



Partial Point Cloud Registration With Deep Local Feature

Yu-Xin Zhang, Zhan-Li Sun , *Member, IEEE*, Zhigang Zeng , *Fellow, IEEE*,
and Kin-Man Lam, *Senior Member, IEEE*

Abstract—How to accurately register partial point cloud still remains a challenging task, because of its irregular and unordered structure in a non-Euclidean space, noise, outliers, and other unfavorable factors. In this article, an effective partial point cloud registration network is proposed by devising a two-stage deep local feature extraction process and an outlier filtering strategy. To be specific, on the one hand, to effectively capture geometric interdependency in the low-level space, a local attention feature extraction module is explored to extract local contextual attention features by highlighting different attention weights on neighborhoods. On the other hand, in the local feature aggregation module, two position encoding blocks are applied to increase the receptive field of each point in the high-level space. Of these, an attentive pooling can automatically learn important local features to alleviate the possible information loss. Furthermore, to derive the weight of the putative correspondence, an outlier filtering module is designed by consisting of point context normalization block, differentiable pooling layer, and differentiable unpooling layer. Moreover, in order to enhance robustness, a weighting point cloud registration model is formulated to alleviate outliers by considering the contribution of each correspondence. Experiments on multiple datasets demonstrate that the proposed approach is competitive to several state-of-the-art algorithms.

Impact Statement—Deep-learning-based point cloud registration techniques have been widely used in various tasks involving computer vision. However, in some cases, the point cloud actually detected by the sensor is incomplete and disturbed by noise. In response to this situation, we propose a new algorithm that uses a well-designed feature extraction network and an outlier filtering module to further improve the performance of partial-to-partial point cloud registration. By providing easy-to-implement algorithms, we hope to help researchers working on other transfer learning tasks, such as classification and segmentation techniques for point clouds.

Index Terms—Deep learning, deep local feature (DLF), outlier filtering, partial point cloud, registration.

Manuscript received 18 March 2022; revised 12 June 2022; accepted 21 August 2022. Date of publication 25 August 2022; date of current version 22 September 2023. This work was supported by the National Natural Science Foundation of China under Grant 61972002. This paper was recommended for publication by Associate Editor Bing Xue upon evaluation of the reviewers' comments. (*Corresponding author: Zhan-Li Sun.*)

Yu-Xin Zhang and Zhan-Li Sun are with the Information Materials and Intelligent Sensing Laboratory of Anhui Province and the School of Electrical Engineering and Automation, Anhui University, Hefei 230093, China (e-mail: yxzhang96@foxmail.com; zhlsun2006@126.com).

Zhigang Zeng is with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: zgeng@hust.edu.cn).

Kin-Man Lam is with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong (e-mail: kin.man.lam@polyu.edu.hk).

Code is publicly available at <https://github.com/zhlsunLab/DLF>
Digital Object Identifier 10.1109/TAI.2022.3201505

I. INTRODUCTION

GIVEN a model point cloud and a target point cloud, the task of rigid point cloud registration is to find the correspondence and the motion parameters that rigidly align two point clouds. As an important component, point cloud registration has been widely applied in many applications, such as medical imaging, shape representation, autonomous vehicle [1], etc.

So far, many effective works have been recorded for point cloud registration. As one pioneer approach, the iterative closest point (ICP) algorithm was presented in [2] to handle rigid registration by alternatively searching for a matching and an alignment. To alleviate local optima and other difficulties of the ICP, many improvements have been gradually developed [3], [4], [5]. However, these methods are slower compared to the ICP.

In recent years, deep learning has taken a dominant role in the field of computer vision. Many computer vision tasks have been shown, which can be better solved by neural-network-based methods. Recently, several learning-based neural network methods have been proposed for point cloud registration [6], [7], [8], [9], [10], [11] and are often much faster than traditional global registration methods. Even though the existing deep learning methods have been developed to solve some partial point cloud registration problems, there are still some challenges due to disordered structures, noise, outliers, and other issues.

In this study, an effective partial point cloud registration network is designed by utilizing a two-stage deep local feature (DLF) extraction process and an outlier filtering module (OFM). For the partial-to-partial registration problem, although the existing methods can capture the local geometric structure to a certain extent, the geometric interdependency is still not fully utilized for each individual point and its neighborhoods. Therefore, by emphasizing different attention weights on neighborhoods, a local attention feature extraction (LAFE) module is initially explored in the low-level space to learn local contextual attention features of point clouds. For extracted attention features, to retain the intricate local structure, we utilize a local feature aggregation (LFA) module by using two-layer position encoding (PE) blocks to expand the receptive field in the high-level space. Among the PE block, an attentive pooling (AP) is adopted instead of the max or mean pooling by considering its ability to automatically learn important local features and then alleviate the possible information loss in feature fusion. Meanwhile, inspired by Gogic et al. [12], an OFM, consisting of point context normalization (PCN) block, differentiable pooling (Diff-pooling) layer,

and differentiable unpooling (Diff-UnPooling) layer, is used to derive the weight of the putative correspondence. In addition, to enhance the robustness, a weighting point cloud registration model is formulated by considering the contribution of each correspondence.

In general, the contributions of the proposed network can be summarized as follows.

- 1) In order to capture geometric interdependency in the low-level space, a LAFE module is explored to extract local contextual attention features by considering self-attention and neighboring-attention mechanisms.
- 2) An LFA module, consisting of PE block and AP, is utilized to obtain a more informative feature in the high-level space by aggregating the extracted local attention features.
- 3) In order to enhance robustness, an OFM is used to derive the weight of the putative correspondence for a weighting point cloud registration model.

II. RELATED WORK

A. Learning on Point Clouds

As an intuitive approach, voxelization converts irregular and sparse point cloud into a standard grid structure. In [13], the volumetric occupancy grid was applied to significantly estimate unknown, occupied, and free space from limited range measurements. A deep network, called VoxelNet, was constructed to convert a set of points in equal spatial voxels into a unified feature representation through the voxel feature coding layer [14]. In [15] and [16], the sparse nature of point cloud was exploited to learn deep 3-D representation at high resolution. For feature learning, the 3-D shape was converted into regular 3-D voxel grids, such as SplineCNN [17], KPConv [18], and GWCNN [19]. Instead of transforming point clouds to other data representations that generate unnecessarily large amounts of data, PointNet [20] and PointNet++ [21] are designed to process the original point clouds directly. Wang et al. [22], Chen et al. [23], and Hu et al. [24] treat the point cloud as a graph of interconnected neighboring points and apply a graph neural network to extract features.

B. Outlier Rejection

Formulated as a classification problem, a deep network is developed to decide which correspondences to retain and which to reject, by exploiting epipolar constraints to learn to reject outliers [25]. In [26], an order-aware network was proposed to infer the probabilities of correspondences being inliers and regress the relative pose encoded by the essential matrix. In [12], multiview 3-D point cloud registration was formulated as an iterative reweighted least squares problem. In order to alleviate outliers, a nonlinear weighting strategy was provided by extending a 3-D outlier filter based on the order-aware blocks [26].

C. Traditional Registration Algorithm

The most widely used traditional registration method is the ICP, whose main steps are correspondence search and transformation estimation. Most other methods are improvements

to the ICP algorithm, such as [3] and [4] using a nonlinear optimization strategy, [5] using a better feature representation, or [27] reducing the computational complexity.

