Locus: LiDAR-based Place Recognition using Spatiotemporal Higher-Order Pooling

Kavisha Vidanapathirana[1,2], Peyman Moghadam[1,2], Ben Harwood[1], Muming Zhao[1],
Sridha Sridharan[2], Clinton Fookes[2]

Abstract— Place Recognition enables the estimation of a globally consistent map and trajectory by providing non-local constraints in Simultaneous Localisation and Mapping (SLAM). This paper presents *Locus*, a novel place recognition method using 3D LiDAR point clouds in large-scale environments. We propose a method for extracting and encoding topological and temporal information related to components in a scene and demonstrate how the inclusion of this auxiliary information in place description leads to more robust and discriminative scene representations. Second-order pooling along with a non-linear transform is used to aggregate these multi-level features to generate a fixed-length global descriptor, which is invariant to the permutation of input features. The proposed method outperforms state-of-the-art methods on the KITTI dataset. Furthermore, *Locus* is demonstrated to be robust across several challenging situations such as occlusions and viewpoint changes in 3D LiDAR point clouds. The open-source implementation is available at: https://github.com/csiro-robotics/locus .

I. INTRODUCTION

Place Recognition (PR) is an essential capability required by autonomous robots and driverless cars, which enables the recognition of previously visited places under changing viewpoints and environmental conditions. Place recognition is crucial for various applications such as loop closure detection for large-scale, global data association in Simultaneous Localization and Mapping (SLAM) or metric localization within known maps [1], [2].

In this paper, we consider the problem of place recognition based on 3D LiDAR point clouds. The majority of the current state-of-the-art LiDAR-based place recognition methods extract representations of 3D point clouds based on either local or global descriptors. Local descriptors such as keypoint features [3] often suffer from low repeatability under noise and minor changes in the environment. The demand for extracting discriminative and generalizable global descriptors from point clouds has led to the development of several different techniques [4]–[6]. However, the majority of global descriptors are not robust to viewpoint variations, occlusions and dynamic scenes [7]. A potential reason for this fragility is that these methods are inherently not capable of capturing topological and temporal interactions between different components in the scene. Such auxiliary information is vital for the construction of robust and discriminative representations since a higher-level understanding is needed to distinguish between different scenes of similar structure.

To this end, we propose a novel place recognition method (named *Locus*) which effectively exploits multi-level features for global place description using 3D LiDAR data. We define multiple levels of feature representations for each point cloud frame. A point cloud frame is defined as the points accumulated from a sweep of the LiDAR [8]. First, segment features are extracted to encode structural information of each point cloud frame. The segment-based representation leverages from the advantages of both local and global representations while not suffering from their individual drawbacks. Compared to local keypoint based representations, they are more descriptive and more robust to noise and changing environments. Next, spatial feature pooling encodes topological relationships between segments within a point cloud frame. Capturing topological relationships between components of a scene enables discrimination between scenes of similar composition but different arrangement.

Finally, temporal feature pooling encodes co-occurrences of segments in the past k_t point cloud frames providing robustness to sensor motion and dynamic objects. Once multi-level features are extracted, second-order pooling aggregates information of local features over a point cloud frame to form a holistic representation for place recognition. Second-order pooling captures the multiplicative interactions between the multi-level features and outputs a fixed-length descriptor which is invariant to the number and permutation of the input features. Furthermore, the fixed dimension of the global descriptor enables maintaining the computational complexity.

Our main contributions are summarised as follows:

- We introduce multi-level features which encode structural appearance, topological relationships and temporal correspondences related to components in a scene.
- We formulate the generation of a global descriptor which encodes these multi-level features into a single viewpoint invariant representation using second-order pooling and demonstrate how these multi-level features contribute to place recognition performance.
- Our proposed method (*Locus*) outperforms the state-of-the-art place recognition methods on the KITTI dataset.
- We quantitatively evaluate the robustness of our *Locus* method on a variety of challenging scenarios such as viewpoint changes and occlusion.

[1] Kavisha Vidanapathirana, Peyman Moghadam, Ben Harwood, Muming Zhao are with the Robotics and Autonomous Systems Group, DATA61, CSIRO, Brisbane, QLD 4069, Australia. E-mails: *firstname.lastname* @data61.csiro.au

