2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

SPOT: Point Cloud Based Stereo Visual Place Recognition for Similar and Opposing Viewpoints

Spencer Carmichael[1], Rahul Agrawal[1], Ram Vasudevan[2], and Katherine A. Skinner[1]

***Abstract*— Recognizing places from an opposing viewpoint during a return trip is a common experience for human drivers. However, the analogous robotics capability, visual place recognition (VPR) with limited field of view cameras under 180 degree rotations, has proven to be challenging to achieve. To address this problem, this paper presents Same Place Opposing Trajectory (SPOT), a technique for opposing viewpoint VPR that relies exclusively on structure estimated through stereo visual odometry (VO). The method extends recent advances in lidar descriptors and utilizes a novel double (similar and opposing) distance matrix sequence matching method. We evaluate SPOT on a publicly available dataset with 6.7-7.6 km routes driven in similar and opposing directions under various lighting conditions. The proposed algorithm demonstrates remarkable improvement over the state-of-the-art, achieving up to 91.7% recall at 100% precision in opposing viewpoint cases, while requiring less storage than all baselines tested and running faster than all but one. Moreover, the proposed method assumes no *a priori* knowledge of whether the viewpoint is similar or opposing, and also demonstrates competitive performance in similar viewpoint cases.**

I. Introduction

Visual place recognition (VPR) is an important capability in robotics that supports loop closure, relocalization, and multimap merging [2], [3], [4]. VPR is challenging because algorithms must avoid false matches between different places with simliar visual appearance while also recognizing revisited places despite possible changes in illumination, viewpoint, weather, seasons, and arrangement of objects (e.g., cars) in the scene. While state-of-the-art methods demonstrate impressive performance in many scenarios [5], [6], [7], the extreme case of opposing viewpoint VPR is particularly difficult and relatively understudied [7], [8], [9], [10], [11], [12].

Opposing viewpoint VPR involves using cameras with a limited field-of-view (FOV) to perform place recognition under 180 degree rotations. Robustness to 180 degree rotations has been demonstrated in place recognition methods using omnidirectional cameras [13] and lidars [14], and 360 degree views fused from multiple cameras [15], [16]. However, such engineering solutions are not without drawbacks. Given the same bandwidth, limited FOV cameras offer higher spatial resolution than omnidirectional cameras, and adding additional cameras increases power consumption, bandwidth, and computing requirements. Opposing viewpoint VPR also offers a fallback in cases where the opposite view is obstructed, such as when a robot is towing a large object or producing a trailing dust cloud. Moreover, existing papers on opposing viewpoint VPR argue that the human capability to solve this problem lends it inherent scientific value [9], [10], [11], [12].

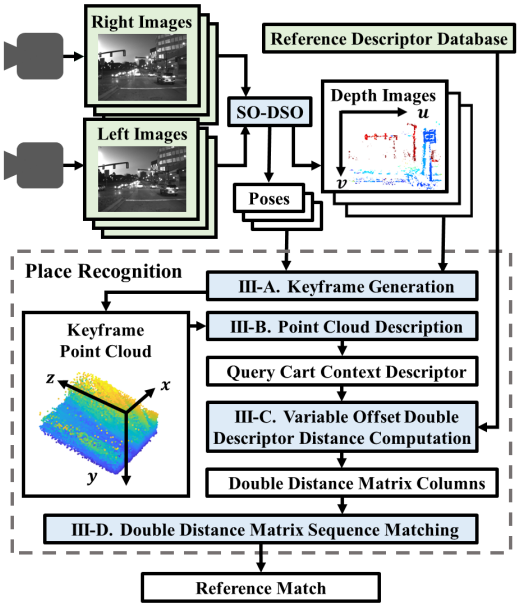

Fig. 1. Overview of SPOT. External inputs are green, processing blocks are blue, outputs are white. The stereo VO algorithm, SO-DSO [29], processes stereo images and outputs estimated poses and scaled depth images. This output is accumulated to form a point cloud at a selected keyframe pose. Next, a Cart Context [14] query descriptor is formed. Two distances between the query and all references are computed and a novel double distance matrix sequence matching scheme produces the final reference match.

Motivated by the relative invariance of structure to changes in illumination, weather, and seasons, several recent papers have used accumulated point clouds from visual odometry (VO) for VPR rather than [17], [18], [19] or in addition to [19], [20] using appearance. In the case of stereo VO, the estimated point clouds have absolute scale and lidar descriptors can be directly applied [19]. While these recent papers have demonstrated promising improvement under appearance changes, prior work leveraging VO-derived structure has not explored the opposing viewpoint case.

This paper introduces an opposing viewpoint VPR method that uses limited FOV stereo cameras and relies exclusively on structure estimated through stereo VO (Fig. 1). We demonstrate our method's remarkable improvement over

This work was supported by a grant from Ford Motor Company via the Ford-UM Alliance under award N028603.

