A Representation Separation Perspective to Correspondences-free Unsupervised 3D Point Cloud Registration

Zhiyuan Zhang¹, Jiadai Sun¹, Yuchao Dai¹ Member, IEEE, Dingfu Zhou², Xibin Song², Mingyi He¹ Senior Member, IEEE

Abstract—3D point cloud registration in remote sensing field has been greatly advanced by deep learning based methods, where the rigid transformation is either directly regressed from the two point clouds (correspondences-free approaches) or computed from the learned correspondences (correspondences-based approaches). Existing correspondences-free methods generally learn the holistic representation of the entire point cloud, which is fragile for partial and noisy point clouds. In this paper, we propose a correspondences-free unsupervised point cloud registration (UPCR) method from the representation separation perspective. First, we model the input point cloud as a combination of pose-invariant representation and pose-related representation. Second, the pose-related representation is used to learn the relative pose w.r.t. a “latent canonical shape” for the source and target point clouds respectively. Third, the rigid transformation is obtained from the above two learned relative poses. Our method not only filters out the disturbance in pose-invariant representation but also is robust to partial-to-partial point clouds or noise. Experiments on benchmark datasets demonstrate that our unsupervised method achieves comparable if not better performance than state-of-the-art supervised registration methods.

Index Terms—Correspondences-free, point cloud registration, representation separation, unsupervised

I. INTRODUCTION

RECENTLY, with the widespread application of LiDAR, point cloud has become an important data source for 3D remote sensing, which presents rich 3D spatial information efficiently [10], [27]. Meanwhile, to obtain a larger perceptive field and ensure the integrity of the objects in point cloud data, 3D point cloud registration has been widely investigated. The 3D point cloud registration task aims to estimate a rigid motion to align two 3D point clouds, which is a key component in many applications ranging from remote object recognition and segmentation [29], [13], simultaneous localization and mapping (SLAM) [8], to autonomous driving [22], etc. Most of the current 3D point cloud registration methods are built upon correspondences, devoting to minimizing the registration error between correspondences [14], [9], [6], [28]. These methods are usually solved through two steps: correspondences building and transformation estimation. However, these correspondences-based methods, whether hand-crafted or learning-based, often suffer from a fatal drawback because they cannot handle outliers well.

As an alternative, the correspondences-free methods solve the rigid motion by comparing the holistic representations of the two point clouds. In the seminal work PointNetLK [1], PointNet [15] is used as the global feature extractor, and a modified LK [12] algorithm is designed for motion estimation. PCRNet [19] inherits this framework and solves the rigid motion through regression. Note that these supervised methods rely on extensive registration label data. Feature-metric [11] is another correspondences-free method, which can be trained in a semi-unsupervised or unsupervised manner. This method devotes to designing feature extractors from the reconstruction perspective but does not exhibit sufficient robustness.

Essentially, these correspondences-free methods [1], [19], [11] can be understood as solving the relative pose consuming the difference in global representations of the source and target point clouds. However, this operation is inaccurate and unreasonable, especially for partial-to-partial cases, which are common in the remote sensing field. Intuitively, the global representation of a point cloud can be viewed as a combination of the pose-related part (i.e., rotation and translation) and the pose-invariant part (e.g., shape, geometry structure, etc.). Thus, using the global representation to regress the final relative pose is problematic. To deal with cases such as partial-view and outliers, which are pose-invariant, the resulting network has to work as a filter due to the asymmetry of the inputs and outputs, thus leading to inaccurate performance.

To solve this problem, we propose a correspondences-free unsupervised 3D point cloud registration (UPCR) method to learn the relative pose only from the pose-related part rather than from the global representation. Specifically, UPCR consists of three steps. First, we separate the input point cloud to a pose-invariant representation and a pose-related representation. Second, the pose-related representation is used to learn a relative pose w.r.t. a “latent canonical shape” for the source and target point clouds correspondingly. Finally, the 3D rigid transformation is obtained from the above two learned relative poses. In this way, the disturbance in pose-invariant representation is filtered out, which enables the high robustness of our method to partial point cloud and outliers. We train our model in an unsupervised learning manner. Experiments
on benchmark datasets demonstrate that our unsupervised method achieves comparable if not better performance than supervised 3D point cloud registration methods.

