2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3503-0129-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/CVPR52729.2023.00127

BUFFER: Balancing Accuracy, Efficiency, and Generalizability in Point Cloud Registration

Sheng Ao[1], Qingyong Hu[2], Hanyun Wang[3], Kai Xu[4], Yulan Guo[1*]
[1]The Shenzhen Campus of Sun Yat-sen University, Sun Yat-sen University, [2]University of Oxford,
[3]Information Engineering University, [4]National University of Defense Technology

Abstract

An ideal point cloud registration framework should have superior accuracy, acceptable efficiency, and strong generalizability. However, this is highly challenging since existing registration techniques are either not accurate enough, far from efficient, or generalized poorly. It remains an open question that how to achieve a satisfying balance between this three key elements. In this paper, we propose BUFFER, a point cloud registration method for ***b****alancing accu****r****acy,* ***eff****iciency, and gen****er****alizability. The key to our approach is to take advantage of both point-wise and patch-wise techniques, while overcoming the inherent drawbacks simultaneously. Different from a simple combination of existing methods, each component of our network has been carefully crafted to tackle specific issues. Specifically, a Point-wise Learner is first introduced to enhance computational efficiency by predicting keypoints and improving the representation capacity of features by estimating point orientations, a Patch-wise Embedder which leverages a lightweight local feature learner is then deployed to extract efficient and general patch features. Additionally, an Inliers Generator which combines simple neural layers and general features is presented to search inlier correspondences. Extensive experiments on real-world scenarios demonstrate that our method achieves the best of both worlds in accuracy, efficiency, and generalization. In particular, our method not only reaches the highest success rate on unseen domains, but also is almost 30 times faster than the strong baselines specializing in generalization. Code is available at* `https://github.com/aosheng1996/BUFFER`.

1. Introduction

Point cloud registration plays a critical role in LiDAR SLAM [23, 25], 3D reconstruction [44], and robotic navigation [22, 36]. An ideal registration framework not only requires aligning geometries accurately and efficiently, but also can be generalized to unseen scenarios acquired by different sensors. However, due to uneven data quality (*e.g.*, noise distribution, non-uniform density, varying viewing angles, domain gaps across different sensors), it remains challenging to simultaneously achieve a satisfactory balance between efficiency, accuracy, and generalization.

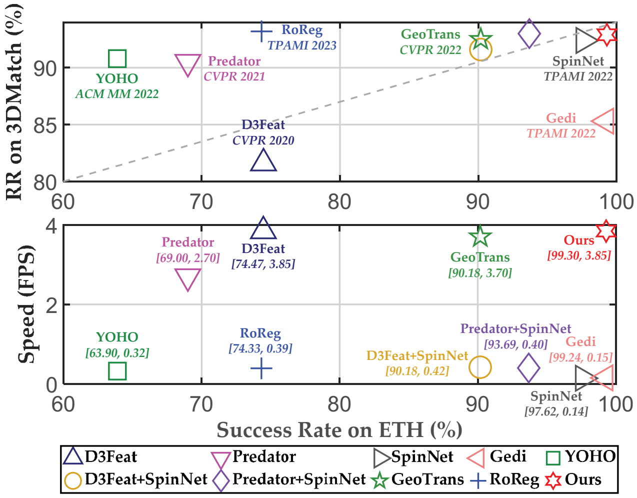


Figure 1. Comparisons of the registration accuracy on the indoor 3DMatch [66] dataset, efficiency, and generalizability on the outdoor ETH [49] dataset of different approaches. Note that, all methods are trained only on the 3DMatch dataset. Our method not only achieves the highest recall on 3DMatch, but also has the best generalization ability and efficiency across the unseen ETH dataset.

Existing registration techniques can be mainly categorized into correspondences-based [30, 63, 66] and correspondences-free methods [3, 60, 61]. By establishing a series of reliable correspondences, the correspondences-based methods usually have better registration performance compared with correspondences-free methods, especially in large-scale scenarios. However, these correspondence-based methods are still not ready for large-scale real-world applications as they are either not accurate enough, far from efficient, or generalized poorly.

[*] Corresponding author: guoyulan@sysu.edu.cn.

979-8-3503-0129-8/23/$31.00 ©2023 IEEE
DOI 10.1109/CVPR52729.2023.00127

Overall, the limitations of existing correspondence-based methods lie in two aspects. **First**, there is currently no unified, efficient, and general feature learning framework. A number of patch-wise methods [21, 66] usually employ complex networks coupled with sophisticated steps to encode the fine-grained geometry of local 3D patches. Benefiting from local characteristics that are inherently robust to occlusion and easy to be discriminated, patch-wise methods usually have good generalization ability whilst low efficiency. To improve computational efficiency, several point-wise methods [5, 26] resort to adopting a hierarchical architecture [51] to consecutively sample raw point clouds. However, the hierarchical architecture tends to capture the global context rather than local geometry, which makes the learned point-wise features easy to homogenize and hard to be matched correctly especially for unseen contexts [1]. **Second**, there is no efficient and general correspondence search mechanism. Most correspondences-based registration frameworks [1, 52] leverage the RANSAC [18] or a coarse-to-fine matching strategy [64] to search reliable correspondences. Considering the efficiency of the RANSAC algorithm is related to the inliers, this mechanism would be time-consuming when inlier rate is very low. Additionally, the coarse-to-fine strategy failed to generalize to unseen domains due to the reliance on global context matching.

A handful of recent works also attempt to leverage unsupervised domain adaptation techniques [24] or simplify the network architecture [1] to achieve a better trade-off between generalization and efficiency. However, they either need an extra target dataset for training or sacrifice the representation capacity of the learned models. Overall, efficiency and generalization seem to contradict each other as existing techniques inherently specialize in one field and do not complement each other.

In this paper, we achieve the best of both worlds on efficiency and generalizability by combining the point-wise and patch-wise methods. An efficient and general search mechanism is also proposed to increase the inlier rate of correspondences. The proposed registration framework, termed BUFFER, mainly consists of a *Point-wise Learner*, a *Patch-wise Embedder*, and an *Inliers Generator*. The input point clouds are first fed into the *Point-wise Learner*, where a novel equivariant fully convolutional architecture is used to predict point-wise saliencies and orientations, further reducing computational cost and enhancing the representation ability of features. With the selected keypoints and learned orientations, the *Patch-wise Embedder* utilizes a lightweight patch-based feature learner, *i.e.,* Mini-SpinNet [1], to extract efficient and general local features and cylindrical feature maps. By matching local features, a set of initial correspondences coupled with corresponding cylindrical feature maps can be obtained. These general cylindrical feature maps are then fed into the *Inlier Generator*, which predicts a rigid transformation for each correspondence using a lightweight 3D cylindrical convolutional network [1] and generates the final reliable set of correspondences by seeking an optimal transformation, followed by RANSAC [18] to estimate a finer transformation.