[1]S. Carmichael, R. Agrawal, and K. A. Skinner are with the Department of Robotics, University of Michigan, Ann Arbor, MI 48109 USA {specarmi,rahulagr,kskin}@umich.edu

[2]R. Vasudevan is with the Department of Robotics and the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109 USA ramv@umich.edu

the state-of-the-art on the publicly available Novel Sensors for Autonomous Vehicle Perception (NSAVP) dataset [21]. Using a 15 meter localization radius to determine correct matches, our method achieves up to 91.7% recall at 100% precision in opposing viewpoint cases while none of the tested baselines exceed 0.2%. Moreover, our method assumes no *a priori* knowledge of whether the viewpoint is similar or opposing, and also demonstrates competitive performance in similar viewpoint cases. Beyond place recognition performance, we show that our method requires less storage than all baselines tested and runs faster than all but one. The code is available through the project webpage [1].

II. Related Work

A. Similar Viewpoint VPR

Place recognition typically proceeds in three stages: a query descriptor is formed, descriptor distances are computed between the query and multiple references, and finally a matching algorithm selects a reference and produces a score. Early VPR techniques used handcrafted local or global image descriptors [3], with methods using local descriptors often employing aggregation schemes such as vector of locally aggregated descriptors (VLAD) [22]. More recently, VPR has trended towards deep learning-based approaches [4], [23]. Inspiration from earlier techniques has led to methods such as NetVLAD [5], wherein the output of a CNN is interpreted as dense local descriptors and VLAD is mimicked with a pooling layer. Deep learning-based methods typically demonstrate better performance at the cost of greater computational requirements [24], [4].

Throughout the literature, VPR is often cast as pure image retrieval [3]. However, in robotics applications such as SLAM, information relating places is often known and can be exploited [3], [23]. For instance, SeqSLAM searches for sequences of query-reference pairs with the knowledge that revisiting an area in the same direction will produce queries in the same order as their matching references [25]. SMART additionally exploits odometry to ensure equi-spaced descriptor formation, improving the search for sequences [26], [27].

Recent methods have gone further and leveraged VO-derived structure [17], [18], [19], [20], effectively performing video retrieval [28]. Most similar to our work, [19] accumulates the output of SO-DSO [29], a stereo extension of DSO [30], and directly applies lidar descriptors for place recognition. Among the lidar descriptors tested, Scan Context [31] demonstrated the best performance and computational efficiency [19]. Although Scan Context is invariant to rotations [31], its potential for opposing viewpoint VPR was not investigated [19].

Inspired by [19], we similarly utilize SO-DSO [29] to estimate structure and apply an updated version of Scan Context, called Cart Context [14], for place description. To further enhance performance, we introduce an improved Cart Context distance metric and a novel double distance matrix sequence matching method, and use pose estimates to ensure equi-spaced descriptors as in SMART [26], [27].

B. Opposing Viewpoint VPR

Opposing viewpoint VPR methods have mainly involved leveraging semantic information with single-image descriptors [10], [9], [11], [12]. In [10], the Local Semantic Tensor (LoST) descriptor was introduced, which is built from both the final output of a dense semantic segmentation network and the output of one of its intermediate layers (conv5). The full method, LoST-X, adds a step to filter the top nearest neighbor candidates by again utilizing the semantic predictions and conv5 features [10]. Improved variations of the LoST-X framework were presented in two subsequent papers [11], [12]. In [11], LoST is replaced by NetVLAD and the fine matching stage is updated to consider candidate reference *sequences* and to leverage learning-based single-view depth estimates [11]. This method, sequence-to-single (Seq2single), uses depth estimates only to filter keypoints [11], unlike our method where depth estimates form the basis of the place description. Most recently, AnyLoc has demonstrated universal place recognition, encompassing diverse viewpoints, with single-image descriptors formed by aggregating features from a foundation model [7]. However, AnyLoc's opposing viewpoint capability was not evaluated outdoors or against methods specific to this task [7].

In approaches with single-image descriptors, matches are often separated by a *visual offset*: the physical distance between two cameras with opposing viewpoints at which the visual overlap between them is maximized [12]. This distance, estimated to be 30-40 meters in a driving scenario [12], may be difficult to overcome in applications where VPR must be followed by precise relative pose estimation (e.g., loop closure in SLAM). Better localization accuracy has been shown with semantic graphs built across multiple images, but this was only demonstrated with short ($<$1 km) sequences [8]. Existing methods also suffer from additional drawbacks, such as required *a priori* knowledge of whether the viewpoint is similar or opposing [10], [12], high storage demands [10], [11], [12], and the resource requirements imposed by deep neural networks [7], [8], [9], [10], [11], [12]. Our proposed method, SPOT, avoids the visual offset problem by forming place descriptors that capture the structure *surrounding* the camera rather than being limited to the FOV of a single image. SPOT is also fully handcrafted, runs quickly on a CPU, requires no *a priori* knowledge of whether the viewpoint is similar or opposing, and requires relatively little storage for the reference database.

