Binary Graph Descriptor for Robust Relocalization on Heterogeneous Data

Xi Wang, Marc Christie, Eric Marchand

***Abstract*— In this paper, we propose a novel binary graph descriptor to improve loop detection for visual SLAM systems. Our contribution is twofold: i) a graph embedding technique for generating binary descriptors which conserve both spatial and histogram information extracted from images; ii) a generic mean of combining multiple layers of heterogeneous data into the proposed binary graph descriptor, coupled with a matching and geometric checking method. We also introduce an implementation of our descriptor into an incremental Bag-of-Words (iBoW) structure that improves efficiency and scalability, and propose a method to interpret Deep Neural Network (DNN) results. We evaluate our system on synthetic and real datasets across different lighting and seasonal conditions. The proposed method outperforms state-of-the-art loop detection frameworks in terms of relocalization precision and computational performance, as well as displays high robustness against cross-condition datasets.**

***Index Terms*—Localization, SLAM, Loop closure**

I. Introduction

LOOP detection is a key module of modern SLAM pipelines. Its rationale is to eliminate the drifting problem during the SLAM process by retrieving already seen locations and taking them into account during the Bundle Adjustment procedure. Two characteristics are crucial: i) the capacity to retrieve already seen locations with high precision despite different views, lighting conditions and weather changes, and ii) the fast computing performance, which includes the detection of features from the current state and the query time within the database of existing locations. Mainstream SLAM loop closures techniques [1, 2] rely on the Bag-of-Words (BoW) or image retrieval methods with the help of low-level handcrafted features ([3, 4]) which are low weight, efficient to compute, and offer good compatibility with feature-based SLAM systems [5, 6].

With the rapid development of Deep Neural Networks (DNN), numerous works demonstrate robust performances on image retrieval and visual localization tasks, especially under extreme challenging conditions ($e.g.$ NetVLAD [7], DenseVLAD [8]). However, besides the GPU requirement, two main factors hinder a general application of the network descriptors into SLAM algorithms: i) facing the increasing scalability and fast query demand of the SLAM loop detection, many networks provide only exhaustive searching methods on the generated descriptors (while classic BoWs rely an inverted indexing scheme to accelerate queries); ii) though many neural networks present high generality towards different input formats, it is complicated to support heterogeneous image layers ($e.g.$ depth, semantics, lidar point clouds) from differently designed networks. Specialization on network structure and training process seems inevitable.

This paper proposes a generic Binary Graph (BiG) descriptor generated by a specific graph embedding technique on image regions. It improves loop detection with the heterogeneous image layers, including DNN outputs. The motivation in using a graph structure lies in its generic and spatial-aware representation to support various inputs and output binary descriptors for a BoW-based framework. Thus, the proposed method benefits from both sides: the image discrimination capacity of heterogeneous information and the accelerated query process of the BoW-based design.

Manuscript received: September, 9th, 2021; Revised November, 30th, 2021; Accepted December, 30th, 2021. This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers' comments.

Authors are with Univ Rennes, Inria, CNRS, Irisa, France. Email: xi.wang@inria.fr marc.christie, eric.marchand@irisa.fr

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

II. Related Work

Loop Detection Methods Mainstream loop closure methods can be divided into two categories: feature-based and image-based. The former group utilizes hand-crafted or learned local features for conducting loop detection ($e.g.$ [4], [9] and [10]), whereas the latter group works on a global approach by exploiting the whole or patched image information. Feature-based methods are widely employed in the feature-based SLAM and VIO systems ($e.g.$ [5, 6]) for the low computational cost and good compatibility when estimating camera poses. Another advantage of the local features lies in its robustness towards perspective warps, illuminative changes and dynamic environments [11, 12]. Following this intuition, a series of Bag-of-Words methods on hand-crafted features such as FabMap [1, 13] and DBoW [2] or learning-based features [14] are proposed and applied in various SLAM implementations [5, 6] for addressing the matching problem. Importantly, the usage of *inverted indexing* technique reduces the computational cost of the query process against the increasing image database whereas the naive image retrieval methods iterate candidates and yield cubic time complexity. The idea of the inverted indexing consists in storing the image index with the extracted word index to augment the filtering efficiency by pre-pruning the searching space. Later, [15] designed an incremental Bag-of-Words (iBoW) structure iBoW-lcd which, instead of training an offline vocabulary, can create an online vocabulary incrementally when the new binary features are added into the database. Another key factor which separates

loop detection tasks from image retrieval problems lies in its evaluation metric: loop detection methods often emphasize the precision over recall factor to avoid erroneous matches that would collapse the optimization. DBoW [2] and many following works [15] propose to apply a series of matching rules on the sequential groups of keyframes, (see [2] for more details) and improve the precision performance during the detection.

Binary Features Binary features display many benefits in computer vision tasks, most of which relate to their computing speed, robustness and good compatibility with the BoW structure. Improving binary descriptor systems w.r.t to their accuracy, speed, robustness even information compressibility performance is an active research topic [16]–[19].

Visual Localization and Image Retrieval: Entering the new era of Deep Neural Network (DNN), countless networks are proposed to handle the image retrieval task. These networks achieved astonishing results compared to the traditional ones ($e.g.$ NetVLAD [7], DenseVLAD [8]). A brief intuition of these networks consists in generating a descriptor of a given image which conserves a robust metric against environmental changes and perspective warps. However, in the SLAM context, except the requirement of GPU computation, another predicament lies in its naive yet heavily searching scheme influenced by the scale of the database. Another branch also investigates the topic of visual localization but under a 2D-3D scenario. The problem is usually described as re-finding the local descriptors from the query image in an already generated point cloud. Some common interests between image retrieval and 2D-3D localization are shared as several works propose to rely upon the image retrieval method as a preprocessing step to narrow the searching space for the 2D-3D localization algorithms [20].

