



Article VPRNet: Virtual Points Registration Network for Partial-to-Partial Point Cloud Registration

Shikun Li¹, Yang Ye², Jianya Liu³ and Liang Guo^{1,*}

- ¹ School of Mathematics & Statistics, Shandong University, Weihai 264209, China; shikun.li@mail.sdu.edu.cn
- ² College of Literature, Science, and the Arts, University of Michgan, Ann Arbor, MI 48109, USA;
 - yanye@umich.edu
- ³ Data Science Institute, Shandong University, Jinan 250100, China; jyliu@sdu.edu.cn
- * Correspondence: liangguo@sdu.edu.cn; Tel.: +86-182-5313-8503

Abstract: With the development of high-precision and high-frame-rate scanning technology, we can quickly obtain scan data of various large-scale scenes. As a manifestation of information fusion, point cloud registration is of great significance in various fields, such as medical imaging, autonomous driving, and 3D reconstruction. The Iterative Closest Point (ICP) algorithm, as the most classic algorithm, leverages the closest point to search corresponding points, which is the pioneer of correspondences-based approaches. Recently, some deep learning-based algorithms witnessed extracting deep features to compress point cloud information, then calculate corresponding points, and finally output the optimal rigid transformation like Deep Closest Point (DCP) and DeepVCP. However, the partiality of point clouds hinders the acquisition of enough corresponding points when dealing with the partial-to-partial registration problem. To this end, we propose Virtual Points Registration Network (VPRNet) for this intractable problem. We first design a self-supervised virtual point generation network (VPGnet), which utilizes the attention mechanism of Transformer and Self-Attention to fuse the geometric information of two partial point clouds, combined with the Generative Adversarial Network (GAN) structure to produce missing points. Subsequently, the following registration network structure is spliced to the end of VPGnet, thus estimating rich corresponding points. Unlike the existing methods, our network tries to eliminate the side effects of incompleteness on registration. Thus, our method expresses resilience to the initial rotation and sparsity. Various experiments indicate that our proposed algorithm shows advanced performance compared to recent deep learning-based and classical methods.

Keywords: virtual points; partial-to-partial; transformer; GAN

1. Introduction

Point cloud registration is a fundamental task that has been widely used in many computational fields, such as object pose estimation [1], SLAM [2], and 3D reconstruction [3]. In its most common incarnation, point correspondence estimation and rigid transformation computation, including rotation and translation, trivializes the problem, which is possibly misled by noise and partiality.

Iterative Closest Point (ICP) [4], as the most representative method, is the gold standard for solving registration problems. It iteratively obtains the point correspondences by nearest neighbor search and estimates the rigid transformation by Singular Value Decomposition (SVD). The ICP algorithm does not require any prior information about the original point clouds. However, the convergence to global minimum puts forward strict requirements for the initial poses because the accuracy and locality of convergence depend heavily on the proportion of the overlapping area [5,6]. Besides, the cover of noise and outliers also prevent the estimation of rigid transformation. Therefore, many works are proposed to overcome the blemish of ICP [7–10]. The Point-to-Plane ICP algorithm [9] modifies the



Citation: Li, S.; Ye, Y.; Liu, J.; Guo, L. VPRNet: Virtual Points Registration Network for Partial-to-Partial Point Cloud Registration. *Remote Sens.* 2022, 14, 2559. https://doi.org/ 10.3390/rs14112559

Academic Editor: Sander Oude Elberink

Received: 12 March 2022 Accepted: 20 May 2022 Published: 27 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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/). cost function from the point-to-point distance to the point-to-plane distance. After finding the closest point, the distance of the closest point is narrowed along the normal direction of the fitted plane. GO-ICP algorithm [11] combines the ICP with the branch-and-bound algorithm to avoid local minima when ICP hovers around the local minimum. Despite the improved performance, the above methods are also sensitive to the initial poses [12].

Recently, advances in deep feature extraction such as PointNet [13] and PointNet++ [14] have been proposed to make it possible for neural networks to directly process disordered points without prior projection [15]. Seminal PointnetLK [16] and DeepLocalization [17] use PointNet to extract global features for registration. Although PCRnet [18] follows a similar policy as PointnetLK, which is a catenation of extracting global embeddings and calculating transformation parameters, it completely relies on neural networks to output regression value rather than an iterative process. Despite the simplicity of global feature-based methods, the global feature vector cannot retain sufficient information for accurate registration when faced with the registration problem of large-scale point clouds [19]. Besides, non-overlapping regions obscure the effectiveness of collected information [12]. Consequently, another class of deep learning-based methods is proposed. Correspondence-based approaches extracting keypoints and correspondence weights firstly, then SVD or MLP (Multilayer Perceptron) is employed to return rigid transformation, such as 3DFeat-Net [20] or DeepVCP [21]. More recently, Deep Closest Point (DCP) [22] incorporates the attention mechanism of the Transformer into the network structure. Essentially, the final hybrid feature is a fusion of two original point clouds. Furthermore, PRnet [23] employs the Gumbel–Softmax Sampler to sample a matching matrix and Actor-Critic Closest Point (ACP) to adjust the "temperatures" of the mapping function of DCP. However, these methods pay more attention to the overlapping regions. The collected corresponding points are still limited due to partiality, which blocks a correspondences-based algorithm for the partial-to-partial point cloud registration problem. OMNet [12] proposes to use deep learning to predict the mask of overlapping regions to erase the matching difficulties caused by shape differences.

In this paper, we propose a novel network called VPRNet, a deep learning network for partial-to-partial point cloud registration. We first generate virtual points to remove the barriers of different partiality ratios of original point clouds. A self-supervised strategy is proposed to extract hybrid features without extra labeled data. Then attention mechanism of the Transformer and Self-Attention (SA) is included in the structure to highlight overlapping regions during feature extraction. Combining virtual points generation with preferred hybrid feature compression profits conjugate points, which desensitizes initial rotation and strengthens the capacity to match partial point clouds. Categorically, we utilize the Generative Adversarial Network (GAN) in VPGnet to generate optimal missing parts and merge them with the original partial point cloud to ensure that the shape information is enriched without destroying the original point cloud geometry. Various experiments indicate that our methods achieve advanced performance compared to advanced deep learning-based and traditional methods. Our main contributions are:

- A self-supervised virtual point generation network (VPGnet) based on GAN is proposed. The VPGnet focuses on the shape information of point clouds and can effectively complete the partial point cloud.
- A combination strategy of virtual point generation and corresponding point estimation is proposed, which can reduce the negative effect of partiality during registration.
- Various experiments demonstrate the advanced performance compared to other advanced approaches.

The rest of this paper is organized as follows: Section 2 reviews previous literature. Section 3 describes the architecture of our proposed network. Experiments are performed in Section 4. The discussion of the experimental results is shown in Section 5. Finally, Section 6 makes a precise summary of our work.

2. Related Work

Point cloud registration aims to find a rigid transformation matrix, including rotation matrix and translation vector, then apply this transformation to align the source point cloud to the target point cloud. In the past few decades, many pieces of literature proposed solutions to this fundamental task. Taking time as the border, we divide the method of solving point cloud registration into traditional and deep learning-based methods. Before 2017, most scholars focused more on conventional methods because of the sparsity and disorder of point cloud. After 2017, benefiting from the landmark PointNet [13] and PointNet++ [14], a large number of researchers tend to the deep learning-based methods [24]. The following text summarizes methods for point cloud registration from the above two aspects.

2.1. Traditional Methods

The most seminal method for solving registration problems is ICP [4]. This algorithm switches from finding correspondences and updating a rigid transformation matrix in a coarse-to-fine manner. Specifically, After obtaining the corresponding point sets, the ICP algorithm employs the least-square method to solve the transformation parameters. ICP can obtain accurate registration results as a fine registration algorithm, but some shortcomings deserve attention. The ICP algorithm needs a good initial value as input, or it is easy to converge to the local optimum [25]. Consequently, the registration accuracy of the ICP algorithm depends heavily on the overlap rate of point clouds [5,26]. Besides, the ICP algorithm requires many iterations to find the optimal corresponding point pair, which is time-consuming [27,28].

