CASSPR: Cross Attention Single Scan Place Recognition

Yan Xia[*,1,2,3,†]   Mariia Gladkova[*,1,5]   Rui Wang[4]   Qianyun Li[1]   Uwe Stilla[1]   João F. Henriques[3]   Daniel Cremers[1,2,3,5]

[1]Technical University of Munich [2]Munich Center for Machine Learning (MCML)

[3] Visual Geometry Group, University of Oxford [4] Microsoft Zurich [5] Munich Data Science Institute

{yan.xia, mariia.gladkova, stilla, cremers}@tum.de, qianyunli0@outlook.com, joao@robots.ox.ac.uk

Abstract

Place recognition based on point clouds (LiDAR) is an important component for autonomous robots or self-driving vehicles. Current SOTA performance is achieved on accumulated LiDAR submaps using either point-based or voxel-based structures. While voxel-based approaches nicely integrate spatial context across multiple scales, they do not exhibit the local precision of point-based methods. As a result, existing methods struggle with fine-grained matching of subtle geometric features in sparse single-shot LiDAR scans. To overcome these limitations, we propose CASSPR as a method to fuse point-based and voxel-based approaches using cross attention transformers. CASSPR leverages a sparse voxel branch for extracting and aggregating information at lower resolution and a point-wise branch for obtaining fine-grained local information. CASSPR uses queries from one branch to try to match structures in the other branch, ensuring that both extract self-contained descriptors of the point cloud (rather than one branch dominating), but using both to inform the output global descriptor of the point cloud. Extensive experiments show that CASSPR surpasses the state-of-the-art by a large margin on several datasets (Oxford RobotCar, TUM, USyd). For instance, it achieves AR@1 of 85.6% on the TUM dataset, surpassing the strongest prior model by ∼15%. Our code is publicly available.[1]

1. Introduction

3D place recognition and localization in a city-scale map is a fundamental challenge in allowing autonomous agents to operate effectively in realistic applications, such as autonomous driving [20, 21, 32, 33, 47, 30] and sidewalk or indoor robot navigation [17, 40, 49, 10, 29, 28].

While GPS signals can provide reliable geo-location under optimal conditions, they require external satellite information and are bound to fail when there is no direct line-
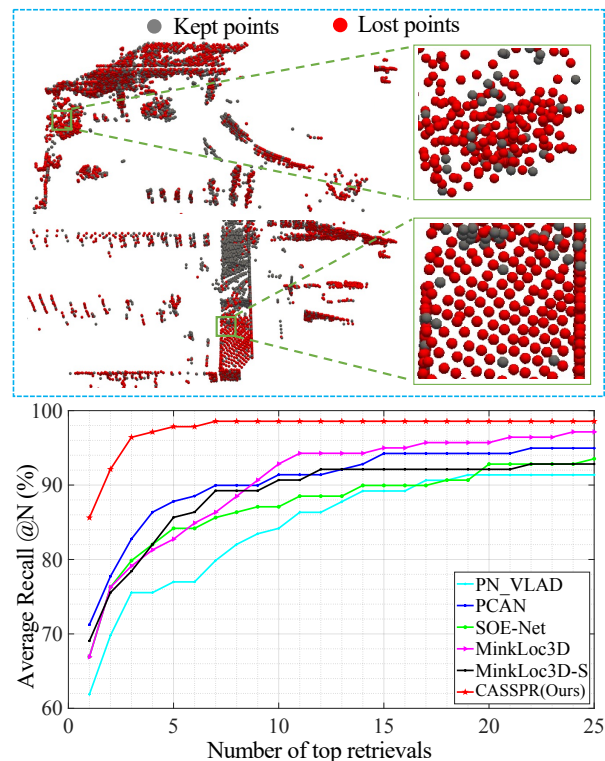

[†]Corresponding author. * Equal contribution.

[1]https://nb.oas-ua.com/Yan-Xia/CASSPR

Figure 1. **(Top)** Voxelization, which is necessary for sparse 3D convolutions [59], loses a large amount of geometric detail (visualized in red). In this example, 64% of the points are lost.
(Bottom) Place recognition performance on the TUM dataset. The proposed CASSPR delivers consistently better performance across all top retrieval numbers. Notably, at top 1 retrieval, it outperforms the best baseline by ∼15%.

of-sight and the signal is absent, such as in tunnels, parking garages, or among tall buildings and vegetation [4]. It is therefore important to develop the capability to perform localization from the on-board sensors only.

Calibrated depth sensors such as LiDAR (producing point clouds) can be used to localize the currently-observed scene by matching it against a pre-built point cloud database. Being focused on the geometry of the scene, point cloud recognition sidesteps several factors that make pure

visual (RGB image-based) matching difficult, such as variations in lighting, weather, and season, where the same geometric structure may appear entirely different.

In the past decade, a variety of point-cloud-based solutions has been proposed that achieve excellent performance. They can be classified into two broad categories: point-based descriptors and voxel-based descriptors. With several works building on PointNetVlad [2], point-based descriptors[2, 53, 35, 50] are dedicated to improving the global context aggregation (e.g. with pooling or self-attention units). However, processing points individually can miss some local context, and does not make use of flexible and robust local pattern matchers such as the filters of convolutional neural networks (CNNs). The voxel-based descriptors [27, 59] generally consist of CNNs leveraging 3D sparse convolutions over a sparse volumetric representation, which can be implemented efficiently using hash tables, most notably with the Minkowski Engine [8]. They achieve good performance with fewer parameters than other models and comparatively fast training. However, sparse voxels can only represent 1 point per voxel cell. Therefore, for larger cells, several points (possibly containing important geometric information) are necessarily lost – for example, 64% of the points in Fig. 1 (a) are lost during voxelization. Small cells, on the other hand, consume large amounts of memory – for example, 82.6 GB of GPU memory would be needed to keep 90% of the points on average [34]. It remains a difficult challenge to capture fine-grained geometric information with point representations, while enjoying the robust pattern matching at multiple context scales of voxel approaches, and without consuming too much memory.

Another challenge is that most previous descriptors have performed place recognition when given dense maps as a reference. While this is an important setting, accurate point cloud maps may not always be available, in unfamiliar scenes and when an environment changes. Ideally, one would therefore prefer localization to be performed from sparse single-shot scans, which can be collected during normal online operation of an autonomous system, instead of requiring a carefully-curated densely captured point cloud to be created first. A single LiDAR such as Ouster OS1-128 [59] can cover an area between 160 and 200 meters in diameter, collecting a relatively sparse set of points. Aggregated LiDAR scans are typically at least 10 times denser. The larger area and lower density represent a problem for performing place recognition from single LiDAR scans.

To tackle these problems, we propose a novel cross attention transformer for single-scan-based place recognition, named CASSPR, aiming to compensate for the quantization losses and integrating long-range spatial relationships. The key to maintaining geometric detail even with coarse voxelization is to use a second branch to compute features independently for each point, inspired by PointNet [42]. This branch is free to partition the input space into regions with boundaries that are not necessarily aligned with voxels, and associate different features with each one. One problem with this strategy is that naively applying a sparse convolution network to such features would still lose geometric detail by removing co-located points due to each voxel only supporting features from a single point. Therefore, we propose to fuse information from both branches, namely point-wise and sparse voxelized, with a hierarchical cross-attention transformer [46] (HCAT). This transformer can flexibly aggregate local and global information into sparse voxel features, and do so very efficiently (see Sec. 6.5). This flexibility and efficiency allows us to surpass the state-of-the-art performance in several challenging point cloud localization tasks (Sec. 6.3).

