On Motion Blur and Deblurring in Visual Place Recognition

Timur Ismagilov[1]*, Bruno Ferrarini[2]*, Michael Milford[3]
Tan Viet Tuyen Nguyen[1], Sarvapali D. Ramchurn[1], Shoaib Ehsan[1,4]

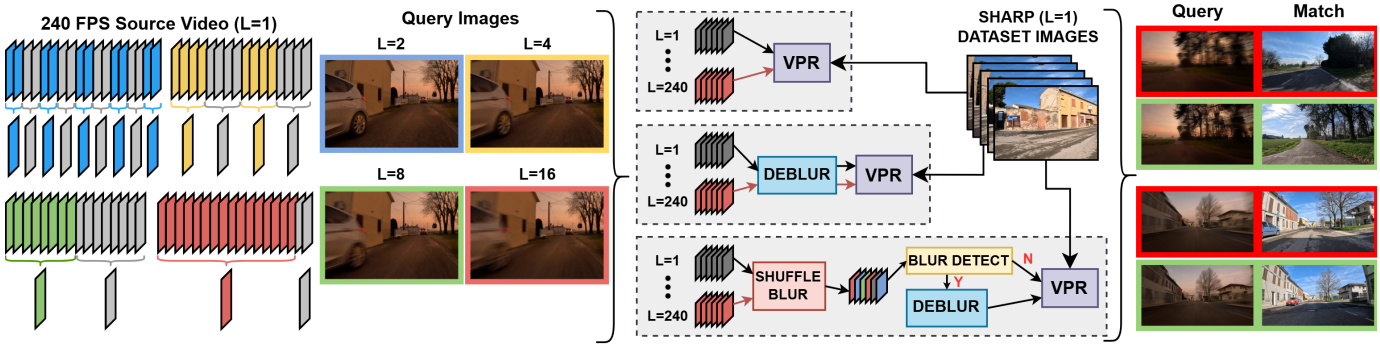

Fig. 1: We introduce a benchmark featuring motion blur and scene variation and evaluate VPR performance under motion blur, deblurring, and adaptive deblurring. Examples highlight incorrect (red) to correct (green) matches after deblurring.

***Abstract*—Visual Place Recognition (VPR) in mobile robotics enables robots to localize themselves by recognizing previously visited locations using visual data. While the reliability of VPR methods has been extensively studied under conditions such as changes in illumination, season, weather and viewpoint, the impact of motion blur is relatively unexplored despite its relevance not only in rapid motion scenarios but also in low-light conditions where longer exposure times are necessary. Similarly, the role of image deblurring in enhancing VPR performance under motion blur has received limited attention so far. This paper bridges these gaps by introducing a new benchmark designed to evaluate VPR performance under the influence of motion blur and image deblurring. The benchmark includes three datasets that encompass a wide range of motion blur intensities, providing a comprehensive platform for analysis. Experimental results with several well-established VPR and image deblurring methods provide new insights into the effects of motion blur and the potential improvements achieved through deblurring. Building on these findings, the paper proposes adaptive deblurring strategies for VPR, designed to effectively manage motion blur in dynamic, real-world scenarios.**

***Index Terms*—Localization, Vision-Based Navigation, Data Sets for Robotic Vision**

Manuscript received: December 9, 2024; Revised February 18, 2025; Accepted March 13, 2025.

This paper was recommended for publication by Pascal Vasseur upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the U.K. Engineering and Physical Sciences Research Council under Grant EP/Y009800/1, Grant EP/V00784X/1 and the EPSRC-UKRI Grant UKRI840 (PRIV-LOC).

*These authors equally contributed to this work.

[1]Timur Ismagilov, Tan Viet Tuyen Nguyen, Sarvapali D. Ramchurn and Shoaib Ehsan are with the School of Electronics and Computer Science, University of Southampton, SO17 1BJ Southampton, U.K. ti1e24@soton.ac.uk, tuyen.nguyen@soton.ac.uk, sdr1@soton.ac.uk, s.ehsan@soton.ac.uk

[2]Bruno Ferrarini is with MyWay srl, Via Osti, 6 - 29010 Vernasca (PC). bferrarini.ac.uk@gmail.com

[3]Michael Milford is with the QUT Centre for Robotics, School of Electrical Engineering and Robotics, Brisbane, QLD 4000, Australia. michael.milford@qut.edu.au

[4]Shoaib Ehsan is also with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, CO4 3SQ, U.K.

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

I. Introduction

In mobile robotics, Visual Place Recognition (VPR) allows robots to identify their position by matching visual data to previously encountered locations. It is a challenging task due to potential variations in illumination, weather, season, and viewpoint between the query image and the reference map in real-world scenarios. The impact of these changes on VPR performance has been extensively investigated [1], [2], [3], through numerous datasets available in the literature [4].

Motion blur is one of the major challenges remaining for VPR methods. It occurs not only in situations where there is rapid camera motion (such as for fast-flying drones [5]) but also in the case of relatively slow motion under low-illumination conditions where longer exposure times are necessary, potentially affecting a wide range of VPR applications. Despite this, the impact of motion blur on VPR methods is relatively unexplored. Although there are some datasets available [6], [7], [8], they do come with certain shortcomings (such as limited range of blur intensities, indoor conditions only, low-light conditions only etc.) which make them unsuitable for performing an in-depth investigation. Similarly, to our best knowledge, there has been no work on systematically analyzing the effects of deblurring when used as a mitigation strategy for VPR under a wide variety of blur intensities and in the presence of VPR-specific challenges, such as changes in illumination, weather, and viewpoint. This paper bridges these research gaps and makes three main contributions (as shown in Fig. 1):

- We introduce a motion blur-specific benchmark with datasets featuring varying blur levels and VPR-specific appearance changes, enabling comprehensive analysis without field image acquisition. Unlike existing VPR datasets that feature only limited intrinsic blur, our benchmark offers controllable motion blur intensity in outdoor scenes. Fig. 2 shows sample images of the same location in sharp and blurred versions across different traverses.

- A comprehensive evaluation of motion blur and deblurring in VPR settings is conducted using the methods depicted in Fig. 1. There has been little study on VPR under motion blur, therefore, this analysis leverages the proposed benchmark to empirically examine its impact on the performance of various state-of-the-art VPR techniques.
- Building on the outcomes of the first two contributions, various adaptive deblurring scenarios are evaluated, along with their associated computational costs. These assessments provide valuable insights to guide future experiments involving selective deblurring tailored to VPR-specific challenges, paving the way for advancements in blur-aware VPR techniques.

The paper is organized as follows: Section II reviews related work on VPR under motion blur and deblurring; Section III introduces the Blurry Places benchmark; Section IV details the experimental setup; Section V presents the results; and Section VI draws conclusions.