The above two drawbacks prohibit the application of the ICP algorithm in real-time and large-scale scenarios. Thus, some scholars proposed solutions. On the one hand, benefiting from the fact that the coarse registration has no hypothesis on the initial poses of point clouds, employing the result of coarse registration as the initial value of the ICP algorithm has become the consensus of the registration task [29]. A popular program utilizes the RANSAC method to find the corresponding triples [30]. The complexity of the RANSAC algorithm regularly degrades to its worst-case $O(n^3)$ complexity in the number n of data samples [29,31]. As improvements to RANSAC, the 4 Points Congruent Sets(4PCS) algorithm [32] and Super 4PCS algorithm [31] intelligently ameliorate the registration process with four selected point pairs instead of three, making the computational complexity reach $O(n^2)$ and O(n), respectively. Moreover, Super Edge 4PCS utilizes the edge of point clouds to finish the registration, thus greatly reducing the running time [33]. On the other hand, some ICP-variant algorithms were proposed, including distances defined as the point to plane [9,34], point to triples [35], and plane to plane [36]. In addition to changing the objective function, improving the search strategy is also a meaningful improvement. Eggert [37] and Vlaminck [38] employed two search strategies, kd-tree and Octree, to speed up the corresponding acquisition. These classical methods are still either easy to fall into local optimal values or time-consuming, which limits the application in large-scale scenarios that require real-time registration [18].

2.2. Learning-Based Methods

Learning-based methods have been gradually been accepted since 3Dmatch [39] was proposed in 2017. After PointNet [13] and PointNet++ [14], scholars can directly employ convolutional neural network to deal with disordered points directly. Therefore, the deep learning methods achieve considerable development. Correspondence-based methods and correspondence-free methods construct two main branches of learning-based methods [24].

2.2.1. Correspondence-Free Methods

The critical step of correspondence-free methods is regressing the global high-dimensional features generated by the deep neural network and outputting the rigid transformation parameters. The PoinetnetLK [16] modifies the traditional Lucas and Kanade (LK) algorithm and unrolls with PointNet into a trainable deep network framework. However, this method

affords many derivation theories instead of simply concatenating the global features to solve R and t, which inevitably causes low computational efficiency [18,19]. As an intelligent improvement to PoitnetLK, PCRnet [18] replaces the approximation of Jacobian as a data-driven technique that is a deep feature alignment layer to output transformation parameters directly. Although PCRnet has improved efficiency and robustness compared to PointnetLK, the latter shows better generalization capabilities across various object categories [18]. Feature-Metric Registration (FMR) believes that the extracted features of point clouds with different poses are different. The transformation is iteratively solved by calculating the differences of global features [40]. Although the above correspondence-free methods straightforwardly follow an end-to-end network architecture, the performances depend heavily on the feature extraction block [24]. Likewise, we follow the analogous end-to-end network architecture but mix the extracted embeddings and other geometric information of opposite point clouds together.

2.2.2. Correspondence-Based Methods

Compared with the straightforward structure of the correspondence-free method, the correspondence-based method often possesses a more complex network architecture. Although employing voxels to represent point clouds and network training is not as popular as the PointNet-based methods due to the vast memory requirement and lost quality [41], some relative voxel-representation methods are still worth discussing [24]. 3DMatch [39] maps the local area that wraps the interest points to a 512-dimensional feature vector as a pioneering approach. Besides, the Perfect Match [42] employs Smoothed Density Value (SDV) voxelization to extract features computed with the Gaussian smoothing kernel. Recently, Huang et al. [43] designed an overlapping attention module in the feature coding stage for early information exchange, which improves the accuracy of registration and is suitable for low overlap scenes.

Inspired by the PointNet framework, PPFnet [44] defines point pair features, including point pairs' coordinates and normals to describe the local regions oriented 3D points. The feature processing leads to a rotation invariance while depending on the estimation of normal excessively [24]. Another representative method of employing the PointNet++ is DeepVCP [21], which utilizes the mini-PointNet++ [44,45] composed of three consecutively stacked fully connected layers and max-pooling layers to extract features and avoid the interference of dynamic targets. The generated corresponding point boosts registration accuracy [21]. In addition to feature extraction, outliers rejection also leverages PointNet's advantage. 3DRegnet [46] affords classification block and registration block, which extend the deep ResNet [47] to extract meaningful features and eliminate incorrect correspondences. However, none of the above methods pays attention to the corresponding points in the non-overlapping area, which influences the accuracy of corresponding points [12].

2.3. Under Partial Overlap

Among point cloud registration tasks, partially overlapping assignments pose a considerable challenge to deep learning methods due to the drastic differences in the global information [24]. Consequently, some algorithms mainly focus on partial-to-partial registration. As a successful case of applying the attention mechanism to registration, Deep Closest Point (DCP) [22] employs the Transformer [48] to absorb information from two point clouds and generates corresponding point pairs via soft pointers. However, the mapping \mathcal{M} produces blurred correspondences in exchange for this differentiability. PRnet [23] extends the DCP algorithm to an iterative pipeline and utilizes Gumbel–Softmax sampling to define a sharp mapping function that accepts backpropagation. A corresponding point generation method similar to DCP appears in [49]. RPMnet proposes a subnetwork to predict annealing parameters and utilizes these two parameters and sinkhorn normalization to generate a match matrix [41]. The above methods contain no targeted measures to deal with partial point clouds. Paying attention to the negative effect of non-overlapping points, OMNet [12] learned the overlapping mask and achieved state-of-the-art performance. Song et al. propose a novel partial point cloud registration network that employs the graph attention module to predict key points [50]. Similarly, Eduardo et al. apply a RANSAC procedure after correspondence matching [51]. These generation processes of corresponding point pairs include only the information fusion of features mapped from original partial point clouds. Based on a brand-new idea, we propose an end-to-end network framework called VPRNet, which includes virtual point generation (VPGnet) and registration (Regnet). VPR-Net utilizes the GAN architecture to continuously generate missing points and applies an attention mechanism to weighted correspondences, ensuring the corresponding quality.

3. Method

3.1. Overview

Our VPRNet is divided into two parts: VPGnet and Regnet. VPGnet is designed to generate virtual points, and the Regnet registers the completed point clouds. Figures 1 and 2 show the framework of VPGnet and registration network, respectively. Structurally speaking, this algorithm contains the framework of GAN using a self-supervised training strategy since the ground-truth missing part is separated from the original complete point cloud, and no auxiliary labeled data is added to the training process. The generator and discriminator confront each other until the discriminator cannot judge whether the virtual point generated by the generator is a ground-truth point or a fake one. The generator extracts the features of the three groups of sampled point clouds with PointNet and DGCNN. Then, the core Transformer and Self-Attention (SA) combine two hybrid features from original point clouds preferentially. Finally, missing parts are generated by MLP and Reshape operations. As for the Regnet, it first combines the virtual points generated by VPGnet with the original point clouds to complete point clouds X and Y. We then convert X into X', Y into Y' according to the rotation matrix R_{i-1} , and translation vector t_{i-1} from the previous iteration. After extracting the hybrid features from X' and Y', the probability volume is calculated according to the feature processed by softmax. Then, the corresponding matrix Σ is obtained as the weighted sum of probability and point coordinates. Finally, the SVD module is applied to generate the new rotation matrix R_i and translation vector t_i .



Figure 1. Architecture of our VPGnet. The self-supervised network is mainly composed of two parts, the generator and the discriminator. The generator sub-network extracts features through Self-Attention and Transformer, then MLP and Reshape operations are used to generate virtual points. Next, the features of the generated points and ground-truth are extracted through the DGCNN of the Discriminator and compared with each other. Finally the probability that the input point cloud is the ground-truth is output.



Figure 2. Architecture of our registration network. The iterative network first applies the rotation and translation calculated in the previous iteration to the input cloud. Through two main components of feature extraction, including Transformer and corresponding matrix acquisition, the network obtains the rigid transformation of the current iteration through SVD.

3.2. VPGnet Architecture

3.2.1. Multi-Resolution Feature Extraction

The first step is to represent the point cloud as embedded features. Deng et al. [52–54] perform convolution operations on the entire point cloud and then duplicate the global features *n* times, where *n* is the number of points. Finally, the mixed point feature is formed by concatenating the global features and local features. There is no transition between point coordinates and global information despite the simplicity. Correspondingly, Qi et al. [13,55] pointed out that the local and global features extracted from different scales can describe the point cloud more efficiently. Consequently, we employ the multi-resolution feature extraction architecture proposed in PointNet++. As shown in Figure 1, we first perform the Farthest Point Sampling (FPS) on the source point cloud and target point cloud. Enlightened by the LRANet [56] and PF-Net [57], we performed FPS three times on the original point clouds. Then, the shared DGCNN encodes the points and their neighbor into latent vector F_l^i where $i \in [1, 3]$ is the scale number, and $l \in [1, 5]$ is the index of convolution. DGCNN integrates the local neighbor information of the point cloud, which is not available in PointNet [22]. After four convolution layers, the dimensions of the feature vector are [64, 64, 128, 256]. Before the fifth convolution layer, the four feature vectors are concatenated together to obtain a 512-dimensional latent vector. Subsequently, we pass this latent vector into the fifth convolution layer to get a 1024-dimensional feature vector F_5 . Putting all F_1^i together, we get a 3 \times 1024 latent map.

