DetarNet: Decoupling Translation and Rotation by Siamese Network for Point Cloud Registration

Zhi Chen, Fan Yang, Wenbing Tao*

National Key Laboratory of Science and Technology on Multispectral Information Processing
School of Artifical Intelligence and Automation, Huazhong University of Science and Technology, China
{z_chen, fanyang, wenbingtao}@hust.edu.cn

Abstract

Point cloud registration is a fundamental step for many tasks. In this paper, we propose a neural network named DetarNet to decouple the translation t and rotation R, so as to overcome the performance degradation due to their mutual interference in point cloud registration. First, a Siamese Network based Progressive and Coherent Feature Drift (PCFD) module is proposed to align the source and target points in high-dimensional feature space, and accurately recover translation from the alignment process. Then we propose a Consensus Encoding Unit (CEU) to construct more distinguishable features for a set of putative correspondences. After that, a Spatial and Channel Attention (SCA) block is adopted to build a classification network for finding good correspondences. Finally, the rotation is obtained by Singular Value Decomposition (SVD). In this way, the proposed network decouples the estimation of translation and rotation, resulting in better performance for both of them. Experimental results demonstrate that the proposed DetarNet improves registration performance on both indoor and outdoor scenes. Our code will be available in https://github.com/ZhiChen902/DetarNet.

Introduction

Point cloud registration is one of the fundamental problems in computer vision, which is widely applied to 3D reconstruction, robotics, autonomous driving and medical tasks. It aims to establish correspondences between two point clouds, and estimate the rigid transformation (translation t and rotation R). The most commonly used way is first establishing coarse correspondences, and then recovering rigid transformation. The main challenge is that there always exist wrong correspondences (outliers). Although some methods attempt to generate more accurate correspondences through handcrafted (Rusu, Blodow, and Beetz 2009; Rusu et al. 2008) or deep-learning technique based descriptors (Zhou et al. 2018; Yew and Lee 2018; Choy, Park, and Koltun 2019), it is hard to be totally outlier-free when dealing with complicated scenarios. Thus, it is worth studying how to better perform point cloud registration in the scenarios when the initial correspondences contain outliers.

Recently, some methods have studied how to use a classification neural networks (Pais et al. 2020; Choy, Dong, and Koltun 2020; Bai et al. 2021; Lee et al. 2021) to find correct 3D correspondences between two point clouds, and then estimate the translation t and rotation R by weighted Singular Value Decomposition (SVD) (Choy, Dong, and Koltun 2020). The core of these methods is to learn the consensus (Bai et al. 2021) of correct correspondences (inliers). Each correspondence can be abstracted as an arrow between a pair of points, as shown in Fig. 1. The consensus of inliers is that the length and direction of inliers are satisfied with some consistency. As shown in Fig. 1 (a), due to the coupling of translation and rotation, the consistency between inliers is difficult to be mined. However, as illustrated in Fig. 1 (b), if we can first eliminate the translation t and only the rotation is left, it is much easier for us to find out the correct correspondences. Inspired by this observation, we expect to decouple the whole rigid transformation into separate translation and rotation estimation.

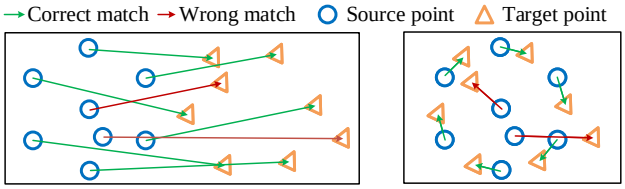

Figure 1: (a): A toy example shows the challenge when translation and rotation are coupled. When accurately estimating translation in advance and there only exists rotation as shown in (b), the consensus between inliers is easier to be mined.

Considering the non-linearity of the rotation space (Peng et al. 2019; Li et al. 2018), it is more feasible to first recover the translation t because it is linear and easy to handle. However, decoupling t and solving it accurately in advance is still challenging in two aspects: 1) It is hard for traditional geometric optimization methods to only recover t without considering R. These methods usually need to jointly optimize t and R. Although centroid alignment (Arun, Huang, and Blostein 1987) can yield a rough t, it can only be used to assist the optimization of the whole rigid transformation due to the existence of outliers. 2) Although deep learning networks have made remarkable progress in point clouds reg-

*Corresponding author.

istration, translation transformation is still hard to be separately modeled in the neural networks while excluding the influence of R. Most of the methods try to regress both t and R (Pais et al. 2020; Aoki et al. 2019) together.

Based on the above analysis, we propose a Siamese Network based Progressive and Coherent Feature Drift (PCFD) module to decouple t from the whole transformation and solve it accurately. PCFD module converts the registration into an alignment process of two point clouds in high-dimensional feature space. First, the features of the two point clouds are respectively extracted by a Siamese Network with shared parameters. Then a global feature offset is learned by establishing global interaction between the two point clouds. The global feature offset forces the source points to move towards the target points coherently as a group to preserve the topological structure of point sets. Thus, the transformation is explicitly encoded by the alignment process, which is named as Coherent Feature Drift (CFD) operation. The whole PCFD module is composed of multiple CFD operations, which progressively align the source and target points to obtain the optimal estimation of t.

The formulation of PCFD module has three advantages: 1) CFD operations explicitly encode the transformation by the global features in the network. Thus, the coupling between R and t can be disentangled by introducing the supervision on the middle layers. We supervise the alignment process by using only the ground truth translation t_{gt}, so that the global features tend to encode the translation transformation. 2) When putative correspondences are given, previous methods usually (Choy, Dong, and Koltun 2020; Pais et al. 2020; Bai et al. 2021) concatenate the two points of a correspondence and form a virtual point to process together. Different from them, our PCFD module adopts a Siamese Network to retain the features of two point clouds. Since the two point clouds are handled respectively, it is easier to establish interaction between them, which benefits the regression for transformation. 3) The network adopts a progressive alignment approach to regress t and gradually eliminates t by using a multi-layer CFD operation. The multiple layers of CFD constitute an iterative optimization structure, so t can be more accurately estimated.