D. Deep Learning Registration Algorithm

In [7], a modified Lucas and Kanade algorithm was unrolled as a recurrent neural network and integrated within the PointNet framework for point cloud registration. Inspired by the PointNetLK algorithm, CorsNet was proposed in [28], which connects local with global features and regresses the correspondence between point clouds. A point cloud registration network, deep closest point (DCP), was proposed in [6] by embedding point cloud into a high-dimensional space via dynamic graph convolutional neural network. In [29], the convolutional layers were used to estimate the corresponding confidence levels, and the transformation parameters were calculated using weighted procrustes. Compared to the above methods, instead of point-to-point registration, a partial registration network (PRNet) [8] was designed for partial-to-partial registration by jointly learning an appropriate geometric representation, a keypoint detector, and keypoint-to-keypoint correspondence. In [30], a hybrid feature was used for guided correspondence search, which consists of both the shape and geometric features. Furthermore, a two-stage point elimination technique was utilized to improve computational efficiency and eliminate point and point pairs that are not conducive to matching. In RPM-Net [31], the soft assignment matrix is learned using the differentiable Sinkhorn [32] layer and the annealing layer, which can alleviate the problem of large differences in spatial coordinates and local geometric dimensions to some extent. In [33], a global feature-based network, OMNet, was more robust to noise due to the rejection of nonoverlapping regions by learning masks. In FINet [34], data interrelation was enhanced by introducing multilevel feature interaction modules and embedding them into a two-branch structure.

III. NETWORK ARCHITECTURE

A. Problem Statement

Considering a pair of potentially overlapping point clouds $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]^T$ with N points and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M]^T$ with M points, each point $\mathbf{x}_i \in R^d$ or $\mathbf{y}_j \in R^d$ can be represented as a d -dimensional feature vector, which may contain 3-D space coordinates, surface normal, color, or intensity. The task of point cloud registration is to estimate a rotation matrix \mathbf{R} and a translation vector \mathbf{t} that rigidly align \mathbf{X} to \mathbf{Y} by minimizing the registration error

$$(\mathbf{R}, \mathbf{t}) = \arg \min_{\mathbf{R}, \mathbf{t}} \sum_i \|\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_{m(\mathbf{x}_i)}\|^2 \quad (1)$$

where $m(\cdot)$ represents the step of obtaining matching pairs of points, where the element $m(\mathbf{x}_i, \mathbf{Y})$ can be seen as a soft pointer for each \mathbf{x}_i to the elements in point cloud \mathbf{Y} [6], [8]. $\mathbf{y}_{m(\mathbf{x}_i)}$ represents a matching point in point cloud \mathbf{Y} to \mathbf{x}_i obtained by means of the operation $m(\cdot)$.

Fig. 1 shows the proposed registration network for point cloud registration. There are four main components: LAFE, LFA,

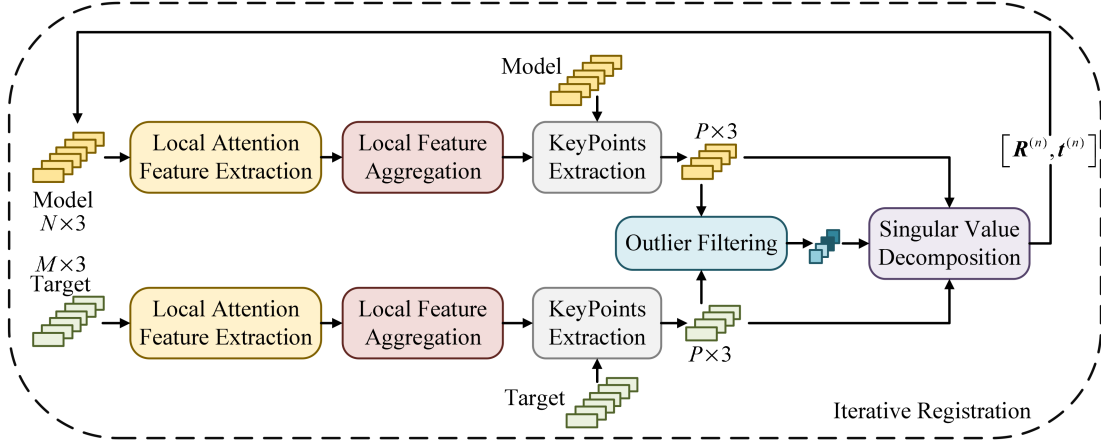


Fig. 1. Architecture of the proposed network for point cloud registration. First, the model point cloud and the target point cloud enter the LAFE and LFA modules for feature extraction. After that, the keypoint extraction is used to obtain subsets with a high degree of matching. Then, the OFM is used to calculate the confidence of the pair of points in the two subsets, which allows us to estimate the rigid transformation by solving a weighted least squares problem through singular value decomposition.

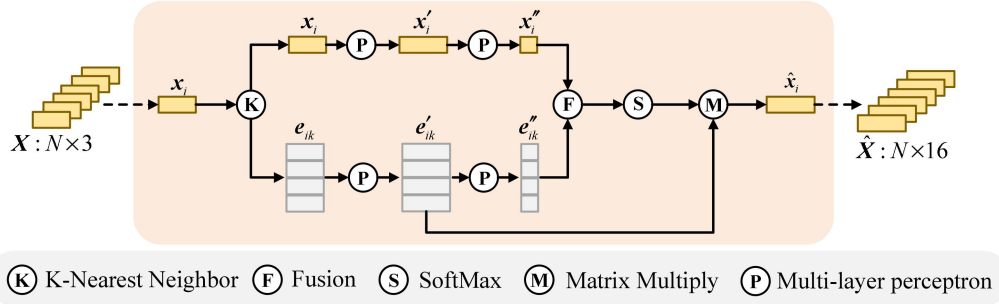


Fig. 2. LAFE. For each point, the attention coefficient is obtained by fusing attention features (self-attention features and attention features of neighboring points) and performing SoftMax operation. The local attention feature is then obtained through linear combination.

outlier filtering, and motion parameter computation. A detailed statement of these four components is made in the following sections.

B. Local Attention Feature Extraction

In the first stage of feature extraction, the LAFE module shown in Fig. 2 is designed, which enables each point to perceive the geometric information of the surrounding points and provides reasonable geometric clues for subsequent feature processing. For each point $x_i (i = 1, \dots, N)$, a set of K neighbors $x_{ik} (k = 1, \dots, K)$ can be searched via comparing the pointwise Euclidean distances. The edge feature e_{ik} between x_i and its k th neighbor x_{ik} can be defined as

$$e_{ik} = x_i - x_{ik}. \quad (2)$$

For x_i and e_{ik} , the higher level features x'_i and e'_{ik} can be obtained via a multilayer perceptron $h_\theta(\cdot)$

$$x'_i = h_\theta(x_i) \quad (3)$$

$$e'_{ik} = h_\theta(e_{ik}) \quad (4)$$

where θ denotes the filter parameter set. Furthermore, the self-attention coefficient x''_i and the neighbor-attention coefficient e''_{ik} are given as

$$x''_i = h_\theta(x'_i) \quad (5)$$

$$e''_{ik} = h_\theta(e'_{ik}). \quad (6)$$

By fusing x''_i and e''_{ik} , the attention coefficient is defined as

$$c_{ik} = g(x''_i + e''_{ik}) \quad (7)$$

where $g(\cdot)$ is a nonlinear activation function, leaky rectified linear unit (ReLU). Then, the SoftMax function is used, which normalizes the attention coefficient to each point