In addition to the local embeddings, we expect that the embeddings can focus on the entire information of point clouds, not limited to the neighborhood of a certain point. Therefore, we choose the PointNet [13] architecture to obtain the global information of input point clouds. The points are encoded into multiple dimensions [64, 128, 256, 512, 1024]. After the Max-pooling operation, we can obtain 1024-dimensional global features F_g . The combination of F_g and F_l^i can juggle the details and overall information. The feature encoding process can be summarized as:

$$F_g = pointnet(x) \tag{1}$$

$$F_l = \{\mathcal{D}[FPS_i(x)]\} \oplus 3 \tag{2}$$

where \mathcal{D} is DGCNN, $x \in \mathcal{R}^{n \times 3}$ is the original point cloud, and n is the point number in x; *FPS*_i is the i-th farthest point sampling with a different sampling size; F_g and $F_l^i \in \mathcal{R}^d$

represent the global and local features, here d is 1024; $\oplus m$ means to repeat the DGCNN and FPS operations *m* times, and stitch the obtained vectors together.

3.2.2. Attention

Both input point clouds suffered from a deficiency of geometric attributes. Thus, we design to employ the shape information of one point cloud to complete the other. Thus, the particular embeddings from two point clouds need to be merged instead of separately decoding the two independent latent maps. Inspired by a recent article [58], we attached two attention mechanisms to change the encoder's attention: Transformer and Self-Attention (SA) modules.

The Transformer is the first composition model that relies on SA to calculate input and output representation [48]. It is first used in natural language processing (NLP) to solve the sequence-to-sequence problem, such as the machine translation task. The Transformer consists of an encoder module and decoder module. Each encoder module and decoder module are stacked with separate sub-encoders and sub-decoders. The encoders are all the same in structure, including two parts: SA and feed-forward neural network. The SA module can help the current element combine the context semantics. Compared with the encoder, the decoder focus on the relevant part of the input sequence. Reviewing the complete encoding and decoding process: input the embedding E_1 obtained after position encoding of the sequence S_1 into the encoder, then output a new embedding E'_1 after SA. There is a residual connection in the sublayers of each encoder, so the output of the encoder is $E_1 = E_1 + E'_1$. In the decoding process, the new E_1 is first decoded to obtain the sequence S_2 , which is then encoded by the decoder and merged with E_1 to output a new sequence S_3 .

We draw inspiration from applying Transformer to solve sequence2sequence problems: Transformer combines two sequences so that the encoder and decoder module can learn co-context information. Consequently, we utilize the Transformer as the first attention method to supplement semantic information of one point cloud to the other. The calculation of the Transformer can be summarized as the following equations:

$$\Theta_x = M_x + \Omega(M_x, M_y)$$

$$\Theta_y = M_y + \Omega(M_y, M_x)$$
(3)

Assuming that the latent maps obtained from the input point clouds are M_x and M_y , where $M_x = \{F_g^x, F_{l1}^x, F_{l2}^x, F_{l3}^x\} \in \mathcal{R}^{r \times d}$, r is the number of latent vectors obtained by DGCNN and PointNet (here is 4), and d is the dimension of latent vectors. Θ_x and Θ_y are the high-dimensional result feature output by Transformer $\Omega \in \mathcal{R}^{r \times d}$. It is worth noting that Ω is not a symmetric function: $\Omega(x, y) \neq \Omega(y, x)$. The decoder realizes meaningful fusion of the contained information from two sequences.

However, this scheme has a premise that we need to know the missing parts of the current point cloud before completing the current point cloud. Therefore, we leverage a separate Self-Attention as a sibling attention mechanism with Transformer, aiming at making the point cloud aware of its distinctive shape. The structure of SA can be described by the equation shown below:

$$\Phi_x = M_x + \mathcal{C}(M_x - \mathcal{C}_v(x) \cdot \phi(\mathcal{C}_q(x) \cdot \mathcal{C}_k(x)))$$
(4)

where ϕ represents the softmax function, C_q and C_k are the convolutions needed to generate query and key vectors. These two vectors are employed to score the high-dimensional feature vectors generated from the coordinates of other points in the point cloud. The scores determine the amount of feature expressions. These scores are multiplied by the value vector generated by C_v to distract attention from the points with less correlation. The entire SA process also follows the structure of residual connections. The latent map obtained by the Transformer is subjected to a Max-pooling layer of [1024–512] and then concatenated with the embedding vectors obtained by SA. Thus, we finally obtain a latent map with dimensions of 1536. The following continuous MLP layers encode the embedding vectors obtained by the attention mechanism into dimensions of 192, so that the final reshape layer can output 256 virtual points.

3.2.3. Discriminator

As another important component of GAN, Discriminator is used to judge the virtual points generated by Generator. Its working mode can be described as:

$$p = \mathcal{L}_3(\varphi(\mathcal{L}_2(\varphi(\mathcal{L}_1(\xi(\mathcal{D}(x)))))))$$
(5)

where *x* represents input point cloud generated virtual points or rendered ground truth, D is the DGCNN operation, ξ is the Maxpooling layer, φ is the Leaky_Relu activation function, and \mathcal{L}_i represents the Linear layer. Discriminator takes virtual points generated by the Generator and ground-truth missing point clouds as inputs, and outputs the predicted probability that the received point cloud is ground-truth. It calculates the adversarial loss between the predicted and the actual label and then feeds it back to the generator. Repeating the above game process until the probability that the predicted label is the virtual point is close to 0.5, means that the discriminator cannot tell the difference between the input point cloud and ground-truth.

3.3. Regnet Architecture

3.3.1. Correspondences Calculation

After obtaining the virtual points, it is first combined with the original points to form complete point clouds PC_g . Then, applying the rotation matrix R and translation vector t generated from the previous iteration to PC_g , we get a new input of the current iteration. Next, DGCNN and Transformer are used to extract and fuse features similar to VPGnet. The Transformer in Regnet enforces the encoder to pay more attention to the spatial information of another point cloud, that is, the orientation and position of the point cloud. The dimension of the embedding vectors obtained after the Transformer is $n \times 1024$, where n is the number of points in the point cloud. In order to obtain the corresponding points in the target complete point cloud, we calculated the correlation between each point in two combined point clouds PC_g^X and PC_g^Y , which is expressed as:

$$\Sigma = \phi(\Theta_x \cdot \Theta_y^{-1}) \tag{6}$$

where Θ_x and $\Theta_y \in \mathcal{R}^{n \times 1024}$ denote the high-dimensional feature maps after Transformer. The dimension of Σ is $n \times m$, where n and m are the scales of source and target point clouds, respectively. Each element Σ_{ij} represents the correlation between the *i*-th point in the source complete point cloud PC_g^x and the *j*-th point in the target complete point cloud PC_g^y . Then, the corresponding points in the target point cloud are calculated as $\Sigma \cdot PC_g^y$.

3.3.2. SVD Module

Now, for each point x_i in the source complete point cloud, there are *m* corresponding points y_j weighted in the target complete point cloud. Therefore, in order to reduce the burden of network training, we employ the SVD module to calculate the final rotation matrix R_{xy} and translation vector t_{xy} . We define the centroids of PC_g^x and PC_g^y as:

$$\overline{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \quad \text{and} \quad \overline{y} = \frac{1}{M} \sum_{j=1}^{M} y_j \tag{7}$$

The covariance matrix can be expressed as:

$$\mathbf{H} = \sum_{i=1}^{N} (x_i - \overline{x})(y_i - \overline{y})$$
(8)

Then, singular value decomposition is performed on $\mathbf{H} \in \mathcal{R}^{3 \times 3}$:

$$\mathbf{H} = \mathbf{U}\mathbf{S}\mathbf{V}^{-1} \tag{9}$$

where **U** and $\mathbf{V} \in SO(3)$ are the matrices formed by the eigenvectors of \mathbf{HH}^T and $\mathbf{H}^T\mathbf{H}$, respectively. **S** is a diagonal matrix whose diagonal elements are eigenvalues of H. Finally, the rotation matrix R_{xy} and the translation vector t_{xy} can be calculated according to Equation (10):

$$\mathbf{R}_{xy} = \mathbf{V}\mathbf{U}^{-1}$$
 and $\mathbf{t}_{xy} = -\mathbf{R}_{xy}\cdot\overline{x} + \overline{y}$ (10)

3.4. Loss Functions

The first loss function is the adversarial loss of Discriminator L_d in VPGnet. We consider four groups of adversarial losses, which are the ground-truth x point clouds, the generated x virtual points, the ground-truth y point cloud, and the generated y virtual points, so L_d is:

$$L_{d} = L_{d}^{gx} + L_{d}^{vx} + L_{d}^{gy} + L_{d}^{vy}$$
(11)

Each L_d^j is defined as:

$$L_d^j = -\frac{1}{N} \sum_{i=1}^N [D(GT_i) * \log(D(GT_i)) + (1 - D(GT_i)) * \log(1 - D(G(x_i)))]$$
(12)

where x_i is the *i*-th point cloud, GT_i is the *i*-th ground-truth missing point cloud, and N is the number of input point clouds. D() and G() represent the Descriminator and Generator.