II. Related Work

This section reviews related work on VPR under motion blur and image deblurring. The impact of motion blur on VPR remains underexplored, with limited and often inadequate datasets available. For example, ETH3D [6] offers only three sequences of a single scene, while [7] matches blurred and sharp images for aerial applications with restricted blur intensity. Similarly, [8] provides a motion blur-specific dataset but is confined to low-light, indoor, and meter-scale trajectories, making it unsuitable for broader VPR applications like self-driving cars. Most VPR datasets seldom emphasize motion blur (see Table I) - typically featuring only fixed or incidental blur effects [9], limiting the analysis. Section III clarifies that realistic blur via frame averaging requires high frame rate data to capture smooth temporal details for motion interpolation and fine control of blur intensity; however, datasets like Pittsburgh250k [10] are collected at low frame rates, hindering this approach.

TABLE I: Dataset comparison: motion blur (MB) and variations in illumination (I), viewpoint (VP), and weather (W).

Dataset	VPR	Scene	fps	MB Emphasis	I	VP	W
St Lucia [11]	✓	Urban	30Hz	✗ (Incidental)	✓	slight	✗
Oxford RobotCar [12]	✓	Urban	<20Hz	✗ (Incidental)	✓	✗	✓
Pittsburgh250k [10]	✓	Urban	Low (GSV)	✗ (Incidental)	✓	✓	✗
KITTI raw [13]	✓	Urban	10Hz	✗ (Incidental)	slight	✗	✗
NYU-VPR [9]	✓	Urban	-	✓ (Incidental)	✓	✓	✓
DE-PR [14]	✓ (Limited)	Indoor	-	✓ (Fixed)	✗	✓	✗
ETH3D [6]	✓ (Limited)	Indoor	∼27.1Hz	✓ (Fixed)	✓	✓	✗
MBA-VO [8]	✓ (Limited)	Indoor	27Hz	✓ (Fixed)	slight	✓	✗
GoPro [15]	✗	Variety	240Hz	✓	✗	✓	✗
HIDE [16]	✗	Variety	240Hz	✓	✗	✓	✗
Blurry Places (Ours)	✓	Urban, Rural	240Hz	✓ (Controlled)	✓	✓	✓

Research on deblurring for VPR is also sparse, with existing datasets offering limited blur intensity and diversity. For example, 'GoPro' [15] uses urban 240fps videos averaging every 7–13 frames, and 'HIDE' [16] focuses on outdoor human movement averaging every 11 frames from 240 fps videos, yet neither are structured for VPR. [17] proposes a SLAM framework that integrates blur detection with DeblurGANv2 deblurring, achieving improved feature matching with direct deblurring, however, they address the method's limited generalization to diverse scenes. Conversely, [14] introduces a selective deblurring approach for indoor scene classification, showing that detecting and excluding blurred frames improves performance while applying any deblurring does not. Overall, while urban and indoor scene deblurring research exists [18], [19], [20], there is little analysis of how deblurring affects VPR performance under complex challenges, such as illumination, weather, and viewpoint variations. These gaps highlight the need for more comprehensive studies in this area.

III. Blurry Places Benchmark

Blurred frame generation is typically achieved either through blur-kernel-based algorithms or by averaging consecutive, short-exposure frames captured at high frame rates [15], [21], [22], [23], [24]. The latter method is considered more realistic [23], as it generates spatially varying (non-uniform) blur patterns that closely model the effects caused by perspective shifts and depth variation. In contrast, kernel-based methods often struggle with occlusion, depth variation and deformations, and kernel estimation is computationally expensive and sensitive to noise saturation. Moreover, studies on deblurring models trained on images produced by frame averaging have demonstrated better performance than those trained on synthetically blurred images generated by kernel-based algorithms [21]. Consequently, recent and well-established motion blur datasets [15], [16], [23] also employ the frame-averaging approach.

The blurred images are obtained by averaging the frames of a slow-motion video at 240 fps captured with a GoPro 11 Black. Unlike filter-based methods (e.g., Gaussian filtering), this approach embeds the motion of a video in a blurred image though mimicking the physical image formation in digital cameras [25]. In detail, this approach models the image formation as the integral over time of the image illuminating the camera's sensors during exposure time (τ):

$$\mathbf{I}(x) = \frac{1}{\tau}\int_0^{\tau} \mathbf{I}(t, x(t))\, dt\,. \tag{1}$$

$\mathbf{I}(t, x(t))$ is the image to which the sensor is exposed at the time t during the exposure time (τ). $\mathbf{I}(x)$ is a frame captured by the camera. This model describes how $\mathbf{I}(t, x(t))$ variations over time induce motion blur. Faster movements generate wider variations and, therefore, more intense motion blur in the part of the sensor where they occur.

We build our datasets from a discrete sequence of video frames, thus Eq. 1 is approximated with the following series:

$$\mathbf{I}(x) \approx \frac{1}{n}\sum_{i=0}^{n-1} \mathbf{I_i}(x)\,. \tag{2}$$

The integral is replaced by a sum of n sharp frames indicated with $\mathbf{I_i}(x)$. Eq. 2 is then parameterized as follows to build multiple blurring intensities:

$$\mathbf{B_j^L}(x) = \frac{1}{L}\sum_{i=j}^{j+L} \mathbf{I_i}(x)\,. \tag{3}$$

Fig. 2: A place in Luzzara in 3 traverses and blur intensities.

$\mathbf{B_j^L}$ is the blurred image from averaging L sharp frames starting at frame j of the source video, with increasing blur intensity as L increases. We use L in the remainder of the paper to indicate the amount of motion blur of an image, with $L = 1$ corresponding to a sharp image. Fig. 1 illustrates the synthetic blur generation process from Eq. 3 with sample images at several blur levels; the colour code used serves only to highlight the contributing sharp frames.

Sample images with motion embedding are shown in Fig. 2, where some areas remain relatively sharp depending on movement direction. Note that L relates to the exposure time in Eq. 1 via the source video frame rate (V_{fps}).

$$\tau \approx \hat{\tau} = \frac{L}{V_{fps}}. \tag{4}$$

Here, $\hat{\tau}$ approximates the shutter time τ of the blur model described by Eq. 1. The higher the video frame rate, V_{fps}, the finer the control on the blur level, as L can vary in a wider range within the same exposure time. For example, consider two videos: one collected at 240 fps and the other at 30 fps at the same speed. With $L = 2$, the 240 fps video results in a 120 fps blurred dataset, while the 30 fps video only achieves a 15 fps blurred dataset with significantly higher exposure time. Alternatively, using the exposure time $\hat{\tau}$ as a reference, the same blur can be applied to videos at different V_{fps}. For example, the blurring corresponding to $\hat{\tau} = 0.5$s requires $L = 120$ and $L = 15$ for the 240 and 30 fps videos, respectively. Furthermore, a higher frame rate enhances motion continuity at the same speed, producing a more realistic motion blur through averaging. Driven by this consideration, the raw data to build the proposed dataset are videos captured with a GoPro 11 black at the highest possible frame rate of 240 fps.