Semantic Segmentation and Graph Embedding: Semantic segmentation information is defined as a partitioning towards multiple segments containing different semantic labels according to human understanding and annotations [21, 22]. Some recent works demonstrate that semantic segmentation can help the loop detection too. [23] exploits random walk graph embedding technique [24] on semantic segmentation images for achieving seasonal and viewpoint robust loop detection results under outdoor conditions. Larsson et al. proposed a fine-grained self-supervised segmentation network FGSN [25] demonstrating good performance in 2D-3D localization case. In FGSN, semantic labels are robust to environmental changes and learned in self-supervised fashion instead of extracted from human annotated labels.

III. Proposed Methods

A. Overview

The contributions of the proposed BiG descriptor are:

- a generic binary graph embedding technique for computing descriptors while conserving spatial relationship and considering textual information from heterogeneous formats by virtue of graph structure.
- a mean of combining multiple layers of information into our descriptor for improving the matching performance and robustness, coupled with a specific matching method and rough geometric checking scheme.

Besides, we propose an implementation of our multi-layered descriptor into an incremental BoW structure, to benefit from the inverted indexing technique that accelerates the query process and increases the scalability when facing a fast growing database of keyframes. Our descriptor also provides a generic way to exploit the inverted indexing on DNN outputs for fast loop detection tasks.

The pipeline to generate the BiG descriptor (see Fig. 1) is composed of the following modules, which we will detail in the subsequent sections: 1) graph structure generation; 2) deterministic graph embedding; 3) histogram generation and binarization; 4) support of heterogeneous image layers.

B. Graph Generation

The rationale in building an image descriptor through graph embedding is that the graph structure preserves relevant characteristics for loop closure tasks. Unlike keypoints frequency-driven BoW methods which ignores the spatial information, or the global methods ($e.g.$ GIST [26], CNN-based global descriptor for localization [27], etc.) prone to be influenced by image distortion and noise, the description of images by graph representations covers both local and global levels of information. The representation of image regions considers pixel-level content while the graph structure provides a comprehensive description of the image's spatial relations. Notably, the generality of graph structure representations enables a generic and extensible encoding of image characteristics which can encompass many different layers (depth, color, semantic, etc).

In this paper, our graph structure is generated from connected superpixelized regions ($e.g.$ SLIC [28]). Superpixels is an image segmentation approach focusing on the local spatial and chromatic similarities and primitive geometries such as compactness. One advantage in using superpixels (or similar methods such as semantic or object segmentation) stands in its repeatability over different images: the generated regions are less sensitive to scaling and slight wrapping distortion since the region's color differences are preserved at a certain level under different views. Therefore we built an unidirectional unweighted graph: $\mathbf{G} = (V, E)$ where V the vertices are in a set of regions in image I, and E presents their adjacency (we build an edge when two superpixel regions are adjacent in an 8-connected layout).

C. Neighbour Regions and Spatial Encoding

Different to Random Walk based stochastic graph embedding methods [24], we seek to design a deterministic graph embedding method: instead of visiting vertices in an arbitrary order, a deterministic method should output the same results when inputting the same graph and starting vertex.

Our proposed graph embedding method relies on the identification of neighbour vertices and regions. We define the k-th order neighbour vertices $N^k(V_c)$ of a given center vertex V_c as a set of vertices $\{V_i\}$ which have geodesic distance equal to k towards the center vertex V_c:

$$N^k(V_c) = \{V_i \in \{V\} : d(V_i, V_c) = k) \tag{1}$$

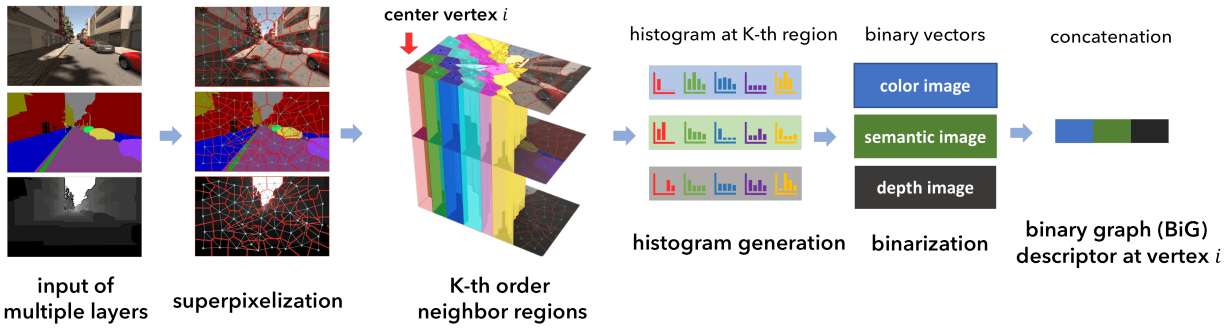

Fig. 1: The proposed pipeline to generate a BiG descriptor for a given superpixel (center vertex i). Multiple layers of heterogeneous image are structured in a graph representation spanning over neighboring regions of the superpixel, and embedding in a BiG descriptor through graph embedding and binarization of multiple layered histograms.

these neighbour vertices can be naively computed through a BFS (Breath-First-Search) search on the graph.

Once the definition of neighbour vertices is given, for a given center vertex, one is able to divide a graph into several united regions, in which each has a different neighbourhood distance $N^k(V_c)$ respectively. We term these order-related regions as $\Omega^k(V_c)$ (see Fig. 2).

