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

That's My Point: Compact Object-centric LiDAR Pose Estimation for Large-scale Outdoor Localisation

Georgi Pramatarov, Matthew Gadd, Paul Newman, and Daniele De Martini
Mobile Robotics Group (MRG), University of Oxford
{georgi,mattgadd,pnewman,daniele}@robots.ox.ac.uk

***Abstract*— This paper is about 3D pose estimation on LiDAR scans with extremely minimal storage requirements to enable scalable mapping and localisation. We achieve this by clustering all points of segmented scans into semantic objects and representing them only with their respective centroid and semantic class. In this way, each LiDAR scan is reduced to a compact collection of four-number vectors. This abstracts away important structural information from the scenes, which is crucial for traditional registration approaches. To mitigate this, we introduce an object-matching network based on self- and cross-correlation that captures geometric and semantic relationships between entities. The respective matches allow us to recover the relative transformation between scans through weighted Singular Value Decomposition (SVD) and RANdom SAmple Consensus (RANSAC). We demonstrate that such representation is sufficient for metric localisation by registering point clouds taken under different viewpoints on the KITTI dataset, and at different periods of time localising between KITTI and KITTI-360. We achieve accurate metric estimates comparable with state-of-the-art methods with almost half the representation size, specifically $1.33\,\mathrm{kB}$ on average.**

***Index Terms*— Localisation, Pose Estimation, Semantic Segmentation, Semantic Mapping, Autonomous Vehicles, Robotics**

I. Introduction

Localisation is crucial for mobile robotics, allowing safe autonomous motion. For this task, LiDAR is popular due to its intrinsically geometric readings and robustness to lighting and appearance, thus providing a very informative and stable representation of its surroundings.

A common approach to localisation is creating a map from a previous traversal of the environment and then registering live sensor readings onto the map to extract the metric ego-vehicle pose. However, modern LiDAR sensors can collect up to tens of thousands of points per scan, making the representation and storage of *compressed but reliable* LiDAR observations compelling. Especially in the Autonomous Driving (AD) domain, maps are prone to being vast and may require a substantial amount of memory to scale [1]. This is also critical in distributed settings, where the map and observations need to be repeatedly transmitted between multiple agents and/or servers [2], [3].

Whilst some methods use pure geometric approaches [1], [4], semantics and objects [5] provide a tradeoff between emphasising local keypoints or global features – while at the same time being human understandable [6]. Each object can be described numerically, and scan matching and registration rely on accurate object correspondences. Motivated by these approaches, this work tries to answer the question: how small can an object descriptor be while still providing enough information for a reliable and accurate scan registration? Here, we show that a descriptor composed of the positions of the objects' centroids and their class – i.e. three floating-points and one byte – is enough for the task, requiring on average $1.33\,\mathrm{kB}$ of storage per LiDAR scan against $1.41\,\mathrm{MB}$ of raw data. To remedy such compact – and thus feature-poor – representation, we enhance state-of-the-art feature embedding and geometric-aware object matching [7], [8] with semantic information, both by encoding it in a network architecture, and by designing a semantically-informed loss to guide the training. We further refine the estimates using RANdom SAmple Consensus (RANSAC) and Iterative Closest Point (ICP) on the object matches, obtaining average translational errors of about $0.1\,\mathrm{m}$ and $0.5°$ respectively on KITTI [9]. Our system is outlined in Fig. 1 and our principal contributions are:

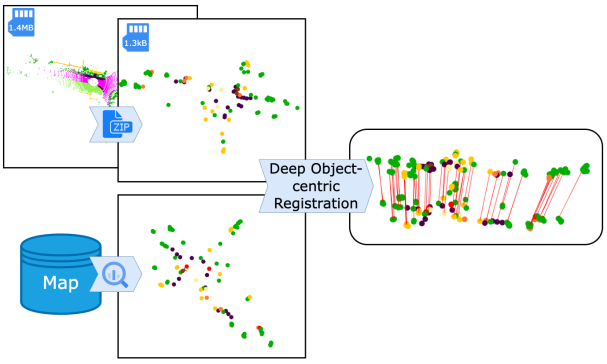

Fig. 1. Method overview. LiDAR scans are represented extremely compactly by only the centroid and semantic class of the corresponding objects in the scene. Sacrificing information in this way, we learn a robust matching function which leverages the remaining geometry in the object scene structure as well as the semantic relationships between entities.

1) An accurate registration approach working on extremely compact object-centric representations of LiDAR scans;
2) A semantic-enhanced neural network architecture and loss function for descriptorless object-matching;
3) Evaluation of the proposed methodology on KITTI [10], [9] and on a long-term localisation scenario between KITTI and KITTI-360 [11]. To the best of our knowledge, this is the first approach that considers long-term cross-dataset registration between these two.

This work was supported by EPSRC Programme Grant "From Sensing to Collaboration" (EP/V000748/1).

II. Related Work

Classical approaches to LiDAR scan registration rely on extracting descriptors from two point clouds, discovering point-to-point correspondences between them and exploiting them to regress the displacement. One of the most widespread and simple approaches is ICP [12], which uses cartesian position to describe and match points. As the environment changes through different viewpoints or over time, dense point-to-point correspondences can be replaced by more robust keypoint-to-keypoint correspondences. As such, keypoint detectors have been developed to discover robust features in the environment and 3D descriptors to distinguish and match them. Examples of classical, hand-crafted keypoint detectors are ISS [13] and KPG [14], which select salient points with large variations in their local neighbourhood. Examples of 3D descriptors include Fast Point Feature Histograms (FPFH) [15] and SHOT [16], which create histogram-like descriptors by taking into consideration the local topology of the neighbouring points.

