SpinNet: Learning a General Surface Descriptor for 3D Point Cloud Registration

Sheng Ao[1*], Qingyong Hu[2*], Bo Yang[3], Andrew Markham[2], Yulan Guo[1,4]
[1]Sun Yat-sen University, [2]University of Oxford,
[3]Hong Kong Polytechnic University, [4]National University of Defense Technology

Abstract

Extracting robust and general 3D local features is key to downstream tasks such as point cloud registration and reconstruction. Existing learning-based local descriptors are either sensitive to rotation transformations, or rely on classical handcrafted features which are neither general nor representative. In this paper, we introduce a new, yet conceptually simple, neural architecture, termed SpinNet, to extract local features which are rotationally invariant whilst sufficiently informative to enable accurate registration. A Spatial Point Transformer is first introduced to map the input local surface into a carefully designed cylindrical space, enabling end-to-end optimization with SO(2) equivariant representation. A Neural Feature Extractor which leverages the powerful point-based and 3D cylindrical convolutional neural layers is then utilized to derive a compact and representative descriptor for matching. Extensive experiments on both indoor and outdoor datasets demonstrate that SpinNet outperforms existing state-of-the-art techniques by a large margin. More critically, it has the best generalization ability across unseen scenarios with different sensor modalities. The code is available at `https://github.com/QingyongHu/SpinNet`.

1. Introduction

Accurate matching of partial 3D surfaces is critical for point cloud registration [17, 38, 6, 21, 29], segmentation [58, 28, 27], and recognition [24, 12, 47]. Given multiple partially overlapped 3D scans, the goal of surface matching is to align these fragments according to a set of point correspondences, thus obtaining a complete 3D scene structure. To achieve this, it is of key importance to identify general and robust local geometric patterns shared between two scans. However, this is challenging, primarily because 1) different scans usually have different viewing angles, 2) the raw 3D scans are typically incomplete, noisy, and have significantly different point densities.

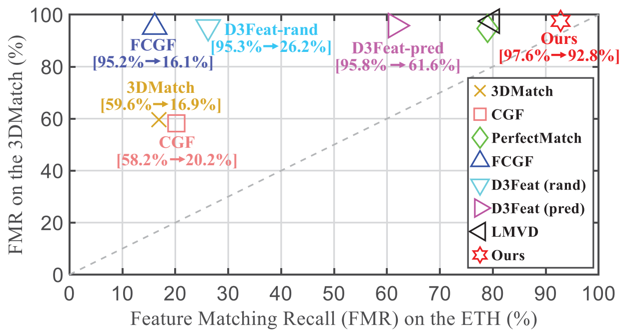

Figure 1: The Feature Matching Recall (FMR) scores of different approaches on the indoor 3DMatch [65] and outdoor ETH [46] dataset. Note that, all methods are trained only on the 3DMatch dataset. Our method not only achieves the highest score on 3DMatch, but also has the best generalization ability across the unseen ETH dataset.

Early methods to extract local geometries include PS [9], 3DSC [19], ISS [67] and SHOT [40], which simply compute the low-level features such as faces [41, 64], corners [52], and handcrafted statistical histograms [43]. Although they achieve encouraging results on high-quality 3D point clouds, they are not capable of generalizing to highly noisy and large-scale real-world 3D point clouds.

Recent learning-based approaches [65, 32, 14, 45] have yielded excellent results in extracting better local point features, thanks to the availability of large-scale labeled 3D datasets. However, they have two major limitations. **First**, many of these methods such as D3Feat [2] and FCGF [8] rely on kernel-based point convolution [53] or submanifold sparse convolution [23] to extract per-point features, resulting in the learned point local patterns being rotationally variant. Consequently, their performance drops dramatically when across datasets with significantly different rotation distributions. **Second**, although a number of recent approaches [14, 13, 22, 37] introduce rotation-invariant descriptors, they simply integrate the handcrafted features such as point-pairs [48, 16] and point density [50, 62, 51], or external local reference frames (LRFs) [37, 33, 66] into

*Equal contribution

the pipeline, fundamentally limiting the representational power of the framework [25]. As a result, the extracted point features, albeit being rotation invariant, are not robust and general when being applied to unseen 3D scans with noise and different point densities.

In this paper, we aim to design a new neural architecture, which is able to learn descriptive local features and generalize well to unseen scenarios. This network clearly satisfies three key properties: 1) It is rotation invariant. Particularly, it learns consistent local features from 3D scans with different rotation angles; 2) It is descriptive. In essence, it preserves the prominent local patterns despite the noise, possible surface incompleteness, or different point densities; 3) It does not include any handcrafted features. Instead, it only consists of multiple point transformations and simple neural layers coupled with true end-to-end optimization. This allows the learned descriptor to be extremely representative and general for complex real-world 3D surfaces.

Our network, named SpinNet, mainly consists of two modules, 1) a **Spatial Point Transformer**[1], which explicitly transforms the input 3D scans into a carefully designed cylindrical space, driving the transformed scans to be SO(2) equivariant, whilst retaining point local information; 2) a **Neural Feature Extractor**, which leverages powerful point-based and convolutional neural layers to learn representative and general local patterns.

The **Spatial Point Transformer** firstly aligns the input 3D surface by a reference axis, eliminating the rotational variance along the Z-axis. This is followed by a coordinate transformation over the XY-plane with the aid of spherical voxelization, further removing the rotation variance of each spherical voxel. Lastly, the transformed local surface is formulated as a simple yet novel 3D cylindrical volume, which is amenable to consumption by the subsequent point-based and convolutional neural layers. The **Neural Feature Extractor** firstly uses simple point-based MLPs to extract a unique signature for each voxel within the cylindrical volume, generating an initial set of cylindrical feature maps. These maps are further fed into a series of novel 3D cylindrical convolutional layers, which fully exploit the rich spatial and contextual information and generate a compact and representative feature vector for the input 3D surface.

These two modules enable our SpinNet to learn remarkably robust and general local features for accurate 3D point cloud registration. It achieves state-of-the-art performance both on the indoor 3DMatch [65] dataset and the outdoor ETH [46] dataset. Notably, it shows superior generalization ability across unseen scenarios. As shown in Figure 1, being trained only on the 3DMatch dataset, the learned descriptor of our SpinNet can achieve an average recall score of 92.8% on the unseen outdoor ETH dataset for feature matching, significantly surpassing the state of the art by nearly 13%.

[1]This is different from the Transformer for natural language processing.

Overall, our contributions are three-fold:

- We propose a new neural feature learner for 3D surface matching. It is rotation invariant, representative, and has superior generalization ability across unseen scenarios.
- By formulating the transformed 3D surface into a cylindrical volume, we introduce a powerful 3D cylindrical convolution to learn rich and general features.
- We conduct extensive experiments and ablation studies, demonstrating the remarkable generalization of our method and providing the intuition behind our choices.