$$\alpha_{ik} = \frac{\exp(c_{ik})}{\sum_{j=1}^K \exp(c_{ij})}. \quad (8)$$

Finally, for each point, the local contextual attention feature is computed by weighting e'_{ik} with α_{ik} , i.e.,

$$\hat{x}_i = g\left(\sum_{k=1}^K \alpha_{ik} e'_{ik}\right). \quad (9)$$

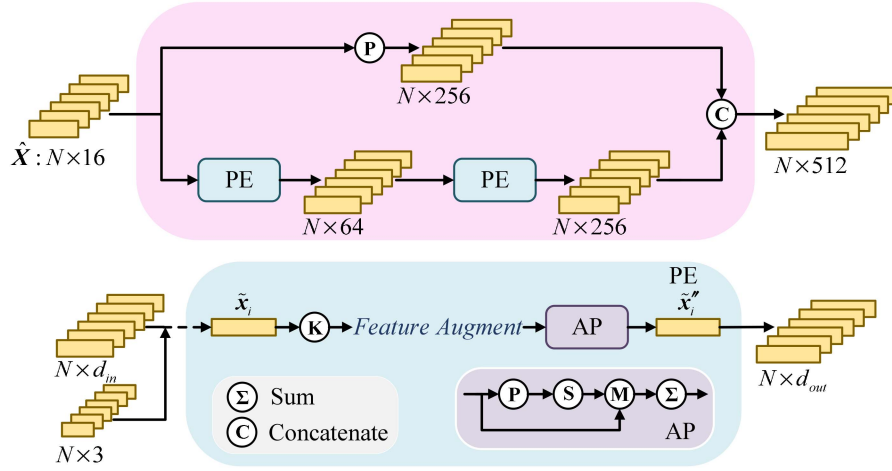


Fig. 3. LFA. The lower part of the figure is the PE block, which is mainly composed of feature augment and AP units. The upper part of the figure shows that the local feature extraction module is composed of two PE blocks and a skip connection.

C. Local Feature Aggregation

The second stage of the designed feature extraction is the LFA module shown in Fig. 3, which learns more fine-grained context features by embedding the attention features obtained in the first stage. Here, the complex local structure of the point cloud is learned by stacking PE blocks. Taking the first layer of PE as an example, the output of LAFe, \hat{x}_i , together with the node feature x_i , is concatenated to the node representing \tilde{x}_i to perceive its relative position

$$\tilde{x}_i = (x_i \oplus \hat{x}_i) \quad (10)$$

where \oplus is a concatenation operation. For \tilde{x}_i , a set of K neighbors $\tilde{x}_{ik} (k = 1, \dots, K)$ are searched to construct a K -nearest neighbor graph. Referred to [24], the node feature \tilde{x}_i , neighbor feature \tilde{x}_{ik} , and edge feature \tilde{e}_{ik} in the graph are concatenated as a local feature representation \tilde{x}'_{ik} to achieve feature augmentation

$$\tilde{x}'_{ik} = h_{\theta}(\tilde{x}_i \oplus \tilde{x}_{ik} \oplus \tilde{e}_{ik}). \quad (11)$$

In most of the existing works [8], [22], max or mean pooling are typically adopted to integrate the neighboring features. Nevertheless, much information may be lost in such a hard integration. Therefore, an AP is applied on \tilde{x}'_{ik} to generate powerful local features. For \tilde{x}'_{ik} , a unique attentive score is computed via a shared function $g_{\theta}(\cdot)$

$$s_{ik} = g_{\theta}(\tilde{x}'_{ik}) \quad (12)$$

where $g_{\theta}(\cdot)$ consists of a 1×1 convolutional layer followed by a SoftMax layer. Given \tilde{x}'_{ik} and s_{ik} , an informative feature is obtained by aggregating the geometric patterns and features of its K nearest points

$$\tilde{x}''_i = \sum_{k=1}^K \tilde{x}'_{ik} \cdot s_{ik}. \quad (13)$$

In order to achieve a balance of effectiveness and efficiency, the process is repeated twice for PE units with a skip connection.

D. Outlier Filtering

In partial point cloud registration, the number of model point clouds and target point clouds is typically not equal ($N \neq M$), and a subset of X and Y matching each other needs to be found. Referring to [8], thanks to the features acquired by the LAFe module, the LFA module, and the transformation module [35], the keypoint extraction module is introduced to discard the points that are not conducive to matching and obtain subset $X^P = [x_1, x_2, \dots, x_P]^T$ and $Y^P = [y_1, y_2, \dots, y_P]^T$ with high degree of matching. Although a subset with a higher matching degree is obtained, there are still unmatched point pairs, which will increase the ambiguity of the registration. In order to alleviate this problem, we adopt the OFM, as shown in Fig. 4(a), to evaluate the reliability of the matching results.

For x_i and y_i , after a concatenation operation

$$z_i = (x_i \oplus y_i) \quad (14)$$

the coordinates of the putative correspondences $Z = \{z_i \in R^{P \times 6}, i = 1, 2, \dots, P\}$ are used as the input of the OFM. As shown in Fig. 4(a), after a multilayer perceptron with an output dimension of 16, Z is fed into a ResNet block

$$Z_l = \text{Res}(h_{\theta}(Z)) \quad (15)$$

which is superimposed by two PCN blocks. The PCN block is composed of one instance normalization layer with a multilayer perceptron. Subsequently, a Diff-pooling layer, as shown in Fig. 4(b), maps the P putative correspondences Z_l to Z_{l+1} with M clusters

$$S_{\text{pool}} = \text{SoftMax}(h_{\text{pool}}(Z_l)) \quad (16)$$

$$Z_{l+1} = S_{\text{pool}}^T Z_l \quad (17)$$

where S_{pool} denotes an assignment matrix and $h_{\text{pool}} \in R^{P \times M}$ has the same structure as PCN. As shown in Fig. 4(c), a spatial correlation layer ϕ is applied on Z_{l+1} to obtain global context information

$$Z'_{l+1} = \phi(Z_{l+1}). \quad (18)$$

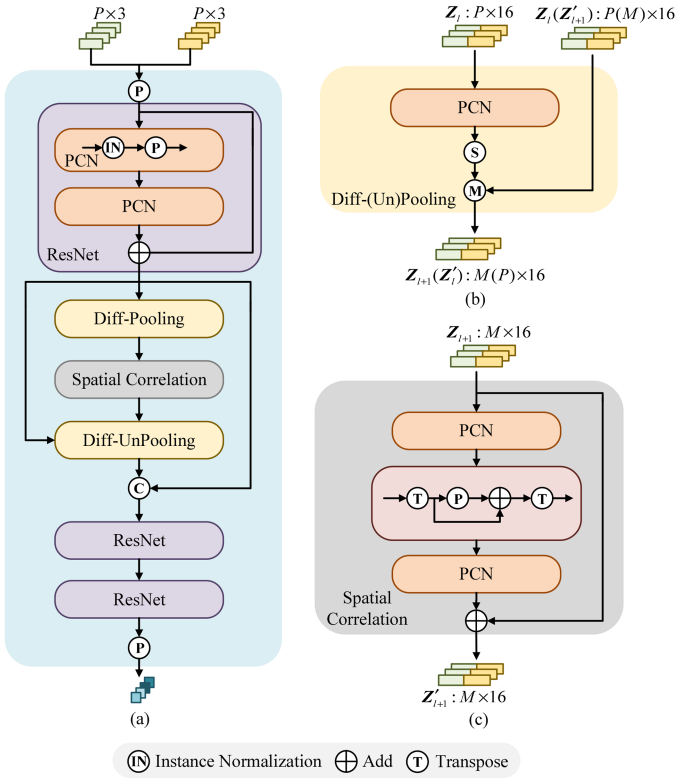


Fig. 4. OFM. (a) Overall module structure. (b) Differentiable (un)pooling (diff-(un)pooling) layer. (c) Spatial correlation layer. Of these, PCN represents point context normalization.