[2] Kavisha Vidanapathirana, Peyman Moghadam, Sridha Sridharan, Clinton Fookes are with the School of Electrical Engineering and Robotics, Queensland University of Technology (QUT), Brisbane, Australia. E-mails: {kavisha.vidanapathirana, peyman.moghadam, s.sridharan, c.fookes}@qut.edu.au
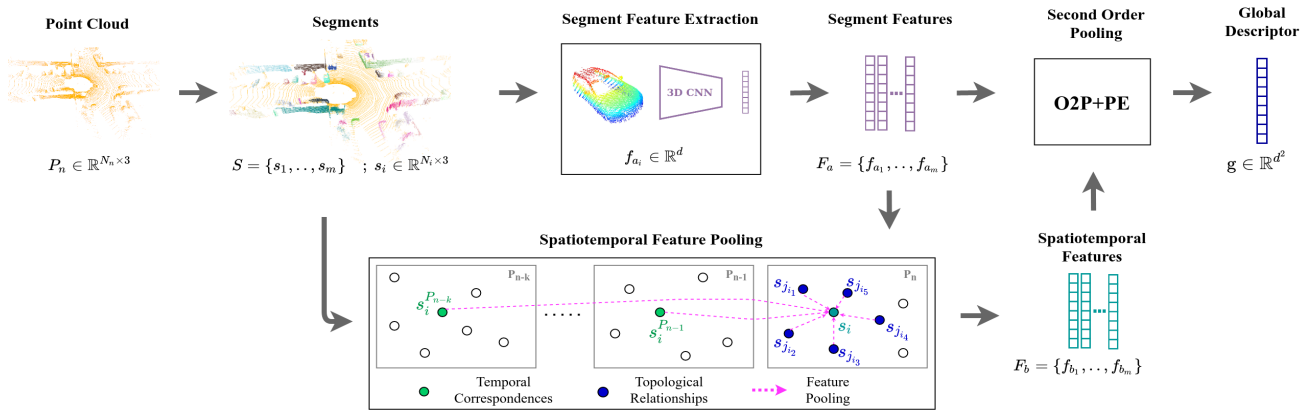
Fig. 1: The overall framework of the proposed *Locus* method. Segments extracted from a LiDAR point cloud frame are described using two complementary sets of features. One feature describes the structural appearance of a segment while the other encodes topological and temporal information related to a segment. The two sets of features are aggregated using second-order pooling (O2P) followed by a Power-Euclidean transform (PE) to obtain a global descriptor of the point cloud.

II. RELATED WORK

A. Point cloud representation and descriptor generation

To address the challenge of place recognition, the scene-level point cloud is often encoded in three different ways; a set of local descriptors, a single global descriptor or a set of object/segment descriptors. Local descriptor based methods first detect a set of keypoints in the point cloud and then form local descriptors by encoding information in each keypoint neighbourhood [3], [9]. Local descriptors and keypoint detection suffer from low repeatability in noisy point clouds and changing environments.

Global descriptors aim to describe the entire point cloud by a single vector representation. M2DP [4] generates a global descriptor by projecting the points into multiple 2D planes and calculating density signatures. PointNetVLAD [5] extracts a global descriptor using an end-to-end process based on PointNet and NetVLAD [10]. PointNetVLAD ignores the spatial distribution of features and hence lacks descriptive power. LPD-Net [11] addresses the limitation of PointNetVLAD by using adaptive local feature extraction along with a graph-based neighborhood aggregation module. Recently, DH3D [12] proposed a network which learns both the keypoint detection and local description and generates a global descriptor using NetVLAD. The aforementioned global descriptors do not demonstrate rotational-invariance and often fail upon reverse revisits.

The current state-of-the-art consists of several rotation-invariant global descriptors. ScanContext [6] records the maximum height of points in a 2D grid-map of the scene and computes pairwise descriptor distances by comparing distances to all column shifted variants of the query descriptor to achieve rotational-invariance. Intensity ScanContext [13] extends ScanContext by including the intensity return of the LiDAR sensor. The ScanContext and its variants are not capable of capturing scene composition or topological relationships and rely on expensive distance calculation to ensure rotational-invariance.

Recently, Kong *et al.* [14] represented point clouds as a semantic graph and recognize places using a graph similarity network. Their network is capable of capturing topological and semantic information from the point cloud and also achieves rotational-invariance along with state-of-the-art performance. However, due to the use of semantic segmentation, their method suffers from the following two bottlenecks. First, it is dependant on the existence of pre-defined semantic classes in the test dataset (*i.e.*, domain gap problem). Second, since each segment is only represented by its class label, the method is not capable of differentiating between two segments of the same class and thus loses valuable information related to intra-class variations.

Euclidean segment based representations [7] are less affected by the aforementioned drawbacks while still being able to capture topological and semantic information of the components in a scene. SegMatch [7] and SegMap [15] address the challenges of local and global descriptors by constructing a global map along the trajectory and performs recognition via segment-wise kNN retrieval on this map followed by a geometric consistency check. In this paper, we also leverage from segment descriptors for point cloud encoding but we avoid the construction of a global map and instead treat place recognition as a retrieval problem similar to recent LiDAR-based place recognition research [5], [6], [13], [14]. In this sense, we avoid segment-wise kNN search and construct a global descriptor through the aggregation of multi-level segment features.