2. Related Work

2.1. Handcrafted Descriptors

Traditional handcrafted descriptors can be roughly divided into two categories: 1) LRF-free methods and 2) LRF-based. The LRF-free descriptors including Spin Images (SIs) [30], Local Surface Patch (LSP) [4] and Fast Point Feature Histograms (FPFHs) [48] are typically constructed by exploiting geometrical properties (*e.g.* curvatures and normal deviations) of a local surface. The main drawback of these descriptors is the lack of sufficient geometric details for the local surface. The LRF-based descriptors such as Point Signature (PS) [9], SHOT [54] and Rotational Projection Statistics (RoPS) [26], albeit being able to exploit more spatial attributes than LRF-free descriptors, inherently introduce lager rotation errors, sacrificing the feature robustness. Overall, all these handcrafted descriptors are usually tailored to specific tasks and sensitive to noise, thus not being sufficiently flexible and descriptive for complicated and novel scenarios.

2.2. Learning-based Descriptors

In contrast to traditional handcrafted descriptors, recent works [10, 68, 3, 15, 63, 61] leverage data-driven deep neural networks to learn local features from large-scale datasets. These learned descriptors tend to have strong descriptive ability and robustness.

Rotation Variant Descriptors. Zeng *et al.* propose the pioneering work 3DMatch [65], which takes the local volumetric patches as input, and then leverages 3D Convolutional Neural Networks (CNNs) to learn local geometric patterns. Yew and Lee introduce a weakly-supervised framework 3DFeat-Net [60] to learn both the 3D feature detector and descriptor simultaneously. Choy *et al.* build a dense feature descriptor FCGF [8] based on [7]. Recently, Bai *et al.* [2] design a pipeline to jointly learn both dense feature detectors and local feature descriptors, achieving the state-of-the-art performance on 3DMatch [65] and KITTI [20] datasets for point cloud registration. However, all these methods are sensitive to rigid transformation in Euclidian space. Extensive data augmentation can be used to alleviate this problem, however, the overall performance of subsequent tasks is still sub-optimal [18].

Rotation Invariant Descriptors. A number of recent methods have started to learn rotation-invariant descriptors. Khoury *et al.* [32] parameterize the raw point clouds with oriented spherical histograms, and then map the high-dimensional embedding to a compact descriptor through a deep neural network. Deng *et al.* [14] encode the local surface using rotation-invariant Point Pair Features (PPFs). These features are then fed into multiple MLPs to learn a global descriptor. In the follow-up work [13], FoldingNet [59] is adopted as the backbone network to learn 3D local descriptors. Gojcic *et al.* [22] introduce the voxelized Smoothed Density Value (SDV) to encode the local surface as a compact and rotation-invariant representation, which is fed into a Siamese architecture to learn the final descriptor. Overall, although these methods are indeed able to learn rotationally invariant features from the local surface, they initially rely on classical handcrafted features which significantly limits the descriptiveness, robustness and generalization ability of the descriptors.

A handful of recent works [50, 62, 33, 37] try to learn rotation-invariant local descriptors with end-to-end optimization. However, they either require the computation of the point density or rely on external LRFs to achieve rotation invariance. This is usually unstable and does not generalize well to unseen datasets. In contrast, our SpinNet learns rotation-invariant, and highly descriptive local features in a truly end-to-end fashion, without relying on any handcrafted features or unstable LRFs. This enables the learned descriptors to be well generalized to novel scenarios.

3. SpinNet

3.1. Problem Statement

Given two partially overlapped point clouds $\mathcal{P} = \{p_i \in \mathbb{R}^3 | i = 1, \dots, N\}$ and $\mathcal{Q} = \{q_j \in \mathbb{R}^3 | j = 1, \dots, M\}$. The task of point cloud registration is to find an optimal rigid transformation $\mathbf{T} = \{\mathbf{R}, \mathbf{t}\}$, as well as the point correspondences to align pairs of fragments, and finally recover the complete scene. The pair of point correspondence (p_i, q_j) is expected to satisfy:

$$q_j = \mathbf{R}p_i + \mathbf{t} + \epsilon_i, \tag{1}$$

where $\mathbf{R} \in \mathrm{SO(3)}$ denotes the rotation matrix, $\mathbf{t} \in \mathbb{R}^3$ is the translation vector, and ϵ_i is the residual error. In practice, it is infeasible to simultaneously find the correspondences and estimate the transformation, due to the non-convexity of this problem [36]. However, if the point subsets $\mathcal{P}^c$ and $\mathcal{Q}^c$ with one-to-one correspondences can be determined, the registration problem can be simplified as a minimization problem for the following L_2 distance:

$$\mathcal{L}(\mathcal{P}^c, \mathcal{Q}^c | \mathfrak{P}, \mathbf{R}, \mathbf{t}) = \frac{1}{\mathcal{N}} \left\| \mathcal{Q}^c - \mathbf{R}\mathcal{P}^c\mathfrak{P} - \mathbf{t} \right\|^2 \tag{2}$$

where $\mathcal{N}$ is the number of successfully matched correspondences, $\mathfrak{P} \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$ is a permutation matrix whose entries satisfy $\mathfrak{P}_{u,v} = 1$ if the u^{th} point in $\mathcal{P}^c$ corresponds to v^{th} point in $\mathcal{Q}^c$ and 0 otherwise.

We propose a new surface feature learner SpinNet, which is a mapping function $\mathcal{M}$, where $\mathcal{M}(p_i)$ is equal to $\mathcal{M}(q_j)$ under arbitrary rigid transformations such as rotation and translation, if p_i and q_j are indeed a correct match. In particular, our feature learner mainly consists of a Spatial Point Transformer and a Neural Feature Extractor.

3.2. Spatial Point Transformer

This module is designed to spatially transform the input 3D surfaces into a cylindrical volume, overcoming the rotation variance, whilst without dropping critical information of local patterns. As shown in Figure 2, it consists of four components, as discussed below.

Alignment with a Reference Axis. Given a specific point $\mathbf{p} \in \mathcal{P}$ in a local surface, we first estimate a reference axis $\mathbf{n_p}$ oriented to the viewpoint [40, 1] from its neighbouring point set $\mathbf{P}^{\mathrm{s}} = \{\mathbf{p}_i : \|\mathbf{p}_i - \mathbf{p}\|^2 \leq R\}$ within a support radius R. We then align $\mathbf{n_p}$ with the Z-axis using a rotation matrix $\mathbf{R_z}$. Compared with the external local reference frames which are likely to be ambiguous and unstable, our estimated $\mathbf{n_p}$ tends to be more robust and stable with regard to rotation changes [44]. Subsequently, the neighbouring point set $\mathbf{P}^{\mathrm{s}}$ is transformed to $\mathbf{P}^{\mathrm{s}}_{\mathrm{r}} = \mathbf{R_z}\mathbf{P}^{\mathrm{s}}$. To achieve translation invariance, we further normalize $\mathbf{P}^{\mathrm{s}}_{\mathrm{r}}$ by offsetting to the center point, *i.e.*, $\widehat{\mathbf{P}}^{\mathrm{s}}_{\mathrm{r}} = \mathbf{P}^{\mathrm{s}}_{\mathrm{r}} - \mathbf{R_z}\mathbf{p}$. Hence, the obtained local patch $\widehat{\mathbf{P}}^{\mathrm{s}}_{\mathrm{r}}$ is aligned with the $\mathbf{z}$-axis, leaving the remaining rotational degree of freedom entirely on the XY-plane.