As a reversed operation to the Diff-Pooling layer, a Diff-UnPooling layer is then adopted to upsample the coarse representation and build a hierarchical architecture. An unpooling assignment matrix, S_{unpool} , is computed via a permutation equivariant function $h_{\text{unpool}}(\cdot)$ followed by a SoftMax layer

$$S_{\text{unpool}} = \text{SoftMax}(h_{\text{unpool}}(Z_l)). \quad (19)$$

A new feature Z'_l is obtained by a map

$$Z'_l = S_{\text{unpool}} Z'_{l+1}. \quad (20)$$

Given Z_l and Z'_l , the weight w can be given as

$$w = \text{Tanh}(\text{ReLU}(h_\theta((\text{Res}^2(Z_l \oplus Z'_l)))) \quad (21)$$

where Res^2 represents a ResNet with two layers, and the output dimension of $h_\theta(\cdot)$ is 1.

E. Motion Parameter Computation

In order to enhance robustness, a weighting matrix is applied in (1) to estimate R and t

$$(R, t) = \sum_i w_i \|R x_i + t - y_{m(x_i)}\|^2 \quad (22)$$

where $w_i \in [0, 1)$ is the putative corresponding weight computed by the OFM; the higher the credibility of point-pair matching, the higher the corresponding weight value, and vice versa. Here, a Gumbel-SoftMax sampler is adopted to sample

the matching matrix to get $y_{m(x_i)}$ [36]. Given the weight w_i , the weighted centers of x_i and $y_{m(x_i)}$ are computed as

$$\bar{x} = \frac{\sum_i w_i x_i}{\sum_i w_i}, \quad \bar{y}_{m(x)} = \frac{\sum_i w_i y_{m(x_i)}}{\sum_i w_i}. \quad (23)$$

The weighted covariance matrix can be given as

$$S = \sum_i (x_i - \bar{x})^\top W (y_{m(x_i)} - \bar{y}_{m(x)}) \quad (24)$$

where $W = \text{diag}(w_i, i \in 1, 2, \dots, P)$. After a singular value decomposition

$$S = U \Sigma V^\top \quad (25)$$

the motion parameters R and t can be given as

$$R = V U^\top \quad (26)$$

$$t = -R \bar{x} + \bar{y}. \quad (27)$$

As shown in Fig. 1, the net parameters and the motion parameters are alternatively optimized to obtain the optimal values.

F. Loss Function

In the n th optimization process, the loss function consists of three terms: a cycle consistency loss L_c^n , a rigid motion loss L_m^n , and a global feature alignment loss L_g^n [8]. The cycle consistency loss L_c^n is defined as

$$L_c^n = \|R_{xy}^n R_{yx}^n - I\|^2 + \|t_{xy}^n - t_{yx}^n\|^2 \quad (28)$$

where $[R_{yx}^n, t_{yx}^n]$ is the rigid motion parameters from Y to X . The rigid motion loss L_m^n is defined as

$$L_m^n = \|R_{xy}^{n*} R_{xy}^n - I\|^2 + \|t_{xy}^n - t_{xy}^{n*}\|^2 \quad (29)$$

where $[R_{xy}^{n*}, t_{xy}^{n*}]$ is the ground-truth motion parameters from X to Y . Given the global features G_x^n and G_y^n obtained via a two-stage feature extraction and transform network, the global feature alignment loss L_g^n can be defined as

$$L_g^n = \|\text{avg } G_x^n - \text{avg } G_y^n\| \quad (30)$$

where avg represents the average pooling operation applied on the channel. For the network, the final loss function L is a weighted sum of three parts: a global feature alignment loss L_g^n , a cycle consistency loss L_c^n , and a rigid motion loss L_m^n

$$L = \sum_{n=1}^3 \gamma^{n-1} (L_g^n + \alpha L_c^n + \beta L_m^n) \quad (31)$$

where the coefficients γ , α , and β are hyperparameters.

IV. EXPERIMENTAL RESULTS

A. Experimental Data and Setup

The performance of the proposed method is demonstrated on three datasets, including ModelNet40 [37], ShapeNetCore.v2 [38], and Real Data [39]. Consistent with the method of processing data in [20], first, 2048 points are sampled from the model in the dataset and then normalized into a sphere. After that, we take 1024 points from them to form a complete point

TABLE I
PARTIAL-TO-PARTIAL POINT CLOUD REGISTRATION RESULTS ON UNSEEN OBJECTS IN MODELNET40

Model	MSE(\mathbf{R})	RMSE(\mathbf{R})	MAE(\mathbf{R})	MSE(\mathbf{t})	RMSE(\mathbf{t})	MAE(\mathbf{t})
ICP	1134.552	33.683	25.045	0.0856	0.293	0.250
Go-ICP	195.985	13.999	3.165	0.0011	0.033	0.012
FGR	126.288	11.238	2.832	0.0009	0.030	0.008
PointNetLK	280.044	16.735	7.550	0.0020	0.045	0.025
DCP-v2	45.005	6.709	4.448	0.0007	0.027	0.020
PRNet	10.235	3.199	1.454	0.0003	0.016	0.010
IDAM	8.225	2.868	0.753	0.0003	0.018	0.004
DLF	0.481	0.694	0.368	0.00004	0.006	0.003

cloud. Subsequently, the complete point cloud on each axis is randomly transformed. The ranges of rotation transformation angle and translation distance are within $[0, 45^\circ]$ and $[-0.5, 0.5]$, respectively. Considering the fact that the sampled point cloud is imperfect due to equipment and other problems in reality, the method in [8] is used to simulate the partial point cloud: randomly put a point in the coordinate frame and select the 768 points closest to this point from the complete point cloud and the transformed point cloud to form the model point cloud and the target point cloud. Finally, we input the generated point cloud pairs into the registration network for training and testing. It is worth noting that the points are jittered and sampled independently.

To evaluate the performance of the proposed DLF model, we compare it to seven state-of-the-art algorithms for point cloud registration, including traditional algorithms (ICP [2], Go-ICP [3], and Fash Global Registration (FGR) [27]) and deep-learning-based algorithms (PointNetLK [7], DCP [6], PRNet [8], and IDAM [30]). The networks are conducted on the Tesla V100 GPU server with dual 3.2-GHz CPU and 4-GB memory. For the first two datasets, the initial learning rate and the number of iterations are set as 0.001 and 150, respectively. Moreover, we divide the initial learning rate by 10 at epochs 30, 60, and 80, respectively. For the last dataset, the number of epochs is set as 50 with a learning rate 0.0001.

