·Review·

Deep learning based point cloud registration: an overview

Zhiyuan ZHANG[1], Yuchao DAI[1*], Jiadai SUN[2]

1. *School of Electronics and Information, Northwestern Polytechnical University, Shaanxi* 710129*, China*

2. *School of Computer Science and Technology, Northwestern Polytechnical University, Shaanxi* 710129*, China*

*** Corresponding author,** daiyuchao@nwpu.edu.cn

Received: 23 March 2020 **Revised:** 8 May 2020 **Accepted:** 11 May 2020

Supported by the National Key Research and Development Program of China under Grant (2018AAA0102803); the National Natural Science Foundation of China under Grants (61871325, 61420106007, 61671387).

Citation: Zhiyuan ZHANG, Yuchao DAI, Jiadai SUN. Deep learning based point cloud registration: an overview. Virtual Reality & Intelligent Hardware, 2020, 2(3): 222—246
DOI: 10.1016/j.vrih.2020.05.002

Abstract Point cloud registration aims to find a rigid transformation for aligning one point cloud to another. Such registration is a fundamental problem in computer vision and robotics, and has been widely used in various applications, including 3D reconstruction, simultaneous localization and mapping, and autonomous driving. Over the last decades, numerous researchers have devoted themselves to tackling this challenging problem. The success of deep learning in high-level vision tasks has recently been extended to different geometric vision tasks. Various types of deep learning based point cloud registration methods have been proposed to exploit different aspects of the problem. However, a comprehensive overview of these approaches remains missing. To this end, in this paper, we summarize the recent progress in this area and present a comprehensive overview regarding deep learning based point cloud registration. We classify the popular approaches into different categories such as correspondences-based and correspondences-free approaches, with effective modules, i.e., feature extractor, matching, outlier rejection, and motion estimation modules. Furthermore, we discuss the merits and demerits of such approaches in detail. Finally, we provide a systematic and compact framework for currently proposed methods and discuss directions of future research.

Keywords Overview; Point cloud registration; Deep learning; Graph neural networks

1 Introduction

In geometric computer vision, point cloud registration is a key task in many applications, including robotics[1], simultaneous localization and mapping (SLAM)[2], autonomous driving[3], and medical imaging[4] (Figure 1). Hence, in this paper, we focus on the point cloud registration problem under a rigid motion, which is defined as follows: Given two point clouds, point cloud registration aims at finding a rigid transformation to align one point cloud to another, potentially obfuscated by noise and partiality[5]. Note that point cloud registration under a non-rigid transformation is an important and separate research branch in geometric computer vision, where various deep and non-deep approaches such as CPD[6], FlowNet3D[7], and HPLFlowNet[8] have been developed. A comprehensive discussion on this topic is beyond the scope of the present study and is left for future research.

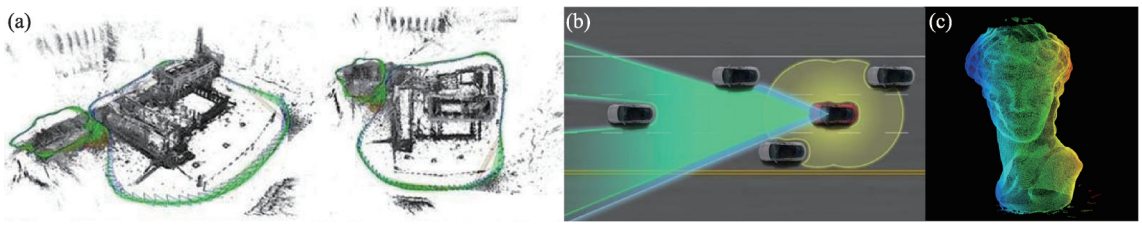

Figure 1 Applications of point cloud registration (figures taken from the Internet). (a) SLAM; (b) Autonomous driving;(c) 3D modeling.

The point cloud registration has been a well-studied problem with a long history, and there have been numerous advanced methods described in the literature. The most seminal approach is the iterative closest point (ICP) [9], which estimates the rigid transformation and updates the point cloud correspondences iteratively to refine the solution in a coarse-to-fine manner. The ICP represents a major milestone in a point cloud registration, which has been extensively applied in various applications, and different variants have been proposed. However, the standard ICP requires a good initialization and usually converges to the local minima. ICP-style methods generally suffer from this drawback. The ICP and its variants developed over the last 30 years but before the deep learning revolution have been summarized[10,11].

The success of deep learning in high-level vision tasks has been extended to various geometric computer vision tasks such as ego-motion estimation[12], stereo matching[13], optical flow[14], and multi-view stereo[15]. However, the extension to 3D point clouds is not straightforward owing to the significant differences between 2D images and 3D point clouds. Sparsity, irregular patterns, unordered permutations, and an imbalanced distribution constitute the major obstacles.

The mesh[16], voxel[17], and multi-view convolutional neural network (CNN) method[18] are pioneering approaches in deep learning based point cloud processing, and many variants such as VoxelNet[19], VV-Net[20], the submanifold sparse convolutional network[21], and MV3D[22] have achieved an impressive performance. PointNet[23] and PointNet++[24] represent two milestones, which first apply deep learning to a point cloud in a straightforward manner and solve the unordered permutation problem through a symmetric function. PointNet[23] generates a descriptor for each point, or a global feature of the entire point cloud. A holistic feature is used for point cloud classification; the corresponding network is called PointNet-Cla herein for a clearer description. In addition, the per-point feature is utilized in the semantic segmentation task, and the corresponding network is called PointNet-Seg. Meanwhile, PointNet++[24] is a key technique used to extract local information in a point cloud. The key stage is the set abstraction module, which consists of sampling, grouping, and PointNet[23]. Another heuristic algorithm is the graph neural network. Such networks consider each point in a point cloud as a vertex of a graph and generate directed edges for the graph based on the neighbors of each point. Feature learning is then applied in the spatial or spectral domains[25]. The dynamic graph CNN (DGCNN) [26] is a graph-based method that constructs a dynamic graph operation to update the relationship between vertexes in the feature space. Whereas PointNet[23] essentially extracts information based on the embedding of each point in a point cloud independently, DGCNN[26] explicitly incorporates the local geometry into its representation. The typical PointNet, set abstraction in PointNet++ , and a graph-based network are demonstrated in Figure 2. However, in addition to a symmetric function strategy, other methods have also been proposed. In PointCNN[27], an χ-transformation is learned, which weights the input features associated with the points and translates the permutation of the original point cloud into a latent and potentially canonical order. Similarly, the tree-structure is utilized in the Kd-network[28] and OctNet[29]. Inspired by a continuous convolution operation, a permutation invariant convolution PointConv[30] was proposed, which treats

convolution kernels as nonlinear functions of the local coordinates of 3D points comprising weight and

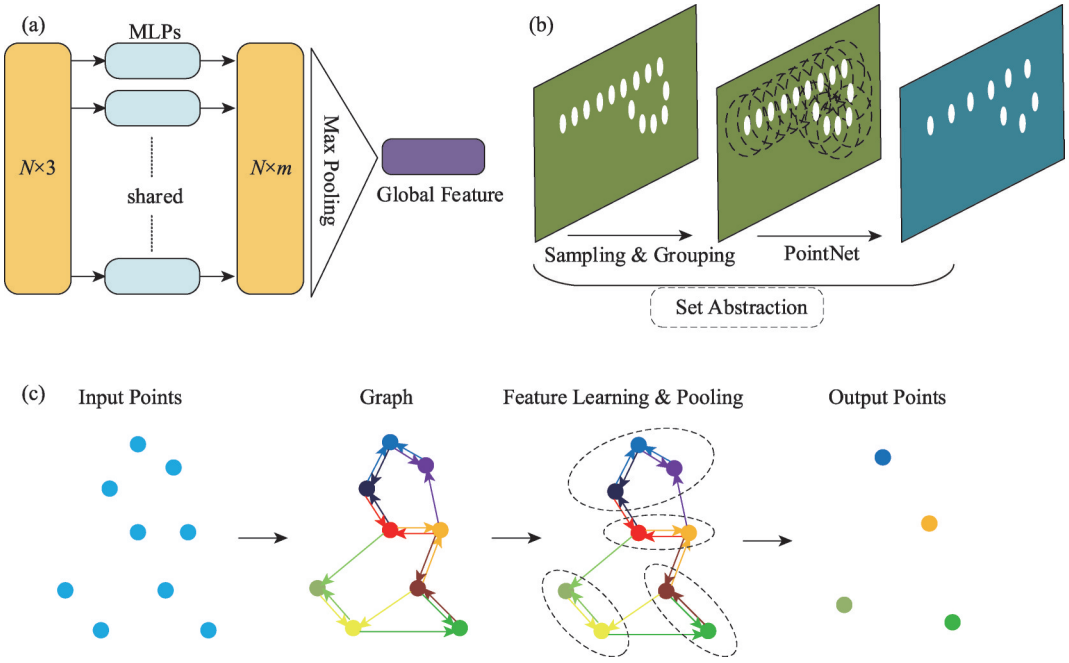

Figure 2 Typical network for point cloud processing based on deep learning. (a) Typical PointNet network; (b) Set abstraction structure; (c) Typical graph-based network.

density functions. In addition, these weights and density functions are learned using two multi-layer perceptron (MLP) networks. PCNN[31] is another method applying convolutional neural networks to point clouds by defining the extension and restriction operators. SO-Net[32] is a permutation invariant network that utilizes the spatial distribution of the point clouds by building a self-organizing map.

There have been an increasing number of methods applying deep learning to address the various problems of point cloud, including 3D shape classification[23], 3D object detection and tracking[33], and 3D point cloud segmentation[24]. A survey of deep learning based point cloud processing has also been provided[34], although the point cloud registration problem has not been involved. In this paper, we focus on the problem of the deep learning based point cloud registration and present a contemporary overview of state-of-the-art methods on this topic.

The major contributions of this study can be summarized as follows:

(1) We present a comprehensive survey of recent state-of-the-art approaches in deep learning based point cloud registration, providing a systematic view of recent developments. We present this survey according to whether the method is correspondences-based or correspondences-free. In particular, in the correspondences-based part, four modules, namely, a feature extractor, matching, outlier rejection, and motion estimation, are described in detail. In addition, some effective refinement strategies are discussed.

(2) To the best of our knowledge, this is the first survey on the deep learning based point cloud registration problem. Existing survey studies have either focused on traditional methods[10,11] or other tasks based on deep learning[34].

The remainder of this survey is organized as follows. In Section 2, we formulate the registration problem mathematically and briefly introduce traditional techniques. Section 3 provides a comprehensive discussion of state-of-the-art methods. The datasets and metrics widely used for this problem are discussed in Section 4. In Section 5, the performances are compared. Finally, we discuss the flourishing of point cloud registration based on deep learning, provide prospects for future research, and offer some concluding remarks in Section 6.