Actually, it is non-trivial to achieve a satisfactory balance between accuracy, efficiency, and generalizability if simply combining existing methods. For example, the point-wise method Predator [26] is vulnerable to unseen scenarios while the patch-wise method SpinNet [1] is highly time-consuming. When combining them together directly, the whole framework is neither efficient nor general as verified in Fig. 1. In contrast, each component of our BUFFER has been carefully crafted to tackle specific issues, and thus a superior balance is more likely to be realized.

As shown in Fig. 1, being trained only on the 3DMatch dataset, our BUFFER not only achieves the highest registration recall of 92.9% on the 3DMatch dataset, but also reaches the best success rate of 99.30% on the unseen outdoor ETH dataset (significantly surpassing the best point-wise baseline GeoTrans [52] by nearly **10%**). Meanwhile, our BUFFER is almost **an order of magnitude faster** than patch-wise methods [1, 48, 57]. Extensive experiments justify the superior performance and compelling efficiency of our method. Overall, our contributions are three-fold:

- We propose a new point cloud registration framework by skillfully combining the point-wise and patch-wise paradigms, achieving the best of both worlds in accuracy, efficiency, and generalizability.
- We introduce an equivariant fully convolutional architecture to predict point-wise orientations and saliencies.
- A new correspondence search strategy is introduced to enhance the inlier ratio of initial correspondences.

2. Related Work

2.1. Correspondences-based Registration

As the name implies, correspondences-based registration first extracts point cloud features, then establishes explicit point correspondences between two scans by feature matching, finally estimates the rigid transformation. From the perspective of features, existing correspondences-based registration work can be roughly divided into two categories: patch-wise and point-wise methods.

Patch-wise Methods. This category of methods exploits a weight-sharing network to characterize the local 3D patches centered at keypoints, generating sparse descriptions for each fragment. The pioneering work of learned descriptors is 3DMatch, which formulates the input local 3D patch into a voxel representation followed by multiple neural layers to extract deep features. Afterwards, several works [14, 21, 37, 42] successively improve the feature descriptiveness. Recently, a handful of works devote to exploring the generalization ability of descriptors. Ao *et al.* propose a general feature descriptor SpinNet [1, 2] by introducing ingenious cylindrical representation and powerful neural architecture. A contemporary work Gedi [48] combines the quaternion network [53] with PointNet++ [51], achieving good generalization ability. Although patch-wise methods can achieve high registration accuracy with strong generalization, they are extremely time-consuming.

Point-wise Methods. This category of methods processes the whole point cloud in a single forward pass based on a hierarchical architecture, yielding dense descriptions for each fragment. Using Minkowski convolutional neural networks [9] as its backbone network, FCGF [10] is the first work to learn dense feature descriptors for point cloud registration. Next, D3Feat [5], as a representative learning framework, is proposed to predict point-wise 3D keypoints and descriptors. Later, a handful of point-wise methods [15,26] follow this paradigm to learn keypoints and feature descriptors jointly. Recently, Yu *et al.* [64] propose a coarse-to-fine registration framework, which first builds coarse correspondences by superpoint matching and then generates fine correspondences by performing point-wise matching within some plausible regions. In particular, this method does not have an explicit detection component. On this basis, Qin *et al.* [52] introduce the Geometric Transformer to learn more representative features for robust superpoint matching by encoding intra- and inter-point-cloud geometric patterns. Overall, due to the superiority of hierarchical architecture, the point-wise methods are usually much faster than patch-wise methods, while the drawback is its low generalization capacity to unseen scenarios.

Apart from the feature-level efforts, several methods resort to improving registration performance from the correspondence level. RANSAC [18] is one of the classic methods which presents the most commonly used hypothesis-verification pipeline for robust pruning outliers. Recently, deep learning techniques have dominated the field of 3D outlier rejection. A handful of works [8, 45] formulate the outlier rejection as an inlier/outlier classification problem, and leverage the neural network to predict the inlier probability of each correspondence. Additionally, several methods [4, 31] combine traditional techniques such as geometric consistency [7] and Hough voting [55] with deep learning architecture to achieve better outlier removal. However, these methods only utilize input point coordinates, ignoring the characteristics underlying local geometry. Our BUFFER considers both global information (spatial coordinates) and local embeddings (cylindrical features), significantly improving the inlier rate.

2.2. Correspondences-free Registration

Correspondences-free registration means directly estimating the rigid transformation between a pair of fragments, usually achieved by establishing an end-to-end differentiable network. Typically, existing correspondences-free registration methods can be divided into soft correspondences-based and direct regression-based, according to the difference in network architecture.

Soft Correspondence Based. Despite no explicit correspondences, this methods [19, 38, 39, 59, 62] usually rely on soft correspondences between features followed by a differentiable Singular Value Decomposition (SVD) [46] to generate the rigid transformation. To acquire accurate soft correspondences, most registration networks combine well-known techniques such as Graph Neural Networks (GNN) [17,43] and Transformers [6,35] to learn more distinctive features. Although encouraging results have been achieved, it remains challenging for these methods to generalize to unseen scenarios captured by different sensors.

Direct Regression Based. Intuitively, these methods aim to regress the rigid transformation without any hard or soft correspondences. PointNetLK [3] is the first attempt at direct regression methods, which first extracts a global feature for each scan using PointNet [50] and then introduces a differentiable Lucas-Kanade algorithm [40] to minimize the feature distance, finally iteratively aligns the two point clouds. Inspired by this, the subsequent methods [27,34,65] generally follow the embedding-regression pipeline, while the major differences only lie in the choice of the regression algorithm. These methods are highly efficient since point-to-point correspondences are not required. However, most of them are only suitable for object-level registration and cannot be generalized to large-scale scenes.

To sum up, existing methods exhibit satisfactory performance on registration accuracy whilst still cannot achieving the trade-off between efficiency and generalization. In this paper, we solve the problem by skillfully integrating patch-wise and point-wise networks and designing a new 3D registration framework, where the point-wise component is mainly responsible for enhancing efficiency and enabling the patch-wise module to extract general features.

