Correspondence-Free Point Cloud Registration with SO(3)-Equivariant Implicit Shape Representations

Anonymous Author(s)
Affiliation
Address
email

Abstract: This paper proposes a correspondence-free method for point cloud rotational registration. We learn an embedding for each point cloud in a feature space that preserves the SO(3)-equivariance property, enabled by recent developments in equivariant neural networks. The proposed shape registration method achieves three major advantages through combining equivariant feature learning with implicit shape models. First, the necessity of data association is removed because of the permutation-invariant property in network architectures similar to PointNet. Second, the registration in feature space can be solved in closed-form using Horn's method due to the SO(3)-equivariance property. Third, the registration is robust to noise in the point cloud because of the joint training of registration and implicit shape reconstruction. The experimental results show superior performance compared with existing correspondence-free deep registration methods.

Keywords: Point cloud registration, implicit shape model, equivariant neural network, representation learning

1 Introduction

Point clouds are a major form of 3D information representation in perception with numerous applications today, including intelligent robots and self-driving cars. The registration of two different point clouds capturing the same underlying object or scene refers to estimating the relative transformation that aligns them. Point cloud registration enables information aggregation across several observations, which is essential for many tasks such as mapping, localization, shape reconstruction, and object tracking. Given perfect point-to-point pairings between the point-cloud-pair, the relative transformation can be solved in closed-form using Horn's method [1].

However, such a pairing is not available or corrupted in real-world problems, posing a significant challenge for the point cloud registration task, namely data association. In the deep learning era, the strong feature learning capacity of neural networks enables better description and discrimination of points, improving their matching. Overall, these methods can be categorized as correspondence-based registration.

An alternative approach, called correspondence-free registration, is to completely circumvent the matching of points by learning a global representation for the entire point cloud and estimating the transformation by aligning their representations. Similar to correspondence-based registration, correspondence-free registration can also be with or without the involvement of deep learning. The main obstacle in this type of approach is that the mapping between the input Euclidean space and the output feature space realized by neural networks is nonlinear and obscure. Thus it is common to rely on iterative optimization with local linearization like Gauss-Newton [2]. The performance of these approaches deteriorates when the initial pose difference gets larger.

In this paper, we innovatively employ equivariant neural networks and implicit shape learning into correspondence-free registration by constructing a feature space with two major advantages. First, thanks to the equivariance property, the feature space preserves the same rotation operation as the Euclidean input space, allowing us to solve the feature-space registration in closed form using Horn's method. As a result, the accuracy is independent of the initial pose difference. Second, we improve

Submitted to the 5th Conference on Robot Learning (CoRL 2021). Do not distribute.

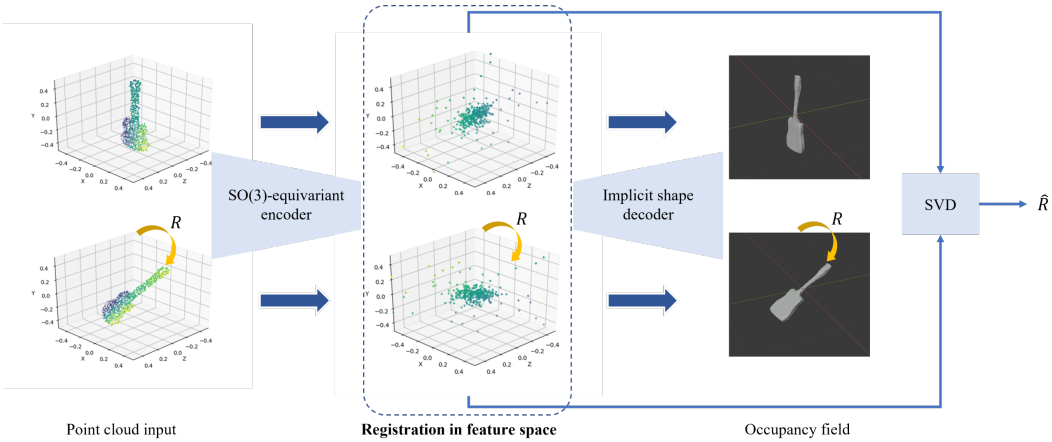

Figure 1: Overview of the correspondence-free rotational registration network. The point cloud input is of shape $\mathbb{R}^{N\times 3}$, and the encoded feature is of shape $\mathbb{R}^{C\times 3}$. N is the number of points, and C is the dimension of features. Occupancy field is a function $v(p) \mapsto [0,1], p \in \mathbb{R}^3$ mapping any 3D coordinate to an occupancy value (see Sec. 3.2 for details). The rotation is estimated by aligning the features using Horn's method [1].

the registration robustness against noise in point clouds by training the features for both the implicit shape reconstruction task and the registration task. The implicit shape learning encourages the network to learn features describing the underlying geometry rather than specific points sampled on the geometry, and the features are better prepared for registration by including the registration in the training loop. In this work, we focus on the rotational registration task.