III. Technical Approach

Figure 1 shows an overview of the full SPOT system, which takes as input stereo images and outputs place recognition matches at selected keyframes. The entry point of the system is a stereo VO algorithm, which outputs poses and sparse depth images with absolute scale. Any stereo VO algorithm could be used for this purpose. We choose SO-DSO [29] due to its demonstrated success at generating point clouds useful for VPR [19]. The following subsections describe the remaining stages of the algorithm. Figure 2 provides an expanded depiction of the final three stages.

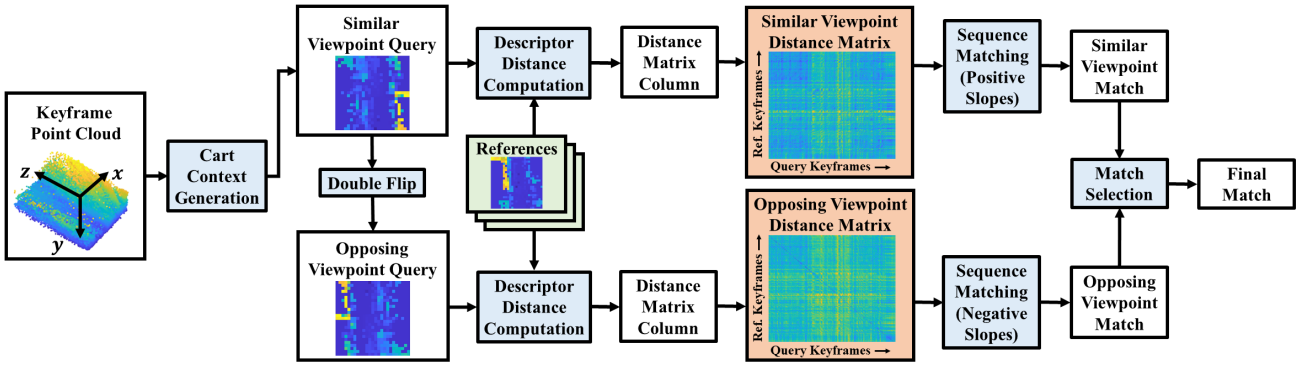

Fig. 2. An expanded depiction of the final three place recognition stages. External inputs are green, processing blocks are blue, outputs are white, and incrementally updated objects are orange. A Cart Context [14] descriptor is formed from the most recent keyframe point cloud and flipped about both axes to produce an additional descriptor for opposing viewpoint VPR. Descriptor distances are computed between each query and every reference to produce two separate distance matrices for similar and opposing viewpoints. Sequence matching, as described in [25], is performed separately in each distance matrix and the final output is selected from the results.

A. Keyframe Generation

The keyframe generation component serves to accumulate the output of SO-DSO, select when a keyframe should be created, and generate point clouds for Cart Context [14] description. Stereo triangulation error increases rapidly with depth [32], so as an initial step to reject noisy points, we discard all pixels of the depth image exceeding a given threshold, r_d. Each depth image from SO-DSO has a corresponding pose estimate. We use this pose estimate and the known intrinsics of the left camera to project each valid pixel of the depth image into a common world frame. We continuously accumulate projected points from each new depth image into a single point cloud. As in SMART [27], [26], we aim to create place descriptors at a constant distance apart to aid subsequent sequence matching. To do this, we compute the path distance, i.e., the sum of Euclidean distances between SO-DSO position estimates, since the last keyframe. When this path distance exceeds the desired descriptor spacing, s, we produce a new keyframe point cloud centered at the current pose. To create a new keyframe point cloud, we transform the accumulated point cloud into the current camera frame and select all points residing within a horizontal radius r_k about the camera position. Every time a new keyframe is generated, we cull faraway points in the accumulated point cloud using a second, larger horizontal radius, r_a. This culling improves efficiency and accounts for large-scale odometry drift. We do not create the first keyframe point cloud until the path distance exceeds $1.5r_k$ in an effort to ensure the point cloud is well-populated.

B. Point Cloud Description

For each keyframe point cloud, a descriptor is formed. As the depth estimates obtained through SO-DSO have absolute scale, it is possible to directly apply a lidar descriptor. We choose to use the Cart Context descriptor introduced in [14] as it is highly efficient to compute and demonstrates impressive performance in lidar-based place recognition.

The Cart Context descriptor is created from the keyframe points lying in a $2r_{lo} \times 2r_{la}$ meter horizontal rectangle centered at the origin of the camera frame, with the $2r_{lo}$ meter side aligned with the longitudinal direction (or forward direction, z) and the $2r_{la}$ meter side aligned with the lateral direction (x). The rectangular domain is divided into m equal-sized rows along the longitudinal axis and n equal-sized columns along the lateral axis to create bins. The maximum height above the ground of the points captured within each bin is recorded to obtain the $m \times n$ Cart Context descriptor. If no point exists within a bin, the corresponding value in the descriptor is set to zero. The height above the ground of a point, $\mathbf{p} = [x, y, z]^T$, is computed as $h_p = h_c - y$, where h_c is the known height of the left camera above the ground. Examples of Cart Context descriptors are visualized in Fig. 2. To ensure the rectangular domain of the Cart Context descriptor is fully populated with points, the following relationships should be satisfied: $r_d \geq r_{lo}$ and $r_k \geq \sqrt{r_{lo}^2 + r_{la}^2}$.