Since we obtain the accurate estimation of t, the correspondences between two point clouds are more obvious and easier to be decided, as shown in Fig. 1 (b). Then we follow the previous works (Moo Yi et al. 2018; Pais et al. 2020) and build a correspondence classification network to prune outliers. Specifically, a Consensus Encoding Unit (CEU) is proposed to remove t when encoding the consensus to make the feature more distinguishable. It combines the spatial and feature consistency items as the feature for each correspondence. Furthermore, we design a Spatial and Channel Attention (SCA) block for the construction of classification network. It simplifies the current spatial attention module (Sun et al. 2020; Chen, Yang, and Tao 2021) and combines it with an instance-unique channel attention. Thus, the network can capture more complex context to better find the consensus of inliers. Finally, according to the established correspondences, R is obtained by Singular Value Decomposition (SVD) (Arun, Huang, and Blostein 1987).

The above modules are integrated into an end-to-end registration network named DetarNet. In a nutshell, our main contributions are threefold: **1.** We propose a Progressive and Coherent Feature Drift (PCFD) module to gradually align the source and target points in feature space. With this process, the t vector can be accurately recovered. **2.** We propose a Consensus Encoding Unit (CEU) to construct a feature for each correspondence and a Spatial and Channel Attention (SCA) block to find correct correspondences. They can establish accurate matches for R estimation. **3.** The above modules are integrated to build decoupling solutions for R and t. Experiments show that the proposed network achieves state-of-the-art performance on both indoor and outdoor datasets.

Related Works

Feature-Based 3D Matching. A common way to establish correspondences between 3D point clouds is by extracting local descriptors. Traditional hand-crafted descriptors are usually generated by extracting the local information, such as histograms of spatial coordinates (Frome et al. 2004; Johnson and Hebert 1999; Tombari, Salti, and Di Stefano 2010) and geometric attributes (Chen and Bhanu 2007; Salti, Tombari, and Di Stefano 2014). Some other methods (Rusu et al. 2008; Rusu, Blodow, and Beetz 2009) aim to design rotation invariant descriptors. Recently, deep learning techniques are explored to learn 3D descriptors. Many of these methods (Su et al. 2015; Zhou et al. 2018; Zeng et al. 2017; Deng, Birdal, and Ilic 2018) take the point cloud patches as input to learn local features. Some other methods (Choy, Gwak, and Savarese 2019; Choy, Park, and Koltun 2019; Yew and Lee 2018; Bai et al. 2020; Huang et al. 2021) use point clouds as input to generate dense feature descriptors on point clouds. Although the methods above can usually establish good initial correspondences, it is hard to be totally outlier-free in the application. Our method is to address the challenge of registration when there are outliers in the correspondences.

Outlier Removal. Given a putative correspondence set that contains outliers, one can use outlier removal methods to remove outliers. The most widely used method is the RANdom SAmple Consensus (RANSAC) (Fischler and Bolles 1981), and its variants (Chum and Matas 2005; Fragoso et al. 2013; Brahmachari and Sarkar 2009; Goshen and Shimshoni 2008).Recently, some methods start adopting deep learning techniques to find good 2D-2D correspondences. The CN-Net (Moo Yi et al. 2018) proposes a Context Normalization (CN) operation for finding correct correspondences. Later works (Plötz and Roth 2018; Zhang et al. 2019; Zhao et al. 2019; Sun et al. 2020; Brachmann and Rother 2019; Liu et al. 2021) aim to capture more context to enhance the performance. Besides, recent attempts try to use deep learning networks for finding 3D correspondences, such as 3DRegNet (Pais et al. 2020), DGR-Net (Choy, Dong, and Koltun 2020) and PointDSC (Bai et al. 2021). Our work aims to better find correct 3D correspondences and align point clouds through decoupling translation and rotation transformations.

Pose Estimation. Pose estimation is the final goal of rigid 3D point registration, i.e., estimating a rigid transformation

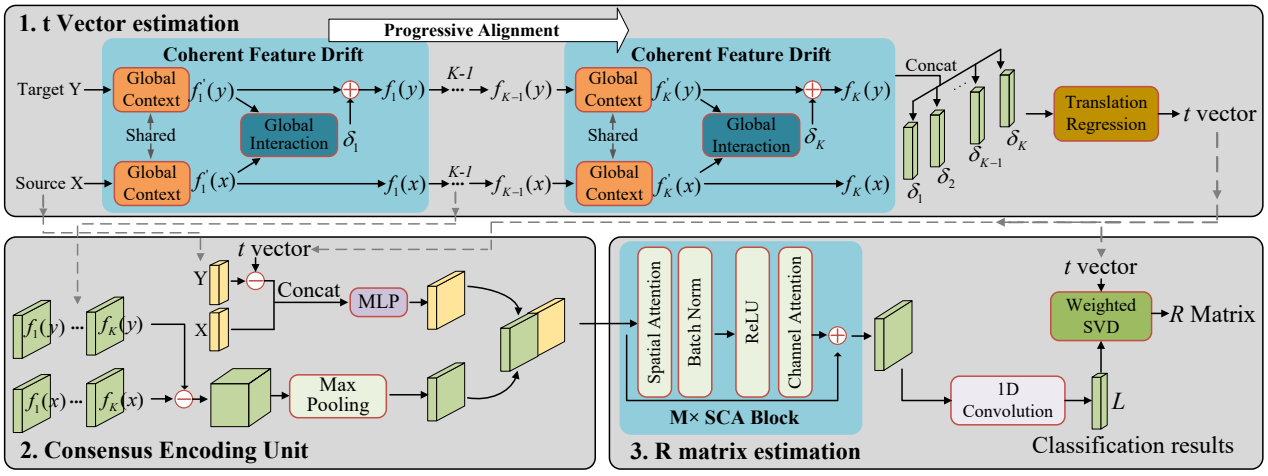

Figure 2: Overview of our network. 1. A Progressive and Coherent Feature Drift (PCFD) module progressively aligns the source and target points in feature space, and recovers t vector from the alignment process. 2. An Consensus Encoding Unit (CEU) constructs feature for each correspondence by combining the spatial and feature consistency. 3. Several Spatial and Channel Attention (SCA) blocks are adopted to find correct correspondences, and followed with weighted SVD for estimating R matrix.