The benchmark is organized in datasets. A dataset includes several sets of images captured along the same route at different times, so it is always possible to use one as a reference (previously visited locations) and the other as a query (a second traverse through the route). The traverses are generally different due to driving trajectories causing lateral shifts, weather, illumination, seasons, and dynamic elements like pedestrians, appearing only in one of the traverses. These appearance changes are used jointly with motion blur to set up different testing scenarios. Specifically, the proposed benchmarks include 9 traverses captured along the three outdoor routes in urban and country-side environments. These are named as the small towns where they are recorded: Luzzara (LZR), Guastalla (GST) and Casoni (CSN), in Italy. The recording list is shown in Table II, indicating the environmental conditions during the recording, the duration of the video, and the number of sharp frames it contains.

TABLE II: Traverse recorded in Luzzara (LZR), Guastalla (GST), Casoni (CSN) routes.

Track	Traverse	Time	Condition	length	duration	Frames
Luzzara (LZR)	01	Morning	Cloudy	3.2 Km	06:52	98810
	02	Dusk	Sunny		06:33	94442
	03	Morning	Sunny		05:57	85679
Guastalla (GST)	01	Afternoon	Sunny	3.6 Km	05:59	86129
	02	Afternoon	Sunny		05:38	81164
	03	Morning	Sunny		05:47	83272
Casoni (CSN)	01	Noon	Cloudy	4.3 Km	06:08	77889
	02	Dusk	Sunny		05:24	88476
	03	Morning	Sunny		05:28	78131

Every video in Table II was processed with the method described by Eq. 3 using the following nine values of L:

$$L \in \{1, 10, 20, 30, 40, 60, 80, 120, 240\}, \tag{5}$$

where $L = 1$ (no blur) establishes a baseline for evaluating the impact of blurring. We found empirically that this set of blur levels ensures a comprehensive characterization of VPR performance. Nevertheless, any blur level can be generated using the provided scripts and data[1], including extending datasets captured in the same manner to domains beyond those tested (e.g., extreme weather, indoors). Traversals can be combined into query-reference pairs to conduct experiments on motion blur at any intensity in L, with or without the environmental conditions available (e.g. night). Moreover, selecting the datasets with $L = 1$ (no blur), the resulting benchmark can be used to assess VPR under the viewing conditions as with any other well-established VPR datasets such as St. Lucia [11] and Oxford Car [12].

IV. Experimental Setup

To demonstrate the utilization of the proposed benchmark, we use the datasets formed by the pairs enlisted in Table III. The first three rows use the same traverse as both query and reference. All the experiments use sharp images as a reference, while those in the query traverse are blurred at 9 increasing intensities as in Eq. 5. This setup excludes any other change but motion blur from the VPR performance analysis. The last three rows present more challenging (and realistic) scenarios where the query and reference images are captured in different traverses. These scenarios include $L = 1$ to establish a performance baseline under weather, illumination, and viewpoint, along with additional motion blur at increasing levels of L. The same traverses are used for analysis of direct deblurring, and a subset of them for adaptive deblurring, where the blur intensities are shuffled into one sequence as described in their respective results. Several well-established VPR methods were systematically evaluated in our benchmark to cover several approaches, including MixVPR [27], AnyLoc [28], CosPlace [29], EigenPlaces [30], HDC-DELF [31], FloppyNet [32], Patch-NetVLAD [33], ORB [34], and SAD [2], with

[1]https://github.com/bferrarini/MotionBlurGenerator

TABLE III: The traverse pairs used for the experiments. Each comes with one or more appearance changes: Motion Blur (**MB**), Weather (**W**), Illumination (**I**), and ViewPoint (**VP**).

Pair ID	Query	Reference	MB	W	I	VP
LZR-MBlur	01-Evening-Sunny (411 images)		X			
GST-MBlur	01-Afternoon-Sunny (358 images)		X			
CSN-MBlur	01-Noon-Cloudy (324 images)		X			
LZR-Mixed	02-Morning-Sunny (393 images)	03-Morning-Cloudy (356 images)	X	X		X
GST-Mixed	02-Afternoon-Sunny (338 images)	01-Afternoon-Sunny (358 images)	X			X
CSN-Mixed	02-Dusk-Sunny (325 images)	03-Morning-Sunny (325 images)	X		X	X

blur intensities ranging from L=1 to L=240. Patch-NetVLAD adopts a two-stage retrieval process with its own matching step, while the others use a single-stage approach with cosine similarity for descriptor comparison. Offline deblurring is applied to each level of blur intensity in all datasets, using three state-of-the-art deblurring methods: DeblurGANv2 [35], FFTFormer [36] and GShift-Net [37]. We then perform the same VPR analysis to get a clear comparison of the impacts of deblurring across blur and scene variation. We evaluate VPR performance using the Area Under Curve (AUC) of Precision-Recall curves. A higher AUC indicates stronger performance, reflecting consistent retrieval of relevant images with minimal false positives, whereas a lower AUC implies difficulty maintaining high precision and recall simultaneously. For VPR methods, we follow the setup shared by the authors, while deblurring is applied to the raw benchmark images to ensure an unbiased evaluation of VPR performance. Adaptive deblurring employs Laplacian variance for blur detection due to its simplicity and efficiency [5]. Empirical tests revealed a wide variance gap between sharp and severely blurred images (L>40), and results confirm deblurring is only significant with frequent, high blur intensities. Consequently, fixed thresholds of 50 for the shuffled CSN-Mixed dataset and 200 for the shuffled GST-Mixed and LZR-Mixed datasets achieved near-perfect classification of such blurred frames.

