 doi:10.20944/preprints202109.0180.v1

*Article*

# MPCR-Net: Multiple partial point clouds registration network using a global template

**Shijie Su [1,*], Chao Wang [2], Ke Chen [1], Jian Zhang [1] and Hui Yang [3]**

[1] School of Mechanical Engineering, Jiangsu University of Science and Technology, Zhenjiang 212003, China
[2] Plasma Environment Technology Group, Research Institute for Environmental Innovation (Suzhou) Tsinghua, Suzhou, 215163, China;
[3] Complex Systems Monitoring, Modeling and Control Lab, The Pennsylvania State University, University Park, PA 16802, USA
* Correspondence: sushijie@just.edu.cn; Tel.+8613511696751

**Abstract:** With the advancement of photoelectric technology and computer image processing technology, the visual measurement method based on point clouds is gradually applied to the 3D measurement of large workpieces. Point cloud registration is a key step in 3D measurement, and its registration accuracy directly affects the accuracy of 3D measurements. In this study, we designed a novel MPCR-Net for multiple partial point cloud registration networks. First, an ideal point cloud was extracted from the CAD model of the workpiece and was used as the global template. Next, a deep neural network was used to search for the corresponding point groups between each partial point cloud and the global template point cloud. Then, the rigid body transformation matrix was learned according to these correspondence point groups to realize the registration of each partial point cloud. Finally, the iterative closest point algorithm was used to optimize the registration results to obtain a final point cloud model of the workpiece. We conducted point cloud registration experiments on untrained models and actual workpieces, and by comparing them with existing point cloud registration methods, we verified that the MPCR-Net could improve the accuracy and robustness of the 3D point cloud registration.

**Keywords:** point cloud registration; template point cloud; multiple partial point cloud; deep learning

## 1. Introduction

Technical advances and market competition have pushed manufacturing companies to offer larger and more precise machine parts. This trend calls for 3D measurement systems with higher measuring efficiency and accuracy [1,2]. The visual measurement method based on point cloud data has the advantages of high measurement speed and measurement accuracy, and has no contact with the workpiece; thus, it is gradually applied to the 3D measurement of large workpieces.

3D point cloud reconstruction involves obtaining a series of partial point clouds of a workpiece under multiple poses through a 3D measuring device and then fusing these point clouds to generate a complete 3D point cloud of the workpiece [3,4]. The accuracy and reliability of 3D reconstruction directly affect the accuracy of the measurement.

A key step in point cloud 3D reconstruction is establishing a partial or global reconstruction model of the workpiece through a series of processing steps such as point cloud denoising, point cloud registration, and surface reconstruction. The purpose of point cloud registration is to move point clouds with different poses to the same posture through rigid body transformation to eliminate the misalignment between pairwise point clouds [5].

In the field of 3D reconstruction, the vast majority of point cloud registration methods belong to multiple partial point cloud registration [6]. Its basic principle is to reduce

or eliminate the cumulative error of the 3D reconstruction by minimizing the registration error between partial point clouds under multiple poses, thereby improving the reconstruction accuracy. However, traditional point cloud registration methods, such as the turntable method [7] and labeling method [8], are handcrafted methods with disadvantages such as low efficiency, high requirements for equipment accuracy, and missing point clouds in labeled areas.

With the development of deep learning technology and increased public datasets for point cloud models, many scholars have attempted to use deep learning methods to achieve point cloud registration and verify its effectiveness through experiments. However, to reduce computational complexity and improve the efficiency of 3D reconstruction, the existing deep learning-based point cloud registration methods often degenerate the multiple partial point cloud registration problem into a pairwise registration problem [9]. Then, by optimizing the relative spatial pose between the pairwise partial point clouds, the final reconstruction model of the workpiece is constructed based on the pairwise registration results.

However, due to the camera pose and environmental limitations, it is often impossible to ensure a sufficient overlap area between the pairwise point clouds. If the relationship between the partial point clouds and the overall structure of the workpiece is ignored, only the registration of partial point clouds is performed. This will increase the difficulty of point cloud registration, reduce the registration accuracy, and increase the final 3D reconstruction error.

There are remaining gaps on point cloud registration algorithms that need to be addressed:

1. Some algorithms require the structures of pairwise point clouds to be the same. If the geometric structures of the pairwise point clouds are quite different, the registration accuracy will decrease;
2. Some algorithms can complete the registration of two partially overlapping point clouds through partial-to-partial point-cloud registration methods. However, these methods rely on the individual training of specific partial data of the point cloud to establish the correspondence point relationship between the pairwise point clouds. Moreover, the registration accuracy is very sensitive to the changes points.

From the above analysis, it can be noted that existing deep learning-based point cloud registration methods can only be used for scene reconstruction and other occasions with low accuracy requirements. It is not suitable for the 3D reconstruction of large workpieces with high accuracy requirements. To solve this problem, we propose a multiple partial point cloud registration network using a global template named MPCR-Net.

MPCR-Net was inspired by PointNet [10]. It uses deep neural networks to extract and fuse the learnable features of partial point clouds and the global template point cloud. It then trains the rigid body transformation matrix for partial point clouds to register the correspondence partial point cloud to the global template point cloud and finally forms a complete point cloud of the workpiece. In MPCR-Net, the partial point cloud has the characteristics of a local geometric structure of the workpiece and is sampled by the measuring device; the global template point cloud has the complete geometric structure of the workpiece and is converted from the CAD model.

The key contributions of our work are listed as follows:

1. A multiple partial point cloud registration method based on a global template is proposed. Each partial point cloud is gradually registered to the global template in a patch-like manner, which can effectively improve the accuracy of the point cloud registration.
2. A clipping network for the global template point cloud, TPCC-Net (clipping network for template point cloud), was designed. In TPCC-Net, the features of partial point clouds and the global template point cloud are extracted and fused through a neural network, and the correspondence points of each partial point cloud are cut out from

the global template point cloud. Compared to the existing registration algorithm based on deep learning, this method can reduce the correspondence point estimation error and improve registration efficiency.

3. A parameter estimation network for rigid body transformation, TMPE-Net (parameter estimating network for transformation matrix), was designed. The learnable features of a partial point cloud and its correspondence points generated through the TPCC-NET were extracted through a neural network, and the parameters of the rigid body transformation matrix were estimated to minimize the learnable feature gap between the partial point cloud and the global template point cloud.

## 2. Related Work

### *2.1. Classic registration algorithms*

Classic registration algorithms for point clouds mainly include the iterative closest point (ICP) algorithm [11,12], variants of ICP [13-19] and geometry-based registration algorithms [20-24].

The ICP algorithm [11,12] represents a major milestone in point cloud registration and is extensively applied in various ways. The essence of ICP is to minimize the sum of the distances between correspondence points of pairwise point clouds using iterative calculations, thereby optimizing the relative pose of the pairwise point clouds. When the relative pose deviation of the pairwise point clouds is small, the algorithm is guaranteed convergence and can obtain an excellent registration result. Scholars have made various improvements to the ICP algorithm to enhance registration efficiency [16] and accuracy [17-19]. However, all ICP-style algorithms still rely on the direct calculation of the closest point correspondences; moreover, they cannot dynamically adjust according to the number of points and easily fall into local minima.

The workflow of geometry-based registration algorithms is as follows: calculate the geometric feature descriptors between pairwise point clouds [25]; determine the correspondence relationship of the pairwise point clouds according to the similarity of the descriptors, and calculate the optimal matrix for rigid body transformation between the pairwise point clouds. Random sample consistency based registration algorithms [20] are the most widely used geometry-based registration algorithms, such as the sample consensus initial alignment (SAC-IA) [24]. In SAC-IA, the correspondence points are established by calculating the local fast point feature histograms (FPFH) of the pairwise point clouds, and registration is subsequently accomplished by minimizing the distance between correspondence points. The algorithm can achieve invariance to the initial pose, and a satisfactory registration result can still be obtained when the overlap of the pairwise point clouds is low.

Unfortunately, the registration results of the geometry-based registration algorithms mainly depend on the calculation accuracy of the geometric feature descriptors between pairwise point clouds. It is necessary to manually adjust the calculation parameters involved in the geometric feature descriptor, such as the neighborhood radius of the FPFH descriptor, to minimize calculation errors and improve registration accuracy. This method consumes a significant amount of time, and it is not easy to determine the optimal parameters.

### *2.2. Deep Learning-Based registration algorithms*

Recent studies have shown that point cloud registration algorithms based on deep learning have higher registration accuracy than classic point cloud registration algorithms [26-30].

Charles et al. proposed an end-to-end deep neural network (PointNet) that could directly take point clouds as a network input [10]. PointNet overcomes the shortcomings of

general deep learning methods that cannot effectively extract features from unstructured point clouds and establishes a learnable structured representation method for unstructured point clouds. PointNet and its variants have been successfully applied in point cloud classification, object detection [31], and point cloud completion tasks [32].

In point cloud registration tasks, some deep learning algorithms use the PointNet architecture to obtain the learnable structural features of the unstructured point clouds, train rigid body transformation matrices for point cloud registration based on these features, and obtain satisfactory point cloud registration results.

PointNetLK [33] is the first deep-learning-based algorithm to use a learnable structured representation method for point cloud registration. PointNetLK modifies the classical Lucas & Kanade (LK) algorithm [34] to circumvent the inherent inability of the PointNet representation to accommodate gradient estimates through convolution. This modified LK framework is then unrolled as a recurrent neural network, in which PointNet is integrated to construct the PointNetLK architecture. However, PointNetLK and its similar algorithms, such as the DCP [35], DeepGMR [36] and PCRNet [37], work on the assumption that all points in the point clouds are inliers by default. Naturally, they perform poorly when one of the point clouds has missing points, as in the case of partial point clouds.