In particular, this work has the following contributions.

1. We propose a correspondence-free point cloud registration method via $\mathrm{SO}(3)$-equivariance representation learning and closed-form pose estimation.
2. The proposed method achieves accurate registration results independent of initial pose differences with robustness against noisy points.
3. Open source software will be available (after the final decision).

2 Preliminaries on Equivariance and Vector Neurons

Symmetry (or equivariance) in a neural network is an important ingredient for efficient learning and generalization. It also allows us to build a clear connection between the changes in the input space and the output space. For example, convolutional neural networks are translation-equivariant. It makes a network generalize to image shifting and enables huge parameter saving. However, other symmetries are less explored. For example, most point cloud processing networks (e.g., PointNet [3], DGCNN [4]) are not equivariant to rotations.

We leverage the recently proposed Vector Neuron layers [5] to enable rotation-equivariant feature learning for point clouds. For the sake of completeness, we briefly introduce the preliminary on equivariance and the design of layers.

A function $f : X \to X$ is equivariant to a set of transformations G, if for any $g \in G$, f and g commutes, i.e., $g \cdot f(x) = f(g \cdot x), \forall x \in X$. For example, applying a translation on a 2D image and then going through a convolution layer is identical to processing the original image with a convolution layer and then shifting the output feature map. Therefore convolution layers are translation-equivariant.

Vector Neurons extend the equvariance property to $\mathrm{SO}(3)$ rotations for point cloud neural networks. The key idea is to augment the scalar feature in each feature dimension to a vector in $\mathbb{R}^3$ and redesign the linear, nonlinear, pooling, and normalization layers accordingly. In a regular network, the feature matrix with feature dimension C corresponding to a set of N points is $V \in \mathbb{R}^{N\times C}$. In Vector Neuron networks, the feature matrix is of form $V \in \mathbb{R}^{N\times C\times 3}$. The mapping between layers can be written as $f : \mathbb{R}^{N\times C_l\times 3} \to \mathbb{R}^{N\times C_{l+1}\times 3}$, where l is the layer index. Following this design, the representation of $\mathrm{SO}(3)$ rotations in feature space is straight-forward: $g(R) \cdot V := VR$, where $g(R)$ denotes the

rotation operation in the feature space, parameterized by the 3-by-3 rotation matrix $R \in \mathrm{SO}(3)$. In the following, we ignore the first dimension N of V for simplicity.

The linear layer for Vector Neurons is similar to a traditional MLP: $f_{\text{lin}}(V) = WV$, where $W \in \mathbb{R}^{C_{l+1} \times C_l}$. It is easy to see that such a mapping is $\mathrm{SO}(3)$-equivariant:

$$g(R) \cdot f_{\text{lin}}(x) = WVR = f_{\text{lin}}(g(R) \cdot V) \tag{1}$$

Designing nonlinearities for Vector Neuron networks is less trivial. A naive ReLu on all elements in the feature matrix will break the equivariance. Instead, a *vectorized* version of ReLu is designed as the truncation of vectors along a hyperplane. Specifically, given an input feature matrix V, predict a canonical direction $k = UV \in \mathbb{R}^{1 \times 3}$ where $U \in \mathbb{R}^{1 \times C}$. Then the ReLU is defined as:

$$v' = \begin{cases} v, & \text{if } \langle v, k \rangle \geq 0 \\ v - \langle v, \frac{k}{\|k\|} \rangle \frac{k}{\|k\|}, & \text{otherwise} \end{cases} \tag{2}$$

for each row $v \in \mathbb{R}^{1 \times 3}$ of V. This design preserves $\mathrm{SO}(3)$ equivariance because k and V will rotate together, and $\langle VR, kR \rangle = \langle V, k \rangle$, meaning that rotations do not change the activation. For the design of pooling layers and normalization layers of Vector Neurons, please refer to the original paper for details [5].

3 Methodology

We employ a $\mathrm{SO}(3)$-equivariant neural network to construct a feature space preserving the same rotation representations as to the input space, thus simplifying solving the rotation for feature alignment. Furthermore, deep implicit representation learning and end-to-end training of registration improves the robustness of the features against noise and allows registration between two different scans of the same geometry.