V. Results and Discussion

A. Motion Blur

The effect of motion blur is examined using the traverse pairs LZR-MBlur, GST-MBlur, and CSN-MBlur. These three datasets use the same traverse as both reference and query, so the images differ only by the amount of motion blur. The results are shown in Fig. 3 for increasing blurring intensities in query images. All the plots start from AUC $= 1$ as at $L = 1$, query and reference images are identical. As expected, every VPR method is negatively impacted. FloppyNet achieves the most robust performance regardless of it being a Binary Neural Network (BNN) - a highly compact variant of neural network where the weights and activations are constrained to 1-bit values to reduce memory and computational cost. FloppyNet's shallow architecture of just three layers makes it tolerant to viewpoint-free appearance changes [32] and exceeds severe motion blur performance in GST-MBlur and CSN-MBlur. SAD is considered to tolerate viewpoint-free changes [38], however, it exhibits steeper performance decay, dropping quickly to an AUC between 0.2 and 0.3. This decay is greater with LZR-MBlur, which may arise from the dataset's reduced lighting and contrast. Patch-NetVLAD starts strongly but degrades quickly, achieving a lower performance among the learning-based methods for larger blur, and comparable or even worse performance than ORB-VLAD in LZR-MBlur and GST-MBlur. ORB-VLAD, being non-learning-based, performs competitively with the learning-based methods on LZR-MBlur and CSN-MBlur, however quickly suffers on GST-MBlur. The remaining models exhibit similar performance and

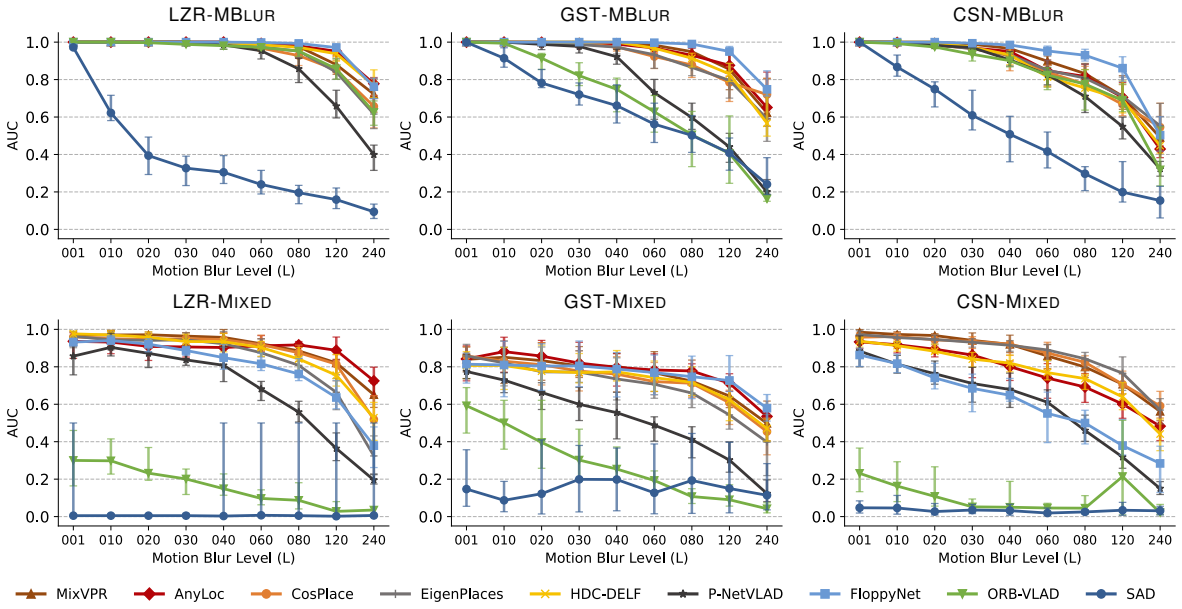

Fig. 3: AUC at various motion blur levels. Blur-only traverse results are shown at the top, while the bottom row shows the results for mixed conditions. The bars indicate the 95% confidence interval obtained using block bootstrapping [26].

degrading behaviour, being the more robust choices out of all the methods capable of handling large motion blur reasonably well. This is reflected in the measured confidence ranges, with top-performing models having little variance in performance, but becoming less certain as blur intensity increases. We expect AnyLoc to outperform in diverse scenes as it utilizes general-purpose features from large-scale pretrained models. Overall, CSN-MBlur proves to be the variation-free dataset that VPR models struggled the most on with severe blur intensity.

B. Mixed Conditions

The mixed-condition datasets used for these experiments are shown in the last three rows of Table III, combining motion blur with other appearance variations that are typical of VPR applications, such as viewpoint, weather, and illumination. Fig. 3 shows the AUC trend for increasing motion blur, with $L = 1$ representing the baseline of VPR performance, which degrades with the addition of motion blur. Regardless of the model, GST-Mixed proves to be a particularly challenging dataset for VPR, indicating that this query traverse introduces non-trivial examples that shift AUC down. The difficulty of these traverses is highlighted by the wider width of the 95% confidence intervals compared to the intervals in the blur-only traverses. AnyLoc extracts per-pixel features using large-scale pretrained models, making it better suited for general, broad-spectrum applications. With increasing motion blur, AnyLoc performs with little degradation on the LZR-Mixed and GST-Mixed datasets, highlighting its robustness to blur in weather and viewpoint variation, yet struggles to achieve competitive performance in the presence of illumination variation. MixVPR shows top-end performance across all dataset variations, with itself, Cosplace and EigenPlaces achieving the best performance on CSN-Mixed by a wide margin. At $L = 1$, these 3 models achieve close to AUC = 1 on CSN-Mixed and LZR-Mixed, revealing their tolerance to illumination and weather differences. However, Cosplace and EigenPlaces degrade at a larger rate on LZR-Mixed and GST-Mixed, so MixVPR may be better equipped to handle motion blur. FloppyNet achieves competitive performance on GST-Mixed, yet in the presence of weather or illumination variance, it struggles to compete with the other methods. P-NetVLAD, ORB-VLAD and SAD stand out for their intolerance to these mixed conditions. P-NetVLAD appears to equally suffer across each dataset, starting close to 0.8 AUC and dropping to 0.2. Although ORB-VLAD and SAD showed some tolerance in the blur-only traverses at sharp and small blur intensity, they suffer in mixed conditions in any case and are incapable of VPR.

C. Deblurring Query Images

We examine the impact of deblurring across various blur intensities to assess its effect on VPR accuracy. For CSN-

TABLE IV: AUC of VPR methods with different deblurring methods on CSN-Mixed. Bold indicates the best performing combinations within each query blur level. Improvements after deblurring highlighted in green; deterioration in red.