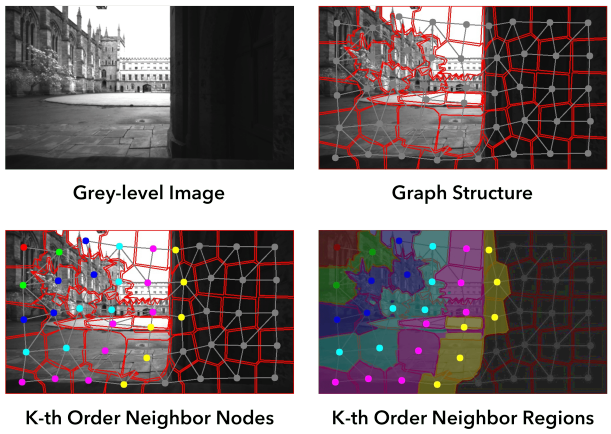

Fig. 2: A graph structure of superpixel connected regions is generated from an input image (left upper), red contour and gray nodes and vertices (right upper); given the center vertex of the red superpixel at left upper corner, we find k-th order neighbour vertices with different colors (red for $k = 0$, green for $k = 1$, etc.) (left lower) ; k-th order neighbour regions are highlighted in the image (right lower). Image from [29].

D. Histogram Generation

Image histogram, frequency statistics of a image, is prevalent in many robotics vision applications such as: image retrieval [30], visual servoing [31], and image registration [32] as it presents good characteristics on describing and manipulating image information. In this paper, we also rely on a histogram for depicting image information and retrieving similar images. Different to common applications computing histogram in whole image space, we rely on regional histograms to compute *localized neighboring histograms* which exploits the graph structure to express both spatial and frequential information.

Describing a histogram as a real vector of a given bins number N_b: $\mathbf{h} \in \mathbb{R}^{N_b}$, we calculate localized neighboring histograms in different k-th order neighbour regions $\Omega^k(V_c)$. Given a center vertex V_c, and its k-th order neighbour regions, a set of k histograms $\mathbf{h}^k(V_c)$ (one per region) can be obtained by computing independently within each regional area expressed as $\mathcal{H}^k(V_c)$:

$$\mathcal{H}^k(V_c) = \{\mathbf{h}^0(V_c), ..., \mathbf{h}^k(V_c)\} \tag{2}$$

E. Deterministic Graph Embedding and Binarization

We then propose a deterministic graph embedding technique (nodes are embedded in a deterministic order in contrast with Random Walk [24] embedding techniques). Given our k-th order neighbour region histogram in Eq. 2, we access graph information by extracting pairs of histograms in a combinatorial order: given K neighbors in total, we select combinations of C^2_{K+1} and sort them in ascending order: $(0, 1), ..., (0, K), (1, 2), ..., (K\text{-}1, K)$. The intuition of this operation consists in improving the discriminating capacity by computing a statistical difference and encoding spatial relationship into a binary vector in a deterministic order. Once the pairs of histograms are selected, we perform a binarization by comparing selected histograms w.r.t each bins b, similar to the BRIEF [3] operator:

$$\tau(h^i, h^j, b) := \begin{cases} 1 & \text{if } h^i(b) < h^j(b) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

We rewrite it into a more compact way for binarizing a pair of histograms of a given center vertex V_c:

$$\boldsymbol{\beta}^j_i(V_c) = \tau(\mathbf{h}^i(V_c), \mathbf{h}^j(V_c)) \tag{4}$$

where $\mathbf{h}^i(V_c)$ is i-th order regions histogram, same for j.

Finally, we generate a long binary sequence vector ($i.e.$ descriptor) $\mathbf{d}_c$ via concatenating the binarized histograms together in combinatorial order.

$$\mathbf{d}_c = \boldsymbol{\beta}^1_0(V_c) \oplus, ..., \oplus \boldsymbol{\beta}^K_{K-1}(V_c) \tag{5}$$

where the $\oplus$ is the concatenation operation, the length of our binary descriptor $\mathbf{d}_c$ is $N_d = N_b \times C^2_{K+1}$. We use a histogram

of 64-bins and 6 orders for neighbor regions, $\mathbf{d}_c$ has a size of $64 \times C_7^2 = 1344$ digits for each vertex.

Given each vertex V_c in graph, we can compute a $\mathbf{d}_c$ for describing its region and the relationship towards its vicinity. A series of binary descriptors $\{\mathbf{d}_c\}$ can therefore be generated for the whole graph ($i.e.$ the input image).

F. Generic Multiple Layer Descriptor

Applying the deterministic graph embedding has another advantage: each binary digit's order is determined and corresponds to its combination order given a graph structure, $i.e.$ its spatial relation. Therefore, we can easily concatenate the binary sequence descriptor with other layers of information as long as they share an identical graph structure.

For example, under the scenario of RGB-D camera, we can apply the previous process for extracting a descriptor $\mathbf{d}_c^g$ with a given center vertex V_c on a color image. Similarly, after transforming the depth and image sensors in a same coordinate, one can also compute the histograms of a depth image with the same graph regional segmentation from the color image and generate a descriptor $\mathbf{d}_c^d$ at the exact center vertex V_c. A concatenation operation helps to combine and generate an extended descriptor $\mathbf{d}_c^*$ for this multiple layer heterogeneous information (see Fig. 1).

$$\mathbf{d}_c^* = \mathbf{d}_c^g \oplus \mathbf{d}_c^d \tag{6}$$

The advantage of this design lies in its compactness and low cost for appending more layers. This provides a simple and fast way to fuse the heterogeneous data from different sources such as depth image, lidar data, semantic images, even the DNN results, though each type of data requires specific preprocessing before integrating into our descriptor.

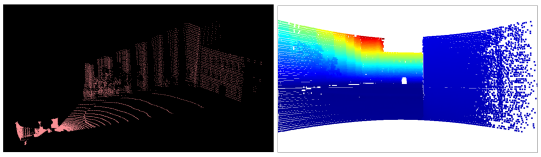

Fig. 3: We project point cloud data into 2D depth images along the camera frame coordinate. The blank arc regions are caused by the limited field-of-view of Lidar device.