Unfortunately, in actual point cloud 3D reconstruction tasks, it is rare that the geometric structures of pairwise point clouds are the same, especially for large workpieces. Each raw point cloud collected by the measuring device can only map the local structure of the workpiece.

A class of point cloud registration algorithms, PRNet [38] and RPM-Net [39], which also contain the PointNet architecture, can handle partial-to-partial point cloud registration. Their application range is wider than that of PointNetLK and other algorithms that can only perform global point cloud registration. Unfortunately, these algorithms do not scale well when the number of points increases. If the number of points of the pairwise point clouds differs significantly, the estimation result of the rigid body transformation matrix will fluctuate with the number of points, resulting in a decrease in registration accuracy.

Other deep learning-based algorithms, such as the DGR [40] and multi-view registration network [41], can use neural networks to filter out some outliers from the correspondence points of the pairwise point clouds, but these algorithms require a clear correspondence between the pairwise point clouds.

## 3. MPCR-Net

### *3.1. Overview*

Most existing deep learning-based partial-to-partial point cloud registration algorithms iteratively perform the registration between partial point clouds to realize the registration between multiple partial point clouds before finally building a complete reconstruction model of the workpiece. However, when the overlap area between the pairwise partial point clouds is low, the registration accuracy cannot be guaranteed, and a large 3D reconstruction error accumulates after the iterative registration of multiple partial point clouds.

We propose MPCR-Net, a multiple partial point cloud registration network, to use a global template to improve the registration accuracy of a large workpiece. In this network, the template point cloud extracted from an ideal CAD model is used as the global registration template, and multiple partial point clouds are gradually pasted onto the global template, and the relative poses of partial point clouds and the template point clouds are subsequently optimized to obtain a fully registered point cloud of all the partial point clouds. Compared with existing registration algorithms, the MPCR-Net can guarantee the overlap rate between the pairwise point clouds (the partial point cloud can be

 doi:10.20944/preprints202109.0180.v1



approximately regarded as a subset of the template point cloud), thus reducing the registration difficulty and error.

MPCR-Net mainly comprises TPCC-Net and TMPE-Net. TPCC-Net uses a deep neural network to extract and merge the features of a partial point cloud and the global template point cloud; it then "cuts out" a correspondence partial template point cloud in the global template point cloud. TMPE-Net merges the features of partial point clouds and the correspondence partial template point clouds, then iteratively learns the optimal rigid body transformation matrix.

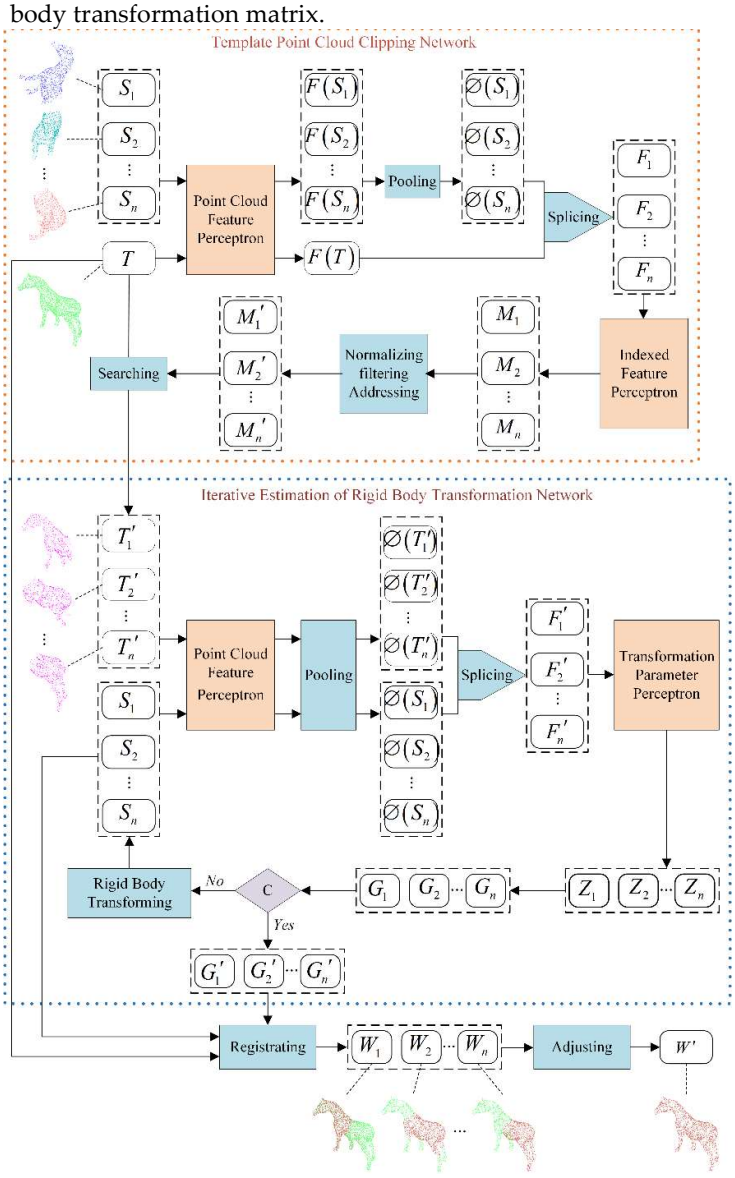



**Figure 1.** The network architecture of the MPCR-Net

As shown in Fig. 1, the overall workflow of the MPCR-Net is as follows:

1. Suppose there are n partial point clouds $S_1,...,S_n$ and a global template point cloud $T$, $S_1,...,S_n$ and $T$ are used as the inputs of TPCC-Net. In the TPCC-Net, the feature matrices $F(S_1),...,F(S_n)$ of $S_1,...,S_n$ and $F(T)$ of $T$ are obtained through the point cloud feature perceptron, respectively;

2. Global feature vectors $\varnothing(S_1),...,\varnothing(S_n)$ are obtained by pooling $F(S_1),...,F(S_n)$, then the fusion features $F_1,...,F_n$ are obtained by splicing $\varnothing(S_1),...,\varnothing(S_n)$ and $F(T)$ [42];
3. Index features $M_1,...,M_n$ of $F_1,...,F_n$ are obtained through the index feature perceptron, and the indexes $M_1',...,M_n'$ are obtained by normalizing, filtering, and addressing $M_1,...,M_n$. According to $M_1',...,M_n'$, partial template clouds $T_1',...,T_n'$ corresponding to $S_1,...,S_n$ respectively are cut out from $T$;
4. Use partial template clouds and partial point clouds as the input of the TMPE-Net in the form of correspondence point groups $\{(S_1,T_1'),(S_2,T_2'),\cdots,(S_n,T_n')\}$. In the TMPE - Net, the global feature vector $\{(\varnothing(S_1),\varnothing(T_1')),(\varnothing(S_2),\varnothing(T_2')),\cdots,(\varnothing(S_n),\varnothing(T_n'))\}$ of $\{(S_1,T_1'),(S_2,T_2'),\cdots,(S_n,T_n')\}$ is obtained through the point cloud feature perceptron and the pooling layer, and each group of global feature vectors is spliced to obtain the global fusion feature $F_1',...,F_n'$;
5. Reduce the dimension of $F_1',...,F_n'$ through the transformation parameter perceptron and output the transformation parameter vectors $Z_1,...,Z_n$, and construct the rigid body transformation matrixes $G_1,...,G_n$ according to $Z_1,...,Z_n$;
6. Perform rigid body transformations on $S_1,...,S_n$ according to transformation matrices $G_1,...,G_n$, repeat above steps to iteratively calculate $G_1,...,G_n$ until $G_1,...,G_n$ meet the stop condition C, and then combine all the iteration results to construct the optimal rigid body transformation matrix $G_1',...,G_n'$;
7. Use $G_1,...,G_n$ to register $S_1,...,S_n$ to $T$, and use the ICP algorithm to optimize the registration results to obtain point clouds $W_1,...,W_n$. Adjust $W_1,...,W_n$ to a same coordinate system, and obtain a fully registered point cloud $W'$ spliced by $W_1,...,W_n$. Subsequent 3D reconstruction tasks, such as surface reconstruction, can be performed based on $W'$.

### *3.2. TPCC-Net*

#### 3.2.1. Mathematical Formulation

Suppose that the index of a partial template point cloud $T_i'$ corresponding to the partial point cloud $S_i, i \in [1,n]$ in the global template point cloud $T$ is $M_i'$; the operator $\otimes$ is used to represent the process of estimating $T_i'$ in $T$, then

$$T_i' = M_i' \otimes T \tag{1}$$

The symbol $\varnothing : \mathbb{R}^{N\times 3} \to \mathbb{R}^{N\times K}$ represents the extraction operation of the global features of a point cloud. For example, $\varnothing(P)$ means to transform point cloud $P$ into a K-dimensional global feature vector.

Assuming that $S_i$ can be registered to $T$ completely, then $S_i$ and $T$ have similar global features of the point cloud, that is:

$$\varnothing(S_i) \approx \varnothing(RT_i' + t) \tag{2}$$

In Eq. 2, $R \in \mathbb{R}^{3\times 3}$ and $t \in \mathbb{R}^{N_y\times 3}$ are the rotation matrix and translation matrix used in the registration, respectively. The pose relationship between $S_i$ and $T$ is uncertain before registration, that is, the values of $R$ and $t$ are unknown, thus we ignore the influence of $R$ and $t$ temporarily and use Eq. 3 to establish a weaker condition to relate $S_i$ to $T_i'$.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 September 2021 doi:10.20944/preprints202109.0180.v1



$$\varnothing(S_i) \approx \varnothing(T_i') \tag{3}$$

that is:

$$\varnothing(S_i) \approx \varnothing(M_i' \otimes T) \tag{4}$$

Eq. 4 indicates that the point clouds $T$ and $S_i$ input to the TPCC-Net are linked together through the index $M_i'$, and all correspondence points that are similar to the features of $S_i$ can be determined from $T$. We assume that $M_i'$ can be calculated by operation $f(\cdot)$, then

$$M_i' = f\left(\varnothing(S_i), \varnothing(T)\right) \tag{5}$$

The search for correspondence points is to learn the index $M_i'$ by training on the TPCC-Net, and then cut out the correspondence point set $T_i'$ of $S_i$ from the global template point cloud $T$ according to $M_i'$. We consider estimating $M_i'$ by fusing the point cloud features of $T$ and $S_i$.