To summarize, the main contributions of this work are:

• We study the extreme sparsity of single (non-aggregated) LiDAR scans, in the context of voxel and point-based neural networks for place recognition, and analyze the complementary roles of each approach.

• We propose a 2-stage hierarchical cross-attention transformer (HCAT) module that is designed to compensate for the shortcomings of point-wise and voxel-wise features, compensating for the loss of the geometric detail caused by the spatial quantization and integrating long-range spatial relationships.

• We assess the computation and memory trade-offs of the different approaches, finding that our proposal reduces inference time and memory consumption by up to 62% and 91%, respectively, when compared to previous attention units used for LiDAR-based place recognition.

• We conduct extensive experiments on several benchmark datasets, including USyd Campus [56], Oxford Robot-Car [38] and TUM City Campus datasets [58] and show that the proposed CASSPR greatly improves over the state-of-the-art methods.

2. Related work

3D point cloud based place recognition is usually expressed as a 3D retrieval task. Numerous methods were proposed to tackle this task, which can be primarily classified into 2 categories: those based on geometric global descriptors, and those based on plane or object descriptors.

Handcrafted 3D global descriptors. Most handcrafted global descriptors describe places using global statistics, which have the advantage of not requiring re-training to adapt to different environments and sensor types. Magnusson *et al*. [39] split the point cloud into overlapping cells and computed shape properties such as spherical, linear, and several types of planar properties of each cell. Rohling *et al*. [44] propose a fast method of describing places through histograms of point elevation, assuming the sensor had a

constant height above the ground plane. Scan-Context [26] is a method that exploits a bird-eye view of the point cloud and encodes height information of the surrounding objects for place recognition. However, handcrafted descriptors cannot improve with more data, placing a hard ceiling on their performance, unlike data-driven methods.

Learning-based 3D global descriptors. With breakthroughs in learning-based image retrieval methods, deep learning of 3D global descriptors for retrieval tasks has become the focus of intense research. In PointNetVlad, Uy *et al.* [2] tackle 3D place recognition with end-to-end learning, using PointNet [42] to extract local descriptors and then aggregating them globally using NetVlad pooling over 3D points [3]. PointNet itself consists of a MLP applied to each point independently, with global max-pooling aggregation of features instead, as well as a predicted linear transform for input points [42]. Subsequently, PCAN [53] explores an attention mechanism for local features aggregation, discriminating local features that contribute positively. LPD-Net [35] enhances local contextual relationships using graph neural networks, but relying on handcrafted features. DH3D [12] introduces a deep hierarchical network to produce more discriminative descriptors, by recognizing places and refining a 3D pose estimation simultaneously. SOE-Net [50] presents a PointOE module introducing orientation encoding into PointNet for generating point-wise local descriptors. Minkloc3D [27] utilizes a Feature Pyramid Network [31] (FPN) based on 3D sparse tensors with generalized-mean (GeM) pooling [43] to compute a compact global descriptor. Several methods [57, 9, 13, 51, 54, 37, 5, 36] introduce different transformer networks (i.e. stacked self-attention blocks [48, 11]) for learning long-range contextual features. Compared to OverlapNet [7] and other attention-based methods, including OverlapTransformer [37], AttDLNet [5] and CVT-Net [36], our CASSPR directly applies to raw point clouds, rather than range images projected from LiDAR scans. The work that is more closely related to ours is Minkloc3D-SI [59], which tackled place recognition with a single 3D LiDAR scan, proposing a non-Cartesian point representation and per-point color intensity information. However, 3D voxelization methods inevitably suffer from lost points due to the quantization step (see Fig. 1).

Point-voxel fusion networks. Recent works have tried to fuse point-wise networks with voxelized representations. The PV-RCNN [45] architecture in LCDNet [6] adopts a voxel-to-keypoint encoding strategy for 3D object detection, where the features of each keypoint are aggregated by grouping the neighboring voxel-wise features. In PVCNN [34] and PVT [52], the voxel-based features are first transformed back to the domain of the point cloud and then simply added with point-wise features for 3D semantic segmentation. Both are different from our method, which

does not lose points (either by aggregation or devoxelization), but rather uses them to inform the voxel branch (using cross-attention).

Based on planes or objects. Some works have proposed using 3D shapes or object-centric recognition for the task of 3D place recognition. Fernandez *et al.* [14] first detect planes in 3D environments and then summarize their pairwise relationships into a graph. A final geometric consistency test over the planes matches sub-graphs of this structure using an interpretation tree. A follow-up work [15] utilizes the covariance of the plane parameters instead of the number of points in planes for matching. Finman *et al.* [16] propose an object-based indoor place recognition based on RGB-D cameras. However, it is only suitable for small, indoor environments with objects of particular classes.

3. Problem statement

We begin by defining the reference map $M_{\text{ref}} = \{m_i : i = 1, ..., M\}$ to be a collection of *single-shot* 3D LiDAR scans m_i. Each scan is tagged with UTM coordinates using GPS/INS at the position where it is captured [56]. Let Q be a query single LiDAR scan at one timestamp. Then, the single-scan 3D place recognition problem is defined as retrieving the scan m^* from M_{ref} that is structurally closest to Q. We aim to design a network $F(\cdot)$ that encodes a single scan as a 3D global descriptor, such that we can retrieve the correct scan $m^* \in M_{\text{ref}}$ by the nearest-neighbour search in a reference map:

$$m^* = \operatorname*{argmin}_{m_i \in M_{\text{ref}}} d(F(Q), F(m_i)), \quad (1)$$

where $d(\cdot)$ is a distance metric (e.g. the Euclidean distance). In practice, the global descriptors of all 3D scans are compiled offline in a dictionary. Given a new query scan, the nearest neighbor in a dictionary is found efficiently using a KD-tree (e.g. using FAISS [25]).

4. Methodology

Fig. 2 shows the network architecture of our CASSPR. The point branch (top) captures fine-grained geometrical features by using a simplification of a PointNet [42], described in Section 4.1. The voxel branch (bottom) performs convolution on sparse voxels in spherical coordinates (based on Minkloc3D-SI [59]) and will be explained in Section 4.2. Point-voxel feature fusion is achieved by a Hierarchical Cross-Attention Transformer (HCAT) module (Section 4.3). Finally, lightweight self-attention (LSA) units are used to encode the spatial relationship among local descriptors (Section 4.4), followed by 3D convolutions and pooling. The training strategy with the loss function is described in Section 5.