B. Feature aggregation

Bag-of-Words (BoW) [16] and its higher-order variants (Vector of Locally Aggregated Descriptors (VLAD) [17], NetVLAD [10], Fisher Vector (FV) [18]) are popular methods in many retrieval tasks which construct global descriptors by aggregating local descriptors using zeroth, first or second order statistics respectively. These codebook-based aggregation methods require a preprocessing step to learn the codebook (a standard BoW, cluster centers in VLAD

and Gaussian Mixture Models (GMM) in FV) and do not transfer well to unseen environments.

Other higher-order aggregation methods such as Second Order Pooling (O2P) [19] do not require such pre-processing steps and and hence enables to be used for online place recognition in unseen environments. They also tend to be more efficient in run-time since the description stage does not require finding the nearest codebook word for each feature. Additionally, O2P methods allow the aggregation of complementary features [20] while maintaining a fixed-length descriptor without violating permutation invariance. In this paper, we propose to use second-order pooling to aggregate multiple levels of feature representations (structural, topological and temporal) into a fixed-length global descriptor for place recognition.

C. Incorporating temporal information

In place recognition, temporal information has been used almost exclusively in the retrieval stage. In visual place recognition (VPR), SeqSLAM [21] incorporated sequential information by comparing feature similarity over time and demonstrated dramatic performance improvement when dealing with changing appearance. Lynen *et al.* [22] modeled sequential retrieval as a probability density estimation in votes versus travel distance space. More recently, SeqLPD [23] extended LPD-Net with SeqSLAM for LiDAR-based place recognition. Our method differs from these sequential retrieval methods by incorporating temporal information in the place description stage instead. Recently, in VPR, Delta descriptors [24] outlined how visual sequential-representation provides inherent robustness to variations in camera motion and environmental changes.

III. METHOD

This section describes our proposed method (named *Locus*) for LiDAR-based place description including the segment based representation of the point cloud, the generation of the multi-level features and their second-order aggregation (see Fig. 1).

A. Segment-based representation

Segments are defined as Euclidean clusters of points in the point cloud representation. Our segment extraction is performed similar to the SegMatch implementation [7] by first removing the ground plane and then extracting Euclidean clusters of points. The maximum Euclidean distance between two neighbouring points in the same segment is set to $0.2m$ and the segments contain a minimum of 100 and a maximum of 15000 points. Thus, each input point cloud $P \in \mathbb{R}^{N \times 3}$ is represented by a set of segments $S = \{s_1, .., s_m\}$, where $s_i \in \mathbb{R}^{N_i \times 3}$, and the number of segments m varies depending on the environment and range of the point cloud.

For each segment, a compact deep feature which encodes the structural appearance of the segment is obtained using the *SegMap-CNN* network proposed in [15]. The network represents each segment in a voxel grid of 32x32x16 of

which the voxel sizes (0.1m by default) are scaled in order to fit larger segments. The description network consists of three 3D convolutional layers plus max-pool layers followed by two fully connected layers. For a given set of m segments S, it outputs a set of compact features $F_a = \{f_{a_1}, .., f_{a_m}\}$ (where $f_{a_i} = f_a(s_i) \in \mathbb{R}^d, d = 64$) which discriminates segments based on structural appearance.

B. Spatiotemporal Pooling

Incorporating topological and temporal information in place description has many advantages when dealing with changing environments and varying sensor motion. With the aim of encoding this information, we compute a complimentary set of features $F_b \in \mathbb{R}^{m \times d}$ for each point cloud frame P_n, in addition to the structural appearance features $F_a \in \mathbb{R}^{m \times d}$ described in section III-A. This is achieved via two stages of feature pooling: spatial and temporal.

Feature pooling computes a new feature for a segment by taking the weighted average of features of all related segments. Encoding topological information is achieved via feature pooling based on spatial relationships within a point cloud frame. Temporal information is encoded via pooling features based on temporal correspondences across multiple point cloud frames. For a query segment $s_i^{P_n}$ from the current point cloud frame P_n, we find topological relationships and temporal correspondences with all other segments $s_j^{P_l} \in \mathbb{S}_{n,k_t}$ from the current frame and k_t frames into the immediate past,

$$\mathbb{S}_{n,k_t} = \{S^{P_n}, .., S^{P_{n-k_t}}\}, \quad S^{P_l} = \{s_1^{P_l}, .., s_{m_l}^{P_l}\}, \quad (1)$$

If a segment $s_j^{P_l}$ is spatially or temporally related to the query segment $s_i^{P_n}$, its structural appearance feature $f_a(s_j^{P_l})$ is included in the feature pooling of the query segment.

