



*Mini review*

## Point cloud registration: a mini-review of current state, challenging issues and future directions

Nathan Brightman, Lei Fan\* and Yang Zhao

Department of Civil Engineering, Xi'an Jiaotong-Liverpool University, 111 Renai Road, Suzhou 215123, China

\* **Correspondence:** Email: [Lei.Fan@xjtlu.edu.cn](mailto:Lei.Fan@xjtlu.edu.cn); Tel: +86-(0)512-81883231.

**Abstract:** A point cloud is a set of data points in space. Point cloud registration is the process of aligning two or more 3D point clouds collected from different locations of the same scene. Registration enables point cloud data to be transformed into a common coordinate system, forming an integrated dataset representing the scene surveyed. In addition to those reliant on targets being placed in the scene before data capture, there are various registration methods available that are based on using only the point cloud data captured. Until recently, cloud-to-cloud registration methods have generally been centered upon the use of a coarse-to-fine optimization strategy. The challenges and limitations inherent in this process have shaped the development of point cloud registration and the associated software tools over the past three decades. Based on the success of deep learning methods applied to imagery data, attempts at applying these approaches to point cloud datasets have received much attention. This study reviews and comments on more recent developments in point cloud registration without using any targets and explores remaining issues, based on which recommendations on potential future studies in this topic are made.

**Keywords:** point cloud; registration; deep learning; LiDAR; benchmark datasets

---

### 1. Introduction

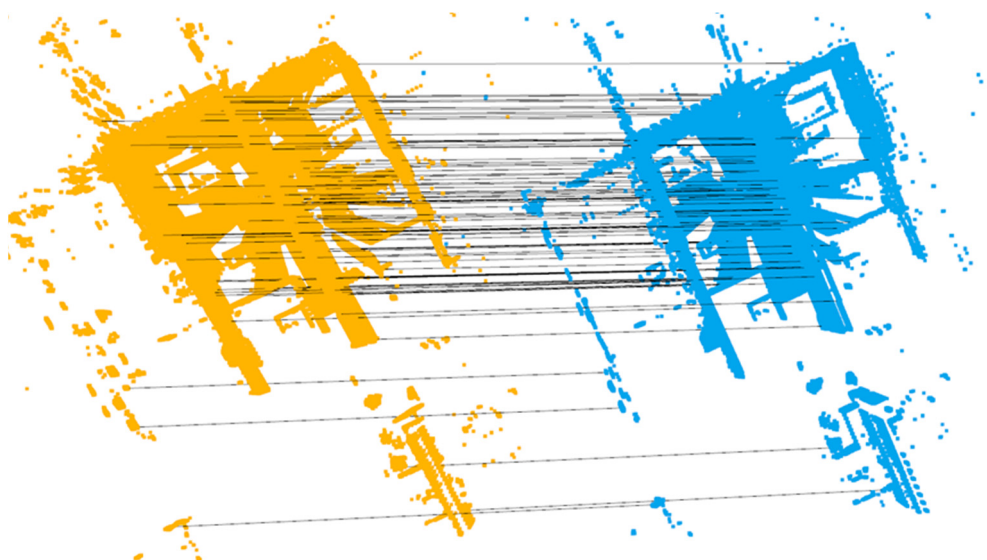
Point cloud data constitute a very basic form of data, i.e., a set of points representing a group of objects and the space between them. As such, this form finds utility in a broad range of applications at vastly different scales, from the very large, such as those in geographic survey in geosciences, through

to the very small, such as those in microbiology or particle physics. Point cloud registration is a basic step in many point cloud processing pipelines. It is the process of aligning two or more 3D point clouds collected from different locations of the same scene. Registration enables point cloud data to be transformed into a common coordinate system, forming an integrated dataset representing the scene surveyed. There are various registration methods available, such as those reliant on targets being placed in the scene before data capture, and others based on using only the data captured [1].

Registration plays an indispensable role in the applications of point clouds. For example, multiple point clouds acquired at different locations must be registered to form a complete documentation of a cultural heritage site or a structure. Therefore, the registration quality significantly affects the overall quality of the documentation and determines the success of the application. There are also applications that aim for change detections or deformation estimations by comparing point clouds obtained at multiple epochs. Typical applications of this type include the estimation of mining excavation volumes, deformation monitoring of heritage buildings and civil structures and quantification of the erosion and deposit of terrain surfaces. In these applications, the accuracy of the registration often dictates the minimum level of detection of changes or deformations, given that the terrestrial laser scanning can usually produce high-quality measurements. In large-scale projects, such as a significant construction project, there are possibly billions of data points spread across hundreds of scans, which imposes great challenges to many registration methods (especially deep learning-based methods) to register such large datasets without using any scan targets. For the navigation of robots and autonomous driving vehicles, the real-time efficiency of registration is a key consideration, in addition to registration accuracy.

Based on the aforementioned applications, the motivation behind point cloud registration may be broadly split into two categories: the desire to build models based on multiple point clouds [2], or the desire to know the relative position or the pose of one point cloud with respect to another [3]. These different motivations place different emphasis on the registration process that is, for example, either toward achieving high precision or high speed. Different applications and data acquisition methods influence the importance of key factors in the registration process, such as the degree of overlap between point clouds, the type of transformation (e.g., rigid or non-rigid) needed to complete the registration and the levels of error and noise present in the data.

Registration between source and target point clouds is commonly a two-step process: (1) establishing 3D-3D point correspondences between the source and target, and (2) finding the optimal transformation between the source and the target. Considering rigid transformations (a combination of rotation and translation), the optimal transformation is usually considered to be one that minimizes the total Euclidean distance between all point correspondences. Figure 1 shows an example of the cloud-to-cloud registration based on the use of feature correspondences established by a deep learning technique. For better visualization of the point feature matches, the two point clouds in Figure 1 have been intentionally set to be parallel to each other.



**Figure 1.** Point feature correspondences for the point cloud registration.

Until recently, point cloud registration methods have generally been centered upon the use of a coarse-to-fine optimization strategy, the best-known element of which is the iterative closest point (ICP). The challenges and limitations inherent in this process have shaped the development of point cloud registration and the associated software tools over the past three decades. Based on recent success for deep learning methods applied to 2D image data, attempts at applying these approaches to 3D datasets have received much attention. The fusion of more recent deep learning methods and conventional optimization approaches is the source of much research and progress. We review the state of the art for both approaches and highlight various remaining issues related to this subject. In terms of its scope, this study briefly reviews and critically discusses the key developments of approaches for point cloud registration without using any targets, but it was not intended to consider all relevant studies in the literature.

