HRegNet: A Hierarchical Network for Large-scale Outdoor LiDAR Point Cloud Registration

Fan Lu[1], Guang Chen[1,*], Yinlong Liu[2], Lijun Zhang[1], Sanqing Qu[1], Shu Liu[3], Rongqi Gu[4]
[1]Tongji University, [2]Technische Universität München, [3]ETH Zurich, [4]Westwell lab
{lufan,guangchen,tjedu_zhanglijun,2011444}@tongji.edu.cn
Yinlong.Liu@tum.de, liush@ethz.ch, rongqi.gu@westwell-lab.com

Abstract

Point cloud registration is a fundamental problem in 3D computer vision. Outdoor LiDAR point clouds are typically large-scale and complexly distributed, which makes the registration challenging. In this paper, we propose an efficient hierarchical network named HRegNet for large-scale outdoor LiDAR point cloud registration. Instead of using all points in the point clouds, HRegNet performs registration on hierarchically extracted keypoints and descriptors. The overall framework combines the reliable features in deeper layer and the precise position information in shallower layers to achieve robust and precise registration. We present a correspondence network to generate correct and accurate keypoints correspondences. Moreover, bilateral consensus and neighborhood consensus are introduced for keypoints matching and novel similarity features are designed to incorporate them into the correspondence network, which significantly improves the registration performance. Besides, the whole network is also highly efficient since only a small number of keypoints are used for registration. Extensive experiments are conducted on two large-scale outdoor LiDAR point cloud datasets to demonstrate the high accuracy and efficiency of the proposed HRegNet. The project website is `https://ispc-group.github.io/hregnet`.

1. Introduction

Point cloud registration aims to estimate the optimal rigid transformation between two point clouds, which is a fundamental problem in 3D computer vision and plays an important role in many applications such as robotics [27] and autonomous driving [24].

Iterative Closest Point (ICP) [3] is the best-known method to solve point cloud registration problem. However, ICP highly relies on the initial guesses of the transformation for iteration and can easily get stuck into local minimum due to the non-convexity of the problem. Several variants of ICP [31, 23, 37] have been proposed to achieve global optimal estimation, however, are typically time-consuming for large-scale point clouds.

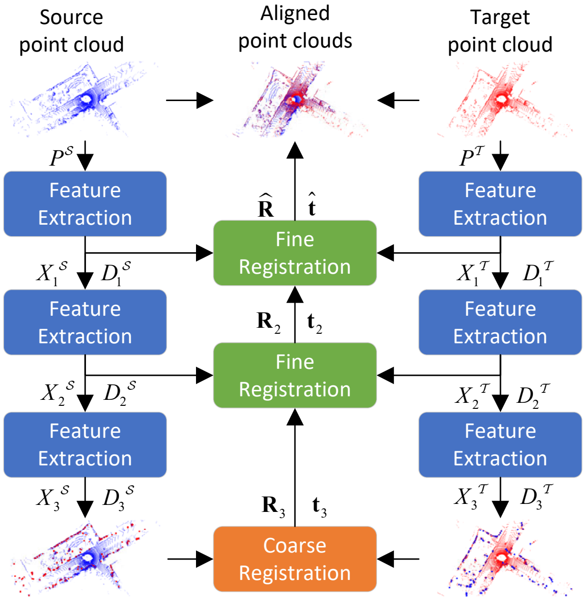


Figure 1. Network architecture of the proposed HRegNet. The point clouds P are hierarchically downsampled to small sets of keypoints X and descriptors D. We perform coarse registration in the bottom layer to leverage the reliable features for keypoints matching and fine registration is followed to refine the transformation by exploiting the precise position information in upper layers.

Recently, deep learning has achieved great success in numerous 3D computer vision tasks such as 3D object detection and semantic segmentation [13]. There also emerge a number of deep learning-based methods for point cloud registration. However, existing methods are mostly designed for object-level point clouds [35, 1, 36, 39, 19] or indoor point clouds [5, 14, 25, 11]. Compared to object-level or indoor point clouds, outdoor LiDAR point clouds typically have higher sparsity, larger spatial range and a more complex and variable distribution, which makes the registration

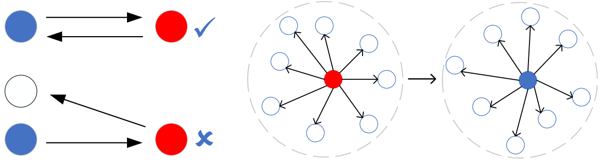
[*] Corresponding author: guangchen@tongji.edu.cn

intractable. Consequently, existing learning-based methods are either unreliable or time-consuming to be applied to outdoor LiDAR point cloud registration.

In this paper, we aim to provide an accurate, reliable and efficient network for large-scale outdoor LiDAR point cloud registration. Inspired by the success of learning-based 3D features on LiDAR point cloud registration [18, 2, 20, 38, 7], we propose a hierarchical keypoint-based point cloud registration network named HRegNet. The overall structure is displayed in Fig. 1. We hierarchically downsample the point clouds to multiple small sets of keypoints and descriptors for registration. Intuitively, as the layer goes deeper, the information of a single keypoint increases, which makes the descriptors more reliable for keypoints matching, however, the increasing sparsity of keypoints may also cause larger position error of corresponding keypoints. Based on the above consideration, the network starts with coarse registration in the bottom layer by globally matching keypoints in descriptor space to leverage the reliable features. Then the coarse transformation is refined by fine registration in upper layers based on local matching in spatial neighborhoods, which exploits the precise position information in shallower layers. Besides, since only a small number of keypoints are used for registration, the network has high efficiency and can be applied in applications requiring real-time performance, such as autonomous driving.

Although the keypoints in the bottom layer have reliable features, possible error of descriptors may lead to a considerable number of mismatches. To improve the robustness and accuracy of registration, we present a learning-based correspondence network to generate corresponding keypoints and reject unreliable correspondences. Here we introduce two important concepts for keypoints matching, namely bilateral consensus and neighborhood consensus. Bilateral consensus, as illustrated in Fig. 2(a), means that a pair of corresponding keypoints should be the nearest neighbor in descriptor space of each other from both sides. As shown in Fig. 2(b), neighborhood consensus indicates that the neighboring keypoints of two corresponding keypoints should also have high similarity. Notably, bilateral consensus and neighborhood consensus have been successfully applied in many cases (*e.g.*, estimate image dense correspondences [30]). To effectively incorporate them into the learning-based registration pipeline, we design novel similarity features based on bi-directional similarity of descriptors and an attention-based neighbor encoding module, which significantly improves the registration performance.

