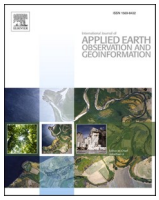
International Journal of Applied Earth Observations and Geoinformation 112 (2022) 102944

Contents lists available at ScienceDirect

International Journal of Applied Earth Observations and Geoinformation

journal homepage: www.elsevier.com/locate/jag



Rethinking of learning-based 3D keypoints detection for large-scale point clouds registration

ShaoCong Liu [a], Tao Wang [a], Yan Zhang [a], Ruqin Zhou [a], Chenguang Dai [a], Yongsheng Zhang [a], Haozhen Lei [a], Hanyun Wang [a,b,*]

[a] School of Surveying and Mapping, Information Engineering University, Zhengzhou 450001, China
[b] State Key Laboratory of Resources and Environmental Information System, Beijing 100000, China

ARTICLE INFO

Keywords:
Large-scale point clouds
3D keypoints detection
Deep learning
Point clouds registration

ABSTRACT

The main solution for large-scale point clouds registration is to first obtain a set of matched 3D keypoint pairs and then accomplish the point cloud registration task based on these matched keypoint pairs. However, at present, many methods study the feature descriptors in the point clouds registration task, but few methods discuss the 3D keypoints detection issues. The commonly used 3D keypoints detection strategy is the voxel-grid-based downsampling method, and the detected 3D keypoints are usually with a relatively huge amount and also with no explicit geometrical properties, which finally leads to a low inlier ratio. In this study, we rethink the 3D keypoints detection problem for large-scale point clouds with deep learning. Specifically, we discuss four kinds of 3D keypoints detection methods based on the joint keypoint detection and description learning framework D3Feat, and carry out extensive analyses on both the indoor large-scale point clouds dataset 3DMatch and the outdoor large-scale point clouds dataset KITTI Odometry. Experimental results demonstrate that the Multi-layer Perceptron (MLP) based method achieves the best inlier ratios under the different numbers of extracted 3D keypoints on both the indoor and outdoor large-scale point clouds. Further, we test these four kinds of keypoints detection methods under the application of large-scale point clouds registration, and the MLP-based method also achieves the state-of-the-art registration performance.

1. Introduction

Point clouds registration aims to estimate the transformation matrix between two overlapping point clouds, and then align these two point clouds into the same coordinate system. Nowadays, point cloud registration has been widely applied in various domains, for instance, augmented reality, autonomous driving, and simultaneous localization and mapping for mobile robots.

In the past few years, point clouds registration has been widely researched and current methods can be grouped into traditional methods and learning-based methods. The majority of traditional methods first match the hand-crafted local feature descriptors extracted from two point clouds and then utilize RANSAC (RANdom SAmple Consensus) (Fischler and Bolles, 1981) to filter out false matched point pairs. The coarse registration matrix is estimated from these matched point pairs, and the fine registration matrix is finally estimated based on the ICP (Iterative Closest Point) (Besl and McKay, 1992) method. Although several recently published methods directly estimate the transformation matrix from the learned global feature descriptors of the two point clouds (Choy et al., 2020; Xu et al., 2021; Wang et al.,2022), most learning-based methods are also based on the local feature descriptors extracting and matching strategies, except that the local features are learned based on the deep learning techniques, not extracted based on the hand-crafted design (Ao et al., 2022; Wu et al., 2021; Jiang et al., 2021).

For large-scale point clouds registration tasks, one critical issue is that we cannot take all points into consideration in both the traditional and learning-based registration methods because of the huge amount of hardware memory consumption and the low computational efficiency. A common strategy is to downsample the original large-scale point clouds, and current downsampling methods can be grouped into the heuristic sampling methods and learning-based sampling methods (Hu et al., 2021; Hu et al., 2020). Typical heuristic sampling methods include the voxel-grid-based sampling method (Shi and Rajkumar, 2020), the

* Corresponding author at: School of Surveying and Mapping, Information Engineering University, Zhengzhou 450001, China.
E-mail address: why.scholar@gmail.com (H. Wang).