		Query Blur Level											
VPR Model	**Deblur Method**	**001**	**010**	**020**	**030**	**040**	**060**	**080**	**120**	**240**	Avg.	Std.	Avg. Deblur + Extract time / query (ms)
MixVPR	No Deblur	0.98	0.97	0.96	0.94	0.92	0.85	0.79	0.70	0.56	0.85	0.13	3.72
	DeblurGANv2	0.98	0.97	**0.97**	**0.97**	**0.95**	**0.91**	**0.88**	**0.81**	0.59	**0.89**	**0.11**	38.41
	GShift-Net	0.98	**0.98**	**0.97**	0.95	0.93	0.88	0.82	0.74	**0.65**	0.88	**0.11**	394.65
	FFTFormer	0.98	0.97	0.96	0.95	0.93	0.88	0.84	0.75	0.52	0.86	0.14	590.33
AnyLoc	No Deblur	0.93	0.91	0.89	0.86	0.80	0.74	0.69	0.60	0.48	0.76	**0.14**	624.84
	DeblurGANv2	0.93	0.92	**0.91**	0.88	**0.83**	0.75	**0.75**	0.63	0.43	**0.78**	0.15	659.53
	GShift-Net	**0.94**	**0.93**	0.90	0.85	**0.83**	0.72	0.72	**0.64**	**0.49**	**0.78**	**0.14**	1015.77
	FFTFormer	**0.94**	0.92	0.90	**0.89**	0.82	**0.76**	0.72	0.63	0.44	**0.78**	0.15	1211.45
CosPlace	No Deblur	0.95	0.96	0.94	0.93	0.92	0.87	0.82	0.70	0.59	0.85	0.12	3.78
	DeblurGANv2	0.96	**0.97**	**0.97**	**0.96**	**0.96**	**0.95**	**0.91**	**0.87**	0.66	**0.91**	**0.09**	38.47
	GShift-Net	**0.98**	**0.97**	**0.97**	0.94	0.92	0.89	0.84	0.71	**0.74**	0.89	**0.09**	394.71
	FFTFormer	0.95	0.95	0.94	0.93	0.91	0.88	0.84	0.79	0.62	0.87	0.10	590.39
EigenPlaces	No Deblur	0.97	0.95	0.94	0.93	0.91	0.89	0.84	0.76	0.56	0.86	0.12	3.76
	DeblurGANv2	0.97	0.97	**0.96**	**0.96**	**0.95**	**0.93**	**0.88**	**0.84**	0.72	**0.91**	**0.08**	38.45
	GShift-Net	**0.98**	**0.98**	**0.96**	0.95	0.92	0.89	0.84	0.77	**0.75**	0.89	**0.08**	394.69
	FFTFormer	0.96	0.95	0.94	0.93	0.92	0.90	0.86	0.79	0.62	0.87	0.10	590.37
HDCDELF	No Deblur	0.93	0.91	0.88	0.84	0.82	0.76	0.73	0.63	0.44	0.77	0.14	610.38
	DeblurGANv2	0.94	**0.93**	**0.92**	**0.90**	**0.86**	**0.83**	**0.78**	**0.74**	0.53	**0.82**	**0.12**	645.07
	GShift-Net	**0.95**	0.92	0.88	0.83	0.78	0.74	0.70	0.62	**0.57**	0.78	**0.12**	1001.31
	FFTFormer	0.94	**0.93**	0.90	0.87	0.84	0.79	0.75	0.68	0.49	0.80	0.13	1196.99
P-NetVLAD	No Deblur	**0.88**	**0.81**	0.76	0.71	0.67	0.61	0.46	0.31	0.15	0.59	0.22	10.49
	DeblurGANv2	0.86	0.78	**0.81**	**0.77**	**0.75**	**0.69**	**0.61**	**0.51**	0.24	**0.66**	0.18	45.18
	GShift-Net	0.78	0.79	0.74	0.68	0.65	0.50	0.49	0.33	**0.30**	0.58	**0.17**	401.42
	FFTFormer	0.82	0.80	0.79	**0.77**	0.68	0.63	0.52	0.39	0.15	0.61	0.21	597.10
FloppyNet	No Deblur	0.86	0.81	0.74	0.68	0.64	0.55	0.50	0.37	0.28	0.60	0.18	41.63
	DeblurGANv2	0.86	**0.84**	**0.80**	**0.78**	**0.75**	**0.70**	**0.59**	**0.48**	0.34	**0.68**	**0.16**	76.32
	GShift-Net	0.86	0.82	0.75	0.69	0.64	0.57	0.52	0.41	**0.34**	0.62	**0.16**	432.56
	FFTFormer	0.86	0.82	0.75	0.70	0.66	0.57	0.51	0.40	0.29	0.62	0.18	628.24
ORB-VLAD	No Deblur	**0.23**	0.16	0.10	0.05	0.05	0.04	0.04	**0.21**	0.02	0.10	0.07	17.26
	DeblurGANv2	0.18	**0.19**	0.14	0.09	**0.08**	**0.07**	0.03	0.04	0.01	0.09	0.06	51.95
	GShift-Net	0.22	0.16	**0.16**	**0.12**	**0.08**	0.03	**0.05**	0.03	**0.04**	0.10	0.06	408.19
	FFTFormer	0.21	0.13	0.10	0.07	**0.08**	0.06	0.04	0.01	0.01	0.08	0.06	603.87
SAD	No Deblur	0.04	0.04	0.02	0.03	0.03	0.01	0.02	0.03	0.03	0.03	0.01	3.99
	DeblurGANv2	0.04	0.03	**0.04**	0.03	**0.04**	**0.02**	0.02	0.02	0.03	0.03	0.01	38.68
	GShift-Net	0.04	0.03	0.02	0.02	0.03	**0.02**	0.02	**0.04**	**0.04**	0.03	0.01	394.92
	FFTFormer	0.03	**0.05**	0.02	**0.04**	0.02	**0.02**	**0.03**	0.01	0.02	0.03	0.01	590.60