To evaluate the proposed HRegNet, extensive experiments are performed on two large-scale outdoor LiDAR point cloud datasets, namely KITTI odometry dataset [10] and NuScenes dataset [4]. The results demonstrate that the proposed method significantly outperforms existing methods in terms of both accuracy and efficiency.



(a) Bilateral consensus.

(b) Neighborhood consensus.

Figure 2. (a) Bilateral consensus: Two corresponding keypoints should be the nearest neighbor in descriptor space of each other. (b) Neighborhood consensus: Spatial neighborhoods of two corresponding keypoints should also be similar.

In summarize, our main contributions are as follows:

- We propose a novel point cloud registration network named HRegNet, which achieves state-of-the-art performance with high computational efficiency.
- The hierarchical paradigm well combines the strengths of keypoints and descriptors in shallower and deeper layers to achieve precise and robust registration.
- We design novel similarity features, which effectively incorporate bilateral consensus and neighborhood consensus into the registration pipeline and significantly improve the registration performance.

2. Related works

We briefly review the related works in two aspects: classical and learning-based point cloud registration methods.

Classical point cloud registration: Iterative closest point (ICP) [3] is the best-known algorithm for point cloud registration, which iteratively finds the closest point and updates the transformation by solving a least square problem. However, ICP is a local registration algorithm and can easily get stuck into local minimum. Several variants [31, 23, 37] aim to break the limitation of ICP. Go-ICP [37] uses a Branch-and-Bound (BnB) algorithm to search a global optimal solution. Several methods also try to extract features from point clouds for registration [9, 32, 33, 34, 15]. For example, Fast Point Feature Histogram (FPFH) [32] builds an oriented histogram using pairwise geometric properties. A comprehensive review of handcrafted features in 3D point clouds can be found in [12]. After feature extraction, *RANdom SAmple Consensus* (RANSAC) [8] is commonly used for robust feature matching by randomly sampling small subsets of correspondences and then finding optimal correspondences for registration.

Learning-based point cloud registration: PointNetLK is a pioneering work of learning-based point cloud registration [1]. It performs registration by combines PointNet [28] and Lucas & Kanade algorithm [22] into a single trainable recurrent deep neural network. Deep Closest Point (DCP) [35] is a well-known learning-based point cloud registration network. It uses a transformer network to predict soft matching between point clouds and provides a differentiable Singular Value Decomposition (SVD) layer to calculate transformation. IDAM [19] utilizes an iterative distance-aware similarity matrix convolution module for pairwise points matching. However, the above methods are basically designed for object-level point clouds and not applicable to complex large-scale LiDAR point clouds.

Recently, there emerge several learning-based methods for indoor point cloud registration. 3DRegNet [25] proposes to use deep network to directly regress the transformation. Feature-metric registration [14] aims to solve the registration problem from a different perspective. It performs registration by minimizing a feature-metric projection error without correspondences rather than minimizing commonly used geometric error. Gojcic *et al.* mainly focus on the registration of multiview 3D point clouds [11]. Deep Global Registration (DGR) [5] proposes to use a learning-based feature named Fully Convolutional Geometric Features (FCGF) [7] to perform registration. A 6D convolutional network [6] is adopted to predict a likelihood for each correspondence. DGR achieves state-of-the-art performance in indoor point cloud registration. DeepVCP [21] is a method designed for LiDAR point cloud registration. It proposes to use virtual points to construct correspondences. However, the keypoints matching in DeepVCP is performed only in local 3D coordinate space, which makes the method can be basically applied to local registration problem.

3. Methodology

Given source and target point clouds $P^{\mathcal{S}}, P^{\mathcal{T}} \in \mathbb{R}^{N\times 3}$, HRegNet aims to predict the optimal rotation matrix $\hat{\mathbf{R}}$ and translation vector $\hat{\mathbf{t}}$ from source to target point clouds in a coarse-to-fine manner. As shown in Fig. 1, here we adopt a 3-layer implementation. Given a point cloud P, we utilize 3 cascaded feature extraction modules to hierarchically downsample the point clouds to multiple small sets of keypoints $X_l \in \mathbb{R}^{M_l\times 3}$, descriptors $D_l \in \mathbb{R}^{M_l\times C_l}$ and also saliency uncertainties $\Sigma_l \in \mathbb{R}^{M_l}$, where $l = \{1, 2, 3\}$ represents the layer number, M_l is the number of keypoints and C_l is the channel of descriptors. To exploit reliable features of keypoints in the bottom layer, coarse registration is performed by globally matching keypoints in descriptor space to estimate a coarse transformation $\mathbf{R}_3, \mathbf{t}_3$, which is further applied to transform the keypoints in upper layer. After that, we adopt fine registration in layer $l = 2$ to refine the coarse transformation. We assume that the coarse transformation can basically align the point clouds, thus, keypoints matching in fine registration is performed locally in spatial neighborhoods. Finally, another fine registration is applied in the top layer to obtain the final estimation $\hat{\mathbf{R}}, \hat{\mathbf{t}}$.

3.1. Feature extraction

The input of each feature extraction module is the keypoints (or the original point cloud), saliency uncertainties, descriptors and also features of keypoints in previous layer. We firstly adopt Weighted Farthest Point Sampling (WFPS) [42, 29] to select a set of candidate keypoints. After that, k-nearest-neighbor (kNN) search is performed to construct clusters centered on the candidate keypoints and a Shared Multi-layer Perceptron (Shared-MLP) [28] is followed to refine the location of candidate keypoints by predicting attentive weight for each neighboring point in the cluster. Saliency uncertainty is also predicted by applying another Shared-MLP to the cluster. Besides, a descriptor network is designed to extract descriptor from the cluster for each keypoint. Since the feature extraction module is not the main focus of this paper, the detailed network structure is provided in our supplementary material.