3. BUFFER

3.1. Problem Statement

Given two partially overlapped point clouds $\mathcal{P} = \{\boldsymbol{p}_i \in \mathbb{R}^3 | i = 1, \ldots, N\}$ and $\mathcal{Q} = \{\boldsymbol{q}_j \in \mathbb{R}^3 | j = 1, \ldots, M\}$, the goal of point cloud registration is to compute an optimal rigid transformation $\mathbf{T} = \{\mathbf{R} \in \mathrm{SO}(3), \boldsymbol{t} \in \mathbb{R}^3\}$ between $\mathcal{P}$ and $\mathcal{Q}$. Referring to [32], if there are ground-truth one-to-one correspondences between subsets $\mathcal{P}_c \subset \mathcal{P}$ and $\mathcal{Q}_c \subset \mathcal{Q}$, the registration problem can be reformulated as a minimization problem:

$$\mathcal{L}(\mathcal{P}_c, \mathcal{Q}_c | \mathbf{P}, \mathbf{R}, \boldsymbol{t}) = \frac{1}{N_c} \left\| \mathcal{Q}_c - \mathbf{R}\mathcal{P}_c\mathbf{P} - \boldsymbol{t} \right\|^2, \qquad (1)$$

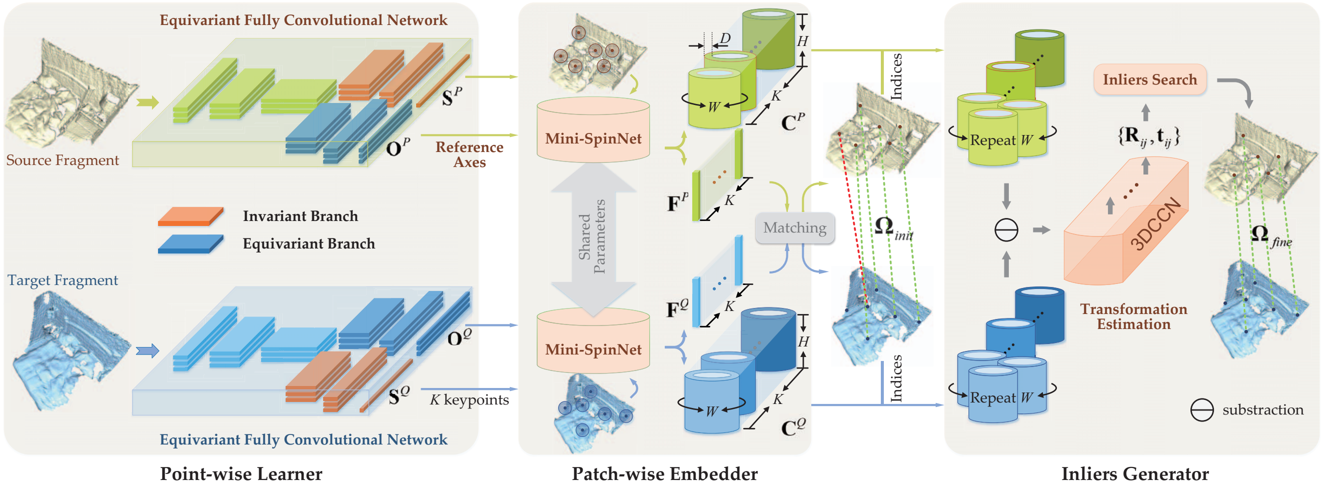


Figure 2. The overall framework of the proposed BUFFER.

where N_c is the number of correctly matched correspondences, $\mathbf{P} \in \mathbb{R}^{N_c \times N_c}$ is a permutation matrix.

To obtain the point subsets $\mathcal{P}_c$ and $\mathcal{Q}_c$, we propose a new registration framework termed BUFFER which mainly consists of a Point-wise Learner, a Patch-wise Embedder, and an Inliers Generator. The pipeline is illustrated in Fig. 2.

3.2. Point-wise Learner

The Point-wise Learner is designed to predict rotation-invariant keypoints and rotation-equivariant orientations, further improving the registration efficiency and accuracy. As shown in Fig. 2, it consists of two components.

Equivariant Fully Convolutional Network. To obtain robust keypoints and point orientations, the first step is to construct a backbone network to learn dense and rotation-equivariant features. Existing methods such as [12, 13] either have extremely high spatial and time complexity, or rely on the global point coordinates which are sensitive to translations. In this paper, we design an Equivariant Fully Convolutional Network (EFCN), which is lightweight and invariant to translations. To ensure the rotational equivariance and translational invariance, we explore the following three geometrical attributes of point cloud $\mathcal{P}$:
(1) *relative coordinate:* $\boldsymbol{p}_{ji} = \boldsymbol{p}_j - \boldsymbol{p}_i$,
(2) *neighborhood center:* $\boldsymbol{c}_i = \frac{1}{|N_i|}\sum_{\boldsymbol{p}_j \in N_i} \boldsymbol{p}_{ji}$,
(3) *initial orientation:* $\boldsymbol{n}_i$ is the eigenvector corresponding to the smallest eigenvalue of $\mathbf{\Sigma} = \frac{1}{|N_i|}\sum_{\boldsymbol{p}_j \in N_i} \boldsymbol{p}_{ji}^{\mathrm{T}}\boldsymbol{p}_{ji}$.
Here, N_i represents all neighboring points of $\boldsymbol{p}_i$ within support radius R. Based on this, the equivariant convolution at a point $\boldsymbol{p}_i$ in the l-th layer can be reformulated as:

$$\boldsymbol{v}_i^{l+1} = \mathbf{VN}(\boldsymbol{v}_j^l; \mathbf{W}), \forall \boldsymbol{p}_j \in N_i, \tag{2}$$

where $\boldsymbol{v}_j^0 = [\boldsymbol{p}_{ji}; \boldsymbol{n}_j; \boldsymbol{n}_j \times \boldsymbol{p}_{ji}; \boldsymbol{c}_i]^{\mathrm{T}}$, $\mathbf{W}$ is a weight matrix, and $\mathbf{VN}$ denotes an equivariant mapping proposed in [13]. Since $\boldsymbol{v}_j^0$ is equivariant to SO(3) rotations and invariant to translations, the whole convolutional network also has the same invariance and equivariance.

Our EFCN is based on the hierarchical architecture of KPConv [54] (details are in supplementary material). Compared with existing equivariant networks [16,28], our EFCN is more efficient and can be applied to scene-level tasks. Though the sampling/upsampling in hierarchical architecture inevitably brings in some quantitative errors, the strict mathematical model behind it already provides a strong inductive bias for the network to learn the equivariant features. The EFCN for the point cloud $\mathcal{Q}$ is the same.

Equivariant and Invariant Branches. The next step is to predict rotation-invariant keypoints and rotation-equivariant orientations. To this end, we feed the equivariant features in the last convolutional layer into two separate decoder branches to produce the dense orientations $\mathbf{O}^P$ and saliencies $\mathbf{S}^P$, as shown in Fig. 2. The same operations are also imposed on the point cloud $\mathcal{Q}$.

Benefiting from the equivariance of each layer, the final learned orientations $\mathbf{O}^P \in \mathbb{R}^{N \times 1 \times 3}$ are naturally equivariant to SO(3) rotations. In another invariant branch, we employ the same invariant transformation as [13] to yield an invariant signal $\mathbf{I}^P \in \mathbb{R}^{N \times C \times 3}$. By flatting $\mathbf{I}^P$ and feeding it into three MLP layers followed by Softplus activation, the final point-wise saliencies $\mathbf{S}^P \in \mathbb{R}^{N \times 1}$ are predicted, where K points with higher saliencies are regarded as keypoints.

