remote sensing

MDPI

Article

Sparse-to-Dense Point Cloud Registration Based on Rotation-Invariant Features

Tianjiao Ma [1,2], Guangliang Han [1,*], Yongzhi Chu [1,2] and Hong Ren [1]

[1] Changchun Institute of Optics, Fine Mechanics and Physics (CIOMP), Chinese Academy of Sciences, Changchun 130033, China; matianjiao@ciomp.ac.cn (T.M.); chuyongzhi@ciomp.ac.cn (Y.C.); renhong@ciomp.ac.cn (H.R.)
[2] University of Chinese Academy of Sciences, Beijing 101408, China
* Correspondence: hangl@ciomp.ac.cn

Abstract: Point cloud registration is a critical problem because it is the basis of many 3D vision tasks. With the popularity of deep learning, many scholars have focused on leveraging deep neural networks to address the point cloud registration problem. However, many of these methods are still sensitive to partial overlap and differences in density distribution. For this reason, herein, we propose a method based on rotation-invariant features and using a sparse-to-dense matching strategy for robust point cloud registration. Firstly, we encode raw points as superpoints with a network combining KPConv and FPN, and their associated features are extracted. Then point pair features of these superpoints are computed and embedded into the transformer to learn the hybrid features, which makes the approach invariant to rigid transformation. Subsequently, a sparse-to-dense matching strategy is designed to address the registration problem. The correspondences of superpoints are obtained via sparse matching and then propagated to local dense points and, further, to global dense points, the byproduct of which is a series of transformation parameters. Finally, the enhanced features based on spatial consistency are repeatedly fed into the sparse-to-dense matching module to rebuild reliable correspondence, and the optimal transformation parameter is re-estimated for final alignment. Our experiments show that, with the proposed method, the inlier ratio and registration recall are effectively improved, and the performance is better than that of other point cloud registration methods on 3DMatch and ModelNet40.

Keywords: point cloud registration; point pair features; rotation-invariant; spatial consistency

check for updates

Citation: Ma, T.; Han, G.; Chu, Y.; Ren, H. Sparse-to-Dense Point Cloud Registration Based on Rotation-Invariant Features. *Remote Sens.* **2024**, *16*, 2485. https://doi.org/10.3390/rs16132485

Academic Editor: Filiberto Chiabrando

Received: 14 April 2024
Revised: 4 June 2024
Accepted: 12 June 2024
Published: 6 July 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Point cloud registration is the fundamental and critical step in many 3D computer vision applications, typically used in 3D reconstruction [1], 3D localization [2], and pose estimation [3], the essence of which is to estimate the transformation matrix between point clouds. Driven by the development of 3D scanning devices and 3D point representation learning [4–6], point cloud registration technology has also been promoted and improved, an approach that has experienced a long development history, with traditional methods being proposed in the early stages and learning-based methods becoming mainstream in the latter stages. Iterative closest point (ICP) [7] was proposed as the most classic of the traditional methods and demonstrates how to compute the transformation matrix with SVD [8] when the correspondences are known. However, ICP does not perform well when the initial pose is poor because the correspondences obtained are based on spatial distance. Feature-based algorithms combined with RANSAC [9] methods have shown advantages in addressing this situation. These methods find correspondences based on the features of points, so they are less sensitive to the initial pose. The performance of registration depends on the expression ability of features. Unfortunately, these features are not robust enough in most cases, resulting in unsatisfactory registration results.

Recently, with the popularity of deep learning applications, many scholars have attempted to use learning-based methods for point cloud registration to improve the poor performance caused by the deficiencies of traditional technologies. Learning-based methods can be roughly divided into feature-based learning [10–14] and end-to-end learning [15–19]. The above models are designed and optimized for the point cloud registration pipeline and improve both robustness and efficiency compared with traditional methods. However, learning-based methods still have shortcomings in some aspects. On the one hand, feature-based learning methods build correspondences according to the salient point features extracted by a neural network and cannot obtain the final transformation matrix directly. The registration result not only depends on the correspondences established but is also affected by the subsequent matching algorithm. On the other hand, end-to-end learning methods utilize an end-to-end network to cope with the registration problem, which regards the estimation of the transformation matrix parameters as a regression problem. Many of these methods cannot be applied to a large point cloud because of a lack of global features, and some are limited by the initial pose.

Based on the aforementioned studies, we propose a method with a sparse-to-dense matching strategy based on rotation-invariant features named STDRIF to address partial point cloud registration robustly. In particular, we focus on improving the accuracy of low-overlap point cloud registration. We adopt a sparse-to-dense matching strategy, which first finds coarse correspondences based on superpoints through the learned distinctive features embedded as point pair features in the coarse stage. The learned features are rotation-invariant, which makes this approach robust for partial overlap even without a good initial pose. Subsequently, correspondences are propagated from superpoints to local dense points and, further, to global dense points. We first find local correspondences and estimate a transformation parameter for each patch. Then, the global correspondences are collected from all patches, and the transformation parameter with the least registration error is selected. Finally, the accuracy of registration is improved through a progressive alignment module. The optimal registration is gradually refined by repeatedly feeding enhanced features updated with spatial consistency into the sparse-to-dense matching module. Overall, our main contributions are as follows:

- A point pair feature Transformer module is introduced, which is composed of point pair feature self-attention (PPF-SA) and feature-base cross-attention (F-CA). PPF-SA embeds point pair features into the self-attention module, which encodes the internal structure for each point cloud. F-CA interacts with the information between two point clouds whose output are hybrid features that are rotation-invariant.
- A sparse-to-dense matching strategy is proposed to address point cloud registration. Seeking superpoint correspondences is the first task, and then the superpoints are propagated to local dense points and further to global dense points in order to obtain global correspondences.
- Spatial consistency is utilized to enhance features after dense matching, and the enhanced features are repeatedly fed into the sparse-to-dense matching module to rebuild more reliable correspondences for optimizing registration.

2. Related Work

2.1. Point Cloud Registration

Point cloud registration methods can be roughly divided into two categories, traditional and learning-based. Traditional point cloud registration methods are diverse and comprehensive. ICP [7] and its variants [20–23] implement point cloud registration in an iterative and progressive manner. Although the ICP family of methods is widely used in point cloud registration, these often fall into local optimality because of a lack of good initialization, which leads to wrong correspondences. Many researchers bypass the localization to solve point cloud registration by finding correspondences directly, the steps of which are as follows: (1) detect key points; (2) compute the feature descriptors for key points using approaches such as those detailed in [24–26]; (3) match features to find correspondences;

and (4) estimate the transformation, typically using RANSAC. However, these methods tend to miss useful correspondences or generate wrong correspondences, which may lead to unsatisfactory registration results. There are other popular methods that attempt to regard the point cloud registration as a probability distribution. GMM-based methods [27,28] are inspired by likelihood maximization and can calculate the transformation matrix and the parameters with an optimal strategy. Although the sensitivity to noise and outliers has been improved to some extent, good localization is still necessary to avoid falling into local optimality. In general, traditional methods are developed based on careful feature design and pipeline optimization, and they have become the fundamental mainstream approaches. However, these methods still face great challenges in partially overlapping point cloud registration because of their sensitivity to noise and lack of good initialization.

Feature-based learning and end-to-end learning methods are the mainstream directions of learning-based point cloud registration. Feature-based learning methods utilize deep neural networks to learn robust features for seeking correspondences, which perform better than early descriptors against clutter and occlusion.

For example, 3DMatch [29] extracts local volumetric patch feature descriptors with a Siamese 3D CNN to establish correspondences. FCGF [30] proposed fully convolutional geometric features with similar performance to the best patch-based descriptor but several orders of magnitude faster. D3Feat [31] proposed a key point detection strategy that uses a 3D fully convolutional network to predict a detection score and a description feature. End-to-end learning methods leverage an end-to-end neural network to align point clouds directly. RPMNET [32] proposed an end-to-end framework combining the Sinkhorn layer with a deep neural network to establish soft correspondence from hybrid features, enhancing its robustness to noise. FMR [33] presents a feature-metric-based framework by combining deep learning and Lucas Kanade optimization, which converts the registration problem to minimizing feature differences. Although the learning-based methods mentioned above leverage the merits of traditional mathematics and deep learning and are effective in point cloud registration, their robustness, accuracy, and adaptability still decline dramatically when facing the mixture of noise, outliers, density differences, and low overlap.

2.2. Transformer-Based Methods

A point cloud is a set of irregular points without a specific order. Transformer [34] is suitable for point cloud data because of its core attention mechanism, which is inherently permutation-based and does not depend on connections between points. For this reason, many researchers have applied transformers to 3D point cloud processing and achieved significant success. The Deep Closest Point (DCP) [35] model was proposed to predict a rigid transformation, utilizing a feature embedding module [36,37] to extract features, applying a transformer to perform context aggregation between two embedded features, and using the Kabsch algorithm [38] to estimate transformation parameters. The deep graph matching-based framework (RGM) [39] was proposed by Fu et al. for robust point cloud registration. Fu et al. utilized a transformer to learn the node features and the soft edges of two nodes, which led to better correspondences in partially overlapping registration. REGTR [40] introduced an end-to-end Transformer framework aiming to find the final correspondences. The features extracted by KPConv [41] were fed into multi-head attention mechanisms to predict the correspondences, and it was argued that rigid transformations can be estimated without RANSAC. GCMTN [42] leverages dense graph convolution and a multilevel interaction transformer to reduce mismatching caused by repeated geometric structures, and it has been shown to perform well in low-overlap registration. A multilevel interaction transformer is applied to refine the internal features and perform feature interaction. The final transformation matrix is estimated according to the distribution overlap region predicted by an overlap prediction module.

The above transformer-based methods aim to extract features and encode contextual information. However, these methods only feed the Transformer with high-level point

features, neglecting spatial structure and lacking geometric and positional discrimination, which leads to a large number of outlier matches.