2 Preliminary knowledge

2.1 Notations and formulation

Three-dimensional point cloud registration aims at estimating a rigid transformation between several partially overlapping point clouds to associate sets of data into a common coordinate system. This allows integrating data from multiple sensors and viewpoints into a bigger model, such as a high precision point cloud map and a complete mesh of objects.

In this paper, suppose we have two point clouds, X (source) and Y (target), with N_X and N_Y points, respectively. Here, X represents the same shape or scene as Y, but is potentially noisy and incomplete. We aim to find a 3D rigid body motion to transform X into X', such that X' best overlaps Y. A rigid motion can be represented by many forms including a quaternion, angle-axis, rotation matrix, and translation vector. The solution represented by rotation matrix $R \in SO_3$ and translation vector $t \in \mathbb{R}^3$ is the most popular.

2.2 Traditional point cloud registration

The ICP[9] is the best-known algorithm for solving rigid registration problems, which alternates between finding the point cloud correspondences and solving the least-squares problem to update a rigid alignment. However, the ICP algorithm often stalls, resulting in a suboptimal result or falling into the local minima. Therefore, numerous variant ICP-based methods have been proposed to remedy these drawbacks. In a previous study[11], the ICP-variants were summarized into six phases, namely, *key point selection*, *matching*, *correspondences weighting*, *outlier rejection*, *error metric assignment*, and *motion estimation*. Note that, in addition to the widely used point-to-point matching metric, other matching metrics such as point-to-plane and plane-to-plane metrics, which are helpful in constructing correspondences from a different perspective, have been developed. However, in deep learning based point cloud registration, the point-to-point matching based metric remains the most commonly applied principle. Therefore, we confine our discussion to point-to-point matching unless otherwise stated. In addition, traditional hand-crafted registration algorithms developed over the last 30 years are introduced in detail[10], whereas we focus on the introduction of deep learning based solutions. For completeness, however, we briefly mention recent studies related to the six aspects above (Figure 3).

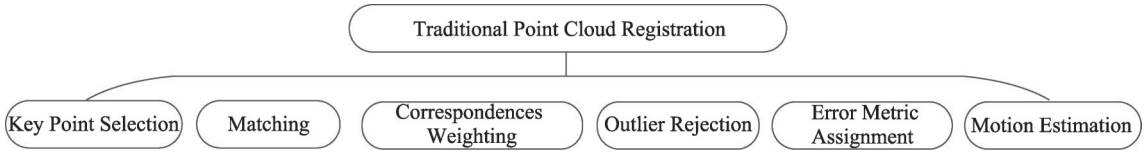

Figure 3 Six phases used in traditional point cloud registration.

For the selection of key points, interest point methods are used to compute and compare local descriptors to estimate the alignment[35,36]. Although these methods are computationally favorable, their usage is often limited to point cloud data having identifiable and unique features that are persistent between input point clouds[37−39].

For the matching stage, k-d tree and Ak-d tree have been used[40,41] to estimate the point correspondences according to the semantic information[42−44] incorporating visual or intensity information.

For the weighting of the correspondences, some 'soft' methods have been proposed, including assigning lower weights to pairs with greater point-to-point distances, weighting based on the compatibility with normals, and weighting based on the expected effect of scanner noise on the uncertainty in the error metric.

Zhou et al. proposed calculating the transformation parameters for each point[45], which can be seen as a highly overparameterized weight adjustment.

The method used to reject outliers may have an effect on the accuracy and stability. Rejecting the worst n% of pairs based on certain proposed metrics[46], semantic information is introduced to help identify the correct correspondences, resulting in a reduction of the iteration number[7].

A well-considered error metric (loss function) plays a leading role in minimizing the error and finding optimal solutions. A point-to-point[9] error is the simplest metric, which has a closed-form solution. A point-to-plane error metric was also proposed[47,48], and a new symmetrized objective function that achieves the simplicity and computational efficiency of point-to-plane optimization was introduced[49].

To minimize the error, the ICP can be jointly viewed as an optimization algorithm used to search for a matching and rigid transformation. Hence, Fitzgibbon et al. proposed using the Levenberg-Marquardt algorithm to optimize the objective function directly, which can yield an improved solution[50]. ICP-style methods are prone to falling into the local minimum from a non-convexity. To find a good optimum within the polynomial time, Go-ICP[51] uses a branch-and-bound (BnB) strategy to search a rigid transformation space SE_3. Other methods have attempted to identify the global optima using a Riemannian optimization[52], convex relaxation[53], and mixed-integer programming[54].

To summarize, traditional point cloud registration methods have made considerable progress, although a bottleneck still occurs.

3 Deep learning based point cloud registration

In this section, we discuss the registration methods on point cloud using deep learning strategies in detail, and recently proposed methods are shown in chronological order in Figure 4. In addition, as shown in Figure 5, the structure of this section is summarized as follows. Deep point cloud registration methods can be classified into two categories, correspondences-based methods and correspondences-free methods. First, we introduce methods without correspondences, which are unusual in the traditional point cloud registration field. These data-driven algorithms regress the rigid motion parameters $R \in SO_3$ and $t \in \mathbb{R}^3$ depending on the differences between the global features of two input point clouds. However, numerous recent deep point cloud registration methods, including DeepVCP[3], deep closest point (DCP)[5], and iterative matching point (IMP)[55], apply a traditional framework, that is, the ICP and ICP-style framework,

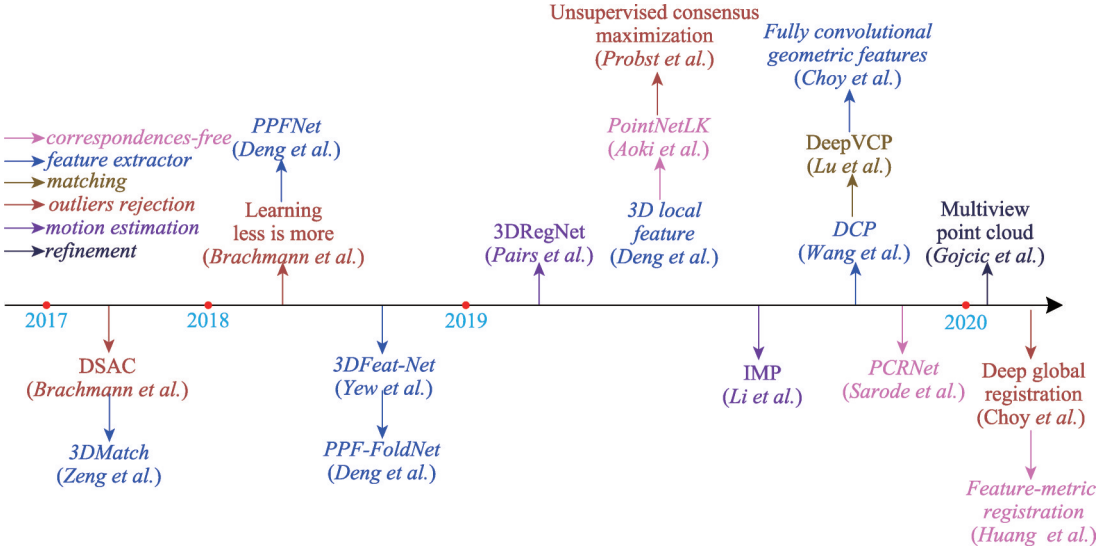

Figure 4 Chronological overview of 3D point cloud registration networks.

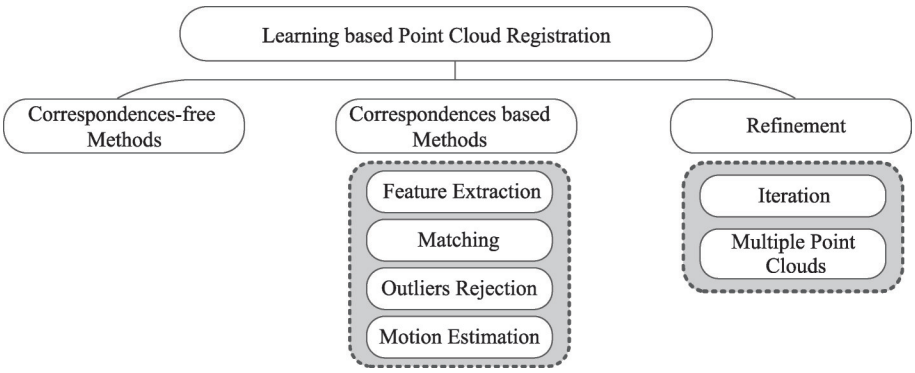

Figure 5 Taxonomy of deep learning based point cloud registration approaches.

which estimate the rigid motion parameters based on explicit correspondences. Learning-based methods improve traditional frameworks, which consist of four parts: a feature extractor, matching, outlier rejection, and motion estimation. In addition to these two categories, some extra special refinement mechanisms are proposed to improve the performance, including an iterative approach and multi-view constraints. The partial point cloud registration problem is finally introduced.

3.1 Correspondences-free methods

The key pipeline of correspondences-free methods is regressing the rigid motion parameters by searching for the difference between global features of two input point clouds. The key stage is the global feature summary, which must be sensitive to the pose. Another important stage is how to solve the motion parameters from such a difference, which is divided into two categories depending on whether they are based on deep learning.

In PointNetLK[56], the global features of input point clouds X and Y, denoted as $\Phi(X)$ and $\Phi(Y)$, are extracted by the PointNet-Cla[23] network. A derivation theory is then presented to solve R and t. Finally, $\left\|\left(G_{est}\right)^{-1} G_{gt} - I_4\right\|_F$ is the loss function, where G_{est} is the prediction, and G_{gt} is the ground truth. In this method, the PointNet[23] network is impressionable to the pose, which becomes a huge advantage in a registration task, and the Jacobian can be approximated through a finite difference gradient computation. This approach allows the application of the computationally efficient inverse compositional Lucas-Kanade algorithm[56].

A recently proposed method, feature-metric registration[57], inherits the framework of PointNetLK, which enforces the optimization by minimizing a feature-metric. This model focuses on a feature extractor, which utilizes an encoder-decoder mechanism. For the principle of two rotated copies of a point cloud, the encoder module should generate different features, and the decoder can recover the different features to their corresponding rotated copies. This model is trained using a semi-supervised or unsupervised approach, which requires limited or no registration label data.

A similar work, PCRNet[58], also utilizes PointNet[23] to extract global features. The framework is shown in Figure 6. However, in the feature alignment module, a data-driven technique is used. First, two global features are concatenated, and then five fully connected layers are applied along with an output layer of the dimension of the

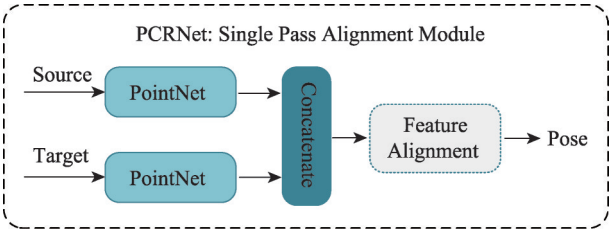

Figure 6 Kernel framework of PCRNet.

parameterization chosen for the pose. This deep principle is more effective than PointNetLK[59]. Another similar approach is AlignNet[60], which concentrates on real scene data.