Depth Image: For depth images acquired by depth sensors, we treat them like grey-level images to generate descriptors.
Lidar Information: Lidar (Light Detection and Ranging) sensors yield 3D point cloud rather than 2D images. Related work believe its difficulty consists in its sparse and three-dimensional nature. In this paper, we render the point cloud to images as the extrinsic calibration between the camera and Lidar sensor is known (see Fig. 3). Though the sparse nature of Lidar data raises problems for local descriptors, histograms remain an appropriate tool that ignores precise spatial information and yields statistical features. In other words, once we project point cloud to images, one can treat them as grey-level images for fitting into the system.

Traditional Semantic Image: For semantic images, the histogram shall not be built on pixels intensities but on labels. Therefore the length of the histogram is not arbitrary, as the quantity of labels is given and fixed. Following the idea of [23] and related work, we manually divide labels into two categories: static and dynamic. Static labels include buildings, plantations, etc, the dynamic ones are mostly non-permanent objects, such as pedestrians and vehicles.
FGSN Semantic Segmentations: FGSN [25] is a self-supervised semantic segmentation method which doesn't provide precisely the meaning for each learned label. Labels are learned for conserving high robustness against environmental variations. We treat FGSN as semantic types but without robust/unrobust separation.
Neural Network Results: Some DNN image retrieval methods rely on environmental invariant image segmentations such as the aforementioned FGSN [25], whereas the mainstream methods usually output directly descriptors of the whole image: $e.g.$ NetVLAD [7] and [8]. For fitting DNN results into our system, we generate respectively a descriptor for each region, and vectorize them directly as histogram.

G. Matching Descriptors and Geometric Checking

During the retrieval stage, namely the matching process, we need to compare and score two sets of descriptor vectors from the candidate and the query images respectively. Commonly, the simplest way is to rely on the Brute-Force Hamming distance matching approach, then average and normalize the computed Hamming distances on the matched descriptors to indicate the similarity between two images. However, unlike other well-defined metrics, a Hamming distance can rarely touch zero (it means inverted XOR digits i.e. two descriptors are inversely correlated), therefore causing an insufficient discrimination ability.

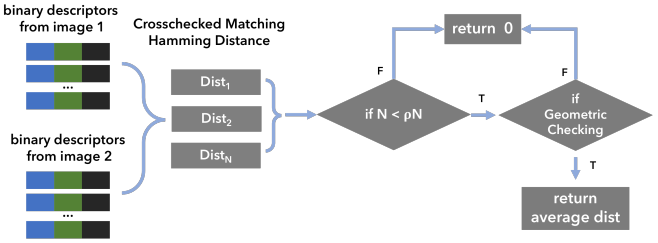

Fig. 4: A pipeline of the matching process of two descriptors. The result is controlled by two stages: it outputs zero (most dissimilar) when low matching number N than ρN after a crosschecked matching or high geometric checking error.

To fix this problem, we set a threshold ρ on the matching number to remove candidates of insufficient matches, as this often suggests multiple wrong matches colliding with each other. Another improvement is proposed at the geometric checking stage. Similar to keypoint BoW methods using fundamental matrix to compute and remove unqualified candidates, we estimate an affine constraint from the query image to the found candidates by computing the center coordinates of each region (regarding them as two sets of repeatable

keypoints from two images), and use the average reprojection error to filter candidates. This adds the extra cost in computing RANSAC but yields better recall performance. We set two thresholds to improve the discrimination ability by outputting 0 if the conditions are not satisfied. Finally a normalised similarity is generated between $[0, 1]$; see Fig. 4.

H. Implementation into iBoW Structure

Classic BoW structures require pre-trained offline vocabularies known beforehand, whereas our descriptors are generated on-the-fly and the format of binary digits varies with relation to different combinations of layers. We therefore expressed our binary descriptor into an incremental BoW system: iBoW-lcd[15] to achieve high precision, faster performance and sequential matching schemes in SLAM loop closure tasks.

IV. EXPERIMENTS

A. Datasets and Methodology

To demonstrate the capacity of our BiG descriptor, we relied on multiple public datasets : i) Synthia Dataset [33], a synthetic dataset across different environmental conditions; ii) Newer College Dataset [29], a real recorded dataset providing infrared images and Lidar data; iii) RobotCar Seasons Dataset [34, 35], a dataset across different weather and time conditions by a car-mounted camera.

B. Synthetic Data: Across Different Seasons

Synthia [33] is a synthetic car-driving dataset on multiple scenes, including various seasonal changes, day-night shift and weather variations. Besides color images, the dataset also provides ground truth semantic labels and depth images and is widely used in relocalization tasks with heterogeneous types of data [23]. We focus on the cross environmental image retrieval capacity. The idea is to retrieve every image from one condition to another: $e.g.$ from the night and rainy condition towards a sunshine one. PR-curves are generated by tuning acceptance similarity threshold on metrics.

We test our proposed method on different combinations of image layers such as Depth and Semantics (D+S). An ablation study on matching number removal threshold ρ (0.5 for our method) and geometric checking is also proposed. Comparison with related work includes i) DBoW [2]: a binary BoW method for SLAM loop detection; ii) FabMap [1] a probabilistic BoW method and iii) NetVLAD [7] a deep-learning method specifically designed for image retrieval under different conditions.

PR-Curve results are shown in Fig. 5. The proposed BiG descriptor handles well the image retrieval task under all conditions (with $\rho = 0.5$), specially the rain and night weather. NetVLAD method demonstrates good results under daylight and non-rainy conditions but our method reaches higher recall on rainy conditions. The improvements of the matching number removal threshold ρ and the geometric checking in BiG descriptor are obvious through the comparison between the first and the second row and the one between dot line and solid line in the figure. Two layers combination (D+S) shows the best working point (highest precision and recall) on two night conditions after applying ρ and geometric checking.