3.1 $\mathrm{SO}(3)$-equivariant point cloud global feature extractor

We choose PointNet as the feature extractor backbone. The permutation invariance property implies that two point clouds correspond to the same feature embeddings if they only differ in the permutation of points. We replace the layers in PointNet with the corresponding Vector Neuron version [5]. Denote the feature extraction network as f, a point cloud $P \in \mathbb{R}^{N \times 3}$, and its rotated and permuted copy $P' = MPR$, where $M \in \mathbb{R}^{N \times N}$ is a permutation matrix and $R \in \mathrm{SO}(3)$ is a 3-by-3 rotation matrix. Then we have

$$Q' = f(P') = f(MPR) = f(MP)R = f(P)R = QR \tag{3}$$

where $Q := f(P) \in \mathbb{R}^{C \times 3}$ is the feature of P extracted by the encoder. Equation (3) is guaranteed by the $\mathrm{SO}(3)$-equivariance and permutation-invariance properties, which is essential for our feature-space registration.

A Vector Neuron network is designed to take inputs of $\mathbb{R}^{N \times C_0 \times 3}$, while the point cloud input is $P \in \mathbb{R}^{N \times 3}$. Simply reshaping P to $\mathbb{R}^{N \times 1 \times 3}$ with $C_0 = 1$ does not work, because a linear layer with weight $W \in \mathbb{R}^{C_1 \times 1}$ will make the resulted feature $WP \in \mathbb{R}^{N \times C_1 \times 3}$ linearly dependent in the third dimension for each point. Following [5], we use one DGCNN edge-convolution [4] layer to initialize the $\mathbb{R}^{N \times C_0 \times 3}$ feature from raw $\mathbb{R}^{N \times 3}$ point cloud using the neighboring points. See [5] for more details. Furthermore, we explore replacing the k-nearest neighbor operation with random sampling of k points in a ball with certain radius in the edge convolution layer. The motivation is to alleviate the sensitivity of the feature against point cloud density (see Sec. 4.2.3). However, the evaluations in this paper are using k-NN by default unless specified.

3.2 Deep implicit representation learning

Being able to register two rotated copies of the same point cloud is not enough for practical use. In real application, the measurements have noise, and the individual points captured by different scans generally do not correspond to the same physical points.

Our solution is to repurpose the global feature (for registration ultimately) for implicit shape representation. Following the Occupancy Network [6], we build an encoder-decoder network. The encoder is the aforementioned $\mathrm{SO}(3)$-equivariant feature extractor f, taking a point cloud $P \in \mathbb{R}^{N \times 3}$

as input. The decoder takes the encoded shape feature $Q = f(P) \in \mathbb{R}^{C\times 3}$ and a queried position $p \in \mathbb{R}^3$ as input, predicting the occupancy value of that position $v(p; Q) \in [0, 1]$. Notice that p need not to be coming from P. In practice, we sample n querying points for each input to calculate the implicit shape reconstruction (occupancy value prediction) loss:

$$L_{\text{occ}} = \sum_{i=1}^{n} L_{\text{cross_entropy}}(v(p_i; f(P)), v_{\text{gt}}(p_i)), \tag{4}$$

where $v_{\text{gt}}(p) \in \{0, 1\}$ is the ground truth occupancy value of a query point p. The sampling strategy of the querying points is consistent with the Occupancy Network [6]. In this way, we implicitly encourage the feature $f(P_1), f(P_2), ...$ to be close as long as $P_1, P_2, ...$ are point clouds of the same geometry.

3.3 Point cloud registration through feature alignment

With the above network design, we can further relax the connection between the two point clouds P' and P in (3). The features Q and $Q'$are *approximately* rotation-equivariant even when P and P' are not permutations of each other, as long as they are sampled on the same geometry. We say it is approximate because a trained implicit shape reconstruction network can only represent the implicit shape approximately. Therefore, we also include the registration step in the training loop, directly encouraging the accurate registration of point clouds with noise. As shown below, the registration is of closed form and differentiable, allowing end-to-end training of better features.

One can see that (3) is exactly in the form of orthogonal Procrustes problem. The learned features Q and Q' can be regarded as two *pseudo* point clouds and each row of them are *automatically matched*. This problem has a standard closed-form solution. First calculate the cross covariance matrix $H = Q^{\mathsf{T}}Q'$. Then conduct Singular Value Decomposition (SVD) on H: $H = USV^{\mathsf{T}}$. The optimal rotation matrix $\hat{R}$ is given as:

$$\hat{R} = V\Lambda U^{\mathsf{T}} \tag{5}$$

where $\Lambda = \text{diag}(1, 1, \det(VU^{\mathsf{T}})) \in \mathbb{R}^{3\times 3}$.