To summarize, correspondences-free methods are straightforward. Their performance strongly relies on the extracted feature descriptors. The accuracy, sensitivity to motion, and outlier robustness of the features determine the lower limit of the methods. At the same time, the align module determines the upper limit, which plays an important role in information integration and the final generation of the results. Because the correspondences-free methods directly regress the relative pose from two holistic point cloud features, the performance is ideal when the point clouds are the same, except for the pose. However, when there is a large difference between the two input point clouds, the performance will greatly depend on the feature extractor network. Therefore, the generalization of these regression-based approaches will be confined by the generalizability of the feature learning methods applied. Recently proposed methods are extremely effective on synthetic data, whereas the future direction is extended to a real scene with a low overlap. Therefore, the major criticism of such correspondences-free approaches is their generalization capability, which aligns with other regression-based geometric vision problems such as stereo matching[13], optical flow estimation[14], and scene flow estimation[8].

3.2 Correspondences-based methods

Algorithms based on correspondences occupy a significant proportion of deep point cloud registration methods. These frameworks are inspired by traditional methods and consist of four major modules: feature extraction, matching, outlier rejection, and motion estimation. However, as a common phenomenon, many learnable registration methods based on the correspondences are not end-to-end, and confuse traditional modules with learning modules. Therefore, in this section, we introduce the methods according to each individual module.

3.2.1 Feature extraction

Different from correspondences-free methods, which extract the global feature per-cloud, the correspondences-based methods extract the per-point or per-patch embeddings of the two input point clouds to generate a mapping and estimate a rigid transformation. However, because the PointNet-Seg network simply summarizes the descriptor by concatenating the per-point information and the global information, which is extreme, this network is rarely used in a correspondences-based point cloud registration. In fact, the local information and other useful information are concentrated.

3DMatch[61] is a pioneering approach, using a 3D CNN to learn the mapping from a volumetric 3D patch to a 512-dimensional feature representation that serves as the descriptor for the local region. For each point of interest, a 3D volumetric representation is extracted for the local region surrounding it. Each 3D region is converted from its original representation into a volumetric voxel grid, and the representations are then fed into the 3D CNN for the final features. However, the volumetric representation has obvious drawbacks, including being constrained by its resolution owing to the data sparsity and the computational cost. Consequently, such a representation tends to be replaced by methods that operate on the original point cloud in a straightforward manner.

Zan et al. proposed a feature extractor based on the smoothed density value (SDV) voxelization[62]. Given a point x in point cloud X, its local spherical support $S \subset X$ is selected such that $S = \left\{x_i : \left\| x_i - x \right\|_2 \leqslant r\right\}$, where r denotes the radius of the local neighborhood used for estimating the local reference frame. This operation is beneficial for feature rotation invariance. The local point is then transformed into the local reference

frame, the estimation of which is notated as $x_i' \in S'$. The SDV voxel grid as a 3-dimensional matrix $X^{SDV} \in \mathbb{R}^{W \times H \times D}$ whose elements $\left(X^{SDV}\right)_{jkl} = :x_{jkl}$ represent the SDV of the corresponding voxel is computed using the Gaussian smoothing kernel with bandwidth h.

$$x_{jkl} = \frac{1}{n_{jkl}} \sum_{i=1}^{n_{jkl}} \frac{1}{\sqrt{2\pi}\, h} exp \frac{-\left\| c_{jkl} - x_i' \right\|_2^2}{2h^2} s.t. \left\| c_{jkl} - x_i' \right\|_2 < 3h \tag{1}$$

where n_{jkl} denotes the number of points $x_i' \in S'$ that lie within the distance $3h$ of the voxel centroid c_{jkl}. Finally, the SDV representation is fed into the stacked convolutional layers and a batch normalization layer, followed by L_2 normalization, to produce the unit length of the local feature descriptors. Similarly, the local reference frame is obtained through a learning strategy[63].

A representative approach is DeepVCP[3], the structure of which is shown in Figure 7. DeepVCP applies PointNet++[24] and a mini-PointNet[23,64,65] structure to learn the descriptors. Given a point cloud with N_X points, the 3D Euclidean coordinates are the input, and a $N_X \times 32$ tensor $F(X)$ is the feature descriptor generated by PointNet++[24]. The most important modification is mini-PointNet, which is helpful in summarizing the local information. Mini-PointNet consists of three stacks of fully connected layers and a max-pooling layer to obtain the feature descriptors. The input is a local information vector $L(X)$ of size $N_X \times K \times 36$, which refers to the local coordinate, the intensity, and the 32-dimensional $F(X)$ feature descriptor. Each point with K-nearest neighbors is utilized to generate a more informative 32-dimensional feature $\Phi(X) = MLP\left(Max\left(L(X)\right)\right)$ and $\Phi(Y)$.

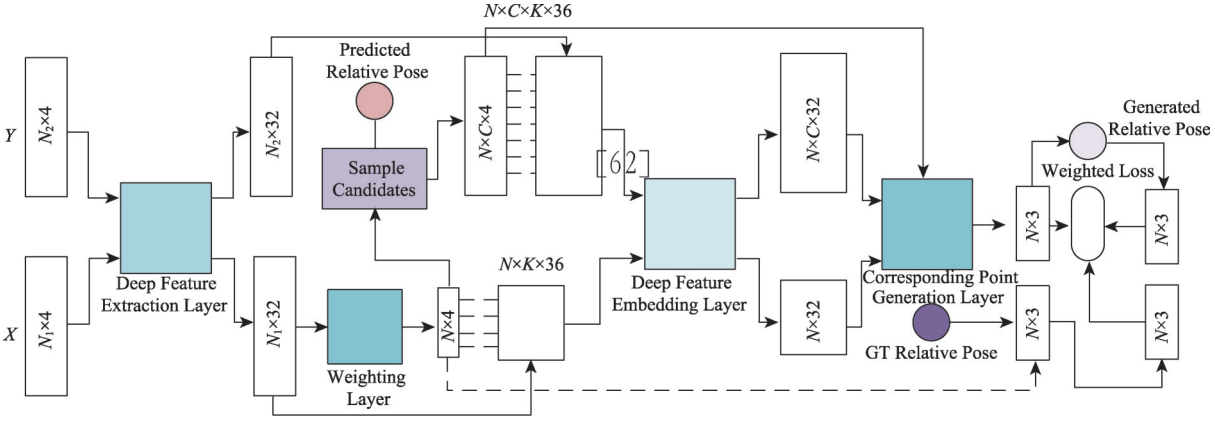

Figure 7 Kernel framework of DeepVCP.

The PointNet++[24] structure is used in 3DFeat-Net[66]. However, because of the high complexity of the multi-set abstractions, only one set abstraction structure is applied. To this end, the sampling layer samples a set of points $x_{i1}, x_{i2}, \cdots, x_{ik}$ from the input point cloud, which are regarded as clustering centers. Next, k clusters of points are generated by the grouping layer. These clusters are used as receptive fields to compute the local descriptors. After applying max-pooling to obtain a cluster feature, which is concatenated with the per-point features to incorporate the context information, a single fully connected layer and max-pooling are applied for a more global contextual feature, which is referred to as f_k of cluster C_k. In other words, another extra symmetric function is added to the feature extractor stage in 3DFeat-Net[66]. The addition of contextual information improves the effectiveness of the descriptor.

Inspired by the PointNet[23] structure, particularly a max-pooling operation, Deng et al. proposed PPFNet[65], which focuses on the rotation invariant descriptors for correct correspondences in a registration task. Given a reference point $x_r \in X$, define a local region $\Omega \in X$ and collect a set of points $\{m_i\} \in \Omega$ within this local vicinity. Then, the normal of the point set is computed[67], each neighboring point x_i is paired with

the reference x_r, and the point pair feature (PPF) is computed[68]. Therefore, the initial features $F_r = \{x_r, n_r, x_i, \cdots, n_i, \cdots, \psi_{ri}, \cdots\}$, where $\psi_{ri} = \left(\|d\|_2, \angle(n_r, d), \angle(n_i, d), \angle(n_r, n_i)\right)$ is the PPF. The input to PPFNet is N local patch features, the first module is mini-PointNet, and another max-pooling is applied. The global feature is then concatenated with the local feature. PPFNet[65] is analogous to PointNet[23], the input of which is PPFs rather than 3D Euclidean coordinates. This operation leads to a rotation invariance while strongly relying on a normal vector estimation.

IMP[55], a variant of DGCNN[26] is selected as the feature extractor. Denoting the point in point cloud X as x_i, $i \in [1, N_X]$. In addition, the information of x_i is simply the 3D Euclidean coordinate. The input to the network is a $N_X \times 3$ tensor and the output is a $N_X \times d$ tensor, where each point feature is of d dimensions, and each layer in the IMP network[55] operates as the following function:

$$u_n(x) = \frac{1}{K} \sum_{x' \in N(x)} g_n\left(u_{n-1}(x) - u_{n-1}(x')\right) \tag{2}$$

where $u_n(x)$ is the feature output by the n-th layer, the input to the first layer is the point coordinates, and g_n is a multi-layer perceptron. Denote the set of K-nearest neighbors for x as $N(x)$. It is worth noting that the relative pose is purely utilized, whereas the absolute position coordinate of the current point is concatenated with the relative pose to extract a descriptor in the original DGCNN[26]. As a result, the feature extraction module is translation invariant.

With DCP[5], the main framework of which is shown in Figure 8, the feature extractor module consists of two stages, the DGCNN and an asymmetric function, i.e., a transformer module[69]. The point cloud features are calculated through the DGCNN and are referred to as F_X and F_Y. Wang et al.[5] claimed that this improves the feature effectiveness of the matching by making the features task-specific, that is, the features are changed depending on the particularities of X and Y together rather than embedding them independently. Analogous to attention-based models[70–72], a module for learning the co-contextual information by capturing the self-attention and conditional attention is designed. This attention module learns a function, $\phi: \mathbb{R}^{N \times P} \times \mathbb{R}^{N \times P} \rightarrow \mathbb{R}^{N \times P}$, where $N_X = N_Y = N$ is assumed, and P is the embedding dimension, which provides new embeddings of the point clouds as follows:

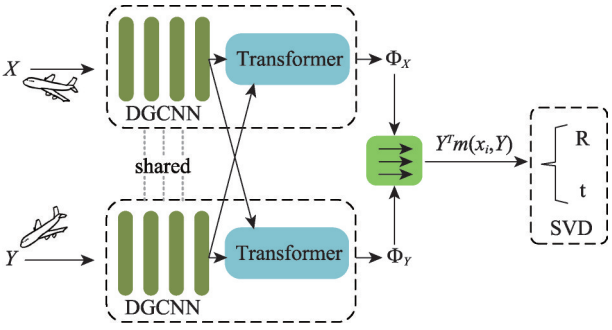

Figure 8 Kernel framework of DCP.

$$\begin{cases} \Phi_X = F_X + \phi\left(F_X, F_Y\right) \\ \Phi_Y = F_Y + \phi\left(F_Y, F_X\right) \end{cases} \tag{3}$$

Treat ϕ as a residual term, providing additive changes to X and Y depending on the order of the inputs. This map, $F_X \mapsto \Phi_X$, modifies the features associated with the points in Y in a fashion that is knowledgeable regarding the structure of Y. An asymmetric function ϕ is then given by a transformer[69].