2.3. Spatial Consistency Embedding Methods

Spatial consistency refers to the invariance property of the relative positions and orientations caused by the rigid transformation between two point clouds. Previous studies have demonstrated that embedding relative positions or orientations into feature extraction modules makes the learned features more distinguished in their geometric structure, resulting in an improvement of the inlier ratio. Point DSC [43] introduced a SCNonlocal module to learn a discriminative embedding space by using spatial consistency. Geotransformer [44] encodes point pair distances and point-triplet angles for each input point cloud for the extraction of distinctive geometric features, leading to high matching accuracy. RoITr [45] constructed a novel attention-based encoder-decoder architecture by embedding point pair features into an attention mechanism to address pose variations. DoPE [46] was proposed to encode positional information by computing the joint origin as the origin of the shared coordinate system for all points. OIF-PCR [47] proposed an efficient position encoding for point cloud registration that requires only a small addition of memory and computing overhead. The author first produced one virtual correspondence for the point cloud registration network and then carried out point-wise position encoding on the basis of two reference points corresponding to the virtual correspondence. Lepard [48] encoded the position information of the point cloud into the feature vector and explicitly represented the 3D relative distance between the point clouds through the dot product of the vectors, thereby improving the accuracy and robustness of point cloud matching and registration.

3. Methods

3.1. Problem State

The essence of point cloud registration is to estimate a transformation matrix so that the distance between two point clouds obtained from two perspectives can be minimized after transformation. We consider that there are two point clouds: the source point cloud $\mathrm{S} = \left\{ \mathrm{S}_{x_i} \in \mathbb{R}^3 \middle| i = 1, \ldots, N \right\}$ and the target point cloud $\mathrm{T} = \left\{ \mathrm{T}_{y_i} \in \mathbb{R}^3 \middle| i = 1, \ldots, M \right\}$, the goal is to estimate the transformation matrix $\mathbf{M}_T^S = \{\mathbf{R}, \mathbf{t}\}$ by solving the following problem:

$$\min_{\mathbf{R},\mathbf{t}} \sum_{(\mathrm{S}_{x_i}^*, \mathrm{T}_{y_i}^*) \in C^*} \left\| \mathrm{T}_{y_i}^* - (\mathbf{R} \cdot \mathrm{S}_{x_i}^* + \mathbf{t}) \right\|_2 \tag{1}$$

where $\mathbf{R} \in SO(3)$ denotes the rotation parameter, $\mathbf{t} \in \mathbb{R}^3$ denotes the translation parameter, and C^* stands for the predicted correspondences between S and T.

3.2. Network Architecture

This section illustrates the network architecture as shown in Figure 1. In the encoder stage, the input point clouds S and T are, respectively, encoded into a set of superpoints with associated features through a KPConv-FPN backbone network, which is then fed into the point pair feature transformer to extract the rotation-invariant hybrid features in both geometric space and feature space. Subsequently, the sparse-to-dense matching module establishes correspondences between superpoints, which are then propagated to local dense points and, further, to global dense points. A set of correspondences is found in each superpoint corresponding patch, and a series of transformation parameters are obtained by utilizing SVD. Finally, spatial consistency is used to enhance the hybrid features, and the enhanced features are repeatedly fed into the sparse-to-dense matching module to rebuild more reliable correspondences and gradually optimize registration, which leads to the final transformation parameters.

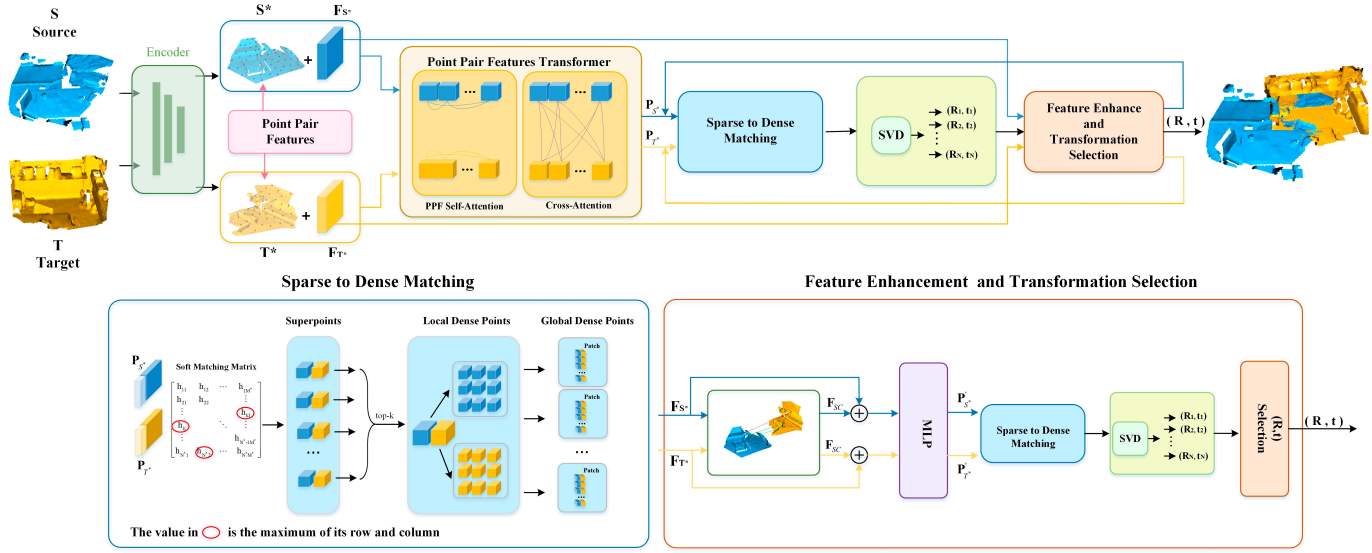


Figure 1. Network architecture overview.

3.3. Encoder to Superpoint

We use the KPConv-FPN backbone [41,44,49] to aggregate the original point clouds into a much coarser subset of points with their associated features. In the above processing, grid sampling is adopted for downsampling, which can uniformly cover the points in space and make KPConv robust to different densities. Moreover, the learned latent features are enhanced by combining KPConv and a feature pyramid network (FPN), which extract multi-level features from point clouds of different resolutions. Many previous studies have proven that establishing correspondences with coarser points can reduce the number of useless correspondences.

In this paper, we treat the first down-sample points as dense points, and the points with the coarsest resolution are the superpoints. For the convenience of the following illustration, the superpoints corresponding to the source and target clouds are denoted as $S^* = \left\{ S^*_{x_i} \middle| i = 1, 2, \ldots N^* \right\}$ and $T^* = \left\{ T^*_{y_i} \middle| i = 1, 2, \ldots, M^* \right\}$, respectively, and their associated learned features are represented by $\widetilde{\mathbf{F}}_{S^*}$ and $\widetilde{\mathbf{F}}_{T^*}$. Moreover, in order to preserve the information from the original point clouds, we integrate extracted features with point clouds as input features for subsequent processing. The integral features are denoted as $\mathbf{F}_{S*} = (\widetilde{\mathbf{F}}_{S*}, S*)$ and $\mathbf{F}_{T*} = (\widetilde{\mathbf{F}}_{T^*}, T*)$.

3.4. Point Pair Features Transformer Module

Capturing local features is not enough for point cloud registration, with the knowledge that it is necessary to unite the global context. The importance of global context has been shown in many registration tasks [50–52]. In order to enrich the contextual information of each input point cloud and exchange the information between the two input point clouds, we proposed a point pair feature [53] transformer (PPF Transformer) module, which contains a point pair feature self-attention (PPF SA) module and a feature-based cross-attention (F CA) module. Previous work has proven that encoding only high-level point features leads to numerous severe outlier matches [35,54]. To address this problem, we encode the internal structure of the point cloud, making it invariant to rigid transformation. PPF self-attention encodes geometric features between the point pairs for each point cloud, and feature-based cross-attention interacts with the feature information between the source point cloud and target point cloud in feature space.

3.4.1. Point Pair Feature Self-Attention

Point pair feature self-attention is designed to learn the rotation-invariant features in both feature space and structure space, which are used to enrich contextual information

and measure the feature similarity for each point cloud. Given the input feature matrix, the output contextual feature matrix $\mathrm{F}_{sa} \in \mathbb{R}^{|S^*| \times d_t}$ is updated by the following equation:

$$\mathbf{F}_{sa} = \sum_{j=1}^{N^*} \mathrm{h}_{MLP}(\mathrm{softmax}(\boldsymbol{\alpha}_{i,j}))(\mathbf{F}_{\mathrm{S}^*_{x_j}} \mathbf{W}^V) \tag{2}$$

where $h_{MLP}(\cdot)$ is a three-layer full network; $\boldsymbol{\alpha}_{i,j}$ denotes the attention score which represents the similarity between $\mathbf{F}_{\mathrm{S}^*_{x_i}}$ and $\mathbf{F}_{\mathrm{S}^*_{x_j}}$; and $\boldsymbol{\alpha}_{i,j}$ is computed as follows:

$$\boldsymbol{\alpha}_{i,j} = \frac{(\mathbf{F}_{S^*_{x_i}} \mathbf{W}^Q)(\mathbf{F}_{S^*_{x_j}} \mathbf{W}^K + \boldsymbol{\delta}_{i,j} \mathbf{W}^{PPF})^T}{\sqrt{\mathrm{d}_t}} \tag{3}$$

$\mathbf{F}_{\mathrm{S}^*_{x_i}} \in \mathbf{F}_{\mathrm{S}^*}$ and $\mathbf{F}_{\mathrm{S}^*_{x_j}} \in \mathbf{F}_{S^*}$ are the associated features of two superpoints $\mathrm{S}^*_{x_i}$ and $\mathrm{S}^*_{x_j}$, respectively; $\boldsymbol{\delta}_{i,j}$ represents the point pair feature between $\mathbf{F}_{\mathrm{S}^*_{x_i}}$ and $\mathbf{F}_{\mathrm{S}^*_{x_j}}$; and $\mathbf{W}^Q$, $\mathbf{W}^K$, $\mathbf{W}^V$, and $\mathbf{W}^{PPF} \in \mathbb{R}^{d_t \times d_t}$ denote the learnable weighted matrices for queries, keys, values, and point pair features, respectively. The structure and calculation method of PPF self-attention is shown in Figure 2, whose process is first carrying out a softmax on attention score $\boldsymbol{\alpha}_{i,j}$, then multiplying the matrix obtained from the previous step by the value vector and then via an MLP layer, and finally summing the output result of the liner layer to obtain the final feature. The PPF self-attention features for the point cloud T^* are obtained in the same way.