SC: Spherical Coordinates   CAT: Cross-Attention Transformer   HCAT: Hierarchical Cross-Attention Transformer   LSA: Lightweight Self-Attention
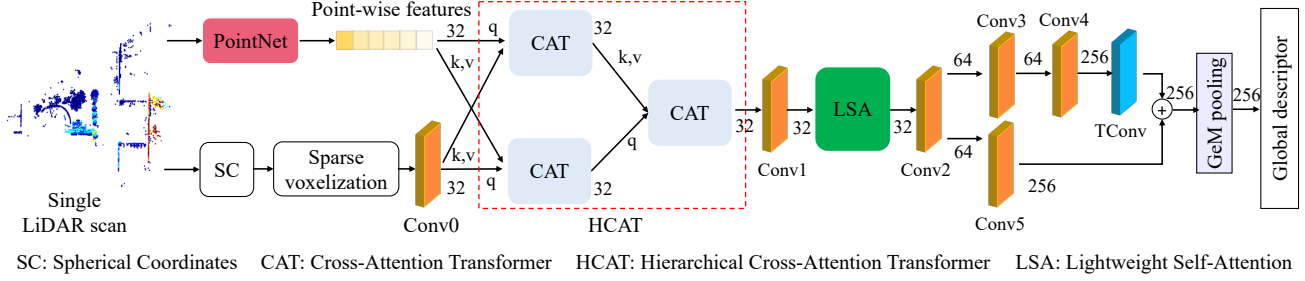
Figure 2. The proposed CASSPR architecture. It consists of two parallel branches: one is extracting features from each point independently (top) and another is using sparse convolutions over sparse voxels (bottom). Cross-attention operators use queries from one branch to look up information in the other branch, encouraging the network to focus on the commonalities between these two views. The result is then fused with cross-attention and (light-weight) self-attention, and finally processed with convolutions, deconvolutions (TConv), and GeM pooling following [27] to generate the final global descriptor. The numerical values (e.g., 32, 64) represent the number of channels of a feature map produced by each block.

4.1. Point branch

This branch is a simplification of the PointNet [42] without the max-pooling stage. To make the discussion self-contained, we describe it briefly here. It consists of a simple 2-layer MLP with 64 channels that operates on each point independently and outputs a 32-dimensional vector for each point. The input 3D points are transformed via multiplication with an adaptive 3×3 matrix, which allows the Point-Net to rotate or stretch the point cloud to align it with a preferred reference frame. This matrix is predicted by an auxiliary network that also processes each point with a 3-layer MLP, applies max-pooling over the points, and finally applies a 2-layer MLP to output the matrix's 9 elements.

The point branch has the ability to recognize fine-grained geometric features at a global level, without discretization at arbitrary voxels [42]. However, it has limited capacity to flexibly identify patterns in local neighborhoods of points.

4.2. Sparse spherical voxel branch

The density of 3D point clouds in urban scenes collected by LiDAR sensors is uneven, with the area closer to the scanner having a much greater density than far-away regions. Consequently, 3D voxelization methods based on Cartesian coordinates face a difficult tradeoff: a fine grid captures nearby details but consumes a lot of memory, limiting the spatial range of the grid; a coarse grid allows a greater range but loses detail close to the scanner. To address this, we use the spherical-voxelization method of MinkLoc3D-S [59] to balance the varying density of points in a LiDAR scan. Each 3D point (x, y, z) is transformed to spherical coordinates $(\gamma, \theta, \varphi)$, with $\gamma = \|(x, y, z)\|$ and

$$\theta = \mathrm{atan}_2(y, x), \quad \varphi = \mathrm{atan}_2(z, \sqrt{x^2 + y^2}). \quad (2)$$

The advantage of spherical coordinates is that the volume of each cell grows quadratically with its distance (γ) from the scanner, taken to be at the origin, resulting in a more even distribution of point density.

Next, we quantize the point cloud with coordinates $(\gamma, \theta, \varphi)$ into a finite number of voxels (Sparse Voxelization block in Fig. 2). This process creates a sparse tensor with 3 spatial dimensions, containing the constant 1 in a 3D voxel whenever there is a point there. This is implemented efficiently using coordinate hash functions with the Minkowski Engine [8]. A 3D convolutional block then aggregates neighboring points, producing a sparse feature map with increasing receptive fields. We explore how to efficiently fuse the features from different branches in the next sections. Note that, while the voxelization allows for robust 3D sparse convolutions to extract geometric information from a wide receptive field, it necessarily loses many details from the point cloud (visualized in Fig. 1, top).

4.3. Hierarchical cross-attention transformer

To efficiently exploit the relationship between the volumetric branch and the point branch, we further propose a Hierarchical Cross-Attention Transformer (HCAT) to fuse the features from the two branches in a unified model. As shown in Fig. 2, the HCAT consists of three Cross Attention Transformers (CAT), each having a different role. The first CAT takes the sparse-voxel features as the Query and the point-wise features as the Key and Value. It extracts point-wise features with reference to the sparse feature maps and outputs sparse tensors that are informed by the point features. Conversely, another CAT produces improved point-wise features by taking the point features as the Query and the sparse-voxel features as the Key and Value. Finally, the aim of the third CAT is to fuse both enhanced sparse tensors and enhanced point-wise features. Fig. 3 shows the architecture of the HCAT in detail. The advantage of this scheme, as opposed to simply using a single CAT with concatenated inputs from both branches, is to ensure that each branch is useful in isolation – as it must be used as the sole
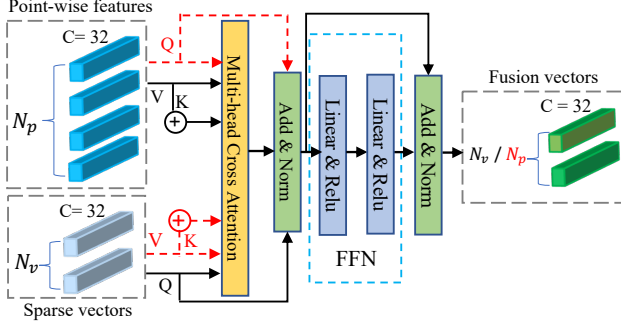
Figure 3. The architecture of the Cross-Attention Transformer. See Sec. 4.3 for details.

source of Queries for a single CAT calculation, but also independently as the sole source of Keys and Values – thus preventing one branch from dominating over the other.

Each CAT is a residual module, consisting of two sublayers: Multi-Head Cross-Attention (MHCA) and Feed-Forward Network (FFN). The feed-forward network includes two linear transformation layers with the ReLU activation function. Formally, each CAT can be formulated as:

$$\mathbf{F}_{CA} = \text{CAT}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$$
$$= \widetilde{\mathbf{F}}_{CA} + \text{FFN}\left(\widetilde{\mathbf{F}}_{CA}\right), \quad (3)$$
$$\widetilde{\mathbf{F}}_{CA} = \mathbf{Q} + \text{MHCA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}),$$

where $\mathbf{Q} = F_s \in \mathbb{R}^{N_s \times d}$ is the query, $\mathbf{K} = \mathbf{V} = F_p \in \mathbb{R}^{N_p \times d}$ are the key and value.

In the MHCA layer, cross-attention is performed by projecting the $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ with h heads (we use $h = 2$). Specially, we first calculate the weight matrix with scaled dot-product attention [46] for the Query, Key, and Value:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{S}\left(\frac{\mathbf{Q} + \mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}, \quad (4)$$

where S is a Softmax function outputting attention weights. Next, we calculate the values for h heads and concatenate them together:

$$\text{Multi-Head}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [\,\text{head}_1, \ldots, \text{head}_h\,]\mathbf{W}^O, \quad (5)$$
$$\text{head}_i = \text{Attention}\left(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V\right), \quad (6)$$

where $\mathbf{W}_i^{\{Q,K,V,O\}}$ are learnable parameters.