## 2. Point cloud data

### 2.1. File formats

There is a wide range of file formats, including ASCII (XYZ, OBJ, PTX and ASC), binary (FLS, PCD and LAS) or those storing data in both binary and ASCII (e.g., PLY, FBX and E57). Which specific one is used will depend on both the source of the data and the tools used when handling them. While the underlying X-Y-Z format is very much standardized, other data are often associated with both the individual points and the point cloud as a whole. The impact that the specific format will have is in terms of the data available to guide any processing: additional characteristic information available for each point (e.g., color, intensity), and/or meta data available to, for example, provide class, time or location parameters. Apart from the likely benefits of those additional data, they (i.e., characteristic information and/or meta data) also present a complicating factor if, for example, point clouds derived from more than one data format are to be used with each other.

## 2.2. Benchmark datasets

In any development work aimed at improving point cloud data processes, there is a need to have access to substantial amounts of point cloud data. When datasets have been specifically collected for a particular set of experiments or real-world applications, they are sometimes shared with the rest of the community. For example, if any comparison is to be made between tools and techniques, access to such common datasets is very valuable. When used in this way, these datasets are often referred to as “benchmarks”. A number of valuable attempts have been made to provide publicly shared evaluation benchmarks. Typically, these cover a specific-use case of point clouds, and they have been collected by using a specific sensor. Preparing data for point cloud related experiments is a substantial task. Some of the benchmark datasets acknowledge this by providing some welcome assistance, perhaps either in the form of documentation or specific software to aid with data preparation. There is a range of available datasets, some of which are more applicable than others to particular problem types, such as cloud-to-cloud registration.

Since deep learning requires large volumes of data to facilitate training of the learning models, simulated data are also of particular value. In general terms, the objective is to closely simulate the characteristics of data collected by using real sensor devices. Some key attributes of simulated point clouds to be considered include the level of random errors, the interaction with reflective surfaces (e.g., glass, water), variable atmospheric conditions (e.g., rain, fog) and point densities. The broad aim is to facilitate an analysis of the performance of registration methods based on the use of simulated point cloud datasets. Therefore, for completeness, corresponding real-world point cloud data should then be used to evaluate the applicability of the models trained on the simulated datasets.

Several sources of point cloud data that may be considered include real-world single-sensor data (e.g., 3DMatch [4], KITTI [5], ETH [6]), real-world multi-sensor data (e.g., 3DCSR [7]) and synthetic (often simulated) point clouds. The synthetic point clouds may be derived from models such as ModelNet40 [8] or simulated sensors such as BLINDER [9], SynthCity [10], CARLA simulator [11] and the LG SVL Simulator [12]. A summary of the information of some representative benchmark datasets is shown in Table 1.

### 2.2.1. 3DMatch

3DMatch is one of the most popular real-life datasets used for developing deep learning-based point cloud registration methods. It contains over 8 million correspondences out of 200 thousand registered RGB-D images from 62 RGB-D real-life indoor scene reconstructions, such as 7-Scenes, SUN3D, RGB-D Scenes v.2 and Halber. The datasets are captured in different environments with different local geometries at varying scales. Subsets or variants (e.g., Rotated-3DMatch, 3DLoMatch) of the original 3DMatch were also derived in other studies for more specific requirements.

### 2.2.2. KITTI

KITTI includes a range of point clouds covering a total length of 39.2 km, which were acquired by a Velodyne laser scanner mounted on a car traveling in rural areas and along highways. The ground-truth registration was given by the global positioning system (GPS) and inertial measurement unit (IMU) measurements synchronized with the scanner. It is an odometry-derived dataset that was

initially designed for stereo matching performance evaluation, and it contains stereo sequences, LiDAR (light detection and ranging) point clouds and ground-truth poses. It consists of 22 stereo sequences, where 11 sequences (i.e., 00–10) have ground-truth trajectories for training and 11 sequences (i.e., 11–21) have no ground truth for evaluation.

### 2.2.3. ETH

ETH contains point cloud data representing both indoor and outdoor scenes, which were collected with laser, IMU and GPS sensors. It consists of 276 LiDAR scans in eight scenes, including two indoor, five outdoor and one mixed environment scenes. The point clouds were captured by using a Hokuyo UTM-30LX. A theodolite was utilized to guarantee the precision of the ground-truth positions. It was often used as a test dataset due to its limited number of scans.

### 2.2.4. Multi-sensor—3DCSR

Multi-sensor datasets are less common. They are relevant because different types of sensors have different imaging mechanisms and sensor noises. One of the datasets of this type is 3DCSR, where two kinds of mixed-source point cloud data were fused: Kinect and Lidar, and Kinect and 3D reconstruction. LiDAR, Kinect and camera sensors were used to collect those data. They were captured by using a Kinect and RGB camera for two scenes and using LiDAR and Kinect sensors for 19 scenes, containing in total 202 pairs of point clouds. They were manually aligned to obtain the ground-truth transformation. The dataset contains the most common objects or scenes in an indoor workspace environment.

### 2.2.5. ModelNet40

As a subset of ModelNet, ModelNet40 consists of 12,311 3D CAD models (with vertices, edges and faces) for objects in 40 categories. Points can be sampled according to certain criteria across the exterior surfaces of the CAD models of ModelNet40 to generate point clouds. During data preparation, it is possible to add random rotation and translation transforms, add random noises and remove some points to simulate partial data. It has been popular in the training and testing of many deep-learning registration models.

### 2.2.6. Simulated sensors

BLAINDER is an add-on package for the open-source 3D modeling software Blender, which enables a largely automated generation of semantically annotated point cloud data in a virtual 3D environment. It supports several depth-sensing techniques, including LiDAR. The add-on is able to load different depth sensors from presets, implement customized sensors and simulate different environmental conditions (e.g., rain, dust). The semantically labeled data can be exported to various 2D and 3D formats and optimized for different applications.

SynthCity is a synthetic full-color mobile laser scanning point cloud, and it contains 367.9 million points. Every point is assigned a label from one of nine categories. The point cloud was generated from

a model of a typical urban/suburban environment by using a plugin for Blender. SynthCity used Blender to simulate sensor data, which is an older variant of BLAINDER based on the same principles.