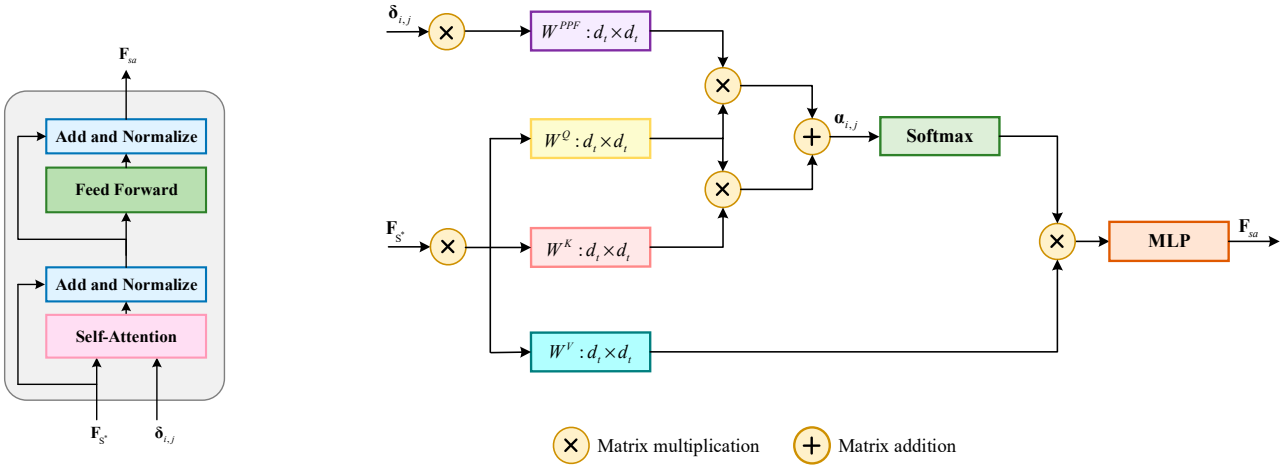


Figure 2. Illustration of (**a**) the structure of the PPF self-attention layer and (**b**) the computation of the feature matrix $\mathbf{F}_{sa}$.

For each superpoint, we construct a patch $\mathcal{P}_i^S$ with the point-to-node strategy [44], and the patch is composed of a series of dense points which are defined as follows:

$$\mathcal{P}_i^S = \left\{ \mathrm{S}_{x_j} \in \mathrm{S} \middle| j = \mathrm{argmin}_i(\|\mathrm{S}_{x_j} - \mathrm{S}^*_{x_i}\|), \mathrm{S}^*_{x_i} \in \mathrm{S}^* \right\} \tag{4}$$

In order to generate rotation-invariant features, we embed the point pair feature $\boldsymbol{\delta}_{i,j}$ into the attention score $\boldsymbol{\alpha}_{i,j}$. Point pair feature $\boldsymbol{\delta}_{i,j}$ consists of four components, as shown in Figure 3, which are defined as follows:

$$\boldsymbol{\delta}_{i,j} = (\|\mathbf{d}_{i,j}\|_2, \angle(\mathbf{n}_{\mathrm{S}^*_{x_i}}, \mathbf{d}_{i,j}), \angle(\mathbf{n}_{\mathrm{S}^*_{x_j}}, \mathbf{d}_{i,j}), \angle(\mathbf{n}_{\mathrm{S}^*_{x_i}}, \mathbf{n}_{\mathrm{S}^*_{x_j}})) \tag{5}$$

where $\mathbf{n}_{\mathrm{S}^*_{x_i}}$ and $\mathbf{n}_{\mathrm{S}^*_{x_j}}$ are the normals of superpoints $\mathrm{S}^*_{x_i}$ and $\mathrm{S}^*_{x_j}$ respectively, which are computed by using the k-nearest neighbors dense points of the superpoints. In Equation (5),

$\|\mathbf{d}_{i,j}\|_2$ is the distance between superpoints $S^*_{x_i}$ and $S^*_{x_j}$, $\angle(\mathbf{n}_{S^*_{x_i}}, \mathbf{d}_{i,j})$ and $\angle(\mathbf{n}_{S^*_{x_j}}, \mathbf{d}_{i,j})$ denote the angles between the respective normals and the vector of two superpoints, $\angle(\mathbf{n}_{S^*_{x_i}}, \mathbf{n}_{S^*_{x_j}})$ is the angle between the two normals, and $\|\mathbf{d}_{i,j}\|_2$ and $\angle$ are calculated as follows:

$$\|\mathbf{d}_{i,j}\|_2 = \|S^*_{x_i} - S^*_{x_j}\|_2 \tag{6}$$

$$\angle(\vec{a}, \vec{b}) = \arctan(\vec{a} \times \vec{b} / \vec{a} \cdot \vec{b}) \tag{7}$$

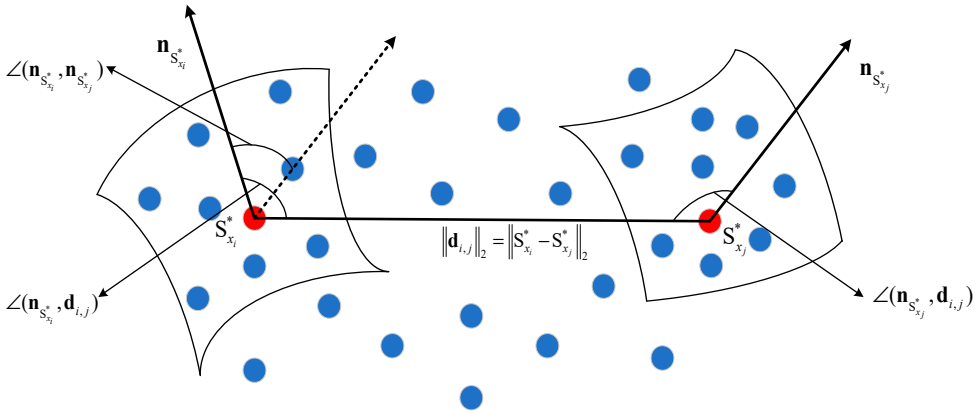


Figure 3. Point pair feature of the superpoint. The red dots denote superpoints, and the blue dots denote the neighbor's dense points of the superpoints.

3.4.2. Feature-Based Cross-Attention

Cross-attention is utilized to exchange the information between two input point clouds, the structure and calculation method of cross-attention is shown in Figure 4. We denote the output feature matrices of the self-attention module for S^* and T^* as $\hat{\mathbf{F}}_{S^*_{x_j}}$ and $\hat{\mathbf{F}}_{T^*_{y_j}}$, respectively, which are the input features of the cross-attention module, and the cross-attention features of S^* can be computed as follows:

$$\mathbf{F}^S_{ca} = \sum_{j=1}^{N^*} h_{MLP}(\text{softmax}(\boldsymbol{\beta}_{i,j}))(\hat{\mathbf{F}}_{T^*_{y_j}} \mathbf{W}^V) \tag{8}$$

where $\boldsymbol{\beta}_{i,j}$ is the correlation score between $\hat{\mathbf{F}}_{S^*_{x_j}}$ and $\hat{\mathbf{F}}_{T^*_{y_j}}$, which is defined as follows:

$$\boldsymbol{\beta}_{i,j} = \frac{(\hat{\mathbf{F}}_{S^*_{x_j}} \mathbf{W}^Q)(\hat{\mathbf{F}}_{T^*_{y_j}} \mathbf{W}^K)^T}{\sqrt{d_t}} \tag{9}$$

The cross-attention features of T^* can be computed in the same way, and so far, the information exchange has been completed. Benefiting from the rotation-invariant features encoded by the PPF SA module, the mixed features are rotation-invariant to rigid transformation and contribute to finding the superpoints' correspondences.

3.5. Sparse-to-Dense Matching

3.5.1. Sparse Matching

Benefiting from the distinctive hybrid features $\mathbf{P}_{S^*}$ and $\mathbf{P}_{T^*}$, we find the superpoints correspondences by utilizing a differentiable optimal transport. The similarity matrix $\mathbf{M} \in \mathbb{R}^{N^* \times M^*}$ is calculated as follows:

$$\mathbf{M}_{i,j} = \mathbf{P}_{S^*_{x_i}} \mathbf{P}^T_{T^*_{y_i}} / \sqrt{d_t} \tag{10}$$

Subsequently, $\bar{\mathbf{M}}$ is computed as the augmented matrix of $\mathbf{M}$ by appending a new row and a new column, as reported in [47], which is filled with a learnable dustbin parameter α. Then, a Sinkhorn algorithm [55] is implemented on $\bar{\mathbf{M}}$ to calculate the soft matching matrix $\bar{\mathbf{H}}$, and the sparse matching is converted to an optimal transport problem by computing the maximization of $\sum_{i,j} \bar{\mathbf{M}} \cdot \bar{\mathbf{H}}$. The soft matching score $\mathbf{H} \in \mathbb{R}^{N^* \times M^*}$ can be obtained by removing the last row and column of $\bar{\mathbf{H}}$. In practical terms, the soft matching score guides the ratio of inliers. For example, when the value $\mathbf{H}_{i,j}$ is higher, the possibility of this point pair being the correct correspondence is greater. Here, we pick reliable superpoints correspondences based on the strategy with top-k matching scores, where a superpoints correspondence is selected if its matching score is the maximum in both its row and column. However, it is possible for the accuracy of superpoints' correspondences to be compromised when there are repetitive and textureless structures in the scene. Thus, the proposed method needs to be further strengthened to reduce the mismatching caused by the above-mentioned issues.

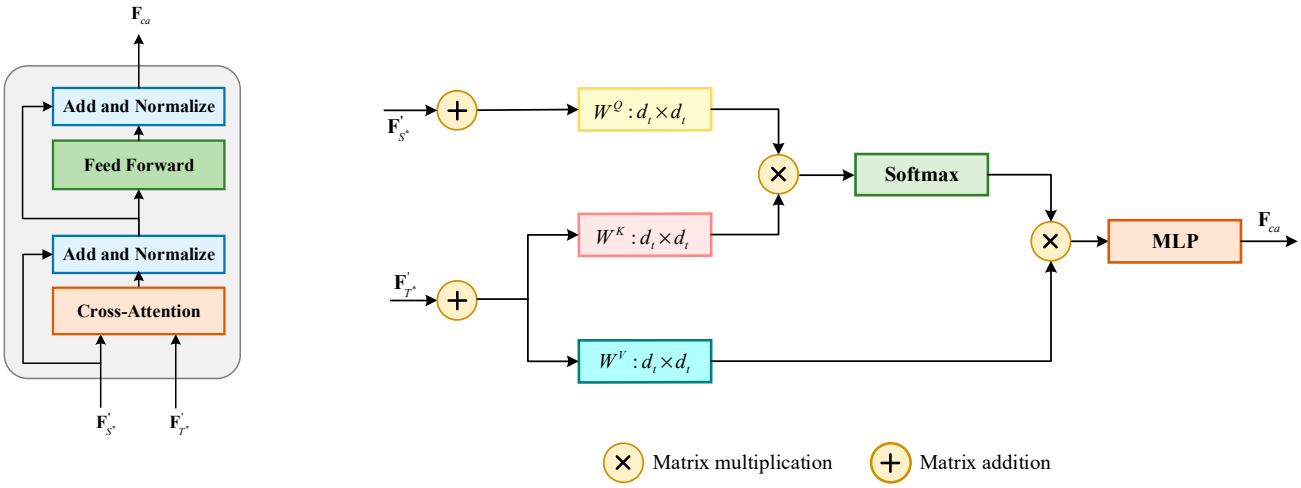


Figure 4. Illustration of (**a**) the structure of the cross-attention layer and (**b**) the computation of the feature matrix $\mathbf{F}_{ca}$.