https://doi.org/10.1016/j.jag.2022.102944
Received 16 June 2022; Received in revised form 20 July 2022; Accepted 28 July 2022
Available online 5 August 2022

Farthest Point Sampling (FPS) method (Qi et al., 2017a), the Inverse Density Importance Sub-Sampling (IDISS) method (Groh et al., 2019), and the Random Sampling (RS) method. The voxel-grid-based sampling method treats the centroids or the closet points to the centroids in each voxel as the downsampled points. Each point sampled by the FPS method is the farthest point with respect to the all already sampled points. The IDISS method first computes the density of each point and then preserves the points in low-density area, and the RS method randomly selects K points from the original point cloud. Typical learning-based sampling methods include the Generator-Based Sampling (GS) method (Dovrat et al., 2019), the Continuous Relaxation Based Sampling (CRS) method (Abubakar et al., 2019), and the Policy Gradient Based Sampling (PGS) method (Xu et al., 2015). The GS method proposes an S-NET to generate a group of points to represent the original point cloud. The CRS method avoids the non-differentiable step in the sampling operation through the reparameterization trick and selects the points through the Markov decision process. However, the FPS, IDIS, GS and CRS methods are with expensive computational cost or huge memory consumption, the RS method cannot control the sampling process, and the PGS method is difficult to converge. The voxel-grid-based sampling method has low memory consumption and high computational efficiency, and is currently the commonly used downsampling method for large-scale point cloud registration. However, the keypoints extracted by the voxel-grid-based method are usually with a relatively huge amount and with no explicit geometrical properties, which finally leads to a low inlier ratio.

To solve this issue, recently many researchers study the 3D keypoints detection problems with deep learning. A typical 3D keypoints detection module usually consists of a point saliency scores calculation layer and a top K points selection layer. (Li et al., 2018; Lu et al., 2019; Du et al., 2019) utilized the MLP layer to obtain the saliency scores. PRNet (Wang and Solomon, 2019b) calculates the saliency scores by using the norm of features. At present, there is little work related to the joint learning of the keypoints detection and feature description. 3DFeatNet (Yew and Lee, 2018) extracts keypoints and descriptors for point cloud registration through weakly supervised learning, but it made insufficient use of context information. USIP (Li and Lee, 2019) extracts 3D keypoints with a probabilistic chamfer loss through an unsupervised learning manner, but these keypoints are located in a certain area when the number of them is small. The D3Feat (Bai et al., 2020) proposes a new density-invariant keypoints selection strategy and treats points with maximal scores across the spatial and channel dimensions as the 3D keypoints. The experiments carried out on the 3DMatch (Zeng and Xiao, 2017) and KITTI Odometry (Geiger et al., 2012) datasets demonstrate its superiority on the 3D keypoints detection problem.

Although many 3D keypoints detection methods have been proposed recently, how to define a 3D keypoint or what is a 3D keypoint is still an unsolved problem, especially for large-scale point clouds. Thus, in this study, we will rethink the 3D keypoints detection problem for the large-scale point clouds with deep learning. Specifically, we discuss four kinds of 3D keypoints detection methods based on the D3Feat network. To evaluate the effectiveness of these four different methods, we carry out experiments on both the indoor large-scale point clouds dataset 3DMatch and the outdoor large-scale point clouds dataset KITTI Odometry. Further, we also test these four kinds of keypoint detection methods for large-scale point clouds registration. In summary, the main contributions are as follows:

- This paper is the first one that systematically analyzes the 3D keypoints detection problem for large-scale point clouds with deep learning.
- This paper discusses four different 3D keypoints detection methods, and compares their accuracy and stability on both the indoor and outdoor large-scale point clouds datasets.
- This paper further tests the effectiveness of these four kinds of 3D keypoints detection methods for large-scale point cloud registration.