C. Variable Offset Double Descriptor Distance Computation

The Cart Context descriptor offers a coarse, birds-eye-view representation of the structure surrounding the keyframe camera pose. In [14], the Cart Context descriptor distance is computed as the minimum column-wise cosine distance between the reference and all circular column shifts of the query. The circular shifts lend the distance computation lateral robustness but have no valid physical interpretation. We instead compute the descriptor distance using the variable offset concept previously applied in SMART [27], [26]. While Sum of Absolute Differences (SAD) is applied to the overlapping patches in SMART, we have empirically observed the best performance using cosine distance on the flattened patches. Specifically, the descriptor distance is computed as:

$$
\begin{aligned}
d(\mathbf{Q}, \mathbf{R}) = & \min_{k \in s_{lo}, l \in s_{la}} \text{cd}\left(\mathbf{Q}[i_Q, j_Q, h, w], \mathbf{R}[i_R, j_R, h, w]\right) \\
& i_Q = \max(1, -k+1), \;\; j_Q = \max(1, -l+1) \\
& i_R = \max(1, k+1), \;\; j_R = \max(1, l+1) \\
& h = m - |k|, \;\; w = n - |l|
\end{aligned} \tag{1}
$$

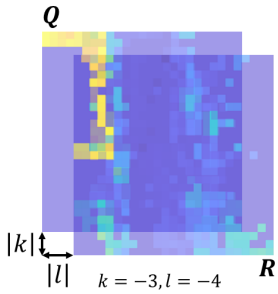

Fig. 3. A depiction, for a single longitudinal (k) and lateral (l) shift, of the overlapping regions of the query $\mathbf{Q}$ and reference $\mathbf{R}$ between which the cosine distance is computed.

where $\mathbf{Q} \in \mathbb{R}^{m \times n}$ and $\mathbf{R} \in \mathbb{R}^{m \times n}$ are the query and reference Cart Context descriptors, $\mathbf{Q}[i, j, h, w]$ denotes the submatrix obtained by selecting the rows $\{i, ..., i + h - 1\}$ and columns $\{j, ..., j + w - 1\}$ from $\mathbf{Q}$, s_{lo} and s_{la} are sets of longitudinal and lateral shifts, and $\text{cd}(\mathbf{A}, \mathbf{B})$ is the cosine distance between flattened matrices $\mathbf{A}$ and $\mathbf{B}$. Figure 3 visualizes how the longitudinal and lateral shifts are applied to the query and reference descriptors in Equation 1.

The descriptor distance procedure described above results in robustness to longitudinal and lateral shifts but not rotations. In [14], the opposing viewpoint case is accounted for by double flipping the reference descriptors, i.e., one flip about each axis. The double flipped versions are then treated as additional reference descriptors corresponding to the same places as their non-flipped counterparts [14]. This method is referred to as Augmented Cart Context (A-CC) [14]. The method works because the double flip intuitively yields a descriptor similar to that which would have been produced from the opposing view. Here, we instead perform the double flip on the query descriptor rather than the references to avoid either doubling the reference database size or requiring that the references be double flipped for each new query. As depicted in Fig. 2, for each new query and its double flipped counterpart, descriptor distances are computed against every reference in the database. For efficiency, computations across separate references are performed in parallel.

D. Double Distance Matrix Sequence Matching

The descriptor distances computed in the previous stage contribute one new column each to two separate, continuously updated distance matrices. The descriptor distances computed with the original query contribute to a distance matrix that captures similar viewpoint matches $\mathbf{D}_{sim}$, while those computed with the double flipped query contribute to a distance matrix that captures opposing viewpoint matches $\mathbf{D}_{opp}$. We expect a sequence of similar viewpoint matches to appear as a line with positive slope in $\mathbf{D}_{sim}$ and to produce no pattern in $\mathbf{D}_{opp}$. Conversely, we expect a sequence of opposing viewpoint matches to appear as a line with negative slope in $\mathbf{D}_{opp}$ and to produce no pattern in $\mathbf{D}_{sim}$. Note that the equi-spaced descriptor formation described in Section III-A should ensure a slope magnitude roughly equal to 1 in either case. The distance matrices shown in Fig. 2 were computed in an opposing viewpoint scenario and illustrate these expectations.

Fig. 4. Opposing viewpoint images from route *R0* [21] at the same place. **Left to right:** noon reference, noon query, sunset query and night query.

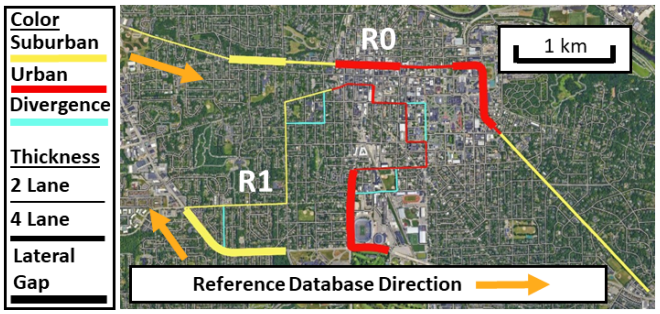

Fig. 5. Evaluated routes from the NSAVP dataset.