There are several other research efforts for synthetic data, primarily aiming at the autonomous driving domain; examples include CARLA (an open-source simulator for autonomous driving research) [11] and the LG SVL Simulator (an autonomous vehicle simulator) [12].

**Table 1.** Summary of some representative benchmark datasets for registration.

Dataset	#Scenes	#Frames or scans	Format	Scenario
3Dmatch [4]	62	200,000	RGB-D	indoor
KITTI [5]	39.2 km	-	Point cloud	outdoor
ETH [6]	8	276	Point cloud	indoor and outdoor
3DCSR [7]	21	202	RGB-D and point cloud	indoor
ModelNet [8]	151,128	-	CAD	synthetic objects
ModelNet40 [8]	12,311	-	CAD	synthetic objects
BLAINDER [9]	-	-	Synthetic point cloud	semantically annotated synthetic objects
SynthCity [10]	-	-	Synthetic point cloud	semantically annotated synthetic urban/suburban environment

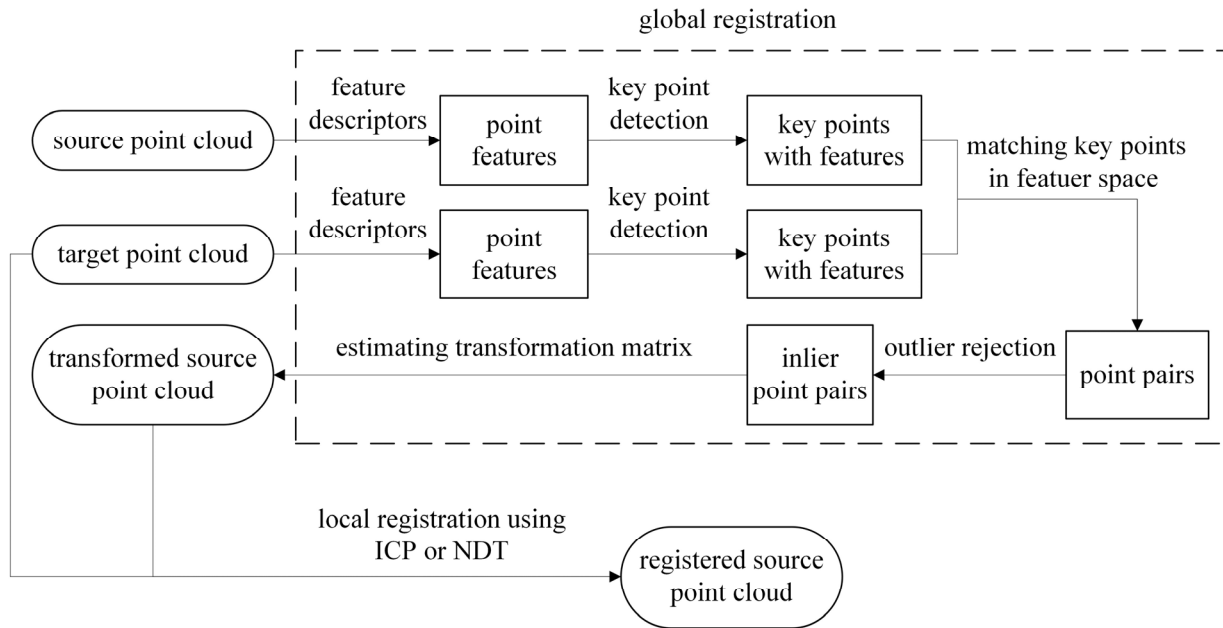
### 3. Registration methods

#### 3.1. Typical registration strategy

Most point cloud registration methods employ a coarse-to-fine strategy. In this approach, a coarse registration (typically a global registration) is first applied to find an approximate rigid transformation (a combination of rotation and translation) for a pair of point clouds. Once a coarse transformation is available, a fine registration (typically a local registration) algorithm, such as the ICP algorithm, normal distribution transform (NDT) algorithm or one of the more efficient variants of ICP and NDT is used to refine the transformation.

A global registration problem is one in which the aim is to align point clouds without additional information on their relative poses. Most global algorithms do not lend themselves to providing precise results. In contrast, local registration algorithms perform better in this respect. As might be expected, algorithms focused on local registration are usually less effective for global problems. One issue is that they make use of local optimization techniques that may get stuck within local minima when used against specific datasets.

As a result, many registration pipelines use a global algorithm to provide an initial estimate that sets up a subsequent local process. Global registration is typically achieved by using point features. Such a feature-based registration is often a slow process, as the extraction of features can be computationally expensive. In contrast, local registration approaches do not usually employ any feature extractions. A typical workflow of the whole registration process is shown in Figure 2.



**Figure 2.** Typical workflow of the point cloud registration.

The key factors that may be considered for developing registration methods include different sensor types for point clouds where different noise patterns may exist, the degree of spatial overlaps, the amount of misalignments, the combination of rotation and translation errors, global or local registration, the type of scene (indoor structured, outdoor structured or outdoor unstructured), whether there are moving objects, the scale of problems, etc.

### 3.2. Local registration

#### 3.2.1. ICP and variants

It is widely accepted that the ICP algorithm [13] is the most widely adopted method for pairwise fine registration. It remains the de facto technique for local registration. Generally, ICP assumes that the point clouds are roughly aligned and aims to calculate the rigid transformation that achieves the alignment. Rather than comparing features, ICP approximates potential correspondences by looking for the closest point in the target point cloud to each point in the source point cloud, which is often an expensive computation process. A large number of variants of ICP have been developed, typically focusing on the speed or quality of the results.

Requiring a good initial transformation to bring it close to registration, ICP converges to a more optimal registration by repeatedly applying a search for point-to-point correspondences, followed by a transformation calculation. LiDAR point clouds are often huge and corrupted by variations in the point densities, noise, outliers (unintended points), occlusions (missing points) and partial overlaps. ICP is challenged by such LiDAR point clouds due to the limited one-to-one correspondences between two point clouds. A significant body of research has been applied to variants of ICP, aiming at dealing with these challenges. Representative ones include point-to-plane [13], point-to-projection [14] and plane-to-plane [15]. Chetverikov et al. [16] proposed the trimmed ICP (TrICP) algorithm. At each

iteration step, TrICP considers the outliers, shape defects and partial overlaps, making it more tolerant of incomplete and noisy data. Yang et al. [17] introduced the global optimal ICP method (Go-ICP) to integrate ICP with a branch-and-bound scheme so that a coarse registration is not needed. However, Go-ICP is much more time-consuming than ICP, and it is sensitive to outliers. Another focus on improving ICP is increasing the efficiency of the correspondence search by, for example, applying a GPU-accelerated way to deal with K-D tree structures [18].