Another effective feature extraction method is fully convolutional geometric learning[73]. This method is first applied to 2D data[74], and later applied to 3D semantic segmentation[21,75–78]. Choy et al. proposed variants for fully convolutional features that integrate negative-mining into contrastive and triplet losses[73]. New losses such as hardest-contrastive and hardest-triplet are presented to learn the feature extractor. Fully convolutional networks can be attributed to three factors. First, fully convolutional networks are effective because they share intermediate activations across neurons with overlapping receptive fields. Second,

neurons in fully convolutional networks can have larger receptive fields because they are not constrained by operating on separately extracted and processed patches. Third, fully convolutional networks produce a dense output, which is well-suited for tasks that call for a detailed scene characterization.

The evolution of the feature extractor is clear and heuristic. PointNet-Cla, PointNet-Seg, and DGCNN, for example, which are designed for point cloud classification, segmentation, and tracking, respectively, are transformed into a registration task through certain adjustments, particularly the attention mechanism. Naturally, more state-of-the-art feature extractors in other applications can be used, such as PointSift[79], SO-Net[32], and KD-Network[28]. Another direction is designing the feature extractor specifically for the registration problem.

3.2.2 Matching module

The matching stage is another key module in a point cloud registration. The rigid motion parameters R and t can be solved using singular value decomposition (SVD) based on the correct correspondences, which has been mathematically proven to be optimal. However, in traditional methods, the common principle is to search the most similar points y_i in Y to x_i in X as the corresponding pair. However, owing to the sparsity and partiality, two point clouds do not always have a point-to-point correspondence. Therefore, the virtual point method is creatively proposed.

In DeepVCP[3], a virtual point is generated according to the initial R_0 and t_0. DeepVCP focuses on a subset constituted by key points denoted as $S \in X$, $N_S \ll N_X$, where N_S and N_X are the point set sizes. The process is given briefly as follows. First, S is transformed using the input initial parameters R_0 and t_0, and generates the corresponding point x_i' for x_i. The neighboring space of x_i' is divided into $\left(\frac{2r}{s}+1, \frac{2r}{s}+1, \frac{2r}{s}+1\right)$ 3D grid voxels, where r is the searching radius and s is the voxel size, both of which are predefined. Denote the centers of the 3D voxels as y_j', $j = 1,...,C$, which are considered candidate corresponding points. Next, all candidates are fed into the feature extractor. A three-layer 3D CNN is applied to learn the similarity between the features of the source point and the candidate points. More importantly, it can smooth (regularize) the matching volume and suppress the matching noise. The softmax operation is applied to convert the matching costs into the probabilities. Finally, the target corresponding point y_i is calculated through a weighted-sum operation as follows:

$$y_i = \frac{1}{\sum_{j=1}^{C} w_j} \sum_{j=1}^{C} w_j y_j' \tag{4}$$

where w_j is the weight of each candidate corresponding to point y_j'.

A similar operation is proposed in DCP[5]. However, the candidates are not 3D voxel centers in the neighbor. All real points in Y are considered. The similarity is generated through a dot product operation, which operates as a weight. Finally, the matching principle can be summarized as follows:

$$x_i' = \sum_{j=1}^{N_Y} w_j y_j, w_j = D\left(\Phi\left(x_i\right), \Phi\left(y_j\right)\right) \tag{5}$$

where N_Y is the size of point cloud Y, and $D(\cdot, \cdot)$ is the unnormalized cosine similarity.

However, not all methods rely on virtual points, and IMP[55] is presented to select the most similar point as a corresponding point. With IMP, a similarity matrix is formed using a dot product operation.

$$M\left(i,j\right) = D\left(\Phi\left(x_i\right), \Phi\left(y_j\right)\right) \tag{6}$$

where $M\left(i,j\right)$ is the unnormalized cosine similarity between x_i and y_j. A softmax function is applied on each row of M to obtain a probability distribution over all points in Y for each point in X.

$$P(i,j) = soft\max\left(M(i,:)\right)[j] \tag{7}$$

where $P(i,j)$ represents the probability of y_j being the corresponding point of x_i. For each point x_i in X, the point y_j in Y that has the highest probability (largest similarity) to be the corresponding point is selected.

The matching stage is a key stage, although it is easily ignored in deep learning methods. In traditional methods, the nearest principle is decisively applied in a 3D Euclidean or feature space. In recently proposed deep learning methods, the virtual points have made progress, which transfers the attention to the weight coefficients. Therefore, a potential direction is the learning of the weight parameters.

3.2.3 Outlier rejection

In fact, incorrect matching points, called outliers, exist at all times even in most state-of-the-art methods with an advanced matching strategy. Because of the sparsity and partiality of the point cloud data, it is common for not all points in point cloud X to have the corresponding points in point cloud Y. At the same time, because of noise, the many-to-one phenomenon is not rare. Outliers significantly deteriorate the registration performance. In traditional methods, RANSAC[80,81] is the most widely used robust fitting algorithm, which utilizes the maximum consistency as a supervised signal to filter out the outlying matching pairs. The solution to a deep learning method relies on the supervised and unsupervised principles.

(1) Supervised

In 3DRegNet[82], the classification principle is transplanted to distinguish the outliers and inliers, as shown in Figure 9. The input to the classification block is a set of 3D point correspondences $\left(x_i, y_i\right), i = 1, \cdots, N$. The output is the weighting coefficients, which represent the probability that the correspondence is an inlier. The classification block is inspired by the network architecture[83]. Each point correspondence (6 tuples) is processed by a fully connected layer with 128 ReLU activation functions. There is weight sharing for each individual point correspondences, and the output has dimensions of $N \times 128$, where 128-dimensional features are generated from every point correspondence. The $N \times 128$ output is then passed through 12 deep ResNet blocks[84], with weight-sharing fully connected layers instead of convolutional layers. Another fully connected layer with ReLU is used, which is followed by tanh $(\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \in (-1, 1))$ units to produce the weights $w_i \in [0, 1)$. In addition, w_i is the probability aiming at the i-th correspondence. Finally, a predefined threshold is applied to filter out the outliers.

Another approach, deep global registration[85], is similar, designing a novel network to learn the weights of the correspondences. To avoid the drawback in which traditional methods disregard a local geometric

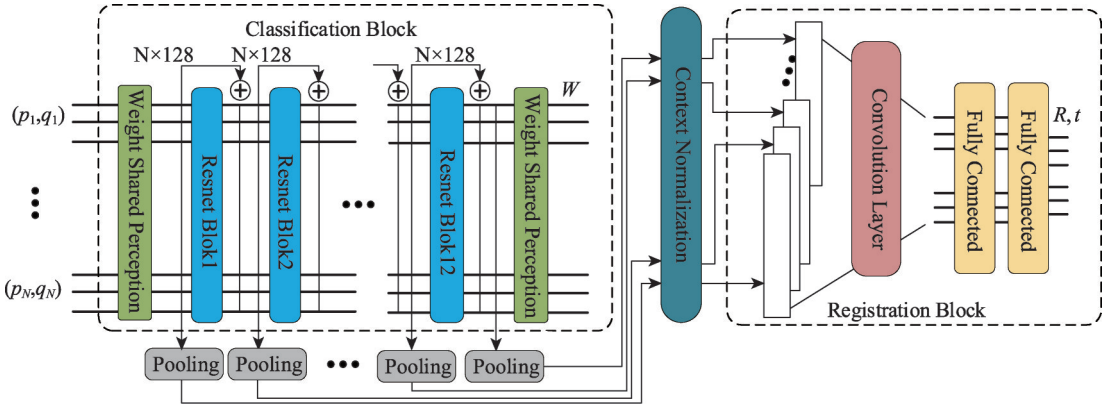

Figure 9 Classification block and registration block in 3DRegNet.

structure, a high-dimensional convolutional network is proposed. The point cloud is converted into regular voxels, and a 6-dimensional convolutional network architecture is presented to encode the local information. This method is outstanding but relies on voxel representation.

(2) Unsupervised

The 3DRegNet[82] strategy is supervised. However, in traditional registration method, an unsupervised pipeline is more common. Unsupervised learning of the consensus maximization[86] for 3D vision problems is proposed, which provides a supervisory signal for learning the consensus maximization from the data.

Given a neural network, as shown in Figure 10, indicated as $w_\theta(\chi):\mathbb{R}^{m\times n}\rightarrow[0,1]^m$ and parametrized by θ, χ is the sample set of correspondences, the learned prediction score of which is indicated as w_i. The network maximizes the number of inliers $(w_i\rightarrow 1)$, while rejecting outliers $(w_i\rightarrow 0)$. To this end, a differentiable supervised signal is defined that requires neither point-wise labels nor knowledge about the ground truth transformation between correspondences. The constraint can be relaxed by minimizing the singular values of a Vandermonde matrix $M_d(\chi)$[87]. For descending singular values $\sigma_1,\sigma_2,\cdots,\sigma_s$ of $M_d(\chi)$, the trailing r singular values must be zero. The network can be modeled as follows: After construction of the Vandermonde matrix $M_d(\chi)\in\mathbb{R}^{m\times s}$[87], every row i is weighted with the corresponding inlier probability w_i. Then, the last r singular values of the weighted Vandermonde matrix are computed using a differentiable SVD operation. Therefore, the final empirical loss $l(\theta,\chi)$ is given as follows:

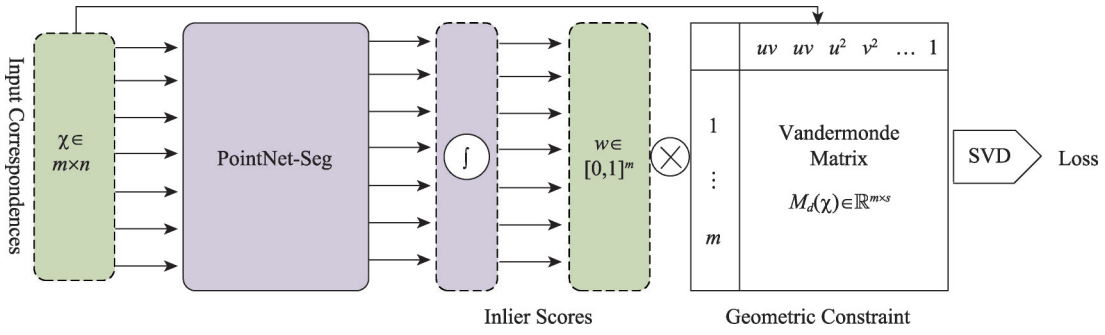

Figure 10 Unsupervised outlier rejection module.

$$l(\theta,\chi)=-\left\|w_\theta(\chi)\right\|_1+\lambda\sigma_{s-k}\left(diag\left(w_\theta(\chi)\right)M_d(\chi)\right) \tag{8}$$

In fact, the outlier rejection stage is challenging but important. In addition, the unsupervised learning strategy is more notable, and is more suitable in various applications. A differentiable extension of RANSAC[80], DSAC[81,88] is heuristic. The development of more effective unsupervised methods is therefore imperative.