3.2.2 NETWORK ARCHITECTURE

TPCC-Net mainly includes a point cloud feature perceptron, pooling layer, and index feature perceptron. The architecture of TPCC-Net is shown in Fig. 2.

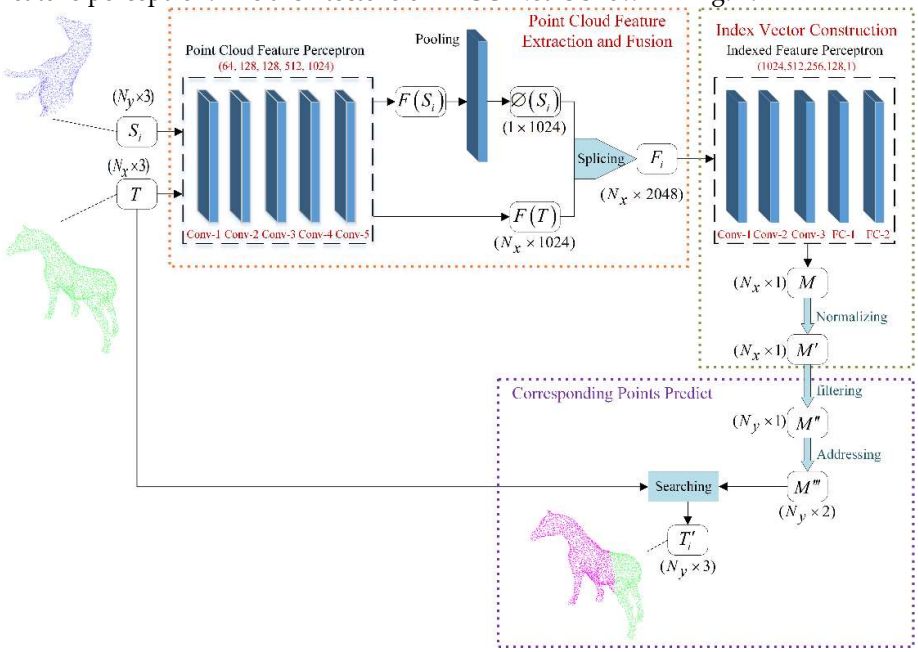



**Figure 2.** The network architecture of the TPCC-Net

The point cloud feature perceptron comprises five convolutional layers with sizes of 3-64, 64-128, 128-128, 128-512, and 512-1024. The main purpose of the point cloud feature perceptron and pooling layer is to generate the feature matrix and global feature vector of two input point clouds. The index feature perceptron consists of three convolutional layers with sizes of 2048-1024, 1024-512, 512-256, and two fully connected layers with sizes of 256-128 and 128-1. Its main purpose is to estimate the index vector for correspondence point searching through dimensional transformation.

3. 2. 3WORKING PROCESS

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 September 2021 doi:10.20944/preprints202109.0180.v1



TPCC-Net is divided into three main functional blocks. Taking the correspondence point estimation process of $S_i$ as an example, the working process of TPCC-Net is as follows.

1. Extract and fuse point cloud features

a. Suppose point clouds $S_i$ and $T$ contain $N_x$ and $N_y$ data points respectively, and $N_x < N_y$. Input $S_i$ and $T$ to the point cloud feature perceptron; it consists of five multi-layered perceptrons (MLPs), similar to PointNet. The dimensions of $S_i$ and $T$ are both increased to 1024 after the convolution processing of the point cloud feature perceptron. Afterward, generate the feature matrices $F(S_i) \in \mathbb{R}^{N_y \times 1024}$ and $F(T) \in \mathbb{R}^{N_x \times 1024}$ of $S_i$ and $T$. Weights are shared between the MLPs used for $S_i$ and $T$.

b. Use the max-pooling function to downsample $F(S_i)$ to generate a global feature vector $\varnothing(S_i)$ corresponding to $S_i$.

c. Join $\varnothing(S_i)$ and $F(T)$ to build a point cloud fusion feature $F_i \in \mathbb{R}^{N_x \times 2048}$.

2. Construct the index vector

a. Input $F_i$ to the indexed feature perceptron; the dimension of $F_i$ is reduced to one, and the index feature $M \in \mathbb{R}^{N_x \times 1}$ is then output.

b. Use the Tanh activation function to normalize $M$ to construct the index vector $M' \in \mathbb{R}^{N_x \times 1}$.

3. Predict the correspondence points

a. Encode all data points in $T$ to construct the index $M_T \in \mathbb{R}^{N_x \times 2}$.

b. Filter out the first $N_y$ elements close to zero from the index vector $M' \in \mathbb{R}^{N_x \times 1}$ to form the index element vector $M'' \in \mathbb{R}^{N_y \times 1}$.

c. Find the address of the above $N_y$ elements in $M'$ to construct the index $M''' \in \mathbb{R}^{N_y \times 2}$.

d. $M''' \in M_T$, according to each index in $M'''$; the elements corresponding to the index in $M'''$ are extracted from the global template point cloud $T$, and all the extracted elements in $T$ are combined to construct the estimated correspondence point set $T_i'$ of $S_i$ in $T$. The correspondence point estimation process is shown in Fig. 3, where the purple part is the correspondence point set $T_i'$.

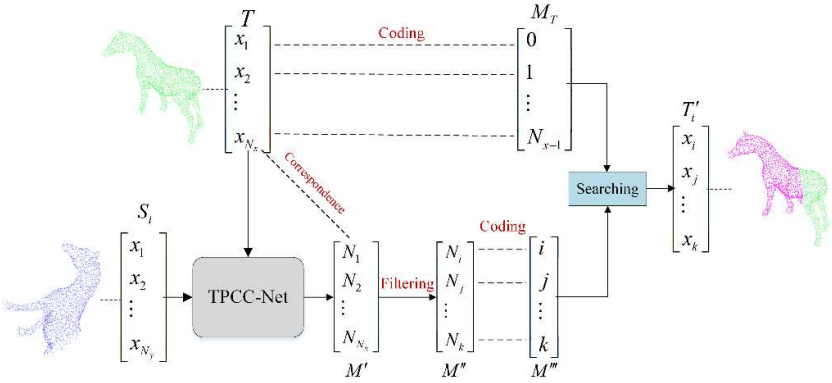



**Figure 3.** The estimating process of the correspondence points

### *3.3. TMPE-Net*

#### 3.3.1. Mathematical Formulation

Suppose that $S_i$ can be fully registered to $T$ through a rigid body transformation, then the global features of the point clouds of $S_i$ and $T_i'$ are similar, where $T_i'$ is the corresponding point set of $T$ that is generated through TPCC-Net. Assuming that the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 September 2021 doi:10.20944/preprints202109.0180.v1



rotation matrix and the translation vector used in the rigid body transformation are $R \in \mathbb{R}^{3\times3}$ and $t \in \mathbb{R}^{N_y\times3}$, respectively, then:

$$\varnothing(S_i) \approx \varnothing(RT_i' + t) \tag{6}$$

Assuming that $R$ and $t$ can be calculated by operation $\mathrm{H}(\cdot)$, then:

$$\{R, t\} = \mathrm{H}(\varnothing(S_i), \varnothing(T_i')) \tag{7}$$

Similar to PointNetLK [33], in the TMPE-Net, we directly estimate $R$ and $t$ by fusing the point cloud features of $T_i'$ and $S_i$, and further optimize $R$ and $t$ by iteration $S_i$.

### 3.3.2. NETWORK ARCHITECTURE

As shown in Fig. 4, TMPE-Net mainly includes a point cloud feature perceptron, pooling layer, and transformation parameter perceptron. The architecture of the point cloud feature perceptron is the same as that of TPCC-Net. The architecture of transformation parameter perceptron is similar to that of TPCC-Net, which consists of three convolutional layers with sizes of 2048-1024, 1024-512, 512-256, and two fully connected layers with sizes of 256-128 and 128-7. Its main function is to estimate $R$ and $t$.