To predict the correct match without *a priori* knowledge of whether the viewpoint is similar or opposing, we perform sequence matching, as described in [25], separately within each distance matrix. Specifically, over the last w queries we sum over lines in $\mathbf{D}_{sim}$ with positive slopes (referred to as velocities in [25]) and in $\mathbf{D}_{opp}$ with negative slopes, searching for a line with the minimum sum. In each case, we evaluate slopes with magnitudes ranging from v_{min} to v_{max}. For efficiency, sums over multiple candidate lines are computed in parallel. Each of the two searches returns a predicted match for the query at the center of the search window. The match with the lowest sum is the final output and its score is computed as the lowest sum divided by the second lowest sum outside of a window centered on the match.

IV. Experimental Setup

A. Dataset Overview

The Oxford RobotCar dataset [33] is commonly used for evaluating opposing viewpoint VPR between front- and rear-facing cameras [9], [10], [11], [12]. However, while the Oxford Robotcar dataset includes a front-facing stereo camera, it has only short segments driven in opposing directions [33]. Therefore, we instead utilize the NSAVP dataset [21], which includes front-facing stereo monochrome and RGB cameras and ∼8 km routes, *R0* and *R1*, driven fully in opposing lanes. The sequences capture a variety of lighting conditions (Fig. 4), scene types (urban vs. suburban), traffic conditions, road widths (two vs. four lanes), and lateral gaps between lanes (e.g., medians and turn lanes). We use 6.7 and 7.6 km subsets of the *R0* and *R1* routes respectively (Fig. 5). We form reference descriptor databases for each route by applying the steps described in Sections III-A and III-B to sequences collected at noon. Query sequences will be referred to by their route, time of day, and viewpoint relative to the reference database. The one exception is a *R1* sequence

TABLE I

SPOT PARAMETER VALUES.

Parameter(s)	Value(s)	Description
r_d, r_k, r_a	35.35 m, 35.35 m, 90 m	Point cloud thresholds
s	2 m	Keyframe spacing
r_{lo}, r_{la}	25 m, 25 m	Descriptor ranges
m, n	25, 25	Descriptor dimensions
s_{lo}	{-2, 1, 0, 1, 2}	Variable offset longitudinal shifts
s_{la}	{-5, -4, ..., 4, 5}	Variable offset lateral shifts
w	75	Sequence length
v_{min}, v_{max}	0.6, 1.4	Sequence matching slope range

TABLE II

LABELS APPLIED TO EACH QUERY

If query i returns an accepted reference match j	
If $\lvert\mathbf{c}_i - \mathbf{c}_j\rvert \leq r_m$	true positive
Otherwise	false positive
If query i returns no accepted match	
If $\exists j' \in J, \lvert\mathbf{c}_i - \mathbf{c}_{j'}\rvert \leq r_m$	false negative
Otherwise	true negative

c_i, c_j, and $c_{j'}$ are ground truth positions
r_m is a localization radius
J is the set of all reference indices

collected at noon with divergences from its route, which will be referred to as *Diverge R1* (these divergences are shown in Fig. 5)[1].

B. Implementation Details

We run SO-DSO on the monochrome images of all sequences with default parameters. Following [20], we discard images during periods where the vehicle is stationary to ensure stable tracking. The parameters used for SPOT are listed in Table I. Despite the NSAVP dataset only containing lateral offsets in one direction, we use a symmetric set of lateral shifts to avoid *a priori* assumptions regarding lane shifts. We additionally conducted ablation and sequence length studies which are included in the appendix [1].

C. Baselines

We evaluate our method against two state-of-the-art opposing viewpoint VPR frameworks: LoST-X [10] and Seq2single [11]. Following [10], we additionally test LoST, LoST-X, and NetVLAD [5] with sequence matching using OpenSeqSLAM [34] (denoted with +SM). We use the same sequence length as used in SPOT. To align with [10] and [11], we use ground truth position data to sample the left RGB images input to these methods at a constant 2 meter distance apart. Note that this sampling lends these baselines an advantage over our proposed approach, as SPOT selects equi-spaced keyframes based on VO pose estimates, rather than ground truth. We also test the method proposed in [19], referred to here as SO-DSO VPR. This method also employs a rotation-invariant Scan Context descriptor with SO-DSO derived points clouds, but it was not originally evaluated against opposing viewpoints.

D. Evaluation Methodology & Metrics

We consider a query i to return an accepted reference match j if the match score is less than or equal to a given threshold. Note that sequence matching produces no match for the first and last $(w-1)/2$ queries [25], and we consider such queries to return no accepted match by default. For a given score threshold, precision and recall are computed by assigning a label to each query according to Table II. Precision-recall (PR) curves are drawn by varying the score threshold from the minimum to the maximum match score across all queries. We compute PR curves using two different values for the localization radius, r_m: 15 meters and 80 meters. The larger 80 meter threshold is chosen to match that used in [10] and the stricter 15 meter threshold is chosen to be just above the maximum lateral shift between opposing lanes in the NSAVP dataset (12.5 meters).