3.2.4 Motion estimation

A rigid motion estimation is the final stage of the point cloud registration task. The motion parameter has different representations, such as the quaternion, angle-axis, rotation matrix R, and translation vector t. In addition, R and t are the most popular, and have been proven to be optimally solvable using SVD based on correspondences. At the same time, based on an end-to-end learning strategy, some methods have been proposed that utilize a regression strategy for estimating the motion.

(1) Regression

In 3DRegNet, the input to the network block is the features extracted from the point correspondences. The pooling operation is used to extract meaningful features from each layer of the classification block. Max-pooling performs the best in comparison with other options such as average pooling. After the pooling is

completed, context normalization and feature map concatenation are applied[83]. The features from the context normalization are then passed to a convolutional layer. The output is then generated as six variables: $v = \left(v_1, v_2, v_3\right)$ and $t = \left(t_1, t_2, t_3\right)$.

Deng et al. proposed obtaining the motion using RelativeNet[89], which is essentially a fully connected network. However, it is worth mentioning that this regression network is straightforward but effective because the input is considered to be related to motion only. This is constructed based on the difference between the rotation invariant network PPF-FoldNet and rotation related network PC-FoldNet[90,91].

(2) Singular Value Decomposition (SVD)

In the rigid alignment problem, assume Y is transformed from X using an unknown rigid motion. Denote the rigid transformation as $R \in SO_3$ and $t \in \mathbb{R}^3$. For ease of expression, supposing $N_X = N_Y$, which is indicated as N, the target function can then be formulated as follows:

$$R^*, t^* = \operatorname{argmin} \frac{1}{N} \sum_{k=1}^{N} \left\| R x_k + t - y_k \right\|^2 \tag{9}$$

Then, the centroids of X and Y are defined as follows:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i, \bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i \tag{10}$$

The cross-covariance matrix H is given by

$$H = \sum_{i=1}^{N} \left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)^T \tag{11}$$

First, we use the SVD to decompose $H = U\Sigma V^T$. Then, the alignment minimizing loss function is given in a closed form as follows:

$$R = VU^T, t = -R\bar{x} + \bar{y} \tag{12}$$

Here, take the convention that $U,V \in SO_3$, where Σ is diagonal but potentially signed. This accounts for orientation reversing choices of H. This classic orthogonal Procrustes problem assumes that the point sets are matched with each other, that is, x_i should be mapped to y_i in the final alignment for all i.

(3) Weighted SVD

Although the SVD decomposition has been proved to be optimal, its success depends on the assumption that all correspondences play an equal role. In fact, the correspondences are not always precise. For example, some points on a flat surface may have features comparable to numerous points, and should be given less weight. In addition, we should pay more attention to those distinctive points that have fewer possible correspondences, such as corners. Therefore, with IMP[55], the inverse of the entropy of the probability distribution is used as the weight:

$$w_i = -\frac{1}{\sum_j P\left(i,j\right) \log\left(P\left(i,j\right)\right) + \varepsilon} \tag{13}$$

where ε is a small constant for numerical stability. Then, (9) becomes the following:

$$R^*, t^* = \operatorname{argmin} \frac{1}{N} w_k \left\| R x_k + t - y_k \right\|^2 \tag{14}$$

This optimization problem can be solved in a closed form. Let

$$H = \sum_{k=1}^{N_X} \left(x_k - \bar{x}\right) W \left(y_k - \bar{y}\right)^T \tag{15}$$

where $\dot{x}$ and $\dot{y}$ are the means. In addition, W is a diagonal matrix with $W\left(i,i\right) = w_i$. Applying SVD to H,

$$H = U\Sigma V^T \tag{16}$$

The solution can be obtained using

$$R^* = VU^T, t^* = -R^*\bar{x} + \bar{y} \tag{17}$$

3.3 Refinement

In addition to the pipeline discussed above, other techniques are proposed to refine the performance. These are unit modules that can be appended to these state-of-the-art registration algorithms. In this section, we provide the most representative methods, which significantly improve the performance.

3.3.1 Iteration

The precision in matching the corresponding points and motion estimation depends on the relative poses of the input point clouds. In fact, the error of the rotation estimation increases with an increase in the initial rotation angle. This motivated the development of an iterative refinement[37,54]. Denote the rotation and translation output by iteration m as R_m and t_m. In iteration $(m+1)$, we transform the source point cloud with R_m and t_m to obtain a new point cloud X_m that has a smaller angle and translation gaps into target Y. Next, X_m is fed into a new feature extraction module with the same structure but different weights to extract new features in this new pose. Again, a similarity matrix is formed, the correspondences are found, and the new R_{m+1} and t_{m+1} are solved. Because X_m is already close to Y, the correspondences found by matching the features extracted under this new pose will be more accurate. The final prediction is formed by compositing all intermediate values of R_m and t_m as follows:

$$\begin{cases} R = \prod_{m=1}^{M} R_m \\ t = \sum_{m=1}^{M-1}\left(t_m \prod_{n=m+1}^{M} R_n\right) + t_m \end{cases} \tag{18}$$

This iterative refinement strategy not only improves the accuracy of the estimation result, it is also beneficial in dealing with noise, even for a partial-to-partial registration[92].

3.3.2 Multi-view point cloud registration

A multi-view point cloud registration means considering multiple point clouds at the same time, and utilizing the pose constraint between every two point clouds to refine the performance[93]. All points are formulated as a graph G and each point cloud is denoted as a vertex. The global transformation parameters can be estimated by dividing the problem into rotation[79] and translation[28] synchronization. In addition, a differentiable, closed-form solution is summarized based on a spectral relaxation[28,79,80,82].

3.3.3 LUT interpolation MLP

The most popular neural network unit in point cloud processing with deep learning is a multi-layer perceptron (MLP). However, traditional MLP consists of layers of a matrix-vector product operation followed by a nonlinearity. For these reasons, the computation of the embedding dominates most of the process. To this end, Yusuke et al. proposed a novel framework that computes the embedding using a linear combination of basis functions stored in a lookup table (LUT), which is called LUT interpolation MLP (LUTI-MLP), and is significantly more efficient than MLP[94].

3.4 Point cloud registration under partial overlap

In this subsection, a point cloud registration under a partial overlap is introduced separately owing to its particularity. A partial point cloud refers to an incomplete point cloud, which means that the objects observed are not entirely consistent within the two input point clouds. In this case, it is much more challenging, particularly with a deep learning method, to estimate the rigid motion because of the limited

overlap between the two point clouds. The correspondences-free method is impracticable owing to the drastic differences between the two global features in addition to the pose. The conventional correspondences-based methods are not satisfactory because only extremely limited correspondences can be observed.

To solve this hard problem, key point correspondences and an iteration refinement strategy have been utilized. In PRNet[92], Wang et al. proposed searching for the key points by comparing the L_2 norms of the learned features, and then estimating the correspondences iteratively in a coarse-to-fine manner. It is worth mentioning that the Gumbel-Softmax[95] strategy is applied to improve the quality of the correspondences.

The partial point cloud registration problem with deep learning remains a challenging problem with extremely few studies conducted in this area. In addition, paying more attention to optimizing the correspondence matrix influenced by a low overlap and outliers will be an interesting research topic in the future.

4 Datasets and metrics

4.1 Datasets

In this section, we introduce common datasets for a 3D point cloud registration. Datasets are indispensable in evaluating the performance across different metrics. The 3D point cloud datasets for a registration task can be divided into two categories, namely, synthetic data and real scene data, which are degraded by noise. The real scene data are obtained using LiDAR, an RGB-D camera directly, or multi-view images indirectly, including outdoor and indoor scenes. We mainly provide essential information of the datasets in the tables below. More detailed information is given in the references.

Table 1 Common and representative datasets with subsets

Datasets	Real/Synthetic	Subsets	Connotation
ShapeNet[96]	Synthetic	ShapeNetCore	51300 3D models from 55 categories
		ShapeNetSem	12000 models from 270 categories
ModelNet[97]	Synthetic	ModelNet10	CAD models from 10 categories
		ModelNet40	CAD models from 40 categories
		Aligned 40-Class Subsets	CAD models from 40 categories
Redwood[22,44,98]	Real	Robust Reconstruction of Indoor Scenes	Two models of indoor scenes
	Synthetic	A Large Dataset of Object Scans	401 models from 10 categories
	Real	Indoor LiDAR-RGBD Scan Dataset	
Make3D Range Image Data[99]	Synthetic/real	Make3D Image and Laser Depth Map	Consists of outdoor scenes (about 1000), indoor (about 50), synthetic objects (approximately 7000), etc.
		Image and Laser and Stereo	
		Image and 1D Laser	
		Image and Depth	
		Video and Depth	

4.2 Metrics

In the point cloud registration task, the evaluation strategy can be divided into two categories. The first focuses on the feature extractor and matching. The second is straightforward, concentrating on a rigid motion estimation.

(1) Evaluation metric for feature extractor

Feature-match recall. The feature-match recall measures the percentage of fragment pairs that can

Table 2 Common and representative datasets without subsets

Datasets	Real/ Synthetic	Ingredients	Connotation
KITTI[100]	Real	Consists of images, optical flows and 3D point clouds	
Intrinsic3D Dataset[101]	Real	RGB-D sequences and reconstructed 3D models in five scenes: lion, gate, hieroglyphics, tomb statuary, and bricks	Camera-to-world pose is supported
Apollo-SouthBay Dataset[3]	Real	Covers various scenarios including residential areas, urban downtown areas, and highways	
ScanNet[76]	Real	RGB-D dataset that consists of 1513 scenes, annotated into 21 categories	1201 scenes for training, 312 for testing
Stanford[102]	Real		A huge dataset contains many 3D models
Oxford RobotCar Dataset[103]	Real		The push-broom 2D scans are accumulated into 3D point clouds using the associated GPS/INS poses
ETH Dataset[104]	Real		Contains largely unstructured vegetation
3DMatch Dataset[61]	Real	62 RGB-D scene reconstructions from some existing datasets	Contains eight sets of scene fragments with ground truth pose
Matterport3D[105]	Real	Contains 10800 panoramic views from 194400 RGB-D images of 90 building-scale scenes	
SUN3D[106]	Real		A large-scale RGB-D video database with camera pose and object labels
ScanObjectNN[107]	Real	Including approximately 15000 objects categorized into 15 categories with 2902 unique object instances	
SceneNN[108]	Real	More than 100 indoor scenes	
Princeton Shape Benchmark[109]	Synthetic	Version 1 contains 1814 models	907 models for training and 907 for testing

recover the pose with high confidence. Mathematically, this is defined as follows:

$$R_{fa} = \frac{1}{M}\sum_{s=1}^{M} 1\left(\left[\frac{1}{\left|\Omega_s\right|}\sum_{(i,j)\in\Omega_s} 1\left(\left\|T^* x_i - y_j\right\| < \tau_1\right)\right] > \tau_2\right) \tag{19}$$

where M is the number of fragment pairs, Ω_s is a set of correspondences between fragment pairs s, x and y are the 3D coordinates from the first and second fragments, and $T^* \in SE_3$ is the ground-truth pose. In addition, τ_1 is the inlier distance threshold, and τ_2 is the inlier recall threshold.