Our main contributions can be summarized as follows: 1) We present a correspondences-free unsupervised point cloud registration method from the representation separation perspective; 2) We design a representation separation module to decouple the pose-related representation from the global representation; 3) Experimental results on benchmark datasets validate the superiority and robustness of our method.

II. PROPOSED APPROACH

Given the source and the target point clouds \( X \in \mathbb{R}^{3 \times N_X} \), \( Y \in \mathbb{R}^{3 \times N_Y} \), the 3D registration problem aims at estimating the relative pose to achieve the best alignment, which can be modeled by a rotation matrix \( R \in SO(3) \) and a translation vector \( t \in \mathbb{R}^3 \). We tackle this problem from a novel representation separation perspective. Specifically, we model the input point clouds as a combination of pose-invariant and pose-related representations. As the pose-invariant part does not provide motion estimation cues, we solve the relative pose only from the pose-related representation. The whole objective is formulated as:

\[
\min_{\{T_X, T_Y\}} \psi(R_X^{-1}(X - t_X), R_Y^{-1}(Y - t_Y)),
\]

where \( T_X = [R_X t_X] \) and \( T_Y = [R_Y t_Y] \) denote the relative poses learned only from the pose-related parts w.r.t. “latent canonical shape” \( X_c, Y_c \), i.e. \( X = R_X X_c + t_X \), \( Y = R_Y Y_c + t_Y \). Note that \( X_c = R_X^{-1}(X - t_X) \) and \( Y_c = R_Y^{-1}(Y - t_Y) \) are defined as the “latent canonical shape” with a “canonical pose”, which are obtained by recovering the learned relative poses from the given shapes. Compared with setting a certain pose as the canonical pose artificially [21], our canonical pose is learned in a data-driven manner. \( \psi(\cdot, \cdot) \) is a metric function, which measures the difference of the input two canonical shapes. In this paper, we first summarize the pose-invariant representations of source and target denoted as \( \Gamma^X, \Gamma^Y \), and the global representations as \( \Gamma^X_G, \Gamma^Y_G \). Then a special subtraction is designed to solve the pose-related representations as \( \Gamma^X_P, \Gamma^Y_P \) to regress respective relative pose \( T_X, T_Y \) by subtracting pose-invariant representation from the global representation respectively. This special subtraction is introduced in detail below imitating the relative entropy.

A. Representation Separation for Registration

Since we are modeling the point cloud as a combination of the pose-invariant part and the pose-related part, a natural idea is to remove the pose-invariant part from the global representation and what remains will be the pose-related part.

Global Representation. The global representation summarizes all information from the spatial coordinates of 3D points. Under the deep learning framework, many global representation extraction networks are proposed as standard modules [4], [5], [7]. A pioneering work PointNet [15] pools an arbitrary number of orderless points into a global descriptor. PointNet cannot summarize the local structure, while PointNet++ [16] alleviates this problem by a hierarchical structure. It is worth mentioning that DGCNN [25] encodes the local geometric characteristics by graph neural network (GNN) and achieves superior performance. Thus, we employ the GNN as our global representation extractor.

Considering a point cloud \( X \), GNN constructs a K-nearest neighbor (KNN) graph, applies a non-linearity to the edge values, and performs vertex-wise aggregation (max-pooling) in each layer for point features, i.e.,

\[
F^\ell(x_i) = f\left( \{ h^\ell_0 (F^{\ell-1}(x_i), F^{\ell-1}(x_{ij})) \mid \forall x_{ij} \in \mathcal{N}(x_i) \} \right),
\]