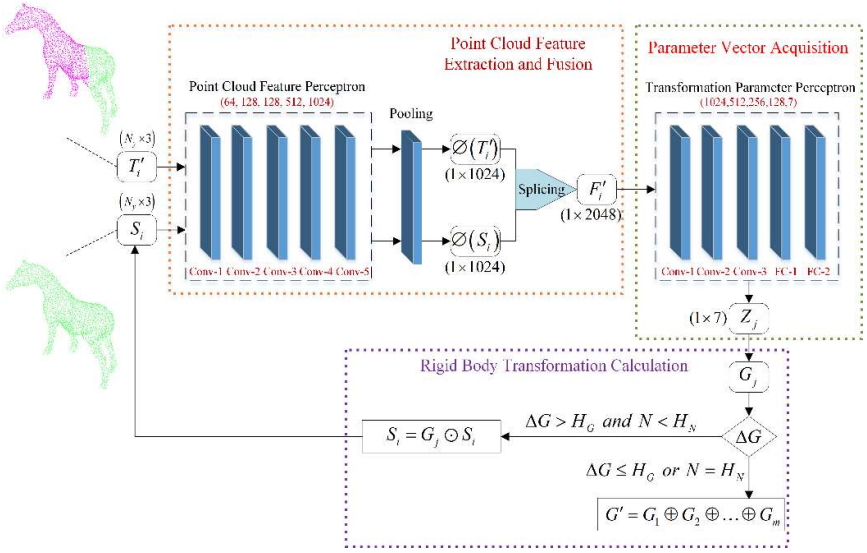



**Figure 4.** The network architecture of the TMPE-Net

### 3.3.3. WORKING PROCESS

Taking the rigid body transformation of $S_i$ as an example, TPCC-Net is working as follows:

1. Extract and fuse the global feature vector of point clouds

a. Input $S_i \in \mathbb{R}^{N_y\times3}$ and its corresponding point set $T_i' \in \mathbb{R}^{N_y\times3}$ into point cloud feature perceptron; then, the dimensions of $S_i$ and $T_i'$ are both increased to 1024, and the feature matrices $F(S_i) \in \mathbb{R}^{N_y\times1024}$ and $F(T_i') \in \mathbb{R}^{N_y\times1024}$ of $S_i$ and $T_i'$ are generated. The weights of all convolutional layers in the point cloud feature perceptron are shared for $S_i$ and $T_i'$.

b. Use the max-pooling function to downsample $F(S_i)$ and $F(T_i')$ to construct the global feature vectors $\varnothing(S_i) \in \mathbb{R}^{1\times1024}$ and $\varnothing(T_i') \in \mathbb{R}^{1\times1024}$ that correspond to $S_i$ and $T_i'$, respectively.

 doi:10.20944/preprints202109.0180.v1



c. Join $\varnothing(S_i)$ and $F(T_i')$ to build a global fusion feature $F_i' \in \mathbb{R}^{N_x \times 2048}$ of the point clouds.

2. Construct the parameter vector

Input $F_i'$ to the transformation parameter perceptron and the dimension of $F_i'$ is reduced to seven; then, output the transformation parameter vector $Z \in \mathbb{R}^{1\times 7}$.

3. Estimate the rigid body transformation

In TMPE-Net, we estimate the rigid body transformation matrix through iterative training, and the process is as follows:

Suppose that the rigid body transformation matrix has been iteratively calculated $m$ times, and parameter vector $Z_j$ can be obtained in the jth ( $j \in [1, m]$ ) iteration. Assuming that $Z_j = [a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6]$, we use the first four elements ( $a_0, ..., a_3$ ) in $Z_j$ to construct the rotation matrix $R_j$ [43]:

$$R_j = \begin{bmatrix} a_0^2 + a_1^2 - a_2^2 - a_3^2 & 2(a_1a_2 - a_0a_3) & 2(a_1a_3 + a_0a_2) \\ 2(a_1a_2 + a_0a_3) & a_0^2 + a_2^2 - a_1^2 - a_3^2 & 2(a_2a_3 - a_0a_1) \\ 2(a_1a_3 - a_0a_2) & 2(a_2a_3 + a_0a_1) & a_0^2 + a_3^2 - a_1^2 - a_1^2 \end{bmatrix} \tag{8}$$

We then use last three elements ( $a_4, ..., a_6$ ) in $Z_j$ to construct the translation vector $t_j$ :

$$t_j = [a_4 \quad a_5 \quad a_6] \tag{9}$$

The rigid body transformation matrix estimated in one iteration is $G_j = \{R_j, t_j\}$.

Suppose that the input partial point cloud of the j-th iteration is $([S_i])_j$, and $G_j$ is the corresponding estimated rigid body transformation matrix for $([S_i])_j$. Use $G_j$ to perform a rigid body transformation on $([S_i])_j$ to form a new partial point cloud $([S_i])_{j+1}$, and use $\odot$ to represent the rigid body transformation operation.

$$([S_i])_{j+1} = G_j \odot ([S_i])_j \tag{10}$$

Input $([S_i])_{j+1}$ to TMPE-Net for the (j+1)-th iteration, and output the rigid body transformation matrix $G_{j+1}$. Then, the difference $\Delta G$ between the results of two consecutive iterations was calculated.

$$\Delta G = \left\| \bar{G}_j \left( \bar{G}_{j+1} \right)^{-1} - I \right\| \tag{11}$$

In Eq. 11, $\bar{G}_j \in \mathbb{R}^{4\times 4}$ and $\bar{G}_{j+1} \in \mathbb{R}^{4\times 4}$ are the exponential mapping of $G_j$ and $G_{j+1}$ respectively, which refer to the rigid body transformation matrix of two consecutive iterations. $\|\cdot\|$ represents the sum of squares of all elements of the matrix.

Set the minimum rigid body transformation threshold $H_G$ and the maximum number of iterations $H_N$, and terminate the iterative training when $\Delta G \le H_G$ or the current iteration number $N \ge H_N$. We assume that TMPE-Net performs $m$ iteration calculations according to this rule. Combine the rigid body transformation matrices from each iteration to obtain the final trained rigid body transformation matrix $G'$.

If the total number of iterations is $m$, the final estimate $G'$ can be computed during the iterative loop:

$$G' = G_1 \otimes G_2 \otimes \cdots \otimes G_m = \{R_1 \times R_2 \times, \cdots \times R_n, t_1 + t_2 +, \cdots + t_m\} \tag{12}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 September 2021 doi:10.20944/preprints202109.0180.v1



In Eq. 12, $\otimes$ represents the combined operation of all the iterated rigid body transformation matrices. The iterative estimation process of the rigid body transformation matrix is shown in Fig. 5.

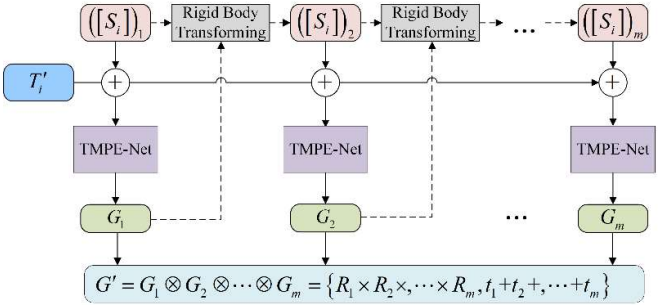



**Figure 5.** The iterative estimation of rigid body transformation

### *3.4. Loss Function*

TPCC-Net uses the loss function $Loss_1$ to maximize the registration performance (i.e., the total number of correspondence points between the partial point clouds in the global template point cloud). In TMPE-Net, the loss function $Loss_2$ is the summation of two terms $Loss_a$ and $Loss_b$. The objective of $Loss_a$ is to minimize the difference between the real rigid body transformation matrix $g$ and the estimated rigid body transformation matrix $G'$ between the partial point clouds and corresponding partial template point clouds. The target of $Loss_b$ is to minimize the difference between the global feature vectors of the partial template point clouds $T_1',...,T_n'$ and partial point clouds $S_1,...,S_n$.

The mean square error (MSE) [44] is used to express the above loss function as follows:

$$\begin{cases} Loss_1 = \left\| N_{TPCC} - N_y \right\|_F \\ Loss_2 = Loss_a + Loss_b = \left\| G' - g \right\|_F + \left\| \varnothing(T') - \varnothing(S) \right\|_F \end{cases} \tag{13}$$

where $\|\cdot\|_F$ represents the MSE between the internal elements calculated; $\varnothing(T')$ and $\varnothing(S)$ represent the global feature vectors of the point cloud $T'$ and $S$, respectively; $N_{TPCC}$ is the number of correspondence points correctly estimated by the TPCC-Net in the global template point cloud, and $N_y$ is the number of points in the partial point cloud.

### *3.5. Training*

#### 3.5.1. PERPROCESSING OF TRAINING DATA

The original data used for training is the ModelNet40[45] dataset, which contains 12468 CAD models in 40 categories. We randomly designated 20 categories of models as the training set and the other 20 categories of models as the test set. The creation process of a global template point cloud and a partial point cloud is illustrated in Fig. 6.

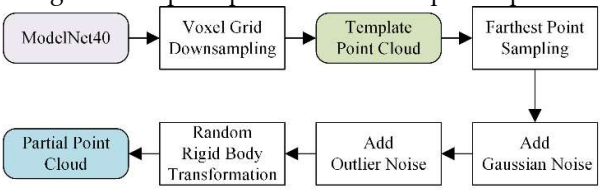



**Figure 6.** The creation process of a template point cloud and a partial point cloud

 doi:10.20944/preprints202109.0180.v1



For each original point cloud in ModelNet40, the global template point cloud is obtained through voxel grid downsampling [46-48], consisting of 1024 data points.

The process of creating a partial point cloud is as follows:

1. Each initial partial point cloud contains 568 data points, which are "cut" from a global template point cloud using the farthest point sampling (FPS) [42] algorithm.
2. Add a Gaussian noise of 0.0075 level to the initial partial point cloud to simulate the deviation of the coordinate value between the scanned data point and the data point in the global template point cloud under noisy conditions.
3. Randomly add 284 outlier noise points to the initial partial point cloud to simulate the disturbance of the scanned point cloud structure by environmental noise and sensor error. This will increase the structural difference between the initial partial point cloud and the template point cloud.
4. Create a random rigid body transformation matrix, through which the initial partial point cloud is subjected to random rotation transformation (±45° around each Cartesian coordinate axis) around the origin of the coordinate and a random translation transformation (±0.5 unit along each Cartesian coordinate axis) to obtain the partial point cloud.

3.5.2. TRAINING METHOD

TPCC-Net and TMPE-Net are trained using transfer learning [43]. First, train TPCC-Net separately to obtain the optimal network model parameter $P_a$, which means that TPCC-Net using parameter $P_a$ has the best performance on the test set. Then, use the model parameter $P_a$ as the pre-training model of TMPE-Net. The rigid body transformation matrix between the correspondence point set and the partial point cloud is iteratively trained and evaluated in the test set. Finally, the optimal network model parameter $P_b$ of TMPE-Net is obtained.