Methods	Precision	Recall	AUC	AP
FabMap 1 (11k voc)	0.477	0.510	0.280	0.404
FabMap 2 (4k voc)	0.528	0.452	0.399	0.418
DBoW (NoGC)	0.636	0.164	0.098	0.380
DBoW (GCDI2)	0.636	0.164	0.553	0.378
DBoW (GCExhaustive)	1.000	0.317	0.750	0.831
iBoW-lcd (1k ORB)	1.000	0.522	0.833	0.744
Ours: IR + Lidar	1.000	0.298	0.730	0.804
Ours: IR + Semantic DpLabv3+	1.000	0.471	**0.849**	**0.856**
Ours: IR + Semantic FGSN	**1.000**	**0.538**	**0,847**	**0.808**
Ours: IR + NetVLAD	**1.000**	**0.567**	0.788	**0.901**
Ours: FGSN + NetVLAD	**1.000**	**0.588**	**0.860**	**0.900**

TABLE I: Table of working point (Precision, Recall), AUC and the average precision (AP) for each method tested in Newer College Dataset [29]. Our proposed methods has the highest recall with the combination of infrared image and NetVLAD DNN descriptor and good performance on AUC and AP along all the recall band.

C. Newer College Dataset: under Static Environment

We then test our method under a static scenario, similar to traditional use-cases in loop detection systems. The Newer College dataset [29] provides stereo infrared images with point clouds acquired by the Lidar device in an asynchronised approach. After manually synchronising the point cloud data with infrared data, we extracted 505 images with maximum distance of at least 300 meters. We apply a distance threshold of 25 meters and an orientation threshold of 25 degrees for generating all ground truth loop closure event frames. Following [2], we set the neighbor ground truth loop closure event as "recalled" when a true positive is found close enough (2 frames). This avoids a too low recall caused by the inter-competing of the candidates ($e.g.$ an outstanding candidate may compete and attract the true recalls of its neighbors lead to lower recall). Five methods are compared: our method implemented in the iBoW-lcd, iBoW-lcd, DBoW and FabMap (1 and 2 on differently trained vocabularies).

We project the lidar point clouds into depth images to adapt to the BiG descriptor according to its extrinsic calibration data (see Fig. 3). Due to the limited Lidar field-of-view, some undetected regions exist on the depth image. We tested combinations including infrared image (IR), Lidar, NetVLAD [7], and two semantic data (generated by DeepLabv3+ [22] and FGSN [25]) with different pre-processing stages proposed in Section III-F.

For the DBoW [2], three geometric checking configurations are evaluated: without checking (NoGC), with a checking on node DI2 (GCDI2), and with an exhaustive checking (GCExhaustive). We tested FabMap1 [1] on an $11k$ vocabulary from the paper and train a $4k$ vocabulary on a similar dataset for the FabMap2 [13].

The results are shown as the best working point (highest precision point when the recall is non-zero), AUC (Area-under-Curve) and average precision of sampling points in the Table. I, we observe that all configurations of our method can yield a working point with precision at 1.0 and relative high recall level. DBoW with geometric checking shows similar results over the combination of infra image and lidar projected depth image. iBoW-lcd has better results except our

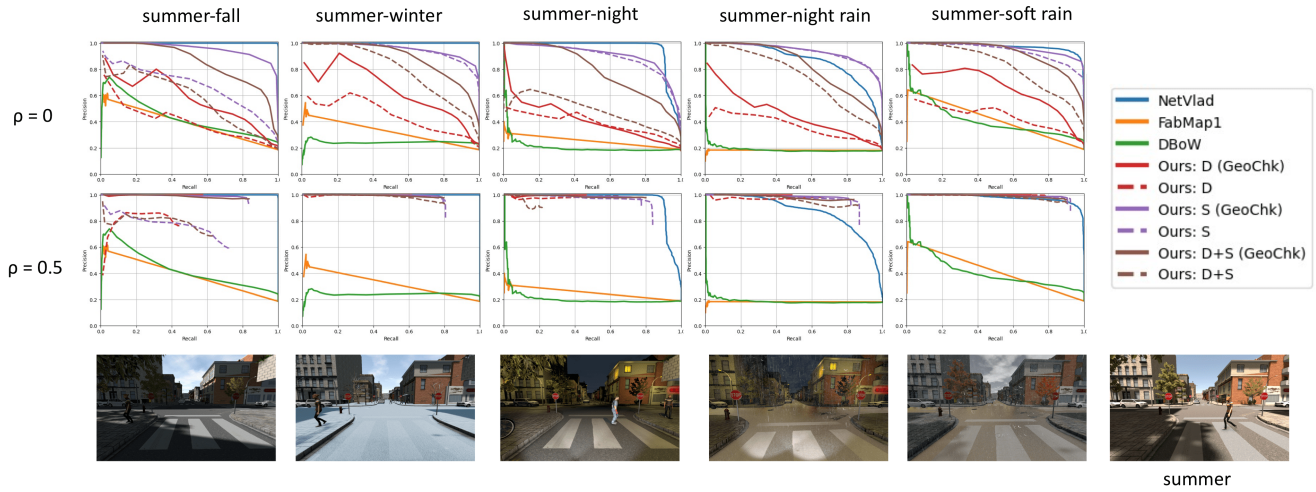

Fig. 5: The PR-Curves (x,y for recall and precision) from Synthia Dataset [33] demonstrate that our BiG descriptor outperforms DBoW and FabMap on all conditions and show better results against NetVLAD under rainy weather conditions. The improvement of ρ and geometric checking can also be observed in the comparison between first and second row, dot line and solid line. In the legend, D and S are Depth and Semantic layers respectively, D+S is the combination of the two.

methods, however by combining with NetVLAD and semantic information, our methods outperform all others.