Spherical Voxelization. To further eliminate the rotational variance on the XY-plane, we leverage a rotation-robust spherical representation. In particular, we treat the patch $\widehat{\mathbf{P}}^{\mathrm{s}}_{\mathrm{r}}$ as a sphere, and evenly divide it into $J \times K \times L$ voxels along the radial distance ρ, elevation angle ϕ and azimuth angle θ. The center of each voxel is denoted as $\mathbf{v}_{jkl}$, where $j \in \{1, ..., J\}$, $k \in \{1, ..., K\}$, $l \in \{1, ..., L\}$. We then explicitly identify a set of neighboring points for the center $\mathbf{v}_{jkl}$ of each voxel. In particular, we use the radius query to find the neighboring points $\mathbf{P}_{jkl} \subset \widehat{\mathbf{P}}^{\mathrm{s}}_{\mathrm{r}}$ based on a fixed radius R_v, where $\mathbf{P}_{jkl} = \{\widehat{\mathbf{p}}_i : \|\widehat{\mathbf{p}}_i - \mathbf{v}_{jkl}\|^2 \leq R_v, \widehat{\mathbf{p}}_i \in \widehat{\mathbf{P}}^{\mathrm{s}}_{\mathrm{r}}\}$. Lastly, we randomly sample and preserve a fixed number of k_v points for each voxel, aiming for efficient computation in parallel. This spherical voxelization step is key to the successive spatial point transformation.

Transformation on the XY-Plane. To enable each spherical voxel to be rotationally invariant on the XY-plane,

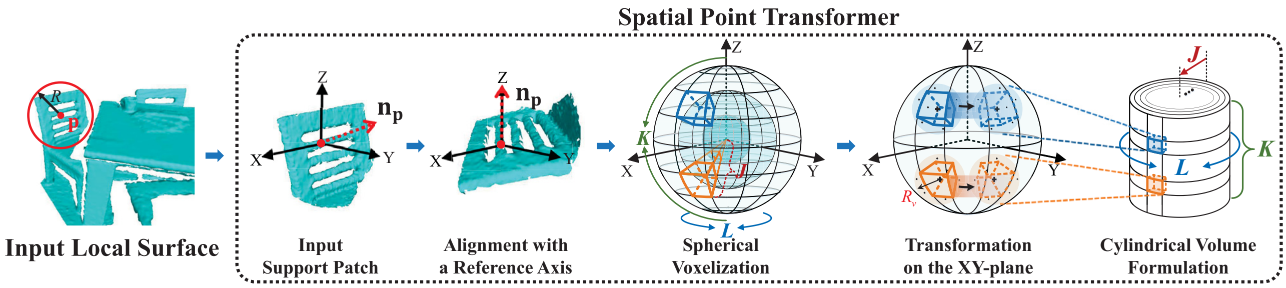

Figure 2: The detailed components and processing steps of our Spatial Point Transformer.

we proactively rotate each voxel around the Z-axis to align its center $\mathbf{v}_{jkl}$ with the YZ-plane, where the rotation matrix $\mathbf{R}_{jkl}$ is defined as:

$$\mathbf{R}_{jkl} = \begin{bmatrix} \cos(\pi/2 - 2\pi l/L) & -\sin(\pi/2 - 2\pi l/L) & 0 \\ \sin(\pi/2 - 2\pi l/L) & \cos(\pi/2 - 2\pi l/L) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

This removes an additional rotational degree of freedom for each voxel on the XY-plane, without dropping any local geometric patterns of each voxel. Note that, the existing methods [50, 62] usually use handcrafted features to achieve rotation invariance, resulting in the loss of the rich local patterns. Uniquely, our simple strategy to transform voxels can preserve these patterns, leaving them to be learned by the powerful neural layers.

Cylindrical Volume Formulation. Once the local patterns of each voxel are transformed, it is crucial to further preserve the larger spatial structures across multiple voxels. This requires the relative positions of all voxels to be represented in the whole framework. To this end, we reformulate the spherical voxels into a cylindrical volume. This is amenable to the proposed 3D cylindrical convolutional network, which guarantees the SO(2) equivariance of the input local surface and preserves the topological patterns of multiple voxels. In particular, given the transformed spherical voxels, each of which has a set of neighbouring points, we logically project them into a cylindrical volume, denoted as $\mathbf{C} \in \mathbb{R}^{J \times K \times L \times k_v \times 3}$ and illustrated in Figure 2.

In summary, given an input surface patch, our Spatial Point Transformer explicitly aligns its Z-axis with a reference axis, and proactively transforms the spherical voxel patterns on the XY-plane, and further preserves the topological surface structures through the cylindrical volume formulation. Clearly, this module keeps all surface patterns intact for the subsequent Neural Feature Extractor to learn.

3.3. Neural Feature Extractor

This module is designed to learn the general features from the transformed points within each cylindrical voxel using the powerful neural layers. As shown in Figure 3, it consists of two components, as discussed below.

Point-based Layers. Given the points within each cylindrical voxel, we use shared MLPs followed by a max-pooling function $\mathcal{A}(\cdot)$ to learn an initial signature for each voxel. Formally, the point-based layers are defined as:

$$\mathbf{f}_{jkl} = \mathcal{A}(\mathbf{MLPs}(\mathbf{R}_{jkl}\mathbf{P}_{jkl})) \tag{4}$$

where $\mathbf{f}_{jkl}$ is the learned features with D dimension. Note that, the MLP weights are shared across all spherical voxels. Eventually, we obtain a set of 3D cylindrical feature maps $\mathbf{F} \in \mathbb{R}^{J \times K \times L \times D}$.