1) Spatial Feature Pooling: We define spatial relationships between segments through a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Vertices $\mathcal{V} = \{1, .., m\}$ represent the segments $S = \{s_1, .., s_m\}$ in the current point cloud frame and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represent which segments relate to which other segments. $\mathcal{G}$ is constructed as a kNN graph where each segment is connected to its k_s nearest neighbours ($k_s = 5$) and the distance between segments is calculated using the minimum translational distance (MTD) [25] between the convex hulls of the segments. We use the `QuickHull` algorithm [26] to compute convex hulls. Since the segment extraction process in Section III-A guarantees that segments do not overlap, the MTD ($\mathcal{D}$) can be computed as follows,

$$\mathcal{D}(s_i, s_j) = \min\{\|p_{i_x} - p_{j_y}\| : \forall\, p_{i_x} \in \hat{s_i}, p_{j_y} \in \hat{s_j}\} \quad (2)$$

where $\|p_{i_x} - p_{j_y}\|$ represents Euclidean distance between points $p_{i_x}, p_{j_y} \in \mathbb{R}^3$ and $\hat{s} = $ `QuickHull`(s). The spatial feature pooling for segment s_i is then carried out as,

$$\Phi(s_i) = \sum_{j:(i,j)\in\mathcal{E}} \phi(i,j) f_a(s_j), \quad (3)$$

$$\phi(i,j) = \texttt{softmax}\{\mathcal{D}(s_i, s_j)\}$$
$$= \frac{exp(-\beta \cdot \mathcal{D}(s_i, s_j))}{\sum_{k:(i,k)\in\mathcal{E}} exp(-\beta \cdot \mathcal{D}(s_i, s_k))}, \quad (4)$$

Algorithm 1: Estimate temporal segment correspondences from frame P_l to P_{l-1}

Result: Corresponding segment indices $S_c \in \mathbb{R}^m$
$S^{P_l}, S^{P_{l-1}} \in \mathbb{R}^{m' \times n' \times 3}$ // segments;
$F_a^{P_l}, F_a^{P_{l-1}} \in \mathbb{R}^{m' \times d}$ // features;
$C^{P_l}, C^{P_{l-1}} \in \mathbb{R}^{m' \times 4}$ // segment centroids;
$^{l-1}T_l \in SE(3)$ // relative pose from P_{l-1} to P_l;
for $s_i \in S^{P_l}$ **do**
  $f_{a_i}, {}^l c_i$ // feature, centroid of segment s_i in P_l;
  $F_N \leftarrow kNN(f_{a_i}, F_a^{P_{l-1}})$;
  $C_N \leftarrow rNN(^{l-1}T_l\, {}^l c_i, C^{P_{l-1}})$;
  $N \leftarrow intersection(F_N, C_N)$;
  if $len(N) > 0$ **then**
  | $S_c[i] \leftarrow arg\_min(N)$;
  else
  | $S_c[i] \leftarrow \varnothing$;
  end
end

---

where $\beta = 0.1$ is a smoothing factor. The spatially-pooled feature $\Phi(s_i)$ is essentially a weighted average of the structural-appearance features (f_a) of the 5 closest segments to segment s_i. This captures information on the immediate neighbourhood of s_i and thus contributes towards encoding topological information in the scene.

*2) **Temporal Feature Pooling**:* Temporal relationships are defined by segment correspondences between frames. The segment $s_i^{P_n}$ in the current point cloud frame P_n will only relate to its corresponding segment in each of the k_t previous frames ($k_t = 3$). Segment correspondence indices S_c are calculated as in Algorithm 1 iteratively k_t times.

First kNN finds the indices of the k nearest neighbours of f_{a_i} from the set of features $F_a^{P_{l-1}}$ of the previous frame. Next, rNN finds the indices of nearest neighbor centroids from the previous frame within a radius $r = 1m$. To increase accuracy, the rNN search takes into account the the sensor's relative pose across frames represented by the homogeneous transformation $^{l-1}T_l$. Finally, common elements in both nearest neighbour sets are found using the `intersection` function and the function `arg_min` finds the index of the segment which minimises both feature-space and Euclidean-space distance.

For the selected segment $s_i^{P_n}$ in the current frame, the set of corresponding segments can be obtained from S_c as $\{s_i^{P_{n-1}}, .., s_i^{P_{n-k_c}}\} = \{\mathbb{S}_{n,k_t}(j) : j \in S_c\}$. We note that $k_c \leq k_t$ since correspondences can be lost between frames (when a segment which minimises both feature space and Euclidean-space distance is not available in the previous frame). The temporal pooling for $s_i^{P_n}$ is then carried out as,

$$\Psi(s_i) = \sum_{j \in S_c} \psi(i,j) f_a(\tilde{s}_j), \quad \tilde{s}_j = \mathbb{S}_{n,k_t}(j), \quad (5)$$

where $\psi(i,j)$ is calculated similar to $\phi(i,j)$ in Eq. 4 with $\mathcal{D}(s_i, s_j)$ replaced by $\|f_a(s_i) - f_a(\tilde{s}_j)\|$ and k is sampled

from S_c. The aggregation of features across multiple sequential frames essentially magnifies the weight of features corresponding to highly repeatable segments (segments which are extracted at every frame in the sequence and mapped to a similar point in the f_a feature space each time) and thus, inherently down-weights non-repeatable segments.