With the above HCAT, point-wise features from the point branch can attend to the sparse tensors from the voxel-based branch, and vice-versa, thus compensating for the loss of points due to the voxel quantization.

4.4. Lightweight self-attention unit

Before going into details concerning the Lightweight Self-Attention (LSA) unit, we give a short review of the

standard self-attention used in previous 3D place recognition or point Transformers [55, 19]. Given a set of points with features $D_L = \{(p_i, f_i)\}_{i=1}^N$, where p_i is the coordinate of the i-th point (of a total of N) and $f_i \in \mathbb{R}^C$ is the feature of p_i. A dot-product self-attention (DPSA) unit on p_i can be formulated as follows:

$$\text{DPSA}(p_i) = \sum_{j \in \chi(i)} \text{S}[\delta(p_i - p_j), f_i]\varphi(f_j), \quad (7)$$

where δ is a linear positional encoding function, φ is a learnable value projection layer, and $\chi(i)$ denotes the indices of neighboring points of p_i.

However, computing DPSA directly by multiplying two $N \times C$ matrices is expensive, with $O(N^2)$ space complexity and $O(N^{2.34})$ time complexity [1]. Additionally, nearest neighbors search with a KD-tree has $O(N \log N)$ space complexity [41]. In this work, we reduce the training and inference time/memory consumption using a Lightweight Self-Attention (LSA) unit, inspired by [41], which we describe next.

Efficient positional encoding. We first voxelize the input point cloud to a set of M triplets $\mathbb{V} = \{(v_i, g_i, c_i)\}_{i=1}^M$, including the i-th voxel coordinate v_i, the corresponding voxel feature g_i, and the centroid coordinate c_i of this voxel. Note that $M < N$ since the sparse voxelization step will lose most points. LSA on c_i is formulated from Eq. 7:

$$\text{LSA}(c_i) = \sum_{j \in \tau(i)} \text{S}[\delta(c_i - c_j), g_i]\varphi(g_j), \quad (8)$$

where $\tau(i)$ represents the neighbor voxel indexes of c_i.

Finding neighboring K voxels via voxel hashing will be quick since it only has $O(M)$ time complexity. However, implementing $\delta(c_i, c_j)$ directly as a linear function requires $O(MKD)$ space complexity. To alleviate this problem, a coordinate decomposition approach is used inspired by [41]. Given a query voxel (v_i, g_i, c_i) and a nearest neighbor voxel (v_j, g_j, c_j), the relative position encoding between c_i and c_j can be decomposed as follows:

$$\delta(c_i - c_j) = \delta[(c_i - v_i) + (v_i - v_j) - (c_j - v_j)]. \quad (9)$$

As seen in Eq. 9, the memory-consuming $\delta(c_i - c_j)$ is decomposed in three parts, since δ is a linear operator. The space complexity of $\delta(c_i, c_j)$ is reduced from $O(MKD)$ to $O(MD + KD)$ because the space complexity of $\delta(c_i - v_j)$ and $\delta(v_i - v_j)$ will be $O(MD)$ and $O(KD)$. The decomposition is illustrated in detail in Fig. 4.

Unfortunately, using the softmax function in Eq. 8 will often result in degenerate weights due to the sparsity of the inputs. The input size is variable and often an input only contains a single point or none. When $|\tau_i|$ is 1 (i.e. a single point in the neighborhood), the attention weights are normalized into the constant 1, so LSA will be downgraded to
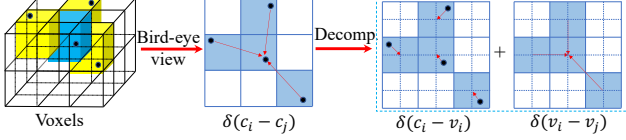
Figure 4. Decomposition of positional encoding. We decompose the positional encoding $\delta(c_i - c_j)$ to $\delta(c_i - v_i)$ and $\delta(v_i - v_j)$. This decomposition can reduce the space complexity from $O(MKD)$ to $O(MD + KD)$.

a linear function φ. To prevent this, we replace the softmax with cosine similarity in Eq. 8, rewriting it as:

$$\text{LSA}(c_i) = \sum_{j \in \tau_i} \frac{\delta^T(c_i - c_j)g_i}{\|\delta(c_i - c_j)\| \|g_i\|} \varphi(g_j) \qquad (10)$$

After the LSA operation, the result is decoded with 3D convolutional and deconvolution operations, with a coarse residual block inspired by Feature Pyramid Networks [31] (illustrated in Fig. 2). Finally, generalized mean pooling (GeM) [43] aggregates over all voxels to obtain a single 256-dimensional embedding vector. GeM is defined as $e_k = \left(\frac{1}{n} \sum_j^n h_{kj}^p \right)^{1/p}$, given the k-th feature of the j-th voxel h_{kj} and a learnable exponent parameter p.

5. Loss function

Similarly to Zywanowski et al. [59], we use a triplet margin loss based on the batch hard negative mining approach. A batch of scans with size n is built by sampling $n/2$ structurally similar scans. The batch is then used as the input of our network to compute global descriptors. Notably, two $n \times n$ boolean masks are computed, one indicating structurally similar pairs and the other structurally dissimilar pairs. Informative training triplets are constructed based on the boolean masks. Additionally, we only mine the hardest positive and the hardest negative scans in one batch. The hard mining triplet loss can be formulated as:

$$L = \max\left(m + d\left(\psi_a, \psi_p\right) - d\left(\psi_a, \psi_n\right), 0\right) \qquad (11)$$

where ψ_a is an anchor point cloud, ψ_p a hard positive point cloud (structurally similar to the anchor), ψ_n a hard negative point cloud (structurally dissimilar to the anchor). m is a pre-defined margin parameter.

6. Experiments

6.1. Datasets

To showcase the performance of our CASSPR on single LiDAR scan data, we first train and evaluate CASSPR on two public LiDAR datasets: USyd Campus [56] and TUM City Campus [58] dataset. To test the generalization ability and compare with other methods fairly, we also train and

evaluate on Oxford RobotCar [38]. Different with USyd and TUM dataset, the point clouds in Oxford RobotCar are generated from 2D scans accumulated over time.

USyd Campus Dataset. The USyd Campus Dataset (USyd) [56] contains data collected by a small electric car driving the same route around the university of Sydney campus over 50 weeks in varying weather conditions. The array of sensors includes a Velodyne VLP-16 LiDAR, six cameras, and GPS/IMU. Following [59], we split consecutive LiDAR scans at intervals of 5 meters traveled distance, resulting in about 735 scans each run. Each scan covers 100 meters in diameter. The point clouds are downsampled using the voxel grid filter to 4096 points. Notably, the ground plane is not removed.

Oxford RobotCar Dataset. The Oxford RobotCar dataset [38] consists of data collected by a SICK LMS-151 2D LiDAR scanner recorded over a year, with a total length over $1000\,\text{km}$. Point clouds in Oxford RobotCar are generated from 2D scans accumulated over time to create a unified 3D map. The 3D map is divided into submaps with $20\,\text{m}$ length, and each submap includes exactly 4096 equally distributed points.