3.2. Coarse registration

After the keypoints and descriptors are extracted by the feature extraction module, the key problem then is how to find correct correspondences between source and target keypoints. The most commonly used method to match two sets of keypoints is nearest neighbor (NN) search in descriptor space. Although the descriptors in the bottom layer are relatively more reliable, they are not perfect. Thus, the simple NN search-based approach may result in a considerable number of mismatches due to possible errors of descriptors, which will cause a large registration error. To address the above problem, in this paper, we adopt a learning-based correspondence network to match two sets of keypoints in the bottom layer $l = 3$ to perform coarse registration.

3.2.1 Correspondence network

To simplify the formulation, the subscripts l indicating layer number are omitted in this section and we denote the source and target keypoints and descriptors as $X^{\mathcal{S}}, X^{\mathcal{T}} \in \mathbb{R}^{M\times 3}$ and $D^{\mathcal{S}}, D^{\mathcal{T}} \in \mathbb{R}^{M\times C}$, respectively. As shown in Fig. 3, for a source keypoint $x^{\mathcal{S}}$, we firstly perform k-nearest-neighbor (kNN) search in descriptor space to find K candidate corresponding keypoints in $X^{\mathcal{T}}$. The K neighboring candidate keypoints $\{x_1^{\mathcal{T}}, \cdots, x_K^{\mathcal{T}}\}$ and the center keypoint $x^{\mathcal{S}}$ form a cluster. The features of the cluster consist of three parts: geometric features F_G, descriptor features F_D and similarity features F_S. F_G is the concatenation of coordinates of the center and neighboring keypoints. Besides, the relative coordinates and distances between neighboring and center keypoints are calculated as additional geometric features. F_D consists of the descriptors of center and neighboring keypoints. In addition, the saliency uncertainties of keypoints are also included in F_D. F_S is introduced to incorporate bilateral consensus and neighborhood consensus

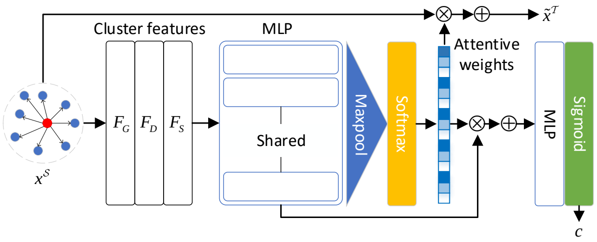


Figure 3. Architecture of correspondence network in coarse registration. The input is the kNN cluster of a source keypoint $x^{\mathcal{S}}$ and the features consist of geometric features F_G, descriptor features F_D and similarity features F_S. The output is the corresponding keypoint $\tilde{x}^{\mathcal{T}}$ and the confidence score c.

and will be described in detail in Section 3.2.2 below.

The cluster features are firstly passed into a 3-layer Shared-MLP to generate a feature map $\tilde{F} = \{f_1, \cdots, f_K\}$. A max-pool layer and a Softmax function are further applied to predict an attentive weight $w_k^{\mathcal{T}}$ for each candidate keypoint $x_k^{\mathcal{T}}$. The estimated corresponding keypoint of $x^{\mathcal{S}}$ can be represented as the weighted sum of the candidate keypoints. Besides, an attentive feature $\bar{F}$ of the cluster is calculated as the weighted sum of $\tilde{F}$. $\bar{F}$ is further fed into a MLP with a Sigmoid function to predict a confidence score c for this correspondence. Then the confidence score is normalized by $\tilde{c}_i = c_i / \sum_{j=1}^{M} c_j$. As we claimed before, using simple NN search can cause a considerable number of mismatches due to the error of descriptors. Intuitively, the proposed attention-based formulation aims to implicitly assign higher weights to the correct candidate corresponding keypoints. The learning-based paradigm incorporates the geometric features, descriptors and also bilateral consensus and neighborhood consensus to generate accurate correspondences and reject unreliable correspondences using the predicted confidence score $\tilde{c}$. Given the corresponding keypoints and confidence scores, the optimal transformation $\mathbf{R}^*, \mathbf{t}^*$ can be calculated as

$$\mathbf{R}^*, \mathbf{t}^* = \underset{\mathbf{R},\mathbf{t}}{\arg\min} \sum_{i}^{M} \tilde{c}_i \left\| \mathbf{R} x_i^{\mathcal{S}} + \mathbf{t} - \tilde{x}_i^{\mathcal{T}} \right\|_2 \tag{1}$$

where $x_i^{\mathcal{S}}$ and $\tilde{x}_i^{\mathcal{T}}$ are corresponding keypoints, $\tilde{c}_i$ is confidence score and $\|\cdot\|_2$ denotes L_2 norm. Eq. 1 can be closed-form solved using weighted Kabsch algorithm [16], which has also been derived in detail in [11].

3.2.2 Similarity features

Bilateral consensus: Based on the kNN search, we can only ensure that the searched K candidate keypoints in $X^{\mathcal{T}}$ are most similar with the keypoint $x^{\mathcal{S}}$. However, this single directional operation can not guarantee the reverse best similarity of the matching. Intuitively, a correct correspondence should satisfy bilateral consensus, which means that if $x_j^{\mathcal{T}}$ is the nearest neighbor (in descriptor space) in $X^{\mathcal{T}}$ of $x_i^{\mathcal{S}}$, then $x_i^{\mathcal{S}}$ should also be the nearest neighbor in $X^{\mathcal{S}}$ of $x_j^{\mathcal{T}}$.

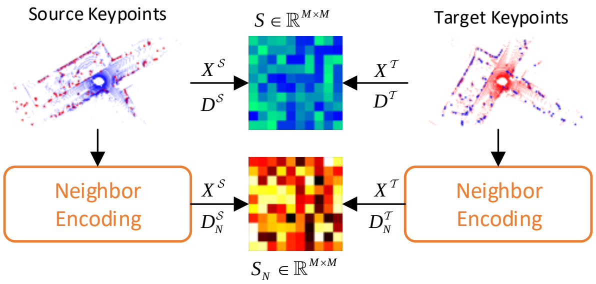


Figure 4. Illustration of similarity matrix. Given source and target keypoints, we calculate the cosine similarity of the descriptors to form $S \in \mathbb{R}^{M\times M}$. The neighbor encoding module is adopted to gather neighborhood information and the similarity matrix S_N is calculated based on the neighbor-aware descriptors $D_N^{\mathcal{S}}$, $D_N^{\mathcal{T}}$.