Registration recall. The registration recall takes a set of overlapping fragments with a ground truth pose and measures how many overlapping fragments can be correctly recovered by the matching algorithm. Specifically, the registration recall uses the following error metric between estimated fragments $\{i,j\}$ and the corresponding pose estimation $\hat{T}_{i,j}$ to define a true positive:

$$E = \sqrt{\frac{1}{\Omega^*}\sum_{(x^*,y^*)\in\Omega^*}\left\|\hat{T}_{i,j}x^* - y^*\right\|^2} < \tau_3 \tag{20}$$

where Ω^* is a set of corresponding ground-truth pairs in fragments $\{i,j\}$, and, x^* and y^* are the 3D coordinates of the ground-truth pair. For fragments $\{i,j\}$ with overlap, τ_3 is the threshold used to judge whether the pair is correct.

The recall is important because it is possible to improve the precision with better pruning, as noted in several studies[60,110].

(2) Evaluation metric for rigid motion estimation

Supposing the rotation matrix and translation vector are the target parameters that are most popular in the registration task. The most common error metrics are the mean relative angular error (MRAE/°) and mean

relative translation error (MRTE/m), which are calculated as follows:

$$MRAE = mean\left(\cos^{-1}\left(\frac{trace\left(R_{pre}^{-1}R_{gt}\right)-1}{2}\right)\right), MRTE = mean\left(\left\|t_{pre}-t_{gt}\right\|_2\right) \tag{21}$$

where the subscripts *pre* and *gt* represent the prediction and ground truth, respectively. However, there are some other more detailed metrics, such as the mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE) between the ground truth and predicted values. Ideally, all error metrics above should be zero if the rigid alignment is perfect.

5 Performance

In this section, the performance of common methods based on the above metrics is given. It is worth mentioning that not all indexes come from the original study. Because the point cloud registration in deep learning is not mature, and different datasets have been tested, it is impractical to obtain comprehensive results from only the original paper. Therefore, some indexes in the tables below are extracted from comparative experiments described in the referenced studies.

Because the experiments were tested under different settings, it is difficult to make a horizontal comparison. Therefore, limited conclusions can be drawn as follows:

(1) According to Table 3, for the 3DMatch benchmark, 3DMatch[61] < PPFNet[65] < PPF-FoldNet[91] < 3D local feature[89] < FCGF[73] in the feature-match recall metric.

(2) According to Table 4, for the 3DMatch benchmark, 3DMatch + RANSAC < PPFNet + RANSAC < FCGF + RANSAC in the registration recall metric.

(3) According to Tables 5, 6, and 7, aiming at the rotation and translation results, in a synthetic dataset ModelNet40, PointNetLK[56] < DCP[5] in several experiment settings include full data, unseen object categories, and Gaussian noise data. In addition, in a real scene dataset, KITTI, random sampling + 3DMatch < 3DFeat-Net[66] < FCGF + RANSAC.

(4) Considering the application of tracking and motion estimation, the computation time is an essential factor. However, a quantitative computation time relies heavily on the device, and only a few experiment results have been provided[5,73]. Thus, we can only compare some qualitative results. In the 3DMatch benchmarks, 3DMatch > PPF-FoldNet > FCGF. In the ModelNet40 dataset, as the point cloud size increases, the experiment results change from PointNetLK > DCP to DCP > PointNetLK.

By analyzing the performance of the methods mentioned above, the pros and cons are elicited as follows. Compared to the 3DMatch[61] method, which describes the 3D space using a voxel resulting in a limited resolution and precision, the PPFNet[65] operates the original point cloud data and learns the local descriptors for a pure geometry and is highly aware of the global context. In addition, PPFNet is able to consume raw point clouds to exploit the full sparsity. PPF-FoldNet[91]

Table 3 Performance comparison in feature-match recall metric

Metrics	Methods	Datasets: 3DMatch/%
Feature-match recall	3DMatch	59.6
	PPFNet	62.3
	PPF-FoldNet	68.0
	3D local feature	74.6
	FCGF	95.2
	3DMatch + RANSAC	66.8
	PPFNet + RANSAC	71.0
	3D local feature + RANSAC	69.0
	3D local feature + RelativeNet	77.7

Table 4 Performance comparison in registration recall metric

Metrics	Methods	Datasets: 3DMatch/%
Registration recall	3DMatch + RANSAC	66.8
	PPFNet + RANSAC	71.0
	FCGF + RANSAC	82.0

Table 5 Performance comparison in MARE metric

Metrics	Methods	Datasets			
		Oxford RobotCar	KITTI	ETH	Apollo-SouthBay
MRAE	Random sampling + 3DMatch	2.02	1.21	4.71	
	3DFeat-Net	1.07	0.57	1.56	0.076
	FCGF + RANSAC		0.17		
	DeepVCP				0.056

Table 6 Performance comparison in MRTE metric

Metrics	Methods	Datasets			
		Oxford RobotCar	KITTI	ETH	Apollo-SouthBay
MRTE	Random sampling+3DMatch	0.616	0.377	0.292	
	3DFeat-Net	0.300	0.259	0.156	0.061
	FCGF+RANSAC		0.049		
	DeepVCP				0.018

Table 7 Performance comparison on ModelNet40 dataset as reported for DCP[5]

Setting	Methods	Rotation			Translation		
		MSE	RMSE	MAE	MSE	RMSE	MAE
Full dataset	**PointNetLK**	227.87	15.10	4.23	0.000487	0.022065	0.005404
	DCP	1.31	1.14	0.77	0.000003	0.001786	0.001195
Unseen object categories	**PointNetLK**	306.323975	17.502113	5.280545	0.000784	0.028007	0.007203
	DCP	9.923701	3.150191	2.007210	0.000025	0.005039	0.003703
Gaussian noise	**PointNetLK**	256.155548	16.004860	4.595617	0.000465	0.021558	0.005652
	DCP	1.169384	1.081380	0.737479	0.000002	0.001500	0.001053

is an unsupervised learning strategy for 3D local descriptors under a pure point cloud geometry. This method is fast, more robust to sparsity than PPFNet, and necessitates neither supervision nor a sensitive local reference frame. Another superiority is the low bottleneck of the point cloud size. Here, 3D local features for a direct pairwise registration method[89] inherit the advantages of PPF-FoldNet, and the method is effective in challenging real scene datasets with a better generalization and a dramatic speed-up. FCGF[73] provides a fully convolutional geometric feature, which is compact, captures a broad spatial context, and scales to large scenes. FCGF is robust in both indoor and outdoor datasets and is much faster than PPF-FoldNet and 3DMatch. 3DFeat-Net[66] can work in a real scene dataset, and is a weakly supervised learning-based approach; however, this method underperforms compared with FCGF. DeepVCP[3] is built upon virtual points, and operates in real scenes while strongly depending on a high-quality initialization. PointNetLK[59] and PCRNet[58] are correspondences-free methods, and utilize the differences between the global features to solve the rigid motion, and improve the performance through an iterative strategy. These methods are robust and clear, particularly PointNetLK, which maintains a better performance as the initial motion and noise increase. DCP[5] outperforms PointNetLK in terms of the precision of the estimation, and is a correspondences-based method, applied in experiment settings of full, noisy, and unseen object categories data.

6 Conclusion and discussion

To summarize, two different approaches have been obtained according to the methods described in the literature. (1) Focus has been shifted from a single module, i.e., the feature extractor, to the entire process. The feature extractor module is important and easily achieves the superiority of the deep learning strategy,

such as 3DMatch[61] and PPFNet[65]. However, the research to the entire registration task has recently become popular, such as DCP[5] and DeepVCP[3]. (2) There is a big gap between synthetic scenes and real scenes. The methods on synthetic and real scene dataset focus on different aspects of registration problem, especially the impact of outliers. For example, DCP conducts tests under ModelNet40, which improves the performance significantly while ignoring outliers. However, an unsupervised consensus maximization method and DeepVCP take outliers and noise into consideration.

The current deep learning based point cloud registration methods still suffer from some obvious drawbacks. First, the feature extraction modules are variants from a module designed for other tasks, such as point cloud classification, segmentation, and tracking. However, the feature extractor in the registration task plays a fundamental role, which is expected to be specific, such as FPFH[39] and PPFH[110] used in traditional methods. PPFNet[65] is a pioneering approach, but relies on a normal vector estimation. Therefore, an advanced feature extractor focusing on a registration problem based on point cloud Euclidean coordinates is a valuable issue. Second, current methods based on correspondences are not end-to-end, mixing the learning modules with traditional modules. However, an advanced and effective deep learning framework for the entire process is being pursued. Thus, the exploration of other modules with deep learning is a meaningful area of future study. Finally, the above methods are mostly supervised, whereas traditional methods, such as ICP, are always unsupervised. It is reasonable to believe that unsupervised deep learning methods are more promising.

As a conclusion, this paper focuses on a deep learning based point cloud registration, providing a contemporary survey of other state-of-the-art methods. Finally, we summarized a unified framework and introduced the detailed merits and demerits of the modules.