From the PR curves, we compute two metrics: maximum recall at 100% precision (MR100) and the area under the PR curve (AUC). MR100 indicates the percentage of all possible matches successfully attained without any false positives, while the AUC provides a summary of the performance that is less sensitive to individual false positives. In the ideal case, both metrics would equal 1. For methods utilizing sequence matching, the maximum possible value for both metrics is less than 1 due to the queries that are not assigned a match, so results with zero incorrect matches are denoted.

V. RESULTS

A. Place Recognition Performance

The MR100 and AUC values achieved by each method with each query sequence are presented in Table III. With the opposing viewpoint query sequences, SPOT demonstrates remarkable improvement over the baselines, achieving 91.7% MR100 with the *Noon R0* sequence and 83.6% MR100 with the dimly lit *Sunset R0* under the strict 15 meter localization radius. Performance is also strong on the *Diverge R1* sequence, which shows SPOT can reliably identify true negatives. The opposing viewpoint *Night R0* sequence is more challenging, with successful matches limited primarily to a well-lit urban portion. Altogether, the results indicate that SPOT can achieve excellent performance under opposing viewpoints and changes in lighting conditions so long as surrounding structure can still be perceived. Note that in some cases, there is even *worse* performance with the 80 meter localization radius because some of the true negatives with the 15 meter radius become false negatives with the 80 meter radius.

In contrast to the proposed method, the baselines achieve at most 10% MR100 under opposing viewpoints with the 80 meter localization radius. The AUC results and full PR curves (Fig. 6), indicate the baselines do generate a meaningful number of true positive matches within the 80 meter localization radius. However, with the stricter 15 meter localization radius, only the structure based SO-DSO VPR attains an AUC greater than 0.09 under opposing viewpoints. This localization inaccuracy is likely due to the *visual offset* problem described in Section II-B.

With similar viewpoints, all methods perform well under the 15 meter localization threshold. In terms of MR100,

[1]Original sequence names [21]: references: R0_RA0, R1_RA0; queries: R0_FA0, R0_FS0, R0_FN0, R1_FA0, R1_DA0, R0_RS0, R0_RN0.

TABLE III

PLACE RECOGNITION PERFORMANCE: MR100 AND AUC USING A 15 METER AND 80 METER LOCALIZATION RADIUS

Method	Opposing Viewpoint										Similar Viewpoint			
	Noon R0		*Sunset R0*		*Night R0*		*Noon R1*		*Diverge R1*		*Sunset R0*		*Night R0*	
	15 m	**80 m**	**15 m**	**80 m**	**15 m**	**80 m**	**15 m**	**80 m**	**15 m**	**80 m**	**15 m**	**80 m**	**15 m**	**80 m**
MR100														
SPOT (Ours)	**0.917**	**0.913**	**0.836**	**0.832**	**0.067**	**0.067**	**0.798**	**0.809**	**0.605**	**0.536**	0.968*	0.968*	0.199	0.199
LoST-X	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.004	0.000	0.007	0.819	0.900	0.018	0.018
LoST+SM	0.000	0.010	0.000	0.009	0.000	0.006	0.000	0.056	0.000	0.049	**0.978***	**0.978***	0.309	**0.335**
LoST-X+SM	0.000	0.036	0.000	0.015	0.000	0.021	0.000	0.100	0.000	0.041	**0.978***	**0.978***	**0.330**	0.330
Seq2single	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.000	0.002	0.166	0.195	0.000	0.008
NetVLAD+SM	0.000	0.029	0.000	0.019	0.000	0.003	0.000	0.000	0.000	0.000	**0.978***	**0.978***	0.269	0.269
SO-DSO VPR	0.000	0.000	0.000	0.008	0.000	0.000	0.002	0.002	0.000	0.000	0.111	0.111	0.000	0.000
AUC														
SPOT (Ours)	**0.970**	**0.965**	**0.968**	**0.963**	**0.262**	0.265	**0.972**	**0.967**	**0.913**	**0.887**	0.968*	0.968*	0.518	0.528
LoST-X	0.046	0.506	0.037	0.341	0.025	0.169	0.057	0.451	0.049	0.392	**0.999**	**1.000**	0.566	0.627
LoST+SM	0.027	0.864	0.006	0.657	0.022	0.259	0.086	0.861	0.074	0.752	0.978*	0.978*	**0.837**	**0.844**
LoST-X+SM	0.019	0.859	0.004	0.640	0.019	**0.296**	0.062	0.873	0.050	0.795	0.978*	0.978*	0.808	0.835
Seq2single	0.032	0.545	0.025	0.289	0.015	0.112	0.049	0.471	0.034	0.379	0.990	0.991	0.338	0.379
NetVLAD+SM	0.001	0.874	0.000	0.361	0.000	0.112	0.014	0.850	0.003	0.791	0.978*	0.978*	0.666	0.672
SO-DSO VPR	0.294	0.444	0.225	0.331	0.018	0.047	0.291	0.442	0.150	0.278	0.946	0.948	0.101	0.117