A. Learning-based Point Cloud Registration Losses

Learning-based approaches have replaced classical ones, typically through end-to-end descriptors supervised by a differentiable Singular Value Decomposition (SVD) operation with ground-truth registration. For instance, Deep Closest Point (DCP) [17] adopts a point-based encoder to extract high-dimensional descriptors and a transformer-based head to compute soft matches to feed into an SVD module. Similar to us – but to subsample the point clouds to use in a DCP setup – DCPCR [18] integrates a compression network, trained to downsample the points clouds while preserving the local information in the feature representation.

A different approach is to directly supervise the matching phase by cross-entropy or contrastive losses. For instance, StickyPillars [19] inspired by the image-based SuperGlue [20], and MDGAT [21] build soft assignments supervised through ground-truth matches obtained by projection of the keypoints from one scan into the other. Empirically, we show that a match-supervision approach tends to perform better in our specific case of few and feature-poor points.

B. Learning-based Point Cloud Registration Architectures

Architectures for point cloud processing for localisation adopt diverse learning architectures. Indeed, whereas some works exploit the vast experience in image-based Convolutional Neural Networks (CNNs) [22], [23], most apply point- and, more recently, transformer- or graph-based approaches.

Although earlier approaches use purely point-based architectures [24], [25], [26], transformer-based architectures are recently enhancing them by applying self- and cross-attention in feature space to share context information from within or across LiDAR scans. One example is GeoTransformer [8], which uses KPConv-FPN [27] and a custom transformer architecture with a keypoint descriptor invariant to rigid transformation. Similarly, DCP [17] and DCPCR [18] exploit DGCNN [7] and KPConv [27] point encoders respectively, and a transformer head for attention-based assignments.

Whereas transformers operate on fully-connected graphs, a purely graph-based method is, for instance, SEM-GAT [6], which exploits semantic and morphological features to find match candidates and compute match attention, leading to introspection capabilities [28]. Similarly, PREDATOR [29] designs an overlap-attention module incorporating a Graph Neural Network (GNN) connecting nearest-neighbour keypoints to extract and refine their feature descriptors.

In our proposed system, we apply a point encoder and transformer approach: we employ a DGCNN module for feature extraction, augmented by encoded semantic features together with the geometry of the scene, and a transformation-invariant matcher adapted from GeoTransformer [8].

C. Use of Semantics in Point Cloud Registration

Non-learned semantic approaches typically extract and describe objects and match them directly. BoxGraph [5], encodes objects with their bounding box achieving a very compact scan representation, while GOSMatch [30] encodes histogram-based descriptors from the distances between the objects in the scene into vertex descriptors for initial pose estimation and verification. For learned registration techniques we can distinguish three types of segmentation usage: augmenting geometric information with semantics, extracting object- or segment-level entities, and its usage in the loss term. Examples of the first approach are DeepSIR [31] and SARNet [32], which learn both geometric and semantic features to constrain feature matching. Differently, SegMatch [33], SegMap [34] and SemSegMap [35] partition point clouds into higher-level segments and extract multidimensional descriptors for each segment, and match them across scenes. More similar to us, InstaLoc [36] and SGPR [37] learn to segment and match individual objects to the prior scene. In particular, [37] applies a neural network architecture to extract vertex features based on graph connectivity. Finally, methods like PADLoc [38] leverage panoptic segmentation of point clouds and use such information during training to aid the convergence to robust descriptors. In our approach, we apply all three rules and extract object-level entities, which we couple with semantic class information as input *and* as an additional term in the loss. The resulting object features are orders of magnitude smaller than the object descriptors in the above-mentioned methods.

III. Method

We address the problem of robot pose estimation for large-scale outdoor LiDAR localisation with minimal, lightweight map and query representations. We formulate this as a registration problem between two point clouds where we seek point correspondences to estimate the relative transformation. By representing the point clouds as small sets of 3D object centroids and their respective semantic types, we obtain structures that yield extremely compact, highly-scalable semantic maps, yet allow very accurate localisation. We employ a learned approach inspired by GeoTransformer [8]. Still, rather than only relying on geometric features for

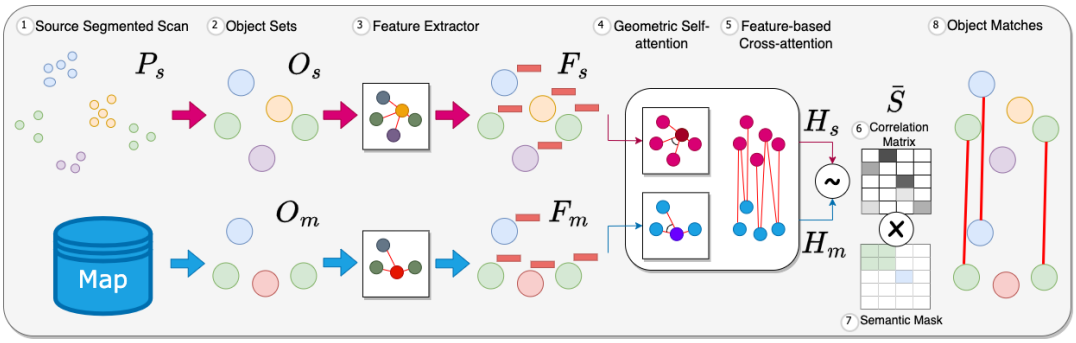

Fig. 2. System diagram. Our method aims to register a semantically labelled query point cloud P_s with a map. It clusters P_s into an object set O_s, keeping only instance centroids and semantic labels. Then, the query and map sets are passed through a semantic embedding and feature extraction module. The resulting object features F_s and F_m are then passed through a geometric self- and feature-based cross-attention matching module, producing a cross-correlation similarity matrix $\bar{S}$. A semantic mask is then applied to filter erroneous matches, resulting in the final object correspondences.

superpoint matching, we integrate semantic information to obtain descriptive features for object matching as in Fig. 2.