Fan et al. proposed two position-invariant metrics to calculate the distance between two point clouds: Chamfer Distance (CD) and Earth Mover's Distance (EMD). CD calculates the average closest point distance between two input point clouds, which is shown as Equation (13). The first term represents the sum of the minimum distance from any point xin S_1 to S_2 . The second term serves the symmetric role. The two sets S_1 and S_2 do not need to be the same size. EMD was first proposed in [59] as a histogram similarity measure based on transportation efficiency. It calculates the minimum distance from one distribution to another. Unlike CD, the calculation of EMD requires that the two sets S_1 and S_2 have the same size. The calculation method is shown in Equation (14):

$$d_{CD}(S_1, S_2) = \frac{1}{S_1} \sum_{x \in S_1} \min_{y \in S_2} ||x - y||_2^2 + \frac{1}{S_2} \sum_{y \in S_2} \min_{x \in S_1} ||y - x||_2^2$$
(13)

$$d_{EMD}(S_1, S_2) = \min_{\varphi: S_1 \to S_2} \frac{1}{|S_1|} \sum_{x \in S_1} \| x - \varphi(x) \|$$
(14)

We calculate the CD and EMD between the missing parts of the virtual point clouds and the ground truth. Apart from that, the CD between the combined point clouds and the ground-truth complete point cloud are employed to ensure that the former has a similar shape and structure to the latter. Therefore, the loss function of the Generator can be summarized as follows:

$$L_g = L_g^X + L_g^Y \tag{15}$$

$$L_{g}^{X} = d_{CD}(V_{x}, GT_{x}) + d_{EMD}(V_{x}, GT_{x}) + d_{CD}(PC_{x}, PC_{x}^{gt})$$
(16)

where V_x , V_y are the virtual point clouds generated from the source partial cloud X and target partial cloud Y; GV_x and GV_y are the ground-truth missing regions of input two point clouds; PC_x and PC_y are the complete point clouds consisting of the original partial cloud and generated virtual points; PC_{gt}^x and PC_{gt}^y are the ground-truth complete point clouds. L_g^y is calculated with the same method and symmetrical parameters.

The last loss function is the registration loss. We directly measure the deviation of predicted *R* and *t* from ground-truth R_g and t_g that are recorded during the original point clouds preprocessing. Equation (17) shows the last loss term:

$$L_{reg} = \sum_{i=1}^{k} \|R_i^T R^g - \mathcal{I}\|^2 + \|t_i - t^g\|^2$$
(17)

Here, *g* denotes ground-truth. *k* represents the total iteration numbers. Therefore, the total loss can be summarized as follows:

$$L_{total} = \alpha L_d + (1 - \alpha)L_g + L_{reg}$$
(18)

3.5. Implementation Details

First, we set the training batch size to 64 and epochs to 250. Adam is the selected optimizer with a learning rate of 0.0002 and weight decay of 0.001 to perform gradient descent stably and efficiently. In order to speed up the training of the GAN network, we first train 50 epochs for the G network so that the G network can generate relatively accurate virtual points after a short training. The total number of iterations in Regnet is three. The α in Equation (18) is set to 0.05.

4. Experiments and Results

4.1. Data and Metrics

4.1.1. Dataset

We trained and evaluated VPRNet on the Modelnet40 dataset. The dataset comprises 12,311 meshed CAD models grouped into 40 artificial categories. We follow the original division of training and testing set in the original Modelnet40 dataset, that is, 9843 for training and 2468 for testing. In the test of unseen category models, we leverage the first 32 categories of shape names file in Modelnet40 for training and the last 8 categories for testing. Coincidentally, the ratio of the training to testing set is close to 8:2, which is 9907 train models and 2404 test models, respectively. We did not use the half-and-half data segmentation strategy provided by Modelnet40, because we added new processing to the original dataset, that is, the separation of the point cloud patch. We arbitrarily select a point inside a point cloud and exclude the nearest k points to construct original partial point clouds. Here, k is set to 256. Such data augmentation makes the training of baseline algorithms more difficult than under clean data. Besides, our algorithm employs the structure of GAN, and the final generation effect can be improved with more samples [60,61]. Consequently, we adjust the ratio of the training to the testing set to 8:2; 1024 points were uniformly sampled from Modelnet40 samples for training and testing of VPGnet. We employed the augmentation strategy for all sampled point clouds, and a rotation and translation was performed along each coordinate axis with a randomly selected angle within $[0, 45^{\circ}]$ and a distance generated from [-0.5, 0.5].

4.1.2. Metrics

We evaluate the network framework according to five registration metrics: *MAE*, *MSE*, *RMSE*, *R_loss*, and *T_loss*. Equations (19)–(21) shows the calculation method of the first three metrics, which evaluate the distance between two vectors. *M* is the length of two vectors, and x_i, y_i are the corresponding elements of two vectors. The smaller the value is, the better the registration effect is. We adopt the L_2 norm between the ground-truth rigid transformation parameters and the predicted results to evaluate the accuracy of the rotation and translation. The calculation methods of *R_loss* and *T_loss* are shown in Formulas (22) and (23), where R_{pre} and t_{pre} are the predicted rotation results, R_{gt} and t_{gt} are the ground truth, respectively. Finally, *Reg_loss* is defined as the sum of *R_loss* and *T_loss*. All angular measurements in our results are in units of radians.

$$MAE = \frac{1}{M} \sum_{i=1}^{M} |x_i - y_i|$$
(19)

$$MSE = \frac{1}{M} \sum_{i=1}^{M} (x_i - y_i)^2$$
(20)

$$RMSE = \sqrt{\frac{1}{M} \sum_{i=1}^{M} (x_i - y_i)^2}$$
(21)

$$R_{loss} = ||R_{pre} * R_{gt}^{-1} - I||_2$$
(22)

$$T_loss = ||t_{pre} - t_{gt}||_2$$
(23)

4.2. Baseline Algorithms

In order to evaluate the proposed network framework more comprehensively, this section divides the baseline algorithms into two categories. One is the most representative traditional algorithm, including ICP [4], Generalized ICP [36], Point-to-Plane ICP [9], and Fast Global Registration (FGR) [62], the other is the state-of-the-art deep learning-based algorithm proposed in recent years, including OMNet [12], PointnetLK [16], DCP [22], and RPMnet [41]. All networks are trained in NVIDIA Tesla v100 GPU and tested in AMD Ryzen 7 at 4800H CPU.

4.2.1. Traditional Algorithms

We choose the feature-based registration algorithm for the traditional method, namely Fast Global Registration (FGR) [62]. The algorithm uses the Fast Point Feature Histograms (FPFH) of point cloud to return corresponding point pairs with similar geometric structures. The other is ICP [4] and its variant version GO-ICP [10] and Point-to-Plane ICP [9]. As a classical point cloud registration algorithm, ICP can accurately complete the registration task under the insurance of a good initial value. The GO-ICP tries to avoid the disadvantage of the ICP algorithm falling into local optimization by employing the branch-and-bound method to search for the optimal value in the global range. The ICP-plane changes the definition of distance from point-to-point to point-to-plane. The implementation of ICP, ICP-plane, and FGR is available in Open3D. The GO-ICP is called from the library *pygoicp* whose parameters of *DT size* and *Factor* are set to 300 and 2.0. ICP and its variant ICP-plane are initialized with a rigid identity matrix, and the distance threshold is set to 1.

4.2.2. Deep Learning Algorithms

The deep learning algorithms we choose are PointnetLK [16], DCP [22], RPMnet [41], and OMNet [12]. As the first deep learning-based registration algorithm, the strategy that uses MLP to extract point cloud features for pose estimation in PointnetLK became attractive after being proposed. This algorithm is compared by many papers [12,22,40], so we chose it as the first baseline algorithm belonging to deep learning. Besides, DCP removes the relevant calculation of Lie algebra in PointnetLK and applies the Transformer to extract hybrid features. Then, the rotation matrix and translation vector are estimated by SVD for corresponding point pairs. As an advanced algorithm for applying the attention mechanism to the registration task, the DCP algorithm expresses competitive performance on the Modelnet40 dataset, so we treat it as the second baseline algorithm based on corresponding point pairs. Moreover, RPMnet utilizes a subnetwork to predict the annealing parameters according to the PPF feature. Then, a sinkhorn normalization is concatenated to the match matrix module, thus outputting a matching matrix. Finally, OMNet is proposed to specially deal with partially overlapping registration tasks with the critical mask prediction module. Although pre-trained models of the above networks are delivered by the original authors, the division of training and testing set of those models is not consistent with the design in

this paper. Therefore, we retrained other all deep learning methods with the same dataset as ours. For a fair comparison, we use the parameter values recommended in the official introduction of baseline algorithms to ensure baseline algorithms achieve the best effect. Some important parameters of all deep learning methods used in training and testing are introduced in Table 1. Note that DCP does not employ an iterative strategy. Thus, it does not contain the Iter_num parameter.