D. RobotCar Season Dataset: under Changing Conditions

RobotCar Seasons dataset is a dataset captured by car-mounted cameras across different seasons (summer, winter), weathers (rainy, sunny, snowing), and times (dawn, dusk, night). The reference includes around $7k$ images taken under the overcast condition (oc), and various query conditions about 250 images each for different conditions. Given its challenging nature, many works exploit this dataset for robustness experiments [25, 34, 36]. In this paper, we use color images for image retrieval and loop detection tests.

We perform two categories of experiments in this section: i) image retrieval experiment: following the methodology of experiments in [25, 34, 37], we organize the experiment as follows: relying on the proposed method, we apply image retrieval experiment from the query conditions towards the reference condition for all images. The measure is defined as the percentage of images from the query dataset have been relocalized under certain distance and angle thresholds. The purpose of this experiment is to demonstrate the discriminative capacity of the proposed BiG descriptor with the combination of various layers of information such as FGSN [25] semantic labels and NetVLAD [7] DNN outputs.

Image Retrieval Experiment: In Table. II, we test two types of combinations of layers: FGSN as single layer (termed BiG FGSN in the table) and the FGSN plus NetVLAD as two layers (termed BiG F+N in the table). With three versions tested for each combination: i) we directly take the first ranked candidate as the output of the query result; ii) we perform an extra geometric checking with the help of SIFT [4] (S in the table) keypoints within the top 10 ranked candidates and output the final candidate according to the geometric checking results; iii) the process is similar to the second one, instead of using SIFT keypoints, we choose a DL keypoint D2-Net [10] (D2 in the table) which demonstrates better robustness against environmental variations.

From the Table. II we achieve some observations: i) the geometric checking improves for all methods: D2-Net demonstrates better performance under difficult scenarios whereas the SIFT feature works well under normal conditions; ii) our methods maintain good performance on daylight conditions, though the gain is slightly negative ($<1\%$) for few, the average result outperformed the geometric checking version NetVLAD (See Table. III); iii) the proposed method shows drastic improvements ($\sim 45\%$) on night conditions and proves the robustness and discriminative capacity; iv) the combination of two layers improves the performance during the daylight but lowers for the night ones, due to the bad performance of NetVLAD during night.

In Table. III, we list the average of the best performance of our methods with two combinations and two geometric checking methods, the NetVLAD method with and without geometric checking and other 2D-2D (image retrieval) methods. The proposed method outperforms all the image retrieval methods, and especially gained drastic performance under night condition even compared to the specific designed DNN method for dark environment [36] and the state-of-the-art 2D-3D relocalization methods. In 2D-3D methods, the relocalization is achieved in an already built 3D point cloud map as a priori. Therefore the difficulty of the retrieval problem is eased and better results can be achieved numerically. Still our proposed 2D-2D method shows a gain of at least 30% even compared to the constraint relaxed 2D-3D methods in night conditions. More specifically, the FGSN [25] is exactly the semantic labels used for our proposed method as our method is 35% better on results when using the FGSN as the only layer. The improvements on the identical input data shows the efficiency of our binary graph descriptor.

Loop Closure Experiment: The second type of experiment aims at testing the loop closure ability of the proposed method by integrating BiG descriptor into an incremental bag-of-words

	day conditions							night conditions	
	sun 0.25/0.5/5 m 2/5/10 deg	oc-summer 0.25/0.5/5 m 2/5/10 deg	oc-winter 0.25/0.5/5 m 2/5/10 deg	dusk 0.25/0.5/5 m 2/5/10 deg	dawn 0.25/0.5/5 m 2/5/10 deg	rain 0.25/0.5/5 m 2/5/10 deg	snow 0.25/0.5/5 m 2/5/10 deg	Night 0.25/0.5/5 m 2/5/10 deg	NightRain 0.25/0.5/5 m 2/5/10 deg
NetVLAD	2,87/12,44/85,17	0,04/25,91/95,91	2,97/19,31/86,63	13,37/37,97/90,37	3,48/17,39/80,43	7,58/25,25/93,43	6,69/24,27/90,38	0,51/2,03/15,23	0,89/3,57/24,55
NetVLAD (S)	3,35/15,31/85,65	5,91/35,45/97,27	8,42/35,64/89,11	18,18/43,85/92,51	7,39/27,83/83,48	10,10/32,32/93,43	10,88/28,87/91,63	0,51/1,52/16,75	0,89/4,46/21,43
NetVLAD (D2)	2,87/15,31/90,43	5,00/36,36/97,73	6,44/34,65/91,58	17,65/42,25/90,91	6,96/25,65/84,78	9,09/27,78/93,94	8,79/29,29/91,63	1,52/3,55/28,93	1,79/7,59/42,86
BiG FGSN	1,44/9,57/71,77	2,73/15,00/81,82	1,98/18,32/79,21	9,63/24,60/81,82	3,04/13,04/79,13	6,57/17,17/88,38	3,35/13,39/78,24	3,55/10,15/58,88	2,23/6,25/63,84
BiG FGSN (S)	3,35/16,27/81,34	6,36/29,09/95,45	7,92/33,17/87,62	17,11/43,32/90,37	6,96/28,70/86,52	10,10/32,83/92,93	9,62/27,20/89,54	1,52/4,57/44,67	2,23/5,80/51,79
BiG FGSN (D2)	5,26/17,70/89,47	4,55/31,36/95,45	6,44/38,12/92,08	16,58/40,11/90,37	6,52/27,83/87,39	10,10/29,29/92,93	7,95/28,03/91,21	3,55/10,66/76,14	3,57/13,84/86,16
BiG F+N	3,35/13,40/76,08	3,18/17,27/86,82	3,47/21,78/81,68	10,70/26,74/88,24	5,65/18,70/83,04	8,08/20,71/90,40	4,18/16,32/84,10	2,03/6,09/44,67	3,13/10,27/66,07
BiG F+N (S)	3,35/15,79/81,82	6,36/34,09/97,27	7,92/33,66/89,11	16,58/42,78/91,98	7,83/30,43/86,96	10,10/32,83/94,95	8,79/26,36/87,87	0,51/2,54/35,03	2,23/7,14/48,66
BiG F+N (D2)	4,31/19,14/90,43	5,00/29,55/96,82	6,93/35,64/92,57	16,58/41,18/91,44	7,83/29,57/87,39	10,10/28,79/93,94	7,11/28,03/89,12	1,02/7,11/74,62	4,46/17,41/84,82
Gain from NetVLAD	**1,44/3,83/0,00**	1,36/-2,27/-0,45	**1,93/-0,72/0,99**	-1,60/-1,07/-0,53	**0,87/3,91/2,61**	**0,0/0,51/1,52**	-0,84/-1,26/-0,42	**2,03/7,11/47,21**	**1,79/6,25/43,30**