### 3.2.2. NDT and variants

There are also local registration algorithms that do not employ such a nearest-point approximation. NDT [19], for example, treats the point clouds as a set of Gaussians, trying to align them by finding the most probable alignment. The NDT algorithm was initially developed for 2D robotics, eventually advancing to 3D [20]. NDT handles the registration process as one of matching probability density functions (PDFs). The registration problem is transformed into a nonlinear optimization problem in which the optimal transformation is based on maximizing the similarity between the PDFs. Whereas, ICP and its variants require a relatively high point density (to obtain accurate correspondences), and NDT is better able to tolerate lower/variable density in point clouds. Similar to ICP, NDT has been refined and continues to be incorporated into new research and developments, such as the work by Zhou et al. [21].

## 3.3. Global registration

### 3.3.1. Classic, non-learning-based

In conventional global registration approaches, features are extracted by using manually defined rules to form the handcrafted feature descriptors. For those based on handcrafted features, a deeper review can be found in the paper by Han et al. [22]. Fast point feature histograms (FPFHs) [23] appear to be the basis of a lot of research, and they have been used in various works claiming state-of-the-art results. It is observed that the FPFH descriptor has been utilized in a number of places, and it continues to provide state-of-the-art results with the TEASER [24,25]. One related approach is fast global registration (FGR) [26], which is commonly used as a benchmark for being a state-of-the-art global registration method. Four-points congruent set (4PCS)-based registration also continues to drive a number of research efforts, and it is the basis of the current state-of-the-art results achieved by K-4PCS [27].

Probabilistic registration methods model the distribution of point clouds as a density function. One key method that adopts the probability density estimation is the coherent point drift (CPD) method. CPD-based methods use a Gaussian mixture model (GMM) to describe a point cloud and then fit the GMM to a second point cloud by maximizing the likelihood of the objective function. CPD sees applications for non-rigid transformations in, for example, medical applications [28]. CPD provides generality, accuracy and good robustness against noise and outliers. It continues to be improved by a number of works, including that conducted by Golyanik et al. [29] and Wang et al. [30].

Using handcrafted features to distinguish correspondences is highly dependent on the experience of their designers [31]. As a result, their generalizability and robustness may be sub-optimal for many



applications. To improve these characteristics, efforts have been made to develop or apply deep learning algorithms to point cloud registration.

### 3.3.2. Deep learning-based

In most cases [32–40], deep learning was considered for some tasks (e.g., feature extraction and key point selection) in a registration pipeline, while the other tasks were completed by using conventional approaches. Based on the existing research, feature extraction is the task wherein deep learning is most widely implemented. This is not only because it is a very fundamental task in point cloud registration, but also because the quality of the features extracted often determines the overall performance of a registration method. Deep learning-based feature extraction algorithms automatically learn more robust feature representations, and they have great potential in registering scenes that have repetitive and symmetrical features and limited overlaps. Although deep learning has also been adopted for key point selection in many studies, there are also methods [40–42] that do not adopt a learned key point detector but were still able to achieve the state-of-the-art performances on the benchmark datasets, suggesting that using a learning-based key point detector is not always essential. There are also methods that embed the whole registration process in a deep learning network [43–47].

Based on the taxonomy described by Dong et al. [31], deep learning-based registration methods can be divided into three categories according to their data representations: voxels, multiviews and points. The voxelization-based and multiview-based registration methods have been the subject of a number of research efforts, but, due to issues around computational inefficiency, they have been largely restricted to good results with small-scale indoor datasets. A proportion of the results from these efforts have been utilized in developments that were focused on point-based representations.

Applying deep learning to 3D point cloud data introduces a number of challenges. Some of these challenges include the general point cloud data characteristics such as occlusions, noise and outliers. However, more specific issues with the application of deep learning on point clouds are (1) irregularity (i.e., the points are not evenly distributed spatially across the different regions of the scene so that some regions contain dense points and others are sparse), (2) unstructured (i.e., not organized in a known pattern as would be the case for image data) and (3) unordered (i.e., the point cloud of a scene is a set of points, usually stored as a list in a file, and a change in the order does not reflect a change in the scene represented).

These issues make the direct application of convolutional neural networks (CNNs) difficult, as they assume ordered and regular structures. Early approaches attempted to overcome these issues by converting a point cloud into a structured grid format. The establishment of PointNet [48] represents a key advancement in point cloud processing. PointNet [48] and PointNet++ [49] are the pioneering methods for directly processing unordered point sets (invariance to transformations), making them the foundation for many of the recent developments where they are used as a feature extractor. For example, PPFNet [34] is based on and extends PointNet to provide some learning of the local geometry.

PointNetLK [46] introduces the Lucas-Kanade (LK) algorithm into 3D point cloud registration and solves the problem iteratively with PointNet. PointNetLK represents a significant milestone in the development of a category of deep learning methods that do not directly seek to identify correspondences across the input point cloud data before proceeding. Being built upon PointNet, PointNetLK uses its learnable structured representation for point clouds and applies it to the task of point cloud registration. To achieve this, it utilizes a classical stereo vision technique, i.e., the LK

algorithm [50]. This connection was motivated by Wang et al. [51], who demonstrated 2D object tracking performance by treating the LK algorithm as a recurrent neural network, effectively extending a successful approach from 2D to 3D. It has provided an important stepping stone for some promising developments [52,53].

PCRNet [45], in a similar way to PointNetLK, utilizes PointNet to extract global features. In contrast to PointNetLK, for the feature alignment module, a data-driven technique is used. Two global features are concatenated before five fully connected layers are applied, which occurs before an output layer provides the registration transformation. Compared to PointNetLK, PCRNet exhibits better generalizability, but it is not robust to noise.

Like PointNet, although not specifically for registration, dynamic graph CNNs (DGCNNs) [54] are used as a component in a number of related registration pipelines, and they include deep closest point (DCP) [43] and PRNet [44]. In DGCNNs, a graph is constructed in the feature space and dynamically updated after each layer of the network. A multilayer perceptron (MLP) is used as the feature learning function for each edge. Channel-wise symmetric aggregation is applied to the edge features associated with the neighbors of each point.