2. Related works

2.1. 3D keypoints detectors

Hand-crafted Detectors: Hand-crafted detectors extract 3D keypoints by specifically designed rules, and usually can only detect 3D keypoints with specific geometrical properties. Harris 3D (Sipiran and Bustos, 2011) extends the 2D Harris corner detection method to 3D keypoints detection. Local Surface Patches (LSP) (Chen and Bhanu, 2007) and Shape Index (SI) (Chitra and Anil, 1997) extract 3D keypoints according to the maximum and minimum principal curvatures of the local surface. Intrinsic Shape Signatures (ISS) (Yu, 2009) and KeyPoint Quality (KPQ) (Mian et al., 2010) extract 3D keypoints by calculating the eigenvalues of the dispersion matrix. LORAX (Gil et al., 2017) uses the PCA to select 3D keypoints with common geometric features from the depth map. (Iqbal et al., 2019) used the 3D geometrical and RGB information to generate candidate 3D keypoints based on the eigenvalues of the covariance matrix and the Difference of Gaussian method. The hand-crafted detectors utilize the specific geometric properties around the points and thus can extract 3D keypoints with high repeatability. However, the distinguishable abilities of feature descriptors are usually limited, and the hand-crafted detectors can only detect specific kinds of 3D keypoints.

Learning-based Detectors: The 3D keypoints extracted by the learning-based detectors can effectively avoid the limits of hand-crated feature descriptors. (Georgakis et al., 2018) used the Region Proposal Network to generate a set of scores and regions of interest on depth images, and used the centroids of the regions of interest as the keypoints. (Li et al., 2018) put the extracted features into the MLP layer and extracted the 3D keypoints in the point cloud by processing the matching matrix and one-hot matrix. Kpsnet (Du et al., 2019) calculates the saliency scores of each point by using an MLP layer. PRNet (Wang and Solomon, 2019b) calculates the L_2 norm of the learned features and selects the top K points as the 3D keypoints. Deep VCP (Lu et al., 2019) network constructs a point weighting layer composed of MLP and softplus, and then selects keypoints with maximal scores through the top K layer. USIP (Li and Lee, 2019) network proposes the FPN (Feature Proposal Network) to compute the saliency uncertainties and selects the points with the smallest salient uncertainties as the 3D keypoints. D3Feat (Bai et al., 2020) network treats points with maximal scores across the spatial and channel dimensions as the 3D keypoints. The trainable keypoint detector network SKD (Salient Keypoint Detector) (Tinchev et al., 2021) firstly concatenates context-aware features, saliency and PCA features, and then feeds them into a fully connected network to extract keypoints. The 3D3L (Streiff et al., 2021) method firstly generates the reliability and repeatability score maps, and then obtains the keypoints score map by multiplying these two score maps. The points with larger values in the score map are extracted as keypoints. In Skeleton Merger (Shi et al., 2021), the original points and their features extracted through the PointNet++ (Qi et al., 2017a) are multiplied to extract 3D keypoints. These approaches demonstrate that the learning-based detectors can extract repeatable and stable keypoints through data-driven manner and outperform the hand-crafted detectors.

2.2. 3D feature descriptors

Before the rise of deep learning, point cloud feature descriptors are mainly hand-crafted descriptors. Because of the data-driven superiority, deep learning techniques have been introduced into the point cloud feature learning in recent years. (Guo et al., 2021) had systematically reviewed the point cloud feature learning methods, and the readers can refer to their paper for details. In this paper, we will shortly introduce typical point clouds feature learning methods commonly used in the point clouds registration task.

Voxel-based methods: VoxNet (Maturana and Scherer, 2015) builds a 3D convolutional neural network on the regularized voxel grids