TUM City Campus Dataset. The TUM City Campus (TUM) dataset [58] contains two recordings taken at the city campus of the Technical University of Munich. Two Velodyne HDL-64E LiDAR sensors are mounted on Mobile Distributed Situation Awareness (MODISS). The acquisition resulted in more than 10,500 scans for each run. A single scan includes $130\,\text{K}$ points per rotation and covers a large area of 120 meters in diameter. The point cloud scans are split roughly every 5 meters without overlapping based on the scanner positions. Ground removal is applied to remove uninformative patterns. We then downsample the point clouds to 4096 points using the voxel grid filter.

In-house datasets. We validate the generalization capabilities of networks trained only on Oxford RobotCar by utilizing three additional in-house datasets collected in a university sector (U.S.), a residential area (R.A.), and a business district (B.D.) with a sensor Velodyne-64 Li-DAR [2]. The datasets contain single-scan sequences of lengths $10\,\text{km}$, $8\,\text{km}$, and $5\,\text{km}$, respectively.

6.2. Evaluation criteria

Following [2], Average Recall at Top N (AR@N) is used as an evaluation metric, which means the location is correctly recognized if the N most similar scans matched from the database contain at least one location within the distance d from the query. Top 1 counts the number of times the first match from the database matches the query location. We also present the Average Recall for Top 1% (AR@1%) results for comparison to the state-of-the-art solutions. For the TUM dataset, we regard the retrieved scan as a correct match if the distance is within $d = 5m$. For the USyd, Ox-

Table 1. Average recall (%) at top 1% (@1%) and top 1 (@1) for each model trained on the USyd and TUM dataset. * indicates the results reported in MinkLoc3D-S [59].

	USyd dataset		TUM dataset	
	AR @1%	AR @1	AR @1%	AR @1
PointNetVlad	81.7	60.7	76.3	61.9
PCAN	86.4	68.7	87.8	71.2
SOE-Net	78.9	52.8	83.5	66.9
MinkLoc3D	98.1*	91.7*	82.7	66.9
MinkLoc3D-S	98.8*	93.9*	85.7	69.1
CASSPR	**98.9**	**97.6**	**97.1**	**85.6**

Table 2. Average recall (%) at top 1% (@1%) and top 1 (@1) for each of the models trained on the Oxford RobotCar. Our CASSPR achieves the best performance on all benchmarks.

AR @ {1, 1%}	Oxford		U.S.		R.A.		B.D.	
	1	1%	1	1%	1	1%	1	1%
PN-VLAD [2]	62.8	80.3	63.2	72.6	56.1	60.3	57.2	65.3
PCAN [53]	69.1	83.8	62.4	79.1	56.9	71.2	58.1	66.8
LPD-Net [35]	86.3	94.9	87.0	96.0	83.1	90.5	82.5	89.1
EPC-Net [23]	86.2	94.7	-	96.5	-	88.6	-	84.9
HiTPR [22]	86.6	93.7	80.9	90.2	78.2	87.2	74.3	79.8
SOE-Net [50]	89.4	96.4	82.5	93.2	82.9	91.5	83.3	88.5
MinkLoc3D [27]	93.0	97.9	86.7	95.0	80.4	91.2	81.5	88.5
NDT-T [57]	93.8	97.7	-	-	-	-	-	-
PPT-Net [24]	93.5	98.1	90.1	97.5	84.1	93.3	84.6	90.0
SVT-Net [13]	93.7	97.8	90.1	96.5	84.3	92.7	85.5	90.7
MinkLoc3D-S [59]	92.8	81.7	83.1	67.7	72.6	57.1	70.4	62.2
CASSPR (Ours)	**95.6**	**98.5**	**92.9**	**97.9**	**89.5**	**94.8**	**87.9**	**92.1**

ford RobotCar, and In-house datasets, the distance is set as 10m, 25m, and 25m, respectively.

6.3. Results

6.3.1 Comparisons on the USyd and TUM Dataset.

We compare our CASSPR with the state-of-the-art methods: PointNetVlad [2], PCAN [53], SOE-Net [50], MinkLoc3D [27], and MinkLoc3D-S [59]. For a fair comparison, we re-trained and tested on the USyd dataset using publicly-available code. Table 1 (left) shows the top 1% and top 1 recall of each method on the USyd datasets. CASSPR achieves the best performance of 98.9% / 97.6% at AR@1% / AR@1, exceeding the performance of the previous SOTA MinkLoc3D-S by 3.9% on AR@1. It demonstrates that CASSPR can generate more discriminative global descriptors compared with point-based or voxel-based baselines.

We also conduct experiments on the TUM dataset. The results in Table 1 (right) show that the proposed CASSPR outperforms others significantly. Notably, CASSPR achieves the recall of 97.1% at top 1%, exceeding the recall of the current state-of-the-art method by 9.3%. Furthermore, our CASSPR achieves an average recall of 85.6% at top 1, which has a significant advantage (16.5%) over MinkLoc3D-S. Fig. 1 (**Bottom**) shows the recall curves of PointNetVLAD, PCAN, SOE-Net, Minkloc3D, MinkLoc3D-S, and ours for the top 25 retrieval results. The plot demonstrates a consistently superior performance of CASSPR over the compared baselines. Comparing the results on the USyd and TUM benchmarks, we notice CASSPR performs much better on the latter. This might be caused by the fact that the TUM dataset has less data compared to the USyd dataset with only two recorded runs.

6.3.2 Comparisons on the Oxford RobotCar datasets.

To further demonstrate the capabilities of our CASSPR, we also conduct experiments on benchmark datasets introduced in [2]. Following the baseline networks proposed in [2, 53, 35, 23, 50, 27, 57, 24, 13, 22], we train and evaluate on the Oxford RobotCar. Three in-house datasets are used to verify the generalization ability of models on unseen scenarios. We use the same settings in Minkloc3D, except that

we add the LSA unit followed by each 3D convolution layer. The HCAT is removed because the spherical representation is not suitable for 2D aggregated scans, as verified in [59].

We compare CASSPR with the state-of-the-art methods, including PointNetVLAD [2], PCAN [53], LPD-Net [35], EPC-Net [23], SOE-Net [50], Minkloc3D [27], NDT-Transformer [57], PPT-Net [24], SVT-Net [13], and HiTPR [22]. The dimensions of all global descriptors are set to 256. The evaluation results are shown in Table. 2. CASSPR achieves state-of-the-art results on the Oxford RobotCar, with 2.6% improvements at AR@1 over MinkLoc3D. Compared with NDT-Transformer, PPT-Net, SVT-Net, and HiTPR, which are all based on Transformers, our CASSPR still achieves a remarkable improvement. Compared with PCAN and SOE-Net, CASSPR exceeds the recall of PCAN and SOE-Net by 26.5% and 6.2% at AR@1, respectively. This suggests that the proposed hierarchical attention with a point branch is more effective than the attention strategies used in PCAN and SOE-Net.

In addition, CASSPR surpasses other methods significantly on the in-house datasets, despite the existence of a large domain shift. This demonstrates that the global descriptors generated by CASSPR can generalize better than the previous state-of-the-art methods.