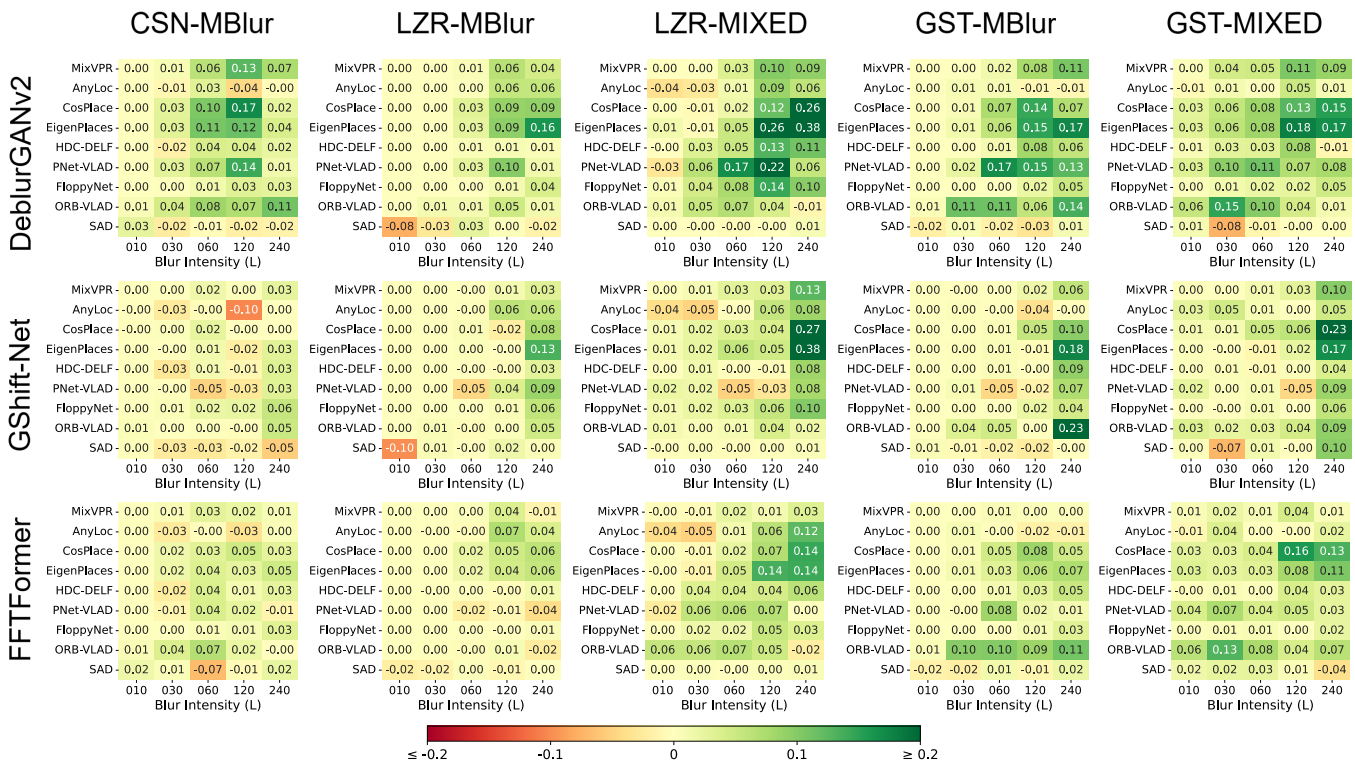

Fig. 4: Heatmaps showing the **difference** in AUC performance after deblurring the other datasets (columns) with each deblurring method (rows), with different VPR method and blur intensity combinations.

Mixed, Table IV presents full results, including the combined average deblurring and query extraction times, while Fig. 4 shows performance differences for the other datasets. Deblurring effectiveness depends on the method, VPR model, and blur intensity, with greater improvements at severe blur levels. Overall, VPR models resilient to appearance variations achieve greater average AUC and lower AUC standard deviation across blur intensities due to enhanced image restoration.

Generally, CosPlace and EigenPlaces respond positively and consistently with deblurring. They use global image descriptors, making them highly scalable and benefiting from deblurring methods that restore spatial context across the image. MixVPR adopts a relatively simple approach too, using a holistic feature aggregation technique, and benefits in most cases with deblurring. Unexpectedly, the effectiveness of all three deblurring methods on AnyLoc performance was limited, with small improvements in CSN-Mixed, LZR-Mixed and LZR-MBlur, reinforcing the fact that it may already be inherently more tolerant to motion blur. Likewise, FloppyNet's strong performance in motion blur-only datasets means it benefits little from deblurring in those cases, however, in mixed conditions, it shows small improvements, particularly with DeblurGANv2. Deblurring provides some improvement for HDC-DELF under severe motion blur, but it is often minor and less impactful than for methods that rely exclusively on sharp, global image features. Both ORB-VLAD and P-NetVLAD show tangible improvement when combined with DeblurGANv2 across all datasets, highlighting their intolerance for motion blur, especially with the GST-Mixed dataset, which degraded all model performance significantly. Regardless of deblurring, SAD is incapable of effective VPR unless the images are sharp and do not contain complex variations.

DeblurGANv2 proves to be the most effective and robust choice of the three when applied to VPR, providing larger and more consistent improvements in VPR performance. Its strong performance can be attributed to its emphasis on multi-scale deblurring. Its use of a feature pyramid network enables deblurring across broad spatial resolutions, which may support larger and more complex blur patterns. Unlike the other methods, GShift-Net is developed for video deblurring and implicitly aggregates temporal and spatial information in its shift blocks. By shifting feature groups across frames, it increases the receptive field. This method performs relatively poorer than the other three, however, interestingly, we can see that GShift-Net accounts for some of the best scores with blur level 240 by a large margin, such as when combined with Eigenplaces on CSN-Mixed, we see a gain of 0.152. This may indicate that adjacent frames could provide valuable information when context is lost under extreme blur conditions. Future work could explore the importance of temporal information under varying degrees of image degradation. The impacts of FFTFormer resemble DeblurGANv2, however, it does not provide significant gains. More often than not, it still shows measurable improvement over no deblurring, and even when performance drops, it is minimal. These results are unexpected as FFTFormer achieves greater performance than DeblurGANv2 on 'GoPro', 'RealBlur' [39] and 'HIDE' evaluation. Their differences in training data may align more

TABLE V: Comparison of inference efficiency and AUC for different deblurring strategies on shuffled mixed-condition datasets, measured with the PyJoules library on an AMD EPYC 7452 CPU and Nvidia A100 GPU.

Setup	Method	Timing (s)				Energy (kJ)				AUC
		Extract	Detect	Deblur	Avg/Query (ms)	Extract	Detect	Deblur	Avg/Query (J)	
DeblurGANv2-CosPlace-CSN-Mixed	No Deblur	1.39	–	–	3.76	0.41	–	–	1.12	0.93
	All Deblur	1.40	–	12.80	38.48	0.42	–	2.80	8.75	0.97
	Detect+Deblur	1.39	0.71	7.76	26.72	0.42	0.10	1.73	6.10	0.97
DeblurGANv2-EigenPlaces-GST-Mixed	No Deblur	1.28	–	–	3.78	0.40	–	–	1.18	0.69
	All Deblur	1.28	–	11.69	38.26	0.38	–	2.57	8.70	0.75
	Detect+Deblur	1.29	0.65	9.41	33.48	0.40	0.094	2.00	7.36	0.75
DeblurGANv2-EigenPlaces-LZR-Mixed	No Deblur	1.34	–	–	3.76	0.41	–	–	1.15	0.93
	All Deblur	1.34	–	12.42	38.65	0.41	–	2.74	8.85	0.95
	Detect+Deblur	1.35	0.72	6.42	23.85	0.41	0.10	1.39	5.34	0.95

or less with the variations present in our benchmark. Notably, DeblurGANv2 and GShift-Net were further trained on the DVD [40] dataset, providing additional videos captured at 240 fps, and DeblurGANv2 on the NFS video dataset [41]. FFTFormer, however, focuses on RealBlur and HIDE datasets, highlighting the need for careful consideration of the domain when building deblurring VPR or SLAM systems.