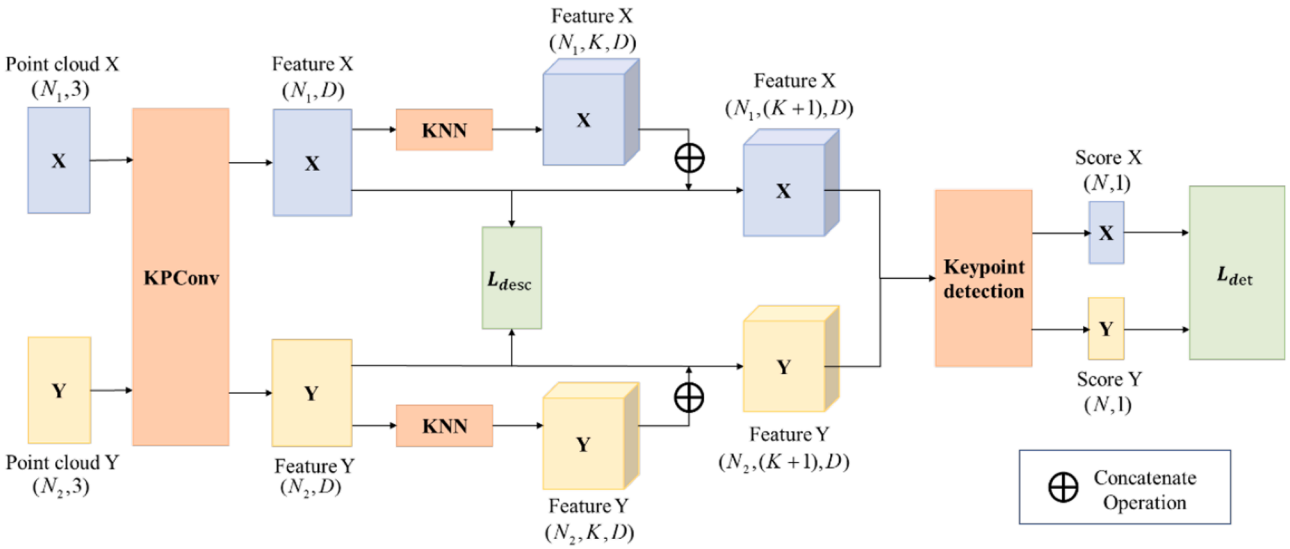


Fig. 1. The pipeline of the training stage.