6.4. Ablation study

To assess the relative contribution of each module, we remove the LSA and the HCAT (including the point branch) from our network one by one, denoted as CASSPR_HCAT and CASSPR_LSA, respectively. We also switch the key/value and query for the fusion unit of HCAT, namely the point branch acting as a query. and the voxel branch as a key and value, denoted as CASSPR_Switch. All networks are trained on the TUM dataset, with results shown in Table 3. CASSPR_HCAT outperforms MinkLoc3D-S in recall by 8.6%, indicating the proposed HCAT is a crucial part

of a successful fusion strategy. The LSA unit brings significant improvements on the average recall metric (7.9%), when compared against MinkLoc3D-S performance. The results are consistent with our expectation that compensating for quantization losses and introducing long-range contexts is essential for generating a more discriminative global descriptor. Combining both modules achieves the best performance, improving the performance by 11.4% and 16.5%. We also observe inferior performance of CASSPR_Switch to the proposed method when changing the sequence of key/value and query in the second stage. Specifically, CASSPR_Switch achieves top 1% recall of 98.1% on the USyd dataset and 89.2% on the TUM dataset, lower than the accuracy achieved by the proposed CASSPR of 98.7% and 97.1% respectively. We thus conclude the voxel branch plays a more significant role compared to the point branch when fusing the features.

Impact of sparsity. We also studied how performance changes as we vary the sparsity of the point clouds. The horizontal scanning angle θ (presented in Sec. 4.2) is chosen to determine the number of points fed into the network. Specifically, more points would be lost as $\Delta\theta$ increases. As we vary $\Delta\theta$ the other two parameters are kept constant at their default values, $\Delta r = 2.5$ and $\Delta\varphi = 1.875$. Fig. 5 shows the influence of increasing the quantization parameter $\Delta\theta$ on the performance of our CASSPR and two baselines from both point-based and voxel-based communities, PCAN [53] and MinkLoc3D-S [59]. We randomly choose 10 sequences from the USyd dataset [56] for the study. We split all scans into train and test sets analogously to [59]. While MinkLoc3D-S naturally quantizes input point clouds in order to construct sparse input tensors, PCAN relies entirely on the input scan data. Therefore, to fairly compare point-based and voxel-based approaches, we manually quantize point clouds based on the chosen quantization parameters before feeding them to PCAN. PCAN is significantly affected by the point cloud quantization with a larger azimuthal angle (Fig. 5). In comparison to the voxel-based MinkLoc3D-S, our CASSPR shows consistently higher accuracy and robustness to increasingly sparse data. Specifically, a larger horizontal angular size (6 to 25 degrees) results in the AR@1 decrease of $\approx 3\%$, while the impact is more significant for MinkLoc3D-S with $\approx 6\%$. We attribute this effect to the interplay between the point-based and voxel-based branches of our method. While the voxel-based branch becomes weaker and less informative with increasing sparseness, the point-based branch plays a more influential role thanks to the mediation of the proposed hierarchical cross-attention unit.

6.5. Computational cost analysis

In this section, we analyze the required computational resources of different global descriptors in terms of the num-
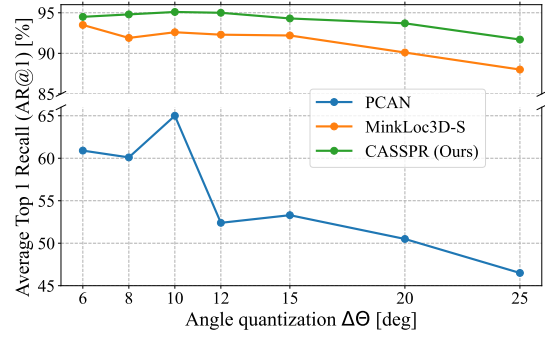

Figure 5. The impact of increasing azimuthal angle quantization $\Delta\theta$ on the average top 1 recall (AR@1) for MinkLoc3D-S [59], PCAN [53] and our CASSPR on the USyd sequences [56]. While point-based PCAN is significantly affected by larger voxelization ranges, CASSPR remains robust across a wide quantization spectrum. Our method is also consistently more accurate than MinkLoc3D-S.

Table 3. Ablations of the hierarchical cross-attention transformer (HCAT) and lightweight self-attention (LSA) on the TUM dataset.

Method	HCAT	LSA	Ave recall @1%	Ave recall @1
MinkLoc3D [27]			82.7	66.9
MinkLoc3D-S [59]			85.7	69.1
CASSPR_HCAT	✓		94.3	77.7
CASSPR_LSA		✓	93.6	84.9
CASSPR_Switch	✓	✓	89.2	77.7
CASSPR (Ours)	✓	✓	**97.1**	**85.6**

Table 4. Computational cost requirements of different 3D global descriptors on the TUM dataset. * indicates the consumption including the HCAT and an LSA.

Methods	Parameters (M)		Time Usage (ms)			Performance
	Total	Attention	Total	Backbone	Attention	AR @1%
PointNetVLAD [2]	19.8	N/A	12.6	12.6	N/A	76.3
PCAN [53]	20.4	0.6	66.8	27.8	39.0	87.8
SOE-Net [50]	19.4	1.6	66.9	29.8	37.1	83.5
MinkLoc3D-S [59]	**1.1**	N/A	**6.0**	6.0	N/A	85.7
CASSPR (Ours)	3.8	**0.057***	29.7	15.6	14.1*	**97.1**

ber of parameters and time efficiency. For a fair comparison, all methods are tested on the TUM dataset with a single NVIDIA V100 (32G) GPU. As shown in Table 4, our CASSPR takes $29.7\,\mathrm{ms}$ to encode one scan into a global descriptor with only $3.8\,\mathrm{M}$ parameters. Although CASSPR is a point-voxel fusion architecture, it has lower trainable parameters compared with the pure point-based methods, including PointNetVLAD, PCAN, and SOE-Net. For inference time, CASSPR is faster than the attention-based methods PCAN and SOE-Net per scan ($29.7\,\mathrm{ms}$ vs. $66.8\,\mathrm{ms}$). In addition, our attention units, including the HCAT and LSA, are significantly more lightweight (91% improvement in memory) and faster (62% improvement in inference time). However, compared with pure voxel-based MinkLoc3D-S, CASSPR has more parameters and running time due to the extra point branch ($5\,\mathrm{ms}$), HCAT module ($12.7\,\mathrm{ms}$) and
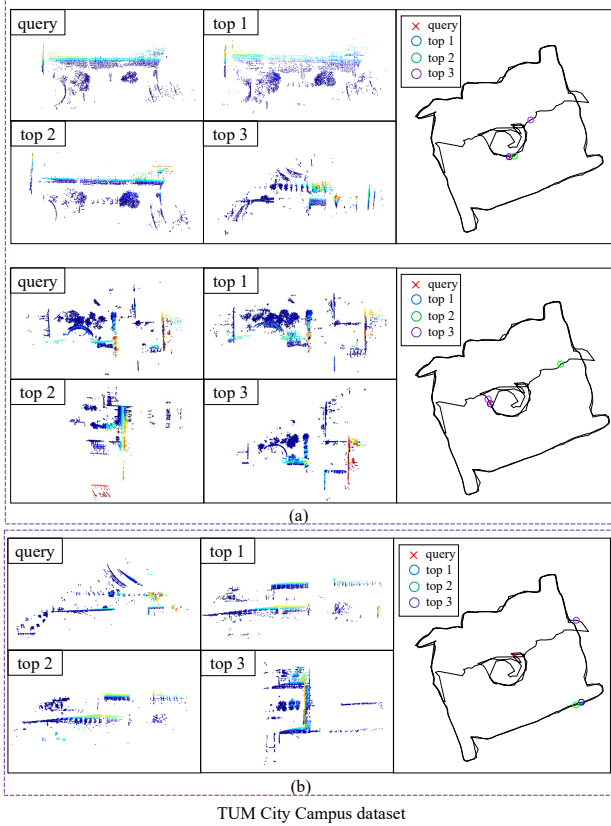
TUM City Campus dataset

Figure 6. Example retrieval results of CASSPR on the TUM dataset, best viewed on a screen. For each retrieval, the query point cloud and top 3 retrievals are shown. The point clouds' origins are also indicated in a reference map on the right. a) shows the successful retrieval cases, while b) shows a failure case.