A. Problem Setup

Consider a pair of point clouds, $P_s = \{p_i \mid p_i \in \mathbb{R}^3\}$ and $P_m = \{p_j \mid p_j \in \mathbb{R}^3\}$, e.g., from a source live sensor stream of a vehicle and a pre-built target map. We aim to estimate the relative transformation $\mathtt{T}_{s,m} = [\mathtt{R}_{s,m}|\mathbf{t}_{s,m}]$ where $\mathtt{R}_{s,m} \in SO(3)$ and $\mathbf{t}_{s,m} \in \mathbb{R}^3$ denote the rotation and translation. We do so by generating key points, $K_s = \{k_i \mid k_i \in \mathbb{R}^3\}$ from P_s and $K_m = \{k_j \mid k_j \in \mathbb{R}^3\}$ from P_m. Crucially, K_s and K_m are stable and visible across multiple revisits of an environment, and $|K_s| \ll |P_s|$ and $|K_m| \ll |P_m|$. Given a set of ground-truth correspondences between the two sets of key points, $\mathtt{T}_{s,m}$ can be obtained using the Kabsch algorithm [39] using SVD. Hence, we aim to obtain an optimal set of correspondences $\mathcal{C} = \{(k_i, k_j) \mid k_i \in K_s, k_j \in K_m\}$ to produce the final pose estimate. We show that using semantic object-like instances as key points provides a consistent basis for pose estimation even across long-term revisits of the same place under variation in object layout.

B. Object Extraction

We follow BoxGraph [5] and SGPR [37], where input point clouds are segmented via a pre-trained semantic segmentation network and then clustered into object instances based on their semantic labels and Euclidean coordinates. Panoptic segmentation or object detection might yield more accurate instances, yet we opt for this simpler approach since datasets with static object labels are scarce.

In particular, consider a point cloud P (① in Fig. 2) and the set L containing a semantic label for each point in P; here, $L = \{l_i \mid l_i \in \mathbb{L}\}$, where $\mathbb{L} \subset \mathbb{N}$ is a finite set of semantic classes. We apply the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm to extract a set of object-like clusters. For each cluster we keep its centroid o – as the mean coordinate of its points – and semantic label l. We then end up with with a labelled object set $O = \{(\mathbf{o}_i, l_i) \mid \mathbf{o}_i \in \mathbb{R}^3, l_i \in \mathbb{L}\}$, which we use as keypoints (② in Fig. 2). Contrary to other approaches that extract object features by exploiting a point cloud's intra- and inter-object geometric structure, we instead utilise only the semantic label and the inter-object relationships, and *learn* any further representations associated with the object.

C. Feature Enhancement

Indeed, we encode each object's semantic class and neighbouring geometric structure to extract discriminative object descriptors (③ in Fig. 2). For this, we apply a learnable function $f_{sem} = h_{sem} \circ e_{emb}$ to each cluster's semantic label l_i, where $e_{emb} : \mathbb{L} \to \mathbb{R}^{d_{emb}}$ is a categorical embedding function and $h_{sem} : \mathbb{R}^{d_{emb}} \to \mathbb{R}^{d_{sem}}$ is parametrised by a small MLP. We then enhance the features with structural context $\mathbf{o}_i$ through a three-layer DGCNN-based module E [7], producing $F = \{f_i \mid f_i = E(\mathbf{o}_i \| f_{sem}(l_i)), (\mathbf{o}_i, l_i) \in O, f_i n \in \mathbb{R}^{d_f}\}$, where $(\cdot\|\cdot)$ denotes concatenation. At each layer, the feature x_i of the i-th point gets transformed into the feature x_i' through a projection network h_{ec} and max-pooling layer as in:

$$x_i' = \max_{j \in \mathrm{kNN}(i)} h_{ec}(x_i \,\|\, x_j - x_i) \tag{1}$$

where $\mathrm{kNN}(i)$ are i's k neighbours in feature space. Whilst x_i and $(x_j - x_i)$ are not transformation invariant yet, the operation informs the resulting features of the local structure.

D. Object Similarity and Matching

The feature vector of each object must be sufficiently descriptive and discriminative to be effectively matched across scans. Since we discard the internal structure of each object instance, it is crucial to exploit the scene's structure by incorporating it into each object's features. Moreover, as the localisation setting is agnostic of the relative transformation between the scans, so must be the object descriptors.