where \( F^\ell(x_i) \) is the feature of point \( x_i \) in the \( \ell \)-th layer GNN, and \( h^\ell_0(\cdot) \) is a multi-layer perceptron (MLP) function parameterized by \( \theta \). \( f(\cdot) \) represents the symmetry function max-pooling, and \( \mathcal{N}(x_i) \) denotes the KNN neighbors of point \( x_i \), i.e. \( x_{ij} \) is the \( j \)-th neighbor point, \( j \in [1, K] \). \( K \) is pre-defined. As a result, the global representation can be obtained by the max-pooling acting on all point features, \( \Gamma^X_G = f(\{ F^\ell(x_i) i = 1, \ldots, N_X \}) \in \mathbb{R}^{1 \times m} \), \( m \) is the pre-defined representation vector dimension. Similarly, \( \Gamma^Y_G = f(\{ F^\ell(y_i) i = 1, \ldots, N_Y \}) \in \mathbb{R}^{1 \times m} \).

Pose-Invariant Part Representation. We extract the pose-invariant part representation of one point cloud via pose invariant feature. There are usually two methods to obtain invariant features, i.e., data augmentation and pose-invariant network. Data augmentation is easy to implement, but it is time-consuming. Thus, we adopt the second strategy to design a concise but effective feature called distance-based feature.

Considering a point cloud \( X \), whose center is denoted as \( o \), the distance-based point feature \( \phi \) is designed as

\[
\phi(x_{ij}) = D(x_{ij}, o), D(x_i, x_{ij}), D(o, x_{ij}),
\]

where \( x_{ij} \in \mathcal{N}(x_i) \) is a neighbor point as mentioned above. Function \( D(\cdot, \cdot) \) calculates the Euclidean distance between two 3D points. \( [\cdot, \cdot, \cdot] \) combines these three input elements to a point feature with 3 dimensions. Hence, \( \phi(x_{ij}) \) maintains invariance to arbitrary rigid transformations. Then, using all neighbor points yields the pose-invariant feature of point \( x_i \),

\[
\Phi^0(x_i) = f(\{ h_\alpha(\phi(x_{ij})) \mid \forall x_{ij} \in \mathcal{N}(x_i) \}),
\]

where \( h_\alpha \) is a MLP network parameterized by \( \alpha \). According to Eq. (2), a GNN network is applied to purify the pose-invariant representation as:

\[
\Gamma^X_P = f(\{ \Phi^\ell(x_i) i = 1, \ldots, N_X \}) \in \mathbb{R}^{1 \times m}.
\]

Note that the dimension of pose-invariant part representation is same as the global representation. Similarly, \( \Gamma^Y_P = f(\{ \Phi^\ell(y_i) i = 1, \ldots, N_Y \}) \in \mathbb{R}^{1 \times m} \). Notably, while our distance-based feature characterizes pose-invariant feature, other pose-invariant feature can also play the same role within our pipeline. In Sec. III, we also report the performance of merging other advanced pose-invariant features including PFH [18], SPFH [17] into our UPCR method. We find that, these more informative pose-invariant feature can generate more accurate registration results. And designed distance-based features helps to achieve advanced time-efficiency and registration performance simultaneously.

Pose-Related Part Representation. Our core idea is to estimate the rigid motion by learning relative poses w.r.t. a “latent canonical shape” for source and target point clouds from pose-related part representations, which is more reasonable and accurate. To model this pose-related part representation, we propose to “subtract” pose-invariant part from global representation. Then, a natural question is, how to formulate...
Chamfer distance to measure the discrepancy between coordinate by source and target. We design a novel and elegant “representation subtraction”.

Considering that the relative entropy often works as a “distance” between distributions, we formulate our subtraction analogously. Specifically, we first construct the probability distribution of the global representation and the pose-invariant representation, i.e. \( q = \text{softmax}(\Gamma_{G}) \), \( p = \text{softmax}(\Gamma_{V}) \). Then, we formulate the pose-related part representation as:

\[
\Gamma_{\mu} = [p_1 \log(p_1/q_1), p_2 \log(p_2/q_2), ..., p_m \log(p_m/q_m)],
\]

where the subscript is the dimension index.