Based on the above consideration, we introduce novel similarity features to take bilateral consensus into consideration. As shown in the top row of Fig. 4, for each keypoint $x_i^{\mathcal{S}}$, we calculate the cosine similarity of the descriptor $d_i^{\mathcal{S}}$ with descriptors of all keypoints in $X^{\mathcal{T}}$. Consequently, we can obtain a $M \times M$ similarity matrix and an entry s_{ij} of the similarity matrix $S \in \mathbb{R}^{M\times M}$ can be calculated as

$$s_{ij} = \frac{\langle d_i^{\mathcal{S}}, d_j^{\mathcal{T}} \rangle}{\left\| d_i^{\mathcal{S}} \right\|_2 \left\| d_j^{\mathcal{T}} \right\|_2} \tag{2}$$

where $\langle \cdot, \cdot \rangle$ and $\|\cdot\|_2$ denote inner product and L_2 norm.

After that, we normalize the similarity matrix in two directions to generate two different similarity matrixes S^f (forward matrix) and S^b (backward matrix) as

$$s_{ij}^f = \frac{s_{ij}}{\max_m s_{im}}, \quad s_{ij}^b = \frac{s_{ij}}{\max_m s_{mj}} \tag{3}$$

Then, the similarity features of the cluster are the concatenation of corresponding similarity scores of the candidate keypoints with the center keypoint in forward and backward similarity matrix S^f and S^b. Take a pair of candidate corresponding keypoints $\{x_i^{\mathcal{S}}, x_j^{\mathcal{T}}\}$ as an example, then the similarity features of this correspondence can be represented as $[s_{ij}^f, s_{ij}^b]$. The similarity features implicitly model bilateral consensus. If $x_j^{\mathcal{T}}$ is the most similar keypoint of $x_i^{\mathcal{S}}$ among all keypoints in $X^{\mathcal{T}}$, then $s_{ij}^f = 1$. Then, if $x_i^{\mathcal{S}}$ is also the most similar keypoint in $X^{\mathcal{S}}$ of $x_j^{\mathcal{T}}$, s_{ij}^b will also be equal to 1, otherwise $s_{ij}^b < 1$ because the best similarity score will not fall in s_{ij} in this case. Thus, s_{ij}^f and s_{ij}^b will both be equal to 1 only if the correspondence between $x_i^{\mathcal{S}}$ and $x_j^{\mathcal{T}}$ satisfies bilateral consensus.

Neighborhood consensus: In addition to bilateral consensus, neighborhood consensus is also important for good correspondence, which means that the neighboring keypoints of two corresponding keypoints should have similar features. To exploit neighborhood consensus, we propose an attention-based neighbor encoding module to gather the information of neighboring keypoints to generate neighbor-aware descriptors. Take a keypoint $x^{\mathcal{S}}$ in $X^{\mathcal{S}}$ as an example, we firstly perform kNN spatially to search K neighboring keypoints in $X^{\mathcal{S}}$ to form a cluster. The features of the cluster consist of the descriptors of neighboring keypoints, relative coordinates and relative distances from neighboring to center keypoints. The cluster features are input into a Shared-MLP to generate a feature map. After that, a max-pool layer and a Softmax function are followed to predict attentive weights for each neighboring keypoint. The neighbor-aware descriptor $d_N^{\mathcal{S}}$ of $x^{\mathcal{S}}$ can be calculated as the weighted sum of the neighboring descriptors. Thus, the similarity of neighbor-aware descriptors can encode the similarity of neighboring keypoints. As shown in the bottom row of Fig. 4, using the neighbor-aware descriptors $D_N^{\mathcal{S}}$ and $D_N^{\mathcal{T}}$, we generate a neighbor-aware similarity matrix $S_N \in \mathbb{R}^{M\times M}$ through the similar method described before.

Finally, the similarity features F_S consist of two parts, namely F_S^O and F_S^N, where F_S^O denotes the similarity features from original similarity matrix S and F_S^N denotes that from the neighbor-aware similarity matrix S_N. Consequently, the introduction of similarity features F_S is able to simultaneously incorporate bilateral consensus and neighborhood consensus into the registration pipeline implicitly.

3.3. Fine registration

After applying coarse registration in layer $l = 3$, we obtain the coarse transformation $\mathbf{R}_3, \mathbf{t}_3$. Fine registration is applied in upper layers to reduce the registration error caused by the sparsity of the keypoints in deeper layers.

Take the middle layer $l = 2$ as an example, we firstly transform the source keypoints using the coarse transformation $\mathbf{R}_3, \mathbf{t}_3$. We assume that the coarse registration can provide a correct but not accurate enough estimation. Thus, the corresponding target keypoint $\tilde{x}^{\mathcal{T}}$ of a source keypoint $x^{\mathcal{S}}$ should be spatially close to $x^{\mathcal{S}}$ after the coarse transformation. Based on the above assumption, for a source keypoint $x^{\mathcal{S}}$, we perform kNN search locally in its spatial neighborhoods rather than in descriptor space to find K candidate corresponding keypoints to construct a cluster. Different from coarse registration, the features of cluster in fine registration only include geometric features F_G and descriptor features F_D. The similarity features are dropped here due to the computational complexity for a larger number of keypoints in upper layers. We then apply a similar correspondence network on the cluster to generate keypoints correspondences and confidence scores. Weighted Kabsch algorithm is followed to calculate the transformation $\Delta\mathbf{R}_2, \Delta\mathbf{t}_2$. Then the transformation $\mathbf{R}_2, \mathbf{t}_2$ after the fine registration in layer $l = 2$ can be calculated as $\mathbf{R}_2 = \Delta\mathbf{R}_2\mathbf{R}_3, \mathbf{t}_2 = \Delta\mathbf{R}_2\mathbf{t}_3 + \Delta\mathbf{t}_2$. Similarly, another fine registration is applied in the top layer $l = 1$ based on the *coarse transformation* $\mathbf{R}_2, \mathbf{t}_2$ to get the final registration result $\hat{\mathbf{R}}, \hat{\mathbf{t}}$.