TABLE II: Comparison of our method of one layer: FGSN semantic (BiG FGSN) and two layers: semantic (F) plus NetVLAD (N) (BiG F+N) and original NetVLAD on RobotCar Dataset [34]. Three versions are applied on each method: i) best ranking candidate; ii) SIFT geometric checking on top 10 ranking candidates (S); iii) D2-Net geometric checking on top 10 ranking candidates (D2). The gains of the best results of ours against the best of NetVLAD are in last row.

		mean daylight .25 / .50 / 5.0 2 / 5 / 10	mean night .25 / .50 / 5.0 2 / 5 / 10
	m deg		
2D	Our method	**8.58 / 31.73 / 92.26**	**3.56 / 12.25 / 81.15**
2D	NetVLAD [7]	5.29 / 23.22 / 88.90	0.70 / 2.80 /19.89
2D	NetVLAD (GC)	7.98 / 30.66 / 91.80	1.65 / 5.57 / 35.90
2D	DenseVLAD [8]	7.71 / 31.26 / 92.26	1.00 / 4.45 / 22.70
2D	FabMap [1]	2.80 / 12.34 / 30.37	0.00 / 0.00 / 0.00
2D	SeqSLAM [38]	1.30 / 6.10 / 15,30	0.20 / 0.70 / 1.50
2D	ToDayGAN [36]	-	2.15 / 11.00 / 50.20
3D	FGSN [25]	-	11.00 / 28.40 / 45.20
3D	DomainAdapt [39]	-	20.65 / 42.50 / 52.15

TABLE III: Comparison on average relocalization rate of our method with mainstream 2D image retrieval methods and 3D relocalization methods on RobotCar Dataset [34], where GC refers to the geometric checking.

loop detection framework. The motivation is to demonstrate the proposed descriptor can complete loop closure tasks under difficult scenarios with high precision, fast speed and good compatibility with all iBoW systems and the ability of combining heterogeneous information.

The methodology is similar to Sec. IV-C, we concatenate all images of reference condition and query images of various conditions respectively for inputting into a loop detection system sequentially. We mark the working point of highest precision and recall as final results. In this experiment we compare to the original iBoW-lcd [15] method which utilizes 1k ORB [9] as feature to build BoW and DBoW [2].

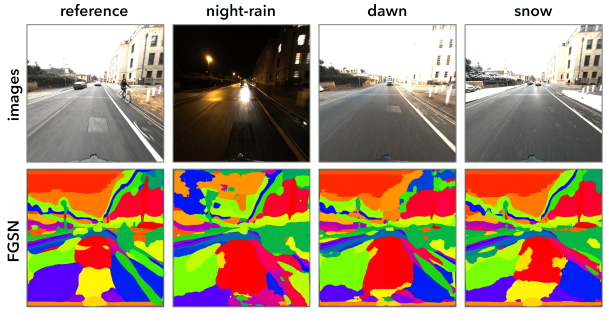

Fig. 6: The retrieval results from reference to various environmental conditions of RobotCar Dataset [34], our method retrieves seen locations despite appearance disparities by the spatial similarity w.r.t semantic distribution on FGSN images.

After integrating the BiG descriptor with multiple layers into an iBoW framework, the proposed method outperforms the original iBoW-lcd in both daylight and night cases with higher recalls and similar yet high precisions (See Table. IV). The iBoW-lcd generates better performance under well lit overcast condition: oc-summer but lacks robustness against environmental noises in sun, dawn and nights conditions.

E. Computational Efficiency

As we mentioned in related works, the binary descriptor displays a good compatibility with BoW techniques not only for improving the retrieval ability but also for achieving a faster query supported by the inverted indexing technique.

In the Table. V, we measure our speed performance on a 2.7GHz Intel i7, as well as the iBoW-lcd, FabMap and DBoW. The other results are taken from [34]. We can see the proposed method only takes 73.5 ms whereas the other methods are more time consuming except FabMap. NetVLAD and DenseVLAD data are taken from [34] with a Intel Xeon 2.6GHz. The explanation for the difference between our method and iBoW-lcd lies in the different descriptor number: we only generate about 50 descriptors for each image (one per region), but iBoW-lcd uses 1k ORB features. Therefore the proposed descriptor shows another advantage: converting the DNN into binary descriptors for accelerating the query time when facing large scale datasets. See our supplementary material for more analysis [40].

V. Conclusion

In conclusion, we proposed a binary graph descriptor which is able to: i) encode image content and spatial information from a graph structure through the design of a graph embedding method; ii) combine heterogeneous layers of information such as semantic images or neural network results to gain better retrieval capacity under dynamic environments. Besides, by relying on an incremental Bag-of-Words structure, the proposed method achieves real-time performance for SLAM loop detection tasks on large datasets.