Then, we regress both rotations and translations from the pose-related representations. The Euler angle \( \Omega \in \mathbb{R}^3 \) is used to represent the rotation, which is compact and facilitates accurate regression. Afterwards, the relative poses \( T_X, T_Y \) w.r.t. the same “latent canonical shape”, which can be decoded as,

\[
\begin{align*}
\Omega_X, t_X &= h_\beta(T_X), R_X = T(\Omega_X) \\
\Omega_Y, t_Y &= h_\beta(T_Y), R_Y = T(\Omega_Y),
\end{align*}
\]

where \( h_\beta \) is the MLP parameterized by \( \beta \), \( T(\cdot) \) is the function that transforms Euler angle to rotation matrix. We obtain the final rigid transformation as \( R = R_Y R_X^T, t = t_Y - R t_X \).

B. Unsupervised Loss

Next, in our representation separation framework, we can transform the source and target point clouds into the canonical coordinate by \( T_X \) and \( T_Y \) respectively, i.e., \( X_c = R_X^T (X - t_X), Y_c = R_Y^T (Y - t_Y) \). Note that the latent canonical shapes \( X_c \) and \( Y_c \) should be consistent here. We use the Chamfer distance to measure the discrepancy between \( X_c \) and \( Y_c \). In this case, any ground truth is not involved. In other words, an unsupervised loss is constructed as,

\[
\text{Loss}(X_c, Y_c) = \frac{1}{N_X} \sum_{x \in X_c} \min_{y \in Y_c} \| x - y \|^2_2 + \frac{1}{N_Y} \sum_{y \in Y_c} \min_{x \in X_c} \| x - y \|^2_2.
\]

III. EXPERIMENTS

Implementation Details. We use GNN as the point cloud representation extractor, where \( K = 24 \) of KNN experimentally.

The dimension \( m = 512 \). The number of layers \( \ell = 5 \). We train our network for 500 epochs with an initial learning rate of \( 1 \times 10^{-3} \). As our method is unsupervised, we fine-tune the model for 200 epochs with an initial learning rate of \( 1 \times 10^{-4} \) on the test split. We use the Adam optimizer and the experiment is performed in PyTorch 1.0.0 with a batch size of 26.

Dataset and Evaluation Metric. For synthetic dataset ModelNet40 [26], following [23], [24], the rotation and translation along each axis between the source and target are in \([0^\circ, 45^\circ], [-0.5, 0.5] \). For real dataset 7Scenes [20], following [11], the rotation is initialized in \([0^\circ, 60^\circ]\) and the translation is initialized in \([0, 1.0]\) along one randomly selected axis. Root mean square error (RMSE) and mean absolute error (MAE) in Euler angle and translation vector, and mean error in \( SE(3) \) [3] are used as evaluation metrics, noted as RMSE(R), MAE(R), RMSE(t), MAE(t), and ME(T) respectively.

A. Evaluation on ModelNet40

Following [23], [24], three experiment settings are applied here: 1) Unseen Point Clouds (UPC): All point clouds are divided into training and test sets with the official setting. 2) Unseen Categories (UC): The first 20 categories in ModelNet40 are used for training and the rest for test. 3) Noisy Data (ND): Random Gaussian noise (i.e. \( \mathcal{N}(0, 0.01) \), and the sampled noise out of the range of \([-0.05, 0.05]\) will be clipped) is added to the UPC setting.

Consistent Point Clouds. Following [23], the input point clouds are consistent here. Two versions of PRNet have been evaluated, i.e., PRNet selects half points as the key points [24] and PRNet* selects all points as the key points. Besides using distance-based feature notated as UPCR w/ Dis, we also test our method using SPFH [17], PFH [18] as pose-invariant feature notated as UPCR w/ SPFH and UPCR w/ PFH respectively. From Table I, for UPC, in all unsupervised methods, UPCR achieves the best performance. Meanwhile, our unsupervised method even obtains better performance under many metrics than supervised methods. Especially, UPCR outperforms the correspondence-free method, PointNetLK by a large margin. For UC, in all unsupervised methods, UPCR significantly outperforms all baselines. Our method achieves higher accuracy under most metrics than those supervised methods. For ND, UPCR is superior to DCP-v2 but inferior to PRNet and PRNet* under RMSE(R) while achieving the best results under other metrics. Then, in all UPCR pipelines, UPCR w/ PFH achieves the most accurate results due to the informative pose-invariant feature.