DCP employs a DGCNN for feature extraction and a singular value decomposition (SVD) module [55] to calculate rotation and translation. Incorporating techniques from both computer vision and natural language processing, it is broadly based on the classic ICP pipeline while aiming to avoid the associated issue of converging to local solutions. As a limitation, there is an assumption of a high degree of correspondences between the point clouds. DCP has three stages. In the first stage, data are embedded (by a point cloud embedding network) into high-dimensional space by using DGCNN to extract features. It was claimed that this improved the feature effectiveness of the matching by making the features task-specific. In the second stage, an attention-based module [56] combined with a pointer generation layer [57] is used to approximate combinatorial matching, which provides a dependency term between the feature sets, i.e., one set is modified in a way that is based on the structure of the other. In the third stage, a differentiable SVD layer is used to extract the final rigid transformation. It has been shown that SVD provided better results than using an MLP.

PRNet uses the network architecture DCP iteratively. Use of DCP in this way was suggested by the developers of DCP as a possible extension of their work to better handle partial overlap scenarios. Registration between point clouds with only limited overlaps is a challenging case to handle. Under such cases, the end-to-end, correspondence-free methods such as PointNetLK can perform poorly. PRNet is aimed at overcoming this problem. In PRNet, a search for the key points is made by comparing the norms of the learned features, which is followed by the estimation of the correspondences iteratively in a coarse-to-fine manner.

RPM-Net [47] illustrates a common theme in deep learning approaches. It adopts and builds upon the classical method of robust point matching (RPM) [58]. RPM aims to avoid some of the issues with ICP by applying a soft assignment scheme combined with “deterministic annealing” to gradually “harden” the assignment. RPM-Net uses this soft assignment approach combined with a Sinkhorn layer [59]. Sinkhorn is a mechanism that finds utility in a number of recent deep learning works, including PRNet. RPM-Net is, in many ways, similar to DCP. The authors claimed that their use of Sinkhorn normalization enabled RPM-Net to better handle outliers and partial visibility. It also uses an iterative pipeline to achieve high precision, which is one of the ways that the DCP developers highlighted for further study. It also makes use of a FPFH-based [23] local feature descriptor that was referenced in many works, including the PPFNet [34], which, in turn, can be

utilized in PointNetLK. RPM-Net provides a useful comparison against DCP since it uses a similar testing methodology and datasets, as well as many of the same benchmark methods, including FGR and PointNetLK.

D3Feat [37] and Predator [38] uses kernel point convolution (KPCConv) [60] to compute local features. In a basic KPCConv layer, a neighborhood radius and multiple convolutional kernel points are defined for each point in a point cloud. The positions and the influence distances of kernel points are estimated based on the number and the distribution of the points within the neighborhood. The convolution of point features is conducted on the points within the neighborhood by using the convolutional kernel points to output a local feature. The KPCConv network connects multiple KPCConv layers to produce final point features. The output point features of the previous layer are used as the input point features of the current layer. Through these multiple connected basic KPCConv layers, incrementally, the original point cloud is subsampled with the pooling of point features while the neighborhood radius increases. Therefore, the receptive fields of point features are also incrementally increased. The KPCConv can reduce excessive computational cost caused by the sparsity of a point cloud by using point convolutional kernels instead of grid convolutional kernels. Meanwhile, it incorporates local spatial information and retains context awareness with a large receptive field.

REGTR [61] applies multiple transformer attention layers to directly predict each downsampled point's corresponding location in the point cloud being registered. The establishment of the correspondences does not require an additional RANSAC step, which is distinct from the typical correspondence-based algorithms. As such, it can efficiently produce an accurate registration, as evidenced by the tested performances based on benchmark datasets, such as ModelNet40. Its decent performances may also be attributed to context-awareness components incorporated into its point features.

Other more recent attempts of deep learning networks for point cloud registration include DDRNet [62], PCAM [63], HRegNet [64], WSDesc [65], OMNet [66] and RCP [67]. Although there has been a gradual increase of studies in this area in recent years (up to 2021), there is a clear sign of slowing interest in this subject, as evidenced by the reduced number of relevant publications in 2022. This is likely due to further improvements of the registration performance being found to be very small. For a more in-depth review of deep learning methods, see Huang et al. [7] and Zhang et al. [33]. So far, deep learning methods have proved effective for the registration of indoor and relatively small-scale outdoor point clouds [31]. However, limitations on the amount of data and complexities mean that scaling to large-scale outdoor point clouds is still a barrier and requires more effort in future research.

## 4. Challenging issues

### 4.1. Point cloud data perspective

There is a wide range of ways to store point clouds in files. In practice, there are far more file extensions than the fundamental differences between the file types. Some formats are versions tailored to proprietary file systems or optimized in one way or another for a particular software tool. These may bring inconveniences when data are shared. In some file types, additional information (other than coordinates) is also available. While such additional information may aid in data visualization and processing, they could also present a complicating factor if, for example, point clouds derived from more than one data format are to be used with each other.

As point cloud data are a standardized form of 3D representation, they find applications in a very wide range of applications with differing requirements and types of sensor-collecting data. Therefore, point cloud registration is not one clearly defined task. Factors such as the levels of overlap, degrees of occlusions and noise and the occurrence of rigid or non-rigid transformations can change the nature of the problem. When compounding these variabilities across the different disciplines, as we have noted, the motivation for registration is either high precision to accurately capture reality or high speed (with sufficient precision) to allow navigation. When attempting to compare the effectiveness of different processes and results from different sets of research, it is very difficult to accommodate the wide range of differences between application types (e.g., robotics, BIM, navigation, change detection), point cloud datasets and sensor type used (e.g., RGB-D, LiDAR).

#### *4.2. Registration method perspective*

Based on our study of recently developed methods, there is not any substantial evidence that there are significant improvements in the effectiveness (i.e., precision) of registration being achieved. There is still a substantial reliance on well-established methods (e.g., ICP and variants, NDT and variants) for precision registration. It is often the case that these classic techniques are utilized to refine the registration efforts of other methods. However, there are clear signs that efficiency (i.e., speed) has benefited from recent developments. This possibly reflects where most of the emphasis in the research is being placed, i.e., it has been broadly directed toward simultaneous localization and mapping (SLAM) to assist with the autonomous vehicle research efforts.