3.5.2. Dense Matching

Sparse matching is effective in solving global ambiguity, and the dense matching module is designed to improve robustness at the level of dense points. The matching process propagates the establishment of superpoints' correspondences to local dense points and, further, to global dense points. The final global correspondences are collected from all local correspondences. Based on the coarse correspondences, we first find the patch-wise correspondences in the same way in which we obtained the superpoints' correspondences. The difference is that the similarity matrix is computed with the features generated by the backbone. Then we estimate a transformation parameter $\mathbf{T}_i = \{\mathbf{R}_i, \mathbf{t}_i\}$ for each superpoints' correspondence with their patches in the dense point cloud, generating the local correspondences in each patch. The transformation matrix is solved by the following:

$$\{\mathbf{R}_i, \mathbf{t}_i\} = \min_{\mathbf{R},\mathbf{t}} \sum\nolimits_{(\mathrm{S}_{x_i}, \mathrm{T}_{y_i}) \in C_i} \omega_j^i \| \mathrm{T}_{y_i} - (\mathbf{R} \cdot \mathrm{S}_{x_i} + \mathbf{t}) \|_2^2 \tag{11}$$

where the soft matching score $\bar{\mathbf{H}}_{i,j}$ calculated above is used as the weighting coefficient ω_j^i, and this equation can be solved via SVD. Then, we choose a transformation parameter $\mathbf{T}_i = \{\mathbf{R}_i, \mathbf{t}_i\}$ from $\mathbf{T} = \{\mathbf{T}_i | i = 1, 2, \ldots, N^*\}$ to be used in global point matching, with the

condition that the patch corresponding to $\mathbf{T}_i = \{\mathbf{R}_i, \mathbf{t}_i\}$ has the highest inlier ratio among all patches.

$$\{\mathbf{R}, \mathbf{t}\} = \max_{R_i, t_i} \sum_{(\mathrm{S}_{x_i}, \mathrm{T}_{y_i}) \in C} [\![\|\mathbf{R}_\mathrm{i} \cdot \mathrm{S}_{x_i} + \mathbf{t}_i - \mathrm{T}_{y_i}\|_2^2 < \tau]\!] \tag{12}$$

Here, $[\![\cdot]\!]$ is the Iverson bracket and τ is the inlier threshold.

3.6. Progressive Alignment with Spatial Consistency

In this stage, spatial consistency features are learned repeatedly to enhance the distinctiveness of the above-learned features in order to find more reliable superpoints' correspondences, and the matching results are gradually optimized with the more reliable correspondences. Spatial consistency is the byproduct of rigid transformation, which is determined by the property of isometric isomorphism. We leverage spatial consistency to obtain correct dense correspondences, which depend on the spatial geometric structure between all inlier points not changing with rotation and translation, as shown in Figure 5.

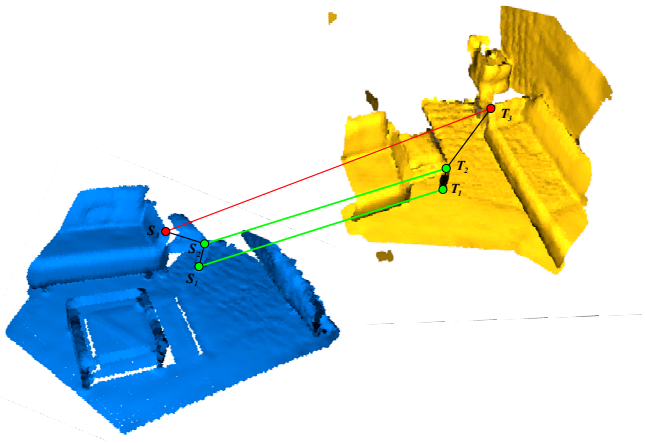


Figure 5. Illustration of spatial consistency. The green line denotes inlier correspondence, and the red line denotes outlier correspondence.

According to the property of isometric isomorphism, we compute spatial consistency features and feed them into an MLP with 4 hidden layers after normalization; the output is a spatial consistency feature that is used to update the hybrid features via direct addition. Subsequently, we compute the spatial consistency feature S^* with Equation (13), and the same goes for T^*. The hybrid features are updated as shown in Equation (14).

$$\mathbf{F}_{SC} = \mathrm{h}_{MLP}(\sum_{i=1}^{N^*} \mathrm{softmax}(\boldsymbol{\alpha}\boldsymbol{\beta})\}(\widetilde{\mathbf{P}}_{\mathrm{S}_i^*})) \tag{13}$$

$$\widetilde{\mathbf{P}}_{\mathrm{T}^*} = \mathbf{P}_{\mathrm{T}^*} + \mathbf{F}_{SC} \tag{14}$$

Here, $\mathbf{F}_{SC}$ first computed with $\widetilde{\mathbf{P}}_{\mathrm{S}^*} = \mathbf{P}_{\mathrm{S}^*}$, and subsequently with $\widetilde{\mathbf{P}}_{\mathrm{S}^*}$ computed using Equation (10); $\}(\cdot)$ is a linear projection function; $\boldsymbol{\alpha}$ represents the dot-product similarity [43]; and, $\boldsymbol{\beta}$ denotes the spatial consistency constraint, calculated as follows:

$$\beta_{i,j} = \max(1 - \frac{\rho_{i,j}^2}{\sigma_d^2}, 0) \tag{15}$$

where $\max(\cdot, 0)$ is used to ensure that $\beta_{i,j}$ is always non-negative, σ_d is a distance parameter sensitive to length difference, and $\rho_{i,j}$ denotes the length difference between the segment of two points in point cloud S^* and the segment of two corresponding points in point cloud T^*; definition of $\rho_{i,j}$ is as follows:

$$\rho_{i,j} = \left| \|\widetilde{\mathrm{S}}_{x_i}^* - \widetilde{\mathrm{S}}_{x_j}^*\| - \|\mathrm{T}_{y_i}^* - \mathrm{T}_{y_j}^*\| \right| \tag{16}$$

where $\widetilde{S}^*_{x_i}$ and $\widetilde{S}^*_{x_j}$ are the updated superpoints with the transformation matrix $\{\mathbf{R}, \mathbf{t}\}$, which is the result of dense matching; $\widetilde{S}^*_{x_i}$ and $\widetilde{S}^*_{x_j}$ are calculated via Equation (17).

$$\widetilde{S}^*_{x_i} = S^*_{x_i} \cdot \mathbf{R} + \mathbf{t} \tag{17}$$

We repeat the operation of feature enhancement to rebuild reliable correspondences from the sparse matching stage, and then we re-estimate the optimal transformation parameters with surviving inliers in the dense matching stage. In fact, the transformation estimated via sparse matching is quite good, but the enhanced features are more distinctive and further improve the accuracy of registration, especially in scenes with many repetitive and textureless structures.

3.7. Loss Functions

In this paper, sparse matching and dense matching directly influence the quality of point cloud registration, so we designed superpoints matching loss L_{sp} and points matching loss L_d to supervise the above two stages, and the total loss function is computed as follows:

$$\mathcal{L}_{loss} = \mathcal{L}_{sm} + \mathcal{L}_{pm} \tag{18}$$

In the following description, we take the loss on the source point cloud S as an example, and the loss on the target point cloud T can be computed in the same way, which will not be repeated later.

3.7.1. Superpoint Loss

We follow [44] and utilize the improved circle loss [56] to supervise the patch-wise feature descriptors for sparse matching in a metric-learning fashion. Here we consider that several patches $\mathcal{P}^S_i$ of superpoints S^*_i exist in the source point cloud S, which have at least one positive patch in the target point cloud T. The above patch pairs form the patch set $\mathcal{P}$. For each patch $\mathcal{P}^S_i \in \mathcal{P}$, the sets of its positive and negative patches in point cloud T are represented as ε^i_p and ε^i_n respectively. Here, a positive patch means that the patch shares a >10% overlap region with patch $\mathcal{P}^S_i$, while a negative patch means that there is no overlap region. The superpoint matching loss on the point cloud S is computed as follows:

$$\mathcal{L}^S_{sm} = \frac{1}{|\mathcal{P}|} \sum_{\mathcal{P}^S_i \in \mathcal{P}} \log \left[1 + \sum_{\mathcal{P}^T_j \in \varepsilon^i_p} e^{\lambda^j_i \beta^{i,j}_p (d^j_i - \Delta_p)} \cdot \sum_{\mathcal{P}^T_j \in \varepsilon^i_n} e^{\beta^{i,k}_n (\Delta_n - d^k_i)} \right] \tag{19}$$

where $d^j_i = \| P_{S^*_i} - P_{T^*_j} \|_2$ denotes the distance in feature space; λ^j_i is calculated as $\lambda^j_i = (o^i_j)^{\frac{1}{2}}$; o^i_j represents the overlap ratio between $\mathcal{P}^S_i$; $\beta^{i,j}_p$ and $\beta^{i,k}_n$ are the positive and negative weights and are calculated as $\beta^{i,j}_p = \gamma(d^j_i - \Delta_p)$ and $\beta^{i,k}_n = \gamma(\Delta_n - d^k_i)$, respectively, γ is a hyper-parameter; and Δ_p and Δ_n denotes the positive and negative margins, respectively, which are set to $\Delta_p = 0.1$ and $\Delta_n = 1.4$ empirically. The overall superpoint matching loss is $\mathcal{L}_{sm} = \mathcal{L}^S_{sm} + \mathcal{L}^T_{sm}$.