3D Cylindrical Convolutional Layers. To further learn wider spatial structures across multiple voxels of the cylindrical volume, we propose an efficient 3D Cylindrical Convolution Network (3DCCN) inspired by [31]. In particular, given a voxel located at the position (j, k, l) on the d^{th} cylindrical feature map in the s^{th} layer, our 3DCCN is defined as follows.

$$\mathbf{F}^{sd'}_{jkl} = \sum_{d=1}^{D} \sum_{r=1}^{R_s} \sum_{y=1}^{Y_s} \sum_{x=1}^{X_s} w^{sd'd}_{ryx} \mathbf{F}^{(s-1)d}_{(j+r)(k+y)(l+x)}. \tag{5}$$

where R_s is the size of the kernel along the radial dimension, Y_s and X_s are the height and width of the kernel respectively, $w^{sd'd}_{ryx}$ are the learnable parameters.

Being quite different from existing convolution operations, our proposed 3DCCN is novel in the following two aspects. **First**, since the cylindrical feature maps are 360° continuous over a cylinder, our 3DCCN is designed to wrap around these feature maps, *i.e.,* over the periodic boundary from $-180°$ to $180°$. Therefore, explicit padding is not required in our 3DCCN, but required by 3D-CNN at the boundary of feature maps. **Second**, compared with the existing 3D manifold sparse convolution [8] or kernel point convolutions [2], the continuous convolution around the 360° volume enables the obtained feature map to be SO(2) equivariant, hence to achieve the final rotation-invariance.

After stacking multiple of these 3DCCN layers followed by max-pooling, the original cylindrical feature maps are compressed to a compact and representative feature vector.

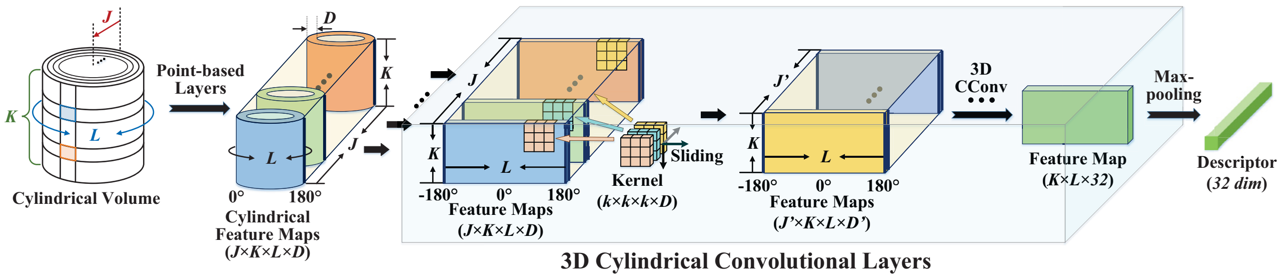

Figure 3: Illustration of the proposed Neural Feature Extractor.

3.4. End-to-end Implementation

The Spatial Point Transformer is directly connected with the Neural Feature Extractor, followed by the existing contrastive loss [2] for end-to-end optimization. The widely-used *hardest in batch* sampling [42] is also adopted on-the-fly to maximize the distance between the closest positive and the closest negative patches. Details of the neural layers are presented in the appendix.

We implement our SpinNet based on the PyTorch framework. The Adam optimizer [35] with default parameters is used. The initial learning rate is set to 0.001 and decayed with a rate of 0.5 for every 5 epochs. We train the network for 20 epochs, the best-performed model on the validation set is then used for testing. For a fair comparison, we keep the same setting for all experiments. All experiments are conducted on the platform with Intel Xeon CPU @2.30GHZ with an NVIDIA RTX2080Ti GPU.

4. Experiments

We first evaluate our SpinNet on the indoor 3DMatch dataset [65] and the outdoor KITTI dataset [20]. We then evaluate the generalization ability of our approach across multiple unseen datasets [65, 20, 46] acquired by different sensors. Lastly, extensive ablation studies are conducted.

Experimental Setup. We follow [2, 65] to generate training samples by only considering the point cloud fragment pairs with more than 30% overlap in the whole dataset. For each paired fragment $\mathbf{P}$ and $\mathbf{Q}$, we randomly sample a fixed number of anchor points from the overlapping region of $\mathbf{P}$, and then apply the ground-truth transformation $\mathbf{T} = \{\mathbf{R}, \mathbf{t}\}$ to determine the corresponding points in fragment $\mathbf{Q}$. For each anchor point, we randomly sample 2048 points from its support region.

4.1. Evaluation on Indoor 3DMatch Dataset

3DMatch is a RGBD-reconstruction dataset, which consists of 62 real-world indoor scenes collected from existing dataset [56, 49, 57, 34, 56, 11]. We follow the official protocol provided in [2] to divide the scenes into training and testing splits. Each scene contains several partially overlapped fragments, and has the ground truth transformation parameters available for evaluation. Feature Matching Recall (FMR) [13] is used as the standard metric.

Comparisons with the state-of-the-arts. We first compare the FMR scores achieved by our SpinNet and strong baselines (including LMVD [37], D3Feat [2], FCGF [8], PerfectMatch [22], PPFNet [14], and PPF-FoldNet [13]) on the 3DMatch dataset, under the conditions of sampling points f=5000, distance threshold τ_1=10 cm and inlier ratio threshold τ_2=5%. To further evaluate the robustness of all approaches against rotations, we follow [14, 2] to build a rotated 3DMatch benchmark by applying arbitrary rotations in SO(3) group to all fragments of the dataset. The rotation distribution of each dataset is showed in the appendix.

	Origin		Rotated		Feat.	Rot.
	FMR (%)	STD	FMR (%)	STD	dim.	Aug.
FPFH [48]	35.9	13.4	36.4	13.6	33	No
SHOT [54]	23.8	10.9	23.4	9.5	352	No
3DMatch [65]	59.6	8.8	1.1	-	512	No
CGF [32]	58.2	14.2	58.5	14.0	32	No
PPFNet [14]	62.3	10.8	0.3	-	64	No
PPF-FoldNet [13]	71.8	10.5	73.1	10.4	512	No
PerfectMatch [22]	94.7	2.7	94.9	2.5	32	No
FCGF [8]	95.2	2.9	95.3	3.3	32	Yes
D3Feat-rand [2]	95.3	2.7	95.2	3.2	32	Yes
D3Feat-pred [2]	95.8	2.9	95.5	3.5	32	Yes
LMVD [37]	97.5	2.8	96.9	-	32	No
SpinNet (Ours)	**97.6**	**1.9**	**97.5**	**1.9**	**32**	**No**

Table 1: Quantitative results on the 3DMatch dataset, STD: standard deviation. The symbol '-' means the results are unavailable or STD under low FMRs (<10%).

As shown in Table 1, the descriptor generated by our method achieves the highest average FMR score and the lowest standard deviation on both the original and rotated datasets, outperforming the state-of-the-art methods. Note that, several baselines [8, 2] require rotation-based data augmentation for training, whilst ours does not.