| Methods | RMSE(R) | MAE(R) | RMSE(t) | MAE(t) | ME(T) |
|---------|---------|---------|---------|---------|--------|
| DCP-v2  | 1.774   | 4.473   | 0.150   | 0.366   | 0.014  |
| PRNet   | 1.774   | 4.473   | 0.150   | 0.366   | 0.014  |
| PRNet*  | 1.774   | 4.473   | 0.150   | 0.366   | 0.014  |
| UPCR w/ Dis | 2.793 | 4.595 | 0.212 | 0.436 | 0.078 |
| UPCR w/ SPFH | 2.793 | 4.595 | 0.212 | 0.436 | 0.078 |
| UPCR w/ PFH | 2.793 | 4.595 | 0.212 | 0.436 | 0.078 |

Partial-to-Partial. Our method is robust to outliers because the disturbance in the pose-invariant part has been filtered out. Following the common protocol of PRNet [24], we simulate...
partial scans of X and Y by randomly placing a point in 3D space and selecting its 768 nearest neighbors from the consistent point clouds, respectively, where there are 1024 points in each consistent point cloud. The performance is reported in Table II. For UPC, our UPCR achieves the best performance in all unsupervised methods. Besides, we obtain comparable or better results compared with the state-of-the-art PRNet. For UC, UPCR achieves the best performance under all metrics, even outperforms PRNet. For ND, our method achieves the best performance under most evaluation metrics over all unsupervised and supervised methods. These results verify the robustness of our method to partial-to-partial cases. Besides, UPCR w/ PFH achieves the best performance among all UPCR pipelines because PFH feature is more informative.

| Methods | Variables | RMSE(R) | MAE(R) | RMSE(t) (× 10−4) | MAE(t) (× 10−4) | ME(T) |
|---------|-----------|---------|--------|-------------------|-------------------|-------|
| PointNet [1] | 16.735 24.930 | 18.300 | 7.530 | 6.855 | 9.785 | 4.005 | 4.061 | 5.625 | 10.013 | 10.264 | 25.417 | 31.547 |
| DCP+PFH [2] | 3.189 3.986 | 4.225 | 1.574 | 2.315 | 2.991 | 0.923 | 0.923 | 0.015 | 0.015 | 3.428 | 10.760 | 4.583 |
| PRNet [3] | 3.181 3.984 | 4.213 | 1.574 | 2.315 | 2.991 | 0.923 | 0.923 | 0.015 | 0.015 | 3.428 | 10.760 | 4.583 |
| DCP [4] | 5.047 6.798 | 6.488 | 5.910 | 6.584 | 6.254 | 3.587 | 4.323 | 3.728 | 3.387 | 11.238 | 9.932 | 27.635 |
| PRNet* [5] | 4.941 6.680 | 6.415 | 5.985 | 6.570 | 6.224 | 3.470 | 4.323 | 3.728 | 3.387 | 11.238 | 9.932 | 27.635 |
| DCP+PFH* [6] | 4.927 6.680 | 6.415 | 5.985 | 6.570 | 6.224 | 3.470 | 4.323 | 3.728 | 3.387 | 11.238 | 9.932 | 27.635 |
| UPCR w/ PFH | 3.271 4.304 | 4.112 | 1.574 | 2.315 | 2.991 | 0.923 | 0.923 | 0.015 | 0.015 | 3.428 | 10.760 | 4.583 |

3. Completeness of Pose-Invariant Feature. We can observe that UPCR with more informative pose-invariant features, e.g., SPFH, PFH achieves better performance than using distance-based feature since SPFH and PFH represent the pose-invariant part more completely. Here, to verified this conclusion further, we employ seven pose-invariant features, i.e., distance-based feature, PFH [5], SPFH [17], PFH [18], distance-based + PPF, distance-based + SPFH and distance-based + PFH + SPFH to extract the pose-invariant representation variants, respectively. Then, the global registration, and the seven pose-related representations returned by “subtracting” these variants are used to perform an ablation experiment.