3.7.2. Point Matching Loss

For point matching loss, we obtain the sets of ground-truth superpoint correspondences and dense correspondences with the ground-truth relative transformation and a matching radius τ. Here, the set of ground-truth dense correspondences in a pair of patches

is denoted as $\mathcal{M}_{\mathrm{i}}$, and those unmatched points of the patch-pairs are represented as $\mathcal{S}_{\mathrm{i}}$ and $\mathcal{T}_{\mathrm{i}}$; and the point matching loss is defined as follows:

$$\mathrm{L_{pm}} = -\frac{1}{\mathrm{C}}\sum_{i}^{\mathrm{C}}\Big(\sum_{(x,y)\in \mathrm{M}_i}\log \mathbf{H}^{i}_{x,y} + \sum_{x\in\mathcal{S}_i}\log \mathbf{H}^{i}_{x,m_i+1} - \sum_{x\in\mathcal{T}_i}\log \mathbf{H}^{i}_{n_i+1,y}\Big) \tag{20}$$

4. Experiments and Results

In this section, a series of experiments and comparisons were implemented on several publicly available datasets to analyze the experimental results and evaluate our method. Firstly, we performed comparative experiments on the 3DMatch [29] and 3DLoMatch [54] datasets, and we evaluated the performance with three metrics: IR, FMR, and RR. Subsequently, we checked the effectiveness of our approach on ModelNet [57] and ModelLoNet [54] with another three metrics: RRE, RTE, and RR. Furthermore, we conducted a series of additional comparison experiments on KITTI datasets to verify the validation of our method in large scenes. Finally, extensive ablation studies were conducted to illustrate the validity of the submodules.

The experiments were trained in 40 epochs on 3DMatch, 200 epochs on ModelNet40, and 80 epochs on KITTI with the Adam optimizer and initial learning rates of 0.005, 0.01, and 0.05, respectively. The batch size was set to 1 and the weight decay was set to 1×10^{-6} in all experiments. The learning rate is exponentially decayed by 0.05 for each epoch on 3DMatch and ModelNet40, and every four epochs on KITTI. We implemented the project on a platform with an Intel i7-11700K CPU and a Geforce RTX 3090 GPU. All programs depended on the environment with Pytorch 1.7.1, Python 3.8, CUDA 11.1, and cuDNN 8.1.0.

4.1. Experiments on 3DMatch and 3DLoMatch

4.1.1. Dataset and Metrics

The 3DMatch dataset is a typical indoor dataset captured by different depth sensors; it contains 62 scenes combining earlier data from RGB-D Scenes and 7 scenes et al., providing great diversity. We divided all scenes into three sets, with 46 scenes, 8 scenes, and 8 scenes, which we used for training, validation, and testing, respectively. Each scene consists of a series of partially overlapped point clouds with their ground-truth transformation. In this paper, we evaluate the proposed approach on 3DMatch and 3DLoMatch datasets, and the difference between 3DMatch and 3DLoMatch is that the overlap of the point cloud pairs in 3DMatch is more than 30%, while in 3DLoMatch it is 10~30%.

We evaluated the performance with reference to [44,54] utilizing three metrics: inlier ratio (IR), feature matching recall (FMR), and registration recall (RR).

IR is computed as the ratio of assumed correspondences, whose residual distance is smaller than a practical value set to $\tau_1 = 10$ cm through the ground-truth transformation $\mathbf{T}_{\mathrm{S}}^{-\mathrm{T}}$; the definition is as follows:

$$\mathrm{IR} = \frac{1}{|\mathcal{C}|}\sum_{(\mathbf{S}_{x_i},\mathbf{T}_{y_i})\in\mathcal{C}}[\![\|\mathbf{T}_{\mathrm{S}}^{-\mathrm{T}}(\mathrm{S}_{x_i}) - \mathrm{T}_{y_i}\|_2 < \tau_1]\!] \tag{21}$$

FMR is the proportion of point cloud pairs whose IR is larger than a practical threshold ($\tau_2 = 0.05$), which is used to measure the potential of success and computed as follows:

$$\mathrm{FMR} = \frac{1}{\mathrm{M}}\sum_{i=1}^{\mathrm{M}}[\![\mathrm{IR}_i > \tau_2]\!] \tag{22}$$

where M denotes the number of all point cloud pairs.

RR is the most reliable metric since it evaluates the end-to-end performance for registration; it is computed as the proportion of correctly registered point cloud pairs

whose transformation error is smaller than a certain threshold ($\tau_3 = 0.2$ m). After applying the estimated transformation, the transformation error can be computed as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{|\mathcal{C}^*|} \sum_{(\mathbf{S}^*_{x_i}, \mathbf{T}^*_{y_i}) \in \mathcal{C}^*} \| \mathbf{T}_{\mathrm{S}}^{\mathrm{T}}(\mathrm{S}^*_{x_i}) - \mathrm{T}^*_{y_i} \|_2^2} \tag{23}$$

And the registration recall is computed as follows:

$$RR = \frac{1}{\mathrm{M}} \sum_{i=1}^{\mathrm{M}} [\![\mathrm{RMSE}_i < \tau_3]\!] \tag{24}$$

4.1.2. Registration Results

The registration results of STDRIF were compared with six recent excellent methods: FCGF [30], D3Feat [31], Predator [54], SpinNet [10], REGTR [40], and Geotransformer [44]. We evaluated the performance with the FMR, IR, and RR metrics and reported the results on 3DMatch and 3DLoMatch with 5000, 2500, 1000, 500, and 250 correspondences, as shown in Table 1.

Table 1. Evaluation results for 3DMatch and 3DLoMatch. The best two performances are highlighted in **bold** and underlined, respectively.

Samples	3DMatch					3DLoMatch				
	5000	2500	1000	500	250	5000	2500	1000	500	250
Feature Matching Recall (%)										
FCGF [30]	97.4	97.3	97.0	96.7	96.6	76.6	75.4	74.2	71.7	67.3
D3Feat [31]	95.6	95.4	94.5	94.1	93.1	67.3	66.7	67.0	66.7	66.5
Predator [54]	96.6	96.6	96.5	96.3	96.5	78.6	77.4	76.3	75.7	75.3
SpinNet [10]	97.6	97.2	96.8	95.5	94.3	75.3	74.9	72.5	70.0	63.6
REGTR [40]	97.8	97.4	96.9	96.1	95.6	74.3	74.4	74.2	73.8	72.9
Geotransformer [44]	97.9	97.9	97.9	**97.9**	97.6	88.3	88.6	88.8	88.6	**88.3**
STDRIF(ours)	**98.0**	**98.0**	**98.0**	**97.9**	**97.7**	**88.9**	**89.1**	**89.0**	**88.8**	88.1
Inlier Ratio (%)										
FCGF [30]	56.8	54.1	48.7	42.5	34.1	21.4	20.0	17.2	14.8	11.6
D3Feat [31]	39.0	38.8	40.4	41.5	41.8	13.2	13.1	14.0	14.6	15.0
Predator [54]	58.0	58.4	57.1	54.1	49.3	26.7	28.1	28.3	27.5	25.8
SpinNet [10]	47.5	44.7	39.4	33.9	27.6	20.5	19.0	16.3	13.8	11.1
REGTR [40]	57.3	55.2	53.8	52.7	51.1	27.6	27.3	27.1	26.6	25.4
Geotransformer [44]	71.9	75.2	76.0	82.2	85.1	43.5	45.3	46.2	52.9	57.7
STDRIF(ours)	**72.3**	**77.5**	**79.1**	**85.3**	**87.2**	**44.7**	**48.2**	**51.1**	**54.1**	**58.9**
Registration Recall (%)										
FCGF [30]	85.1	84.7	83.3	81.6	71.4	40.1	41.7	38.2	35.4	26.8
D3Feat [31]	81.6	84.5	83.4	82.4	77.9	37.2	42.7	46.9	43.8	39.1
Predator [54]	89.0	89.9	90.6	88.5	86.6	59.8	61.2	62.4	60.8	58.1
SpinNet [10]	88.6	86.8	85.5	83.5	70.2	59.8	54.9	48.3	39.8	26.8
REGTR [40]	92.0	91.2	89.7	90.6	90.4	64.8	64.4	64.2	62.3	59.7
Geotransformer [44]	92.0	91.8	91.8	91.4	**91.2**	75.0	**74.8**	74.2	74.1	**73.5**
STDRIF(ours)	**92.4**	**92.0**	**92.4**	**91.7**	91.0	**75.9**	**74.8**	**74.7**	**74.7**	**73.5**

The evaluation results for 3DMatch and 3DLoMatch are exhibited separately on the left and right sides of Table 1, respectively. With the FMR metric, the performance was close to Geotransformer on 3DMatch, and it showed a better improvement on 3DLoMatch, verifying the validity of the submodules designed for finding correspondences. With the IR metric, the performance was superior to that of other methods, demonstrating the more reliable correspondences produced. RR was the most reliable and intuitive metric used to evaluate the performance of final registration, and the evaluation result directly proves the effectiveness of our approach. Overall, our approach performed best on 3DMatch, with close results compared to Geotransformer and significant improvements over other

comparison methods. The performance on 3DLoMatch was still strongly competitive. The evaluation results verify the robustness of our method.

In order to show the reliability of our approach intuitively, a visualization of the registration results on 3DMatch and 3DLoMatch with 5000 correspondences is displayed in Figure 6, which contains five examples represented in five separate lines. In each example, the first two columns denote the source and target point clouds, respectively. The third column denotes the ground truth, and the last column denotes the registration result. From the comparison of the registration results and ground truth, it can be seen that our method performs satisfactorily, especially in low-overlap scenes.