to align point clouds. Besl and McKay (Besl and McKay 1992) propose the iterative closest point (ICP) algorithm to align point cloud through iteratively establishing point correspondence and performing least squares optimization. The variants of ICP (Rusinkiewicz and Levoy 2001; Segal, Haehnel, and Thrun 2009; Bouaziz, Tagliasacchi, and Pauly 2013) are proposed to address the challenges existing in ICP, such as efficiency, partiality and sparsity. Recently, some methods adopt end-to-end frameworks for directly estimating the rigid transformation between point clouds. Deep Closest Point (DCP-Net) (Wang and Solomon 2019a) uses deep global features to form correspondences and estimate relative pose. Later works (Yew and Lee 2020; Wang and Solomon 2019b) aim to address the problem of partial visibility to further improve the performance of registration.

Methods

Given two point clouds to be registered: $X = \{x_i \in \mathbb{R}^3 \mid i = 1, ..., N_x\}$ and $Y = \{x_j \in \mathbb{R}^3 \mid j = 1, ..., N_y\}$, we first form N pairs of correspondences as follows:

$$C = \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & \dots & y_n \end{bmatrix} \in \mathbb{R}^{2\times N\times 3}, \quad (1)$$

where x_i and y_i $(1 < i < N)$ are a pair of matched points. These putative correspondences are established by extracting local descriptors and matching. Limited by the distinctiveness of descriptors, many of these correspondences are wrong (outliers) . The goal of the network is to recover the rigid transformation from these noisy correspondences. It takes the coordinates of these correspondences as input, and outputs the probability of being correct (inliers) for each correspondence and rigid transformation as follows:

$$t,\ R,\ L = \Phi(C); t \in \mathbb{R}^3, R \in \mathbb{R}^{3\times 3}, L \in \mathbb{R}^{N\times 1}, \quad (2)$$

where $\Phi(\cdot)$ is the network with trained parameters. t and R are the estimated translation and rotation respectively. L is the logit value of each correspondence being inlier. In this paper, we propose a decoupling solution for the t and R. The pipeline of our method is shown in Fig. 2. We will explain the details of each module in the following sections.

Translation Estimation

Progressive and Coherent Feature Drift. The PCFD module transforms point cloud registration into a process of coherently moving source points to target points. Since deep neural network can extract more informative feature for each point, we convert the coherent drift operation to high-dimensional feature space. As shown in Fig. 2, the PCFD module is composed of K Coherent Feature Drift (CFD) operations. Each CFD tries to align the features of source and target points generated by the previous CFD layer, so it is a progressive process.

Specifically, the CFD first encodes feature for each point in a Siamese architecture. We use the CN Block (Moo Yi et al. 2018), which is a variant of PointNet (Qi et al. 2017), for encoding global context. Formally, let $f_{l-1}(x)$ and $f_{l-1}(y)$ be the output of $l-1$ layer, then l-th CFD encodes the features as follows:

$$f_l^{'}(x) = \mathrm{CN}(f_{l-1}(x)), f_l^{'}(y) = \mathrm{CN}(f_{l-1}(y)), \quad (3)$$

where $f_l^{'}(x)$ and $f_l^{'}(y)$ are the extracted features for source and target points. Note that the CN operations for source and target points are parameter-shared. In this way, the feature difference between a pair of correspondences is completely caused by the rigid transformation between them. Then we perform coherent drift by moving the features of source points to target points. A core of coherent drift is to force the source points to move coherently as a group to preserve the topological structure of the point sets (Myronenko and Song 2010). To ensure coherent constraints, a global feature offset $(\delta_l, 1 \le l \le K)$ shared by all the source points is learned the

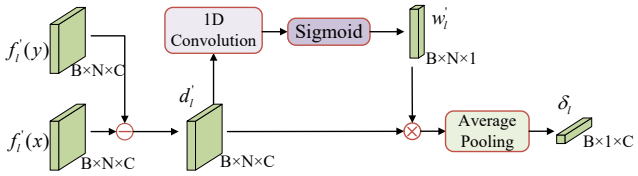

Figure 3: The global interaction operation.

l-th CFD. After that, we hold the target points and move the source points are moved by the δ_l to the target points.

$$f_l(x_i) = f_l^{'}(x) + \delta_l, f_l(y_i) = f_l^{'}(y_i); 1 \leq i \leq N, \quad (4)$$

where x_i and y_i are the i-th point in the source and target points. $f_l(x_i)$ and $f_l(y_i)$ are the output feature.

An important issue of CFD is how to learn the global feature offset δ_l. Each δ_l needs to make the source points gradually approach the target points in the feature space. In the CFD operation, a global interaction is adopted to learn it as in Fig. 3. It first computes the feature difference between the feature of source and target points, as follows:

$$d_l^{'} = f_l^{'}(y) - f_l^{'}(x). \quad (5)$$

As mentioned before, x_i and y_i are a pair of putative correspondences. So $d_l^{'}$ is the feature offset between putative correspondences. We then learn a weight ($w_l^{'}$) for each correspondence by a convolution and sigmoid function:

$$w_l^{'} = sigmoid(Conv(d_l^{'})) \quad (6)$$

Finally, we use an average pooling to integrate the feature difference of all correspondences to produce the global feature offset δ_l. The 1D convolution plays two roles in the learning of δ_l: 1) Since there are many outliers in the putative correspondences, the weights produced by the convolution and sigmoid function is expected to suppress the outliers. 2) There are learnable parameters in the convolution operation to increase the flexibility of global interaction.