To summarize, the hierarchical structure leverages robust features in bottom layer and accurate position information in upper layers to achieve reliable and precise registration.

3.4. Loss function

The loss function $\mathcal{L} = \mathcal{L}_{trans} + \alpha\mathcal{L}_{rot}$, where $\mathcal{L}_{trans}$ and $\mathcal{L}_{rot}$ are translation and rotation loss, respectively. Given estimated and ground truth transformation $\hat{\mathbf{R}}, \hat{\mathbf{t}}$ and $\mathbf{R}, \mathbf{t}$, $\mathcal{L}_{trans}$ and $\mathcal{L}_{rot}$ can be calculated as

$$\mathcal{L}_{trans} = \left\|\mathbf{t} - \hat{\mathbf{t}}\right\|_2 \tag{4}$$

$$\mathcal{L}_{rot} = \left\|\hat{\mathbf{R}}^T\mathbf{R} - \mathbf{I}\right\|_2 \tag{5}$$

where $\mathbf{I}$ denotes identity matrix.

4. Experiments

4.1. Experiment settings

Datasets: We perform extensive experiments on two large-scale outdoor LiDAR point cloud datasets, namely KITTI odometry dataset [10] and NuScenes dataset [4]. KITTI dataset consists of 11 sequences (00 to 10) with ground truth vehicle poses and we use Sequence 00 to 05 for training, 06 to 07 for validation and 08 to 10 for testing. We use the current frame with the 10th frame after that to form a pair of point clouds. To reduce the noise of ground truth vehicle poses, we perform Iterative Closest Point (ICP) algorithm in Open3D library [41] to refine the noisy relative transformation between two point clouds. NuScenes dataset includes 1000 scenes, among which 850 scenes are used for training and validation and 150 scenes for testing. We use the first 700 scenes in the 850 scenes to train the network and the other 150 scenes for validation. NuScenes dataset only provides the ground truth poses of the given samples and the time interval between two consecutive point cloud samples is about 0.5s. We use the current point cloud sample with the second sample after it as a pair of point clouds.

Implementation details: In the pre-processing, we firstly voxelize the input point clouds and the voxel size is set to 0.3m. After that, we randomly sample 16384 points from the point clouds in KITTI dataset and 8192 points in NuScenes dataset. The network is implemented using PyTorch [26] and we use Adam [17] as the optimizer. The learning rate is initially set to 0.001 and decreases by 50% every 10 epochs. The hyperparameter α in the loss function $\mathcal{L}$ is set to 1.8 for KITTI dataset and 2.0 for NuScenes dataset. When training the network, we firstly pre-train the feature extraction module and then train the whole network based on the pre-trained features. The whole network is trained on an NVIDIA RTX 3090 GPU. The details of pre-training and the network architecture are described in the supplementary material.

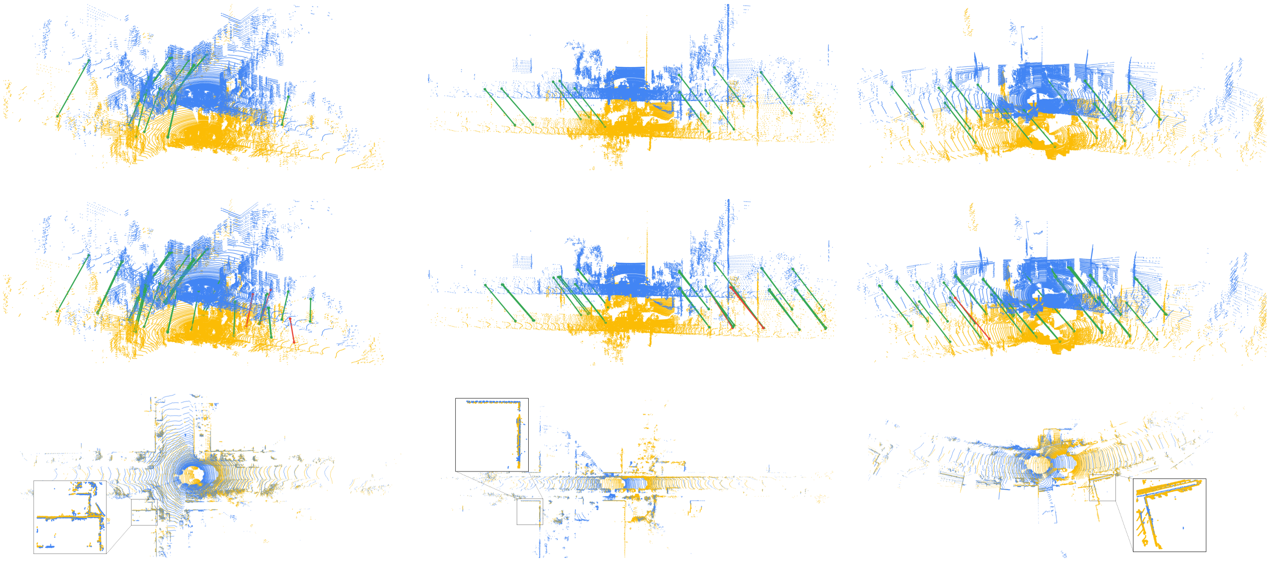


Figure 5. Qualitative visualization of the proposed point cloud registration method. We display 3 samples of point cloud registration here. The first row displays the correspondences between source and target keypoints in coarse registration with confidence score $\tilde{c} > 0.005$ and the second row displays the correspondences with confidence score $\tilde{c} > 0.0005$. The green lines and red lines represent inlier and outlier correspondences, respectively. The bottom row shows the aligned two point clouds and we zoom in an area for better visualization.

Baseline methods: We compare the performance of the proposed HRegNet with both classical and learning-based methods. All of the methods are tested on an Intel i9-10920X CPU and an NVIDIA RTX 3090 GPU.