In order to measure the estimation performance, three indices, including mean square error (MSE), root-mean-square error (RMSE), and mean absolute error (MAE), are computed via the estimated values and the true values of the rotation and translation parameters.

B. Experimental Comparisons

1) *ModelNet40*: ModelNet40 is an artificially synthesized dataset, which contains 12 311 computer-aided design models in a total of 40 categories. In this section, we conducted three different experiments on ModelNet40 to verify the effectiveness, generalization, and robustness of the proposed model.

a) *Unseen objects*: As split in [8], the training data and the testing data contain 9843 and 2468 models, respectively. Table I shows the partial-to-partial registration results on unseen objects. To facilitate the comparison of the performance between different algorithms, the best results are bolded. It can be seen from Table I that the performance of deep-learning-based algorithms is better than that of traditional algorithms. The proposed algorithm DLF captures informative features using

two-stage feature extraction, which facilitates the subsequent keypoint selection. In addition, DLF also introduces an OFM, which calculates the matching confidence between point pairs that gives low weights to false-positive point pairs. Therefore, it stands out among the deep-learning-based algorithms with superior performance. Fig. 5 shows the registration results of some examples for the proposed model. The blue points and the red points denote the model point cloud and the target point cloud, respectively. We can see that most points are correctly matched for two point clouds; this implies that the estimated transformation parameters are highly accurate.

b) *Unseen categories*: In order to investigate the generalization capability, for ModelNet40, the first 20 categories are used for training and the rest for testing. Table II tabulates the performance metrics of the various methods on unseen categories. As traditional algorithms are not required to learn parameters from the training set, unseen categories have few side effects on these algorithms. However, the performance of all the learning-based algorithms degrade to some extent for the lack of *a priori* knowledge of the tested categories. Nevertheless, thanks to the DLF and high weights given to inlier correspondences, DLF shows less degradation in performance and remained superior to all the other algorithms.

c) *Unseen objects with Gaussian noise*: In practical scenarios, the point cloud frequently encounters noise or outliers. To further test the robustness of the algorithm, the experiments are performed for ModelNet40 with Gaussian noise. Specifically, the Gaussian noise is independently sampled from $(0, 0.01)$ and clipped to $[-0.05, 0.05]$. Table III shows the experimental results of the partial-to-partial registration algorithms on unseen objects with Gaussian noise. It can be seen that, as Gaussian noise has a greater impact on the shape of the point cloud, the results of all the indicators are inferior to Table I. As the OFM can effectively filter mismatched pairs, DLF still achieves the highest registration accuracy.

2) *ShapeNetCore.v2*: To further demonstrate the capabilities of the model, we compared it on the ShapeNetCore.v2 with PRNet and IDAM. The ShapeNetCore.v2 dataset contains 51 127 prealigned models from 55 categories, which are split into 35 708 models for training and 15 419 models for testing. Table IV shows the registration results between PRNet, IDAM, and DLF. Compared to ModelNet40, ShapeNetCore.v2 has richer training data. Compared to Table I, all the metrics are reduced for three methods. Nevertheless, DLF trained on ShapeNetCore.v2 still gives competitive results on three metrics of rotation and translation parameters, demonstrating its strong registration ability.

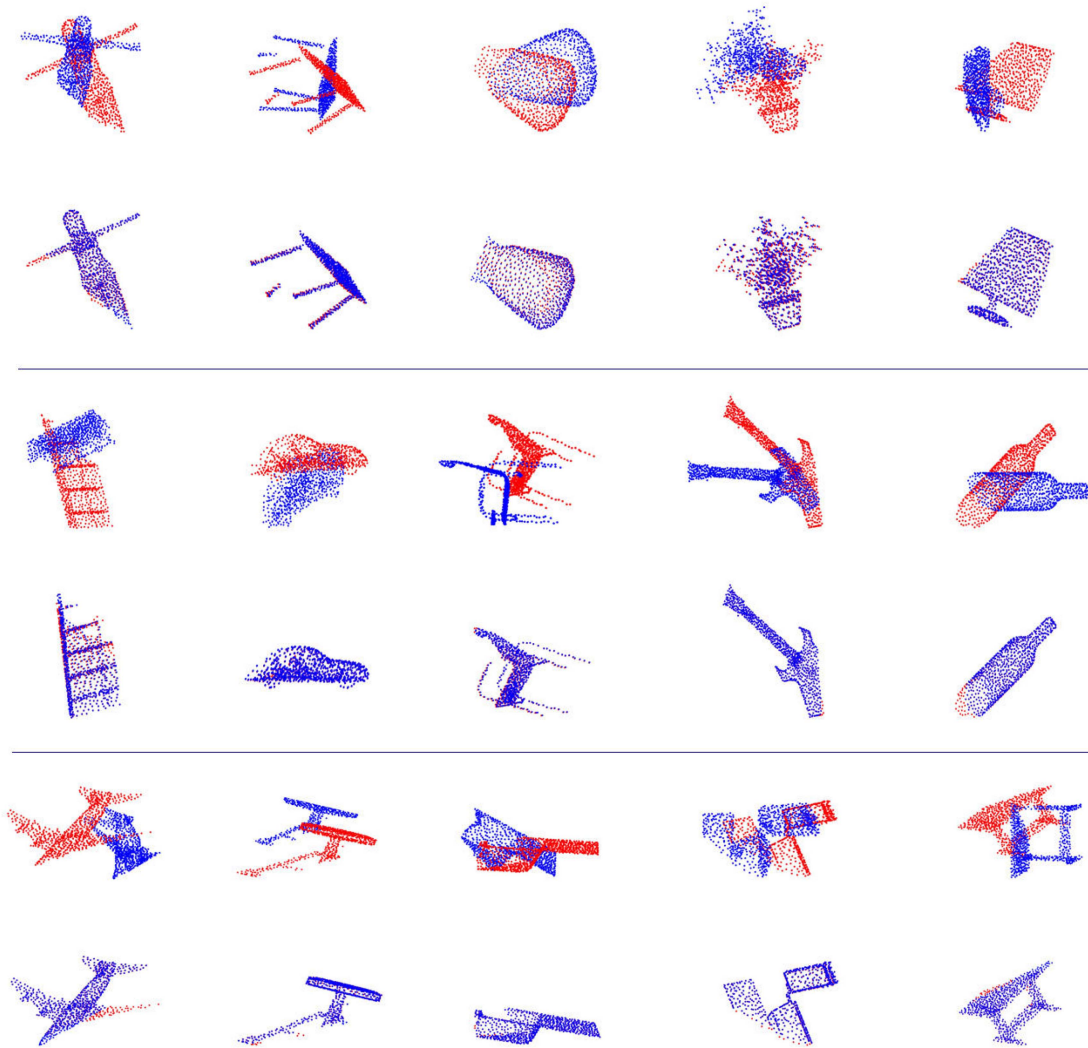


Fig. 5. Registration results of some examples for the proposed model. In each cell separated by the straight line, the upper part displays the initial position of the two partial point clouds, and the lower part displays the registration results.