Since cross-correlation between features across scenes benefits the matching task [17], [20], [8], we employ the Superpoint Matching Module of GeoTransformer [8], which explicitly models the layout within scenes through a geometric self-attention module (④ in Fig. 2), and the latent feature similarity across scenes through a feature-based cross-attention module (⑤ in Fig. 2). The geometric self-attention module encodes the global structure of the scene in a transformation-invariant manner by operating on the relative distances between pairs and the relative angles between triplets of objects. The feature-based cross-attention module, instead, facilitates the feature exchange between the

two scenes. Further details can be found in the original paper [8]. The self- and cross-attention modules are interleaved N_t times, and, from O_s, O_m and the corresponding object features above, output the hybrid features $H_s = \{h_i^s \mid h_i^s \in \mathbb{R}^{d_h}\}$ and $H_m = \{h_j^m \mid h_j^m \in \mathbb{R}^{d_h}\}$, suitable for matching. Following [8], they produce a normalised Gaussian correlation matrix $\bar{S} \in \mathbb{R}^{|O_s| \times |O_m|}$ which serves as matching scores (⑥ in Fig. 2).

At this point, we extend [8] and exploit the semantics of the scene to refine this score further by filtering mismatched objects. We mask out the similarity scores of objects with different labels (⑦ in Fig. 2), i.e. set $\bar{s}_{i,j} \coloneqq 0$, if $l_i^s \neq l_j^m$. The optimal set of object correspondences is then obtained by selecting the top N_c matches with the highest similarity:

$$\hat{\mathcal{C}} = \{(\mathbf{o}_i, \mathbf{o}_j) \mid \mathbf{o}_i \in O_s, \mathbf{o}_j \in O_m, (i,j) \in \mathrm{topk}_{i,j}(\bar{S})\} \tag{2}$$

E. Pose Estimation and Refinement

To find the relative transformation $\mathtt{T}_{s,m} = [\mathtt{R}_{s,m} | \mathbf{t}_{s,m}]$ between object sets O_s and O_m given the above correspondences (⑧ in Fig. 2), we solve

$$\hat{\mathtt{R}}_{s,m}, \hat{\mathbf{t}}_{s,m} = \min_{\mathtt{R},\mathbf{t}} \sum_{(\mathbf{o}_i,\mathbf{o}_j) \in \hat{\mathcal{C}}} \bar{S}_{i,j} \|\mathtt{R}\mathbf{o}_i + \mathbf{t} - \mathbf{o}_j\|_2^2 \tag{3}$$

directly via the weighted SVD algorithm. Alternatively, we can apply the RANSAC algorithm that iteratively selects a subset of matches and tries to maximise the number of inliers up to a distance tolerance for a fixed set of iterations. We explore both approaches in the experimental evaluation; yet, it is important to note that, whereas RANSAC is computationally expensive for dense point clouds, its added complexity is negligible given the low cardinality of the object sets O_s and O_m. Finally, we find it helpful to refine further the relative transformation with ICP, reducing the noise in the coarse estimate above, which again adds insignificant computational overhead.

F. Loss Function

Deep registration approaches are typically supervised in two different ways: a regression loss on the final pose estimate that exploits the differentiability of the soft-assignment matrix between source and target point clouds [17], [18], [38], or a cross-entropy/contrastive loss which uses ground-truth matches to directly supervise the similarity scores [20], [19]. Empirically, when key points are sparse, as in our case, even small ambiguity in the soft assignment matrix introduces large errors in the final estimate; thus, we opt to supervise the object features in a metric fashion similar to GeoTransformer [8], [40], which employs an overlap-aware circle loss on the superpoint features.

The overlap-aware circle loss aims to bring together the features of positive pairs of objects, i.e. spatially proximal, and push apart those of negative pairs, i.e., objects far in space. It does so by weighting the positive matches according to the overlap ratio of their clusters. As we discard object geometries, we cannot recover a measure of overlap. For this reason, we introduce a semantic distance-aware circle loss which focuses the learning on spatially-proximal object pairs proportionally to their Euclidean distance, *and* ensures the positive pairs are of objects of the same semantic class.

Formally, given an anchor object $\mathbf{o}_i \in O_s$, we consider an object $\mathbf{o}_j \in O_m$ a positive if $d_{i,j} \coloneqq \|\mathbf{o}_i - \mathbf{o}_j\|_2^2 < \tau_{match}$ and $l_i == l_j$. We set $\tau_{match} \coloneqq 1\,\mathrm{m}$ and denote the set of anchor objects with $\mathcal{A}_s \subset O_s$, the set of positive corresponding objects with $\mathrm{pos}(i) \subset O_m$, and the rest as negatives with $\mathrm{neg}(i) \subset O_m$. If we let the distance ratio be $\rho_{i,j} \coloneqq 1 - \frac{d_{i,j}}{\tau_{match}} \in [0,1]$ for positive matches, we can formulate the semantic distance-aware circle loss with respect to $\mathcal{A}_s$ as:

$$\mathcal{L}_{DC}^s = \frac{1}{|\mathcal{A}_s|} \sum_{i \in \mathcal{A}_s} \log \left(1 + \sum_{j \in \mathrm{pos}(i)} e^{\beta_{i,j}^p (h_{i,j} - \Delta_p)} \cdot \sum_{j \in \mathrm{neg}(i)} e^{\beta_{i,k}^n (\Delta_n - h_{i,k})} \right) \tag{4}$$

where $h_{i,j} = \left\| h_i^s - h_j^m \right\|_2^2$ is the distance in feature space. The positive pairs and negative pairs are weighted according to weight factors $\beta_{i,j}^p = \sqrt{\rho_{i,j}}\gamma(h_{i,j} - \Delta_p)$ and $\beta_{i,k}^n = \gamma(\Delta_n - h_{i,k})$, where $\gamma = 40$ is a scaling factor and $\Delta_p = 0.1$ and $\Delta_n = 1.4$ are the corresponding margins. We define the loss $\mathcal{L}_{DC}^m$ for the target objects $\mathcal{A}_m$ analogously, yielding the overall loss $\mathcal{L}_{DC}$ as the average of $\mathcal{L}_{DC}^s$ and $\mathcal{L}_{DC}^m$.