We design the registration loss as the mean square error between the residual rotation matrix and the identity matrix:

$$L_{\text{reg}} = \left\| R_{\text{gt}}^{\mathsf{T}} R_{\text{est}} - I_3 \right\|_F, \tag{6}$$

where R_{est} is calculated from (5), and $\|\|_F$ denotes the Frobenius norm.

4 Experiments

4.1 Overview

Our network is trained on the ModelNet40 dataset [7]. It is composed of 12311 CAD models from 40 categories of objects. Following the official split, 9843 models are used for training, and 2468 models are used for testing. The models are preprocessed using the tool provided by Stutz and Geiger [8] to obtain watertight and simplified models that are centered and scaled to a unit cube. Given watertight models, the occupancy value of any 3D coordinate can be determined depending on whether it is inside or outside the model.

We also evaluate our method on a real-world dataset: 7Scenes [9], which is a subset of 3DMatch [10]. We choose it in order to be consistent and comparable with one of our baselines FMR [11]. 7Scenes consists of 7 sequences of RGBD scans of indoor environments. RGB information is not used in the experiments.

The training of the network is composed of two stages. The network is first trained for occupancy value prediction only, after which the network is trained for the occupancy prediction and registration tasks jointly. In the first stage, 1024 points are randomly sampled on each mesh model as the input point cloud. with batch size 24. In the second stage, a pair of point clouds are sampled on the same shape model with different pose and number of points. The registration loss between the pair is calculated, together with the occupancy loss of each of the point clouds. The occupancy value is queried at 2048 points in the unit cube. The batch size of such pairs is 10 in this stage.

Table 1: Rotational registration error given different random initial rotations.
Tested using ModelNet40 official test set. The *max initial rotation angle* refers to the maximum rotation angle allowed in the generation of initial random rotations during testing. The rotation axis is fully randomized. The last column (180 degrees) corresponds to unconstrained initial rotations. All values are in degrees.

Max initial rotation angle	0	30	60	90	120	150	180
Methods	Rotation error after registration						
PCR-Net	7.077	9.498	27.379	68.223	109.608	129.289	133.486
FMR	0.000	0.469	5.292	21.947	42.256	66.385	79.432
Ours	0.019	0.018	0.019	0.020	0.021	0.018	0.021

Table 2: Rotational registration error of noisy point clouds on ModelNet40.

Max initial rotation angle	0	30	60	90	120	150	180
Methods	Rotation error after registration						
PCR-Net	7.212	9.295	27.397	69.956	108.425	125.547	132.396
FMR	1.564	2.012	5.784	19.950	45.728	67.653	82.048
Ours	2.837	2.947	2.785	3.582	3.390	3.402	3.261

The dimension of the learned global feature is set as $C = 342$, corresponding to $342 \times 3 = 1026$ dimensions for a traditional feature vector, close to the dimension chosen by previous literature including PointNetLK [12], PCRNet [13], and Feature-Metric Registration (FMR) [11]. The network is trained on a single Tesla V100 GPU.

4.2 Evaluation on synthetic data: ModelNet40

4.2.1 Registration with rotated copies of point clouds

We first tested the registration of two points clouds that are only different by rotation and permutation. Both point clouds have 1000 points sampled from the mesh model. The initial rotations are generated randomly using axis-angle representation to specify the maximum magnitude. The error is also measured as the angle in axis-angle representations. We compare with two representative correspondence-free deep registration methods as baselines: PCR-Net [13] and FMR [11]. The experiments on the baseline methods are based on their official open-source implementation and pretrained weights. The results are shown in Table 1. We can see that both PCR-Net and FMR are sensitive to the initial rotation angle and only work properly when the initial rotation is small because they rely on iterative refinements. In comparison, the registration accuracy of our method is independent of the initial rotation angle, and it provides almost perfect rotational registration.

4.2.2 Registration with rotated copies of point clouds corrupted with Gaussian noise

In practice, the pair of point clouds for registration will not be identical to each other. Therefore we test the case where the source and target point clouds are both corrupted by noise. We add a Gaussian noise with $\sigma = 0.01$ to all points before putting them through the networks. The rest of the setup is the same as in Sec. 4.2.1. As shown in Table 2, PCR-Net and FMR show a similar trend of increasing error when the initial angle gets larger. Our method, though showing a slightly larger registration error, still outperforms the baseline in most columns. Most importantly, the error remains consistent across different initialization. The result shows that our method not only works in an ideal situation where the point clouds are noise-free but also delivers accurate rotation estimation under noise corruption.