It is noted that the significant efforts have been directed at applying deep learning techniques to handle point cloud data, including registration. There are clear benefits of employing deep learning-based feature extractors to avoid the need to invest in the development of their hand-crafted counterparts. These can be utilized now to give immediate benefits in terms of identifying features to facilitate global registration. One example of this is the use of deep learning modules like DGCNN and PPFNet feature extractors. These are shown to be more effective than the non-learning-based feature extractors such as FPFHs. However, it is also clear that deep learning techniques are still some way from being able to handle large-scale projects where there are possibly billions of data points spread across hundreds of scans, as might be faced in a significant construction project, for example. The current state of the art is effectively limited to dealing with small- to medium-scale datasets.

A note of caution is needed, as it remains to be demonstrated that the deep learning approaches can generalize sufficiently and do not fall into the trap of overfitting to their training data. This represents a situation in which a deep learning model actually models the training data too well and cannot be used outside of that dataset effectively.

#### *4.3. Pre-trained models for real-life applications*

There is a very limited number of pre-trained deep learning models for various typical real-life scenes, partially due to the lack of benchmark datasets. Pre-trained models are a useful way to distribute the result of training a deep learning model. Given the length of time (many hours or days) that the training process of some models takes, it is highly desirable to store the model parameters derived from the training in files to enable re-loading of the model at a later date. For example, if a developer of a specific deep learning model provides a relevant pre-trained model, which matches the

users' experiment requirements, a significant amount of time can be saved if the model is to be used as a benchmark for future work. Providing both the codes and pre-trained models helps to make the work a very good platform to study and a useful benchmark.

#### *4.4. Performance evaluation*

Based on our literature study, the typical performance metrics used to evaluate registration methods include the feature match recall, registration recall, success rate, rotation error and translational error. However, it is rare in the existing studies to consider all of these metrics. In most cases, researchers often chose one or some of these performance metric(s) to test on some of the commonly used benchmark datasets. The choices of performance metrics and benchmark datasets are probably favorable for their methods. As such, the comparisons between different methods may not be consistent or fair.

The performances of many target-less registration methods are often evaluated by using benchmark datasets where the ground-truth registrations are known. However, in real-life applications, the ground truth is often unknown. As such, it is challenging to quantitatively assess the quality of registration in those cases.

#### *4.5. Multi-temporal data*

Registration methods are typically tested on the point cloud data obtained via a single survey campaign, and rarely on multi-temporal point clouds. Transforming multi-temporal data into a common coordinate system is very important for many applications, such as deformation measurements and change detection. The registration of multi-temporal point clouds is often more challenging because scenes may have changed over time. As such, how to determine the static areas/objects and/or eliminate the impacts of moving/changing objects in a scene is an important consideration for registration.

### **5. Future research**

Based on our study, the following future research opportunities are recommended.

- The application of deep learning methods to large-scale point cloud datasets in a systematic and reproducible way, which will require access to large datasets that are appropriate for a specific application area.
- Further development of the tools to facilitate the generation of simulated sensor data based on realistic environments (e.g., works such as BLINDER and SynthCity show the way forward on this).
- The establishment of competent pre-trained deep learning models for typical real-life scenes of a particular application would be very useful for the future efficient utility of the models.
- Based on more synthetic dataset generators, the application of generative adversarial network techniques might yield benefits in terms of being able to deliver more realistic datasets, much like some of the remarkable results being seen with synthetic 2D imagery.
- A greater emphasis on the inclusion of meta data (e.g., class, time and location) and additional characteristic information (e.g., color, intensity and multispectral bands) in the development

of registration processes. This will be particularly relevant in the context of SLAM-based capture devices.

- For applications such as land surface processes and building information modeling, a focus on using registration techniques to monitor changes in a scene over time could be fruitful.
- More efforts should be dedicated to the registration of multi-temporal point clouds in cases in which the scene is dynamic with moving and/or changing objects.
- It would be useful to investigate and establish approaches for a more accurate assessment of the registration uncertainty and, possibly, its spatial patterns in real-life applications wherein the ground truth is unknown.

## Acknowledgments

The author is grateful for the financial support from Xi'an Jiaotong–Liverpool University, which includes the Research Enhancement Fund (grant number REF-21-01-003) and Key Program Special Fund (grant number KSF-E-40).

## Conflict of interest

There is no conflict of interest.

## References

1. Fan L, Smethurst JA, Atkinson PM, et al. (2015) Error in target-based georeferencing and registration in terrestrial laser scanning. *Comput Geosci* 83: 54–64. <https://doi.org/10.1016/j.cageo.2015.06.021>
2. Cai Y, Fan L (2021) An efficient approach to automatic construction of 3D watertight geometry of buildings using point clouds. *Remote Sens* 13: 1947. <https://doi.org/10.3390/rs13101947>
3. Fan L (2020) A comparison between structure-from-motion and terrestrial laser scanning for deriving surface roughness: a case study on a sandy terrain surface. *Int Arch Photogramm Remote Sens Spatial Inf Sci* 42: 1225–1229. <https://doi.org/10.5194/isprs-archives-XLII-3-W10-1225-2020>
4. Zeng A, Song S, Nießner M, et al. (2017) 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1802–1811. <https://doi.org/10.1109/CVPR.2017.29>
5. Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving? The KITTI vision benchmark suite. *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3354–3361. <https://doi.org/10.1109/CVPR.2012.6248074>
6. Pomerleau F, Liu M, Colas F, et al. (2012) Challenging data sets for point cloud registration algorithms. *Int J Rob Res* 31: 1705–1711. <https://doi.org/10.1177/0278364912458814>
7. Huang X, Mei G, Zhang J, et al. (2021) A comprehensive survey on point cloud registration. ArXiv, abs/2103.02690. <https://doi.org/10.48550/arXiv.2103.02690>
8. Wu Z, Song S, Khosla A, et al. (2015) 3D Shapenets: A deep representation for volumetric shapes. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1912–1920.