LSA ($1.4\,\mathrm{ms}$) unit.

6.6. Results visualization on the TUM dataset

In addition to quantitative results, we show qualitative results of two correctly retrieved matches and one failure case in Fig. 6. A full traversal is made as the reference map on the TUM dataset. Then we choose three query point clouds from the traversal, with each representing a single scan from the testing areas. For each scan, the query point cloud and the top 3 retrieved matches are shown on the left. It is clear that the best match has a very similar geometry to the query point cloud. We also display the location of each point cloud in the reference map on the right. For each query, the location of the top 1 result (indicated by the blue circle) is correctly overlapped with the query location (represented by the red cross).

6.7. Evaluation of loop closure detection

In this section, we evaluate loop closure detection performance on the KITTI Odometry benchmark [18], sequences

Table 5. Comparison of loop closure detection performance on the KITTI Odometry benchmark. The sequence number is reported in parentheses, e.g. (00).

	AP (00)	AR@1 (00)	AUC (00)	F1 (08)
LCDNet [6]	**0.97**	-	-	-
OverlapNet [7]	-	81.6	86.7	-
OverlapTransfomer [37]	-	90.6	90.7	-
CVTNet [36]	-	-	91.1	-
AttDLNet [5]	-	-	-	0.13
CASSPR (Ours)	0.96	**97.9**	**92.9**	**0.30**

00 and 08, using the same evaluation protocol as in the previous works [6, 7, 37, 36, 5]. The performance is reported in Table 5 above. Note that only values that are available in the original publications are reported. CASSPR achieves the best performance, and only for Average Precision (AP) on the sequence 00, our method is on par with LCDNet [6].

7. Conclusion

We proposed CASSPR as a cross attention transformer for single scan based place recognition. In a dual-branch hierarchical cross attention transformer, it combines both the multi-scale spatial context of voxel-based approaches with the local precision of point-based approaches. This way, we compensate for quantization losses of voxel-based approaches and introduce long-range context dependency. Extensive experiments demonstrate that CASSPR improves the retrieval performance over the state-of-the-art significantly. In particular, the results confirm that CASSPR is robust to quantization losses. Future work will explore its integration into Simultaneous Localization and Mapping (SLAM) pipelines for efficient temporal aggregation of point clouds and robustness to moving objects.

References

[1] Josh Alman and Virginia Vassilevska Williams. A refined laser method and faster matrix multiplication. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 522–539. SIAM, 2021. 5

[2] Mikaela Angelina Uy and Gim Hee Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4470–4479, 2018. 2, 3, 6, 7, 8

[3] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016. 3

[4] Roberto Arroyo Contera et al. Topological place recognition for life-long visual localization. 2017. 1

[5] Tiago Barros, Luís Garrote, Ricardo Pereira, Cristiano Premebida, and Urbano J Nunes. Attdlnet: Attention-based deep network for 3d lidar place recognition. In *Iberian Robotics conference*, pages 309–320. Springer, 2022. 3, 9

[6] Daniele Cattaneo, Matteo Vaghi, and Abhinav Valada. Lcdnet: Deep loop closure detection and point cloud registration for lidar slam. *IEEE Transactions on Robotics*, 38(4):2074–2093, 2022. 3, 9

[7] Xieyuanli Chen, Thomas Läbe, Andres Milioto, Timo Röhling, Olga Vysotska, Alexandre Haag, Jens Behley, and Cyrill Stachniss. Overlapnet: Loop closing for lidar-based slam. *arXiv preprint arXiv:2105.11344*, 2021. 3, 9

[8] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 2, 4

[9] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 195–205, 2018. 3

[10] Zhiyang Dou, Cheng Lin, Rui Xu, Lei Yang, Shiqing Xin, Taku Komura, and Wenping Wang. Coverage axis: Inner point selection for 3d shape skeletonization. In *Computer Graphics Forum*, volume 41, pages 419–432. Wiley Online Library, 2022. 1

[11] Zhiyang Dou, Qingxuan Wu, Cheng Lin, Zeyu Cao, Qiangqiang Wu, Weilin Wan, Taku Komura, and Wenping Wang. Tore: Token reduction for efficient human mesh recovery with transformer. *arXiv preprint arXiv:2211.10705*, 2022. 3

[12] Juan Du, Rui Wang, and Daniel Cremers. Dh3d: Deep hierarchical 3d descriptors for robust large-scale 6dof relocalization. In *European Conference on Computer Vision*, pages 744–762. Springer, 2020. 3

[13] Zhaoxin Fan, Zhenbo Song, Hongyan Liu, Zhiwu Lu, Jun He, and Xiaoyong Du. Svt-net: Super light-weight sparse voxel transformer for large scale place recognition. AAAI, 2022. 3, 7

[14] Eduardo Fernández-Moral, Walterio Mayol-Cuevas, Vicente Arevalo, and Javier Gonzalez-Jimenez. Fast place recognition with plane-based maps. In *2013 IEEE International Conference on Robotics and Automation*, pages 2719–2724. IEEE, 2013. 3

[15] Eduardo Fernández-Moral, Patrick Rives, Vicente Arévalo, and Javier González-Jiménez. Scene structure registration for localization and mapping. *Robotics and Autonomous Systems*, 75:649–660, 2016. 3

[16] Ross Finman, Liam Paull, and John J Leonard. Toward object-based place recognition in dense rgb-d maps. In *ICRA Workshop Visual Place Recognition in Changing Environments*, volume 76, page 480, 2015. 3

[17] Yanping Fu, Qingan Yan, Long Yang, Jie Liao, and Chunxia Xiao. Texture mapping for 3d reconstruction with rgb-d sensor. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4645–4653, 2018. 1

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 9

[19] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021. 5

[20] Christian Häne, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer, Paul Furgale, Torsten Sattler, and Marc Pollefeys. 3d visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection. *Image and Vision Computing*, 68:14–27, 2017. 1

[21] Gim Hee Lee, Friedrich Faundorfer, and Marc Pollefeys. Motion estimation for self-driving cars with a generalized camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2746–2753, 2013. 1

[22] Zhixing Hou, Yan Yan, Chengzhong Xu, and Hui Kong. Hitpr: Hierarchical transformer for place recognition in point cloud. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2612–2618, 2022. 7