4.2.3 Registration with point clouds with different densities

To further test the robustness of the proposed method against variations in the point clouds, we tested the registration performance when given two point clouds of different densities. Here we sample 1000 and 500 points for the pair of point clouds, respectively. The registration accuracy is shown in Table 3. The trend of the three methods is similar to what is shown in Sec. 4.2.1 and Sec. 4.2.2. We do observe an increase of error in our method in this experiment. A possible reason is that the DGCNN layer for feature initialization at the beginning of the encoder is sensitive to the point cloud density, since it takes the relative coordinate of neighboring points as inputs.

Therefore, we experimented replacing the k-nearest neighbor with random sampling in a ball centered at each point when collecting neighboring points for the edge-convolution of that point. In practice, we sample 20 neighboring points for both strategy, and the radius for ball sampling is 0.2.

Table 3: Rotational registration error of point clouds with different densities on ModelNet40.

Max initial rotation angle	0	30	60	90	120	150	180
Methods	Rotation error after registration						
PCR-Net	7.365	9.702	27.044	65.953	109.220	126.430	130.369
FMR	1.951	2.712	6.913	23.748	45.141	66.972	81.558
Ours	16.161	15.203	16.594	15.740	16.928	17.163	16.497

Table 4: Ablation study on the point sampling strategy in feature initialization layer on ModelNet40. Since our method is invariant to the initial rotation angle, in this table we only report results with random and unrestricted initial rotation angles (up to 180 degrees). The three columns of input conditions correspond to the input condition of Table 1, 2, and 3.

Condition of input point clouds	noise-free	with noise	of different density
Samping strategy	Rotation error after registration		
k-nearest neighbors	0.021	3.261	16.497
random sampling in a ball	3.352	4.837	13.231

The result is in Table 4. One can see that the ball sampling strategy is able to improve the registration performance for point clouds of different density. However, the error is slightly higher compared with k-NN when the point clouds are rotated copies or with Gaussian noise with the same number of points. It is understandable since the ball sampling introduces randomness in the feature initialization, and could slightly worsen the registration when the inputs are originally ideal.

With either of the strategies, under the condition of density difference, the registration error is slightly larger than that of the baselines when the initial rotation angle is small, but is much smaller when the initial rotation is arbitrary. A possible way for further improvement is to use our method as initialization, and then run an iterative-optimization-based method such as FMR or PointNetLK for refinement. It is not closely related to our key innovation, therefore not further discussed in this paper.

4.3 Evaluation on real-world data: 7Scenes

The model evaluated on 7Scenes is first trained on ModelNet40 with the two stages mentioned in Sec. 4.1, and then finetuned on 7Scenes with the registration loss only. The train/test split is consistent with [11].

4.3.1 Registration with point clouds that are sampled differently

The experiments on 7Scenes consists of two parts. In the first part, the pair of point clouds are from the same frame, but sampled differently. The original dense RGBD scan has more than 100,000 points. 1000 points are sampled randomly to form an input point cloud. The pair of point clouds then go through random rotation to form the final input. In this case, the points in one point cloud is neither a noisy version nor a subset of the other point cloud. To our knowledge, we are one of the first to report registration results under this setting in correspondence-free methods [12, 11, 13, 14]. Table 5 shows that our method is working properly on real world scan of point clouds that are different samplings of the underlying geometry.

4.3.2 Registration with point clouds that are partially overlapping

In the second part, we report registration results of point clouds coming from adjacent frames in the sequence. Because of camera movement, the pair of point clouds only overlaps partially. Since our method only estimates rotations, we first align the adjacent frames using ground truth pose, then randomly rotate two point clouds separately in a common frame, so that the input point clouds are related by a rotation. To our knowledge, we are also one of the first to show quantitative results on partially overlapping point clouds in correspondence-free methods [12, 11, 13, 14].

The results are in Table 6. Both our method and the baseline deteriorate in this setting. For FMR, the system completely fails when taking partially-overlapping inputs. Our method consistently gives better rotation estimation under all initial conditions. Still, the accuracy has a lot of room for improvement, indicating that there is still a long way to go for correspondence-free registration methods. For example, recent work by Huang et al. [15] predicting the overlapping part of point cloud pairs may inspire correspondence-free methods to better handle partial overlapping inputs by predicting features focused on the overlapping parts.

Table 5: Rotational registration error of point clouds sampled differently on 7Scenes.