9. Reitmann S, Neumann L, Jung B (2021) BLAINDER-A Blender AI Add-On for Generation of Semantically Labeled Depth-Sensing Data. *Sensors* 21: 2144. <https://doi.org/10.3390/s21062144>
10. Griffiths D, Boehm J (2019) SynthCity: A large scale synthetic point cloud. *ArXiv*, abs/1907.04758. <https://doi.org/10.48550/arXiv.1907.04758>
11. CARLA Simulator (2021) Readthedocs.io. 2021. Available from: <https://carla.readthedocs.io/en/0.9.11/>.
12. LGSVL (2021) lgsvl/simulator. GitHub. Available from: <https://github.com/lgsvl/simulator>.
13. Chen Y, Medioni GG (1992) Object modelling by registration of multiple range images. *Image Vision Comput* 10: 145–155. [https://doi.org/10.1016/0262-8856\(92\)90066-C](https://doi.org/10.1016/0262-8856(92)90066-C)
14. Campbell RJ, Flynn PJ (2001) A survey of free-form object representation and re-cognition techniques. *Comput Vision Image Understanding* 81: 166–210. <https://doi.org/10.1006/cviu.2000.0889>
15. Segal A, Haehnel D, Thrun S (2009) Generalized-ICP. *Robotics: Science and Systems*, 5: 168–176. <https://doi.org/10.15607/RSS.2009.V.021>
16. Chetverikov D, Svirko D, Stepanov D, et al. (2002) The trimmed iterative closest point algorithm. *IEEE Object Recognition Supported by User Interaction for Service Robots*, 3, 545–548. <https://doi.org/10.1109/ICPR.2002.1047997>
17. Yang J, Li H, Jia Y (2013) Go-ICP: solving 3D registration efficiently and globally optimally. *Proceedings of the IEEE Conference on Computer Vision*, 1457–1464. <https://doi.org/10.1109/ICCV.2013.184>
18. Qiu D, May S, Nüchter A (2009) GPU-accelerated nearest neighbor search for 3D re-gistration. *International Conference on Computer Vision Systems*, Springer, Berlin, 194–203. [https://doi.org/10.1007/978-3-642-04667-4\\_20](https://doi.org/10.1007/978-3-642-04667-4_20)
19. Biber P, Straßer W (2003) The normal distributions transform: A new approach to laser scan matching. *Proceedings of International Conference on Intelligent Robots and Systems (IROS 2003)*, Cat. No. 03CH37453, 3: 2743–2748. <https://doi.org/10.1109/IROS.2003.1249285>
20. Takeuchi E, Tsubouchi T (2006) A 3-D scan matching using improved 3-D normal distributions transforms for mobile robotic mapping. *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3068–3073. <https://doi.org/10.1109/IROS.2006.282246>
21. Zhou Z, Zhao C, Adolfsson D, et al. (2021) NDT-Transformer: Large-Scale 3D Point Cloud Localisation using the Normal Distribution Transform Representation. *ArXiv*, abs/2103.12292.
22. Han X, Jin J, Xie J, et al. (2018) A comprehensive review of 3D point cloud descriptors. *ArXiv*, abs/1802.02297.
23. Rusu RB, Blodow N, Beetz M (2009) Fast point feature histograms (FPFH) for 3D registration. *IEEE International Conference on Robotics and Automation (ICRA)*, 3212–3217. <https://doi.org/10.1109/ROBOT.2009.5152473>
24. Yang H, Carlone L (2019) A Polynomial-time Solution for Robust Registration with Extreme Outlier Rates. *ArXiv*: abs/1903.08588. <https://doi.org/10.48550/arXiv.1903.08588>
25. Yang H, Shi J, Carlone L (2020) TEASER: Fast and Certifiable Point Cloud Registration. *arXiv:2001.07715*.
26. Zhou Q, Park J, Koltun V (2016) Fast global registration. *Computer Vision - ECCV 2016 - 14th European Conference*, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II, 766–782. [https://doi.org/10.1007/978-3-319-46475-6\\_47](https://doi.org/10.1007/978-3-319-46475-6_47)

27. Theiler PW, Wegner JD, Schindler K, (1996) Keypoint-based 4-Points Congruent Sets – Automated marker-less registration of laser scans. *ISPRS J Photogram Remote Sens* 96: 149–163. <https://doi.org/10.1016/j.isprsjprs.2014.06.015>
28. Leong-Hoi A, Chambrial A, Collet M, et al. (2020) Non-rigid registration of 3D points clouds of deformed liver models with Open3D and PyCPD. *Proceedings of Unconventional Optical Imaging II*. <https://doi.org/10.1117/12.2555673>
29. Golyanik V, Taetz B, Reis G, et al. (2016) Extended coherent point drift algorithm with correspondence priors and optimal subsampling. *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1–9. <https://doi.org/10.1109/WACV.2016.7477719>
30. Wang L, Chen J, Li X, et al. (2019) Non-Rigid Point Set Registration Networks. *ArXiv*, abs/1904.01428. <https://doi.org/10.48550/arXiv.1904.01428>
31. Dong Z, Liang F, Yang B, et al. (2020) Registration of large-scale terrestrial laser scanner point clouds: A review and benchmark. *ISPRS J Photogram Remote Sens* 163: 327–342. <https://doi.org/10.1016/j.isprsjprs.2020.03.013>
32. Khoury M, Zhou QY, Koltun V (2017) Learning compact geometric features. *2017 Proceedings of the IEEE international conference on computer vision*, 153–161. <https://doi.org/10.1109/ICCV.2017.26>
33. Zhang Z, Dai Y, Sun J (2020) Deep learning based point cloud registration: an overview. *Virtual Reality & Intelligent Hardware* 2: 222–246. <https://doi.org/10.1016/j.vrih.2020.05.002>
34. Deng H, Birdal T, Ilic S (2018) PPFNet: Global context aware local features for robust 3D point matching. *2018 Proceedings of the IEEE conference on computer vision and pattern recognition*, 195–205. <https://doi.org/10.1109/CVPR.2018.00028>
35. Choy C, Park J, Koltun V (2019) Fully convolutional geometric features. *2019 Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8958–8966. <https://doi.org/10.1109/ICCV.2019.00905>
36. Choy C, Dong W, Koltun V (2020) Deep Global Registration. *2020 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2511–2520. <https://doi.org/10.1109/CVPR42600.2020.00259>
37. Bai X, Luo Z, Zhou L, et al. (2020) D3feat: Joint learning of dense detection and description of 3d local features. *2020 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6359–6367. <https://doi.org/10.1109/CVPR42600.2020.00639>
38. Huang S, Gojcic Z, Usvyatsov M, et al. (2021) PREDATOR: Registration of 3D Point Clouds with Low Overlap. *2021 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4267–4276. <https://doi.org/10.1109/CVPR46437.2021.00425>
39. Qin Z, Yu H, Wang C, et al. (2022) Geometric transformer for fast and robust point cloud registration. *2022 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11143–11152. <https://doi.org/10.1109/CVPR52688.2022.01086>
40. Poiesi F, Boscaini D (2022) Learning general and distinctive 3D local deep descriptors for point cloud registration. *IEEE Trans Pattern Anal Mach Intell*. <https://doi.org/10.1109/TPAMI.2022.3175371>
41. Horache S, Deschaud JE, Goulette F (2021) 3D Point Cloud Registration with Multi-Scale Architecture and Self-supervised Fine-tuning. *arXiv:2103.14533*. 2021.