Table 1. Important training and testing parameters used in deep learning methods. \ means that there is no such parameter in the algorithm. Iter_num represents the number of iterations, Train_bsz and Val_bsz represent the batch size during training and testing, LR represents the learning rate, and Epochs represents the number of times all the training data is recycled.

Method	Iter_num	Train_bsz	Val_bsz	LR	Epochs
PointnetLK	10	64	1	0.01	200
DCP	\	32	1	0.001	250
RPMnet	5	4	1	0.0001	200
OMNet	4	64	1	0.001	1000
VPRNet(Ours)	3	24	1	0.0002	250

4.3. Generalizability Test

Table 2 shows the statistical results of the registration indicators of all algorithms under unseen category point clouds. For comparative purposes, we define the relative error rate to normalize indicators of different orders of magnitude. The calculation method is: $\varepsilon = |M_1 - M_2|/M_1$. As shown in Table 2, we can clearly obtain that the accuracy of the deep learning methods significantly exceed the traditional algorithm since the average relative error ratio of RMSE(R) and RMSE(t) is reduced by 54.21% and 32.40% over traditional algorithms. Thanks to the high-dimensional features map extraction of deep learning methods, the calculation of corresponding points is more accurate than the traditional algorithm. Specifically, our algorithm expresses good competitiveness in accuracy compared with all deep learning-based algorithms. Compared with DCP and PointnetLK, our algorithm's average relative error ratio in registration loss is reduced by 68.25% and 80.68%, respectively. We have to admit that our algorithm does have a certain gap with RPMnet and OMNet in some aspects. However, we can find that the differences are not too large to be accepted after a detailed analysis. For example, in terms of MAE(R), our value is only 5.84 larger than RPMnet, that is, the average error of rotation of three rotation axes is only 0.10°. Compared with the difference (32.57, 22.17) between RPMNet and DCP and PointnetLK, the error rate is up to 82.07% and 73.66%, respectively. Besides, focusing on translation estimation, the disparity between ours and RPMnet becomes smaller. In numerical terms, both RMSE(t) are equal to 0.16, and the MAE(t) of our method is 0.01 lower than RPMnet. Moreover, it can be seen that the robustness of the RPMnet algorithm is inferior to our algorithm in the subsequent experiments. In the comparison with OMNet, our algorithm and OMNet are both aimed at partial registration, but these adopt totally different processing ideas. Ours attempts to complete, while OMNet tries to mask the meaningless part. From the results, the difference of MAE(R) is 5.08 (0.08°), the error rates reach 84.03% and 76.27% compared with the difference between OMNet and DCP and PointnetLK, which shows that the disagreements over MAE(R) between ours and OMNet are not as sharp as DCP or PointnetLK. In summary, our algorithm achieves competitive performance in unseen category tests. The conclusion can be inferred that the self-supervised VPRNet first generates virtual corresponding points with Transformer and Self Attention, which makes up for the negative impact of the incompleteness of point clouds.

			D3 (67(4)		.	- 1		— • ()
Method	RMSE(R)	MAE(R)	RMSE(t)	MAE(t)	R_loss	T_loss	Reg_loss	Time(s)
ICP	17.29	14.85	0.19	0.16	0.73	0.21	0.94	0.005
ICP_plane	33.01	28.26	0.33	0.28	0.97	0.39	1.37	0.01
GO-ICP	48.09	43.2	0.55	0.48	1.84	0.92	2.76	0.53
FGR	28.11	24.47	0.22	0.19	0.95	0.2	1.15	0.08
PointnetLK	25.28	22.6	0.55	0.48	1.08	0.99	2.07	0.09
DCP	37.27	33	0.2	0.17	0.9	0.36	1.26	0.43
RPMnet	0.51	0.43	0.16	0.15	0.02	0.004	0.03	0.59
OMNet	2.09	1.19	0.02	0.01	0.06	0.03	0.09	0.06
VPRNet(Ours)	7.26	6.27	0.16	0.14	0.28	0.11	0.40	2.04

Table 2. Results on point clouds of unseen categories in ModelNet40. Bold numbers are the smallest in the current column, and represent the best performance. Lower is better.

The visualization of samples after registration is shown in Figure 3. The histogram on the right shows the proportions of different colors. Different colors represent the distance of the closest point. The closer to blue, the closer the closest point in the opposite point cloud is to this point. It is worth noting that the OMNet algorithm needs a ground-truth pose matrix to calculate the overlapping mask, so the calculation of registration parameters cannot be completed with only the residual clouds. Therefore, the registration of OMNet is excluded from the results in Figure 3. The case of unseen categories are shown in sub-figure (a) of Figures 3 and 4. It can be clearly seen from the figure that the color of our registration result tends to be blue. Besides, there is no obvious visual difference between the registration effect of this algorithm and the RPMnet algorithm despite the leadership in data.



Figure 3. Registration examples on (**a**) unseen categories, (**b**) noisy data, (**c**) sparse data, and (**d**) data with rotation of $30-40^{\circ}$. The histograms on the right show the distance between the points. The closer to the blue, the smaller the distance between the points.



Figure 4. Completion results on (**a**) unseen categories, (**b**) noisy data, (**c**) sparse data, and (**d**) data with rotation of 30–40°. Red points represent the original incomplete point cloud, and black points represent the points generated by the network.

4.4. Robustness Test

The following three experiments tested the resistance of the proposed algorithm and baseline algorithms to noise, sparsity levels, and initial rotation angles.

4.4.1. Noise Test

We randomly sampled jittering noise from $\mathcal{N}(0, 0.002)$ and cropped it to [-0.05, 0.05]. All the deep learning-based algorithms are retrained with noisy point clouds. The results are shown in Table 3. The registration and completion results of noisy data are summarized in Figures 3b and 4b, which shows that the proposed algorithm can still finish the completion and registration of point clouds under the influence of noise. These two figures do not show great visual deficiency. The precise analysis of noisy data is stated below.

Table 3. Results on noisy point clouds in ModelNet40. Bold numbers are the smallest in the current column, and represent the best performance. Lower is better. Our algorithm is in the front rank among all measurements.

Method	RMSE(R)	MAE(R)	MSE(t)	RMSE(t)	MAE(t)	R_loss	T_loss	Reg_loss	Time(s)
ICP	17.29	14.85	0.04	0.19	0.16	0.72	0.21	0.94	0.002
ICP_plane	33.10	28.33	1.36	0.30	0.26	0.97	0.33	1.30	0.01
GO-ICP	48.16	43.23	0.33	0.55	0.48	1.84	0.92	2.76	0.07
FGR	27.40	23.83	0.06	0.22	0.19	0.94	0.20	1.13	0.62
PointnetLK	43.87	38.91	0.32	0.54	0.47	1.68	0.93	2.61	0.1
DCP	37.67	33.40	0.06	0.20	0.17	0.92	0.36	1.28	0.32
RPMnet	5.49	4.6	0.04	0.17	0.15	0.23	0.05	0.27	0.54
OMNet	3.58	2.64	0.0005	0.02	0.01	0.13	0.03	0.15	0.06
VPRNet(Ours)	7.69	6.60	0.03	0.16	0.14	0.31	0.12	0.43	2.10

Evidently, it can be seen from Table 3 that our algorithm has significant leadership in the estimation of rotation compared with the PointnetLK and DCP, which is proved by the fact that the relative error rates of MAE(R) reach 83.04% and 80.24%. In addition,

our algorithm is still not much behind RPMnet. The performance of this algorithm on MAE(R) is only 2.00 lower than the RPM at algorithm, that is, the average rotation error of three rotation axes is only 0.03°. Compared with 28.80 and 34.31 of DCP and PointnetLK algorithm, the error rates are reduced by 93.06% and 94.17%, respectively. Additionally, the translation estimation of the VPRNet algorithm is better than the RPMnet algorithm. For example, the MAE(t), MSE(t), and RMSE(t) of VPRNet are all 0.01 lower than RPMNet. Compared with the OMNet algorithm, there are still some gaps, for example, the error rates of RMSE(R) and T_{loss} are 53.45% and 75.50%, respectively. Nevertheless, the gap between them is reduced compared with the previous clean data. In the aspect of MAE(R), the error between our algorithm and OMNet under noise interference is reduced by 22.05% compared with that under clean data, and T_{loss} is reduced by 10.00%. Therefore, it can be proved that our algorithm is closer to the advanced OMNet algorithm under noisy data than clean data. Besides, compared with clean data, our algorithm's error rate of *Reg_loss* was reduced by 12.24%, which shows that our algorithm is the only one among all deep learning methods whose registration results under noise interference are better than that of clean data. Especially in the estimation of rotation, the relative error rate of MAE(R) of RPMNet is 10.70 times higher than clean data. Not only the RPMnet algorithm but also the OMNet algorithm is worse under noise than that under clean data. For example, the change error rate of MAE(R) under clean and noisy data is 54.92%. Consequently, our method demonstrates competitive robustness among all deep learning methods under noisy data. Exploring deeper reasons, we can infer that the extracted high-dimensional embeddings contain some wrong position information, which results in biased virtual points and estimation of the registration parameters. However, the feature fusion of Transformer and Self-Attention in VPGnet and Regnet can still focus on more relevant parts in noisy data. Thus, these two modules reduce the impact of global noise on completion and registration.