## 4. Experiments

### *4.1. Experimental environment*

The software and hardware specifications used in the experiment are shown in Table 1.

**Table 1.** Experimental environment

| Environment | Configuration | |
|---|---|---|
| Software | Operating system | Windows 10 |
| | Deep learning framework | Pytorch 1.8.1 + CUDA 11.0 + cuDNN |
| | Programming language | Python 3.8.3 |
| | Point cloud processing library | Open3D |
| Hardware | CPU | Intel(R) Core(TM) i5-9400F |
| | Memory | 16GB |
| | Graphics card | Nvidia GeForce GTX 1070 8GB |

### *4.2. Experiments based on untrained models*

We used untrained models in ModelNet40 to carry out correspondence point estimation and registration experiments, and explored the influence of correspondence points estimation accuracy, point cloud registration accuracy, and registration efficiency.

4.2.1. EVALUATION CRITERIA FOR EXPERIMENTS

1. Estimation accuracy

The estimation accuracy of correspondence points refers to the proportion of correspondence points correctly estimated by the algorithm during the experiment to the actual number of correspondence points. Taking MPCR-Net as an example, the estimation process of TPCC-Net is as follows:

Suppose the point clouds input into the network are partial point cloud A and global template point cloud B, and the point number of B is $N_B$. Encode data points in B and extract the index addresses $M_{True} \in \mathbb{R}^{N_x \times 2}$ of all correspondence points of A and B. Input A and B to TPCC-Net to estimate the index address $M_{ES} \in \mathbb{R}^{N_x \times 2}$; Count the number of elements that are the same in $M_{True}$ and $M_{ES}$, and denote it as $N_{ES}$. The actual number of correspondence points is equal to $N_B$; thus, the estimation accuracy of correspondence points of TPCC-Net is

$$P_1 = \frac{N_{ES}}{N_B} \tag{14}$$

2. Registration error

Rigid body transformation is composed of translation and rotation transformations. The registration error is subdivided into rotation and translation transformation errors. The registration error is calculated as follows:

Suppose the actual rotation angle of the partial point cloud relative to the template point cloud in the three Cartesian coordinate axis directions is $R=\{\alpha,\beta,\gamma\}$, and the actual translation distance is $t=\{d_0,d_1,d_2\}$. Input the global template point cloud and the corresponding point set of the partial point cloud in the global template point cloud into TMPE-Net, and estimate the parameter vector $Z=[a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6]$.

Elements $a_0,...,a_3$ in $Z$ represents the four parameters for the quaternion rotation matrix. The rotation Euler angles in the three directions corresponding to the parameter vector can be obtained according to the relationship between the quaternion rotation matrix and the Euler angle rotation matrix:

$$R' = \begin{Bmatrix} \alpha' \\ \beta' \\ \gamma' \end{Bmatrix} = \begin{Bmatrix} \arctan \dfrac{2(a_0a_1 + a_2a_3)}{1-2\left(a_1^{\,2} - a_2^{\,2}\right)} \\ 2\arcsin\left(a_0a_2 - a_3a_1\right) \\ \arctan \dfrac{2(a_0a_3 + a_1a_2)}{1-2\left(a_2^{\,2} - a_3^{\,2}\right)} \end{Bmatrix} \tag{15}$$

Calculate the mean absolute error $\Delta R$ between $R'$ and $R$:

$$\Delta R = \frac{|\alpha' - \alpha| + |\beta' - \beta| + |\gamma - \gamma'|}{3} \tag{16}$$

$\Delta R$ is the rotation transformation error of TMPE-Net.

Elements $a_4, a_5, a_6$ in $Z$ represents the translation distance $t' = \{a_4, a_5, a_6\}$ of the partial point cloud relative to the template point cloud in the three directions estimated by TMPE-Net, and the average absolute difference $\Delta t$ between $t'$ and $t$ is calculated as follows:

$$\Delta t = \frac{|a_4 - d_0| + |a_5 - d_1| + |a_6 - d_3|}{3} \tag{17}$$

$\Delta t$ is the translation transformation error of TMPE-Net.

3. Evaluation of work efficiency

We consider the correspondence point search between partial point clouds and the template point cloud as the preparation stage for the point cloud registration, and the total

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 September 2021 doi:10.20944/preprints202109.0180.v1



time of the correspondence point estimation and rigid body transformation matrix calculation (registration) is used to measure the efficiency of the point cloud registration.

4.2.2. CORRESPONDENCE POINT ESTIMATION

We used deep learning-based algorithms such as MPCR-Net, PRNet, and RPM-Net, which have correspondence point estimation and point cloud local registration functions to carry out the correspondence point estimation experiments.

In the experiment, each global template point cloud contained 1024 data points and was sampled from each original point cloud model in the ModelNet40 dataset through the voxel grid downsampling [44-46] algorithm. The process of creating a partial point cloud is as follows:

1. Using FPS algorithm, the initial partial point cloud is sampled from the global template point cloud according to a sampling ratio of 0.05 to 0.95, and the sampling ratio refers to the ratio of the data volume of the initial partial point cloud to the global template point cloud.
2. Rotate the initial local point cloud by 20° around the three Cartesian coordinate axes with the coordinate origin as the center, and translate 0.5 units along the three coordinate axes to obtain the local point cloud.

Taking the chair model in the ModelNet40 dataset as an example, its creation process of the global template point cloud, the initial partial point cloud, and the partial point cloud is shown in Fig. 7, where the correspondence point estimation experiment is carried out under the condition that the proportion of correspondence points is 0.5. The green part is a global template point cloud that contains 1024 data points, the orange part is the initial partial point cloud that contains 512 data points sampled from the template point cloud at a sampling ratio of 0.5, and the blue part is the partial point cloud obtained after the initial partial point cloud is rotated and translated.

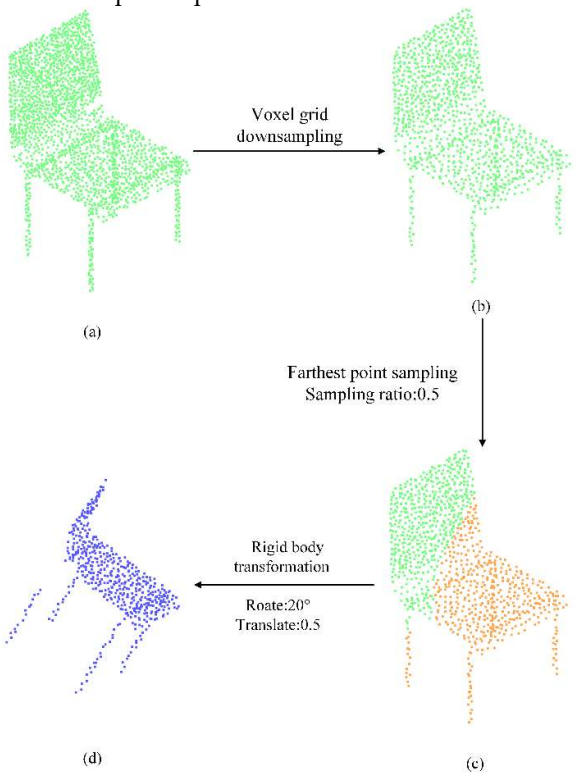
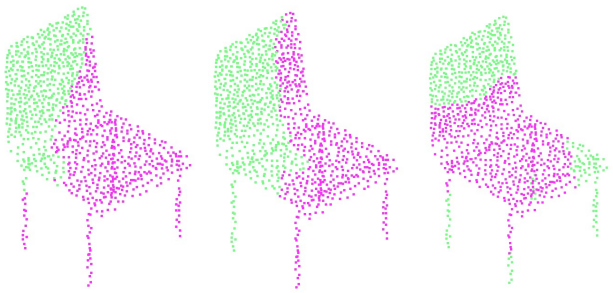



**Figure 7.** The creation process of a global template point cloud, initial partial point cloud, and partial point cloud:(a) Chair model in ModelNet40; (b) Global template point cloud; (c) Initial partial point cloud; (d) Partial point cloud.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 September 2021 doi:10.20944/preprints202109.0180.v1



Taking the chair model as an example, when the proportion of correspondence points is 0.5, the accuracy of estimation of MPCR-Net, PRNet, and RPM-Net algorithms are shown in Fig. 8, where green parts are the global template point clouds, and purple parts are the correspondence points estimated by three algorithms, respectively.



(a) (b) (c)

**Figure 8.** Accuracy of correspondence point estimation: (a) TPCC-Net, P=0.90; (b) PRNet, P=0.51; (c) RPM-Net, P=0.25

The distribution of these estimated correspondence points and the initial partial point cloud (Fig. 8(c)) in the global template point cloud are compared. The closer two distributions are, the more correspondence points are correctly estimated, and the higher the estimation accuracy P of the correspondence point estimation.

Under the condition that the proportion of correspondence points is 0.05 to 0.95, estimation experiments are performed on all models in the test set. Then, the average estimation accuracy of the correspondence point of the three algorithms is calculated.