References

[1] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The Int. J. of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

	sun	oc-summer	oc-winter	dusk	dawn	rain	snow	night	night-rain
BiG FGSN	**36.3/85.4**	36.8/97.6	50.5/97.1	58.3/90.1	**76.5/93.1**	83.8/96.5	39.8/94.1	**38.6/83.5**	**65.2/75.3**
BiG F+S	30.6/85.3	40.0/97.8	**58.9/98.4**	**78.1/95.4**	71.3/91.6	**88.9/99.4**	**56.6/99.1**	18.3/66.7	26.8/72.3
iBoW-lcd	8.6/89.5	**55.9/99.1**	53.9/99.1	75.4/94.0	33.9/89.7	87.9/99.3	40.6/100	0.00/0.00	0.00/0.00
DBoW	2.8/100	9.55/91.3	13.9/100	45.4/97.7	22.6/98.1	49.0/100	23.6/96.5	0.00/0.00	0.00/0.00

TABLE IV: Comparison of Recall/Precision on different conditions of RobotCar Dataset [35]. we take 5.0 m and 10 deg as threshold for computing results, the best performances are highlighted as the maximum sum of precision plus recall.

methods	Ours	iBoW-lcd	FabMap	DBoW	NetVLAD*	DenseVLAD*
time (ms)	73.5	715.2	57.1	262.5	137	338

TABLE V: Comparison of average query time on the RobotCar Season Dataset per image. *: data from [34].

[2] D. Gálvez-López and J. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. on Robotics*, vol. 28, no. 5, pp. 1188–1197, October 2012.

[3] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *European Conf. on Computer Vision*, 2010, pp. 778–792.

[4] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[5] R. Mur-Artal, J. Montiel, and J. Tards, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Trans. on Robotics*, vol. 31, no. 5, pp. 1147–1163, Oct 2015.

[6] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[7] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.

[8] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, pp. 257–271, Feb. 2018.

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

[10] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-net: A trainable cnn for joint description and detection of local features," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 8092–8101.

[11] X. Wang, M. Christie, and E. Marchand, "Optimized contrast enhancements to improve robustness of visual tracking in a slam relocalisation context," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2018, pp. 103–108.

[12] G. Yu and J.-M. Morel, "Asift: An algorithm for fully affine invariant comparison," *Image Processing On Line*, vol. 1, pp. 11–38, 2011.

[13] M. Cummins and P. Newman, "Appearance-only slam at large scale with fab-map 2.0," *The Int. J. of Robotics Research*, vol. 30, no. 9, pp. 1100–1123, 2011.

[14] D. Li, X. Shi, Q. Long, S. Liu, W. Yang, F. Wang, Q. Wei, and F. Qiao, "Dxslam: A robust and efficient visual slam system with deep features," in *2020 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. IEEE, pp. 4958–4965.

[15] E. Garcia-Fidalgo and A. Ortiz, "iBoW-LCD: An appearance-based loop-closure detection approach using incremental bags of binary words," *IEEE Robotics and Automation Letters*, 2018.

[16] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Computer Vision – ECCV 2008*, D. Forsyth, P. Torr, and A. Zisserman, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 304–317.

[17] D. Schlegel and G. Grisetti, "Adding cues to binary feature descriptors for visual place recognition," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 5488–5494.

[18] ——, "Hbst: A hamming distance embedding binary search tree for feature-based visual place recognition," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3741–3748, 2018.

[19] T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit, "Boosting binary keypoint descriptors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2874–2881.

[20] T. Sattler, M. Havlena, K. Schindler, and M. Pollefeys, "Large-scale location recognition and the geometric burstiness problem," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.

[21] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[22] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Eur. Conf. on Computer Vision*, 2018.

[23] A. Gawel, C. Del Don, R. Siegwart, J. Nieto, and C. Cadena, "X-view: Graph-based semantic multi-view localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1687–1694, 2018.

[24] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2014, pp. 701–710.

[25] M. Larsson, E. Stenborg, C. Toft, L. Hammarstrand, T. Sattler, and F. Kahl, "Fine-grained segmentation networks: Self-supervised segmentation for improved long-term visual localization," in *IEEE/CVF Int. Conf. on Computer Vision*, 2019, pp. 31–41.

[26] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid, "Evaluation of gist descriptors for web-scale image search," ser. CIVR '09. NY, USA: Association for Computing Machinery, 2009.

[27] T. Naseer, W. Burgard, and C. Stachniss, "Robust visual localization across seasons," *IEEE Trans. on Robotics*, 2018.

[28] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[29] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon, "The newer college dataset: Handheld lidar, inertial and vision with ground truth," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2020.

[30] R. Brunelli and O. Mich, "Histograms analysis for image retrieval," *Pattern Recognition*, vol. 34, no. 8, pp. 1625–1637, 2001.

[31] Q. Bateux and E. Marchand, "Histograms-based visual servoing," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 80–87, 2016.

[32] A. Dame and E. Marchand, "Second-order optimization of mutual information for real-time image registration," *IEEE Trans. on Image Processing*, vol. 21, no. 9, pp. 4190–4203, Sept 2012.

[33] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. Lopez, "The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2016.

[34] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic *et al.*, "Benchmarking 6dof outdoor visual localization in changing conditions," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018.

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

[36] A. Anoosheh, T. Sattler, R. Timofte, M. Pollefeys, and L. Van Gool, "Night-to-day image translation for retrieval-based localization," arXiv 1809.09767, 2019.

[37] M. Larsson, E. Stenborg, L. Hammarstrand, M. Pollefeys, T. Sattler, and F. Kahl, "A cross-season correspondence dataset for robust semantic segmentation," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 9532–9542.

[38] M. Milford and G. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *IEEE Int. Conf. on Robotics and Automation*, 2012, pp. 1643–1649.

[39] S. Baik, H. J. Kim, T. Shen, E. Ilg, K. M. Lee, and C. Sweeney, "Domain adaptation of learned featuresfor visual localization," in *BMVC*, 2020.

[40] X. Wang, M. Christie, and E. Marchand, "Supplementary Material: Binary Graph Descriptor for Robust Relocalization on Heterogeneous Data," Nov. 2021, preprint. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03442119