4.4.2. Sparsity Level Test

Subsequently, we tested the influence of different sparsity levels on predicted rotation and translation metrics. We first performed FPS on the original two point clouds and retained four sparsity levels, 0.5, 0.25, 0.125, and 0.0625. The statistical performances of all baseline algorithms under different sparsity levels are shown in Figure 5 and 6. The registration and completion results of sparse point clouds from algorithms are shown in Figures 3c and 4c. As can be seen from the figure, only our algorithm and RPMnet algorithm can finish the registration between sparse and partial point clouds. Our algorithm can complete point clouds with a sparse level of 0.5. In order to intuitively see the impact of increased sparsity level, we calculate the *x*-axis as 0.5 - x, where *x* is the sparsity level. Although the point cloud tends to sparse, our algorithm can still alleviate the limitation of sparseness and guarantee registration quality. The detailed analysis is as follows.

We can see from Figure 5 that, no matter how the sparsity level changes, the predicted rotation and translation errors of our algorithm consistently rank high in all methods. Among the traditional algorithms, only the estimation of rotation from ICP is near our algorithm. The ICP algorithm surpasses our algorithm when the sparsity level is 0.0625, but the average error rate of the two algorithms is only 3.90%. Meanwhile, the minimum average error rate of MAE(R) between the remaining traditional algorithms and our algorithm is 54.48%. Therefore, we can conclude that our algorithm is ahead of all the traditional algorithms in the accuracy of rotation estimation. Focusing on the deep learning methods, our algorithm, RPMnet, and OMNet always maintain a leading position. Notably, compared with DCP and PointnetLK algorithms, our algorithm performs significantly better on MAE(R) since the average error rates at different sparsity levels are 68.07% and 61.75%, respectively. As a future improvement focus, our algorithm still has a gap in the overall accuracy revealed in the 49.77% of average error rate compared with RPMnet algorithms. Thankfully, the mean gap in the degree system is only 0.09°. Besides, the variance of MAE(R) metric of our algorithm under different sparsity levels is 10.39, which is close to 12.22 of RPMnet, and the average error rate is only 14.98%. Peculiarly, when

dealing with the point cloud registration problem with a sparsity level greater than 0.25, the average variation of our algorithm is 3.68, which is close to 3.70 of the RPMnet algorithm. However, our algorithm increases by 4.31 compared with MAE(R) under the previous sparse levels, which is lower than 4.66 of RPMnet when turning to the case of extreme sparsity level of 0.0625. Compared with the previous sparse level, the error growth rate of our algorithm under extreme sparse conditions is 38.90%, while the error growth rates of RPMNet and OMNet are 73.71% and 49.47%, respectively. Therefore, the above data prove that our algorithm illustrates outstanding robustness among deep learning-based methods especially under extreme sparse conditions. We can infer from the above situations that the additional virtual point completion in this paper makes the source points that do not have ground-truth correspondences produce virtual corresponding points, thus making up for the lack of shape information caused by the increased sparseness of the point cloud.



Figure 5. Influence of different sparsity levels on *MAE*(*R*).

As for the estimation of translation from Figure 6, except for the PointnetLK and GO-ICP algorithms, the estimated values of the other algorithms are relatively close, which is validated by the 0.01 variance of the mean value under different sparsity levels. Nevertheless, our algorithm is still the third performer with an average of 0.13. Although RPMnet is highly ranked with an average value of 0.08, its overall variance is 0.001 higher than ours. In other words, our translation estimation is more stable than RPMNet in the estimation of translation. Finally, the translation estimation error of our algorithm is only 0.09 higher than that of OMNet at different sparse levels. The reason for our improved performances is that the VPG module enriches the shape information of partial point clouds, so that more conjugate point pairs mean a stronger guarantee of translation estimation.



Figure 6. Influence of different sparsity levels on *T_loss*.

4.4.3. Initial Rotation Angles Test

We followed the suggestion of FMR and evenly divided the initial rotation angle range of 0–60° into 6 groups with an interval of 10°. Then, we calculated the indicators about predicted rotation in the above groups of initial angles to explore the robustness of algorithms to different initial rotation angles. The statistical figure about registration is shown in Figure 7. The broken lines with different colors in the figures represent the performance of different algorithms at various initial rotation angles. Figures 3d and 4d show the registration and completion results with an initial rotation angle of 30–40°. It can be seen from the two pictures that the completion and registration of our algorithm under this initial rotation angle are visually reasonable.

In view of the overall tendency, the prediction errors of all algorithms show a surging trend as the initial rotation angle increases. The reason is that the overlapping region between point clouds decreases with the increase of rotation angles. Besides, the FPFH feature used in the FGR algorithm is also rotation-sensitive, which debilitates the registration ability under different initial rotation angles. As can be seen from Figure 7, although our algorithm lags behind OMNet slightly, it is ahead of other algorithms in the test of all initial rotation angles with a mean value of 11.04. For a more detailed analysis, we divide the initial rotation angles into small angles of $0-40^{\circ}$ and large angles of $40-60^{\circ}$. For the small angles registration, the average value of our algorithm on MAE(R) gets a smaller 5.10, which is still ahead of other algorithms. For example, compared with RPMNet, the relative error rate is 58.27%. It indicates that our method expresses good applicability for registration with a large overlap rate. However, once facing the rotation angle of 40–60°, MAE(R) of all the deep learning methods rise in different steep degrees. The PointnetLK algorithm is the most seriously affected. The average error of the PointnetLK at the initial rotation angle of 40–60° is 193.36% higher than the average value of 0–40°. Specifically, our algorithm still takes the lead in all algorithms even in large rotation angles. The relative error rate of the MAE(R) of ours and RPMNet under 40–60° is 27.63%. Moreover, the average amplitude of RPMNet at large rotation angles is 12.43, which is greater than our 11.97. Therefore, the proposed algorithm achieves more stable and excellent performance than all methods except OMNet regarding resistance to various initial rotation angles. The apparent gap can still be observed from Figure 7 when compared with OMNet, but the average discrepancy between them is 0.06° under large rotation angles, and 0.17° under small rotation angles, respectively. Let us pay special attention to the case where the initial

rotation angle is 50–60°. In this case, the increased error rate of our algorithm is 57.14%, which is lower than 88.99% of OMNet. Therefore, our method performs with close accuracy to OMNet and better robustness to different initial rotation angles than OMNet. Notably, the stability of our method even exceeds the OMNet algorithm in extreme cases, that is, the initial rotation angle of 50–60°. By inspection, we conclude the interpretation that the VPG module enriches the corresponding point pairs after completing the missing points. The Transformer is employed to consider the position information of the opposite cloud in the structure of Regnet, so the corresponding points generated by the registration network can be keenly aware of the position change of the opposite cloud.



Figure 7. Influence of different initial angles on MAE(R).

4.5. Ablation Study

We conduct several ablation studies in this section to dissect VPRNet. Specifically, we replace the important module with an alternative to better understand how various components influence the measurement of the proposed algorithm. All experiments are performed in the same setting as the experiments in Section 4.2.

4.5.1. Without VPGnet

Firstly, we exclude VPGnet and only retain Regnet to test the effectiveness of our completion subnetwork. The network is retrained according to the contents of Section 3.5, and the number of iterations is 1. The comparison data between the retrained network and the original VPRNet is shown in Table 4. As seen from the table, VPRNet with GAN has lower rotation and translation error than the network without GAN. Mainly, R_ Loss and t_ Loss decreased by 55.56% and 66.67% than the network without the VPG module. Therefore, the network structure we designed to complete before registration plays a positive role in registering partial point clouds.