References

1 Pomerleau F, Colas F, Siegwart R. A review of point cloud registration algorithms for mobile robotics. Foundations and Trends® in Robotics. 2015, 4(1):1−104
DOI:10.1561/2300000035

2 Fioraio N, Konolige K. Realtime visual and point cloud slam. In: Proceedings of the RGB-D workshop on advanced reasoning with depth cameras at robotics: Science and Systems Conf. (RSS). 2011

3 Lu W X, Wan G W, Zhou Y, Fu X Y, Yuan P F, Song S Y. DeepVCP: an end-to-end deep neural network for point cloud registration. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, Korea (South), IEEE, 2019, 12−21
DOI:10.1109/iccv.2019.00010

4 Min Z, Wang J L, Meng M Q H. Robust generalized point cloud registration using hybrid mixture model. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). Brisbane, QLD, Australia, IEEE, 2018, 4812−4818
DOI:10.1109/icra.2018.8460825

5 Wang Y, Solomon J. Deep closest point: learning representations for point cloud registration. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, Korea (South), IEEE, 2019, 3522−3531
DOI:10.1109/iccv.2019.00362

6 Myronenko A, Song X B. Point set registration: coherent point drift. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(12): 2262−2275
DOI:10.1109/tpami.2010.46

7 Liu X Y, Qi C R, Guibas L J. FlowNet3D: learning scene flow in 3D point clouds. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA, IEEE, 2019, 529−537
DOI:10.1109/cvpr.2019.00062

8 Gu X Y, Wang Y J, Wu C R, Lee Y J, Wang P Q. HPLFlowNet: hierarchical permutohedral lattice FlowNet for scene flow estimation on large-scale point clouds. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern

Recognition (CVPR). Long Beach, CA, USA, IEEE, 2019, 3249–3258
DOI:10.1109/cvpr.2019.00337

9 Besl P J, McKay N D. A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992, 14(2): 239–256
DOI:10.1109/34.121791

10 Bellekens B, Spruyt V, Berkvens R, Weyn M. A survey of rigid 3D pointcloud registration algorithms. In: AMBIENT 2014: the Fourth International Conference on Ambient Computing, Applications, Services and Technologies. Rome, Italy, 2014, 8–13

11 Rusinkiewicz S, Levoy M. Efficient variants of the ICP algorithm. In: Proceedings Third International Conference on 3-D Digital Imaging and Modeling. Quebec City, Quebec, Canada, IEEE, 2001, 145–52
DOI:10.1109/im.2001.924423

12 Zhou T, Brown M, Snavely N, Lowe D G. Unsupervised learning of depth and ego-motion from video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017, 1851–1858
DOI: 10.1109/CVPR.2017.700

13 Shaked A, Wolf L. Improved stereo matching with constant highway networks and reflective confidence learning. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA, IEEE, 2017, 4641–4650
DOI:10.1109/cvpr.2017.730

14 Dosovitskiy A, Fischer P, Ilg E, Häusser P, Hazirbas C, Golkov V, Smagt P V D, Cremers D, Brox T. FlowNet: learning optical flow with convolutional networks. In: 2015 IEEE International Conference on Computer Vision (ICCV). Santiago, Chile, IEEE, 2015, 2758–2766
DOI:10.1109/iccv.2015.316

15 Huang P H, Matzen K, Kopf J, Ahuja N, Huang J B. DeepMVS: learning multi-view stereopsis. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA, IEEE, 2018, 2821–2830
DOI:10.1109/cvpr.2018.00298

16 Yi L, Su H, Guo X W, Guibas L. SyncSpecCNN: synchronized spectral CNN for 3D shape segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA, IEEE, 2017, 6584–6592
DOI:10.1109/cvpr.2017.697

17 Maturana D, Scherer S. VoxNet: a 3D Convolutional Neural Network for real-time object recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Hamburg, Germany, IEEE, 2015, 922–928
DOI:10.1109/iros.2015.7353481

18 Su H, Maji S, Kalogerakis E, Learned-Miller E. Multi-view convolutional neural networks for 3D shape recognition. In: 2015 IEEE International Conference on Computer Vision (ICCV). Santiago, Chile, IEEE, 2015, 945–953
DOI:10.1109/iccv.2015.114

19 Zhou Y, Tuzel O. VoxelNet: end-to-end learning for point cloud based 3D object detection. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA, IEEE, 2018: 4490–4499
DOI:10.1109/cvpr.2018.00472

20 Meng H Y, Gao L, Lai Y K, Manocha D. VV-net: voxel VAE net with group convolutions for point cloud segmentation. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, Korea (South), IEEE, 2019, 8499–8507
DOI:10.1109/iccv.2019.00859

21 Graham B, Engelcke M, Maaten L V D. 3D semantic segmentation with submanifold sparse convolutional networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA, IEEE, 2018, 9224–9232
DOI:10.1109/cvpr.2018.00961

22 Chen X Z, Ma H M, Wan J, Li B, Xia T. Multi-view 3D object detection network for autonomous driving. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA, IEEE, 2017, 6526–6534
DOI:10.1109/cvpr.2017.691

23 Charles R Q, Su H, Mo K C, Guibas L J. PointNet: deep learning on point sets for 3D classification and segmentation.

In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA, IEEE, 2017, 652–660
DOI:10.1109/cvpr.2017.16

24 Qi C R, Yi L, Su H, Guibas L J. PointNet++: deep hierarchical feature learning on point sets in a metric space. 2017

25 Simonovsky M, Komodakis N. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA, IEEE, 2017, 3693–3702
DOI:10.1109/cvpr.2017.11

26 Wang Y, Sun Y B, Liu Z W, Sarma S E, Bronstein M M, Solomon J M. Dynamic graph CNN for learning on point clouds. ACM Transactions on Graphics, 2019, 38(5): 1–12
DOI:10.1145/3326362

27 Li Y, Bu R, Sun M, Wu W, Di X, Chen B. Pointcnn: Convolution on x-transformed points. In: Advances in neural information processing systems. 2018, 820–830

28 Klokov R, Lempitsky V. Escape from cells: deep kd-networks for the recognition of 3D point cloud models. In: 2017 IEEE International Conference on Computer Vision (ICCV). Venice, Italy, IEEE, 2017, 863–872
DOI:10.1109/iccv.2017.99

29 Riegler G, Ulusoy A O, Geiger A. OctNet: learning deep 3D representations at high resolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA, IEEE, 2017, 3577−3586
DOI:10.1109/cvpr.2017.701

30 Wu W X, Qi Z, Fuxin L. PointConv: deep convolutional networks on 3D point clouds. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA, IEEE, 2019, 9621–9630
DOI:10.1109/cvpr.2019.00985

31 Atzmon M, Maron H, Lipman Y. Point convolutional neural networks by extension operators. ACM Transactions on Graphics, 2018, 37(4): 1–12
DOI:10.1145/3197517.3201301

32 Li J X, Chen B M, Lee G H. SO-net: self-organizing network for point cloud analysis. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA, IEEE, 2018, 9397–9406
DOI:10.1109/cvpr.2018.00979

33 Qi C R, Liu W, Wu C X, Su H, Guibas L J. Frustum PointNets for 3D object detection from RGB-D data. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA, IEEE, 2018, 918–927
DOI:10.1109/cvpr.2018.00102

34 Guo Y L, Wang H Y, Hu Q Y, Liu H, Liu L, Bennamoun M. Deep learning for 3D point clouds: a survey. 2019

35 Gelfand N, Mitra N J, Guibas L J, Pottmann H. Robust global registration. In: Symposium on geometry processing. 2005

36 Guo Y L, Bennamoun M, Sohel F, Lu M, Wan J W. 3D object recognition in cluttered scenes with local surface features: a survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(11): 2270–2287
DOI:10.1109/tpami.2014.2316828

37 Makadia A, Patterson A, Daniilidis K. Fully automatic registration of 3D point clouds. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). New York, NY, USA, IEEE, 2006, 1297–1304
DOI:10.1109/cvpr.2006.122

38 Ovsjanikov M, Mérigot Q, Mémoli F, Guibas L. One point isometric matching with the heat kernel. Computer Graphics Forum, 2010, 29(5): 1555–1564
DOI:10.1111/j.1467-8659.2010.01764.x

39 Rusu R B, Blodow N, Beetz M. Fast point feature histograms (FPFH) for 3D registration. In: 2009 IEEE International Conference on Robotics and Automation. Kobe, Japan, IEEE, 2009, 3212–3217
DOI:10.1109/robot.2009.5152473

40 Greenspan M, Yurick M. Approximate k-d tree search for efficient ICP. In: Fourth International Conference on 3-D Digital Imaging and Modeling. Banff, Alta., Canada, IEEE, 2003, 442–448
DOI:10.1109/im.2003.1240280

41 Nüchter A, Wulf O, Lingemann K, Hertzberg J, Wagner B, Surmann H. 3D mapping with semantic knowledge. In: RoboCup 2005: Robot Soccer World Cup IX. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, 335–346

DOI:10.1007/11780519_30

42 Pandey G, Savarese S, McBride J R, Eustice R M. Visually bootstrapped generalized ICP. In: 2011 IEEE International Conference on Robotics and Automation. Shanghai, China, IEEE, 2011, 2660–2667
DOI:10.1109/icra.2011.5980322

43 Akca D. Matching of 3D surfaces and their intensities. ISPRS Journal of Photogrammetry and Remote Sensing, 2007, 62 (2): 112–121
DOI:10.1016/j.isprsjprs.2006.06.001

44 Park J, Zhou Q Y, Koltun V. Colored point cloud registration revisited. In: 2017 IEEE International Conference on Computer Vision (ICCV). Venice, Italy, IEEE, 2017, 143–152
DOI:10.1109/iccv.2017.25

45 Zhou Q Y, Miller S, Koltun V. Elastic fragments for dense scene reconstruction. In: 2013 IEEE International Conference on Computer Vision. Sydney, NSW, Australia, IEEE, 2013, 473–480
DOI:10.1109/iccv.2013.65

46 Pulli K. Multiview registration for large data sets. In: Second International Conference on 3-D Digital Imaging and Modeling (Cat. No.PR00062). Ottawa, Ontario, Canada, IEEE, 1999, 160–168
DOI:10.1109/im.1999.805346

47 Chen Y, Medioni G. Object modelling by registration of multiple range images. Image and Vision Computing, 1992, 10 (3): 145–155
DOI:10.1016/0262-8856(92)90066-c

48 Low K. Linear least-squares optimization for point-to-plane ICP surface registration. Chapel Hill, University of North Carolina, 2004(February): 2–4

49 Rusinkiewicz S. A symmetric objective function for ICP. ACM Transactions on Graphics, 2019, 38(4): 85
DOI:10.1145/3306346.3323037

50 Fitzgibbon A W. Robust registration of 2D and 3D point sets. Image and Vision Computing, 2003, 21(13/14): 1145–1153
DOI:10.1016/j.imavis.2003.09.004

51 Yang J L, Li H D, Campbell D, Jia Y D. Go-ICP: a globally optimal solution to 3D ICP point-set registration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(11): 2241–2254
DOI:10.1109/tpami.2015.2513405

52 Rosen D M, Carlone L, Bandeira A S, Leonard J J. SE-Sync: a certifiably correct algorithm for synchronization over the special Euclidean group. The International Journal of Robotics Research, 2019, 38(2/3): 95–125
DOI:10.1177/0278364918784361

53 Maron H, Dym N, Kezurer I, Kovalsky S, Lipman Y. Point registration via efficient convex relaxation. ACM Transactions on Graphics, 2016, 35(4): 1–12
DOI:10.1145/2897824.2925913

54 Izatt G, Dai H K, Tedrake R. Globally optimal object pose estimation in point clouds with mixed-integer programming. In: Springer Proceedings in Advanced Robotics. Cham: Springer International Publishing, 2019, 695–710
DOI:10.1007/978-3-030-28619-4_49

55 Li J H, Zhang C H. Iterative matching point. 2019

56 Baker S, Matthews I. Lucas-kanade 20 years on: a unifying framework. International Journal of Computer Vision, 2004, 56(3): 221–255
DOI:10.1023/b:visi.0000011205.11775.fd