In summary, based on our EFCN, the Point-wise Learner can predict dense saliencies to select the easier matched keypoints, thereby improving the registration efficiency. Meanwhile, the Point-wise Learner is able to learn robust point orientations, which is beneficial for the subsequent Patch-wise Embedder to learn highly descriptive features.

3.3. Patch-wise Embedder

This module is designed to learn efficient and general features for the selected keypoints. It contains two key components, as discussed below.

Mini-SpinNet. We leverage a local feature learner, *i.e.*, SpinNet [1], to extract general features. However, the vanilla SpinNet is time-consuming and memory-intensive. To alleviate these problems, we develop a lightweight architecture, termed Mini-SpinNet (see more details in appendix), to extract general local patch features.

Reference Axes. While obtaining a great improvement in efficiency, we must admit that this lightweight structure inevitably deteriorates the discriminability of features. To compensate for the performance, we adopt the learned orientations as reference axes, which are more repeatable and robust than the handcrafted Z-axes used in vanilla SpinNet (as shown in Sect. 4.4), to extract more distinctive features.

Lastly, a set of general local features $\mathbf{F}^P$ and cylindrical feature maps $\mathbf{C}^P$ can be obtained. By performing feature matching between $\mathbf{F}^P$ and $\mathbf{F}^Q$, a series of initial correspondences $\boldsymbol{\Omega}_{init}$ can be established, as shown in Fig. 2. In general, our Patch-wise Embedder is not only typically lightweight and efficient, but also can learn distinctive and general local features for feature matching.

3.4. Inliers Generator

This module is designed to search inliers from a series of initial correspondences, improving the registration performance of the whole framework. As shown in Fig. 2, it consists of two components, as discussed below.

Transformation Estimation. Here, we intend to deal with the inlier search problem from a new perspective of features. Given a list of initial point correspondences $\boldsymbol{\Omega}_{init}$, two cylindrical feature maps $\mathbf{C}_i^P, \mathbf{C}_j^Q \in \mathbb{R}^{H\times W\times D}$ are also obtained for each pair of correspondence $\{\boldsymbol{p}_i, \boldsymbol{q}_j\} \in \boldsymbol{\Omega}_{init}$, where H, W, D represents the height, width, and feature dimensionality of the unfolding cylindrical feature map, respectively. In Sect. 3.3, we can know that the local patch centered at keypoint $\boldsymbol{p}_i$ is pre-aligned with a learned orientation using a rotation matrix $\mathbf{R}_i^p$. Therefore, there is only an SO(2) rotation $\mathbf{R}_{ij}^c$ between $\mathbf{C}_i^P$ and $\mathbf{C}_j^Q$. Based on this, we aim to estimate the SO(2) rotation between two cylindrical feature maps, thereby recovering the rigid transformation between two matched local patches.

Inspired by disparity regression in stereo matching [29], we first construct a 4D matching cost volume $\mathbf{V} \in \mathbb{R}^{H\times W\times W\times D}$ by calculating the difference between two cylindrical feature maps, computed at different width values. Note that, the cost volume is continuous in 360° over a cylinder. To retain this property, a lightweight 3D cylindrical convolutional network (3DCCN) [1] is exploited for cost aggregation $\mathcal{C} : \mathbb{R}^{H\times W\times W\times D} \rightarrow \mathbb{R}^W$. Following a softmax operation $\sigma(\cdot)$, the probability of each offset is obtained. The predicted offset d is then computed by a soft-argmax operation:

$$d = \sum_{i=1}^{W} i \times \sigma_i(\mathcal{C}(\mathbf{V})). \tag{3}$$

Accordingly, the SO(2) rotation between two cylindrical feature maps $\mathbf{C}_i^P$, $\mathbf{C}_j^Q$ can be derived by:

$$\mathbf{R}_{ij}^c = \begin{bmatrix} \cos(2\pi d/W) & -\sin(2\pi d/W) & 0 \\ \sin(2\pi d/W) & \cos(2\pi d/W) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

Further, a rotation matrix $\mathbf{R}_{ij} = (\mathbf{R}_j^q)^{\mathrm{T}}\mathbf{R}_{ij}^c\mathbf{R}_i^p$ and a translation vector $\boldsymbol{t}_{ij} = \boldsymbol{q}_j - \boldsymbol{p}_i\mathbf{R}_{ij}^{\mathrm{T}}$ are produced for each pair of correspondence $\{\boldsymbol{p}_i, \boldsymbol{q}_j\}$ for inliers search.

Inliers Search. Since inlier correspondences have similar estimated transformations, it is easy to find them from a number of putative correspondences. Specifically, we first seek the best transformation $\hat{\mathbf{T}} = \{\hat{\mathbf{R}}, \hat{\boldsymbol{t}}\}$ based on the number of correspondences satisfied by each transformation,

$$\{\hat{\mathbf{R}}, \hat{\boldsymbol{t}}\} = \underset{\mathbf{R}_{ij}, \boldsymbol{t}_{ij}}{\arg\max} \sum_{\{\boldsymbol{p},\boldsymbol{q}\}\in\boldsymbol{\Omega}_{init}} \mathbb{1}(\|\boldsymbol{p}\mathbf{R}_{ij}^{\mathrm{T}} + \boldsymbol{t}_{ij} - \boldsymbol{q}\| < \tau), \tag{5}$$

where $\mathbb{1}$ is the indicator function, $\|\cdot\|$ denotes the Euclidean distance, and τ denotes an inlier distance threshold. The inlier correspondences are then obtained,

$$\boldsymbol{\Omega}_{fine} = \Big\{\{\boldsymbol{p}_i, \boldsymbol{q}_j\} \,|\, \Big\|\hat{\mathbf{R}}\boldsymbol{p}_i + \hat{\boldsymbol{t}} - \boldsymbol{q}_j\Big\| < \tau\Big\}. \tag{6}$$

Overall, our Inliers Generator first exploits simple neural layers and general cylindrical features to estimate a coarse rigid transformation for each pair of correspondence, and then searches reliable correspondences according to the transformation similarity between inliers. Notice, our Inliers Generator prunes the outliers from the feature level rather than the correspondence level. Therefore, the proposed Inliers Generator does not contradict existing outlier rejection methods [4, 8, 45] and can also be combined with these methods to estimate a finer rigid transformation.