Method	RMSE(R)	MAE(R)	RMSE(t)	MAE(t)	R_loss	T_loss	Reg_loss	Time(s)	
VPRNet(No VPG)	37.27	33.00	0.20	0.17	0.90	0.36	1.26	0.43	
VPRNet(Original)	9.85	8.47	0.16	0.14	0.40	0.12	0.52	0.73	

Table 4. The results of network without VPGnet and original network on unseen category data. Bold numbers are the smallest in the current column, and represent the best performance.

4.5.2. Without Transformer

Sequentially, we exclude the Transformer module in the Regnet and explore its significance for feature fusion. The new hybrid features are extracted from the source point cloud and the target point cloud independently. There is no communication between feature information. The results are shown in Table 5.

Table 5. The results of the network without Transformer and original network on unseen category data. Bold numbers are the smallest in the current column, and represent the best performance.

Method	RMSE(R)	MAE(R)	MSE(t)	RMSE(t)	MAE(t)	R_loss	T_loss	Reg_loss	Time(s)
VPRNet(No Trans)	10.44	9.05	0.03	0.16	0.14	0.43	0.13	0.56	0.09
VPRNet(Original)	7.26	6.27	0.03	0.16	0.14	0.29	0.11	0.40	2.04

From Table 5, we can see that the Regnet without Transformer module is inferior to the original one regardless of the rotation or the translation measurement. Especially in the rotation estimation, the relative error ratio ε of our RMSE(R) and MAE(R) are reduced by 30.45%, and 30.72%, respectively. Some conclusions can be inferred from the data that the Transformer provides not only the shape information of the opposing point clouds but also includes the position information. Combining one's feature map with the others' makes the matching of corresponding points more accurate.

4.5.3. Change Iteration Numbers

Finally, we tested the influence of different iteration times in Regnet on the registration effect, and the specific results are shown in Table 6. As can be seen from the table, the number of iterations with the best performance is 3, so we take 3 as the number of iterations in the final registration network. Since too many iterations lead to the registration network relying too much on training data and reducing the generalization of test data, a moderate number of iterations guarantees generalization ability.

Table 6. Results on unseen categories point clouds in ModelNet40 under different iteration numbers. Bold numbers are the smallest in the current column, and represent the best performance.

Method	MAE(R)	RMSE(R)	MAE(t)	RMSE(t)	R_loss	T_loss	Reg_loss	Time(s)
iter = 1	8.47	9.85	0.14	0.16	0.40	0.12	0.52	0.73
iter = 3	6.27	7.26	0.14	0.16	0.28	0.11	0.40	2.04
iter = 5	6.70	7.76	0.15	0.17	0.31	0.12	0.43	3.48
iter = 7	8.77	10.18	0.15	0.17	0.40	0.14	0.54	4.79

5. Discussion

It can be concluded that VPRNet is a novel and competitive registration algorithm for partial assignment tasks from the above extensive experiments. Mainly, some meaningful discussions are summarized below.

5.1. Generalizability Test

We tested the registration under unseen category point clouds by dividing the dataset into the training and testing set according to the category. The experiment shows that the accuracy of deep learning methods significantly exceed the traditional algorithm. Specifically, our algorithm ranks high in registration accuracy among all algorithms and is close to the better RPMNet and OMNet. Therefore, we believe that our algorithm is competitive and outstanding in generalization ability.

5.2. Noise Test

We added N(0, 0.002) jittering noise to the original data to test the robustness of the baselines and our methods to the noise. The results described in Table 3 state that the accuracy of the proposed algorithm under the influence of noise is still ahead of DCP and PointnetLK algorithms. Although it is slightly inferior to OMNet and RPMnet, the gap between the two methods is smaller than clean data. Moreover, it is the only one that still produces more accurate registration under noise interference than clean data. Therefore, the proposed algorithm shows advanced robustness in point cloud registration under noise interference.

5.3. Sparsity Test

We performed FPS on the original data with different ratios to construct data with different sparsity levels. our algorithm, RPMnet, and OMNet still maintain a leading position under different sparsity levels in the sparsity test. Specifically, our algorithm illustrates the best robustness and stability among deep learning-based methods under extreme sparse conditions.

5.4. Initial Rotation Angle Test

We divided the initial rotation angles into six groups at an interval of 10° and tested the influence of different rotation angles on the registration results. Experiments show that our algorithm has better results than RPMNet at all initial rotation angles. Specifically, our registration under large initial rotation angle of 50–60° is more stable than OMNet.

5.5. Ablation Study

We excluded the influence of the VPGnet and Transformer in the ablation study to explore the role of each component in the network. Besides, we changed the number of iterations to determine the number of iterations that perform best. Experimental results show that VPGnet and transformation of the network have positive significance for the final registration, and the registration is the most accurate when the number of iterations is three.

6. Conclusions

We have proposed a novel neural network architecture called VPRNet to solve the partial-to-partial cloud registration task. The network first generates virtual points to complete the partial point clouds via a self-supervised VPGnet. Then, an iterative Regnet is designed to estimate the registration parameters. Various experimental results obtained from Modelnet40 indicate that our algorithm commands a leading position in the aspects of generality and robustness during the competition with traditional and advanced deep learning algorithms. Therefore, we can summarize that our proposed VPRNet achieves advanced performance for partial-to-partial registration. In the future, we plan to improve the algorithm from the following aspects:

- (1) We will add other loss functions and effective modules to improve the accuracy of the completion.
- (2) We will try to incorporate our method into a large system like SLAM to ensure the completeness and accuracy of reconstructed scenes.

Author Contributions: Conceptualization, S.L.; methodology, S.L., Y.Y. and L.G.; software, S.L. and Y.Y.; experiment design, S.L. and Y.Y.; experiment implementation, S.L.; investigation, S.L. and J.L.; resources, S.L. and Y.Y.; data curation, S.L., Y.Y. and L.G.; writing, S.L.; supervision, L.G.; project administration, S.L. and L.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The public dataset used in this article is ModelNet40: http://modelnet.cs. princeton.edu/ accessed on 1 January 2020.

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

Abbreviations

The following abbreviations are used in this manuscript:

ICP	Iterative Closest Point
4PCS	4 Points Congruent Sets
SA	Self-Attention
FPS	Farthest Point Sampling
SVD	Singular Value Decomposition
DCP	Deep Closest Point
ACP	Actor-Critic Closest Point
GAN	Generative Adversarial Network
FMR	Feature-metric Registration
VPG	Virtual Point Generation
VPRNet	Virtual Points Registration Network
CD	Chamfer Distance
EMD	Earth Move Distance
FGR	Fast Global Registration

References

- Wong, J.M.; Kee, V.; Le, T.; Wagner, S.; Mariottini, G.L.; Schneider, A.; Hamilton, L.; Chipalkatty, R.; Hebert, M.; Johnson, D.M.S.; et al. Segicp: Integrated deep semantic segmentation and pose estimation. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 5784–5789.
- Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. Orb-slam: A versatile and accurate monocular slam system. *IEEE Trans. Robot.* 2015, 31, 1147–1163. [CrossRef]
- Izadi, S.; Kim, D.; Hilliges, O.; Molyneaux, D.; Newcombe, R.; Kohli, P.; Shotton, J.; Hodges, S.; Freeman, D.; Davison, A. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, Santa Barbara, CA, USA, 16–19 October 2011; pp. 559–568.
- Besl, P.J.; McKay, N.D. Method for registration of 3-d shapes. Sens. Fusion Iv Control. Paradig. Data Struct. 1992, 1611, 586–606. [CrossRef]
- Prokop, M.; Shaikh, S.A.; Kim, K.-S. Low overlapping point cloud registration using line features detection. *Remote Sens.* 2019, 12, 61. [CrossRef]
- Zhang, X.; Yang, B.; Li, Y.; Zuo, C.; Wang, X.; Zhang, W. A method of partially overlapping point clouds registration based on differential evolution algorithm. *PLoS ONE* 2018, 13, e0209227. [CrossRef]
- 7. Bouaziz, S.; Tagliasacchi, A.; Pauly, M. Sparse iterative closest point. Comput. Graph. Forum 2013, 32, 113–123. [CrossRef]
- Rusinkiewicz, S.; Levoy, M. Efficient variants of the icp algorithm. In Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling, Quebec City, QC, Canada, 28 May–1 June 2001; pp. 145–152.
- Chen, Y.; Medioni, G. Object modelling by registration of multiple range images. *Image Vis. Comput.* 1992, 10, 145–155. [CrossRef]
 Yang, J.; Li, H.; Jia, Y. Go-icp: Solving 3d registration efficiently and globally optimally. In Proceedings of the 2013 IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 1457–1464.
- 11. 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.* **2015**, *38*, 2241–2254. [CrossRef]