Performance under different number of sampled points. We further evaluate the performance of our SpinNet on the

#Sampled points	5000	2500	1000	500	250	Average
		Feature Matching Recall (%)				
PerfectMatch [22]	94.7	94.2	92.6	90.1	82.9	90.9
FCGF [8]	95.2	95.5	94.6	93.0	89.9	93.6
D3Feat-rand [2]	95.3	95.1	94.2	93.6	90.8	93.8
D3Feat-pred [2]	95.8	95.6	94.6	94.3	93.3	94.7
SpinNet (Ours)	**97.6**	**97.5**	**97.3**	**96.3**	**94.3**	**96.6**

Table 2: Quantitative results on the 3DMatch dataset using different numbers of sampled points.

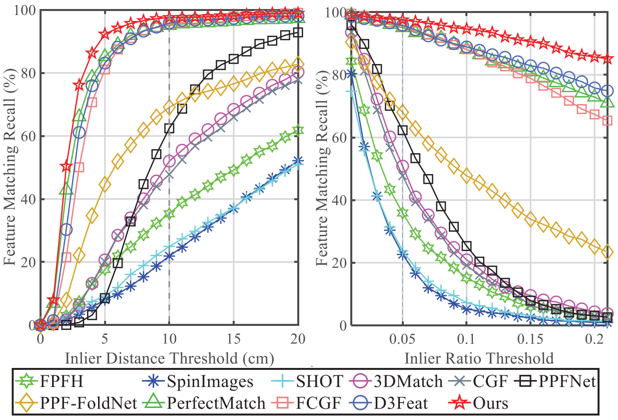

Figure 4: Feature matching recall on the 3DMatch dataset under different inlier distance threshold τ_1 (Left) and inlier ratio threshold τ_2 (Right).

3DMatch by taking different number of sampled points as input. As shown in Table 2, the descriptor learned by our SpinNet consistently achieves the best FMR scores when the number of sampled points is reduced from 5000 to 250. In particular, by randomly selecting points, our method even outperforms D3Feat-pred which has an explicit keypoint detection module. This demonstrates our network is highly robust and not sensitive to the number of sampled points.

	RTE (cm)		RRE (°)		Success (%)
	AVG	STD	AVG	STD	
3DFeat-Net [60]	25.9	26.2	0.57	0.46	95.97
FCGF [8]	9.52	1.30	0.30	0.28	96.57
D3Feat-rand [2]	8.78	0.44	0.32	0.07	99.81
D3Feat-pred [2]	**6.90**	**0.30**	**0.24**	**0.06**	**99.81**
SpinNet (Ours)	9.88	0.50	0.47	0.09	99.10

Table 3: Quantitative results of different approaches on the KITTI odometry dataset. The scores of baselines are retrieved from [2].

Performance under Different Error Thresholds. Additionally, we evaluate the robustness of SpinNet by varying the error thresholds (τ_1 and τ_2). As shown in Figure 4, the descriptor generated by SpinNet consistently outperforms other methods under all thresholds. It is worth noting that the FMR score of our method is significantly higher than others, when the inlier ratio threshold increases. For a stricter condition $\tau_2 = 0.2$, our method maintains a high FMR score of 85.7%, while D3Feat and FCGF drop to 75.8% and 67.4%, respectively. This highlights that our approach is more robust in harder scenarios.

4.2. Evaluation on Outdoor KITTI Dataset

KITTI odometry [20] is an outdoor sparse point cloud dataset acquired by Velodyne-64 3D LiDAR scanners. It consists of 11 sequences of outdoor scans. For fair comparison, we follow the same dataset splits and preprocessing methods as used in D3Feat [2, 8]. Similar to [39], Relative Translational Error (RTE), Relative Rotation Error (RRE), and Success rate are used as the evaluation metrics. The registration is regarded as successful if the RTE and RRE of a pair of fragments are both below the predefined thresholds 2m and 5° in [60]. It is noted that the point clouds are gravity-aligned in this dataset, we follow [60] to skip the alignment with a reference axis in our method. As shown in Table 3, the results of our SpinNet are on par with the strong baseline D3Feat. Admittedly, our SpinNet is marginally lower than the state-of-the-art D3Feat-pred, primarily because D3Feat has a powerful joint learned descriptor and keypoint detector. Also, the well aligned point clouds in this dataset are indeed in favor of D3Feat. We leave the integration of keypoint detection for future exploration.

4.3. Generalization across Unseen Datasets

We have conducted several groups of experiments to extensively evaluate the generalization ability of our SpinNet. In each group, our network is trained on one dataset, and then directly tested on a completely unseen dataset.

	Param.	Gazebo		Wood		Avg.
	(Mb)	Summer	Winter	Autumn	Summer	
FPFH† [48]	-	38.6	14.2	14.8	20.8	22.1
SHOT† [54]	-	73.9	45.7	60.9	64.0	61.1
3DMatch [65]	13.40	22.8	8.3	13.9	22.4	16.9
CGF [32]	1.86	37.5	13.8	10.4	19.2	20.2
PerfectMatch [22]	3.26	91.3	84.1	67.8	72.8	79.0
FCGF [8]	33.48	22.8	10.0	14.8	16.8	16.1
D3Feat (rand) [2]	13.42	45.7	23.9	13.0	22.4	26.2
D3Feat (pred) [2]	13.42	85.9	63.0	49.6	48.0	61.6
LMVD [37]	2.66	85.3	72.0	84.0	78.3	79.9
SpinNet (Ours)	2.16	**92.9**	**91.7**	**92.2**	**94.4**	**92.8**

Table 4: Quantitative results on the ETH dataset, where † denotes handcrafted descriptor. Note that, all learned methods are only trained on the indoor 3DMatch dataset. The FMR scores at τ_1 = 10cm, τ_2 = 5% are compared.

Generalization from 3DMatch to ETH dataset. Following the settings in [2], all models are only trained on the 3DMatch dataset, and then directly tested on the ETH dataset [46]. Note that, the ETH dataset consists of four scenes, *i.e.*, Gazebo-Summer, Gazebo-Winter, Wood-Summer, and Wood-Autumn. Different from 3DMatch, the ETH dataset is acquired by static terrestrial scanners and dominated by outdoor vegetation, such as trees and bushes. In addition, the fragments of point clouds in the ETH dataset have lower resolution and contain more complex geometries compared with the 3DMatch dataset. The large domain gap between these two datasets poses a great challenge to the generalization of all approaches.

As shown in Table 4, the performance of all baselines, namely D3Feat, FCGF, 3DMatch, and CGF, show a significant drop on the ETH dataset. Their FMR scores decrease up to 80% compared with their results on the original 3DMatch dataset, as shown in Table 1, and some techniques are even lower in performance than handcrafted descriptors such as SHOT. Fundamentally, the poor generalization of these methods is attributed to the fact that the descriptors learned by D3Feat, FCGF, and 3DMatch are variant to rigid transformations such as rotation and translation.