IV. Experimental Setup

A. Datasets

We evaluate our method on KITTI [9] and KITTI-360 [11], which include traversal sequences of different areas in Karlsruhe, Germany, by a vehicle equipped with a Velodyne HDL-64 LiDAR sensor. We first explore short-term revisits – applicable to loop closing and SLAM scenarios – on KITTI sequence `08`, challenging due to reverse revisits. The second considers long-term revisits in a teach&repeat setup where we select KITTI sequence `07` as a map and localise KITTI-360 sequence `09`, which has been recorded two years apart, with potentially significant changes in the scenes. In all our evaluations, we input to our model semantic labels as predicted by a top-performing segmentation network, Cylinder3D [41], pre-trained on SemanticKITTI [9].

We consider the KITTI sequences that contain revisits and use `00`, `05`, `06`, and `09` for training and `02` for validation, by selecting pairs of scans within $3\,\mathrm{m}$ that are at least 50 frames apart, as in [37], [42], [5]. While $3\,\mathrm{m}$ may be a limited threshold for general registration, we argue that a coarse location estimate could be achieved with raw GPS readings or place recognition approaches such as SGPR [37], which use a similar compact object representation as ours.

The raw ground-truth poses are noisy, so we further refine them via ICP. To register KITTI and KITTI-360, we use the raw GPS measurements to select pairs within $3\,\mathrm{m}$. Then we remove points belonging to dynamic objects and the road class using Cylinder3D predictions, and again refine with ICP to obtain accurate ground-truth poses, as in Fig. 3.

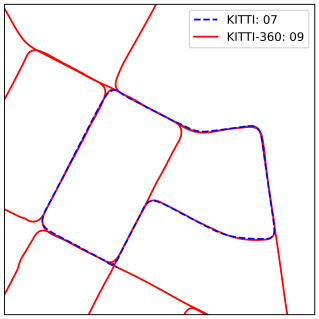

Fig. 3. KITTI-360 sequence 09, registered on KITTI sequence 07.

B. Baselines

We compare our method with geometric, learning-based and semantic approaches. 1) RANSAC-based, hand-crafted FPFH features as a dense geometric approach – with downsampled input point clouds with a voxel size of 30 cm; 2) BoxGraph [5], a compact hand-crafted semantic method; 3) PADLoC [38][1], a learned semantic global localisation approach; 4) superpoint-matching module of the vanilla GeoTransformer [8] (here, GeoTF-SP). We re-train the original architecture on our training data and, at test time, replace the fine registration module on the dense point patches with RANSAC-based matching on superpoint correspondences. We also refine all methods with ICP for a fair comparison.

C. Metrics

To measure the performance, we report the Relative Translation Error (RTE) and Relative Rotation Error (RRE):

$$\mathrm{RTE} = \left\|\hat{\mathbf{t}} - \mathbf{t}\right\|_2, \mathrm{RRE} = \cos^{-1}\left(\frac{\mathrm{Tr}\left(\hat{\mathtt{R}}^T\mathtt{R}\right) - 1}{2}\right) \quad (5)$$

between the estimated $[\hat{\mathtt{R}}_{s,m}|\hat{\mathbf{t}}_{s,m}]$ and ground-truth $[\mathtt{R}_{s,m}|\mathbf{t}_{s,m}]$ rotation and translation, averaged over successful registrations. Subscript is removed for brevity. We also report the registration recall that measures the percentage of successful registrations such that $\mathrm{RTE} < \tau_t$ and $\mathrm{RRE} < \tau_R$.

D. Implementation Details

As in [5], we only keep static objects (sidewalk, building, fence, vegetation, trunk, pole, and traffic sign), stable across revisits. Our semantic embedding dimension is 4, followed by an MLP(32, 64, 128). We use 3 EdgeConv(64, 64) layers in DGCNN with 30 nearest neighbours, followed by an MLP(1024, 512, 256, 256) with ReLU. Normalisation and dropout deteriorate performance, so we omit them. GeoTransformer's matching module is set as in [8], with 3 sets of self- and cross-attention layers with 4 attention heads, and output dimension of 256. We train on an NVIDIA RTX 3090 Ti GPU with a batch size of 32 for 50 epochs, a learning rate of $1 \cdot 10^{-3}$ and ADAM optimizer, halving the learning rate whenever the loss has plateaued for 5 epochs. We apply random yaw rotations up to 360°, randomly subsample raw point clouds to 24000 points before clustering and add random jitter to object centroids to simulate sensor noise. At inference time, we set the number of object correspondences N_c to 15 when using SVD, and 60 with RANSAC. This allows SVD to focus on high-quality matches, while RANSAC has higher outlier tolerance by design. Segmentation and clustering take 62 ms and 80 ms, respectively, while registering a pair of scans takes 20 ms including refinement, showing the efficiency of the approach.

V. Results

A. Map Compactness

Here we compare our storage requirements to other methods. Our representations are extremely compact, producing on average 105 and maximum 238 objects-like instances per point cloud, similar to BoxGraph [5]. For each instance, we store the three floating point coordinates of the centroid and a byte for the semantic class, which on average requires 1.33 kB of storage. With our representation, the 3.2 km-long KITTI sequence 08 can be stored with 4.47 MB only.