Max initial rotation angle	0	30	60	90	120	150	180
Methods	Rotation error after registration						
FMR	1.591	1.605	6.614	19.829	39.496	68.463	84.597
Ours	4.701	4.319	4.932	6.070	5.474	5.189	5.388

Table 6: Rotational registration error of partially overlapping point clouds on 7Scenes.

Max initial rotation angle	0	30	60	90	120	150	180
Methods	Rotation error after registration						
FMR	38.806	42.424	48.588	62.433	80.903	80.279	101.595
Ours	25.122	24.039	23.671	24.245	25.309	24.318	24.487

Overall, our proposed rotational registration method can estimate consistent results independent of initial rotations. It outperforms the correspondence-free baselines in all evaluation settings when initial rotational angle is arbitrary. Experiments on both synthetic data and real-world data are conducted, covering conditions with noisy inputs, varying density, different sampling, and partial overlapping. Although outperforming the baseline, the partial overlapping case is not perfectly handled by our method yet, and we briefly discussed possible future directions.

5 Related Work and Discussion

5.1 Correspondence-based point cloud registration

A major challenge here is to recover the corresponding pairs of points from a pair of point clouds. ICP simply matches the closest points together, solves the transformation, and iteratively rematches the closest points after aligning the two point clouds using the estimated transformation [16]. Point-to-line [17], point-to-plane [18], plane-to-plane [19], and Generalized-ICP [20] build local geometric structures to the loss formulation.

Finding correspondences require strong feature descriptor for points, on which deep learning approaches show expertise. Through metric learning, a feature descriptor can be learned such that matching points are close to each other in the feature space, while non-matching points are far away. Approaches following this idea include PPFNet [21], 3DSmoothNet [22], SpinNet [23], and FCGF [24]. Good point matching leads to accurate pose estimation when solving the orthogonal Procrustes problem. Therefore, the error in pose estimation can be used to supervise point matching and feature learning. DCP [25], RPM-Net [26], DGR [27], and 3DRegNet [28] are some of the methods leveraging ground truth pose to supervise point feature learning. Neural networks are designed to better pick the keypoints (e.g., USIP [29] and SampleNet [30]). Keypoint selection and feature learning may also be considered jointly. Representative works include 3DFeat-Net [31], D3Feat [32], DeepICP [33], and PRNet [34].

The main remaining challenge is that such matching pairs may not exist in the input point clouds in the first place, because point clouds are sparse and a point may not be captured repeatedly by different scans. Soft matching (or many to many) [35, 36] is proposed to address this problem, but it is at best an approximation of the underlying true matching using sparse samples, which can deteriorate the performance when the sparsity increases.

5.2 Correspondence-free point cloud registration

Correspondence-free registration treats a point cloud as a whole rather than a collection of points, requiring a global representation for an entire point cloud. In this way, the limitation brought by the sparsity as mentioned in Sec. 5.1 is circumvented. CVO [37, 38] represents a point cloud as a function in a reproducing kernel Hilbert space, transforming the registration problem to maximizing the inner product of two functions. A series of deep-learning-based methods attempt to extract a global feature embedding for a point cloud and solve the registration by aligning the global features. Examples include PointNetLK [12], PCRNet [13], and Feature-Metric Registration [11]. A limitation of global-feature-based registration is that the nonlinearity of the feature extraction networks leaves us few clues about the structure and properties of the feature space. Therefore these methods rely on iterative local optimization such as the Gauss-Newton algorithm. Consequently, these methods require good initialization to achieve decent results. Our method differentiates from previous work

in that the feature extraction network is $\mathrm{SO}(3)$ equivariant, enabling well-behaved optimization in the feature space (detailed in Sec. 3).

5.3 Group equivariant neural networks

Neural networks today mainly preserve symmetry against translation, limiting the capacity of networks to deal with broader transformations presented in data. One line of work forms kernels as some steerable functions so that the rotations in the input space can be transformed into rotations in the output space [39, 40, 41]. However, this strategy can limit the expressiveness of the feature extractor since the form of the kernels is constrained.

Another strategy is to *lift* the input space to a higher-dimensional space where the group is contained so that equivariance is obtained naturally [42, 43]. However, for each lifted input to K group elements, we need to integrate over the entire group for computing its convolution. While being mathematically sound, this approach involves the use of group $\exp$ and $\log$ maps (for mapping from and to the Lie algebra) and a Monte Carlo approximation to compute the convolution integral over a fixed set of group elements [43, Sections 4.2-4.4].