Classical methods: We evaluate the performance of point to point ICP (ICP (P2Point)), point to plane ICP (ICP (P2Plane)) [3], RANSAC [8], and Fast Global Registration (FGR) [40]. All of the classical methods are implemented using Open3D library [41]. For RANSAC and FGR, we extract Fast Point Feature Histograms (FPFH) [32] from 0.3m-voxel-downsampled point clouds. The maximum iteration number of RANSAC is set to 2e6[1].

Learning-based methods: We choose 4 representative learning-based methods to compare with the proposed HRegNet[2]. (1) Deep Closest Point (DCP) [35]: DCP is a pioneering work for learning-based point cloud registration. For the pre-processing of point clouds, 4096 points are randomly sampled from 0.3m-voxel-downsampled point clouds for both datasets. (2) IDAM [19]: IDAM is one of the state-of-the-art object-level point cloud registration methods. The pre-processing is the same as that for DCP. (3) Feature-metric Registration (FMR) [14]: FMR has been evaluated for both object-level and indoor point cloud registration. The pre-processing of point clouds is the same as that in our methods. (4) Deep Global Registration (DGR) [5]: DGR achieves state-of-the-art performance in indoor point cloud registration. The point clouds are voxelized with 0.3m voxel size. All the learning-based baseline methods are retrained on both datasets for better performance.

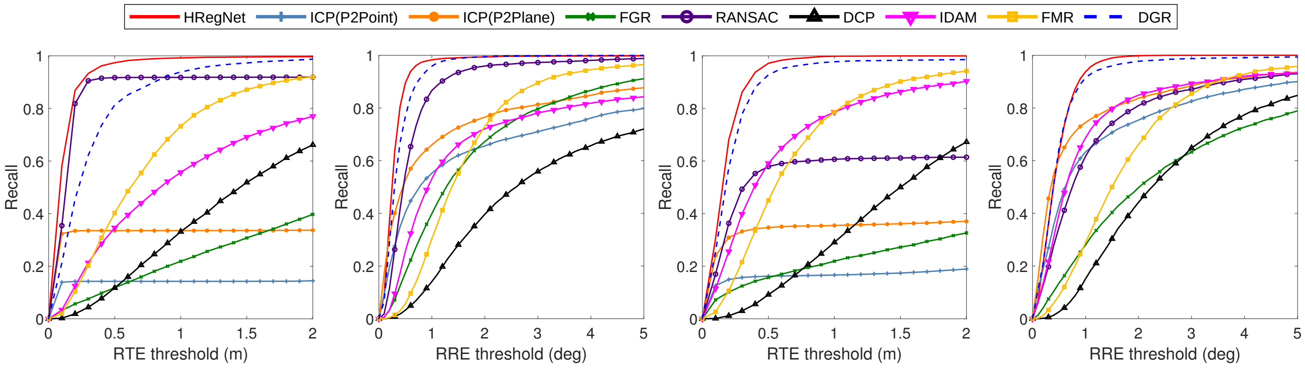
4.2. Evaluation

Qualitative visualization: We display several qualitative samples of point cloud registration in Fig. 5. Corresponding keypoints in coarse registration with confidence scores $\tilde{c} > 0.005$ and $\tilde{c} > 0.0005$ are displayed in the first and second row respectively. Two corresponding keypoints are considered as an inlier if the relative position error (after applying the ground truth relative transformation) less than a distance threshold $\epsilon_d = 1$m. The green and red lines represent inlier and outlier correspondences, respectively. According to the results, the correspondences with larger confidence score ($\tilde{c} > 0.005$) are basically all inliers and several mismatches start to appear when reducing the threshold of $\tilde{c}$ to 0.0005. The qualitative results show that the correspondence network can generate accurate and correct correspondence of keypoints and the predicted confidence score can effectively reject unreliable correspondences. The third row of Fig. 5 displays the two aligned point clouds, which demonstrates that the network can precisely predict the transformation. More qualitative results are displayed in our supplementary material.

Quantitative evaluation: We adopt relative translation error (RTE) and relative rotation error (RRE) to evaluate the registration performance. RTE can be calculated as Eq. 4

[1] We have tried more iterations, however, the accuracy will not be obviously improved while the computational time will increase significantly.

[2] We also try to compare our method with DeepVCP[21], however, the source code has not been released by the author and the self-implemented version does not provide reasonable results.



(a) KITTI dataset (b) NuScenes dataset

Figure 6. Registration recall with different RRE and RTE thresholds on KITTI dataset and NuScenes dataset.

Table 1. Registration performance on KITTI dataset and NuScenes dataset.

Methods	KITTI dataset				NuScenes dataset			
	RTE (m)	RRE (deg)	Recall	Time (ms)	RTE (m)	RRE (deg)	Recall	Time (ms)
ICP (P2Point) [3]	0.04 ± 0.05	0.11 ± 0.09	14.3%	472.2	0.25 ± 0.51	0.25 ± 0.50	18.8%	82.0
ICP (P2Plane) [3]	0.04 ± 0.04	0.14 ± 0.15	33.5%	461.7	0.15 ± 0.30	0.21 ± 0.31	36.8%	44.5
FGR [40]	0.93 ± 0.59	0.96 ± 0.81	39.4%	506.1	0.71 ± 0.62	1.01 ± 0.92	32.2%	284.6
RANSAC [8]	0.13 ± 0.07	0.54 ± 0.40	91.9%	549.6	0.21 ± 0.19	0.74 ± 0.70	60.9%	268.2
DCP [35]	1.03 ± 0.51	2.07 ± 1.19	47.3%	46.4	1.09 ± 0.49	2.07 ± 1.14	58.6%	45.5
IDAM [19]	0.66 ± 0.48	1.06 ± 0.94	70.9%	33.4	0.47 ± 0.41	0.79 ± 0.78	88.0%	32.6
FMR [14]	0.66 ± 0.42	1.49 ± 0.85	90.6%	85.5	0.60 ± 0.39	1.61 ± 0.97	92.1%	61.1
DGR [5]	0.32 ± 0.32	0.37 ± 0.30	98.7%	1496.6	0.21 ± 0.18	0.48 ± 0.43	98.4%	523.0
HRegNet	$\mathbf{0.12 \pm 0.13}$	$\mathbf{0.29} \pm 0.25$	**99.7%**	106.2	$\mathbf{0.18 \pm 0.14}$	$\mathbf{0.45 \pm 0.30}$	**99.9%**	87.3