BoxGraph stores three additional floating point numbers – the objects' bounding boxes – almost doubling the memory requirement. The difference is even larger compared to dense methods like GeoTranformer [8], which require the full LiDAR scan: given 120k as the number of sampled points, common in AD applications, these methods would require three floats for point, i.e. 1.4 MB. Even when voxelised representations are used to reduce the point count – reaching usually 20k points – storage requirements are about 234.4 kB. Compressed methods require an intermediate storage space: for instance, OctSqueeze [1] ranges between 2 and 15 bits per point (bpp), thus requiring from 29.3 kB to 219.7 kB. DCPCR, instead, can compress a point cloud 100-folds[2], compared to our average ratio of about 1:1000.

B. Pose Estimation

We first explore in Tab. I the pose estimation performance of our approach in the short-term setting, where revisits happen within the same sequence. Both our method and BoxGraph take semantic labels as inputs explicitly, so entries with (GT) represent results with ground-truth *semantic* labels from SemanticKITTI, as opposed to Cylinder3D predictions.

Whilst classical RANSAC-based approaches struggle to deduce accurate pose, we achieve accurate results with RANSAC and SVD. In particular, our method using Cylinder3D semantic labels estimates the pose within 0.3 m and 1° more than 85% of the time. BoxGraph struggles with reverse revisits when using Cylinder3D labels, even after ICP-refinement. Its performance improves significantly when using ground-truth semantics, so we conclude it strongly depends on the quality of input segmentation. In contrast, our method shows greater stability when using ground-truth semantic labels as input. In addition, using the vanilla GeoTF-SP (with RANSAC) proves ineffective in this setting, even though its deeper features are directly supervised for matching. This justifies our use of object centroids as distinctive and repeatable keypoints.

[1]PADLoC's public weights are trained on a slightly larger superset of our training data, containing pairs within 4 m, and including sequence 07.

[2]As DCPCR uses aggregated point clouds; here we use the ratio they report for comparison.

We observe similar trends in the second setting, where we investigate the long-term localisation performance. We see that PADLoC, which incorporates semantics implicitly, struggles to register reliably without ICP refinement. This might mean it is difficult to generalize the semantic features across long periods. Our method models objects explicitly and demonstrates high recall of pose estimates with errors within $0.5\,\mathrm{m}$ and $5°$. Note that neither Cylinder3D, nor our method are trained on KITTI-360, showing further robustness across large temporal intervals. This is in line with the other explicit object-based method, BoxGraph, whose performance improves. Comparing the different variants of our method, we see that weighted SVD achieves lower metric errors since it operates on high-quality matches, while RANSAC indeed maximises the consensus between matches, yielding high recall.

		$0.3\,\mathrm{m}/1°$			$0.5\,\mathrm{m}/5°$		
	Method	**RR**	**RTE**	**RRE**	**RR**	**RTE**	**RRE**
Seq: 08	FPFH	3.48	0.18	0.63	15.42	0.28	1.59
	GeoTF-SP	20.29	0.19	0.52	42.26	0.27	0.79
	GeoTF-SP + ICP	20.49	0.19	0.51	40.83	0.26	0.79
	PADLoC	14.24	0.20	0.66	67.78	0.30	1.21
	PADLoC + ICP	49.28	**0.11**	**0.28**	80.02	0.20	0.56
	BoxGraph	18.75	0.16	0.60	44.67	0.22	1.16
	BoxGraph + ICP	53.28	0.12	0.34	59.38	0.13	0.45
	Ours - RANSAC	**87.50**	0.13	0.45	**99.80**	0.14	0.54
	Ours - SVD	85.66	0.12	0.44	95.18	**0.12**	**0.52**
	BoxGraph (GT)	51.23	0.12	0.50	77.61	0.16	0.85
	BoxGraph + ICP (GT)	80.64	0.11	0.40	85.09	0.11	0.44
	Ours - RANSAC (GT)	88.52	0.11	0.45	**99.49**	0.13	0.52
	Ours - SVD (GT)	**89.60**	**0.09**	**0.41**	96.52	**0.10**	**0.47**
Seq: 09, Map: 07	FPFH	42.47	0.16	0.48	72.34	0.22	0.93
	GeoTF-SP	38.22	0.17	0.38	56.01	0.23	0.51
	GeoTF-SP + ICP	39.32	0.13	0.31	54.00	0.20	0.45
	PADLoC	0.01	0.23	0.84	5.22	0.38	3.31
	PADLoC + ICP	46.93	0.13	**0.30**	87.00	0.21	0.73
	BoxGraph	41.92	0.14	0.44	60.20	0.19	0.74
	BoxGraph + ICP	62.52	0.12	0.32	71.29	0.15	0.41
	Ours - RANSAC	78.14	**0.11**	0.35	**94.14**	0.15	0.47
	Ours - SVD	**80.52**	**0.11**	**0.30**	92.66	**0.14**	**0.39**

TABLE I. Registration errors $[\mathrm{m}|°]$ at different thresholds for short-term revisits on KITTI sequence 08 and long-term localisation on KITTI-360 sequence 09 with KITTI sequence 07 as map.

C. Ablation Studies and Analysis

We analyse the different components of our method to gain further insight into our design choices. We report results on KITTI sequence 08 and use weighted SVD on $N_c = 60$ matches. SVD is more strongly affected by erroneous matches, so this better models the match quality.