The descriptor generated by our SpinNet achieves the highest FMR scores on all four scenes, significantly surpassing the second-best method (LMVD) by about 13%. This clearly shows that our method has excellent generalization ability across the unseen dataset collected by a new sensor modality. This is primarily because our SpinNet is explicitly designed to achieve rotational invariance. The first row of Figure 5 shows the qualitative results.

	Origin		Rotated	
	FMR (%)	STD (%)	FMR (%)	STD (%)
FPFH† [48]	35.9	13.4	<u>36.4</u>	13.6
SHOT† [54]	23.8	10.9	23.4	9.5
FCGF [8]	32.5	7.4	1.0	1.0
D3Feat-rand [2]	60.7	7.7	17.2	4.6
D3Feat-pred [2]	<u>62.7</u>	8.1	17.8	3.2
SpinNet (Ours)	**84.5**	**5.9**	**84.2**	5.8

Table 5: Quantitative results of different methods on the indoor 3DMatch dataset, where † denotes handcrafted descriptor. Note that, all learned methods are only trained on the outdoor KITTI dataset.

	RTE (cm)		RRE (°)		Success (%)
	AVG	STD	AVG	STD	
FCGF [8]	27.1	5.58	1.61	1.51	24.19
D3Feat-rand [2]	37.8	9.98	1.58	1.47	18.47
D3Feat-pred [2]	31.6	10.1	1.44	1.35	<u>36.76</u>
SpinNet (Ours)	**15.6**	**1.89**	**0.98**	**0.63**	**81.44**

Table 6: Quantitative results on the KITTI dataset. Note that, all models are trained on the indoor 3DMatch dataset, while being directly tested on the outdoor KITTI dataset.

Generalization from KITTI to 3DMatch dataset. All models are trained on the outdoor KITTI dataset which is mainly composed of sparse LiDAR point clouds, and then directly tested on the indoor 3DMatch dataset which consists of dense point clouds reconstructed from RGBD images. As presented in Table 5, both D3Feat and FCGF achieve poor results on the 3DMatch dataset, especially when arbitary rotation in SO(3) exists. Their scores are even lower than the traditional methods such as FPFH, primarily because both D3Feat and FCGF have large numbers of parameters and tend to overfit the KITTI dataset, without learning the representative and general local patterns that can be applicable to the unseen dataset. By comparison, our SpinNet achieves an overall FMR score of 79.6%, demonstrating the superior generalization across novel scenarios. The second row of Figure 5 shows the qualitative results.

Generalization from 3DMatch to KITTI dataset. Additionaly, we evaluate the generalization ability from 3DMatch to KITTI dataset. All models are only trained on the indoor 3DMatch dataset, and then directly tested on the outdoor KITTI dataset. Table 6 presents the quantitative results. Because these two datasets are collected by different types of sensors, there is a large gap between the data distributions. Neither FCGF nor D3Feat can effectively generalize from 3DMatch to KITTI dataset. However, our method still demonstrates an excellent success rate of 69.19%, doubling that of the second best method. The third row of Figure 5 shows the qualitative results.

4.4. Ablation Study

To systematically evaluate the effectiveness of each component in our SpinNet, we conduct extensive ablative experiments on both 3DMatch and ETH datasets. In particular, we train all ablated models on the 3DMatch dataset, and then directly test them on both 3DMatch and ETH datasets.

(1) Only removing the alignment with a reference axis. Initially, the reference axis is computed to align the input patch with the Z-axis. By removing this step, the rotation invariance on SO(3) is no longer maintained.

	3DMatch				ETH	
	Origin		Rotated		Origin	Rotated
Inlier ratio τ_2 =	0.05	0.2	0.05	0.2	0.05	0.05
(1) W/o reference axis	95.1	80.0	63.0	23.1	83.9	13.5
(2) W/o transformation	93.5	70.8	87.7	44.8	60.5	47.7
(3) W/o Point Nets	94.0	66.3	93.8	66.1	42.6	42.4
(4) Replacing 3DCCN	64.4	10.6	64.4	10.4	0.0	0.0
(5) The full method	**97.6**	**85.7**	**97.5**	**86.1**	**92.8**	**92.4**

Table 7: The FMR scores of all ablated models on the 3DMatch and ETH datasets with $\tau_1 = 0.1$cm.

(2) Only removing the transformation on the XY-plane. The transformation employed on the XY-plane is originally

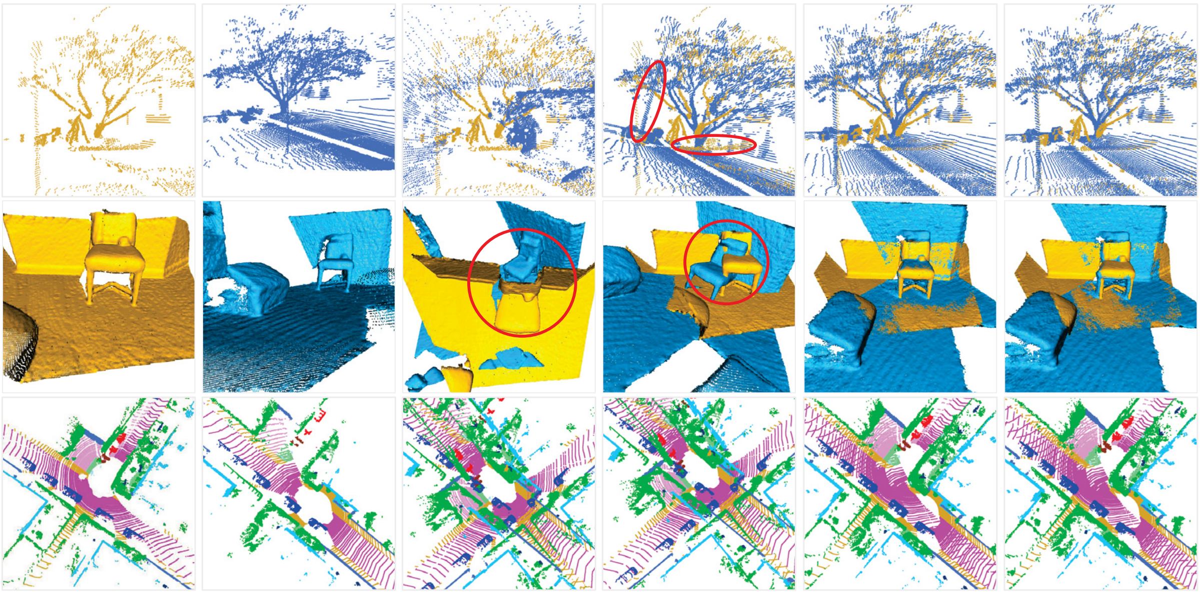

Figure 5: Qualitative results of our method on unseen datasets. The first row is from 3DMatch to ETH, the second row is from KITTI to 3DMatch, and the third row is from 3DMatch to KITTI.

designed to eliminate the rotation variance of each voxel in the plane. In this experiment, we remove the transformation and directly operate on the non-transformed spherical voxels to formulate the cylindrical volume.