and RRE can be represented as $\arccos(\mathrm{Tr}(\hat{\mathbf{R}}^T\mathbf{R} - 1)/2)$, where $\hat{\mathbf{R}}$ and $\mathbf{R}$ are the estimated and ground truth rotation matrix. Registration recall is defined as the ratio of successful registration. A registration is considered as successful when the RTE and RRE are within the thresholds ϵ_{trans} and ϵ_{rot}. We display the registration recall with different RTE and RRE thresholds on two datasets in Fig. 6. According to the results, the proposed HRegNet outperforms all baseline methods by an obvious margin on both two datasets. Besides, for a more detailed comparison of the registration performance, we calculate the average RRE and RTE and display the results in Table 1. Noting that a part of failed registrations can result in dramatically large RRE and RTE, which can cause unreliable error metrics. Thus, the average RTE and RRE are only calculated for successful registrations and the thresholds are set as $\epsilon_{trans} = 2$m and $\epsilon_{rot} = 5$deg. The registration recall at the given threshold is also displayed in Table 1.

According to the results, ICP algorithms (for both ICP (P2Point) and ICP (P2Plane)) fail to generate reasonable relative transformation in most cases due to the lack of precise initial transformation between two point clouds. FGR performs slightly better than ICP, however, the registration recall is still below 50%, which is unacceptable in applications. RANSAC achieves the best performance among the classical methods thanks to the powerful outlier rejection mechanism, however, the iterative paradigm can also result in poor efficiency. The average RTE of RANSAC is similar to ours method, however, it is due to a number of mismatches are omitted in the calculation and the registration recall of RANSAC is obviously lower than the proposed method according to Fig. 6. Moreover, the runtime of our method is almost 1/5 of RANSAC on KITTI dataset.

As for the learning-based methods, the recall of DCP on KITTI and NuScenes dataset are both less than 60% and the average RTE and RRE are also quite large. IDAM performs better than DCP, however, the recall is still only about 70% on KITTI dataset and the RTE and RRE are much higher than the proposed method, which indicates the poor applicability of the object-level point cloud registration methods to complex large-scale LiDAR point clouds. FMR achieves a slightly faster speed than our method, however, the registration error is much higher than ours. For example, the RTE of FMR on KITTI dataset is more than 5 times of our method. DGR achieves the best registration performance among all the learning-based baseline methods. However, the 6D convolutional network-based outlier rejection method is time-consuming and the voxel-based representation of point clouds limits the precision of the registration. The RTE of our method is almost 1/3 of that of DGR on KITTI dataset. Moreover, our method achieves almost 15× faster speed than DGR on KITTI dataset.

Overall, extensive experiments demonstrate that the proposed HRegNet achieves state-of-the-art performance in

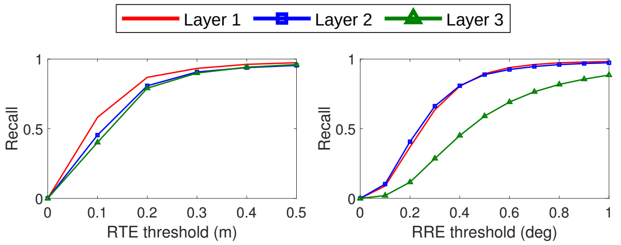


Figure 7. Registration recall of different output layers on KITTI dataset. We set the range of RTE threshold as from 0 to 0.5m and RRE threshold as from 0 to 1deg for better visualization. Layer 1: top layer; Layer 2: middle layer; Layer 3: bottom layer.

terms of both accuracy and efficiency.

4.3. Ablation study

We perform abundant ablation studies on KITTI dataset to demonstrate the effectiveness of the hierarchical structure and the introduction of the similarity features.

Hierarchical structure: To validate the effectiveness of the hierarchical structure, we use the output transformation $\mathbf{R}, \mathbf{t}$ from layer 3 to layer 1 as the final estimation respectively to evaluate the performance. The network with different output layers is trained separately using the same hyperparameters. The registration recall with different output layers is displayed in Fig. 7. The detailed average RRE and RTE is shown in Table 2 and the calculation settings are the same as that in Table 1. According to the results, the average RTE and RRE are gradually reduced with the layer-by-layer refinement. The results in layer 2 achieve much lower rotation error than layer 3. And the translation accuracy in layer 1 (*i.e.*, the full model) is also obviously improved compared to layer 2, which demonstrates the validity of hierarchical refinement strategy. Noting that the registration recall with different RRE thresholds of layer 1 is almost the same as layer 2 and we found that further increasing the number of layers will not result in significant improvements in registration performance, however, will deteriorate the efficiency of the network. Considering the trade-off between accuracy and efficiency, we choose the 3-layer implementation.

Similarity features: As we described before, the similarity features F_S consist of two parts, namely the original similarity features F_S^O and neighbor-aware similarity features F_S^N. To analysis the impact of the two parts on the performance, we drop F_S^O and F_S^N separately and retrain the network. The registration recall and the average RRE and RTE of the full model and the model without F_S^O, F_S^N and F_S are displayed in Fig. 8 and Table 2. According to the results, the registration recall without both similarity features F_S is inferior to the other cases by a significant margin, which demonstrates the importance of the bilateral consensus. The neighbor-aware similarity features F_S^N incorporate the information of neighboring keypoints into consideration, however, may also lead to the neglect of the own unique features of the keypoint. Thus, the original and neighbor-aware similarity features are complementary to each other and the combination of the two (*i.e.*, the full model) outperforms other cases. Overall, the results demonstrate that the introduction of the similarity features significantly improves the performance.

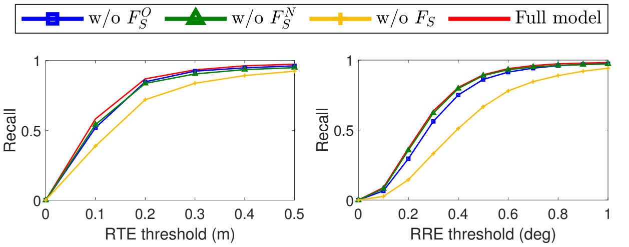


Figure 8. Registration recall with and without (w/o) similarity features on KITTI dataset. F_S^O: original similarity features. F_S^N: neighbor-aware similarity features. F_S: both similarity features.