[23] Le Hui, Mingmei Cheng, Jin Xie, Jian Yang, and Ming-Ming Cheng. Efficient 3d point cloud feature learning for large-scale place recognition. *IEEE Transactions on Image Processing*, 31:1258–1270, 2022. 7

[24] Le Hui, Hang Yang, Mingmei Cheng, Jin Xie, and Jian Yang. Pyramid point cloud transformer for large-scale place recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6098–6107, 2021. 7

[25] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. 3

[26] Giseop Kim and Ayoung 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)*, pages 4802–4809. IEEE, 2018. 3

[27] Jacek Komorowski. Minkloc3d: Point cloud based large-scale place recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1790–1799, 2021. 2, 3, 4, 7, 8

[28] Haoang Li, Yazhou Xing, Ji Zhao, Jean-Charles Bazin, Zhe Liu, and Yun-Hui Liu. Leveraging structural regularity of atlanta world for monocular slam. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2412–2418, 2019. 1

[29] Haoang Li, Jian Yao, Jean-Charles Bazin, Xiaohu Lu, Yazhou Xing, and Kang Liu. A monocular slam system leveraging structural regularity in manhattan world. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2518–2525, 2018. 1

[30] Haoang Li, Ji Zhao, Jean-Charles Bazin, Pyojin Kim, Kyungdon Joo, Zhenjun Zhao, and Yun-Hui Liu. Hong kong world: Leveraging structural regularity for line-based slam.

IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 1–18, 2023. 1

[31] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. 3, 6

[32] Liu Liu, Hongdong Li, Yuchao Dai, and Quan Pan. Robust and efficient relative pose with a multi-camera system for autonomous driving in highly dynamic environments. *IEEE Transactions on Intelligent Transportation Systems*, 19(8):2432–2444, 2017. 1

[33] Weiquan Liu, Baiqi Lai, Cheng Wang, Xuesheng Bian, Chenglu Wen, Ming Cheng, Yu Zang, Yan Xia, and Jonathan Li. Matching 2d image patches and 3d point cloud volumes by learning local cross-domain feature descriptors. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 516–517. IEEE, 2021. 1

[34] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. *Advances in Neural Information Processing Systems*, 32, 2019. 2, 3

[35] Zhe Liu, Shunbo Zhou, Chuanzhe Suo, Peng Yin, Wen Chen, Hesheng Wang, Haoang Li, and Yun-Hui Liu. Lpd-net: 3d point cloud learning for large-scale place recognition and environment analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2831–2840, 2019. 2, 3, 7

[36] Junyi Ma, Guangming Xiong, Jingyi Xu, and Xieyuanli Chen. Cvtnet: A cross-view transformer network for place recognition using lidar data. *arXiv preprint arXiv:2302.01665*, 2023. 3, 9

[37] Junyi Ma, Jun Zhang, Jintao Xu, Rui Ai, Weihao Gu, and Xieyuanli Chen. Overlaptransformer: An efficient and yaw-angle-invariant transformer network for lidar-based place recognition. *IEEE Robotics and Automation Letters*, 7(3):6958–6965, 2022. 3, 9

[38] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017. 2, 6

[39] Martin Magnusson, Henrik Andreasson, Andreas Nüchter, and Achim J Lilienthal. Automatic appearance-based loop detection from three-dimensional laser data using the normal distributions transform. *Journal of Field Robotics*, 26(11-12):892–914, 2009. 2

[40] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 1

[41] Chunghyun Park, Yoonwoo Jeong, Minsu Cho, and Jaesik Park. Fast point transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16949–16958, 2022. 5

[42] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2, 3, 4

[43] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1655–1668, 2018. 3, 6

[44] Timo Röhling, Jennifer Mack, and Dirk Schulz. A fast histogram-based similarity measure for detecting loop closures in 3-d lidar data. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 736–741. IEEE, 2015. 2

[45] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. 3

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 5

[47] Qiangqiang Wu, Jia Wan, and Antoni B. Chan. Progressive unsupervised learning for visual object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2993–3002, June 2021. 1

[48] Qiangqiang Wu, Tianyu Yang, Ziquan Liu, Baoyuan Wu, Ying Shan, and Antoni B Chan. Dropmae: Masked autoencoders with spatial-attention dropout for tracking tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14561–14571, 2023. 3

[49] Yaqi Xia, Yan Xia, Wei Li, Rui Song, Kailang Cao, and Uwe Stilla. Asfm-net: Asymmetrical siamese feature matching network for point completion. In *Proceedings of the 29th ACM international conference on multimedia*, pages 1938–1947, 2021. 1

[50] Yan Xia, Yusheng Xu, Shuang Li, Rui Wang, Juan Du, Daniel Cremers, and Uwe Stilla. Soe-net: A self-attention and orientation encoding network for point cloud based place recognition. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11348–11357, 2021. 2, 3, 7, 8

[51] Tian-Xing Xu, Yuan-Chen Guo, Zhiqiang Li, Ge Yu, Yu-Kun Lai, and Song-Hai Zhang. Transloc3d: point cloud based large-scale place recognition using adaptive receptive fields. *Communications in Information and Systems*, 23(1):57–83, 2023. 3

[52] Cheng Zhang, Haocheng Wan, Xinyi Shen, and Zizhao Wu. Pvt: Point-voxel transformer for point cloud learning. *International Journal of Intelligent Systems*, 37(12):11985–12008, 2022. 3

[53] Wenxiao Zhang and Chunxia Xiao. Pcan: 3d attention map learning using contextual information for point cloud based retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12436–12445, 2019. 2, 3, 7, 8

[54] Wenxiao Zhang, Huajian Zhou, Zhen Dong, Qingan Yan, and Chunxia Xiao. Rank-pointretrieval: Reranking point

cloud retrieval via a visually consistent registration evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 2022. 3

[55] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. 5

[56] Wei Zhou, Julie Stephany Berrio, Charika De Alvis, Mao Shan, Stewart Worrall, James Ward, and Eduardo Nebot. Developing and testing robust autonomy: The university of sydney campus data set. *IEEE Intelligent Transportation Systems Magazine*, 12(4):23–40, 2020. 2, 3, 6, 8

[57] Zhicheng Zhou, Cheng Zhao, Daniel Adolfsson, Songzhi Su, Yang Gao, Tom Duckett, and Li Sun. Ndt-transformer: Large-scale 3d point cloud localisation using the normal distribution transform representation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5654–5660. IEEE, 2021. 3, 7

[58] Jingwei Zhu, Joachim Gehrung, Rong Huang, Björn Borgmann, Zhenghao Sun, Ludwig Hoegner, Marcus Hebel, Yusheng Xu, and Uwe Stilla. Tum-mls-2016: An annotated mobile lidar dataset of the tum city campus for semantic point cloud interpretation in urban areas. *Remote Sensing*, 12(11):1875, 2020. 2, 6

[59] Kamil Żywanowski, Adam Banaszczyk, Michał R Nowicki, and Jacek Komorowski. Minkloc3d-si: 3d lidar place recognition with sparse convolutions, spherical coordinates, and intensity. *IEEE Robotics and Automation Letters*, 7(2):1079–1086, 2021. 1, 2, 3, 4, 6, 7, 8