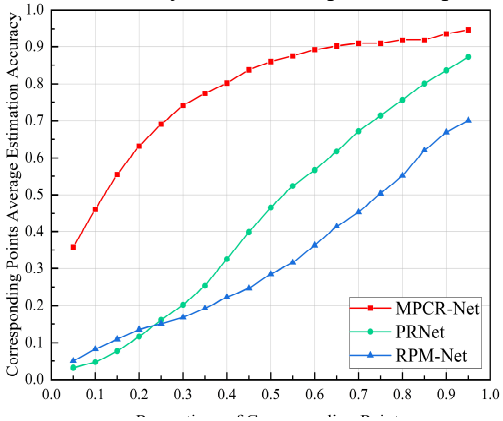



**Figure 9.** Average estimation accuracy of correspondence points with different proportions of correspondence points

As shown in Fig. 9, the average estimation accuracy of the corresponding points of the three algorithms increased as the proportion of correspondence points increased. However, as the proportion of correspondence points decreased, the accuracy gap between PRNet, RPM-Net and MPCR-Net gradually widened. When the proportion of correspondence points was 0.95, the estimation accuracies of the correspondence points of the PRNet and RPM-Net were 26.0% and 8.0% lower than that of MPCR-Net, respectively, and when the proportion of correspondence points was 0.05, the estimation accuracies of correspondence points of PRNet and RPM-Net were 86.9% and 91.2% lower than that of MPCR-Net, respectively. This indicates that compared to the PRNet and RPM-Net algorithms, the MPCR-Net algorithm proposed in this paper can effectively improve the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 September 2021 doi:10.20944/preprints202109.0180.v1

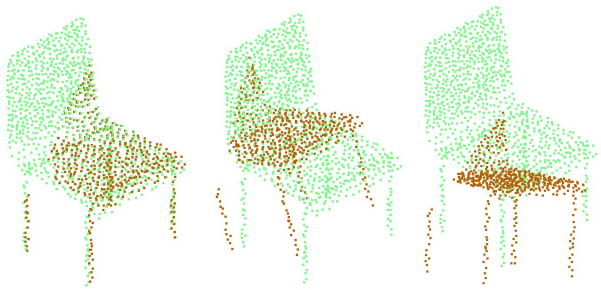


estimation accuracy of the correspondence point, especially when the proportion of correspondence points is low.

4.2.3. POINT CLOUD REGISTRATION

To explore the influence of the proportion of correspondence points on the accuracy of point cloud registration, we used the MPCR-Net, PRNet, and RPM-Net algorithms to perform point cloud registration experiments.

The point cloud registration results of three algorithms on the chair model in the ModelNet40 dataset when the proportion of correspondence points is 0.5 are shown in Fig. 10. The green parts are the template point cloud, and the red parts are the positions of the partial point cloud after registration by the three algorithms. $R$ and $t$ represent the calculation error of the rotation angle and the translation distance of each algorithm, respectively; the smaller the value of $\Delta R$ and $\Delta t$, the higher the accuracy of point cloud registration.



(a) (b) (c)

**Figure 10.** Effect of registration:(a) TMPE-Net $\Delta R = 3.7°$, $\Delta t = 0.029$; (b) PRNet $\Delta R = 17.5°$, $\Delta t = 0.269$; (c) RPM-Net $\Delta R = 21.7°$, $\Delta t = 0.328$.

Under the condition that the proportion of correspondence points was 0.05 to 0.95, the registration experiment was performed on all models in the test set, and the average registration errors of the three algorithms were calculated. Fig. 11(a) and Fig. 11(b) show the average calculation error of the rotation angle, and the average calculation error of the translation distance, respectively. The average calculation error of the rotation angle of the MPCR-Net is 23.8 - 72.8% smaller than that of the PRNet and 27.1 - 77.6% smaller than that of the RPM-Net. The average calculation error of the translation distance of the MPCR-Net is 27.0 - 90.9% smaller than that of the PRNet and 30.2 - 94.3% smaller than that of the RPM-Net. This indicates that the registration accuracy of the MPCR-Net is higher than that of the other two algorithms, especially when the proportion of correspondence points is low.

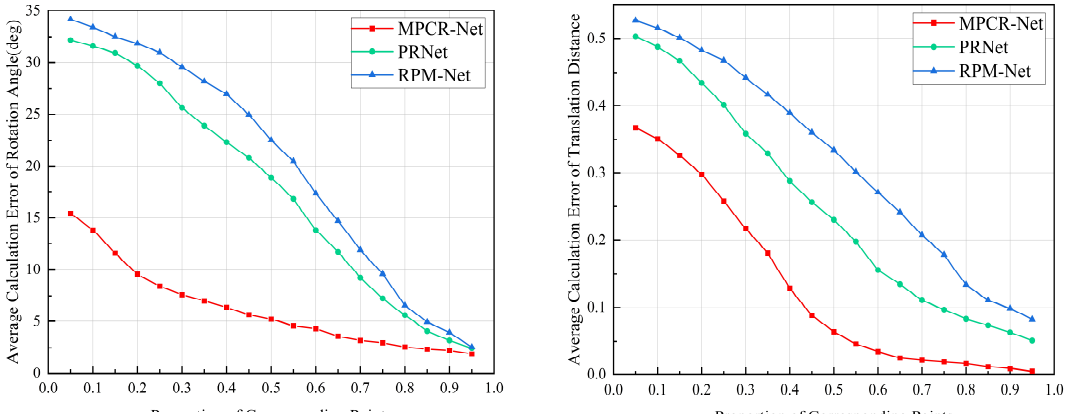
Fig. 11 shows that average calculation errors of the rotation angle and the translation distance are negatively correlated with the proportion of correspondence points, indicating that the point cloud registration accuracy decreases as the proportion of correspondence points decreases. This is because the coordinate information of the correspondence points directly participates in the calculation of the rigid body transformation matrix in the point cloud registration. Since the average estimation accuracy of the correspondence points of PRNet and RPM-Net is significantly lower than that of MPCR-Net, the average registration accuracy is also lower than that of MPCR-Net.

When the proportion of correspondence points changes from 0.95 to 0.05, the average calculation errors of the rotation angle of the MPCR-Net increase by 13.6°, whereas the calculation errors of PRNet and RPM-Net increase by 29.7° and 31.6°, respectively. Simultaneously, the average calculation error of the translation distance also changed similarly. This indicates that compared to the PRNet and RPM-Net algorithms, the MPCR-Net algorithm is more robust to changes in the number of correspondence points. When the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 September 2021 doi:10.20944/preprints202109.0180.v1



correspondence point is low, the MPCR-Net algorithm can still maintain a high registration accuracy and effectively register two point clouds with a large difference in the data amount.




(a) (b)

**Figure 11.** Average registration error with different proportions of correspondence points: (a) Average calculation error of rotation angle; (b) Average calculation error of translation distance

4.2.4. WORK EFFICIENCY

As shown in Fig. 12, under the condition that the proportion of correspondence points was 0.05 to 0.95, the average values of the total time for estimating the correspondence points and calculating the rigid body transformation matrix using the three algorithms for all models in the test set was recorded. Then, the values were used to measure the efficiency of the point cloud registration of the three algorithms.

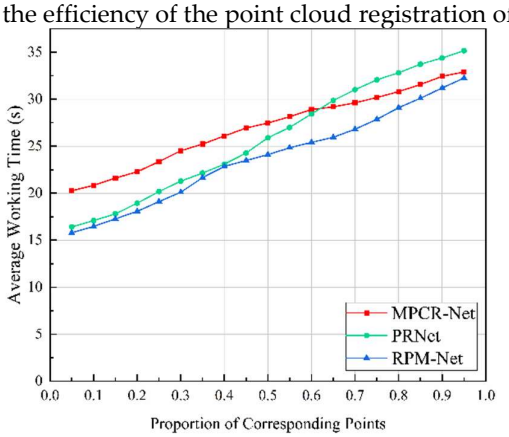



**Figure 12.** Average time of MPCR-Net, PRNet and RPM-Net under different proportions of correspondence points

Fig. 12 shows that as the proportion of correspondence points increases, the average time of each algorithm increases; this is because the number of point cloud data to be processed increases. Furthermore, as the proportion of correspondence points increases, the time-consuming growth rate of the MPCR-Net is slightly lower than that of the PRNet and RPM-Net. This is because the MPCR-Net dynamically adjusts the iterations based on the rigid body transformation matrix difference calculated by two consecutive iterations. As the proportion of correspondence points increases, the iterations required to obtain the optimal rigid body transformation matrix gradually decreases, thereby reducing the time consumption.

 doi:10.20944/preprints202109.0180.v1



*4.3. Experiments with actual workpieces*

We used the MPCR-Net to perform a point cloud registration experiment on actual workpieces and then generated the surface reconstruction models. Then, we evaluated the point cloud registration accuracy by detecting the deviations between surface reconstruction models and actual digital models. Finally, we compared with other point cloud registration algorithms, such as PR-Net and RPM-Net, to verify the effectiveness and advancement of MPCR-Net.

4.3.1. DATA SAMPLING AND PROCESSING

Actual workpieces used in the reconstruction experiment were the egg-shaped pressure hulls [47], which were tailor-welded using multiple stainless steel plates with a thickness of 2 mm. All the egg-shaped pressure hulls were numbered *a-i* sequentially, as shown in Fig. 13. The overall dimensions of the egg-shaped pressure hull 'a' are: long-axis L = 256mm and short-axis B = 180mm; the egg-shaped coefficient S = 0.69. The ideal CAD model and the global template point cloud of the shell 'a' are shown in Fig. 14.

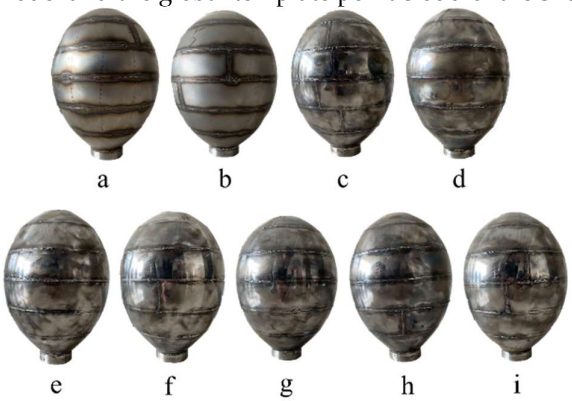



**Figure 13.** Photo of egg-shaped pressure hulls

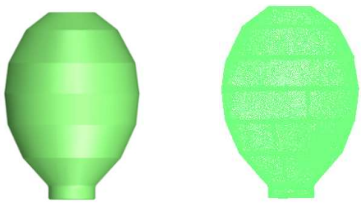



**Figure 14.** Ideal CAD model and global template point cloud of the hull 'a': (a) Ideal CAD model; (b) Global template point cloud.