3.5. Training and Inference

Loss Functions. We train the entire model with a loss $\mathcal{L}$ composed of four terms: $\mathcal{L} = \mathcal{L}_o + \mathcal{L}_f + \mathcal{L}_k + \mathcal{L}_g$. Given a set of ground-truth correspondences $\boldsymbol{\Omega}^* = \{\boldsymbol{p}_i, \boldsymbol{q}_i\}_{i=1\ldots N_c}$ and the ground-truth transformation $\mathbf{T} = \{\mathbf{R}, \boldsymbol{t}\}$, the corresponding orientations $\{\boldsymbol{o}_i^p, \boldsymbol{o}_i^q\}$ can be obtained by the Point-wise Learner. Insipred by the the probabilistic chamfer loss proposed in [33], we define a probabilistic cosine loss $\mathcal{L}_o$ as the learned orientations supervision:

$$\mathcal{L}_o = \frac{1}{N_c}\sum_{i=1}^{N_c}\left(\ln \epsilon_i + \frac{e_i}{\epsilon_i}\right), \quad e_i = 1 - \cos\left(\boldsymbol{o}_i^p \mathbf{R}^{\mathrm{T}}, \boldsymbol{o}_i^q\right) \tag{7}$$

where N_c is the number of ground-truth matched correspondences and ϵ_i is a learnable parameter. Next, we follow D3Feat [5] to leverage the contrastive loss $\mathcal{L}_f$ for feature learning and detection loss $\mathcal{L}_k$ for keypoints detection. To train the proposed Inliers Generator, we first calculate the ground-truth offset d^* between two cylindrical feature maps, and then adopt the L1 loss as the transformation estimation supervision:

$$\mathcal{L}_g = \frac{1}{N_c}\sum_{i=1}^{N_c} | \left\| d_i - d_i^* \right\|_1 . \tag{8}$$

Hypothesis Generation. We find that the RANSAC [18] algorithm is very efficient for the correspondences $\mathbf{\Omega}_{fine}$ with a high inlier rate. In the whole registration framework, the time consumption by RANSAC is almost negligible. Therefore, we perform RANSAC on the point correspondences $\mathbf{\Omega}_{fine}$ to calculate an accurate rigid transformation.

4. Experiments

In this section, we first test the registration performance of our BUFFER when both training and test sets belong to the same domains. Extensive comparative experiments are then performed on unseen domains to evaluate generalizability. Finally, a set of ablation studies are conducted.

4.1. Datasets and Settings

Following [1], we select four datasets, *i.e.,* indoor 3DMatch [66] and 3DLoMatch [26], outdoor KITTI [20] and ETH [49], to evaluate the registration performance of all methods. On both 3DMatch and 3DLoMatch datasets, we use Registration Recall (RR) [66] as our evaluation metrics. On both KITTI and ETH datasets, Relative Translational Error (RTE), Relative Rotation Error (RRE), and Success rate are used as the evaluation metrics [41]. For more details please see the appendix.

Implementation Details. Our BUFFER is implemented with PyTorch. To ensure fairness, we leverage the code and trained models released by the baselines to conduct comparative experiments. All methods are implemented with PyTorch and run on a computer with an Intel Xeon CPU @2.30GHZ and an NVIDIA RTX 3090 GPU. For more details please see the appendix.

4.2. Evaluation on Datasets of Same Domains

Results on Indoor 3DMatch Datasets. We compare the proposed BUFFER with the state-of-the-art methods w.r.t registration recall and running time on both the 3DMatch and 3DLoMatch datasets. As shown in Table 1, our BUFFER achieves the highest registration recall and remarkable computational efficiency on the 3DMatch dataset. Benefiting from the efficient submanifold sparse convolution, FCGF is the quickest approach. However, the registration recall achieved by FCGF is almost the worst among all methods, nearly 8% lower than our BUFFER. On the low-overlap 3DLoMatch dataset, the results of our BUFFER are on par with the state-of-the-art methods. In particular, our BUFFER is the most lightweight method, and is around 35 times faster than the vanilla SpinNet.

Method	*3DMatch*		*3DLoMatch*		#Param.
	RR(%)↑	Time(s)↓	RR(%)↑	Time(s)↓	(M)↓
FCGF [10]	85.1	**0.16**	40.1	**0.16**	8.76
D3Feat [5]	81.6	0.40	37.2	0.40	24.30
Predator [26]	90.5	0.54	62.5	0.54	7.43
YOHO [57]	90.8	3.31	65.2	3.30	12.38
Gedi [48]	85.3	6.65	48.7	6.65	3.11
SpinNet [1]	92.4	7.12	71.6	7.11	1.41
GeoTrans [52]	92.5	0.23	**74.0**	0.23	9.83
RoReg [58]	**93.2**	2.67	71.2	2.64	12.71
Ours	92.9	0.20	71.8	0.20	**0.92**

Table 1. Results on the 3DMatch and 3DLoMatch datasets.

Method	RTE	RRE	Success	Time	#Param.
	(cm)↓	(°)↓	(%)↑	(s)↓	(M)↓
FCGF [10]	13.3	0.31	81.80	**0.18**	8.76
D3Feat [5]	5.55	0.23	97.12	-	14.08
Predator [26]	**5.17**	0.25	96.94	0.69	22.77
SpinNet [1]	5.55	0.24	97.48	14.57	1.41
GeoTrans [52]	7.02	0.23	96.76	0.31	25.50
Ours	5.37	**0.22**	**97.66**	0.30	**0.92**

Table 2. Results on the KITTI odometry dataset.

Results on Outdoor KITTI Datasets. We compare the proposed BUFFER with strong baselines on the KITTI dataset, as shown in Table 2. It is obvious that our BUFFER achieves the highest success rate and is the most lightweight model and highly efficient. Though FCGF is quicker than our method, its registration success rate is significantly lower than our BUFFER by 15.86%. It is also observed all methods spend more time on the KITTI dataset to register point clouds than on the 3DMatch dataset. This is because the outdoor KITTI dataset contains larger scenarios with a higher number of sampling points.

4.3. Generalizing to Unseen Domains

To extensively evaluate the generalizability of the proposed BUFFER on unseen domains, we conduct three groups of experiments following the settings in [1]: generalizations from indoor to outdoor, from outdoor to indoor, and from outdoor to outdoor. In each group of experiments, all methods are trained on one dataset and then directly tested

Method	**RTE** (cm)↓	**RRE** (°)↓	**Success** (%)↑	**Time** (s)↓	**#Param.** (M)↓
FCGF$^\diamond$ [10]	9.98	0.94	61.43	**0.17**	8.76
D3Feat$^\diamond$ [5]	5.54	**0.65**	74.47	0.26	24.30
Predator$^\diamond$ [26]	7.82	0.89	69.00	0.37	7.43
YOHO$^\diamond$ [57]	7.79	1.07	66.90	3.13	12.38
GeoTrans$^\diamond$ [52]	7.97	**0.65**	90.18	0.27	9.83
RoReg$^\diamond$ [58]	7.24	0.99	74.33	2.56	12.71
Gedi† [48]	4.80	0.75	99.24	6.65	3.11
SpinNet† [1]	**3.88**	**0.65**	97.62	7.12	1.41
Ours	5.22	**0.65**	**99.30**	0.26	**0.92**

Table 3. Results of generalization from 3DMatch to ETH. ◇ represents point-wise methods while † denotes patch-wise methods.