D. Adaptive Deblurring

Results have shown that certain models such as Cosplace and Eigenplaces exhibit top VPR performance on the benchmark, being more robust to scene variations and responding well to deblurring. Regardless, the performance gains are often small or even degraded with small motion blur intensity, motivating an adaptive deblurring approach which may achieve improvement in computational cost with minimal change in performance.

Table V compares several significant adaptive deblurring scenarios using the best method, DeblurGANv2, which yielded larger gains across blur levels (see Fig. 4). We measured inference time and energy consumption using PyJoules on a single AMD EPYC 7452 CPU and Nvidia A100 GPU. To avoid null values in short per-image measurements - a limitation of the profiling software, we measured the total cost of inference in each stage separately and aggregated them, reflecting the cumulative inference overhead while ensuring consistent and reliable values across all scenarios. Based on prior results, adaptive strategies are justified only in scenarios with frequent severe blur. To represent these scenarios, the shuffled datasets CSN-Mixed and LZR-Mixed include roughly half sharp ($L = 1$) images, with the remainder spanning blur levels $L = 60$ to $L = 240$. The GST-Mixed shuffled dataset, which exhibited relatively less improvement, contains $\approx 20\%$ sharp, the rest at blur level $L = 120$ and $L = 240$. The same adaptive method can be applied to specific cases with other VPR or deblur methods, such as FFTFormer and Anyloc in LZR-Mixed, which had degradation with small motion blur. In these shuffled mixed-condition scenarios, an all-deblur strategy yields an AUC gain between 0.02 and 0.06. Given that FFTFormer and GShift-Net did not perform as well in previous analyses, they would struggle more in mixed datasets, making them unsuitable for real adaptive deblurring regardless of their computational overhead. Having used the largest backbone, Inception-ResNet-v2, for DeblurGANv2, both the energy consumption and total time significantly increase (processing $\sim$25 fps). In comparison, using a prior detection stage achieves the same performance gain with less additional overhead, as seen in the shuffled CSN-Mixed and LZR-Mixed scenarios. The proportion of sharp images affects the trade-off between computational cost and performance, with sharp images avoiding redundant and costly deblurring operations, while blurred images, though requiring additional time and energy expenditure, enable potential gains in recognition accuracy. Adaptive deblurring is shown to be beneficial over direct deblurring when the VPR model tolerates complex scene variations and when severe, frequent blur allows for significant restoration improvements. In these cases, since most of the overhead comes from deblurring, an adaptive strategy offers substantial benefits.

E. Practical Implications

While these systems run sufficiently fast on GPUs, deploying them on resource-constrained, motion-blur-prone devices like UAVs is challenging. [5] used a Raspberry Pi v2.1 camera for non-learning-based VSLAM, which was limited to 20 fps for successful tracking, even with offline detection and deblurring. Traditional deblurring methods (e.g., Wiener, Lucy-Richardson) are also iterative and slow for large images, limiting real-time use. Consequently, one could explore techniques like quantization in deblurring pipelines [42], developing blur-robust VPR methods that bypass explicit deblurring, or employing spiking neural networks [43] with event-based cameras or neuromorphic hardware - a promising avenue for improving efficiency.

VI. Conclusions

Our benchmark enables comprehensive VPR analysis under motion blur. In certain configurations, combining VPR methods with deblurring improves matching performance and robustness, though real-time adaptive deblurring on resource-limited devices remains challenging despite improvements in efficiency when blur is significant. Future work could extend the blur model to other domains, or explore end-to-end sequence-based pipelines that make use of temporal information [37], [44]. In extreme conditions (e.g., adverse weather), generic deblurring may struggle. Exploring domain-adaptive deblurring or multi-modal sensing using event-based cameras, LiDAR or IMU data could improve accuracy, particularly in high-speed or low-light scenarios.

REFERENCES

[1] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, "On the performance of ConvNet features for place recognition," in *2015 IEEE/RSJ IROS*, 2015, pp. 4297–4304.

[2] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *2012 IEEE ICRA*. IEEE, 2012, pp. 1643–1649.

[3] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi, "Visual place recognition with repetitive structures," in *2013 IEEE CVPR*, 2013, pp. 883–890.

[4] M. Zaffar, S. Garg, M. Milford, J. Kooij, D. Flynn, K. McDonald-Maier, and S. Ehsan, "VPR-Bench: An open-source visual place recognition evaluation framework with quantifiable viewpoint and appearance change," *IJCV*, pp. 1–39, 2021.

[5] B. Şimşek and H. Ş. Bilge, "A novel motion blur resistant vSLAM framework for Micro/Nano-UAVs," *Drones*, vol. 5, no. 4, 2021.

[6] T. Schops, T. Sattler, and M. Pollefeys, "BAD SLAM: Bundle Adjusted Direct RGB-D SLAM," in *2019 IEEE/CVF CVPR*. Long Beach, CA, USA: IEEE, Jun. 2019, pp. 134–144.

[7] D. Dai, L. Zheng, G. Yuan, H. Zhang, Y. Zhang, H. Wang, and Q. Kang, "Real-time and high precision feature matching between blur aerial images," *PLOS ONE*, vol. 17, no. 9, p. e0274773, Sep. 2022, publisher: Public Library of Science.

[8] P. Liu, X. Zuo, V. Larsson, and M. Pollefeys, "MBA-VO: Motion blur aware visual odometry," in *2021 IEEE/CVF ICCV*, 2021, pp. 5530–5539.

[9] D. Sheng, Y. Chai, X. Li, C. Feng, J. Lin, C. Silva, and J.-R. Rizzo, "NYU-VPR: Long-term visual place recognition benchmark with view direction and data anonymization influences," in *2021 IEEE/RSJ IROS*, 2021, pp. 9773–9779.

[10] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi, "Visual place recognition with repetitive structures," in *CVPR*, 2013.

[11] M. Warren, D. McKinnon, H. He, and B. Upcroft, "Unaided stereo vision based pose estimation," in *ACRA*, G. Wyeth and B. Upcroft, Eds. Brisbane: ARAA, 2010.

[12] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *IJRR*, vol. 36, no. 1, pp. 3–15, 2017.

[13] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *CVPR*, 2012.

[14] P. Wozniak and B. Kwolek, "Deep embeddings-based place recognition robust to motion blur," in *Proc. IEEE/CVF ICCV Workshops*, October 2021, pp. 1771–1779.