The final sptiotemporal feature f_b is obtained as the average of a spatially-pooled feature $\Phi(s_i)$ and a temporally-pooled feature $\Psi(s_i)$,

$$f_b(s_i) = (\Phi(s_i) + \Psi(s_i))/2. \quad (6)$$

C. Second-order pooling

Given a set of segments $S = \{s_1, .., s_m\}$ and two complementary sets of features $F_a \in \mathbb{R}^{m \times d}$ and $F_b \in \mathbb{R}^{m \times d}$ the second-order pooling F^{O_2} of the features is defined as,

$$F^{O_2} = \{F_{xy}^{O_2}\}, \quad F_{xy}^{O_2} = \max_{s \in S} f_{xy}^{o_2}(s), \quad (7)$$

where F^{O_2} is a matrix with elements $F_{xy}^{O_2}(1 \leq x, y \leq d)$ and $f^{o_2}(s) = f_a(s) f_b(s)^T \in \mathbb{R}^{d \times d}$ is the outer product of the two complementary features of segment s ($f_a(s), f_b(s) \in \mathbb{R}^d$). This accounts to taking the element-wise maximum of the second-order features of all segments in the scene.

In order to make the scene descriptor matrix F^{O_2} more discriminative, it is decomposed in to singular values as $F^{O_2} = U\lambda V$ and transformed non-linearly using the Power-Euclidean (PE) transform [27] [28] into $F_\alpha^{O_2}$ by raising each of its Eigen values by a power of α as follows,

$$F_\alpha^{O_2} = U\hat{\lambda}V, \quad \hat{\lambda} = diag(\lambda_{1,1}^\alpha, .., \lambda_{d,d}^\alpha), \quad (8)$$

where $\alpha = 0.5$. The matrix $F_\alpha^{O_2}$ is flattened and normalized to obtain the final global descriptor vector $\mathbf{g} \in \mathbb{R}^{d^2}$.

The aggregation of multi-level features using higher-order pooling has multiple advantages for segment-based place description. First, it allows the encoding of complementary features for each segment thus enabling the incorporation of structural-appearance information along with topological and temporal information. Second, even though different point clouds will consist of varying number of segments m of varying sizes N_i, the output dimension of the aggregated feature is fixed and therefore computational time can be greatly reduced. Finally, the aggregated feature is invariant to the permutation of its inputs, which results in a viewpoint invariant global descriptor for place recognition.

IV. EXPERIMENTAL SETUP

A. Evaluation criteria

We evaluate performance using the Precision-Recall curve and its scalar metric: the maximum $F1$ score ($F1_{max}$). Additionally, we use the Extended Precision (EP) metric proposed in [29] as it highlights the maximum recall at 100% precision which is vital for robustness in place recognition. For Precision (P) and Recall (R), the $F1_{max}$ and EP scores are defined as,

$$F1_{max} = \max_\tau 2\frac{P_\tau \cdot R_\tau}{P_\tau + R_\tau}, \quad EP = \frac{P_{R0} + R_{P100}}{2}, \quad (9)$$

Feature Types	EP	$F1_{max}$
structural $\otimes$ structural	0.739	0.796
structural $\otimes$ spatial	0.847	0.895
structural $\otimes$ temporal	0.865	0.900
structural $\otimes$ spatiotemporal	**0.912**	**0.931**

TABLE I: A comparison of *Locus* performance on KITTI sequence 08 when different input feature types are used.

Fig. 2: Precision-Recall curves of our proposed *Locus* method on the KITTI dataset.

where τ is the threshold for positive prediction, P_{R0} is the Precision at minimum Recall, and R_{P100} is the maximum Recall at 100% Precision.

Retrieval is performed based on the comparison of the cosine distance of the query descriptor with a database of descriptors of previously visited places. Inline with the evaluation criteria of [14], previous entries adjacent to the query by less than $30s$ time difference are excluded from search to avoid matching to the same instance. The top-1 retrieval is considered a positive if the associated distance to query is less than the test threshold (τ). A positive retrieval is considered a true-positive if it is less than 3m from the ground truth pose of the query and a false-positive if it is greater than 20m away to maintain consistency with the evaluation in [14].

B. Dataset

Evaluation is performed on the KITTI odometry dataset [8] which consists of Velodyne HDL-64E LiDAR scans collected from a moving vehicle in multiple dynamic urban environments in Karlsruhe, Germany. In line with the most recent evaluation on this dataset [14], we evaluate on sequences 00, 02, 05, 06, 07, and 08.

V. RESULTS

In this section, we demonstrate the contribution of each component in our proposed method. We also provide quantitative and qualitative results to validate our method in comparison to state-of-the-art methods and we quantitatively evaluate the robustness our proposed method against viewpoint changes and sensor occlusion.