Method	**RTE** (cm)↓	**RRE** (°)↓	**Success** (%)↑	**Time** (s)↓	**#Param.** (M)↓
FCGF [10]	14.4	0.49	62.52	**0.15**	8.76
D3Feat [5]	12.6	0.46	69.55	-	24.3
Predator [26]	16.5	1.38	46.13	0.48	7.43
GeoTrans [52]	10.2	0.37	88.47	0.29	9.83
Gedi [48]	**5.99**	**0.28**	95.68	6.31	3.11
SpinNet [1]	7.00	0.33	94.59	14.6	1.41
Ours	8.58	0.30	**95.86**	0.28	**0.92**

Table 4. Results of generalization from 3DMatch to KITTI.

on other unseen datasets.

From Indoor 3DMatch to Outdoor ETH and KITTI. Table 3 and Table 4 list the results of generalization from 3DMatch to ETH and from 3DMatch to KITTI, respectively. It can be noticed that all point-wise methods exhibit a low success rate when being directly generalized to unseen datasets. This is mainly because they adopt a hierarchical network architecture to learn feature descriptors, which is detrimental for generalization [1]. It is also noted that patch-wise methods have excellent generalization ability, but they are very time-consuming, almost an order of magnitude slower than point-wise methods. In contrast, our BUFFER skillfully combines the two methods, which not only achieves the highest success rate across unseen domains, but also is far more efficient than patch-wise methods. Admittedly, our BUFFER is slightly worse than the SpinNet on RTE and RRE, primarily because SpinNet utilizes more sampling points.

From Outdoor KITTI to Indoor 3DMatch. As shown in Table 5, those point-wise methods *i.e.,* FCGF, D3Feat, Predator, and GeoTrans, exhibit poor generalization results due to the large domain gap. It is noticed that our method surpasses SpinNet by 3.6% recall on the 3DMatch dataset, while the performance gap is widened to 11.7% on the 3DLoMatch dataset. This is primarily because the SpinNet can only generate the point correspondences with a lower inlier rate on the low-overlap 3DLoMatch dataset. In contrast, the proposed Inliers Generator can significantly increase the inlier rate, further improving the registration performance of the whole framework. Notably, our BUFFER achieves the highest RR of 91.2% when being directly generalized to the unseen 3DMatch, which even surpasses those strong baselines (such as Predator and YOHO) trained on the 3DMatch. This further demonstrates the strong generalization ability of our BUFFER as well as its potential utility.

Method	*3DMatch* RR(%)↑	Time(s)↓	*3DLoMatch* RR(%)↑	Time(s)↓	#Param. (M)↓
FCGF [10]	19.7	**0.16**	2.26	**0.16**	8.76
D3Feat [5]	53.6	-	11.6	-	14.08
Predator [26]	23.2	0.30	3.31	0.30	22.77
GeoTrans [52]	54.4	0.27	13.8	0.27	25.50
SpinNet [1]	87.6	7.12	52.8	7.12	1.41
Ours	**91.2**	0.24	**64.5**	0.24	**0.92**

Table 5. Results of generalization from KITTI to 3DMatch.

Method	**RTE** (cm)↓	**RRE** (°)↓	**Success** (%)↑	**Time** (s)↓	**#Param.** (M)↓
FCGF [10]	6.13	0.80	39.55	**0.17**	8.76
D3Feat [5]	4.04	0.60	98.18	-	14.08
Predator [26]	7.88	0.87	71.95	0.38	22.77
GeoTrans [52]	8.01	0.89	93.55	0.27	25.50
SpinNet [1]	**3.63**	0.62	99.44	7.10	1.41
Ours	3.85	**0.57**	**99.86**	0.26	**0.92**

Table 6. Results of generalization from KITTI to ETH.

From Outdoor KITTI to Outdoor ETH. As shown in Table 6, compared to the generalization experiments from 3DMatch to ETH, the point-wise methods such as GeoTrans and D3Feat have a significant performance improvement under this experimental setting. This is because both KITTI and ETH datasets merely contain the same SO(2) rotations and the domain gap between the two datasets is not large. Though this generalization experiment is indeed in favor of point-wise methods which are sensitive to rotations and domain gap, our BUFFER still achieves the best success rate.

4.4. Ablation Studies

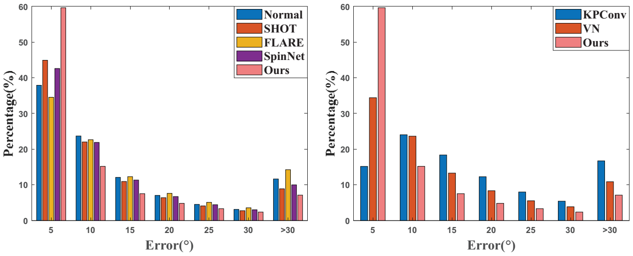
To demonstrate the efficacy of the proposed Equivariant Fully Convolutional Network, we conduct a series of ablative experiments on the 3DMatch dataset. Next, we conduct extensive ablative experiments to systematically evaluate the contribution of each component in our BUFFER.

Ablation of Learned Orientation. To investigate the impact of different settings on the repeatability of orientation, we conduct the following 3 ablation studies.

(1) Replacing our learned orientation by handcrafted methods. In this setting, the orientations are computed by handcrafted methods *i.e.,* normal [11], SHOT [56], FLARE [47], and SpinNet [2].

(2) Replacing the proposed equivariant convolution by KPConv [54]. In this setting, the ablated model is invariant to translations but not equivariant to rotations.

**(3) Replacing the proposed equivariant convolution by



(a) Comparisons with handcrafted methods (b) Comparisons with learned methods

Figure 3. Histogram of the orientation repeatability on the 3DMatch dataset, where the errors denote the deviation angles of estimated orientations between the ground-truth point pairs.

No.	LO	PK	IG	*3DMatch* RR(%)↑	*3DLoMatch* RR(%)↑	*Generalized to ETH* Success(%)↑
1	✗	✗	✗	89.9	66.4	90.74
2	✓	✗	✗	90.1	67.8	93.69
3	✗	✓	✗	90.6	68.9	94.95
4	✓	✓	✗	90.9	69.3	95.93
5	✗	✗	✓	90.9	70.7	97.05
6	✓	✓	✓	**92.9**	**71.8**	**98.88**

Table 7. The quantitative results of all ablated models. Note that, all methods are only trained on the indoor 3DMatch dataset.

vector neuron (VN) [13]. In this setting, the ablated model is equivariant to rotations but not invariant to translations.