Supervising Drift Process. In order to reduce the interference of R for better encoding the translation transformation, we introduce supervision in the middle of the network. Supervision is applied to the global feature offset δ_l ($1 < l < N$). Specifically, we first add up all the previous offsets, then use a 1D convolution to regress a temporary $t_l \in \mathbb{R}^3$ vector as follows:

$$t_l = Conv(Sum(\delta_1, ..., \delta_{l-1}, \delta_l)). \quad (7)$$

Then a drift loss is used to supervise all layers of t_l:

$$\mathcal{L}_{align} = \frac{1}{K}\sum_{l=1}^{K}\{\frac{1}{N}\sum_{i=1}^{N} m_i \cdot \rho(y_i, R_{gt}x_i + t_l)\}, \quad (8)$$

where $\rho(.,.)$ is the distance metric function. R_{gt} is the ground truth rotation. m_i is the mask for correspondence i. m_i is set to 1 if the ground-truth of correspondence i is inliers. Otherwise, it will be set to be 0. It is a semi-alignment loss that uses the estimated translation t_l in l-th layer and ground-truth rotation R_{gt} to align the two point cloud and penalizes the alignment error. Thus, it expects all alignment to approximate the accurate t vector.

Translation Regression. As mentioned before, every time a CFD is performed, the source point cloud is globally aligned to the target point cloud by the global feature offset, and δ_l encodes the alignment process. We can naturally solve the t matrix by integrating the offsets of all layers. We concatenate all of the δ_l ($1 \leq l \leq K$) and then adopt a 1D convolution to regress t vector.

Consensus Encoding Unit

An important issue for finding correct correspondences from putative correspondences is to mine consensus of inliers (Pais et al. 2020; Choy, Dong, and Koltun 2020; Bai et al. 2021), so that outliers can be distinguished from inliers. As introduced in Introduction Section and Fig. 1, it becomes easier to mine the consensus when removing translation t and remaining only rotation R between two point clouds. Since our PCFD module can regress t vector in advance, the Consensus Encoding Unit (CEU) tries to remove translation for better encoding consensus. The architecture of CEU is shown in Fig. 2, it combines the consensuses in coordinate and feature space.

For the consensus in coordinate space, it is intuitive to remove the t vector. We subtract the estimated t vector from the target point cloud, so that the translation t between the source point cloud and the target point cloud is removed. Then the coordinate offset between the source point cloud and the target point cloud is followed with a 1D convolution to be as a feature.

Meanwhile, CEU also tries to mine feature consistency between the correct correspondences. It utilizes the feature produced by the previous PCFD module to construct feature for correspondence to integrate more information. As introduced before, in PCFD module, source points are aligned to target points in feature space. By introducing the supervision of the intermediate layer, the t transformation between the source points and the target points is removed in feature space. Thus, we use the feature difference of these layers to construct the feature for the correspondence, which can encode consensus without translation. In order to make full use of the context of shallow and deep networks, all the layers of PCFD module are used to form a multi-layer correlation feature. Then a max-pooling, which performs the best with other choices based on our experiments, is adopted to integrate multi-layer context. The consensus feature in coordinate and feature space is combined by a concatenation operation.

Rotation Estimation

After Consensus Encoding Unit constructs a feature for each correspondence, a classification network is adopted to finding correct correspondences, and followed with weighted SVD to recover R matrix.

Classification Network. As shown in Fig. 2, the classification network is composed of M Spatial and Channel Attention (SCA) Blocks. Each SCA block integrates a spatial attention, batch normalization (Ioffe and Szegedy 2015), ReLU and a channel attention in a ResNet architecture. The spatial attention is already proposed for finding 2D-2D corre-

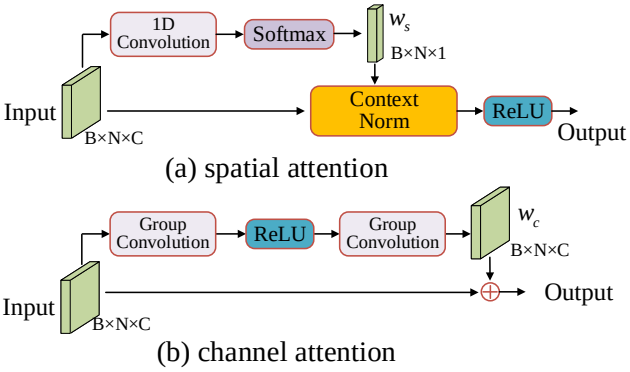

Figure 4: The Spatial Attention and Channel Attention in SCA block.

spondences (Sun et al. 2020; Chen, Yang, and Tao 2021) operation. They integrate local, global and prior information to learn weight for learning global context to ignore outliers. In this paper, since the CEU can construct more distinguishable features, we simplified the spatial attention to reduce the parameters of the network as Fig. 4 (a). For the input feature, it first learns a weight vector ($w_s \in \mathbb{R}^{B\times N\times 1}$) by means of a 1D convolution and Softmax function. Then the weight vector is utilized as guidance to perform a weighted context normalization (Moo Yi et al. 2018) for encoding global context. The weight vector is to allow outliers to be ignored when performing context normalization. Meanwhile, interdependencies between feature channels are proved to be helpful for feature learning (Hu, Shen, and Sun 2018). So we also introduce a channel attention operation as Fig. 4 (b). Different from the commonly used SE-Net, we use instance-unique channel attention. It learns an independent weight vector for each instance instead of a shared one. Thus, it can capture more complex channel information for each correspondence. In order to reduce network computation for learning the instance-unique weight map, the group convolution (Cohen and Welling 2016) is used instead of the regular one.