A. Contribution of individual components

We investigate the contributions of each component (ablation study) of our proposed method using $F1_{max}$ and EP on KITTI sequence 08. First, we demonstrate the importance of the Power Euclidean (PE) non-linear transform in our second-order pooling. For the generation of feature f_a, we use the model trained on extracted segments from the sequences 05 and 06 [15]. The most basic second-order pooling method (*O2P*) is the aggregation of second-order statistics of the same feature (Eq. 7 with $f_b = f_a$). This can be extended with the Power Euclidean (PE) non-linear transform to get *O2P+PE* (Eq. 8 with $f_b = f_a$). The *O2P+PE* shows a dramatic increase of 44.7% $F1_{max}$ (from 0.550 to 0.796) and and improvement of 32.4% EP (from 0.564 to 0.739) when including the non-linear transform. This demonstrates the importance of PE for obtaining discriminative descriptors.
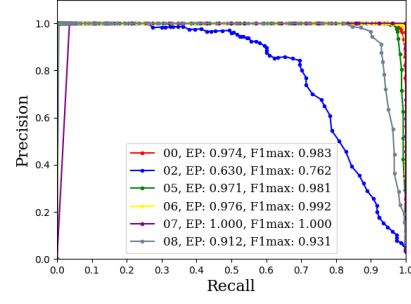
The rest of the evaluation uses the *O2P+PE* feature pooling and compares the contribution of each input feature type. The input f_a is the structural appearance feature and the input feature f_b is varied across all multi-level feature types. Table I shows the respective performance for each of the following cases: $f_b = f_a$ (structural), $f_b = \Phi$ (structural $\otimes$ spatial), $f_b = \Psi$ (structural $\otimes$ temporal), and $f_b = (\Phi+\Psi)/2$ (structural $\otimes$ spatiotemporal).

Table I demonstrates that pooling information from complementary features improves the place recognition performance. Furthermore, incorporating topological or temporal information individually makes the final descriptor more discriminative and improves $F1_{max}$ by 12.4% and 13.1% respectively. The best performance of our proposed method is achieved when incorporating both topological and temporal information along with the structural appearance (+ 17% $F1_{max}$).

B. Comparison to State-of-the-Art

In this section, we compare the results of our method against other state-of-the-art results. Table II shows summary comparison of $F1_{max}$ scores. *Locus* outperforms all other methods with the highest mean $F1_{max}$ score (0.942). Our method sets a new state-of-the-art place recognition performance on KITTI dataset. *Locus* achieves a significant improvement on the challenging sequence 08 which consists of number of reverse or orthogonal revisits thereby demonstrating the robustness to viewpoint variations.

The Precision-Recall curves of *Locus* across all sequences are presented in Fig. 2. The curves highlight that the performance of *Locus* on sequence 02 is significantly poor compared to other sequences. As depicted in Fig. 3, sequence 02 has a long stretch of road with many false negatives. This road consists of non-descriptive point clouds where the segment extraction process fails to obtain a high number of descriptive segments.

The qualitative visualization of our method at R_{P100} (*i.e.*, maximum Recall at 100% Precision) is shown in Fig. 3. The figure shows the locations of true positive (red) and false negative (black) retrievals along the trajectory. This demonstrates that our method is capable of obtaining accurate retrievals in a wide range of environment and revisit types. We observe that most failure cases (black dots) occur

Methods	00	02	05	06	07	08	Mean
M2DP [4]	0.836	0.781	0.772	0.896	0.861	0.169	0.719
ScanContext [6]	0.937	0.858	0.955	**0.998**	0.922	0.811	0.914
PointNetVLAD [5]	0.785	0.710	0.775	0.903	0.448	0.142	0.627
PointNetVLAD* [5]	0.882	0.791	0.734	0.953	0.767	0.129	0.709
SemGraph-RN [14]	0.960	**0.859**	0.897	0.944	0.984	0.783	0.904
Locus (Ours)	**0.983**	0.762	**0.981**	0.992	**1.0**	**0.931**	**0.942**

TABLE II: $F1_{max}$ scores on the KITTI dataset. Results of M2DP, ScanContext, PointNetVLAD, and SemGraph are as presented in [14] with the same evaluation criteria. PointNetVLAD* is the PointNetVLAD method retrained on KITTI [14].
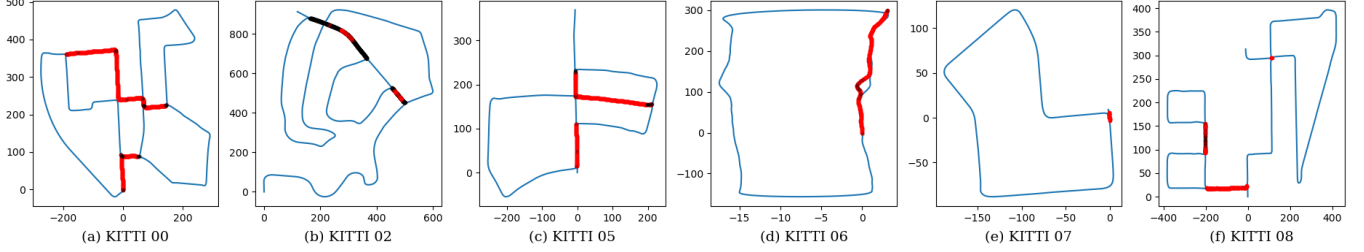