Figure 3 shows the quantitative results of the orientation errors of all ablated models on the 3DMatch dataset. It can be seen that: 1) Compared with the handcrafted techniques, our method is more repeatable and robust for real-world point clouds. This is primarily because the proposed EFCN can learn robust deep equivariant features, while hand-crafted methods only rely on low-level geometrical attributes to compute orientations. 2) If the neural network is not equivariant to rotations or invariant to translations, it is hopeless to estimate repeatable and robust orientations. This is reasonable because the network can only memorize orientations brutely, which undoubtedly fails for new data. We can also find that the proposed EFCN is significant for equivariant feature learning and orientation estimation, and has great potential to be extended to more tasks.

Ablation of BUFFER Framwork. Our BUFFER introduces three key components: learned orientation (LO), predicted keypoint (PK), and inliers generator (IG). To investigate the impact of each module, we therefore conduct the following 6 ablation studies to demonstrate the effectiveness of each component. In particular, we train all ablated models on the 3DMatch dataset, and then directly test them on the 3DMatch, 3DLoMatch, and ETH datasets.

Table 7 shows the quantitative results of all ablated networks. We can see that: 1) Without using any of the proposed components, the baseline (Mini-SpinNet [1]) achieves the lowest registration recall on both indoor 3DMatch and 3DLoMatch datasets, and the worst generalization ability on the outdoor ETH dataset. 2) When the proposed LO or PK is utilized (Nos. 2 and 3), the registration accuracy and generalization capability of the entire framework are improved. When both LO and PK are adopted (No. 4), the ablated model significantly surpasses the baseline by 2.9% recall on the 3DLoMatch dataset and 5.19% success rate on the ETH dataset, respectively. This clearly demonstrates that the proposed Point-wise Learner not only can improve the registration accuracy, but also is beneficial to the generalization of the model. 3) When the proposed IG is only employed (No. 5), the registration performance of the whole framework is still greatly improved. This is because the proposed IG can effectively prune a number of outliers from the initial correspondences (the qualitative results are shown in Fig. 4), making it easier to solve for the correct pose in the subsequent hypothesis generation stage.

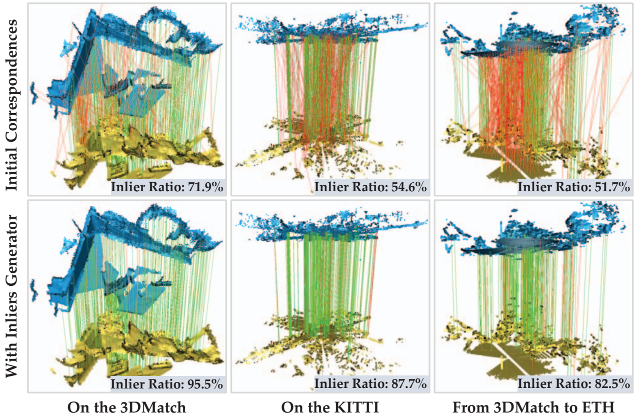


Figure 4. Qualitative results of the ablated model (No. 5), where green lines and red lines denote inliers and outliers.

5. Conclusion

In this paper, we proposed a new deep learning framework termed BUFFER for point cloud registration. The proposed BUFFER introduces an efficient and general feature learning architecture and a correspondence search mechanism. The extensive experiments demonstrate that our method achieves the best trade-off between accuracy, efficiency, and generalizability, outperforming the state-of-the-art by a large margin. In the future, we will investigate the integration of overlap estimation and outlier rejection.

Acknowledgements. This work was partially supported by the National Natural Science Foundation of China (No. U20A20185, 61972435, 62132021), the Guangdong Basic and Applied Basic Research Foundation (2022B1515020103), the Shenzhen Science and Technology Program (No. RCYX20200714114641140), and the Huawei UK AI Fellowship.

References

[1] Sheng Ao, Yulan Guo, Qingyong Hu, Bo Yang, Andrew Markham, and Zengping Chen. You Only Train Once: Learning General and Distinctive 3D Local Descriptors. IEEE TPAMI, 2022. 2, 3, 5, 6, 7, 8, 13

[2] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. SpinNet: Learning a General Surface Descriptor for 3D Point Cloud Registration. In CVPR, 2021. 3, 7

[3] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. PointNetLK: Robust & Efficient Point Clou Registration using PointNet. In CVPR, 2019. 1, 3

[4] Xuyang Bai, Zixin Luo, Lei Zhou, Hongkai Chen, Lei Li, Zeyu Hu, Hongbo Fu, and Chiew-Lan Tai. PointDSC: Robust Point Cloud Registration using Deep Spatial Consistency. In CVPR, 2021. 3, 5, 14

[5] 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. 2, 3, 6, 7, 13

[6] Anh-Quan Cao, Gilles Puy, Alexandre Boulch, and Renaud Marlet. PCAM: Product of Cross-Attention Matrices for Rigid Registration of Point Clouds. In ICCV, 2021. 3

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

[8] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep Global Registration. In CVPR, 2020. 3, 5

[9] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal Convnets: Minkowski Convolutional Neural Networks. In CVPR, 2019. 3

[10] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully Convolutional Geometric Features. In ICCV, 2019. 3, 6, 7, 13

[11] Chin Seng Chua and Ray Jarvis. Point Signatures: A New Representation for 3D Object Recognition. IJCV, 1997. 7

[12] Taco S. Cohen, Mario Geiger, Jonas Koehler, and Max Welling. Spherical CNNs. In ICLR, 2018. 4

[13] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. Vector Neurons: A General Framework for SO(3)-Equivariant Networks. In CVPR, 2021. 4, 8, 11

[14] Haowen Deng, Tolga Birdal, and Slobodan Ilic. PPFNet: Global Context Aware Local Features for Robust 3D Point Matching. In CVPR, 2018. 2

[15] Juan Du, Rui Wang, and Daniel Cremers. DH3D: Deep Hierarchical 3D Descriptors for Robust Large-Scale 6DoF Relocalization. In ECCV, 2020. 3

[16] Carlos Esteves, Yinshuang Xu, Christine Allen-Blanchette, and Kostas Daniilidis. Equivariant Multi-View Networks. In ICCV, 2019. 4

[17] Kai Fischer, Martin Simon, Florian Olsner, Stefan Milz, Horst-Michael Gross, and Patrick Mader. StickyPillars: Robust and Efficient Feature Matching on Point Clouds using Graph Neural Networks. In CVPR, 2021. 3

[18] Martin A Fischler and Robert C Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. COMMUN ACM, 1981. 2, 3, 6, 14

[19] Kexue Fu, Shaolei Liu, Xiaoyuan Luo, and Manning Wang. Robust Point Cloud Registration Framework Based on Deep Graph Matching. In CVPR, 2021. 3

[20] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In CVPR, 2012. 6, 12, 13

[21] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The Perfect Match: 3D Point Cloud Matching with Smoothed Densities. In CVPR, 2019. 2