Cohen and Welling [44] studied the problem of learning the irreducible representations of commutative Lie groups. Kondor and Trivedi [42] studied the generalization of equivariance and convolution in neural networks to the action of compact groups. The latter also discussed group convolutions and their connection with Fourier analysis [45, 46]. In this context, the convolution theorem naturally extends to the group-valued functions and the representation theory [47].

A new design of $\mathrm{SO}(3)$ equivariant neural networks is proposed by Deng et al. [5]. This approach allows incorporating the $\mathrm{SO}(3)$ equivariance property into existing network architectures such as PointNet and DGCNN by replacing the linear, nonlinear, pooling, and normalization layers with their *vector neuron* version. Our equivariant feature learner builds on this simple and useful idea.

5.4 Deep implicit shape modeling

Implicit shape modeling represents a shape as a field on which a function such as an occupancy or signed distance function is defined. The surface is typically defined by the equation $f(x, y, z) = c$ for some constant c. In recent years, deep models are developed to model such implicit function fields, where a network is trained to predict the function value given a query input point coordinate. Some representative works include DeepSDF [48], OccNet [6], NeRF [49].

Implicit functions model a shape continuously, up to the infinite resolution, which offers opportunities in registration to circumvent the challenge in data association due to sparsity in point clouds. DI-Fusion [50] leverages deep implicit models to do surface registration and mapping by optimizing the position of the query points to minimize the absolute SDF value. iNeRF [51] performs RGB-only registration through deep implicit models by minimizing the pixel residual between observed images and neural-rendered images. In this work, we also exploit the continuous nature of the deep implicit model but by using the learned latent feature instead of the decoded function field.

6 Conclusion

In this paper, we presented a correspondence-free rotational registration method for point clouds. The method is built upon the developments in equivariant neural networks and implicit shape representation learning. We construct a feature space where the rotational relations in Euclidean space are preserved, in which registration can be done efficiently. We circumvent the need for data association and solve the rotation in closed form, achieving low errors independent of initial rotation angles. Furthermore, we leverage implicit representation learning and end-to-end registration to enhance the network robustness against non-ideal cases where the points are not exactly corresponding.

We conducted experiments on synthetic and real-world data, showing the network's ability to handle various input imperfections. Nevertheless, the robustness of our method against partial overlapping inputs still has room for improvement. Furthermore, there are two open issues unsolved. First, generalizing the $\mathrm{SO}(3)$-equivariant neural network to $\mathrm{SE}(3)$ so that the registration of rotation and translation can be solved together. Secondly, generalizing the registration to handle outdoor LiDAR scans. These problems are left for future studies.

References

[1] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *J. Opt. Soc. Am. A*, 5(7):1127–1135, Jul 1988. doi:10.1364/JOSAA.5.001127. URL http://josaa.osa.org/abstract.cfm?URI=josaa-5-7-1127.

[2] R. W. Wedderburn. Quasi-likelihood functions, generalized linear models, and the gauss—newton method. *Biometrika*, 61(3):439–447, 1974.

[3] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016.

[4] 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 Transactions on Graphics (TOG)*, 38(5):1–12, 2019.

[5] C. Deng, O. Litany, Y. Duan, A. Poulenard, A. Tagliasacchi, and L. Guibas. Vector neurons: A general framework for SO(3)-equivariant networks. *arXiv preprint arXiv:2104.12229*, 2021.

[6] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.

[7] 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*, pages 1912–1920, 2015.

[8] D. Stutz and A. Geiger. Learning 3d shape completion under weak supervision. *CoRR*, abs/1805.07290, 2018. URL http://arxiv.org/abs/1805.07290.

[9] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proc. Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2013. URL https://www.microsoft.com/en-us/research/publication/scene-coordinate-regression-forests-for-camera-relocalization-in-rgb-d-images/.

[10] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, 2017.

[11] X. Huang, G. Mei, and J. 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.

[12] 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*, June 2019.

[13] V. Sarode, X. Li, H. Goforth, Y. Aoki, R. A. Srivatsan, S. Lucey, and H. Choset. Pcrnet: Point cloud registration network using pointnet encoding. *ArXiv*, abs/1908.07906, 2019.

[14] W. Yuan, B. Eckart, K. Kim, V. Jampani, D. Fox, and J. Kautz. Deepgmr: Learning latent gaussian mixture models for registration. In *European Conference on Computer Vision*, pages 733–750. Springer, 2020.

[15] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, and K. Schindler. Predator: Registration of 3d point clouds with low overlap. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4267–4276, 2021.

[16] 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, Feb 1992. ISSN 1939-3539. doi:10.1109/34.121791.

[17] A. Censi. An ICP variant using a point-to-line metric. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 19–25. IEEE, 2008.