42. Poiesi F, Boscaini D (2021) Distinctive 3D local deep descriptors. *The 25th International Conference on Pattern Recognition (ICPR)*, 5720–5727. <https://doi.org/10.1109/ICPR48806.2021.9411978>
43. Wang Y, Solomon JM (2019) Deep closest point: Learning representations for point cloud registration. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 3523–3532. <https://doi.org/10.1109/ICCV.2019.00362>
44. Wang Y, Solomon JM (2019) PRNet: Self-supervised learning for partial-to-partial registration. *arXiv: 1910.12240*. <https://doi.org/10.48550/arXiv.1910.12240>
45. Sarode V, Li X, Goforth H, et al. (2019) PCRNet: Point Cloud Registration Network using PointNet Encoding. *ArXiv: 1908.07906*. <https://doi.org/10.48550/arXiv.1908.07906>
46. Aoki Y, Goforth H, Srivatsan RA, et al. (2019) PointNetLK: Robust & efficient point cloud registration using PointNet. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7163–7172. <https://doi.org/10.1109/CVPR.2019.00733>
47. Yew ZJ, Lee GH (2020) RPM-Net: Robust point matching using learned features. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11824–11833. <https://doi.org/10.1109/CVPR42600.2020.01184>
48. Qi CR, Su H, Mo K, et al. (2017) Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 652–660.
49. Qi CR, Yi L, Su H, et al. (2017) Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv Neural Inf Process Syst*, 5099–5108.
50. Lucas BD, Kanade T (1981) An Iterative Image Registration technique with an application to stereo vision. *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI'81)*.
51. Wang C, Galoogahi HK, Lin CH, et al. (2018) Deep-LK for efficient adaptive object tracking. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 627–634. <https://doi.org/10.1109/ICRA.2018.8460815>
52. Huang X, Mei G, Zhang J (2020) Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11366–11374. <https://doi.org/10.1109/CVPR42600.2020.01138>
53. Li X, Pontes JK, Lucey S (2020) Deterministic PointNetLK for Generalized Registration. *ArXiv: abs/2008.09527*.
54. Wang Y, Sun Y, Liu Z, et al. (2018) Dynamic graph CNN for learning on point clouds. *arXiv: 1801.07829*.
55. Papadopoulos T, Lourakis MI (2000) Estimating the Jacobian of the singular value decomposition: Theory and applications. *European Conference on Computer Vision. Springer, Berlin*, 554–570. [https://doi.org/10.1007/3-540-45054-8\\_36](https://doi.org/10.1007/3-540-45054-8_36)
56. Vaswani A, Shazeer N, Parmar N, et al. (2017) Attention Is All You Need. *31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, Advances*, 5998–6008.
57. Vinyals O, Fortunato M, Jaitly N (2015) Pointer networks. *Proceedings of the 28th International Conference on Neural Information Processing Systems*, 2692–2700.

58. Gold S, Rangarajan A, Lu CP, et al. (1998) New algorithms for 2D and 3D point matching) pose estimation and correspondence. *Pattern Recognit* 31: 1019–1031. [https://doi.org/10.1016/S0031-3203\(98\)80010-1](https://doi.org/10.1016/S0031-3203(98)80010-1)
59. Sinkhorn R (1964) A relationship between arbitrary positive matrices and doubly stochastic matrices. *Ann Math Stat* 35: 876–879. <https://doi.org/10.1214/aoms/1177703591>
60. Thomas H, Qi CR, Deschaud JE, et al. (2019) KPConv: Flexible and Deformable Convolution for Point Clouds. 2019 *IEEE/CVF International Conference on Computer Vision (ICCV)*, 6410–6419. <https://doi.org/10.1109/ICCV.2019.00651>
61. Yew ZJ, Lee GH (2022) REGTR: End-to-end Point Cloud Correspondences with Transformers. 2022 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6677–6686. <https://doi.org/10.1109/CVPR52688.2022.00656>
62. Zhang Z, Chen G, Wang X, et al. (2021) DDRNet: Fast point cloud registration network for large-scale scenes. *ISPRS J Photogram Remote Sens* 175: 184–198. <https://doi.org/10.1016/j.isprsjprs.2021.03.003>
63. Cao AQ, Puy G, Boulch A, et al. (2021) PCAM: Product of Cross-Attention Matrices for Rigid Registration of Point Clouds. 2021 *IEEE/CVF International Conference on Computer Vision (ICCV)*, 13209–13218. <https://doi.org/10.1109/ICCV48922.2021.01298>
64. Lu F, Chen G, Liu Y, et al. (2021) HRegNet: A Hierarchical Network for Large-scale Outdoor LiDAR Point Cloud Registration. 2021 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16014–16023. <https://doi.org/10.1109/ICCV48922.2021.01571>
65. Li L, Fu H, Ovsjanikov M (2022) WSDesc: Weakly Supervised 3D Local Descriptor Learning for Point Cloud Registration. *IEEE Transactions on Visualization and Computer Graphics*. <https://doi.org/10.1109/TVCG.2022.3160005>
66. Xu H, Liu S, Wang G, et al. (2021) OMNet: Learning Overlapping Mask for Partial-to-Partial Point Cloud Registration. 2021 *IEEE/CVF International Conference on Computer Vision (ICCV)*, 3112–3121. <https://doi.org/10.1109/ICCV48922.2021.00312>
67. Gu X, Tang C, Yuan W, et al. (2022) RCP: Recurrent Closest Point for Point Cloud. 2022 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8216–8226. <https://doi.org/10.1109/CVPR52688.2022.00804>



AIMS Press

© 2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)