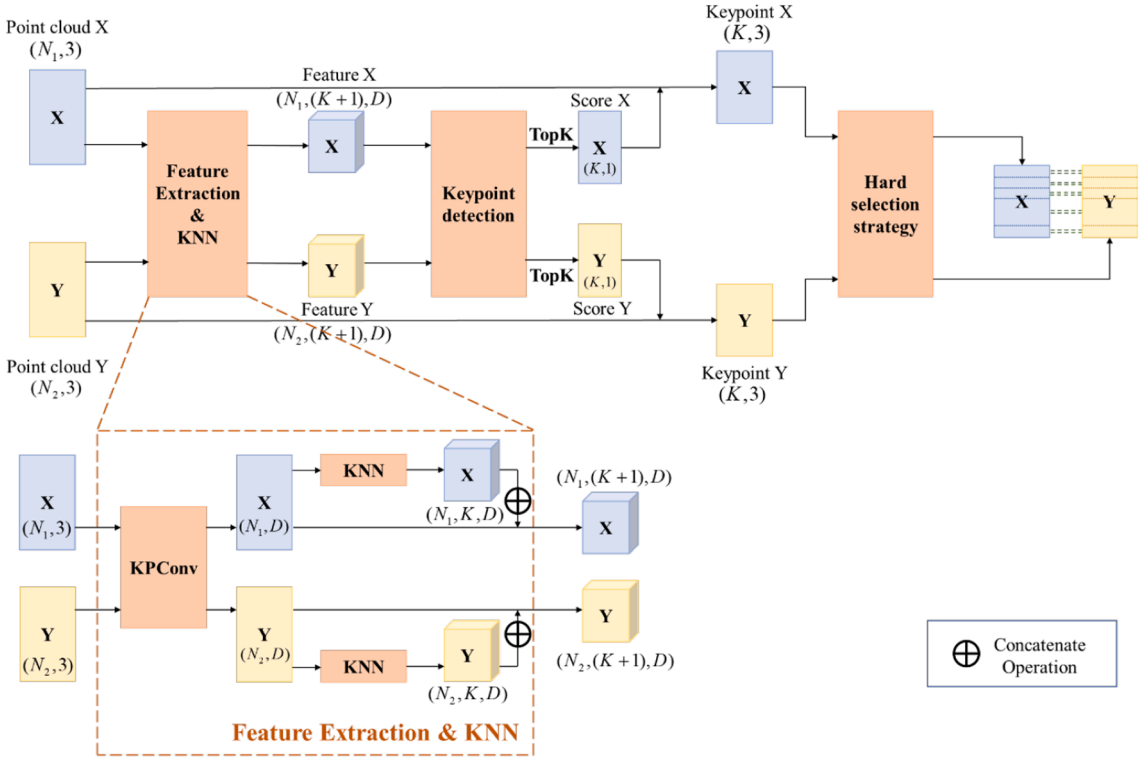


Fig. 2. The pipeline of the testing stage.

after point cloud voxelization. 3DMatch (Zeng and Xiao, 2017) establishes the point correspondences by learning the descriptors of local voxels. FCGF (Choy et al., 2019b) uses the sparse tensors to represent the point cloud data and uses Minkowski Engine (Choy et al., 2019a) to extract the pointwise features.

Point-based methods: PointNet (Qi et al., 2017b) directly inputs the point clouds and extracts pointwise features through MLP layers. However, the PointNet learns pointwise features just from its properties such as coordinates, which leads to a poor distinguishable ability. Subsequently, PointNet++ (Qi et al., 2017a) enlarges the feature receptive field by referring to the idea of CNN, and utilizes the local neighborhood of each point to learn pointwise features. The PPFNet (Deng et al., 2018a) utilizes simple geometric attributes consisting of coordinates, normals, and point pair features (PPFs) to generate distinguishable and rotation-resistant local features. DCP (Wang and Solomon, 2019a) introduces the attention module to enhance the capability of feature learning. The SiamesePointNet (Zhou et al., 2020) generates keypoint descriptors through a series of hierarchical encoder-decoder network structures based on PointNet.

Graph-based methods: The DGCNN utilizes the EdgeConv to learn the features of each node from the local neighborhood information in the graph (Wang et al., 2018). KCNet (Shen et al., 2018) firstly utilizes KNN (K-Nearest Neighbor) to build a graph and then exploits the kernel correlation to extract local features. Graphit (Saleh et al., 2020) learns

pointwise features through a graph convolution network (GCN) (Kipf and Welling, 2017) and extracts the salient points by the GCN and fully connected layers. (Kong et al., 2020) constructed a semantic graph based on the node features composed of the semantic category and the centroids of the instances, and then learned the features through the node embedding layer.

3. Methodology

In this paper, we choose the D3Feat network as our evaluation framework because of its joint keypoints detection and feature learning ability. To better understand our work, we first shortly introduce the overview of the D3Feat network, and then discuss four different 3D keypoints detection methods.

3.1. Network architecture overview

To understand the D3Feat network more clearly, we rewrite the training and testing stages and also redraw the pipelines of these two stages.

Training Stage: The pipeline of the training stage is shown in the Fig. 1. Given two point clouds denoted as $X \in R^{N_1 \times 3}$ and$Y \in R^{N_2 \times 3}$, the D3Feat network first uses KPConv (Thomas et al., 2019) to learn pointwise features, which utilizes the predefined kernel points and continuous convolution kernel to aggregate context information from the local neighbors of each point. However, the original formulation of KPConv is susceptible to the variation of point cloud density, thus the D3Feat network adds a density normalization term to ensure the convolutional sparsity invariance. The pointwise features extracted from the two input point clouds are denoted as $F_X \in R^{N_1 \times D}$ and$F_Y \in R^{N_2 \times D}$, respectively.