- 12. Xu, H.; Liu, S.; Wang, G.; Liu, G.; Zeng, B. Omnet: Learning overlapping mask for partial-to-partial point cloud registration. *arXiv* **2021**, arXiv:2103.00937.
- Qi, C.R.; Su, H.; Mo, K.C.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21-26 July 2017; pp. 77–85.
- Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
- Li, Q.; Chen, S.Y.; Wang, C.; Li, X.; Wen, C.L.; Cheng, M.; Li, J. Lo-net: Deep real-time lidar odometry. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 8465–8474.
- Aoki, Y.; Goforth, H.; Srivatsan, R.A.; Lucey, S. Pointnetlk: Robust & efficient point cloud registration using pointnet. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7156–7165.
- Engel, N.; Hoermann, S.; Horn, M.; Belagiannis, V.; Dietmayer, K. Deeplocalization: Landmark-based self-localization with deep neural networks. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 926–933.
- Sarode, V.; Li, X.; Goforth, H.; Aoki, Y.; Srivatsan, R.A.; Lucey, S.; Choset, H. Pcrnet: Point cloud registration network using pointnet encoding. *arXiv* 2019, arXiv:1908.07906.
- Horn, M.; Engel, N.; Belagiannis, V.; Buchholz, M.; Dietmayer, K. Deepclr: Correspondence-less architecture for deep end-to-end point cloud registration. In Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–7.
- 20. Yew, Z.J.; Lee, G.H. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In *Computer Vision—ECCV 2018*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2018; Volume 11219, pp. 630–646. [CrossRef]
- Lu, W.X.; Wan, G.W.; Zhou, Y.; Fu, X.Y.; Yuan, P.F.; Song, S.Y. Deepvcp: An end-to-end deep neural network for point cloud registration. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 12–21.
- 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 (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 3523–3532.
- 23. Wang, Y.; Solomon, J. Prnet: Self-supervised learning for partial-to-partial registration. arXiv, 2019, arXiv:1910.12240.
- Zhang, Z.; Dai, Y.; Sun, J. Deep learning based point cloud registration: An overview. Virtual Real. Intell. Hardw. 2020, 2, 222–246. [CrossRef]
- 25. Kang, Z.; Li, J.; Zhang, L.; Zhao, Q.; Zlatanova, S. Automatic registration of terrestrial laser scanning point clouds using panoramic reflectance images. *Sensors* 2009, 9, 2621–2646. [CrossRef] [PubMed]
- Bae, K.-H.; Lichti, D.D. A method for automated registration of unorganised point clouds. *ISPRS J. Photogramm. Remote Sens.* 2008, 63, 36–54. [CrossRef]
- 27. Xu, Y.; Boerner, R.; Yao, W.; Hoegner, L.; Stilla, U. Pairwise coarse registration of point clouds in urban scenes using voxel-based 4-planes congruent sets. *ISPRS J. Photogramm. Remote Sens.* **2019**, *151*, 106–123. [CrossRef]
- Bustos, A.P.; Chin, T. Guaranteed outlier removal for point cloud registration with correspondences. *IEEE Trans. Pattern Anal. Mach. Intell.* 2017, 40, 2868–2882. [CrossRef]
- 29. Huang, J.; Kwok, T.-H.; Zhou, C. V4pcs: Volumetric 4pcs algorithm for global registration. J. Mech. Des. 2017, 139, 11. [CrossRef]
- 30. 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]
- Mellado, N.; Aiger, D.; Mitra, N.J. Super 4pcs fast global pointcloud registration via smart indexing. *Comput. Graph. Forum* 2014, 33, 205–215. [CrossRef]
- 32. Aiger, D.; Mitra, N.J.; Cohen-Or, D. 4-points congruent sets for robust pairwise surface registration. In ACM SIGGRAPH 2008 *Papers*; Association for Computing Machinery: New York, NY, USA, 2008; pp. 1–10.
- Li, S.; Lu, R.; Liu, J.; Guo, L. Super edge 4-points congruent sets-based point cloud global registration. *Remote Sens.* 2021, 13, 3210. [CrossRef]
- 34. Tao, W.; Hua, X.; Yu, K.; He, X.; Chen, X. An improved point-to-plane registration method for terrestrial laser scanning data. *IEEE Access* **2018**, *6*, 48062–48073. [CrossRef]
- 35. Al-Durgham, M.; Detchev, I.; Habib, A. Analysis of two triangle-based multi-surface registration algorithms of irregular point clouds. *Int. Arch. Photogram. Remote Sens. Spat. Inform. Sci.* 2011, 61–66. [CrossRef]
- 36. Segal, A.; Haehnel, D.; Thrun, S. Generalized-icp. Robot. Sci. Syst. 2009, 2, 435.
- Eggert, D.W.; Fitzgibbon, A.W.; Fisher, R.B. Simultaneous registration of multiple range views for use in reverse engineering of cad models. *Comput. Vis. Image Underst.* 1998, 69, 253–272. [CrossRef]
- Vlaminck, M.; Luong, H.; Philips, W. Multi-resolution icp for the efficient registration of point clouds based on octrees. In Proceedings of the 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA), Nagoya, Japan, 8–12 May 2017; pp. 334–337.

- Zeng, A.; Song, S.; Nießner, M.; Fisher, M.; Xiao, J.; Funkhouser, T. 3DMatch: Learning local geometric descriptors from rgb-d reconstructions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 June 2017; pp. 1802–1811.
- 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 (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11366–11374.
- 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 (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11824–11833.
- Gojcic, Z.; Zhou, C.; Wegner, J.D.; Wieser, A. The perfect match: 3D point cloud matching with smoothed densities. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5545–5554.
- Huang, S.; Gojcic, Z.; Usvyatsov, M.; Wieser, A.; Schindler, K. Predator: Registration of 3d point clouds with low overlap. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 4267–4276.
- 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 (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 195–205.
- Lu, W.; Zhou, Y.; Wan, G.; Hou, S.; Song, S. L3-net: Towards learning based lidar localization for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 6389–6398.
- Pais, G.D.; Ramalingam, S.; Govindu, V.M.; Nascimento, J.C.; Chellappa, R.; Miraldo, P. 3dregnet: A deep neural network for 3d point registration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 7193–7203.
- He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* 2017, 30, 5998–6008.
- Wei, H.; Qiao, Z.; Liu, Z.; Suo, C.; Yin, P.; Shen, Y.; Li, H.; Wang, H. End-to-end 3d point cloud learning for registration task using virtual correspondences. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October–24 January 2021; pp. 2678–2683.
- 50. Song, Y.; Shen, W.; Peng, K. A novel partial point cloud registration method based on graph attention network. *Vis. Comput.* **2022**, 1–12. [CrossRef]
- Arnold, E.; Mozaffari, S.; Dianati, M. Fast and robust registration of partially overlapping point clouds. *IEEE Robot. Autom. Lett.* 2021, 7, 1502–1509. [CrossRef]
- 52. Deng, H.; Birdal, T.; Ilic, S. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *Computer Vision—* ECCV 2018; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2018; pp. 602–618. [CrossRef]
- Huang, H.; Chen, H.; Li, J. Deep neural network for 3d point cloud completion with multistage loss function. In Proceedings of the 2019 Chinese Control And Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 4604–4609.
- Wen, X.; Li, T.Y.; Han, Z.Z.; Liu, Y.S. Point cloud completion by skip-attention network with hierarchical folding. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1936–1945.
- 55. Zhang, W.; Yan, Q.; Xiao, C. Detail preserved point cloud completion via separated feature aggregation. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2020; pp. 512–528.
- Wu, H.; Miao, Y.B. Lra-net: Local region attention network for 3d point cloud completion. Proc. SPIE 2021, 11605, 1160519. [CrossRef]
- Huang, Z.; Yu, Y.; Xu, J.; Ni, F.; Le, X. PF-net: Point fractal network for 3d point cloud completion. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 7662–7670.
- Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
- 59. Rubner, Y.; Tomasi, C.; Guibas, L.J. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vis.* **2000**, *40*, 99–121. [CrossRef]
- 60. Tran, N.-T.; Tran, V.-H.; Nguyen, N.-B.; Nguyen, T.-K.; Cheung, N.-M. On data augmentation for gan training. *IEEE Trans. Image Process.* **2021**, *30*, 1882–1897. [CrossRef]
- 61. Tanaka, F.H.K.d.S.; Aranha, C. Data augmentation using gans. arXiv 2019, arXiv:1904.09135.
- 62. Zhou, Q.-Y.; Park, J.; Koltun, V. Fast global registration. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 766–782.