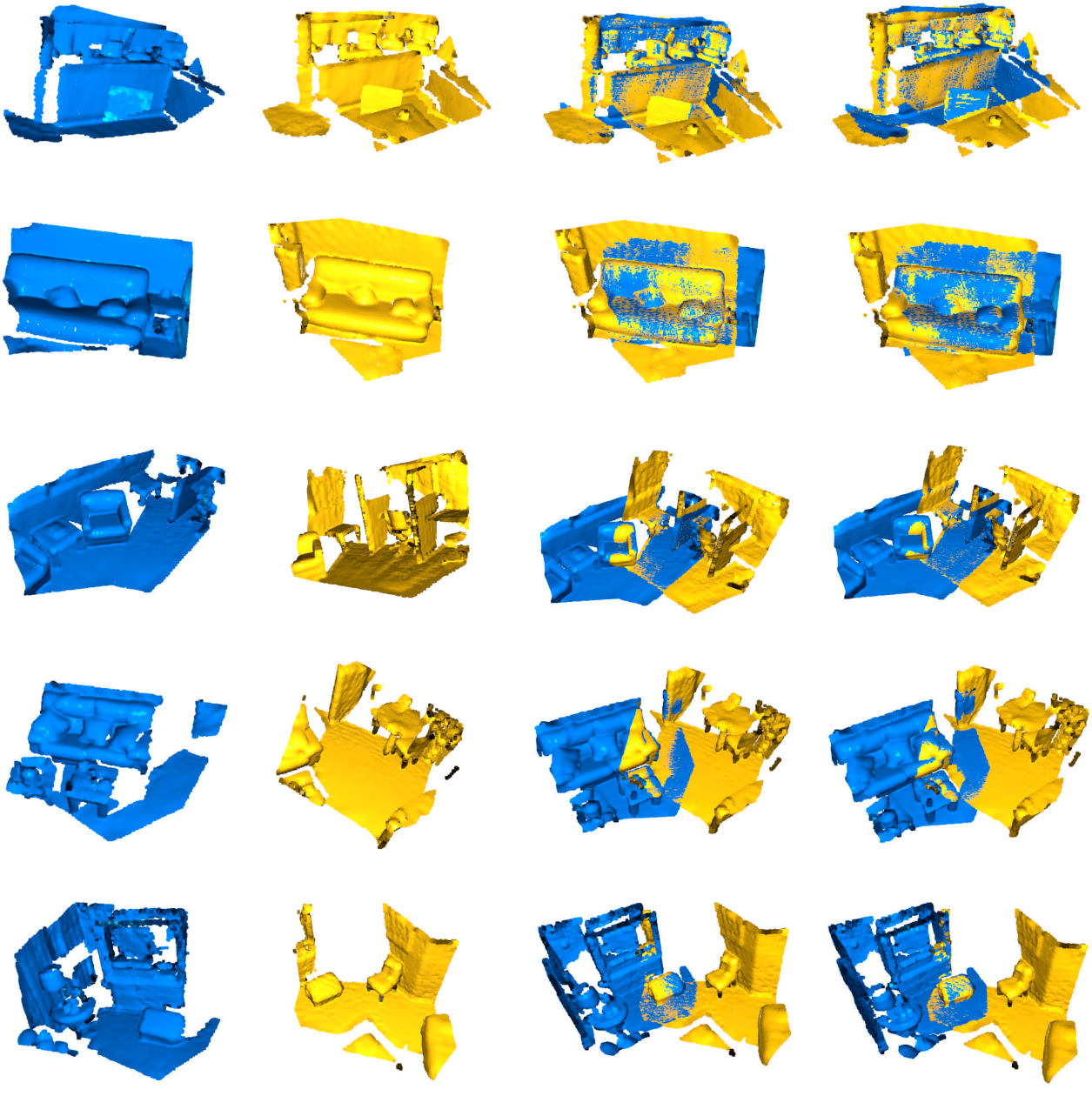


Figure 6. Visualization of five registration result examples on 3DMatch and 3DLoMatch. In each example, the source point cloud S, target point cloud T, ground truth, and registration result are represented from left to right. The first two rows display the results of high-overlap registration, and the last three rows display the results of low-overlap registration.

4.2. Experiments on ModelNet40

4.2.1. Dataset and Metrics

ModelNet40 contains 40 different categories of CAD models, each of which consists of training and test data collected from these CAD models. We used the processed dataset as described in [54], among which are 5112 models for training, 1202 models for validation, and 1266 models for testing. We performed the experiments on both ModelNet and ModelLoNet, where the overlap of point cloud pairs in ModelNet was ~70%, while in ModelLoNet it was ~50%. On ModelNet40, we reported the relative rotation error (RRE), the relative translation error (RTE), and the chamfer distance (CD) to evaluate our method [32].

RRE is the geodesic and Euclidean distance between the estimated and ground-truth rotation matrices, which measures the differences between the predicted and ground-truth rotation matrix, computed as follows:

$$\mathrm{RRE} = \arccos\left(\frac{\mathrm{trace}(\mathbf{R}^T \cdot \bar{\mathbf{R}}) - 1}{2}\right) \tag{25}$$

RTE is the Euclidean distance between the estimated and ground-truth translation vectors, which measures the differences between the predicted and ground-truth translation matrix, which is computed as follows:

$$\mathrm{RTE} = \|\mathbf{t} - \bar{\mathbf{t}}\|_2 \tag{26}$$

CD is the average nearest squared distance between the predicted and ground-truth point clouds, which measures the similarity between the predicted and ground-truth point clouds, defined as follows:

$$d_{CD}(\mathrm{P}, \mathrm{G}) = \frac{1}{\mathrm{P}} \sum_{\mathrm{P_x} \in \mathrm{P}} \min_{\mathrm{G}_y \in \mathrm{G}} \|\mathrm{P_x} - \mathrm{G}_y\|_2^2 + \frac{1}{\mathrm{G}} \sum_{\mathrm{G}_y \in \mathrm{G}} \min_{\mathrm{P_x} \in \mathrm{P}} \|\mathrm{G}_y - \mathrm{P_x}\|_2^2 \tag{27}$$

4.2.2. Registration Results

We compared the registration results of our method with those of five other methods on ModelNet40: PointNetLK [15], DCP [35], RPM-Net [32], Predator [54], and REGTR [40]. Three metrics (RRE, RTE, and CD) were used to measure the performance of ModelNet and ModelLoNet, as shown in Table 2. The left and right sides of Table 2 present the evaluation results for ModelNet and ModelLoNet, respectively. According to the evaluation on ModelNet, although the result with RTE was inferior to that of REGTR, it can be seen that STDRIF performed better than the other comparison methods with the RTE and CD metrics in high-overlap and low-overlap models.

Table 2. The registration results on ModelNet and ModelLoNet. The best two performances are highlighted in **bold** and underlined, respectively.

	ModelNet			**ModelLoNet**		
Method	**RRE**	**RTE**	**CD**	**RRE**	**RTE**	**CD**
PointNetLK [15]	29,275	0.297	0.0235	48.567	0.507	0.0367
DCP [35]	11.975	0.171	0.0117	16.501	0.300	0.0268
RPM-Net [32]	1.712	0.018	0.00085	7.342	0.124	0.0050
Predator [54]	1.739	0.019	0.00089	5.235	0.132	0.0083
REGTR [40]	**1.473**	0.014	0.00078	**3.930**	0.087	**0.0037**
STDRIF (ours)	1.507	**0.013**	**0.00073**	4.287	**0.086**	**0.0037**

Figure 7 presents the registration results on ModelNet and ModelLoNet. Figure 8 displays the results of a comparison experiment between our method and Predator on ModelNet with different overlap ratios using the RRE and RTE metrics. In Figure 7, it

can be seen that the registration results are close to the ground truth, which intuitively demonstrates the effectiveness of the proposed method. Figure 8 shows that the registration performance became worse with the decline in the overlap ratio; however, the results of our method were always superior to those of Predator with RRE and RTEe, which further proves that our method is competitive even for low-overlap registration.



Figure 7. Visualization of registration results on ModelNet and ModelLoNet. In each example, the input source point cloud S, input target point cloud T, ground truth, and registration result are represented from left to right. The first two rows display the results on ModelNet, and the last three rows display the results on ModelLoNet.

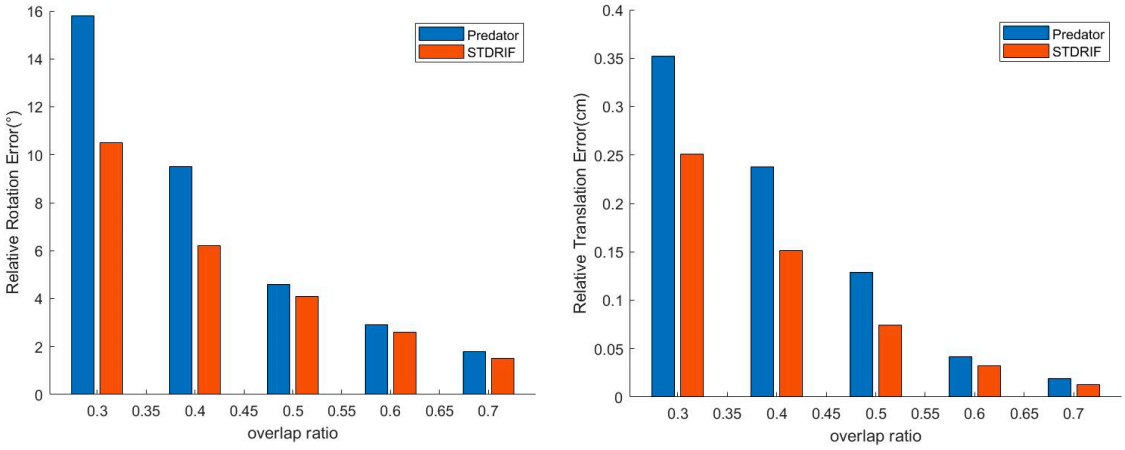


Figure 8. Comparison performance of STDRIF and Predator on ModelNet40 with different overlap ratios.

4.3. Experiments on KITTI

4.3.1. Dataset and Metrics

OdometryKITTI [30,31,44,54] includes 11 sequences of outdoor driving scenarios scanned using Velodyne laser LIDAR, among which sequences 0–5 are for training, sequences 6–7 are for validation, and sequences 8–10 are for testing. We used the method of [44,54] to refine the ground truth pose with ICP, and we evaluated our method with the point cloud pairs that are up to 10 m away from one another. The evaluation metrics used on KITTI are RRE, RTE, and RR; these three metrics have already been introduced in Sections 4.1.1 and 4.2.1 and they will not be repeated here.

4.3.2. Registration Results

A comparison of the registration results is shown in Table 3, and the comparison methods included five recent outstanding methods: 3DFeat-Net [58], FCGF [30], D3Feat [32], Predator [54], and Geotransformer [44]. With the RTE and RR metrics, we obtained the same performance as with Predator and Geotransformer. With RRE, the performance of our method was slightly inferior to that of Geotransformer. Overall, our method is still competitive in large scenes.

Table 3. The registration results on KITTI. The best two performances are highlighted in **bold** and underlined, respectively.