(a) KITTI 00   (b) KITTI 02   (c) KITTI 05   (d) KITTI 06   (e) KITTI 07   (f) KITTI 08

Fig. 3: Qualitative performance visualization of our method at R_{P100} (zero false positives) along the trajectory. Red: true positives, Black: false negatives, Blue: true negatives (not revisits).

C. Robustness tests

We evaluate the robustness of our method against the state-of-the-art on a variety of challenging scenarios which simulate real-word adverse conditions. We simulate these adverse conditions by introducing a set of distortions to the input point cloud. Performance is compared based on the $F1_{max}$ metric on the same KITTI sequences.

1) Viewpoint changes: We simulate viewpoint changes by rotating point clouds about the z-axis with a random angle. The change in mean $F1_{max}$ over all sequences on this test compared to the normal results was -0.007 for our method. The respective change for ScanContext was +0.002 and for SemGraph-RN it was -0.003 as presented in [14]. *Locus*, ScanContext and SemGraph-RN show less than 1% difference implying that random rotation of the input point cloud does not effect the final performance of these methods. Other methods such as M2DP, PointNetVLAD have been shown not to be rotation invariant as mentioned in [14].

2) Occlusion: This test aims to simulate occlusion of the LiDAR where the field-of-view of the sensor can be greatly reduced due to nearby dynamic objects or self-occlusion. We extend the occlusion test in [14] which only considered occlusions of 30° by evaluating performance of various occlusion angles θ_{occ} from 0° up to 180°. The occlusion test consists of removing all points which lie within a sector of a randomly selected azimuth. We compare our method with ScanContext (SC) [6], which achieved the second best overall performance in the Table II. ScanContext (SC) mean $F1_{max}$ performance drops by more than 20% at 45° occlusion and around 50% at 90° occlusion. Our method shows small performance degradation of 3.2% at 45° occlusion and 9.98% at 90° occlusion. Our method outperforms ScanContext (SC) with a large margin of 45.2% mean $F1_{max}$ at 90° occlusion. Also note that in our method, the $F1_{max}$ of sequences 00
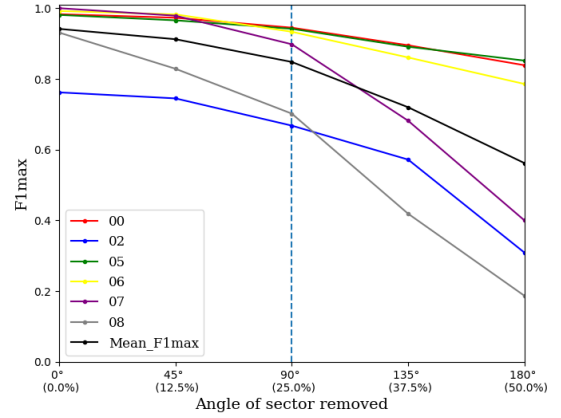

Fig. 4: Robustness of *Locus* to scan occlusion. Vertical blue line shows when the mean $F1_{max}$ drops by 10%.

and 05 remains above 80% even at occlusions of 180° where 50% of the point cloud is removed. Results for all sequences are depicted in Figure 4.

VI. CONCLUSION

In this paper, we presented *Locus*, a novel LiDAR-based place recognition method for large-scale environments. We presented the advantages of scene representation via the aggregation of multi-level features related to components in a scene. We quantitatively showed how the inclusion of topological and temporal information in the description stage leads to an improvement in final place recognition performance. We formulated the generation of a global descriptor which incorporates all multi-level features without violating rotational-invariance. We validated our method through evaluation on the KITTI dataset where it surpassed the state-of-the art. Furthermore, we demonstrated the robustness of *Locus* against viewpoint changes and occlusion.

REFERENCES

[1] C. Park, P. Moghadam, S. Kim, A. Elfes, C. Fookes, and S. Srid-haran, "Elastic LiDAR Fusion: Dense Map-Centric Continuous-Time SLAM," in *Proceedings - IEEE International Conference on Robotics and Automation*, sep 2018, pp. 1206–1213.

[2] C. Park, S. Kim, P. Moghadam, J. Guo, S. Sridharan, and C. Fookes, "Robust photogeometric localization over time for map-centric loop closure," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1768–1775, 2019.