As shown in Fig. 15, the process of obtaining partial point clouds of the hull 'a' is:

1. Paint the surface of the hull; place the painted hull on the rotating table and ensure that the 3D scanner is aligned with the geometric center of the hull.
2. Control the rotating table to rotate the hull to a certain angle.
3. Use the 3D scanner to scan the hull and obtain its partial point cloud under the initial angle.
4. Repeat steps b and c to obtain partial point clouds $S_2 - S_7$ of the hull at certain angles. The partial point clouds obtained are shown in Fig. 16.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 September 2021 doi:10.20944/preprints202109.0180.v1



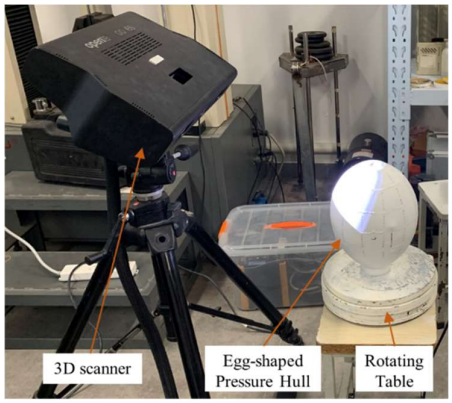



**Figure 15.** Scanning scene photo of the hull 'a' to obtain partial point clouds

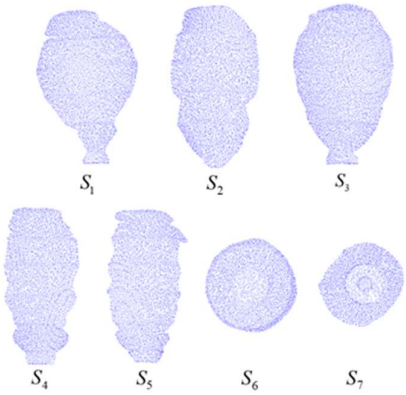
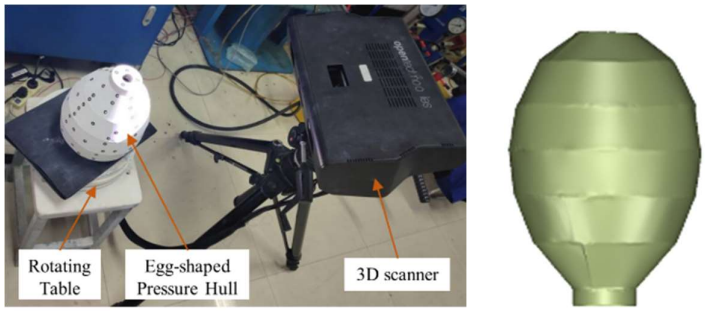



**Figure 16.** Partial point clouds of the hull 'a' at different scanning angles




(a) (b)

**Figure 17.** Scanning scene photo of the hull 'a' and generate the actual digital model: (a) Scanning scene; (b) Actual digital model.

Compared to the ideal CAD model, the actual digital model includes machining errors. The generation process of the actual digital model of the hull 'a' is shown in Fig. 17, and the details are as follows:

1. Paste labels on the surface of the painted hull 'a'.
2. Using scan steps similar to the process of obtaining partial point clouds of the egg-shaped pressure hull, obtain partial point clouds $V_1 - V_7$ of the hull at angles $A_1 - A_7$, respectively.
3. In the measurement software Optical RevEng 2.4, which is provided by the 3D scanner manufacturer, use the turntable method and the label method to register point

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 September 2021 doi:10.20944/preprints202109.0180.v1

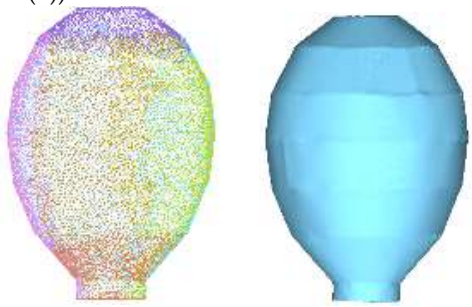


clouds $V_1 - V_7$, and use the ICP algorithm to fine-register the registration point clouds.

4. Continuously adjust the fine registration point clouds manually according to the measurement results to make the registration point clouds closer to the hull.
5. Perform surface reconstruction on the manually adjusted point clouds to obtain the actual digital model (Fig. 17(b)) of hull 'a'.

4.3.2. ANALYSIS OF REGISTRATION ACCURACY OF MULTIPLE PARTIAL POINT CLOUDS

Take the egg-shaped pressure hull 'a' as an example, use MPCR-Net to register all the partial point clouds to the global template point cloud, and use the ICP algorithm to optimize the registration results, then generate the full registered point cloud. Fig. 18 shows the full registration point cloud (Fig. 18(a)) and its surface reconstruction model (Fig. 18(b)) of hull 'a'.



(a) (b)

**Figure 18.** The final 3D reconstruction results: (a) Full registered point cloud; (b) Surface reconstruction model.

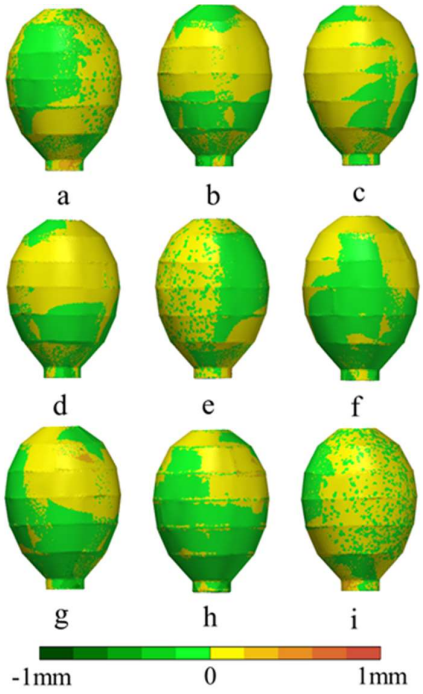



**Figure 19.** Cloud maps of contour deviation of hulls a-i calculated by MPCR-Net.

The above method is used to generate surface reconstruction models of all egg-shaped pressure hulls a-i, and the registration accuracies can be indicated by detecting the

 doi:10.20944/preprints202109.0180.v1



surface contour deviation between surface reconstruction models and actual digital models. The contour deviation cloud maps of hulls a-i are shown in Fig. 19; the smaller the deviation, the higher the registration accuracy.

The maximum positive deviation, maximum negative deviation and the area ratio exceeding the distance tolerance $\pm 0.5mm$ of the surface contour between the surface reconstruction model and the actual digital model are denoted as indicators A, B and C, respectively.

MPCR-Net, PRNet and RPM-Net algorithms were used to perform surface reconstruction experiments on all egg-shaped pressure hulls a-i, and indicators A, B, and C were used to evaluate the reconstruction accuracy of three algorithms. The results are presented in Fig. 20.

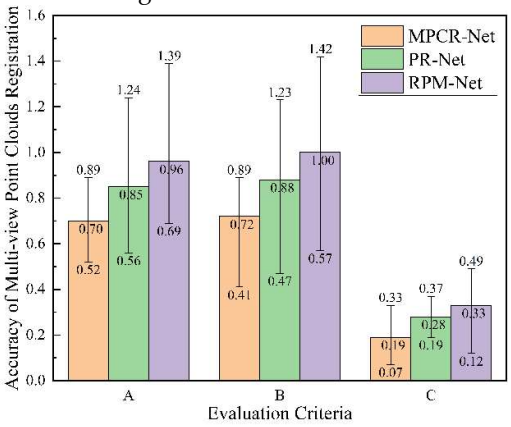



**Figure 20.** Registration accuracy of MPCR-Net, PRNet and RPM-Net on hulls a-i.

As shown in Fig. 20, the smaller the values corresponding to indicators A, B, and C, the higher the registration accuracy. Each bar represents the average registration accuracy of each algorithm under different indicators for hulls a-i, and each error bar indicates the distribution range of the indicator on hulls a-i.

For indicator A, the average maximum positive deviation of the MPCR-Net was 17.6% smaller than that of PRNet and 27.1% smaller than that of RPM-Net. For indicator B, the average maximum negative deviation of MPCR-Net is 18.2% smaller than that of PRNet and 28.0% smaller than that of RPM-Net. For indicator C, the area ratio exceeding the distance tolerance of MPCR-Net was 32.1% lower than that of PRNet and 42.4% lower than that of RPM-Net. In summary, the performance of MPCR-Net for the accuracy indicators A, B and C is better than that of PRNet and RPM-Net, indicating that MPCR-Net can provide higher registration accuracy for actual workpieces.

## 5. CONCLUSIONS

We designed a multiple partial point cloud registration network based on deep learning, called the MPCR-Net. MPCR-Net uses the global template point cloud converted from the ideal CAD model of the workpiece to guide the registration of partial point clouds. All partial point clouds are registered to the global template point cloud through TPCC-Net and TMPE-Net in MPCR-Net, forming a fully registered point cloud of a workpiece. This can effectively reduce errors in multiple partial point cloud reconstruction.

Experiment results demonstrated that MPCR-Net has the following advantages:

1. Using a global-template-based multiple partial point cloud registration method can fully guarantee the overlap rate between each partial point cloud and its corresponding partial template point cloud, thereby reducing the registration error and improving the point cloud reconstruction accuracy.
2. Searching for correspondence points between partial point clouds and the global template point cloud through TPCC-Net, does not require separate training for

specific local data of point clouds, thereby effectively reducing the correspondence point estimation error.