Weighted SVD. We use weighted SVD (Choy, Dong, and Koltun 2020), which reformulates the traditional SVD (Arun, Huang, and Blostein 1987) into a weighted version, to recover R matrix. Specifically, x_i and y_i $(1 \le i \le N)$ are the points in source and target point clouds respectively. We first use the estimated t vector to process the target points:

$$y_i^{'} = y_i - t, 1 \le i \le N. \tag{9}$$

Then a weighted matrix H for SVD is computed as follows:

$$H = \sum_{i\in\mathcal{I}} w_i x_i y_i^{'T}, H \in \mathbb{R}^{3\times 3}, \tag{10}$$

where the weight w_i is computed by the logit value of classification as follows:

$$w_i = \tanh(\mathrm{ReLU}(L_i)), 1 \le i \le N, \tag{11}$$

Finally, R can be obtained by performing SVD on H matrix as follows:

$$R = Udiag(1, 1, det(UV^T))V^T, H = U\sum V^T. \tag{12}$$

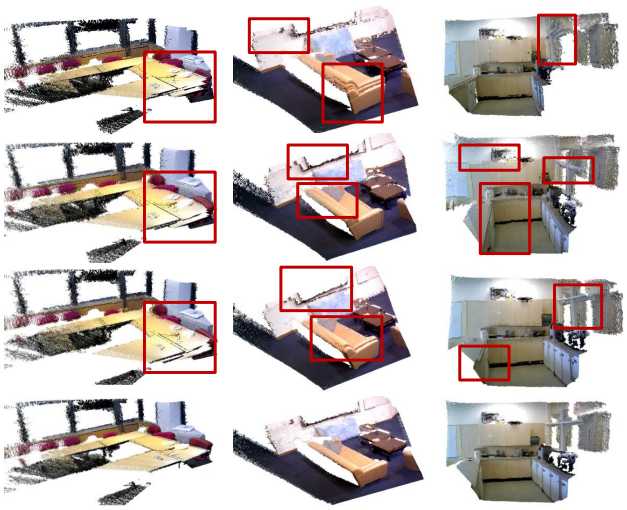

Figure 5: The visualized alignment results of four different methods. From top row to bottom: 3DRegNet (Pais et al. 2020), RANSAC (Fischler and Bolles 1981), FGR (Zhou, Park, and Koltun 2016) and ours. The alignment areas with large errors are marked with red boxes. Our method achieves the best alignment result among these methods.

Loss Function

We formulate our training objective as a combination of four types of loss functions, including translation loss (l_{trans}), classification loss (l_{cls}), alignment loss (l_{align}) and drift loss (l_{drift}) as follows:

$$loss = \lambda_1 l_{trans} + \lambda_2 l_{cls} + \lambda_3 l_{align} + \lambda_4 l_{drift} \tag{13}$$

l_{trans} is the L2 loss between the ground-truth and estimated t vector. l_{cls} is the cross entropy loss. l_{align} penalizes the wrong alignment between correct correspondences as follows:

$$l_{align} = \frac{1}{N}\sum_{i=1}^{N} m_i \cdot \rho(y_i, Rx_i + t), \tag{14}$$

where $\rho(.,.)$ is Euclidean distance. t and R are the estimated translation and rotation transformation. N is the number of correspondences. m_i is also the mask for correspondence i to label inliers, as introduced in Eq. 8. l_{drift} is to supervise the middle layer as Eq. 8.

Experiments

Experimental Setup

Outdoor Dataset. We use the KITTI (Geiger, Lenz, and Urtasun 2012) odometry dataset, which contains 11 outdoor driving scenarios of points clouds. We follow the splitting way of previous works (Bai et al. 2020; Choy, Park, and Koltun 2019) and use scenario 0 to 5 for training, 6 to 7 for validation and 8 to 10 for testing. Then for each point cloud, we construct 30cm voxel grid to downsample the point cloud (Choy, Park, and Koltun 2019).

Indoor Datasets. We use the SUN3D dataset (Xiao, Owens, and Torralba 2013) to generate the dataset for training

Table 1: Quantitative results on the KITTI, SUN3D and 7Scenes Datasets. The mean rotation error (MRE), mean translation error (MTE), mAP and recall under the threshold of (5°, 15 cm) are reported.

	KITTI				SUN3D				7Scenes (Generalization)				
	MRE	MTE	mAP	recall	MRE	MTE	mAP	recall	MRE	MTE	mAP	recall	Time
ICP	1.208	90.21	0.54	1.41	6.178	15.40	19.6	49.7	5.504	12.44	27.8	50.3	0.28
RANSAC	0.759	29.65	43.1	80.9	3.580	15.16	43.9	80.5	2.107	12.41	32.2	68.5	2.79
GCRANSAC	0.152	70.41	1.89	3.12	1.920	9.672	37.6	75.6	1.946	10.43	30.7	78.1	0.82
FGR	0.298	12.13	31.4	72.0	2.895	10.85	39.2	73.6	2.913	13.52	31.5	66.3	0.32
DGR	0.157	9.773	41.6	82.0	2.239	9.663	41.3	82.8	2.166	13.54	30.0	63.4	0.76
PointLK	5.352	43.84	4.01	7.25	7.732	27.64	17.2	32.0	26.49	32.37	5.12	9.61	0.13
PointDSC	0.152	8.966	46.9	**91.5**	1.913	7.283	50.1	89.7	**1.902**	11.31	38.6	78.4	0.09
3DRegNet	0.752	31.62	12.3	28.7	2.889	13.13	31.2	68.6	13.58	23.37	20.1	58.3	0.03
Ours	**0.148**	**8.126**	**48.1**	88.1	**1.840**	**5.317**	**56.3**	**93.1**	2.011	**7.739**	**42.7**	**84.9**	0.04

and testing. Sun3D is composed of 268 sequences of RGB-D videos. We randomly select 115 sequences for training and validation, and 20 sequences for testing. For each video sequence, we first subsample the videos by a factor of 10. Then for each frame, we recover the point cloud by depth map, and construct 5cm voxel grid to downsample the point cloud (Zhou, Park, and Koltun 2018). The 7scenes (Shotton et al. 2013) dataset contains 46 RGBD sequences under various camera motion statuses, we follow the official split to use the 18 sequences of them as test dataset. It is adopted for generalization experiments.