B. Evaluation on 7Scenes

In Table V, we evaluate our UPCR on 7Scenes [20], which is a remote scene RGB-D data including Heads, Office, Chess, Fire, Pumpkin, RedKitchen, Stairs scenes. We use the 3D point cloud information only. Since the real scene data is very challenging, state-of-the-art supervised point cloud registration methods, such as DCP-v2 and PRNet fail to estimate the transformation. For UPCR, we provide the model trained on ModelNet40, fine-tuned on Heads, fine-tuned on the corresponding test scene respectively. Results are consistent: 1) The model trained on ModelNet40 shows poor performance on scene data but is still better than baselines. 2) Models fine-tuned on Heads and corresponding test scenes show accurate and very similar performance, i.e., the model fine-tuned on one scene can obtain accurate results on other scenes.

C. Ablation Study

1. Rotation Solver. There are several rotation representation, including rotation matrix, 6D expression [31], quaternion, Euler angle, etc., which are compared in Table III. The Euler angle representation achieves the best performance. We reckon this is because the network can converge more easily to achieve better results in a more compact representation.

2. Performance vs. Outliers Ratios. Fig. 2 shows the performance degeneration as outliers ratio increases. Compared to PRNet, our unsupervised UPCR w/ Dis method has comparable rotation estimation and better translation estimation results. Totally speaking, when outliers ratio increases, our method is more robust than these supervised methods.

IV. Conclusion

In this paper, we have proposed a correspondence-free unsupervised 3D point cloud registration method from the representation separation perspective. Our proposed method not only filters out the disturbance in pose-invariant representation but also is robust to partial-to-partial or noisy point clouds. Experiments on benchmark datasets demonstrate the superiority and robustness of our unsupervised 3D point cloud registration method, which achieves comparable if not better performance than the state-of-the-art supervised methods.
| Methods | Heads | Office | Choco | Fire | Pumpkin | Rodinchenko | Stars |
|---------|-------|-------|-------|------|---------|-------------|-------|
| UPCR w/ heads | 0.185 | 0.123 | 0.176 | 0.105 | 0.192 | 0.136 | 0.201 | 0.134 | 0.193 | 0.113 | 0.085 | 0.070 | 0.090 | 0.070 | 0.086 | 0.065 | 0.081 | 0.058 | 0.085 | 0.070 |
| UPCR w/ PFH | 0.048 | 0.031 | 0.048 | 0.032 | 0.051 | 0.038 | 0.047 | 0.037 | 0.061 | 0.045 | 0.048 | 0.036 | 0.054 | 0.041 | 0.053 | 0.039 | 0.027 | 0.031 | 0.035 | 0.028 | 0.032 |
| UPCR w/ SPFH | 0.025 | 0.020 | 0.025 | 0.021 | 0.035 | 0.027 | 0.022 | 0.013 | 0.024 | 0.021 | 0.017 | 0.017 | 0.013 | 0.016 | 0.013 | 0.012 | 0.006 | 0.022 | 0.012 | 0.006 | 0.008 |
| UPCR w/ DCP-v2 | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 |
| UPCR w/ PFH | 0.023 | 0.015 | 0.023 | 0.017 | 0.027 | 0.018 | 0.017 | 0.011 | 0.021 | 0.015 | 0.017 | 0.011 | 0.019 | 0.012 | 0.010 | 0.010 | 0.005 | 0.016 | 0.010 | 0.010 | 0.010 |
| UPCR w/ PFH w/ heads | 0.022 | 0.016 | 0.022 | 0.018 | 0.027 | 0.022 | 0.017 | 0.010 | 0.018 | 0.017 | 0.013 | 0.010 | 0.018 |

**Table V: Evaluation on 7Scenes.** ModelNet40 indicates that the model is trained on ModelNet40, Heads means that the model is fine-tuned on Heads, test scene means that the model is fine-tuned on the corresponding test scene.