Then, 3D keypoints are extracted based on their pointwise features. To utilize the neighborhood information, KNN is adopted to search the K nearest points for each point in the coordinate space. The features of neighbors are then concatenated with the features of each point and denoted as $F_X \in R^{N_1 \times (K+1) \times D}$ and$F_Y \in R^{N_2 \times (K+1) \times D}$, respectively. To calculate the saliency score of each point, the concatenated features are fed into the keypoints detection module. The saliency score represents the probability of a point to be a keypoint. The details of keypoints detection layer in D3feat are described in the next subsection. The loss functions used in D3Feat are composed of two components, that are the descriptor loss and the detector loss. The details of these two loss functions are described as follows:

$$L_{desc} = \frac{1}{n}\sum_{i}[\max(0, d_{pos}(i) - M_{pos}) + \max(0, M_{neg} - d_{neg}(i))] \tag{1}$$

$$L_{\det} = \frac{1}{n}\sum_{i}[(d_{pos}(i) - d_{neg}(i))(S_{x_i} + S_{y_i})] \tag{2}$$

$$L = \alpha L_{desc} + (1 - \alpha)L_{det} \tag{3}$$

For descriptor loss function, $d_{pos}(i)$ and $d_{neg}(i)$ are the distances of corresponding and non-corresponding point pairs, M_{pos} and M_{neg} are the distance margins for these two kinds of point pairs, respectively. For detector loss function, (x_i, y_i) is a correspondence pair, and (S_{x_i}, S_{y_i}) is their saliency scores. The feature descriptor loss L_{desc} and the detector loss $L_{\det}$ are finally summed together, and α is the weight coefficient.

Testing Stage: As shown in Fig. 2, the feature extraction layer consists of a KPConv-based pointwise feature learning layer and a KNN-based neighborhood feature aggregation layer, which are described in the training stage. Let us denote the learned pointwise features of X and Y as $F_X \in R^{N_1 \times (K+1) \times D}$ and$F_Y \in R^{N_2 \times (K+1) \times D}$, the keypoints detection module calculates the pointwise saliency score, and selects K points with the maximal saliency scores through the top K layer, which are denoted as $K_X \in R^{K \times 3}$ and$K_Y \in R^{K \times 3}$, respectively. Finally, the hard selection strategy is carried out to obtain corresponding point pairs in terms of distance in the feature space. The whole testing stage is formulated as follows:

$$\begin{aligned} X_k &= topK(\phi(F(x_1, x_2, x_3 \ldots x_m))) \in R^{K \times 3} \\ Y_k &= topK(\phi(F(y_1, y_2, y_3 \ldots y_m))) \in R^{K \times 3} \end{aligned} \tag{4}$$

where $x_i \in X$ and $y_i \in Y$ belong to two inputted point clouds, F denotes the feature description layer, ϕ denotes the keypoint detection layer, $topK$ denotes the top K layer, and X_k and Y_k denote the extracted keypoints of two inputted point clouds, respectively.

3.2. Keypoints detection methods

Ideally, keypoints are distinctive and repeatable with respect to the variations of views and noises. Hand-crafted keypoints detectors usually define points with specific geometrical properties as 3D keypoints, and thus they can only detect specific kind of 3D keypoints. Learning-based keypoints detectors utilize deep learning techniques to automatically learn rich representative pointwise features, but how to define 3D keypoints in the deep learning framework is still an unsolved problem. Thus, in this section, we will discuss four kinds of 3D keypoints detection methods based on the joint keypoints detection and description learning network D3Feat. The details are described as follows.

Local-Maximum-based Detector (LM-Detector): Inspired by (Dusmanu et al., 2019.), the D3Feat network treats points with maximal scores across the spatial and channel dimensions as the 3D keypoints. To measure the saliency of a point, the D3Feat network defines two kinds of saliency scores in both the spatial dimension and the channel dimension of feature maps. For the spatial saliency score α_i^k of point x_i at the k-th channel, D3Feat first calculates the distance of the feature of point x_i from the average feature of its neighboring points at the k-th channel, and further normalizes this difference through softplus function, as described in the equation (5):

$$\alpha_i^k = In(1 + \exp(F_i^k - \frac{1}{|N_{x_i}|}\sum_{x_j \in N_{x_i}} F_j^k)) \tag{5}$$

where F_i^k and F_j^k represent the k-th channel of the feature descriptors of point x_i and x_j, and N_{x_i} is the neighboring points of x_i. By computing the relative difference between each point and its mean feature of local neighbors, the spatial saliency achieves density invariance. For the channel saliency score β_i^k of point x_i at the k-th channel, D3Feat calculates the ratio of the feature at the k-th channel to its largest feature value at all channels of the feature map, as described in the equation (6):