Data Processing. Following 3DReg-Net (Pais et al. 2020), we use FPFH descriptors (Rusu, Blodow, and Beetz 2009) to generate 2560 pairs of correspondences between adjacent frames as input. Then we generate the ground-truth rotation and translation according to the offered camera pose of each frame and label the correspondences as inlier/outlier (1 refers inliers and 0 refers outliers) by a predefined distance threshold.

Evaluation Metrics. For a pair of point clouds, we evaluate the results by computing the errors between the estimated and ground-truth rigid transformation. The errors of rotation (RE) are evaluated by the isotropic error (Ma et al. 2012). The errors of translation (TE) are evaluated by the $L2$ error (Choy, Dong, and Koltun 2020). For the whole test dataset, we first report the mean of rotation (MRE) and translation (MTE) errors. Then, given an error threshold of R and t, we can determine whether each estimated pose is accurate or not. We build a normalized cumulative precision curve of pose estimation in the whole test set. After that, we use (5°, 15 cm) as threshold to figure the recall (Choy, Dong, and Koltun 2020) and the area under the curve as mean average precision (mAP) (Moo Yi et al. 2018).

Implementation Details. In the PCFD module, we use 10 layers of CFD to progressively align the two point clouds (K = 10 in Fig. 2). In the classification module, 4 SCA blocks are utilized to build classification network (M = 4 in Fig. 2). The number of channels in all layers of the network is set to 128. During training, λ_1, λ_2, λ_3 and λ_4 in loss function (Eq. 13) are set to 2, 1, 1 and 0.05 respectively. The network is trained by Adam optimizer (Kingma and Ba 2015) with a learning rate being 10^{-3} and batch size being 16. All the experiments are conducted on a machine with an INTEL Xeon E5-2620 CPU and a single NVIDIA GTX1080Ti. For time-consuming, to do a fair comparison for all the methods, all computation timings are obtained using CPU.

Comparison to Other Baselines

We compare our method with other baselines, including ICP (Besl and McKay 1992), FGR (Zhou, Park, and Koltun 2016), RANSAC (Fischler and Bolles 1981), GCRANSAC (Barath and Matas 2018), DGR (Choy, Dong, and Koltun 2020), PointLK (Aoki et al. 2019), PointDSC (Bai et al. 2021) and 3DRegNet (Pais et al. 2020). ICP, FGR, RANSAC and GCRANSAC are classical methods while DGR, PointLK, PointDSC and 3DRegNet are learning based methods. All the learning based networks are retrained with the same training data. For ICP, RANSAC and FGR, we use the version Open3D implemented, while the released codes are adopted for other methods. We present the quantitative results on the KITTI, SUN3D and 7Scenes Datasets. The results on 7Scenes dataset is obtained by the model trained on SUN3D dataset as generalization experiments. As shown in Tab. 1, the recall and mAP of our method are higher than other methods. It shows the overall performance of our methods. More specifically, the t error of our method is much smaller than other methods, especially on indoor scenes. The t error of our method is 7.81cm and 15.63cm smaller than that of our baseline network (3DRegNet) on SUN3D and 7Scenes datasets. It implies that the proposed Progressive and Coherent Feature Drift (PCFD) module can boost the performance of t estimation. For time-consuming, since our network can output the results without repeated sampling as RANSAC (Fischler and Bolles 1981) and post-processing, it is faster than other methods except for 3DRegNet.

Finally, in order to visually demonstrate the registration performance, we present the visualized alignment results in Fig. 5. We select multiple point clouds and calculate the relative pose between each point cloud and its neighbor. Then we transform these point clouds into the same coordinate frame. The results of 3DRegNet (Pais et al. 2020), RANSAC (Fischler and Bolles 1981), FGR (Zhou, Park, and Koltun 2016) are presented as comparison. Our method achieves the best alignment results with fewer errors.

Registration Robustness

So far, we have demonstrated the overall performance of the proposed network. In order to further analyze the registra-

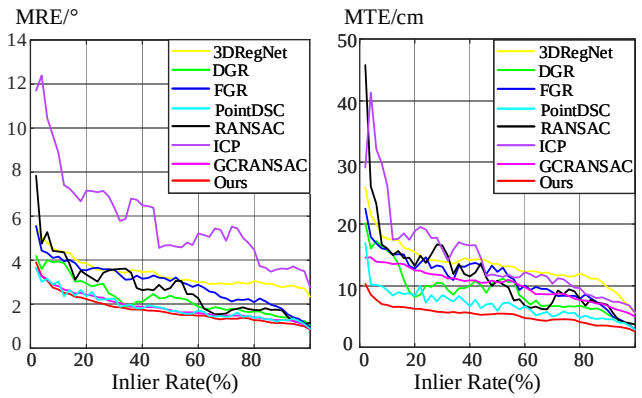

Figure 6: The error curves of R (MRE) and t (MTE) under different inlier ratios of initial correspondences on SUN3D dataset.

tion robustness anti-noise, we test the performance under the scenarios with the different inlier ratios of initial correspondence set. Specifically, we divide the test set of SUN3D dataset into several subsets according to the inlier ratio, and respectively compute the mean errors of R (MRE) and t (MTE) estimation at each inlier ratio, as shown in Fig. 6. As we can see, our method has obtained results with smaller errors under the scenarios of different inlier ratios for both R and t. It demonstrates that our method is robust to outliers. Besides, when the inlier ratio changes, the error range of our method is also smaller, which shows that the performance of our method is relatively stable.

Method Analysis

In this section, we will analyze our method in detail. Expect the previously introduced four evaluation metrics, we also report the classification accuracy (Acc in Tab. 2 and 3) for better understanding the effect of each module.