(3) Only replacing the point-based layers with density. Instead of using the Point-based layers to learn a signature for each cylindrical voxel, we manually compute the point density of each voxel as its signature. Basically, this is to validate whether our point-based learned features are more general and representative than the commonly used, yet limited, handcrafted feature.

(4) Only replacing 3DCCN by MLPs. The 3DCNN is designed to learn larger spatial structures from multiple voxels, whilst maintaining rotation equivariance. In this experiment, we replace the 3DCNN layers with the same number of MLP layers shared by all cylindrical voxels. These MLPs are unable to learn a wide context.

Analysis. Table 7 shows the results of all ablated networks on the 3DMatch dataset, as well as the generalization performance on the ETH datasets. It can be seen that: 1) Without using the alignment of a reference axis or the transformation of spherical voxels, the ablated models are unable to effectively match the point clouds either in 3DMatch or ETH datasets, especially for the point clouds with random rotations. This shows that the proposed Spatial Point Transformer indeed plays an important role to achieve rotation invariance in our SpinNet. 2) Without using the advanced point-based neural layers to learn the signatures for spherical voxels, the ablated method can obtain consistent results on the 3DMatch dataset using the simple handcrafted feature, *i.e.,* point density, but fails to generalize to the unseen ETH dataset. This clearly demonstrates that the learned local features tend to be much more powerful and general than the handcrafted features. 3) Without using the 3DCCN to learn larger surface structures, the ablated model only obtains significantly lower scores on both the 3DMatch and ETH datasets. This demonstrates that our 3DCCN is a key to preserving the local spatial patterns.

5. Conclusion

In this paper, we present a new neural descriptor to learn compact representations for complex 3D surfaces. The learned representations are rotation invariant, descriptive, and able to preserve complex local geometric patterns. Extensive experiments demonstrate that our descriptor has remarkable generalization ability across unseen scenarios and achieves superior results for 3D point cloud registration. In future, we will investigate the integration of keypoint detector, as well as the fully-convolutional architecture.

Acknowledgements. This work was partially supported by the National Natural Science Foundation of China (No. U20A20185, 61972435), the Natural Science Foundation of Guangdong Province (2019A1515011271), and the Shenzhen Science and Technology Program (No. RCYX20200714114641140, JCYJ20190807152209394).

References

[1] Sheng Ao, Yulan Guo, Jindong Tian, Yong Tian, and Dong Li. A Repeatable and Robust Local Reference Frame for 3D Surface Matching. PR, 2020. 3

[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 CVPR, 2020. 1, 2, 4, 5, 6, 7, 12, 13, 14, 15, 16, 17

[3] Chao Chen, Guanbin Li, Ruijia Xu, Tianshui Chen, Meng Wang, and Liang Lin. ClusterNet: Deep Hierarchical Cluster Network with Rigorously Rotation-Invariant Representation for Point Cloud Analysis. In CVPR, 2019. 2

[4] Hui Chen and Bir Bhanu. 3D Free-form Object Recognition in Range Images Using Local Surface Patches. PRL, 2007. 2

[5] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust Reconstruction of Indoor Scenes. In CVPR, 2015. 12

[6] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep Global Registration. In CVPR, 2020. 1

[7] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal Convnets: Minkowski Convolutional Neural Networks. In CVPR, 2019. 2

[8] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully Convolutional Geometric Features. In ICCV, 2019. 1, 2, 4, 5, 6, 7, 12, 13, 14, 15, 16, 17

[9] Chin Seng Chua and Ray Jarvis. Point Signatures: A New Representation for 3D Object Recognition. IJCV, 1997. 1, 2

[10] Taco S. Cohen, Mario Geiger, Jonas Koehler, and Max Welling. Spherical CNNs. In ICLR, 2018. 2

[11] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. BundleFusion: Real-Time Globally Consistent 3D Reconstruction Using On-the-Fly Surface Reintegration. TOG, 2017. 5

[12] Michaël Defferrard, Martino Milani, Frédérick Gusset, and Nathanaël Perraudin. DeepSphere: A Graph-based Spherical CNN. In ICLR, 2020. 1

[13] Haowen Deng, Tolga Birdal, and Slobodan Ilic. PPF-FoldNet: Unsupervised Learning of Rotation Invariant 3D Local Descriptors. In ECCV, 2018. 1, 3, 5, 12, 14

[14] Haowen Deng, Tolga Birdal, and Slobodan Ilic. PPFNet: Global Context Aware Local Features for Robust 3D Point Matching. In CVPR, 2018. 1, 3, 5, 12, 14

[15] Haowen Deng, Tolga Birdal, and Slobodan Ilic. 3d local features for direct pairwise registration. In CVPR, 2019. 2, 12

[16] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model Globally, Match Locally: Efficient and Robust 3D Object Recognition. In CVPR, 2010. 1

[17] Gil Elbaz, Tamar Avraham, and Anath Fischer. 3D Point Cloud Registration for Localization using a Deep Neural Network Auto-Encoder. In CVPR, 2017. 1

[18] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning SO(3) Equivariant Representations with Spherical CNNs. In ECCV, 2018. 3

[19] Andrea Frome, Daniel Huber, Ravi Kolluri, Thomas Bülow, and Jitendra Malik. Recognizing Objects in Range Data using Regional Point Descriptors. In ECCV, 2004. 1, 12

[20] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In CVPR, 2012. 2, 5, 6, 13, 15

[21] Zan Gojcic, Caifa Zhou, Jan D. Wegner, Leonidas J. Guibas, and Tolga Birdal. Learning Multiview 3D Point Cloud Registration. In CVPR, 2020. 1

[22] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The Perfect Match: 3D Point Cloud Matching with Smoothed Densities. In CVPR, 2019. 1, 3, 5, 6, 14

[23] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In CVPR, 2018. 1

[24] Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, and Jianwei Wan. 3D Object Recognition in Cluttered Scenes with Local Surface Features: A Survey. IEEE TPAMI, 2014. 1

[25] Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, Jianwei Wan, and Ngai Ming Kwok. A Comprehensive Performance Evaluation of 3D Local Feature Descriptors. IJCV, 2016. 2

[26] Yulan Guo, Ferdous Sohel, Mohammed Bennamoun, Min Lu, and Jianwei Wan. Rotational Projection Statistics for 3D Local Surface Description and Object Recognition. IJCV, 2013. 2, 12

[27] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep Learning for 3D Point Clouds: A Survey. IEEE TPAMI, 2020. 1