TABLE II
PARTIAL-TO-PARTIAL POINT CLOUD REGISTRATION RESULTS ON UNSEEN CATEGORIES IN MODELNET40

Model	MSE(\mathbf{R})	RMSE(\mathbf{R})	MAE(\mathbf{R})	MSE(\mathbf{t})	RMSE(\mathbf{t})	MAE(\mathbf{t})
ICP	1217.618	34.894	25.455	0.086	0.293	0.251
Go-ICP	157.072	12.533	2.940	0.0009	0.031	0.010
FGR	98.635	9.932	1.952	0.0014	0.038	0.007
PointNetLK	526.401	22.943	9.655	0.0037	0.061	0.033
DCP-v2	95.431	9.769	6.954	0.0010	0.034	0.025
PRNet	24.857	4.986	2.329	0.0004	0.021	0.015
IDAM	11.546	3.398	0.837	0.0004	0.020	0.005
DLF	0.599	0.774	0.478	0.00005	0.007	0.004

3) *Real Data*: We also conduct the experiments on real data, i.e., the Stanford Bunny, Happy, and Dragon datasets, which contain ten, nine, and seven real point cloud models scanned by sensors, respectively. Compared with the ModelNet40 and the ShapeNetCore.v2 datasets, the distribution of the point cloud data acquired by sensor scanning is not uniform, which is more challenging for point cloud registration. Fig. 6 shows the registration results of the proposed method. It can be seen that for the point cloud in the real dataset, most of the points are

correctly matched. Note that we only use the model trained on ModelNet40 with unseen objects to test on the Stanford Bunny, Happy, and Dragon, which further proves the generalization ability of the proposed model.

C. Ablation Study

The above experimental results illustrate the performance of DLF for point cloud registration. The success of DLF is the

TABLE III
PARTIAL-TO-PARTIAL POINT CLOUD REGISTRATION RESULTS ON UNSEEN OBJECTS WITH GAUSSIAN NOISE IN MODELNET40

Model	MSE(\mathbf{R})	RMSE(\mathbf{R})	MAE(\mathbf{R})	MSE(\mathbf{t})	RMSE(\mathbf{t})	MAE(\mathbf{t})
ICP	1229.670	35.067	25.564	0.0860	0.294	0.250
Go-ICP	150.320	12.261	2.845	0.0008	0.028	0.029
FGR	764.671	27.653	13.794	0.0048	0.070	0.039
PointNetLK	397.575	19.939	9.076	0.0032	0.0572	0.032
DCP-v2	47.378	6.883	4.534	0.0008	0.028	0.021
PRNet	18.691	4.323	2.051	0.0003	0.017	0.012
IDAM	13.861	3.723	1.850	0.0004	0.020	0.011
DLF	3.530	1.879	0.979	0.0001	0.010	0.007

TABLE IV
PARTIAL-TO-PARTIAL POINT CLOUD REGISTRATION RESULTS ON SHAPENETCORE.V2

Model	MSE(\mathbf{R})	RMSE(\mathbf{R})	MAE(\mathbf{R})	MSE(\mathbf{t})	RMSE(\mathbf{t})	MAE(\mathbf{t})
PRNet	7.638	2.764	1.113	0.0002	0.015	0.010
IDAM	6.991	2.644	0.676	0.0002	0.013	0.003
DLF	0.354	0.595	0.336	0.00002	0.005	0.003

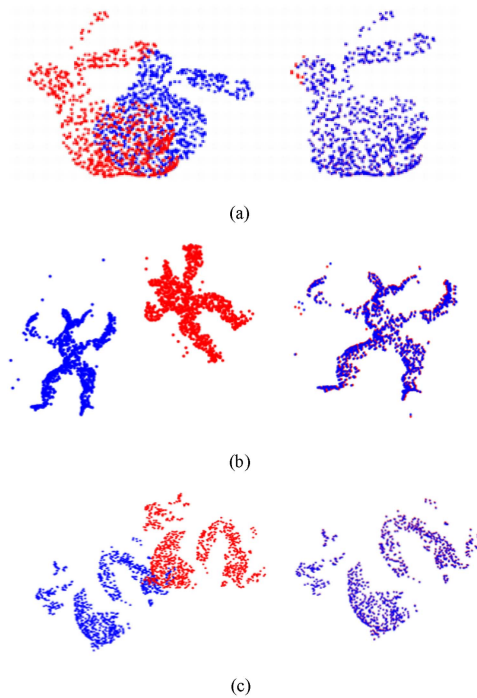


Fig. 6. Registration results of three examples in the real data of the proposed model. The left and right sides of the figure are the input partial point clouds and the transformed partial point clouds, respectively. (a) Bunny. (b) Happy. (c) Dragon.

result of its design: two-stage feature extraction (LAFE and LFA) and an OFM. Here, we elaborate the effectiveness of the three components by considering the following variants of DLF.

- 1) Only LFA is used in the point cloud registration network, denoted as DLF w/ LFA.
- 2) Both LAFE and LFA are adopted in the point cloud registration network, denoted as DLF w/ LAFE+LFA.
- 3) Both LFA and OFM are adopted in the point cloud registration network, denoted as DLF w/ LFA+OFM.
- 4) LAFE, LFA, and OFM are all adopted in the point cloud registration network, i.e., DLF.

Tables V–VII show the registration results of DLF and its three variants under the ModelNet40 dataset. The best results are shown in bold. We can observe the following.

- 1) Two-stage feature extraction is more conducive to registration than just using LFA, i.e., deep local context feature is effective.
- 2) In the experiment with Gaussian noise, when the OFM is added to the registration network, all the indicators have been greatly improved, which means that the OFM can effectively filter out mismatched point pairs to increase the robustness of the model. In addition, we present the visualization of OFM filtered outliers, as shown in Fig. 7. The pairs of points connected by green lines are the filtered outliers. As can be seen from Fig. 7, the OFM has an excellent ability to filter point cloud pairs that do not have a corresponding relationship (i.e., outliers).
- 3) The performance is best when all three modules are used in the proposed point cloud registration network, i.e., DLF, which is sufficient to validate the effectiveness as well as the reasonableness of the proposed network.

D. Efficiency

In order to compare the efficiency of the different models, the inference time of each model is calculated. We conducted efficiency tests on the point cloud with the number of points 512, 1024, and 2048, respectively. The inference time is presented in Table VIII. It can be observed that DLF is faster than FGR, which performs best among traditional algorithms. Although the speed of DLF is slightly lower than that of other deep-learning-based methods, the overall performance is competitive by considering the efficiency and accuracy.

E. Discussions and Analysis

In the above experiments, the initial angle range is set to $[0, 45^\circ]$. In order to verify the effectiveness of the proposed algorithm at a larger initial angle, we compare it with PRNet and IDAM, which are the best performing in deep learning algorithms, at an initial angle range of $[0, 60^\circ]$. The relevant

TABLE V
PERFORMANCE OF DLF AND DIFFERENT VARIANTS ON UNSEEN OBJECTS IN MODELNET40

Model	MSE(\mathbf{R})	RMSE(\mathbf{R})	MAE(\mathbf{R})	MSE(\mathbf{t})	RMSE(\mathbf{t})	MAE(\mathbf{t})
DLF w/ LFA	3.863	1.965	1.078	0.0002	0.015	0.010
DLF w/ LAFE+LFA	2.075	1.440	0.949	0.0003	0.018	0.012
DLF w/ LFA+OFM	1.284	1.133	0.435	0.00004	0.006	0.004
DLF	0.481	0.694	0.368	0.00004	0.006	0.003