Tab. II reports the results, where $\mathcal{L}_{DC}$ denotes the proposed distance-aware circle loss and $\mathcal{L}_{DC}^{-}$ ignores semantic classes and allows mislabelled matches, while $\mathcal{L}_{SVD} = ||\mathbf{t}_{gt} - \hat{\mathbf{t}}|| + ||\mathbf{I} - \mathbf{R}_{gt}^{\mathbf{t}}\hat{\mathbf{R}}||$ denotes the explicit pose loss estimated with SVD on the full similarity matrix $\bar{S}$ (⑥ in Fig. 2).

We can see in (1)-(5) that performance deteriorates without the input semantic embeddings and the semantic filtering in our loss. We notice that having no input semantics and filtering in the loss, (1)-(2), performs better than omitting either one of them (3)-(5), potentially meaning that having semantics at only one end confuses the network while ignoring them altogether allows it to learn matches purely based on spatial proximity. The largest drop in performance happens when replacing the $\mathcal{L}_{DC}$-loss (4), which justifies our design choice. (6) also shows that the DGCNN feature enhancement is crucial for learning discriminative object features. In (7) we replace the GeoTransformer Superpoint Matching Module with a KPConv-based feature encoder and a feature-based transformer head as in [18]. The deteriorated performance shows the descriptive power of the geometric self-attention of GeoTransformer, which supports our design.

			Architecture		Metric Errors $[0.3\,\mathrm{m}/1°]$		
	Sem. Emb.	**DGCNN**	**Matching Module**	**Loss**	**RR**	**RTE**	**RRE**
(1)	✗	✓	GeoTF	$\mathcal{L}_{DC}^{-}$	36.17	0.14	0.40
(2)	✗	✓	GeoTF + ICP	$\mathcal{L}_{DC}^{-}$	68.75	**0.11**	**0.38**
(3)	✗	✓	GeoTF	$\mathcal{L}_{DC}$	26.59	0.15	0.46
(4)	✓	✓	GeoTF	$\mathcal{L}_{SVD}$	0.31	0.21	0.70
(5)	✓	✓	GeoTF	$\mathcal{L}_{DC}^{-}$	31.25	0.14	0.40
(6)	✓	✗	GeoTF	$\mathcal{L}_{DC}$	17.01	0.17	0.56
(7)	✓	✓	KPConv + TF	$\mathcal{L}_{SVD}$	16.09	0.13	0.57
(8)	✓	✓	GeoTF	$\mathcal{L}_{DC}$	41.09	0.14	0.42
(9)	✓	✓	GeoTF + ICP	$\mathcal{L}_{DC}$	**75.67**	**0.11**	0.41

TABLE II. Ablation on architecture on KITTI sequence 08, $[\mathrm{m}|°]$.

D. Reducing Registration Overlap

So far we consider pairs of point clouds with sufficient overlap. We now briefly study the performance of our method with lesser overlap. We follow a common setup [8], [23], [29] which aims to register point cloud pairs that are at least $10\,\mathrm{m}$ apart. We train on KITTI sequences 00-05 and report the averaged results on sequences 08-10 in Tab. III. We see that while our performance is limited due to the low quality of object instances we extract, our RANSAC-based approach still obtains more than 90% registration recall within the commonly used boundaries of $\tau_t = 2\,\mathrm{m}$ and $\tau_R = 5°$. This is a significant improvement compared to BoxGraph which uses the same instances as our approach, and showcases the potential of learned centroid-only object-based registration.

	$2\,\mathrm{m}/5°$		
Method	**RR**	**RTE**	**RRE**
BoxGraph	10.85	0.61	1.95
Ours - RANSAC	**95.53**	**0.27**	**1.05**
Ours - SVD	67.97	0.33	1.07
BoxGraph (GT)	16.49	0.52	1.84
Ours - RANSAC (GT)	**92.65**	**0.26**	**1.06**
Ours - SVD (GT)	65.90	0.27	0.96

TABLE III. Registration errors $[\mathrm{m}|°]$ for scans at least $10\,\mathrm{m}$ apart, averaged over KITTI sequences 08-10.

VI. Conclusion

We presented a novel approach for global registration of LiDAR point clouds with extremely compact object representations. We show that using only the 3D centroids and semantic type of object instances is enough to estimate accurate poses in challenging conditions, including reverse and long-term revisits of the same place. We employ a low-level geometric-attention-based matching module, and enhance it with high-level semantics to increase its discriminative power. The resulting maps have very low storage requirements, which enables drastic scaling of the mapped areas. In addition, working directly on objects can be a subject for cross-modal localisation or semantic analysis and reasoning, which is the focus of future work.

REFERENCES

[1] L. Huang, S. Wang, K. Wong, J. Liu, and R. Urtasun, "Octsqueeze: Octree-structured entropy model for lidar compression," in *IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[2] T. Cieslewski, S. Choudhary, and D. Scaramuzza, "Data-efficient decentralized visual slam," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[3] B. Ramtoula, R. de Azambuja, and G. Beltrame, "Capricorn: Communication aware place recognition using interpretable constellations of objects in robot networks," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[4] C. Cao, M. Preda, and T. Zaharia, "3d point cloud compression: A survey," in *International Conference on 3D Web Technology*, 2019.

[5] G. Pramatarov, D. De Martini, M. Gadd, and P. Newman, "Boxgraph: Semantic place recognition and pose estimation from 3d lidar," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 7004–7011.

[6] E. Panagiotaki, D. De Martini, G. Pramatarov, M. Gadd, and L. Kunze, "Sem-gat: Explainable semantic pose estimation using learned graph attention," in *International Conference on Advanced Robotics (ICAR)*, 2023, pp. 367–374.

