

[13] A.I. Maqueda, A. Loquercio, G. Gallego, N. García, and D. Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5419–5427, 2018.

[14] M. Milford, H. Kim, S. Leutenegger, and A. Davison. Towards visual slam with event-based cameras.

[15] M. J. Milford and G. F. Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *2012 IEEE International Conference on Robotics and Automation*, pages 1643–1649, 2012.

[16] R. Paul and P. Newman. Fab-map 3d: Topological mapping with spatial and visual appearance. In *2010 IEEE International Conference on Robotics and Automation*, pages 2649–2656, 2010.

[17] E. Pepperell, P.I. Corke, and M. Milford. All-environment visual place recognition with smart. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1612–1618, 2014.

[18] H. Rebecq, G. Gallego, E. Mueggler, and D. Scaramuzza. Emvs: Event-based multi-view stereo—3d reconstruction with an event camera in real-time. *International Journal of Computer Vision*, 11 2017. doi: 10.1007/s11263-017-1050-6.

[19] H. Rebecq, D. Gehrig, and D. Scaramuzza. Esim: an open event camera simulator. In *Conference on Robot Learning*, pages 969–982, 2018.

[20] T. Stone, M. Mangan, P. Ardin, and B. Webb. In *2014 Robotics: Science and Systems Conference*, 07 2014.

[21] B. Talbot, S. Garg, and M. Milford. Openseqslam2.0: An open source toolbox for visual place recognition under changing conditions. pages 7758–7765, 10 2018. doi: 10.1109/IROS.2018.8593761.

[22] O. Vysotska and C. Stachniss. Lazy data association for image sequences matching under substantial appearance changes. *IEEE Robotics and Automation Letters*, 1(1): 213–220, 2016.

[23] O. Vysotska and C. Stachniss. Effective visual place recognition using multi-sequence maps. *IEEE Robotics and Automation Letters*, 4(2):1730–1736, 2019.

[24] A. Zhu, L. Yuan, K. Chaney, and K. Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. In *Robotics: Science and Systems 2018*, 06 2018.

[25] A. Zhu, L. Yuan, K. Chaney, and K. Daniilidis. *Unsupervised Event-Based Optical Flow Using Motion Compensation: Munich, Germany, September 8-14, 2018, Proceedings, Part VI*, pages 711–714. 01 2019. ISBN 978-3-030-11023-9. doi: 10.1007/978-3-030-11024-6_54.