Regression or SVD - Tab. 2. In our network, t is estimated by regression while R is solved by SVD. We discuss these two estimation heads for t and R. The regression and weighted SVD are adopted as the estimation heads for t and R, respectively. Through permutation and combination, we can generate four alternatives. For each alternative, we use the proposed network as the backbone for feature extraction. By analyzing these four groups of control experiments, we can get the following observations: 1) When the estimation heads of R are consistent, direct regression will get better results than the SVD method for t estimation. When the estimated head of t is the same, the R result obtained by SVD is better. This proves that regressing t in advance, which adopted in our method is a good choice. 2) We further compare the results of the 2-th group and the 4-th group (ours). We can find that our method obtains more accurate t. Meanwhile, although the 2-th and the 4-th group use the same estimation head for R estimation, the 4-th group still achieves better results for R estimation. This can be explained as our method can estimate and remove t in advance before finding the correspondence. Thus, there exists only rotation between correspondence, which helps the classification of inliers and outliers. In fact, the 4-th group dose achieve a better classification result than that of the 2-th group. The above results prove the effectiveness of the idea of decoupling t and R.

Table 2: The registration result of using different estimation heads.

Tag	t	R	Acc	MRE	MTE	mAP	recall
1	Reg	Reg	65.7	2.45	6.22	49.7	86.5
2	SVD	SVD	69.1	2.29	9.17	43.2	80.0
3	SVD	Reg	60.2	2.69	9.01	35.7	62.3
4	Reg	SVD	**72.9**	**1.84**	**5.32**	**56.3**	**93.1**

Table 3: Ablation studies of proposed modules.

Baseline	PCDF	CEU	SCA	Acc	MRE	MTE	mAP	recall
✓				63.2	2.66	13.1	33.1	70.2
✓	✓			60.0	2.92	6.02	41.7	78.8
✓	✓	✓		70.1	2.02	5.99	52.1	90.5
✓	✓	✓	✓	**72.9**	**1.84**	**5.32**	**56.3**	**93.1**

Ablation Study - Tab. 3. Finally, we perform ablation studies on SUN3D dataset to further analyze the effect of the proposed modules, including Progressive and Coherent Feature Drift (PCFD), Consensus Encoding Unit (CEU) and Spatial and Channel Attention (SCA) block. The 3DRegNet (Pais et al. 2020) is adopted as our baseline model. Since we have already proved that regression for t and SVD for R is the most suitable combination of estimation head, we use 3DRegNet with this alternative instead of the vanilla version. We gradually add the proposed modules into the baseline model. First, we use the PCDF module to replace the CN Blocks (Moo Yi et al. 2018) of 3DRegNet. The error of t estimation significantly has decreased, which confirms the effectiveness of PCDF for regressing t. Then we adopt CEU to construct features for correspondence classification. As we can see, the classification accuracy is improved by 10% compared with only using PCFD, leading to a better result of R. It shows that the proposed CEU can construct better classification features. Finally, we replace the CN blocks in 3DRegNet with SCA blocks, the performance of correspondences classification and R estimation are further enhanced.

Conclusion

In this work, we develop a point cloud registration network named DetarNet, which decouples the estimation of rotation and translation. Specifically, we first propose a Progressive and Coherent Feature Drift (PCFD) module. It transforms the point cloud alignment process into a coherent drift operation in high-dimensional feature space and gradually estimates the translation. Then, we adopt a classification module to perform outlier pruning. It uses the proposed Consensus Encoding Unit (CEU) to construct feature for each correspondence, and adopts a Spatial and Channel Attention (SCA) for classification. Thus, the network can establish correct matches by taking advantages of the estimated t. Finally, R matrix is obtained by performing weighted SVD. Extensive experiments on real scenes demonstrate the effectiveness of the proposed DetarNet.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants 62176096 and 61991412.

References

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

Arun, K. S.; Huang, T. S.; and Blostein, S. D. 1987. Least-squares fitting of two 3-D point sets. *IEEE Transactions on pattern analysis and machine intelligence*, (5): 698–700.

Bai, X.; Luo, Z.; Zhou, L.; Chen, H.; Li, L.; Hu, Z.; Fu, H.; and Tai, C.-L. 2021. PointDSC: Robust Point Cloud Registration using Deep Spatial Consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15859–15869.

Bai, X.; Luo, Z.; Zhou, L.; Fu, H.; Quan, L.; and Tai, C.-L. 2020. D3Feat: Joint Learning of Dense Detection and Description of 3D Local Features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6359–6367.

Barath, D.; and Matas, J. 2018. Graph-cut RANSAC. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6733–6741.

Besl, P. J.; and McKay, N. D. 1992. Method for registration of 3-D shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, 586–606. International Society for Optics and Photonics.

Bouaziz, S.; Tagliasacchi, A.; and Pauly, M. 2013. Sparse iterative closest point. In *Computer graphics forum*, volume 32, 113–123. Wiley Online Library.

Brachmann, E.; and Rother, C. 2019. Neural-guided RANSAC: Learning where to sample model hypotheses. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4322–4331.

Brahmachari, A. S.; and Sarkar, S. 2009. BLOGS: Balanced local and global search for non-degenerate two view epipolar geometry. In *2009 IEEE 12th International Conference on Computer Vision*, 1685–1692. IEEE.

Chen, H.; and Bhanu, B. 2007. 3D free-form object recognition in range images using local surface patches. *Pattern Recognition Letters*, 28(10): 1252–1262.

Chen, Z.; Yang, F.; and Tao, W. 2021. Cascade Network with Guided Loss and Hybrid Attention for Finding Good Correspondences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 1123–1131.

Choy, C.; Dong, W.; and Koltun, V. 2020. Deep Global Registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2514–2523.

Choy, C.; Gwak, J.; and Savarese, S. 2019. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3075–3084.

Choy, C.; Park, J.; and Koltun, V. 2019. Fully convolutional geometric features. In *Proceedings of the IEEE International Conference on Computer Vision*, 8958–8966.

Chum, O.; and Matas, J. 2005. Matching with PROSAC-progressive sample consensus. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, 220–226. IEEE.

Cohen, T.; and Welling, M. 2016. Group equivariant convolutional networks. In *International conference on machine learning*, 2990–2999.