	KITTI		
Method	**RRE**	**RTE**	**RR**
3DFeat-Net [58]	0.35	25.9	96.0
FCGF [30]	0.30	9.5	96.0
D3Feat [32]	0.30	7.2	**99.8**
Predator [54]	0.27	**6.8**	**99.8**
Geotransformer [44]	**0.24**	**6.8**	**99.8**
STDRIF (ours)	0.26	**6.8**	**99.8**

Figure 9 displays the five examples of visualization results of registration on odometryKITTI, which intuitively shows the reliability of our approach. In Figure 8, each line exhibits an example of the registration result. In each example, the first two columns denote the source and target point clouds, respectively. The third column denotes the ground truth, and the last column denotes the registration result. From the comparison of the registration results and ground truth, it can be seen that the proposed method performs satisfactorily in large scenes.

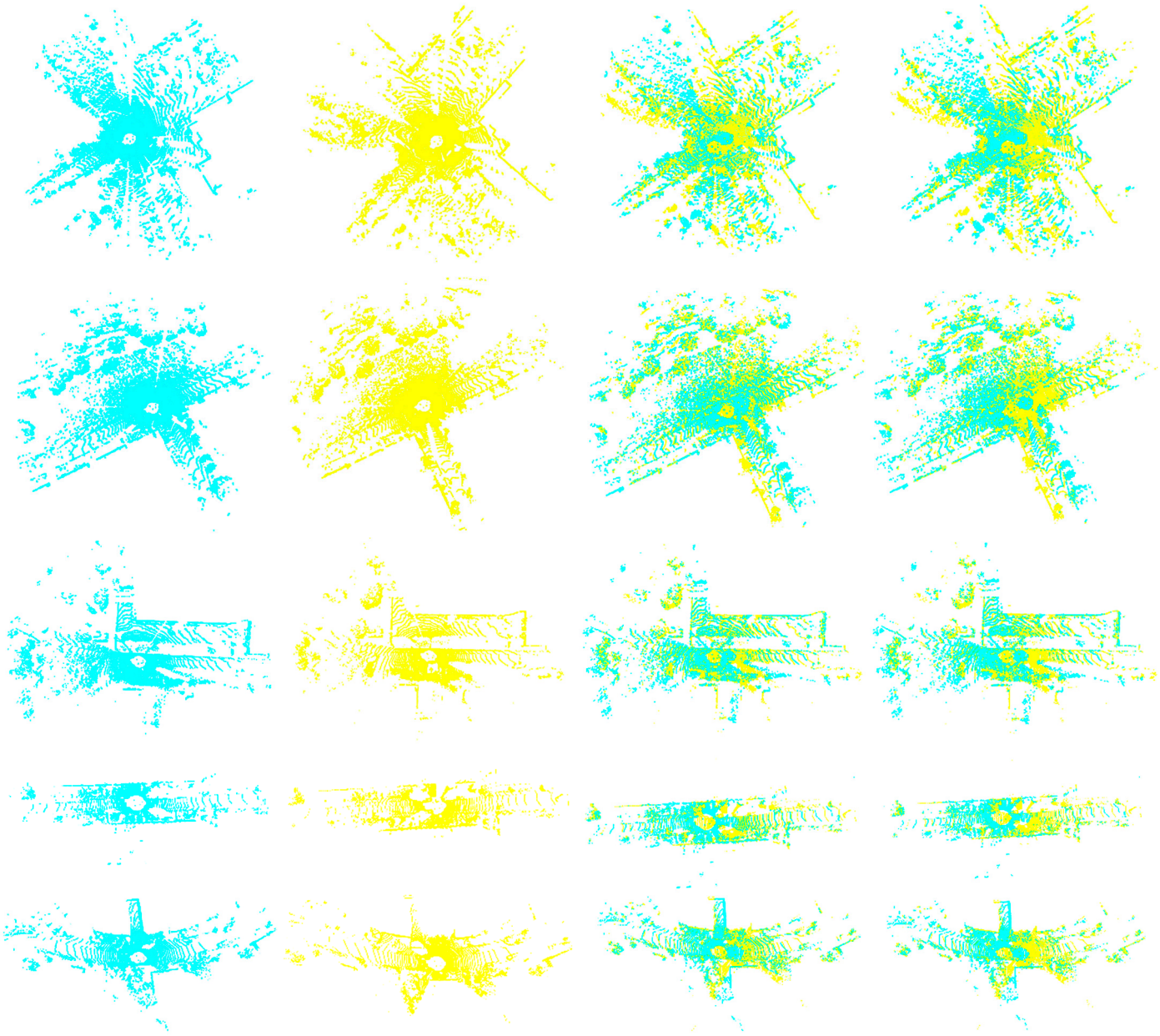


Figure 9. Visualization of registration results on KITTI. In each example, the input source point cloud S, input target point cloud T, ground truth, and registration result are represented from left to right.

4.4. Ablation Studies

Extensive ablation studies were conducted by applying one proposed submodule to the baseline at each time. Since the registration results of our method were close to those reported in [40], we chose [54] as the baseline to present the corresponding contribution of each submodule. Table 4 shows the extensive results on 3DMatch and 3DLoMatch. Our method benefits from three key components: PPF Transformer (PPF), sparse-to-dense matching (STD), and progressive alignment (PA). It can be seen that FMR, IR, and RR improved with each component used in the baseline, and the performance was better when submodules were used in superposition. The method using three submodules on the 3DMatch and 3DLoMatch datasets led to an increase of 3% and 3.7% in the RR metric, respectively, compared to the baseline with no submodules. Furthermore, using three submodules on the 3DMatch and 3DLoMatch datasets, the RR increased more obviously on

the 3DLoMatch, which demonstrates that our approach is more competitive for low-overlap point cloud registration tasks.

Table 4. Ablation study for each submodule, tested with 5000 correspondences. The best two performances are highlighted in **bold** and underlined, respectively. The √ and x indicate whether or not the module is in use, respectively.

Key Component			3DMatch			3DLoMatch		
PPF	**STD**	**PA**	**FMR (%)**	**IR (%)**	**RR (%)**	**FMR (%)**	**IR (%)**	**RR (%)**
x	x	x	96.7	66.9	89.4	87.2	40.1	72.2
√	x	x	97.1	67.4	89.9	88.1	41.4	72.8
√	√	x	97.2	70.4	90.8	88.2	42.5	73.9
√	x	√	97.5	71.4	91.1	88.7	43.1	74.3
√	√	√	**98.0**	**72.3**	**92.4**	**88.9**	**44.7**	**75.9**

5. Discussion

5.1. Performance and Effectiveness Analysis

In this study, several experiments were implemented on the 3DMatch, ModelNet40, and KITTI datasets. For all datasets, in order to show the excellent performance of our method, we reported the evaluation results with respective metrics and displayed the visualization of the registration results in the Sections 4.1–4.3. The evaluation results verify that the registration performance of our approach is superior to that of other recent excellent methods, even in large scenes, and the visualization of the registration results demonstrates the practicability and feasibility of our method. In the Section 4.4, ablation studies were conducted to explore the influence of each submodule with the FMR, IR, and RR metrics on 3DMatch. The value of the metrics increased with each submodule applied to the baseline, demonstrating the validity of the proposed submodules. Ablation studies on 3DLoMatch showed more significant performance improvements than on 3DMatch, demonstrating that our approach remains competitive in low-overlap scenes.

5.2. Limitations and Future Improvement Directions

A series of experiments have shown that STDRIF performs excellently with high accuracy and robustness, especially for low-overlap registration tasks. However, the limitations of our approach must be discussed below for the improvement of future work. In the above illustration, the PPF Transformer module was leveraged to learn rotation-invariant features. We focused on the point pair features in the self-attention stage to enrich contextual information, but we neglected the awareness of spatial positions when exchanging and aggregating information in the cross-attention stage. Moreover, we followed the method of [44] to downsample the input point clouds with the grid sampling method, which may lead to numerous superpoints and cause an increase in the calculation cost. The limitations mentioned above will be improved upon in future work. Except for the above details in need of improvement, we hope our method can cover a wider range of applications, such as non-rigid registration, cross-modality registration, etc.

6. Conclusions

In this paper, we introduced a point cloud registration network based on rotation-invariant features with a sparse-to-dense matching strategy, and the registration results benefit from three key submodules: PPF, STD, and PA. The PPF transformer module embeds point pair features into the transformer, which learns the distinctive hybrid features that make it invariant to rigid transformation. Sparse-to-dense matching establishes reliable correspondences from sparse to dense. The progressive alignment module further optimizes the registration by finding higher inlier ratio superpoints correspondences based on consistency. We conducted several experiments on the 3DMatch, ModelNet40, and KITTI datasets, and we showed the visualization and evaluation of the registration re-

sults. We further conducted ablation studies on the 3DMatch dataset and demonstrated the effectiveness of the three key submodules. All of the experimental results verify the excellent performance of our method, especially for low-overlap registration tasks and in large scenes. Finally, we discussed the limitations and future improvement directions. The remaining issues will be addressed in our future work for more robust and applicable point cloud registration.

Author Contributions: Conceptualization, T.M.; validation, T.M. and G.H.; formal analysis, T.M., Y.C. and H.R.; investigation, T.M., Y.C. and H.R.; original draft preparation, T.M.; review and editing, T.M., Y.C., H.R. and G.H.; visualization, T.M.; funding acquisition, G.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by the Department of Science and Technology of Jilin Province under Grant 20210201132GX.