Table 2. Ablation studies on KITTI dataset.

Model	RTE (m)	RRE (deg)	Recall	Time (ms)
Full	$\mathbf{0.12 \pm 0.13}$	$\mathbf{0.29 \pm 0.25}$	**99.7%**	106.2
Layer 2	0.15 ± 0.18	0.29 ± 0.27	99.2%	101.4
Layer 3	0.16 ± 0.18	0.55 ± 0.45	99.7%	96.9
w/o F_S^O	0.15 ± 0.19	0.31 ± 0.30	99.1%	98.6
w/o F_S^N	0.14 ± 0.17	0.33 ± 0.29	99.4%	96.4
w/o F_S	0.19 ± 0.22	0.46 ± 0.36	98.7%	88.0

5. Conclusion

In this paper, we provide an efficient hierarchical network for large-scale outdoor LiDAR point cloud registration. The hierarchical paradigm leverages different characteristics of keypoints and descriptors in deeper and shallower layers by introducing coarse registration and fine registration in different layers. To construct reliable correspondences between keypoints, we propose a correspondence network to generate corresponding keypoints. Moreover, novel similarity features are designed to effectively incorporate bilateral consensus and neighborhood consensus into the registration pipeline. Abundant ablation studies demonstrate the effectiveness of the hierarchical paradigm and the introduction of similarity features. Besides, the network is also highly efficient since we only use a small number of keypoints for registration. Extensive experiments on two large-scale LiDAR point cloud datasets demonstrate the high accuracy and efficiency of the proposed HRegNet.

Acknowledgments: This work is funded by the Key Technologies Developement and Application of Piloted Autonomous Driving Trucks Project, and the Shanghai Rising Star Program (No. 21QC1400900), and the National Key Research and Development Program of China (No.2016YFB0100901).

References

[1] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7163–7172, 2019.

[2] Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3feat: Joint learning of dense detection and description of 3d local features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6359–6367, 2020.

[3] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[5] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2514–2523, 2020.

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

[7] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8958–8966, 2019.

[8] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[9] Alex Flint, Anthony Dick, and Anton Van Den Hengel. Thrift: Local 3d structure recognition. In *9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA 2007)*, pages 182–188. IEEE, 2007.

[10] 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.

[11] Zan Gojcic, Caifa Zhou, Jan D Wegner, Leonidas J Guibas, and Tolga Birdal. Learning multiview 3d point cloud registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1759–1769, 2020.

[12] Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, Jianwei Wan, and Ngai Ming Kwok. A comprehensive performance evaluation of 3d local feature descriptors. *International Journal of Computer Vision*, 116(1):66–89, 2016.

[13] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[14] Xiaoshui Huang, Guofeng Mei, and Jian Zhang. Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11366–11374, 2020.

[15] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence*, 21(5):433–449, 1999.

[16] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015.

[18] Jiaxin Li and Gim Hee Lee. Usip: Unsupervised stable interest point detection from 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 361–370, 2019.

[19] Jiahao Li, Changhao Zhang, Ziyao Xu, Hangning Zhou, and Chi Zhang. Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration. In *European Conference on Computer Vision (ECCV)*, 2020.

[20] Fan Lu, Guang Chen, Yinlong Liu, Zhongnan Qu, and Alois Knoll. Rskdd-net: Random sample-based keypoint detector and descriptor. *Advances in Neural Information Processing Systems*, 33, 2020.

[21] Weixin Lu, Guowei Wan, Yao Zhou, Xiangyu Fu, Pengfei Yuan, and Shiyu Song. Deepvcp: An end-to-end deep neural network for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–21, 2019.

[22] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[23] Haggai Maron, Nadav Dym, Itay Kezurer, Shahar Kovalsky, and Yaron Lipman. Point registration via efficient convex relaxation. *ACM Transactions on Graphics (TOG)*, 35(4):1–12, 2016.

[24] Balázs Nagy and Csaba Benedek. Real-time point cloud alignment for vehicle localization in a high resolution 3d map. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[25] G Dias Pais, Srikumar Ramalingam, Venu Madhav Govindu, Jacinto C Nascimento, Rama Chellappa, and Pedro Miraldo. 3dregnet: A deep neural network for 3d point registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7193–7203, 2020.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit

Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. 2019.

[27] François Pomerleau, Francis Colas, and Roland Siegwart. A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends in Robotics*, 4(1):1–104, 2015.

[28] 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.

[29] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[30] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[31] David M Rosen, Luca Carlone, Afonso S Bandeira, and John J Leonard. Se-sync: A certifiably correct algorithm for synchronization over the special euclidean group. *The International Journal of Robotics Research*, 38(2-3):95–125, 2019.

[32] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE, 2009.

[33] Ivan Sipiran and Benjamin Bustos. Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. *The Visual Computer*, 27(11):963–976, 2011.

[34] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *European conference on computer vision*, pages 356–369. Springer, 2010.

[35] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3523–3532, 2019.

[36] Yue Wang and Justin M Solomon. Prnet: Self-supervised learning for partial-to-partial registration. *arXiv preprint arXiv:1910.12240*, 2019.

[37] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-icp: A globally optimal solution to 3d icp point-set registration. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2241–2254, 2015.

[38] Zi Jian Yew and Gim Hee Lee. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 607–623, 2018.

[39] Wentao Yuan, Benjamin Eckart, Kihwan Kim, Varun Jampani, Dieter Fox, and Jan Kautz. Deepgmr: Learning latent gaussian mixture models for registration. In *European Conference on Computer Vision*, pages 733–750. Springer, 2020.

[40] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *European Conference on Computer Vision*, pages 766–782. Springer, 2016.

[41] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.

[42] Yao Zhou, Guowei Wan, Shenhua Hou, Li Yu, Gang Wang, Xiaofei Rui, and Shiyu Song. Da4ad: End-to-end deep attention-based visual localization for autonomous driving. In *European Conference on Computer Vision*, pages 271–289. Springer, 2020.