TABLE VI
PERFORMANCE OF DLF AND DIFFERENT VARIANTS ON UNSEEN CATEGORIES IN MODELNET40

Model	MSE(\mathbf{R})	RMSE(\mathbf{R})	MAE(\mathbf{R})	MSE(\mathbf{t})	RMSE(\mathbf{t})	MAE(\mathbf{t})
DLF w/ LFA	5.129	2.265	1.196	0.0002	0.015	0.010
DLF w/ LAFE+LFA	2.725	1.651	1.065	0.0003	0.018	0.012
DLF w/ LFA+OFM	3.274	1.810	0.890	0.00012	0.011	0.007
DLF	0.599	0.774	0.478	0.00005	0.007	0.004

TABLE VII
PERFORMANCE OF DLF AND DIFFERENT VARIANTS ON UNSEEN OBJECTS WITH GAUSSIAN NOISE IN MODELNET40

Model	MSE(\mathbf{R})	RMSE(\mathbf{R})	MAE(\mathbf{R})	MSE(\mathbf{t})	RMSE(\mathbf{t})	MAE(\mathbf{t})
DLF w/ LFA	10.827	3.290	1.684	0.0003	0.016	0.011
DLF w/ LAFE+LFA	8.022	2.832	1.571	0.0003	0.016	0.011
DLF w/ LFA+OFM	4.635	2.153	1.134	0.0001	0.010	0.007
DLF	3.530	1.879	0.979	0.0001	0.010	0.007

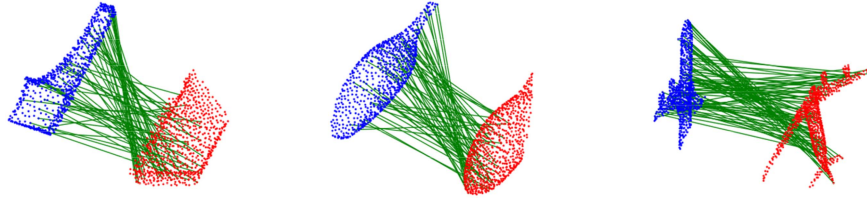


Fig. 7. Visualization of OFM filtering outliers. Filtered outlier points are connected by green lines.

TABLE VIII
INFERENCE TIME (IN SECONDS)

Points	ICP	Go-ICP	FGR	PointNetLK	DCP	PRNet	IDAM	DLF
512	0.128	14.053	0.172	0.059	0.015	0.024	0.022	0.082
1024	0.162	14.140	0.187	0.074	0.026	0.042	0.033	0.143
2048	0.231	14.212	0.185	0.084	0.063	0.087	0.048	0.297

TABLE IX
REGISTRATION EXPERIMENTAL RESULTS AT INITIAL ANGLES OF $[0, 60^\circ]$ ON UNSEEN OBJECTS WITH GAUSSIAN NOISE IN MODELNET40

Model	MSE(\mathbf{R})	RMSE(\mathbf{R})	MAE(\mathbf{R})	MSE(\mathbf{t})	RMSE(\mathbf{t})	MAE(\mathbf{t})
PRNet	62.917	7.932	3.983	0.0009	0.030	0.018
IDAM	192.210	13.864	6.076	0.0060	0.079	0.038
DLF	30.530	5.525	2.107	0.0004	0.022	0.013

settings are the same as in the experiments with Gaussian noise. The experimental results are presented in Table IX. It can be seen that the registration errors of all three algorithms rise compared to Table III when the range of initial rotation angles increases. As the OFM can increase the robustness of DLF to some extent, the performance of DLF is still advantageous compared to PRNet and IDAM.

For the structure of DLF, first, the related experiments are performed by using different numbers of branches to obtain the neighbor-attention coefficient in LAFE. The experimental setup

is consistent with that in the unseen categories. Table X shows the experimental results of DLF with different numbers of branches, in which DLF w/ Two-NAC and DLF w/ Three-NAC denote that two and three branches are adopted in LAFE, respectively. It can be clearly seen that, in LAFE, it is better to use the branch of neighbor-attention coefficients only once.

In addition, in the OFM, only the original coordinates of keypoints are used to calculate the matching relationship. As the inlier correspondences are close to each other [11], the Euclidean distance between the keypoints of the model point cloud and the target point cloud is added to the input features of the OFM to better constrain the outlier filtering. Denote DLF w/ ED as the model after adding the Euclidean distance. The experimental results are shown in Table XI. It can be seen that the registration error decreases slightly after adding the Euclidean distance feature. From the above experimental results, it indicates that the geometric constraint of two point clouds is

TABLE X
RESULTS OF REGISTRATION EXPERIMENTS USING DIFFERENT NUMBERS OF BRANCHES WITH NEIGHBOR-ATTENTION COEFFICIENT

Model	MSE(\mathbf{R})	RMSE(\mathbf{R})	MAE(\mathbf{R})	MSE(\mathbf{t})	RMSE(\mathbf{t})	MAE(\mathbf{t})
DLF	0.599	0.774	0.478	0.00005	0.007	0.004
DLF w/ Two-NAC	0.920	0.959	0.562	0.00006	0.008	0.005
DLF w/ Three-NAC	1.118	1.057	0.624	0.00010	0.010	0.006

TABLE XI
EXPERIMENTAL RESULTS WITH/WITHOUT INTRODUCING EUCLIDEAN DISTANCE

Model	MSE(\mathbf{R})	RMSE(\mathbf{R})	MAE(\mathbf{R})	MSE(\mathbf{t})	RMSE(\mathbf{t})	MAE(\mathbf{t})
DLF	0.599	0.774	0.478	0.00005	0.007	0.004
DLF w/ ED	0.567	0.753	0.465	0.00005	0.007	0.004

beneficial for outlier filtering. Therefore, how to improve outlier filtering may be investigated further by exploring the various geometric constraints.

V. CONCLUSION

In this article, an effective network was proposed for partial point cloud registration. In order to take advantage of the interdependence between each point and its neighbors, we explicitly applied the LAPE module to estimate the geometric cues in the low-level space. Accordingly, the LFA module we introduced can learn the complex geometric context representation in the high-level space more effectively. Besides, the OFM assists the registration model to filter out unmatched point pairs, which can alleviate the ambiguity in the registration. Experiments on multiple datasets demonstrated that the proposed approach has an obviously better performance than that of several state-of-the-art algorithms.