3. The rigid body transformation matrix parameters in the registration are estimated through TMPE-Net, and estimation results are robust to changes in data points. It eliminates the shortcomings of other algorithms that cannot effectively register two point clouds with significant differences in the amount of data.

## 6. Patents

**Author Contributions:** Conceptualization, Shijie Su and Yang Hui; Formal analysis, Chao Wang; Funding acquisition, Jian Zhang; Investigation, Chao Wang; Methodology, Shijie Su; Resources, Chao Wang and Ke Chen; Software, Chao Wang; Supervision, Jian Zhang; Validation, Shijie Su; Visualization, Ke Chen; Writing – original draft, Shijie Su; Writing – review & editing, Yang Hui. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Excellent Youth Foundation of Jiangsu Scientific Committee of China (Grant No. BK20190103) and Jiangsu Government Scholarship for Overseas Studies (Grant No. JS-2018-258).

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** The study data can be obtained by email request to the authors.

**Acknowledgments:** We would like to thank Editage (www.editage.cn) for English language editing.

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

## References

1. Zhong, K.; Li, Z.; Zhou, X.; Li, Y.; Shi, Y.; Wang, C. Enhanced phase measurement profilometry for industrial 3D inspection automation. *The International Journal of Advanced Manufacturing Technology* **2015**, *76*, 1563-1574.
2. Han, L.; Cheng, X.; Li, Z.; Zhong, K.; Shi, Y.; Jiang, H. A Robot-Driven 3D Shape Measurement System for Automatic Quality Inspection of Thermal Objects on a Forging Production Line. *Sensors (Basel)* **2018**, *18*, doi:10.3390/s18124368.
3. Liu, D.; Chen, X.; Yang, Y.-H. Frequency-Based 3D Reconstruction of Transparent and Specular Objects. In Proceedings of the Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014; pp. 660-667.
4. Yang, H.; Liu, R.; Kumara, S. Self-organizing network modelling of 3D objects. *CIRP Annals* **2020**, *69*, 409-412.
5. Cheng, X.; Li, Z.; Zhong, K.; Shi, Y. An automatic and robust point cloud registration framework based on view-invariant local feature descriptors and transformation consistency verification. *Optics and Lasers in Engineering* **2017**, *98*, 37-45.
6. Pulli, K. Multiview registration for large data sets. In Proceedings of the Second international conference on 3-d digital imaging and modeling (cat. no. pr00062), 1999; pp. 160-168.
7. Verdie, Y.; Yi, K.M.; Fua, P.; Lepetit, V. TILDE: A Temporally Invariant Learned DEtector. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
8. Ouellet, J.-N.; Hébert, P. Precise ellipse estimation without contour point extraction. *Machine Vision and Applications* **2008**, *21*, 59-67, doi:10.1007/s00138-008-0141-3.
9. Zhang, Z.; Dai, Y.; Sun, J. Deep learning based point cloud registration: an overview. *Virtual Reality & Intelligent Hardware* **2020**, *2*, 222-246, doi:10.1016/j.vrih.2020.05.002.
10. Charles, R.Q.; Su, H.; Kaichun, M.; 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; pp. 77-85.
11. Besl, P.; McKay, N.D. A method for registration of 3-D shapes, Pattern Analysis and Machine Intelligence. *IEEE Transactions on 14*, 239.
12. Chen, Y.; Medioni, G. Object modelling by registration of multiple range images. *Image and vision computing* **1992**, *10*, 145-155.
13. Rusinkiewicz, S.; Levoy, M. Efficient variants of the ICP algorithm. In Proceedings of the Proceedings third international conference on 3-D digital imaging and modeling, 2001; pp. 145-152.

14. Rusinkiewicz, S. A symmetric objective function for ICP. *ACM Transactions on Graphics* **2019**, *38*, 1-7, doi:10.1145/3306346.3323037.
15. Kamencay, P.; Sinko, M.; Hudec, R.; Benco, M.; Radil, R. Improved feature point algorithm for 3D point cloud registration. In Proceedings of the 2019 42nd International Conference on Telecommunications and Signal Processing (TSP), 2019; pp. 517-520.
16. Yang, J.; Li, H.; Campbell, D.; Jia, Y. Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2016**, *38*, 2241-2254.
17. Srivatsan Rangaprasad, A.; Xu, M.; Zevallos-Roberts, N.; Choset, H. Bingham Distribution-Based Linear Filter for Online Pose Estimation. In Proceedings of the Robotics: Science and Systems XIII, 2017.
18. Eckart, B.; Kim, K.; Kautz, J. Fast and Accurate Point Cloud Registration using Trees of Gaussian Mixtures. *arXiv e-prints* **2018**, arXiv: 1807.02587.
19. Jost, T.; Hugli, H. A multi-resolution scheme ICP algorithm for fast shape registration. In Proceedings of the Proceedings. First International Symposium on 3D Data Processing Visualization and Transmission, 2002; pp. 540-543.
20. Fischler, M.A.; Bolles, R.C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* **1981**, *24*, 381-395.
21. Chen, C.-S.; Hung, Y.-P.; Cheng, J.-B. A fast automatic method for registration of partially-overlapping range images. In Proceedings of the Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271), 1998; pp. 242-248.
22. Aiger, D.; Mitra, N.J.; Cohen-Or, D. 4-points congruent sets for robust pairwise surface registration. In *ACM SIGGRAPH 2008 papers*; 2008; pp. 1-10.
23. Mellado, N.; Aiger, D.; Mitra, N.J. Super 4pcs fast global pointcloud registration via smart indexing. In Proceedings of the Computer graphics forum, 2014; pp. 205-215.
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, 2009; pp. 3212-3217.
25. Frome, A.; Huber, D.; Kolluri, R.; Bülow, T.; Malik, J. Recognizing objects in range data using regional point descriptors. In Proceedings of the European conference on computer vision, 2004; pp. 224-237.
26. Kurobe, A.; Sekikawa, Y.; Ishikawa, K.; Saito, H. Corsnet: 3d point cloud registration by deep neural network. *IEEE Robotics and Automation Letters* **2020**, *5*, 3960-3966.
27. 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 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017; pp. 199-208.
28. 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 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020; pp. 7191-7201.
29. Lu, W.; Wan, G.; Zhou, Y.; Fu, X.; Yuan, P.; Song, S. DeepVCP: An End-to-End Deep Neural Network for Point Cloud Registration. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019; pp. 12-21.
30. 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 Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16, 2020; pp. 378-394.
31. Qi, C.R.; Liu, W.; Wu, C.; Su, H.; Guibas, L.J. Frustum PointNets for 3D Object Detection from RGB-D Data. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018; pp. 918-927.
32. Yuan, W.; Khot, T.; Held, D.; Mertz, C.; Hebert, M. PCN: Point Completion Network. In Proceedings of the 2018 International Conference on 3D Vision (3DV), 2018; pp. 728-737.

33. Aoki, Y.; Goforth, H.; Srivatsan, R.A.; Lucey, S. PointNetLK: Robust & Efficient Point Cloud Registration Using PointNet. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019; pp. 7156-7165.
34. Lucas, B.D.; Kanade, T. An iterative image registration technique with an application to stereo vision. 1981.
35. Wang, Y.; Solomon, J.M. Deep closest point: Learning representations for point cloud registration. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019; pp. 3523-3532.
36. Yuan, W.; Eckart, B.; Kim, K.; Jampani, V.; Fox, D.; Kautz, J. Deepgmr: Learning latent gaussian mixture models for registration. In Proceedings of the European Conference on Computer Vision, 2020; pp. 733-750.
37. Sarode, V.; Li, X.; Goforth, H.; Aoki, Y.; Srivatsan, R.A.; Lucey, S.; Choset, H. PCRNet: Point Cloud Registration Network using PointNet Encoding. *arXiv e-prints* **2019**, arXiv: 1908.07906.
38. Wang, Y.; Solomon, J.M. PRNet: Self-Supervised Learning for Partial-to-Partial Registration. *arXiv e-prints* **2019**, arXiv: 1910.12240.
39. Yew, Z.J.; Lee, G.H. RPM-Net: Robust Point Matching Using Learned Features. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020; pp. 11821-11830.
40. 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; pp. 2511-2520.
41. Gojcic, Z.; Zhou, C.; Wegner, J.D.; Guibas, L.J.; Birdal, T. Learning Multiview 3D Point Cloud Registration. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020; pp. 1756-1766.
42. Moenning, C.; Dodgson, N.A. *Fast marching farthest point sampling*; University of Cambridge, Computer Laboratory: 2003.
43. Olivas, E.S.; Guerrero, J.D.M.; Sober, M.M.; Benedito, J.R.M.; Lopez, A.J.S. Handbook Of Research On Machine Learning Applications and Trends: Algorithms, Methods and Techniques-2 Volumes. **2009**.
44. Orts-Escolano, S.; Morell, V.; Garcia-Rodriguez, J.; Cazorla, M. Point cloud data filtering and downsampling using growing neural gas. In Proceedings of the The 2013 International Joint Conference on Neural Networks (IJCNN).
45. Sanyuan, Z.; Fengxia, L.; Yongmei, L.; Yonghui, R. A New Method for Cloud Data Reduction Using Uniform Grids. **2013**.
46. Benhabiles, H.; Aubreton, O.; Barki, H. Fast simplification with sharp feature preserving for 3D point clouds. In Proceedings of the Programming and Systems (ISPS), 2013 11th International Symposium, 2013.
47. Zhang, J.; Wang, M.; Wang, W.; Tang, W.; Zhu, Y. Investigation on egg-shaped pressure hulls. *Marine Structures* **2017**, *52*, 50-66.