15 m / 80 m: 15 and 80 meter localization radii, *: zero incorrect matches

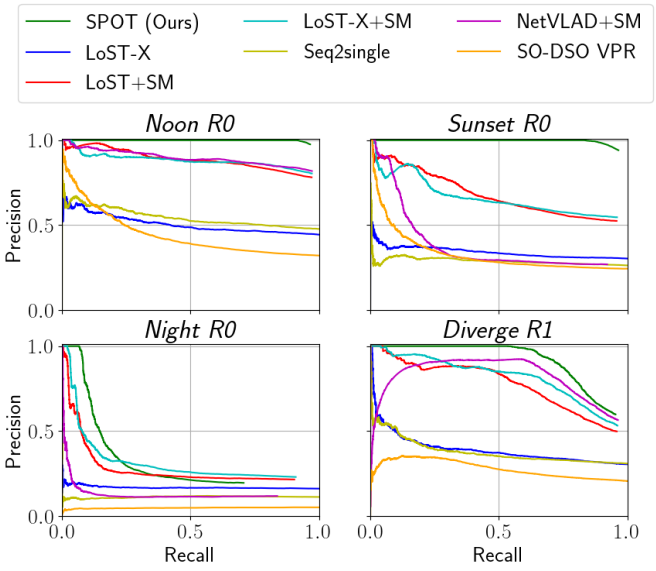

Fig. 6. Precision-recall curves for selected opposing viewpoint query sequences using an 80 meter localization radius.

methods with sequence matching perform better. SPOT performs competitively on *Night R0*, although it is surpassed by several of the baselines on this sequence.

B. Computation Time and Storage Requirements

Table IV lists computation time and reference storage space requirements for each method. All experiments were run on a machine with an AMD Ryzen 9 5950x 16-core, 32-thread CPU and a NVIDIA RTX A6000 GPU. The hardware utilized by each stage of each method is specified in the table. SPOT is more efficient in both aspects than nearly all baselines tested. LoST-X and Seq2single require the high dimensional output of an intermediate convolutional layer to be retained for each reference image resulting in storage requirements three orders of magnitude greater than all other methods tested.

TABLE IV

COMPUTATION TIME PER QUERY (OPPOSING *Noon R0*) AND STORAGE

Method	Query Place Description		Matching		Storage Per Reference
	Average Time (ms)	HW	Average Time (ms)	HW	
SPOT (Ours)	**0.16**	C-1	11.86	C-M	**5.0 KB**
LoST-X	131.17	G/C-1	85.03	C-1	15.8 MB
LoST+SM	131.17	G/C-1	21.64	C-1/M	49.2 KB
LoST-X+SM	131.17	G/C-1	93.06	C-1/M	15.8 MB
Seq2single	128.39	G	1962.16	C-1	16.3 MB
NetVLAD+SM	32.52	G	12.23	C-1/M	16.4 KB
SO-DSO VPR	0.66	C-1	**5.39**	C-M	19.2 KB

HW: Hardware, **G**: GPU process, **C-1/M**: single/multi-threaded CPU process

VI. LIMITATIONS

Several assumptions are implicit in our method that are frequently valid in driving scenarios but not generally applicable. Specifically, the method assumes: (1) predominantly straight motion, (2) similar or opposing viewpoint (yaw variation of 0 or 180 degrees), (3) limited roll-pitch rotations, and (4) a known, constant camera height above the ground. The current implementation also assumes forward-facing cameras, so modifications would be required to apply this method to configurations where the cameras oppose the direction of travel.

VII. CONCLUSIONS & FUTURE WORK

In this paper, we presented SPOT, a technique for opposing viewpoint VPR that relies exclusively on structure estimated through stereo VO. Evaluating SPOT against several baselines on the NSAVP dataset, we demonstrated remarkable improvement over the state-of-the-art. Overall, we believe SPOT further signals the potential of VO-derived structure for VPR. The relatively low localization error of our approach makes it promising for future integration within a SLAM system to support opposing viewpoint loop closure or multi-map merging. Moreover, future work could assess the proposed method's suitability with monocular visual inertial odometry and for cross-modality place recognition, using point clouds as a common representation.

REFERENCES

[1] Project webpage: https://umautobots.github.io/spot.

[2] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[3] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, Feb. 2016.

[4] X. Zhang, L. Wang, and Y. Su, "Visual place recognition: A survey from deep learning perspective," *Pattern Recognition*, vol. 113, p. 107760, May 2021.

[5] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016, pp. 5297–5307.

[6] A. Ali-Bey, B. Chaib-Draa, and P. Giguére, "MixVPR: Feature mixing for visual place recognition," in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA, Jan. 2023, pp. 2997–3006.

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

[8] A. Gawel, C. D. 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, July 2018.