REFERENCES

- [1] T. P. Zou et al., "KAM-Net: Keypoint-aware and keypoint-matching network for vehicle detection from 2D point cloud," *IEEE Trans. Artif. Intell.*, vol. 3, no. 2, pp. 207–217, Apr. 2022.
- [2] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.
- [3] J. L. Yang, H. D. Li, D. Campbell, and Y. D. Jia, "Go-ICP: A globally optimal solution to 3-D ICP point-set registration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2241–2254, Nov. 2016.
- [4] H. Maron, N. Dym, I. Kezurer, S. Kovalsky, and Y. Lipman, "Point registration via efficient convex relaxation," *ACM Trans. Graph.*, vol. 38, no. 2, Jul. 2016, Art. no. 73.
- [5] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3-D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 1848–1853.
- [6] Y. Wang and J. M. Solomon, "Deep closest point: Learning representations for point cloud registration," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 3522–3531.
- [7] H. Goforth, Y. Aoki, R. A. Srivatsan, and S. Lucey, "PointNetLK: Robust & efficient point cloud registration using PointNet," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7156–7165.
- [8] Y. Wang and J. M. Solomon, "Prnet: Self-supervised learning for partial-to-partial registration," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 8812–8824.
- [9] H. Zhao, H. Zhuang, C. Wang, and M. Yang, "G3DOA: Generalizable 3D descriptor with overlap attention for point cloud registration," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2541–2548, Apr. 2022.
- [10] S. Ao, Q. Hu, B. Yang, A. Markham, and Y. Guo, "SpinNet: Learning a general surface descriptor for 3D point cloud registration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11748–11757.
- [11] X. Y. Bai et al., "PointDSC: Robust point cloud registration using deep spatial consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15854–15864.
- [12] Z. Gojcic, C. F. Zhou, J. D. Wegner, L. J. Guibas, and T. Birdal, "Learning multiview 3D point cloud registration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1756–1766.
- [13] D. Maturana and S. Scherer, "VoxNet: A 3-D convolutional neural network for real-time object recognition," in *Proc. IEEE Int. Conf. Intell. Robot. Syst.*, 2015, pp. 922–928.
- [14] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3-D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4490–4499.
- [15] G. Riegler, U. A. Osman, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proc. IEEE Int. Conf. Intell. Robot. Syst.*, 2017, pp. 6620–6629.
- [16] P. S. Wang, Y. Liu, Y. X. Guo, C. Y. Sun, and X. Tong, "O-CNN: Octree-based convolutional neural networks for 3-D shape analysis," *ACM Trans. Graph.*, vol. 36, no. 4, Aug. 2017, Art. no. 72.
- [17] M. Fey, J. E. Lenssen, F. Weichert, and H. Muller, "SplineCNN: Fast geometric deep learning with continuous B-spline kernels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 869–877.
- [18] H. Thomas, C. R. Qi, J. E. Deschard, B. Marcotequi, F. Goulette, and L. G. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6420–6429.
- [19] D. Ezuz, J. Solomon, V. G. Kim, and M. Ben-Chen, "GWCNN: A metric alignment layer for deep shape analysis," *Comput. Graph. Forum*, vol. 36, no. 5, pp. 49–57, Aug. 2017.
- [20] C. R. Qi, H. Su, K. C. Mo, and L. G. Guibas, "PointNet: Deep learning on point sets for 3-D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 77–85.
- [21] C. R. Qi, L. Yi, H. Su, and L. G. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5105–5114.
- [22] Y. Wang, Y. B. Sun, Z. W. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Nov. 2019.
- [23] C. Chen, L. Z. Fragonara, and A. Tsourdos, "GAPointNet: Graph attention based point neural network for exploiting local feature of point cloud," *Neurocomputing*, vol. 438, pp. 122–132, May 2019.
- [24] Q. Y. Hu et al., "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11108–11117.
- [25] K. M. Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua, "Learning to find good correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2666–2674.
- [26] J. H. Zhang et al., "Learning two-view correspondences and geometry using order-aware network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5845–5854.
- [27] Q. Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 766–782.
- [28] A. Kurobe, Y. Sekikawa, K. Ishikawa, and H. Saito, "CorsNet: 3D point cloud registration by deep neural network," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 3960–3966, Jul. 2020.
- [29] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2511–2520.
- [30] J. H. Li, C. H. Zhang, Z. Y. Xu, H. H. Zhou, and C. Zhang, "Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 766–782.

- [31] Z. J. Yew and G. H. Lee, "RPM-Net: Robust point matching using learned features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11824–11833.
- [32] R. Sinkhorn, "A relationship between arbitrary positive matrices and doubly stochastic matrices," *Ann. Math. Statist.*, vol. 35, pp. 876–879, 1964.
- [33] H. Xu, S. Liu, G. Wang, G. Liu, and B. Zeng, "OMNet: Learning overlapping mask for partial-to-partial point cloud registration," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 3522–3531.
- [34] H. Xu, N. Ye, S. Liu, G. Liu, and B. Zeng, "FINet: Dual branches feature interaction for partial-to-partial point cloud registration," 2021, *arXiv:2106.03479*.
- [35] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
- [36] E. Jang, S. X. Gu, and B. Poole, "Categorical reparameterization with Gumbel-Softmax," 2017, *arXiv:1611.01144*.
- [37] Z. R. Wu et al., "3-D shapenets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1912–1920.
- [38] A. X. Chang et al., "ShapeNet: An information-rich 3-D model repository," 2015, *arXiv:1512.03012*.
- [39] G. Turk and M. Levoy, "Zippered polygon meshes from range images," in *Proc. Annu. Conf. Comput. Graph. Interact. Techn.*, 1994, pp. 311–318.



Yu-Xin Zhang received the M.Sc. degree in control science and engineering from the School of Electrical Engineering and Automation, Anhui University, Hefei, China, in 2022.

Her research interests include machine learning and image and signal processing.



Zhan-Li Sun (Member, IEEE) received the Ph.D. degree in control science and engineering from the University of Science and Technology of China, Hefei, China, in 2005.

Since 2006, he has worked with The Hong Kong Polytechnic University, Hong Kong, Nanyang Technological University, Singapore, and National University of Singapore, Singapore. He is currently a Professor with the School of Electrical Engineering and Automation, Anhui University, Hefei. His research interests include machine learning and image and

signal processing.

Dr. Sun is an Associate Editor for IEEE ACCESS.



Zhigang Zeng (Fellow, IEEE) received the Ph.D. degree in systems analysis and integration from the Huazhong University of Science and Technology, Wuhan, China, in 2003.

He is currently a Professor with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. He is also with the Key Laboratory of Image Processing and Intelligent Control of the Education Ministry of China, Wuhan. He has authored or coauthored more than 100 international journal papers. His current research interests

include theory of functional differential equations and differential equations with discontinuous right-hand sides, and their applications to dynamics of neural networks, memristive systems, and control systems.

Dr. Zeng was an Associate Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS from 2010 to 2011. He has been an Associate Editor for IEEE TRANSACTIONS ON CYBERNETICS since 2014 and IEEE TRANSACTIONS ON FUZZY SYSTEMS since 2016. He has been a Member of the Editorial Board of *Neural Networks* since 2012, *Cognitive Computation* since 2010, and *Applied Soft Computing* since 2013.



Kin-Man Lam (Senior Member, IEEE) received the Associateship in electronic engineering with distinction from the Hong Kong Polytechnic University (formerly called Hong Kong Polytechnic), Hong Kong, in 1986, the M.Sc. degree in communication engineering from the Department of Electrical Engineering, Imperial College of Science, Technology and Medicine, London, U.K., in 1987, and the Ph.D. degree in electrical engineering from the Department of Electrical Engineering, University of Sydney, Sydney, NSW, Australia, in 1996.

From 1990 to 1993, he was a Lecturer with the Department of Electronic Engineering, The Hong Kong Polytechnic University. In October 1996, he joined the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, as an Assistant Professor, where he became an Associate Professor in 1999 and has been a Professor since 2010. His current research interests include human face recognition, image and video processing, and computer vision.

Dr. Lam is a Member of the Organizing Committee and Program Committee of many international conferences. He is a General Co-Chair of 2012 IEEE International Conference on Signal Processing, Communications, and Computing. He is a Member of the Board of Governors of the Asia-Pacific Signal and Information Processing Association and the Director–Student Services of the IEEE Signal Processing Society. He is an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING, *APSIPA Transactions on Signal and Information Processing*, and *EURASIP International Journal on Image and Video Processing*.