[22] 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

[23] 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

[24] Sofiane Horache, Jean-Emmanuel Deschaud, and François Goulette. 3D Point Cloud Registration with Multi-Scale Architecture and Self-supervised Fine-tuning. In 3DV, 2021. 2

[25] 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

[26] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. PREDATOR: Registration of 3D Point Clouds with Low Overlap. In CVPR, 2021. 2, 3, 6, 7, 12, 13

[27] 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. 3

[28] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael JL Townshend, and Ron Dror. Learning From Protein Structure with Geometric Vector Perceptrons. ICLR, 2020. 4

[29] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In ICCV, 2017. 5

[30] Marc Khoury, Qian-Yi Zhou, and Vladlen Koltun. Learning Compact Geometric Features. In ICCV, 2017. 1

[31] Junha Lee, Seungwook Kim, Minsu Cho, and Jaesik Park. Deep Hough Voting for Robust Global Registration. In ICCV, 2021. 3

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

[33] Jiaxin Li and Gim Hee Lee. USIP: Unsupervised Stable Interest Point Detection From 3D Point Clouds. In ICCV, 2019. 5, 14

[34] Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. PointNetLK Revisited. In CVPR, 2021. 3

[35] Yang Li and Tatsuya Harada. Lepard: Learning Partial Point Cloud Matching in Rigid and Deformable Scenes. In CVPR, 2022. 3

[36] Ming-Yu Liu, Oncel Tuzel, Ashok Veeraraghavan, Yuichi Taguchi, Tim K Marks, and Rama Chellappa. Fast Object Localization and Pose Estimation in Heavy Clutter for Robotic Bin Picking. IJRR, 2012. 1

[37] Fan Lu, Guang Chen, Yinlong Liu, Zhongnan Qu, and Alois Knoll. RSKDD-Net: Random Sample-based Keypoint Detector and Descriptor. In NeurIPS, 2020. 2

[38] Fan Lu, Guang Chen, Yinlong Liu, Lijun Zhang, Sanqing Qu, Shu Liu, and Rongqi Gu. HRegNet: A Hierarchical Network for Large-scale Outdoor LiDAR Point Cloud Registration. In ICCV, 2021. 3

[39] 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. 3

[40] Bruce D Lucas, Takeo Kanade, et al. An Iterative Image Registration Technique with an Application to Stereo Vision. In DARPA Image Understanding Workshop, 1981. 3

[41] 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

[42] Marlon Marcon, Riccardo Spezialetti, Samuele Salti, and Luigi Di Stefano. Unsupervised Learning of Local Equivariant Descriptors for Point Clouds. IEEE TPAMI, 2021. 2

[43] Taewon Min, Chonghyuk Song, Eunseok Kim, and Inwook Shim. Distinctiveness Oriented Positional Equilibrium for Point Cloud Registration. In ICCV, 2021. 3

[44] Przemyslaw Musialski, Peter Wonka, Daniel G Aliaga, Michael Wimmer, Luc Van Gool, and Werner Purgathofer. A Survey of Urban Reconstruction. CGF, 2013. 1

[45] G Dias Pais, Srikumar Ramalingam, Venu Madhav Govindu, Jacinto C Nascimento, Rama Chellappa, and Pedro Miraldo. 3DRegNet: A Deep Neural Network for 3D Point Registration. In CVPR, 2020. 3, 5

[46] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In NeurIPS, 2017. 3

[47] Alioscia Petrelli and Luigi Di Stefano. On the Repeatability of The Local Reference Frame for Partial Shape Matching. In ICCV, 2011. 7

[48] Fabio Poiesi and Davide Boscaini. Learning General and Distinctive 3D Local Deep Descriptors for Point Cloud Registration. IEEE TPAMI, 2022. 2, 3, 6, 7

[49] François Pomerleau, Ming Liu, Francis Colas, and Roland Siegwart. Challenging Data Sets for Point Cloud Registration Algorithms. IJRR, 2012. 1, 6, 13

[50] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Deep Learning on Point Sets for 3D Classification and Segmentation. In CVPR, 2017. 3

[51] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In NeurIPS, 2017. 2, 3

[52] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric Transformer for Fast and Robust Point Cloud Registration. In CVPR, 2022. 2, 3, 6, 7

[53] Vinit Sarode, Xueqian Li, Hunter Goforth, Yasuhiro Aoki, Rangaprasad Arun Srivatsan, Simon Lucey, and Howie Choset. PCRNet: Point Cloud Registration Network using PointNet Encoding. arXiv preprint arXiv:1908.07906, 2019. 3

[54] 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. 4, 7

[55] Federico Tombari and Luigi Di Stefano. Object Recognition in 3D Scenes with Occlusions and Clutter by Hough Voting. In PSIVT, 2010. 3

[56] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique Signatures of Histograms for Local Surface Description. In ECCV, 2010. 7

[57] Haiping Wang, Yuan Liu, Zhen Dong, Wenping Wang, and Bisheng Yang. You Only Hypothesize Once: Point Cloud Registration with Rotation-equivariant Descriptors. In ACM MM, 2022. 2, 6, 7

[58] Haiping Wang, Yuan Liu, Qingyong Hu, Bing Wang, Jianguo Chen, Zhen Dong, Yulan Guo, Wenping Wang, and Bisheng Yang. RoReg: Pairwise Point Cloud Registration with Oriented Descriptors and Local Rotations. TPAMI, 2023. 6, 7

[59] Yue Wang and Justin M. Solomon. Deep Closest Point: Learning Representations for Point Cloud Registration. In ICCV, 2019. 3

[60] Hao Xu, Shuaicheng Liu, Guangfu Wang, Guanghui Liu, and Bing Zeng. OMNet: Learning Overlapping Mask for Partial-to-Partial Point Cloud Registration. In ICCV, 2021. 1

[61] Hao Xu, Nianjin Ye, Shuaicheng Liu, Guanghui Liu, and Bing Zeng. FINet: Dual Branches Feature Interaction for Partial-to-Partial Point Cloud Registration. In AAAI, 2022. 1

[62] Zi Jian Yew and Gim Hee Lee. RPM-Net: Robust Point Matching using Learned Features. In CVPR, 2020. 3

[63] Zi Jian Yew and Gim Hee Lee. REGTR: End-to-end Point Cloud Correspondences with Transformers. In CVPR, 2022. 1, 13

[64] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. CoFiNet: Reliable Coarse-to-fine Correspondences for Robust Point Cloud Registration. In NeurIPS, 2021. 2, 3

[65] Wentao Yuan, Benjamin Eckart, Kihwan Kim, Varun Jampani, Dieter Fox, and Jan Kautz. DeepGMR: Learning Latent Gaussian Mixture Models for Registration. In ECCV, 2020. 3

[66] 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, 6, 12, 13