57 Huang X S, Mei G F, Zhang J. Feature-metric registration: a fast semi-supervised approach for robust point cloud registration without correspondences. 2020

58 Sarode V, Li X Q, Goforth H, Aoki Y, Dhagat A, Srivatsan R A, Lucey S, Choset H. One framework to register them all: PointNet encoding for point cloud alignment. 2019

59 Aoki Y, Goforth H, Srivatsan R A, Lucey S. PointNetLK: robust & efficient point cloud registration using PointNet. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA, IEEE, 2019, 7156–7165

DOI:10.1109/cvpr.2019.00733

60 Groß J, Osep A, Leibe B. AlignNet-3D: fast point cloud registration of partially observed objects. 2019

61 Zeng A, Song S R, Nießner M, Fisher M, Xiao J X, Funkhouser T. 3DMatch: learning local geometric descriptors from RGB-D reconstructions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA, IEEE, 2017, 199–208
DOI:10.1109/cvpr.2017.29

62 Gojcic Z, Zhou C F, Wegner J D, Wieser A. The perfect match: 3D point cloud matching with smoothed densities. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA, IEEE, 2019: 5540–5549
DOI:10.1109/cvpr.2019.00569

63 Zhu A F, Yang J Q, Zhao W Y, Cao Z G. LRF-net: learning local reference frames for 3D local shape description and matching. 2020

64 Lu W X, Zhou Y, Wan G W, Hou S H, Song S Y. L3-net: towards learning based LiDAR localization for autonomous driving. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA, IEEE, 2019, 6382–6391
DOI:10.1109/cvpr.2019.00655

65 Deng H W, Birdal T, Ilic S. PPFNet: global context aware local features for robust 3D point matching. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA, IEEE, 2018, 195–205
DOI:10.1109/cvpr.2018.00028

66 Yew Z J, Lee G H. 3DFeat-net: weakly supervised local 3D features for point cloud registration. In: Computer Vision-ECCV 2018. Cham: Springer International Publishing, 2018, 630–646
DOI:10.1007/978-3-030-01267-0_37

67 Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Surface reconstruction from unorganized points. In: Proceedings of the 19th annual conference on Computer graphics and interactive techniques-SIGGRAPH'92. New York, ACM Press, 1992
DOI:10.1145/133994.134011

68 Birdal T, Ilic S. Point pair features based object detection and pose estimation revisited. In: 2015 International Conference on 3D Vision. Lyon, France, IEEE, 2015, 527–535
DOI:10.1109/3dv.2015.65

69 Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł, Polosukhin I. Attention is all you need. Advances in neural information processing systems. 2017, 5998–6008

70 Devlin J, Chang M W, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. 2018

71 Santoro A, Raposo D, Barrett D G T, Malinowski M, Pascanu R, Battaglia P, Lillicrap T. A simple neural network module for relational reasoning. 2017

72 Wang X L, Girshick R, Gupta A, He K M. Non-local neural networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA, IEEE, 2018, 7794–7803
DOI:10.1109/cvpr.2018.00813

73 Choy C, Park J, Koltun V. Fully convolutional geometric features. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, Korea (South), IEEE, 2019, 8957–8965
DOI:10.1109/iccv.2019.00905

74 Sun W W, Wang R S. Fully convolutional networks for semantic segmentation of very high resolution remotely sensed images combined with DSM. IEEE Geoscience and Remote Sensing Letters, 2018, 15(3): 474–478
DOI:10.1109/lgrs.2018.2795531

75 Choy C, Gwak J, Savarese S. 4D spatio-temporal ConvNets: minkowski convolutional neural networks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA, IEEE, 2019, 3070–3079
DOI:10.1109/cvpr.2019.00319

76 Dai A, Chang A X, Savva M, Halber M, Funkhouser T, Nießner M. ScanNet: richly-annotated 3D reconstructions of

indoor scenes. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA, IEEE, 2017, 2432–2443
DOI:10.1109/cvpr.2017.261

77 Graham B. Sparse 3D convolutional neural networks. In: Procedings of the British Machine Vision Conference 2015. Swansea, British Machine Vision Association, 2015
DOI:10.5244/c.29.150

78 Rethage D, Wald J, Sturm J, Navab N, Tombari F. Fully-convolutional point networks for large-scale point clouds. In: Computer Vision–ECCV 2018. Cham: Springer International Publishing, 2018, 625–640
DOI:10.1007/978-3-030-01225-0_37

79 Jiang M Y, Wu Y R, Zhao T Q, Zhao Z L, Lu C W. PointSIFT: a SIFT-like network module for 3D point cloud semantic segmentation. 2018

80 Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. 1987, 726–740
DOI:10.1016/b978-0-08-051581-6.50070-2

81 Brachmann E, Krull A, Nowozin S, Shotton J, Michel F, Gumhold S, Rother C. DSAC: differentiable RANSAC for camera localization. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA, IEEE, 2017, 2492–2500
DOI:10.1109/cvpr.2017.267

82 Pais G D, Ramalingam S, Govindu V M, Nascimento J C, Chellappa R, Miraldo P. 3DRegNet: a deep neural network for 3D point registration. 2019

83 Yi K M, Trulls E, Ono Y, Lepetit V, Salzmann M, Fua P. Learning to find good correspondences. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA, IEEE, 2018, 2666–2674
DOI:10.1109/cvpr.2018.00282

84 He K M, Zhang X Y, Ren S Q, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA, IEEE, 2016, 770–778
DOI:10.1109/cvpr.2016.90

85 Christopher C, Wei D, Vladlen K. Deep Global Registration. 2020

86 Probst T, Paudel D P, Chhatkuli A, van Gool L. Unsupervised learning of consensus maximization for 3D vision problems. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA, IEEE, 2019, 929–938
DOI:10.1109/cvpr.2019.00102

87 Hu Y, Zhang D B, Ye J P, Li X L, He X F. Fast and accurate matrix completion via truncated nuclear norm regularization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(9): 2117–2130
DOI:10.1109/tpami.2012.271

88 Brachmann E, Rother C. Learning less is more-6D camera localization via 3D surface regression. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA, IEEE, 2018, 4654–4662
DOI:10.1109/cvpr.2018.00489

89 Deng H W, Birdal T, Ilic S. 3D local features for direct pairwise registration. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA, IEEE, 2019, 3239–3248
DOI:10.1109/cvpr.2019.00336

90 Yang Y Q, Feng C, Shen Y R, Tian D. FoldingNet: point cloud auto-encoder via deep grid deformation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA, IEEE, 2018, 206–215
DOI:10.1109/cvpr.2018.00029

91 Deng H W, Birdal T, Ilic S. PPF-FoldNet: unsupervised learning of rotation invariant 3D local descriptors. In: Computer Vision–ECCV 2018. Cham: Springer International Publishing, 2018, 620–638
DOI:10.1007/978-3-030-01228-1_37

92 Wang Y, Solomon J M. PRNet: self-supervised learning for partial-to-partial registration. 2019

93 Gojcic Z, Zhou C F, Wegner J D, Guibas L J, Birdal T. Learning multiview 3D point cloud registration. 2020

94 Sekikawa Y, Suzuki T. Tabulated MLP for fast point feature embedding. 2019

95 Jang E, Gu S X, Poole B. Categorical reparameterization with Gumbel-Softmax. In: Proceedings of the International Conference on Learning Representations. Toulon, France, 2017

96 Savva M, Yu F, Su H, Aono M, Chen B, Cohen-Or D, Deng W, Su H, Bai S, Bai X. Shrec16 track: largescale 3D shape retrieval from shapenet core55. EG 2016 workshop on 3D Object Recognition. 2016
DOI: 10.2312/3dor.20161092

97 Wu Z R, Song S R, Khosla A, Yu F, Zhang L G, Tang X O, Xiao J X. 3D ShapeNets: a deep representation for volumetric shapes. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, MA, USA, IEEE, 2015, 1912–1920
DOI:10.1109/cvpr.2015.7298801

98 Choi S, Zhou Q Y, Miller S, Koltun V. A large dataset of object scans. 2016

99 Saxena A, Chung S H, Ng A Y. 3-D depth reconstruction from a single still image. International Journal of Computer Vision, 2008, 76(1): 53–69
DOI:10.1007/s11263-007-0071-y

100 Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. Providence, RI, USA, IEEE, 2012, 3354–3361
DOI:10.1109/cvpr.2012.6248074

101 Maier R, Kim K, Cremers D, Kautz J, Nießner M. Intrinsic3D: high-quality 3D reconstruction by joint appearance and geometry optimization with spatially-varying lighting. In: 2017 IEEE International Conference on Computer Vision (ICCV). Venice, Italy, IEEE, 2017, 3133–3141
DOI:10.1109/iccv.2017.338

102 Turk G, Levoy M. The Stanford 3D scanning repository. Stanford University Computer Graphics Laboratory. 2005

103 Maddern W, Pascoe G, Linegar C, Newman P. 1 year, 1000 km: The Oxford RobotCar dataset. The International Journal of Robotics Research, 2017, 36(1): 3–15
DOI:10.1177/0278364916679498

104 Pomerleau F, Liu M, Colas F, Siegwart R. Challenging data sets for point cloud registration algorithms. The International Journal of Robotics Research, 2012, 31(14): 1705–1711
DOI:10.1177/0278364912458814

105 Chang A X, Dai A, Funkhouser T. Matterport3D: learning from RGB-D data in indoor environments. In: Proceedings of the International Conference on 3D Vision. Qingdao, China, IEEE, 2017
DOI: 10.1109/3DV.2017.00081

106 Xiao J X, Owens A, Torralba A. SUN3D: a database of big spaces reconstructed using SfM and object labels. In: 2013 IEEE International Conference on Computer Vision. Sydney, NSW, Australia, IEEE, 2013, 1625–1632
DOI:10.1109/iccv.2013.458

107 Uy M A, Pham Q H, Hua B S, Nguyen T, Yeung S K. Revisiting point cloud classification: a new benchmark dataset and classification model on real-world data. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, Korea (South), IEEE, 2019, 1588–1597
DOI:10.1109/iccv.2019.00167

108 Hua B S, Pham Q H, Nguyen D T, Tran M K, Yu L F, Yeung S K. SceneNN: a scene meshes dataset with aNNotations. In: 2016 Fourth International Conference on 3D Vision (3DV). Stanford, CA, USA, IEEE, 2016, 92–101
DOI:10.1109/3dv.2016.18

109 Shilane P, Min P, Kazhdan M. The princeton shape benchmark. In: Proceedings of the International Conference on Shape Modeling Applications. IEEE, 2004
DOI: 10.1109/SMI.2004.1314504

110 Rusu R B, Marton Z C, Blodow N, Beetz M. Persistent point feature histograms for 3D point clouds. In: Proceeding of the Int Conf Intel Autonomous Syst (IAS-10). Baden-Baden, Germany, 2008, 119–128
DOI: 10.3233/978-1-58603-887-8-119