[7] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, oct 2019. [Online]. Available: https://doi.org/10.1145/3326362

[8] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, S. Ilic, D. Hu, and K. Xu, "Geotransformer: Fast and robust point cloud registration with geometric transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[9] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *International Conference on Computer Vision*, 2019.

[10] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition*, 2012.

[11] Y. Liao, J. Xie, and A. Geiger, "Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[12] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.

[13] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3d object recognition," in *2009 IEEE 12th international conference on computer vision workshops, ICCV Workshops*, 2009, pp. 689–696.

[14] A. Mian, M. Bennamoun, and R. Owens, "On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes," *International Journal of Computer Vision*, 2010.

[15] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *International Conference on Robotics and Automation*, 2009.

[16] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *European Conference on Computer Vision*, 2010.

[17] Y. Wang and J. M. Solomon, "Deep closest point: Learning representations for point cloud registration," in *IEEE/CVF international conference on computer vision*, 2019, pp. 3523–3532.

[18] L. Wiesmann, T. Guadagnino, I. Vizzo, G. Grisetti, J. Behley, and C. Stachniss, "Dcpcr: Deep compressed point cloud registration in large-scale outdoor environments," *IEEE Robotics and Automation Letters*, 2022.

[19] K. Fischer, M. Simon, F. Olsner, S. Milz, H.-M. Gross, and P. Mader, "Stickypillars: Robust and efficient feature matching on point clouds using graph neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[20] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.

[21] C. Shi, X. Chen, K. Huang, J. Xiao, H. Lu, and C. Stachniss, "Keypoint matching for point cloud registration using multiplex dynamic graph attention networks," *IEEE Robotics and Automation Letters*, 2021.

[22] X. Chen, T. Läbe, A. Milioto, T. Röhling, J. Behley, and C. Stachniss, "Overlapnet: A siamese network for computing lidar scan similarity with applications to loop closing and localization," *Autonomous Robots*.

[23] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *International Conference on Computer Vision*, 2019.

[24] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "Pointnetlk: Robust & efficient point cloud registration using pointnet," in *IEEE/CVF conference on computer vision and pattern recognition*, 2019.

[25] G. D. Pais, S. Ramalingam, V. M. Govindu, J. C. Nascimento, R. Chellappa, and P. Miraldo, "3dregnet: A deep neural network for 3d point registration," in *IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[26] W. Yuan, B. Eckart, K. Kim, V. Jampani, D. Fox, and J. Kautz, "Deepgmr: Learning latent gaussian mixture models for registration," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, 2020.

[27] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *IEEE/CVF international conference on computer vision*, 2019.

[28] E. Panagiotaki, D. De Martini, and L. Kunze, "Semantic interpretation and validation of graph attention-based explanations for gnn models," in *International Conference on Advanced Robotics (ICAR)*, 2023.

[29] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, and K. Schindler, "Predator: Registration of 3d point clouds with low overlap," in *IEEE/CVF Conference on computer vision and pattern recognition*, 2021, pp. 4267–4276.

[30] Y. Zhu, Y. Ma, L. Chen, C. Liu, M. Ye, and L. Li, "Gosmatch: Graph-of-semantics matching for detecting loop closures in 3d lidar data," in *International Conference on Intelligent Robots and Systems*, 2020.

[31] Q. Li, C. Wang, C. Wen, and X. Li, "Deepsir: Deep semantic iterative registration for lidar point clouds," *Pattern Recognition*, 2023.

[32] C. Liu, J. Guo, D.-M. Yan, Z. Liang, X. Zhang, and Z. Cheng, "Sarnet: Semantic augmented registration of large-scale urban point clouds," *arXiv preprint arXiv:2206.13117*, 2022.

[33] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "Segmatch: Segment based place recognition in 3d point clouds," in *International Conference on Robotics and Automation*, 2017.

[34] R. Dubé, A. Cramariuc, D. D. H. Sommer, M. Dymczyk, J. N. R. Siegwart, and C. Cadena, "SegMap: Segment-based mapping and localization usingdata-driven descriptors," *The International Journal of Robotics Research*.

[35] A. Cramariuc, F. Tschopp, N. Alatur, S. Benz, T. Falck, M. Brühlmeier, B. Hahn, J. Nieto, and R. Siegwart, "Semsegmap–3d segment-based semantic localization," in *International Conference on Intelligent Robots and Systems*, 2021, pp. 1183–1190.

[36] L. Zhang, T. Digumarti, G. Tinchev, and M. Fallon, "Instaloc: One-shot global lidar localisation in indoor environments through instance learning," *Robotics: Science and Systems (RSS)*, 2023.

[37] 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 *International Conference on Intelligent Robots and Systems*, 2020.

[38] J. Arce, N. Vödisch, D. Cattaneo, W. Burgard, and A. Valada, "Padloc: Lidar-based deep loop closure detection and registration using panoptic attention," *IEEE Robotics and Automation Letters*, 2023.

[39] W. Kabsch, "A solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A*, 1976.

[40] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei, "Circle loss: A unified perspective of pair similarity optimization," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[41] H. Zhou, X. Zhu, X. Song, Y. Ma, Z. Wang, H. Li, and D. Lin, "Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation," *arXiv preprint arXiv:2008.01550*, 2020.

[42] L. Li, X. Kong, X. Zhao, T. Huang, W. Li, F. Wen, H. Zhang, and Y. Liu, "Ssc: Semantic scan context for large-scale place recognition," in *International Conference on Intelligent Robots and Systems*, 2021.