**References**

[1] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, “Pointnetlk: Robust & efficient point cloud registration using pointnet,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7163–7172.

[2] P. J. Besl and N. D. McKay, “A method for registration of 3-d shapes,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 14, no. 2, 1992.

[3] R. Brégier, F. Devernay, L. Leyrit, and J. L. Crowley, “Defining the pose of any 3d rigid object and an associated distance,” *International Journal of Robotics and Automation*, vol. 17, no. 4, pp. 721–725, 2020.

[4] W. Sun, A. Tagliasacchi, B. Deng, S. Sabour, S. Yazdani, G. Hinton, and K. M. Yi, “Canonical capsules: Unsupervised capsules in canonical pose,” *arXiv:2012.04718*, 2020.

[5] C. Zhao, H. Guo, J. Lu, D. Yu, D. Li, and X. Chen, “ALS point cloud matching with smoothed densities,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5108–5118.

[6] Y. Wang and J. M. Solomon, “Prnet: Self-supervised learning for rotation-invariant 3d local descriptors,” in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 620–630.

[7] Y. Wang and J. M. Solomon, “P-net: Self-supervised learning for partial-to-partial registration,” in *Proceedings of the Advances in Neural Information Processing Systems*, 2019, pp. 8814–8826.

[8] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. Bronstein, and J. M. Solomon, “Dynamic graph cnn for learning on point clouds,” *ACM Transactions on Graphics*, vol. 38, no. 5, pp. 1–12, 2019.

[9] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, “3d shapenets: A deep representation for volumetric shapes,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1912–1920.

[10] W. Y. Yan, A. Shaker, and N. El-Alshamy, “Urban land cover classification using airborne lidar data: A review,” *Remote Sens. Environ.*, vol. 158, pp. 295–310, 2015.

[11] Y. Zhang and J. M. Solomon, “Deep closest point: Learning representations for point cloud registration,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3573–3582.

[12] Y. Zhang and J. M. Solomon, “Pnet: Self-supervised learning for partial-to-partial registration,” in *Proceedings of the Advances in Neural Information Processing Systems*, 2019, pp. 8814–8826.

[13] W. Sun, A. Tagliasacchi, B. Deng, S. Sabour, S. Yazdani, G. Hinton, and K. M. Yi, “Canonical capsules: Unsupervised capsules in canonical pose,” *arXiv:2012.04718*, 2020.

[14] G. Wan, X. Yang, R. Cai, H. Li, Y. Zhou, H. Wang, and S. Song, “Robust and precise vehicle localization based on multi-sensor fusion in diverse scenes,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2018, pp. 4670–4677.

[15] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proceedings of the International Joint Conference on Artificial Intelligence*, 1981, pp. 674–679.

[16] H. Luo, Q. Zheng, C. Wang, and W. Guo, “Boundary-aware and semiautomatic segmentation of 3-d object in point clouds,” *IEEE Geosci. Remote. Sens. Lett.*, vol. 18, no. 5, pp. 910–914, 2021.

[17] C. Zhao, H. Guo, J. Lu, D. Yu, D. Li, and X. Chen, “ALS point cloud classification with small training data set based on transfer learning,” *IEEE Geosci. Remote. Sens. Lett.*, vol. 17, no. 8, pp. 1406–1410, 2020.

[18] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, “On the continuity of rotation representations in neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5745–5753.

[19] C. Zhao, H. Guo, J. Lu, D. Yu, D. Li, and X. Chen, “ALS point cloud classification with small training data set based on transfer learning,” *IEEE Geosci. Remote. Sens. Lett.*, vol. 17, no. 8, pp. 1406–1410, 2020.

[20] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, “On the continuity of rotation representations in neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5745–5753.