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

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

Fragoso, V.; Sen, P.; Rodriguez, S.; and Turk, M. 2013. EVSAC: accelerating hypotheses generation by modeling matching scores with extreme value theory. In *Proceedings of the IEEE International Conference on Computer Vision*, 2472–2479.

Frome, A.; Huber, D.; Kolluri, R.; Bülow, T.; and Malik, J. 2004. Recognizing objects in range data using regional point descriptors. In *European conference on computer vision*, 224–237. Springer.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3354–3361. IEEE.

Goshen, L.; and Shimshoni, I. 2008. Balanced exploration and exploitation model search for efficient epipolar geometry estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7): 1230–1242.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141.

Huang, S.; Gojcic, Z.; Usvyatsov, M.; Wieser, A.; and Schindler, K. 2021. PREDATOR: Registration of 3D Point Clouds with Low Overlap. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4267–4276.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Johnson, A. E.; and Hebert, M. 1999. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on pattern analysis and machine intelligence*, 21(5): 433–449.

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

Lee, J.; Kim, S.; Cho, M.; and Park, J. 2021. Deep Hough Voting for Robust Global Registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15994–16003.

Li, Y.; Wang, G.; Ji, X.; Xiang, Y.; and Fox, D. 2018. DeepIM: Deep Iterative Matching for 6D Pose Estimation. In *European Conference on Computer Vision (ECCV)*.

Liu, Y.; Liu, L.; Lin, C.; Dong, Z.; and Wang, W. 2021. Learnable Motion Coherence for Correspondence Pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3237–3246.

Ma, Y.; Soatto, S.; Kosecka, J.; and Sastry, S. S. 2012. *An invitation to 3-d vision: from images to geometric models*, volume 26. Springer Science & Business Media.

Moo Yi, K.; Trulls, E.; Ono, Y.; Lepetit, V.; Salzmann, M.; and Fua, P. 2018. Learning to find good correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2666–2674.

Myronenko, A.; and Song, X. 2010. Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12): 2262–2275.

Pais, G. D.; Ramalingam, S.; Govindu, V. M.; Nascimento, J. C.; Chellappa, R.; and Miraldo, P. 2020. 3DRegNet: A deep neural network for 3D point registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7193–7203.

Peng, S.; Liu, Y.; Huang, Q.; Zhou, X.; and Bao, H. 2019. PVNet: Pixel-wise Voting Network for 6DoF Pose Estimation. In *CVPR*.

Plötz, T.; and Roth, S. 2018. Neural nearest neighbors networks. In *Advances in Neural Information Processing Systems*, 1087–1098.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Rusinkiewicz, S.; and Levoy, M. 2001. Efficient variants of the ICP algorithm. In *Proceedings third international conference on 3-D digital imaging and modeling*, 145–152. IEEE.

Rusu, R. B.; Blodow, N.; and Beetz, M. 2009. Fast point feature histograms (FPFH) for 3D registration. In *2009 IEEE international conference on robotics and automation*, 3212–3217. IEEE.

Rusu, R. B.; Marton, Z. C.; Blodow, N.; and Beetz, M. 2008. Persistent point feature histograms for 3D point clouds. In *Proc 10th Int Conf Intel Autonomous Syst (IAS-10), Baden-Baden, Germany*, 119–128.

Salti, S.; Tombari, F.; and Di Stefano, L. 2014. SHOT: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125: 251–264.

Segal, A.; Haehnel, D.; and Thrun, S. 2009. Generalized-icp. In *Robotics: science and systems*, volume 2, 435. Seattle, WA.

Shotton, J.; Glocker, B.; Zach, C.; Izadi, S.; Criminisi, A.; and Fitzgibbon, A. 2013. Scene coordinate regression forests for camera relocalization in RGB-D images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2930–2937.

Su, H.; Maji, S.; Kalogerakis, E.; and Learned-Miller, E. 2015. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, 945–953.

Sun, W.; Jiang, W.; Trulls, E.; Tagliasacchi, A.; and Yi, K. M. 2020. Acne: Attentive context normalization for robust permutation-equivariant learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11286–11295.

Tombari, F.; Salti, S.; and Di Stefano, L. 2010. Unique shape context for 3D data description. In *Proceedings of the ACM workshop on 3D object retrieval*, 57–62.

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

Wang, Y.; and Solomon, J. M. 2019b. PRNet: Self-supervised learning for partial-to-partial registration. In *Advances in Neural Information Processing Systems*, 8814–8826.

Xiao, J.; Owens, A.; and Torralba, A. 2013. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of the IEEE international conference on computer vision*, 1625–1632.

Yew, Z. J.; and Lee, G. H. 2018. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In *European Conference on Computer Vision*, 630–646. Springer.

Yew, Z. J.; and Lee, G. H. 2020. Rpm-net: Robust point matching using learned features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11824–11833.

Zeng, A.; Song, S.; Nießner, M.; Fisher, M.; Xiao, J.; and Funkhouser, T. 2017. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1802–1811.

Zhang, J.; Sun, D.; Luo, Z.; Yao, A.; Zhou, L.; Shen, T.; Chen, Y.; Quan, L.; and Liao, H. 2019. Learning Two-View Correspondences and Geometry Using Order-Aware Network. *arXiv preprint arXiv:1908.04964*.

Zhao, C.; Cao, Z.; Li, C.; Li, X.; and Yang, J. 2019. NM-Net: Mining Reliable Neighbors for Robust Feature Correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 215–224.

Zhou, L.; Zhu, S.; Luo, Z.; Shen, T.; Zhang, R.; Zhen, M.; Fang, T.; and Quan, L. 2018. Learning and matching multi-view descriptors for registration of point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 505–522.

Zhou, Q.-Y.; Park, J.; and Koltun, V. 2016. Fast global registration. In *European Conference on Computer Vision*, 766–782. Springer.

Zhou, Q.-Y.; Park, J.; and Koltun, V. 2018. Open3D: A modern library for 3D data processing. *arXiv preprint arXiv:1801.09847*.