[15] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *2017 IEEE CVPR*. Los Alamitos, CA, USA: IEEE Computer Society, Jul. 2017, pp. 257–265.

[16] Z. Shen, W. Wang, X. Lu, J. Shen, H. Ling, T. Xu, and L. Shao, "Human-aware motion deblurring," in *2019 IEEE/CVF ICCV*, 2019, pp. 5571–5580.

[17] J. Guo, R. Ni, and Y. Zhao, "DeblurSLAM: A novel visual slam system robust in blurring scene," in *2021 IEEE 7th ICVR*, 2021, pp. 62–68.

[18] T. Kim, H. Cho, and K.-J. Yoon, "Frequency-aware event-based video deblurring for real-world motion blur," in *Proc. IEEE/CVF CVPR*, June 2024, pp. 24 966–24 976.

[19] Y. Xiang, H. Zhou, C. Li, F. Sun, Z. Li, and Y. Xie, "Deep learning in motion deblurring: current status, benchmarks and future prospects," *The Visual Computer*, pp. 1–27, 09 2024.

[20] M. Suin, K. Purohit, and A. N. Rajagopalan, "Spatially-attentive patch-hierarchical network for adaptive motion deblurring," in *2020 IEEE/CVF CVPR*, 2020, pp. 3603–3612.

[21] M. Noroozi, P. Chandramouli, and P. Favaro, "Motion deblurring in the wild," in *Pattern Recognition*, V. Roth and T. Vetter, Eds. Cham: Springer International Publishing, 2017, pp. 65–77.

[22] T. H. Kim, S. Nah, and K. M. Lee, "Dynamic video deblurring using a locally adaptive blur model," *IEEE PAMI*, vol. 40, no. 10, pp. 2374–2387, 2018.

[23] Q. Guo, W. Feng, R. Gao, Y. Liu, and S. Wang, "Exploring the Effects of Blur and Deblurring to Visual Object Tracking," *IEEE Transactions on Image Processing*, vol. 30, pp. 1812–1824, 2021.

[24] T. Brooks and J. T. Barron, " Learning to Synthesize Motion Blur ," in *2019 IEEE/CVF CVPR*. Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2019, pp. 6833–6841.

[25] K. Ikeuchi, Ed., *Computer Vision: A Reference Guide*. Cham: Springer International Publishing, 2021.

[26] W. Härdle, J. Horowitz, and J.-P. Kreiss, "Bootstrap methods for time series," *International Statistical Review*, vol. 71, no. 2, pp. 435–459, 2003.

[27] A. Ali-Bey, B. Chaib-Draa, and P. Giguére, "MixVPR: Feature mixing for visual place recognition," in *2023 IEEE/CVF WACV*, 2023, pp. 2997–3006.

[28] N. Keetha, A. Mishra, J. Karhade, K. M. Jatavallabhula, S. Scherer, M. Krishna, and S. Garg, "AnyLoc: Towards universal visual place recognition," *IEEE RA-L*, vol. 9, no. 2, pp. 1286–1293, 2024.

[29] G. Berton, C. Masone, and B. Caputo, "Rethinking visual geo-localization for large-scale applications," in *Proc. IEEE/CVF CVPR*, June 2022, pp. 4878–4888.

[30] G. Berton, G. Trivigno, B. Caputo, and C. Masone, "EigenPlaces: Training viewpoint robust models for visual place recognition," in *Proc. IEEE/CVF ICCV*, October 2023, pp. 11 080–11 090.

[31] P. Neubert and S. Schubert, "Hyperdimensional computing as a framework for systematic aggregation of image descriptors," in *2021 IEEE/CVF CVPR*. Nashville, TN, USA: IEEE, Jun. 2021, pp. 16 933–16 942.

[32] B. Ferrarini, M. J. Milford, K. D. McDonald-Maier, and S. Ehsan, "Binary Neural Networks for memory-efficient and effective visual place recognition in changing environments," *IEEE T-RO*, vol. 38, no. 4, pp. 2617–2631, 2022.

[33] S. Hausler, S. Garg, M. Xu, M. Milford, and T. Fischer, "Patch-NetVLAD: Multi-Scale Fusion of Locally-Global Descriptors for Place Recognition," in *2021 IEEE/CVF CVPR*. Nashville, TN, USA: IEEE, Jun. 2021, pp. 14 136–14 147.

[34] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Computer Vision (ICCV), 2011 IEEE ICCV*. IEEE, 2011, pp. 2564–2571.

[35] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better," in *IEEE ICCV*, Oct 2019.

[36] L. Kong, J. Dong, J. Ge, M. Li, and J. Pan, "Efficient frequency domain-based transformers for high-quality image deblurring," in *2023 IEEE/CVF CVPR*, 2023, pp. 5886–5895.

[37] D. Li, X. Shi, Y. Zhang, K. C. Cheung, S. See, X. Wang, H. Qin, and H. Li, "A simple baseline for video restoration with grouped spatial-temporal shift," in *Proc. IEEE/CVF CVPR*, June 2023, pp. 9822–9832.

[38] N. Sünderhauf, P. Neubert, and P. Protzel, "Are we there yet? challenging seqslam on a 3000 km journey across all four seasons," in *Proc. of Workshop on Long-Term Autonomy, IEEE ICRA*, 2013, p. 2013.

[39] J. Rim, H. Lee, J. Won, and S. Cho, "Real-world blur dataset for learning and benchmarking deblurring algorithms," in *Proc. ECCV*, 2020.

[40] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang, "Deep video deblurring for hand-held cameras," in *Proc. IEEE CVPR*, July 2017.

[41] H. K. Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey, "Need for Speed: a benchmark for higher frame rate object tracking," in *2017 IEEE ICCV*, 2017, pp. 1134–1143.

[42] C.-M. Chiang, Y. Tseng, Y.-S. Xu, H.-K. Kuo, Y.-M. Tsai, G.-Y. Chen, K.-S. Tan, W.-T. Wang, Y.-C. Lin, S.-Y. R. Tseng, W.-S. Lin, C.-L. Yu, B. Shen, K. Kao, C.-M. Cheng, and H.-J. Chen, "Deploying image deblurring across mobile devices: A perspective of quality and latency," in *Proc. IEEE/CVF CVPR Workshops*, June 2020.

[43] S. Hussaini, M. Milford, and T. Fischer, "Applications of spiking neural networks in visual place recognition," *IEEE T-RO*, vol. 41, pp. 518–537, 2025.

[44] C.-T. Li, W.-C. Siu, and D. P. Lun, "Vision-based place recognition using ConvNet features and temporal correlation between consecutive frames," in *2019 IEEE ITSC*, 2019, pp. 3062–3067.