[28] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. In CVPR, 2020. 1

[29] Xiaoshui Huang, Guofeng Mei, and Jian Zhang. Feature-metric Registration: A Fast Semi-supervised Approach for Robust Point Cloud Registration without Correspondences. In CVPR, 2020. 1

[30] Andrew E. Johnson and Martial Hebert. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. IEEE TPAMI, 1999. 2, 12

[31] Sunghun Joung, Seungryong Kim, Hanjae Kim, Minsu Kim, Ig-Jae Kim, Junghyun Cho, and Kwanghoon Sohn. Cylindrical Convolutional Networks for Joint Object Detection and Viewpoint Estimation. In CVPR, 2020. 4

[32] Marc Khoury, Qian-Yi Zhou, and Vladlen Koltun. Learning Compact Geometric Features. In ICCV, 2017. 1, 3, 5, 6, 14

[33] Seohyun Kim, Jaeyoo Park, and Bohyung Han. Rotation-Invariant Local-to-Global Representation Learning for 3D Point Cloud. In NeurIPS, 2020. 1, 3

[34] Vladimir G Kim, Siddhartha Chaudhuri, Leonidas Guibas, and Thomas Funkhouser. Shape2pose: Human-Centric Shape Analysis. TOG, 2014. 5

[35] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In ICLR, 2015. 5

[36] Hongdong Li and Richard Hartley. The 3D-3D Registration Problem Revisited. In ICCV, 2007. 3

[37] Lei Li, Siyu Zhu, Hongbo Fu, Ping Tan, and Chiew-Lan Tai. End-to-End Learning Local Multi-view Descriptors for 3D Point Clouds. In CVPR, 2020. 1, 3, 5, 6, 14

[38] 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 ICCV, 2019. 1
[39] Yanxin Ma, Yulan Guo, Jian Zhao, Min Lu, Jun Zhang, and Jianwei Wan. Fast and Accurate Registration of Structured Point Clouds with Small Overlaps. In CVPRW, 2016. 6, 13
[40] Ajmal Mian, Mohammed Bennamoun, and Robyn Owens. On the Repeatability and Quality of Keypoints for Local Feature-based 3D Object Retrieval from Cluttered Scenes. IJCV, 2010. 1, 3
[41] Ajmal S Mian, Mohammed Bennamoun, and Robyn Owens. Three-Dimensional Model-Based Object Recognition and Segmentation in Cluttered Scenes. IEEE TPAMI, 2006. 1
[42] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working Hard to Know Your Neighbor's Margins: Local Descriptor Learning Loss. In NeurIPS, 2017. 5
[43] John Novatnack and Ko Nishino. Scale-Dependent/Invariant Local 3D Shape Descriptors for Fully Automatic Registration of Multiple Sets of Range Images. In ECCV, 2008. 1
[44] Alioscia Petrelli and Luigi Di Stefano. On the Repeatability of The Local Reference Frame for Partial Shape Matching. In ICCV, 2011. 3
[45] Fabio Poiesi and Davide Boscaini. Distinctive 3D Local Deep Descriptors. In ICPR, 2021. 1
[46] François Pomerleau, Ming Liu, Francis Colas, and Roland Siegwart. Challenging Data Sets for Point Cloud Registration Algorithms. IJRR, 2012. 1, 2, 5, 7, 13
[47] Yongming Rao, Jiwen Lu, and Jie Zhou. Spherical Fractal Convolutional Neural Networks for Point Cloud Recognition. In CVPR, 2019. 1
[48] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast Point Feature Histograms (FPFH) for 3D Registration. In ICRA, 2009. 1, 2, 5, 6, 7, 14
[49] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In CVPR, 2013. 5
[50] Riccardo Spezialetti, Samuele Salti, and Luigi Di Stefano. Learning an Effective Equivariant 3D Descriptor Without Supervision. In ICCV, 2019. 1, 3, 4
[51] Riccardo Spezialetti, Federico Stella, Marlon Marcon, Luciano Silva, Samuele Salti, and Luigi Di Stefano. Learning to Orient Surfaces by Self-supervised Spherical CNNs. In NeurIPS, 2020. 1, 12
[52] Fridtjof Stein, Gérard Medioni, et al. Structural Indexing: Efficient 3D Object Recognition. IEEE TPAMI, 1992. 1
[53] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In ICCV, 2019. 1
[54] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique Signatures of Histograms for Local Surface Description. In ECCV, 2010. 2, 5, 6, 7
[55] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique Signatures of Histograms for Local Surface Description. In ECCV, 2010. 14
[56] Julien Valentin, Angela Dai, Matthias Nießner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to Navigate the Energy Landscape. In 3DV, 2016. 5
[57] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A Database of Big Spaces Reconstructed using SfM and Object Labels. In ICCV, 2013. 5
[58] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning Object Bounding Boxes for 3D Instance Segmentation on Point Clouds. In NeurIPS, 2019. 1
[59] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. FoldingNet: Point Cloud Auto-encoder via Deep Grid Deformation. In CVPR, 2018. 3
[60] Zi Jian Yew and Gim Hee Lee. 3DFeat-Net: Weakly Supervised Local 3D Features for Point Cloud Registration. In ECCV, 2018. 2, 6, 13
[61] Zi Jian Yew and Gim Hee Lee. RPM-Net: Robust Point Matching using Learned Features. In CVPR, 2020. 2
[62] Yang You, Yujing Lou, Qi Liu, Yu-Wing Tai, Lizhuang Ma, Cewu Lu, and Weiming Wang. Pointwise Rotation-Invariant Network with Adaptive Sampling and 3D Spherical Voxel Convolution. In AAAI, 2020. 1, 3, 4
[63] Ruixuan Yu, Xin Wei, Federico Tombari, and Jian Sun. Deep Positional and Relational Feature Learning for Rotation-Invariant Point Cloud Analysis. In ECCV, 2020. 2
[64] Andrei Zaharescu, Edmond Boyer, Kiran Varanasi, and Radu Horaud. Surface Feature Detection and Description with Applications to Mesh Matching. In ICCV, 2009. 1
[65] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. In CVPR, 2017. 1, 2, 5, 6, 13, 14, 16, 17
[66] Yongheng Zhao, Tolga Birdal, Jan Eric Lenssen, Emanuele Menegatti, Leonidas Guibas, and Federico Tombari. Quaternion Equivariant Capsule Networks for 3D Point Clouds. In ECCV, 2020. 1
[67] Yu Zhong. Intrinsic Shape Signatures: A Shape Descriptor for 3D Object Recognition. In ICCVW, 2009. 1
[68] Lei Zhou, Siyu Zhu, Zixin Luo, Tianwei Shen, Runze Zhang, Mingmin Zhen, Tian Fang, and Long Quan. Learning and Matching Multi-view Descriptors for Registration of Point Clouds. In ECCV, 2018. 2