$$\beta_i^k = \frac{F_i^k}{\max_t(F_i^t)} \tag{6}$$

Finally, we compute the product of α_i^k and β_i^k for each point, and take the largest value along the channel dimension as the saliency score for this point, as described in the equation (7):

$$S_i = \max_k(\alpha_i^k \cdot \beta_i^k) \tag{7}$$

L_2-norm-based Detector (L_2-Detector): PRNet (Wang and Solomon, 2019b) defines the saliency score of each point as the L_2 norm of the learned feature vectors, and takes the top K points with largest saliency scores as the 3D keypoints. In this paper, we also take the L_2 norm of the learned feature vectors as a measurement of keypoints' saliency scores. However, during training we found that the training process cannot converge when we directly connected the L_2 norm layer to the feature extraction layer. The convergent ability of the network is affected by calculating the L_2 norm of the learned features and taking them as salience scores directly. In view of the above reasons, a fully connected layer is added after the feature extraction layer, and then the pointwise saliency is calculated through the L_2 norm. The equation is formulated as follows:

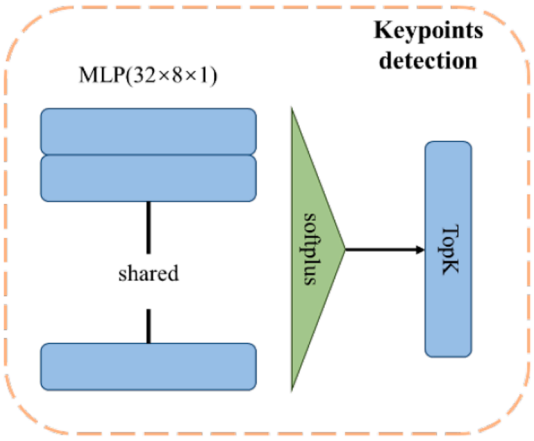


Fig. 3. Architecture of the MLP-based 3D keypoints detector.

Table 1
The inlier ratio and registration recall on 3DMatch.

Method	Inlier Ratio (%)					Registration Recall (%)
# Keypoints	250	500	1000	2500	5000	250
LM-Detector	45	44.1	42.7	40.6	40.7	83.4
L2-Detector	27.25	32.47	38.3	43.25	45.66	85.3
FDN-Detector	39.7	42.03	44.6	45.74	45.97	**85.8**
MLP-Detector	**59.28**	**56.91**	**54.38**	**50.7**	**48.04**	**85.8**

$$S_i = \sqrt[2]{\sum_{k=1}^{D} \left(F_i^k\right)^2} \tag{8}$$

where F_i^k represents the k-th channel of the learned feature for point x_i. The S_i represents the saliency score of the point x_i.

Features-Difference-with-Neighborhood Detector (FDN-Detector): Inspired by the keypoint detectors such as FAST (Rosten and Drummond, 2006), SURF (Herbert et al., 2008) and SuperGlue (Sarlin et al, 2020) in the 2D images, a keypoint must be obviously different from its neighborhood in the feature space. Thus, we also introduce a novel 3D keypoint detector which has not been used in the previous work. The mathematical principle is that the L_2 norm of the distance vector of the 3D keypoint from its neighbors in the feature space should be relatively large, and is formulated as follows:

$$S_i = \sum_{k=1}^{D} \left\| F_i^k - \frac{1}{|N_{x_i}|} \sum_{x_j \in N_{x_i}} F_j^k \right\|_2 \tag{9}$$

$$\{S_1, S_2, ..., S_K\} = topK\{S_1, S_2, ..., S_M\} \tag{10}$$

MLP-based Detector (MLP-Detector): MLP has exhibited excellent non-linear fitting ability in the 2D images and 3D point clouds keypoints detection, such as SuperGlue (Sarlin et al., 2020), Deep VCP (Lu et al., 2019), USIP (Li and Lee, 2019), etc. Usually the MLP-based keypoints detection layer consists of an MLP layer, a softplus activation function and a top K layer, as shown in Fig. 3. The learned pointwise features $F_x \in R^{N_1 \times (K+1) \times D}$ and $F_y \in R^{N_2 \times (K+1) \times D}$ are fed into the MLP layer to learn the saliency scores, and then the learned saliency scores are normalized by the softplus activation function. The 3D keypoints are finally extracted by selecting K points with the largest saliency scores through the top K layer.