[3] S. Salti, F. Tombari, and L. Di Stefano, "SHOT: Unique signatures of histograms for surface and texture description," *Computer Vision and Image Understanding*, vol. 125, pp. 251–264, 2014.

[4] L. He, X. Wang, and H. Zhang, "M2DP: A novel 3D point cloud descriptor and its application in loop closure detection," in *IEEE International Conference on Intelligent Robots and Systems*, nov 2016, pp. 231–237.

[5] M. A. Uy and G. H. Lee, "PointNetVLAD: Deep Point Cloud Based Retrieval for Large-Scale Place Recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, dec 2018, pp. 4470–4479.

[6] G. Kim and A. Kim, "Scan Context: Egocentric Spatial Descriptor for Place Recognition Within 3D Point Cloud Map," in *IEEE International Conference on Intelligent Robots and Systems*, dec 2018, pp. 4802–4809.

[7] R. Dube, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "SegMatch: Segment based place recognition in 3D point clouds," in *Proceedings - IEEE International Conference on Robotics and Automation*, jul 2017, pp. 5266–5272.

[8] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, sep 2013.

[9] J. Guo, P. V. Borges, C. Park, and A. Gawel, "Local Descriptor for Robust Place Recognition Using LiDAR Intensity," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1470–1477, apr 2019.

[10] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437–1451, jun 2018.

[11] Z. Liu, S. Zhou, C. Suo, P. Yin, W. Chen, H. Wang, H. Li, and Y. Liu, "LPD-Net: 3D Point Cloud Learning for Large-Scale Place Recognition and Environment Analysis," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 2831–2840.

[12] J. Du, R. Wang, and D. Cremers, "DH3D: Deep Hierarchical 3D Descriptors for Robust Large-Scale 6DoF Relocalization," in *European Conference on Computer Vision (ECCV)*, jul 2020, pp. 744–762.

[13] H. Wang, C. Wang, and L. Xie, "Intensity Scan Context: Coding Intensity and Geometry Relations for Loop Closure Detection," in *IEEE International Conference on Robotics and Automation (ICRA)*, sep 2020, pp. 2095–2101.

[14] X. Kong, X. Yang, G. Zhai, X. Zhao, X. Zeng, M. Wang, Y. Liu, W. Li, and F. Wen, "Semantic Graph Based Place Recognition for 3D Point Clouds," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, aug 2020, pp. 8216–8223.

[15] R. Dubé, A. Cramariuc, D. Dugas, H. Sommer, M. Dymczyk, J. Nieto, R. Siegwart, and C. Cadena, "SegMap: Segment-based mapping and localization using data-driven descriptors," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 339–355, mar 2020.

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

[17] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1704–1716, 2012.

[18] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222–245, dec 2013.

[19] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu, "Semantic segmentation with second-order pooling," in *European Conference on Computer Vision (ECCV)*, vol. 7578, 2012, pp. 430–443.

[20] T. Y. Lin, A. Roychowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1449–1457.

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

[22] S. Lynen, M. Bosse, P. Furgale, and R. Siegwart, "Placeless Place-Recognition," in *Proceedings - 2014 International Conference on 3D Vision, 3DV 2014*, feb 2015, pp. 303–310.

[23] Z. Liu, C. Suo, S. Zhou, F. Xu, H. Wei, W. Chen, H. Wang, X. Liang, and Y. H. Liu, "SeqLPD: Sequence Matching Enhanced Loop-Closure Detection Based on Large-Scale Point Cloud Description for Self-Driving Vehicles," in *IEEE International Conference on Intelligent Robots and Systems*, nov 2019, pp. 1218–1223.

[24] S. Garg, B. Harwood, G. Anand, and M. Milford, "Delta Descriptors: Change-Based Place Representation for Robust Visual Localization," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5120–5127, oct 2020.

[25] S. Cameron and R. Culley, "Determining the minimum translational distance between two convex polyhedra," in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, vol. 3, pp. 591–596.

[26] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The Quickhull Algorithm for Convex Hulls," *ACM Transactions on Mathematical Software*, vol. 22, no. 4, pp. 469–483, 1996.

[27] P. Koniusz, F. Yan, P. Gosselin, and K. Mikolajczyk, "Higher-order occurrence pooling for bags-of-words: Visual concept detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 2, pp. 313–326, 2017.

[28] P. Li, J. Xie, Q. Wang, and W. Zuo, "Is Second-Order Information Helpful for Large-Scale Visual Recognition?" in *Proceedings of the IEEE International Conference on Computer Vision*, dec 2017, pp. 2089–2097.

[29] B. Ferrarini, M. Waheed, S. Waheed, S. Ehsan, M. J. Milford, and K. D. McDonald-Maier, "Exploring performance bounds of visual place recognition using extended precision," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1688–1695, apr 2020.