[18] Y. Chen and G. G. Medioni. Object modeling by registration of multiple range images. *Image Vision Comput.*, 10(3):145–155, 1992.

[19] N. J. Mitra, N. Gelfand, H. Pottmann, and L. Guibas. Registration of point cloud data from a geometric optimization perspective. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '04, page 22–31, New York, NY, USA, 2004. Association for Computing Machinery. doi:10.1145/1057432.1057435.

[20] A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA, 2009.

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

[22] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser. The perfect match: 3d point cloud matching with smoothed densities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5545–5554, 2019.

[23] S. Ao, Q. Hu, B. Yang, A. Markham, and Y. Guo. Spinnet: Learning a general surface descriptor for 3d point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[24] C. Choy, J. Park, and V. Koltun. Fully convolutional geometric features. In *ICCV*, 2019.

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

[26] Z. J. Yew and G. H. Lee. Rpm-net: Robust point matching using learned features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11824–11833, 2020.

[27] C. Choy, W. Dong, and V. Koltun. Deep global registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[28] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7193–7203, 2020.

[29] J. Li and G. H. Lee. Usip: Unsupervised stable interest point detection from 3d point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 361–370, 2019.

[30] I. Lang, A. Manor, and S. Avidan. SampleNet: Differentiable point cloud sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7578–7588, 2020.

[31] Z. J. Yew and G. H. Lee. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In *ECCV*, 2018.

[32] X. Bai, Z. Luo, L. Zhou, H. Fu, L. Quan, and C.-L. 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.

[33] W. Lu, G. Wan, Y. Zhou, X. Fu, P. Yuan, and S. 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.

[34] Y. Wang and J. M. Solomon. Prnet: Self-supervised learning for partial-to-partial registration. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8814–8826. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/9085-prnet-self-supervised-learning-for-partial-to-partial-registration.pdf.

[35] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness. New algorithms for 2D and 3D point matching: Pose estimation and correspondence. *Pattern recognition*, 31(8):1019–1031, 1998.

[36] S. Granger and X. Pennec. Multi-scale EM-ICP: A fast and robust approach for surface registration. In *Proceedings of the European Conference on Computer Vision*, pages 418–432, 2002.

[37] M. Ghaffari, W. Clark, A. Bloch, R. M. Eustice, and J. W. Grizzle. Continuous direct sparse visual odometry from RGB-D images. In *Proceedings of the Robotics: Science and Systems Conference*, Freiburg, Germany, June 2019.

[38] R. Zhang, T.-Y. Lin, C. E. Lin, S. A. Parkison, W. Clark, J. W. Grizzle, R. M. Eustice, and M. Ghaffari. A new framework for registration of semantic point clouds from stereo and RGB-D cameras. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1–8, 2021.

[39] T. S. Cohen and M. Welling. Steerable cnns. *arXiv preprint arXiv:1612.08498*, 2016.

[40] C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis. Learning so (3) equivariant representations with spherical cnns. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–68, 2018.

[41] F. B. Fuchs, D. E. Worrall, V. Fischer, and M. Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *arXiv preprint arXiv:2006.10503*, 2020.

[42] R. Kondor and S. Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *International Conference on Machine Learning*, pages 2747–2755. PMLR, 2018.

[43] M. Finzi, S. Stanton, P. Izmailov, and A. G. Wilson. Generalizing convolutional neural networks for equivariance to Lie groups on arbitrary continuous data. In *International Conference on Machine Learning*, pages 3165–3176. PMLR, 2020.

[44] T. Cohen and M. Welling. Learning the irreducible representations of commutative lie groups. In *International Conference on Machine Learning*, pages 1755–1763. PMLR, 2014.

[45] G. S. Chirikjian and A. B. Kyatkin. *Engineering applications of noncommutative harmonic analysis: with emphasis on rotation and motion groups*. CRC Press, 2001. doi:10.1201/9781420041767.

[46] G. S. Chirikjian. *Stochastic models, information theory, and Lie groups, volume 2: Analytic methods and modern applications*, volume 2. Springer Science & Business Media, 2009.

[47] B. Hall. *Lie groups, Lie algebras, and representations: an elementary introduction*, volume 222. Springer, 2015.

[48] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[49] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020.

[50] J. Huang, S.-S. Huang, H. Song, and S.-M. Hu. Di-fusion: Online implicit 3d reconstruction with deep priors. *arXiv preprint arXiv:2012.05551*, 2020.

[51] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin. inerf: Inverting neural radiance fields for pose estimation. *arXiv preprint arXiv:2012.05877*, 2020.