Data Availability Statement: All datasets used in this work are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Eldefrawy, M.; King, S.A.; Starek, M. Partial Scene Reconstruction for Close Range Photogrammetry Using Deep Learning Pipeline for Region Masking. *Remote Sens.* **2022**, *14*, 3199. [CrossRef]
2. Dubé, R.; Gollub, M.G.; Sommer, H.; Gilitschenski, I.; Siegwart, R.; Cadena, C.; Nieto, J.I. Incremental-Segment-Based Localization in 3-D Point Clouds. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1832–1839. [CrossRef]
3. Chua, C.-S.; Jarvis, R. Point Signatures: A New Representation for 3D Object Recognition. *Int. J. Comput. Vis.* **1997**, *25*, 63–85. [CrossRef]
4. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; IEEE Computer Society: Washington, DC, USA, 2017; pp. 77–85.
5. Ali, S.A.; Kahraman, K.; Reis, G. RPSRNet: End-to-End Trainable Rigid Point Set Registration Network using Barnes-Hut 2D-Tree Representation. *arXiv* **2021**, arXiv:2104.05328.
6. Noh, J.; Lee, S.; Ham, B. HVPR: Hybrid Voxel-Point Representation for Single-stage 3D Object Detectio. *arXiv* **2021**, arXiv:2104.00902.
7. Besl, P.J.; McKay, N.D. Method for registration of 3-D shapes. In Proceedings of the Sensor Fusion IV: Control Paradigms and Data Structures, Boston, MA, USA, 12–15 November 1991; SPIE: Bellingham, WA, USA, 1992; Volume 1611, pp. 586–606.
8. Abdi, H. Singular value decomposition (SVD) and generalized singular value decomposition. *Encycl. Meas. Stat.* **2007**, *907*, 44.
9. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
10. Ao, S.; Hu, Q.; Yang, B.; Markham, A.; Guo, Y. SpinNet: Learning a General Surface Descriptor for 3D Point Cloud Registration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, Virtual, 19–25 June 2021; IEEE: New York, NY, USA, 2021; pp. 11753–11762.
11. Deng, H.; Birdal, T.; Ilic, S. Ppfnet: Global context aware local features for robust 3d point matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 195–205.
12. Li, J.; Zhang, C.; Xu, Z.; Zhou, H.; Zhang, C. Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration. In Proceedings of the 16th European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Cham, Switzerland, 2020; pp. 378–394.
13. Chen, Z.; Yang, F.; Tao, W. Detarnet: Decoupling translation and rotation by siamese network for point cloud registration. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 22 February–1 March 2022; pp. 401–409.
14. Gojcic, Z.; Zhou, C.; Wegner, J.D.; Wieser, A. The Perfect Match: 3D Point Cloud Matching with Smoothed Densities. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, 16–20 June 2019; IEEE: New York, NY, USA, 2019; pp. 5545–5554.
15. Aoki, Y.; Goforth, H.; Srivatsan, R.A.; Lucey, S. PointNetLK: Robust&Efficient Point Cloud Registration Using PointNet. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, 16–20 June 2019; IEEE: New York, NY, USA, 2019; pp. 7163–7172.
16. Wang, Y.; Solomon, J.M. Prnet: Self-supervised learning for partial-to-partial registration. *arXiv* **2019**, arXiv:1910.12240.
17. Yang, Z.; Pan, J.Z.; Luo, L.; Zhou, X.; Grauman, K.; Huang, Q. Extreme Relative Pose Estimation for RGB-D Scans via Scene Completion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, 16–20 June 2019; IEEE: New York, NY, USA, 2019; pp. 4531–4540.

18. Elbaz, G.; Avraham, T.; Fischer, A. 3D Point Cloud Registration for Localization Using a Deep Neural Network Auto-Encoder. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; IEEE Computer Society: Washington, DC, USA, 2017; pp. 4631–4640.
19. Choy, C.; Dong, W.; Koltun, V. Deep global registration. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, 13–19 June 2020; IEEE: New York, NY, USA, 2020; pp. 2514–2523.
20. Yang, J.; Li, H.; Campbell, D.; Jia, Y. Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 2241–2254. [CrossRef] [PubMed]
21. Segal, A.; Hhnel, D.; Thrun, S. Generalized-ICP. In *Robotics: Science and Systems V*; IEEE: Seattle, WA, USA, 2019; p. 435.
22. Parkison, S.A.; Gan, L.; Jadidi, M.G.; Eustice, R.M. Semantic Iterative Closest Point through Expectation-Maximization. In Proceedings of the British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, 3–6 September 2018; BMVA Press: Durham, UK, 2018; p. 280.
23. Biber, P.; Straßer, W. The normal distributions transform: A new approach to laser scan matching. In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, USA, 27 October–1 November 2003; IEEE: New York, NY, USA, 2003; pp. 2743–2748.
24. Rusu, R.B.; Blodow, N.; Beetz, M. Fast point feature histograms (FPFH) for 3D registration. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; IEEE: Kobe, Japan, 2009; pp. 3212–3217.
25. Aiger, D.; Mitra, N.J.; Cohen-Or, D. 4-points congruent sets for robust pairwise surface registration. *ACM Trans. Graph.* **2008**, *27*, 1–10. [CrossRef]
26. Tombari, F.; Salti, S.; Di Stefano, L. Unique signatures of histograms for local surface description. In Proceedings of the 11th European Conference on Computer Vision, Heraklion, Greece, 5–11 September 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 356–369.
27. Jian, B.; Vemuri, B.C. Robust Point Set Registration Using Gaussian Mixture Models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1633–1645. [CrossRef] [PubMed]
28. Eckart, B.; Kim, K.; Kautz, J. HGMR: Hierarchical Gaussian Mixtures for Adaptive 3D Registration. In Proceedings of the Computer Vision—ECCV 2018—15th European Conference, Munich, Germany, 8–14 September 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Proceedings, Part XV. Springer: Berlin/Heidelberg, Germany, 2018; Volume 11219, pp. 730–746.
29. Zeng, A.; Song, S.; Nießner, M.; Fisher, M.; Xiao, J.; Funkhouser, T.A. 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; IEEE Computer Society: Washington, DC, USA, 2017; pp. 199–208.
30. Choy, C.B.; Park, J.; Koltun, V. Fully Convolutional Geometric Features. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Republic of Korea, 27 October–2 November 2019; IEEE: New York, NY, USA, 2019; pp. 8957–8965.
31. Bai, X.; Luo, Z.; Zhou, L.; Fu, H.; Quan, L.; Tai, C.-L. D3Feat: Joint Learning of Dense Detection and Description of 3D Local Features. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, 13–19 June 2020; IEEE: New York, NY, USA, 2020; pp. 6358–6366.
32. Yew, Z.J.; Lee, G.H. Rpm-net: Robust point matching using learned features. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11824–11833.
33. Huang, X.; Mei, G.; Zhang, J. Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11366–11374.
34. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
35. Wang, Y.; Solomon, J.M. Deep closest point: Learning representations for point cloud registration. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3523–3532.
36. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural Information Processing Systems*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2017; p. 30.
37. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph.* **2019**, *38*, 1–12. [CrossRef]
38. Kabsch, W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallogr. Sect. A Cryst.Phys.Diffr. Theor. Gen. Crystallogr.* **1976**, *32*, 922–923. [CrossRef]
39. Fu, K.; Liu, S.; Luo, X.; Wang, M. Robust point cloud registration framework based on deep graph matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2021; pp. 8893–8902.
40. Yew, Z.J.; Lee, G.H. REGTR: End-to-end Point Cloud Correspondences with Transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, 18–24 June 2022; IEEE: New York, NY, USA, 2022; pp. 6667–6676.
41. Thomas, H.; Qi, C.R.; Deschaud, J.E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. Kpconv: Flexible and deformable convolution for point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6411–6420.

42. Wang, X.; Yuan, Y. GCMTN: Low-Overlap Point Cloud Registration Network Combining Dense Graph Convolution and Multilevel Interactive Transformer. *Remote Sens.* **2023**, *15*, 3908. [CrossRef]
43. Bai, X.; Luo, Z.; Zhou, L.; Chen, H.; Li, L.; Hu, Z.; Fu, H.; Tai, C.L. Pointdsc: Robust point cloud registration using deep spatial consistency. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 15859–15869.
44. Qin, Z.; Yu, H.; Wang, C.; Guo, Y.; Peng, Y.; Xu, K. Geometric Transformer for Fast and Robust Point Cloud Registration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, 18–24 June 2022; IEEE: New York, NY, USA, 2022; pp. 11133–11142.
45. Yu, H.; Qin, Z.; Hou, J.; Mahdi, S.; Li, D.; Benjamin, B.; Slobodan, I. Rotation-Invariant Transformer for Point Cloud Matching. *arXiv* **2023**, arXiv:2303.08231.
46. Taewon, M.; Chonghyuk, S.; Eunseok, K.; Inwook, S. Distinctiveness oriented positional equilibrium for point cloud registration. In Proceedings of the IEEE/CVF Conference on Computer Vision, Montreal, Canada, 17 October–27 October 2021; pp. 5490–5498.
47. Yang, F.; Guo, L.; Chen, Z.; Tao, W. One-inlier is first: Towards efficient position encoding for point cloud registration. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022; pp. 6982–6995.
48. Li, Y.; Harada, T. Lepard: Learning partial point cloud matching in rigid and deformable scenes. In Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, 18–24 June 2022; IEEE: New York, NY, USA, 2022; pp. 5554–5564.
49. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, Virtual, 19–25 June 2021; IEEE: New York, NY, USA, 2021; pp. 2117–2125.
50. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Houlsby, N. An image is worth 16 $\times$ 16 words: Transformers for image recognition at scale. *arXiv* **2010**, arXiv:2010.11929.
51. Sun, J.; Shen, Z.; Wang, Y.; Bao, H.; Zhou, X. LoFTR: Detector-Free Local Feature Matching with Transformers. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, Virtual, 19–25 June 2021; IEEE: New York, NY, USA, 2021; pp. 4267–4276.
52. Yu, H.; Li, F.; Saleh, M.; Busam, B.; Ilic, S. Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 23872–23884.
53. Drost, B.; Ulrich, M.; Navab, N.; Ilic, S. Model globally, match locally: Efficient and robust 3D object recognition. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; IEEE: San Francisco, CA, USA, 2010; pp. 998–1005.
54. Huang, S.; Gojcic, Z.; Usvyatsov, M.; Wieser, A.; Schindler, K. Predator: Registration of 3D Point Clouds with Low Overlap. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, Virtual, 19–25 June 2021; IEEE: New York, NY, USA, 2021; pp. 8922–8931.
55. Sinkhorn, R.; Knopp, P. Concerning nonnegative matrices and doubly stochastic matrices. *Pac. J. Math.* **1967**, *21*, 343–348. [CrossRef]
56. Sun, Y.; Cheng, C.; Zhang, Y.; Zhang, C.; Zheng, L.; Wang, Z.; Wei, Y. Circle loss: A unified perspective of pair similarity optimization. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, 13–19 June 2020; IEEE: New York, NY, USA, 2020; pp. 6398–6407.
57. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1912–1920.
58. Yew, Z.; Lee, G. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In Proceedings of the European conference on computer vision, ECCV 2018, Munich, Germany, 8–14 September 2018; pp. 607–623.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.