[9] S. Garg, N. Suenderhauf, and M. Milford, "Don't look back: Robustifying place categorization for viewpoint- and condition-invariant place recognition," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, Australia, May 2018, pp. 3645–3652.

[10] ——, "LoST? appearance-invariant place recognition for opposite viewpoints using visual semantics," in *Proceedings of Robotics: Science and Systems*, Pittsburgh, PA, USA, June 2018, pp. 1–10.

[11] S. Garg, M. Babu V, T. Dharmasiri, S. Hausler, N. Suenderhauf, S. Kumar, T. Drummond, and M. Milford, "Look no deeper: Recognizing places from opposing viewpoints under varying scene appearance using single-view depth estimation," in *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada, May 2019, pp. 4916–4923.

[12] S. Garg, N. Suenderhauf, and M. Milford, "Semantic–geometric visual place recognition: a new perspective for reconciling opposing views," *The International Journal of Robotics Research*, vol. 41, no. 6, pp. 573–598, May 2022.

[13] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, J. J. Yebes, and S. Gámez, "Bidirectional loop closure detection on panoramas for visual navigation," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, Dearborn, MI, USA, June 2014, pp. 1378–1383.

[14] G. Kim, S. Choi, and A. Kim, "Scan context++: Structural place recognition robust to rotation and lateral variations in urban environments," *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 1856–1874, June 2022.

[15] A. Chapoulie, P. Rives, and D. Filliat, "A spherical representation for efficient visual loop closing," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, Barcelona, Spain, Nov. 2011, pp. 335–342.

[16] X. Xu, Y. Jiao, S. Lu, X. Ding, R. Xiong, and Y. Wang, "Leveraging BEV representation for 360-degree visual place recognition," *arXiv preprint arXiv:2305.13814*, pp. 1–11, May 2023.

[17] T. Cieslewski, E. Stumm, A. Gawel, M. Bosse, S. Lynen, and R. Siegwart, "Point cloud descriptors for place recognition using sparse visual information," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, May 2016, pp. 4830–4836.

[18] Y. Ye, T. Cieslewski, A. Loquercio, and D. Scaramuzza, "Place recognition in semi-dense maps: Geometric and learning-based approaches," in *British Machine Vision Conference*, London, UK, Sept. 2017, pp. 1–13.

[19] J. Mo and J. Sattar, "A fast and robust place recognition approach for stereo visual odometry using LiDAR descriptors," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA, Oct. 2020, pp. 5893–5900.

[20] A. Oertel, T. Cieslewski, and D. Scaramuzza, "Augmenting visual place recognition with structural cues," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5534–5541, Oct. 2020.

[21] S. Carmichael, A. Buchan, M. Ramanagopal, R. Ravi, R. Vasudevan, and K. A. Skinner, "Dataset and benchmark: Novel sensors for autonomous vehicle perception," *arXiv preprint arXiv:2401.13853*, pp. 1–11, Jan. 2024.

[22] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, June 2010, pp. 3304–3311.

[23] C. Masone and B. Caputo, "A survey on deep visual place recognition," *IEEE Access*, vol. 9, pp. 19 516–19 547, Jan. 2021.

[24] M. Zaffar, A. Khaliq, S. Ehsan, M. Milford, and K. McDonald-Maier, "Levelling the playing field: A comprehensive comparison of visual place recognition approaches under changing conditions," in *IEEE ICRA Workshop on Dataset Generation and Benchmarking of SLAM Algorithms for Robotics and VR/AR*, Montreal, Canada, Apr. 2019, pp. 1–8.

[25] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *2012 IEEE International Conference on Robotics and Automation*, Saint Paul, MN, USA, May 2012, pp. 1643–1649.

[26] E. Pepperell, P. I. Corke, and M. J. Milford, "All-environment visual place recognition with SMART," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, May 2014, pp. 1612–1618.

[27] E. Pepperell, P. Corke, and M. Milford, "Towards persistent visual navigation using SMART," in *Proceedings of the 2013 Australasian Conference on Robotics and Automation*, Sydney, Australia, Dec. 2013, pp. 1–9.

[28] S. Garg, T. Fischer, and M. Milford, "Where is your place, visual place recognition?" in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, Montreal, Canada, Aug. 2021, pp. 4416–4425.

[29] J. Mo and J. Sattar, "Extending monocular visual odometry to stereo camera systems by scale optimization," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, Nov. 2019, pp. 6921–6927.

[30] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, Mar. 2018.

[31] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, Oct. 2018, pp. 4802–4809.

[32] L. Matthies and S. Shafer, "Error modeling in stereo navigation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 3, pp. 239–248, June 1987.

[33] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The Oxford RobotCar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, Jan. 2017.

[34] N. Sünderhauf, P. Neubert, and P. Protzel, "Are we there yet? challenging SeqSLAM on a 3000 km journey across all four seasons," in *IEEE ICRA Workshop on Long-Term Autonomy*, Karlsruhe, Germany, May 2013, pp. 1–3.