4. Experiments

4.1. Implementation details

The framework of our experiments is based on TensorFlow 1.12 and Ubuntu 20.04. All experiments in our study are carried out on a workstation with an Intel Core i9-9700 CPU, a 16-GB RAM, and a 24-GB NVIDIA Titan GPU. The performances of these four methods are evaluated on two public available large-scale indoor and outdoor point clouds datasets 3DMatch and KITTI Odometry. The batch size, the initial parameters of momentum optimizer lr and the momentum are set to 1, 0.1 and 0.98. The M_{pos},M_{neg} and α are set to 0.1, 1.4 and 0.5 respectively.

4.2. Evaluation on 3DMatch

The 3DMatch contains a total of 62 indoor point clouds scenes and has been used as the benchmark for point clouds registration. We follow the D3Feat to preprocess this dataset. In the training process, point cloud pairs with overlapping areas larger than 30 % are selected, and a total of 35,297 pairs of point clouds are finally selected. In the training dataset, the minimum value, maximum value and average value of overlap are 0.30, 0.995, and 0.515 respectively. For each pair of point clouds, the data augmentation including noise, rotation and scaling disturbances is applied. Specifically, the rotation disturbance is randomly selected between 0° and 360°, the scaling disturbance is randomly selected between 0.9 and 1.1, and the gaussian noise is generated with a standard deviation of 0.005. According to the transformation matrix contained in the 3DMatch dataset, the keypoints correspondences are constructed and the point pairs with spatial distance lower than 0.1 m after transformation are considered as corresponding points. The minimum number, maximum number and average number of keypoints are 850, 197,343 and 27,127 respectively. The testing dataset contains a total of 433 pairs of point clouds in 8 scenes.

Evaluation metrics: Following the D3Feat network, three metrics including the *inlier ratio*, the *registration recall* and the *feature matching recall (FMR)* are adopted. First, the saliency scores of the two point clouds are calculated, and the top K points with the largest saliency scores are selected as the 3D keypoints. Then, the distances in the feature space between these two keypoint sets are computed and the point pairs with the nearest distance are considered as the corresponding points. That is to say, if x_i and y_j are the corresponding points, then.

$$j = \text{argmin}_t\left(\|F_{x_i} - F_{y_t}\|\right) \;\&\; i = \text{argmin}_t\left(\|F_{y_j} - F_{x_t}\|\right)$$

where F_{x_i} and F_{y_j} are the learned features of the x_i and y_j respectively, F_{x_t} and F_{y_t} are the learned features of the x_t and y_t respectively.

The extracted corresponding point pairs are then transformed to the same coordinate system according to the ground truth transformation matrix, and the points with distances smaller than the inlier distance threshold are treated as the inlier points. The inlier ratio I is computed as follows:

$$I = \frac{1}{N} \sum_{s=1}^{N} \left[\frac{1}{|\Omega_s|} \sum_{(x_i, y_j) \in \Omega_s} \mathbf{1}\left(\left\| (Rx_i + T) - y_j \right\| < \beta \right) \right]$$

where N is the number of corresponding point pairs for all pairwise point clouds, x_i and y_j are the matched keypoints belonging to the set of corresponding point pairs Ω_s in the s-th pairwise point clouds, and $\mathbf{1}(\cdot)$ is the counting function. The R and T respectively denote the ground truth rotation matrix and translation vector contained in the 3DMatch dataset, and $\beta = 0.1$ m is the inlier distance threshold.

The registration recall (Choi et al., 2015) measures the ratio of successfully registered point cloud pairs based on the matched keypoints, and is evaluated through the RMSE metric, shown as follows:

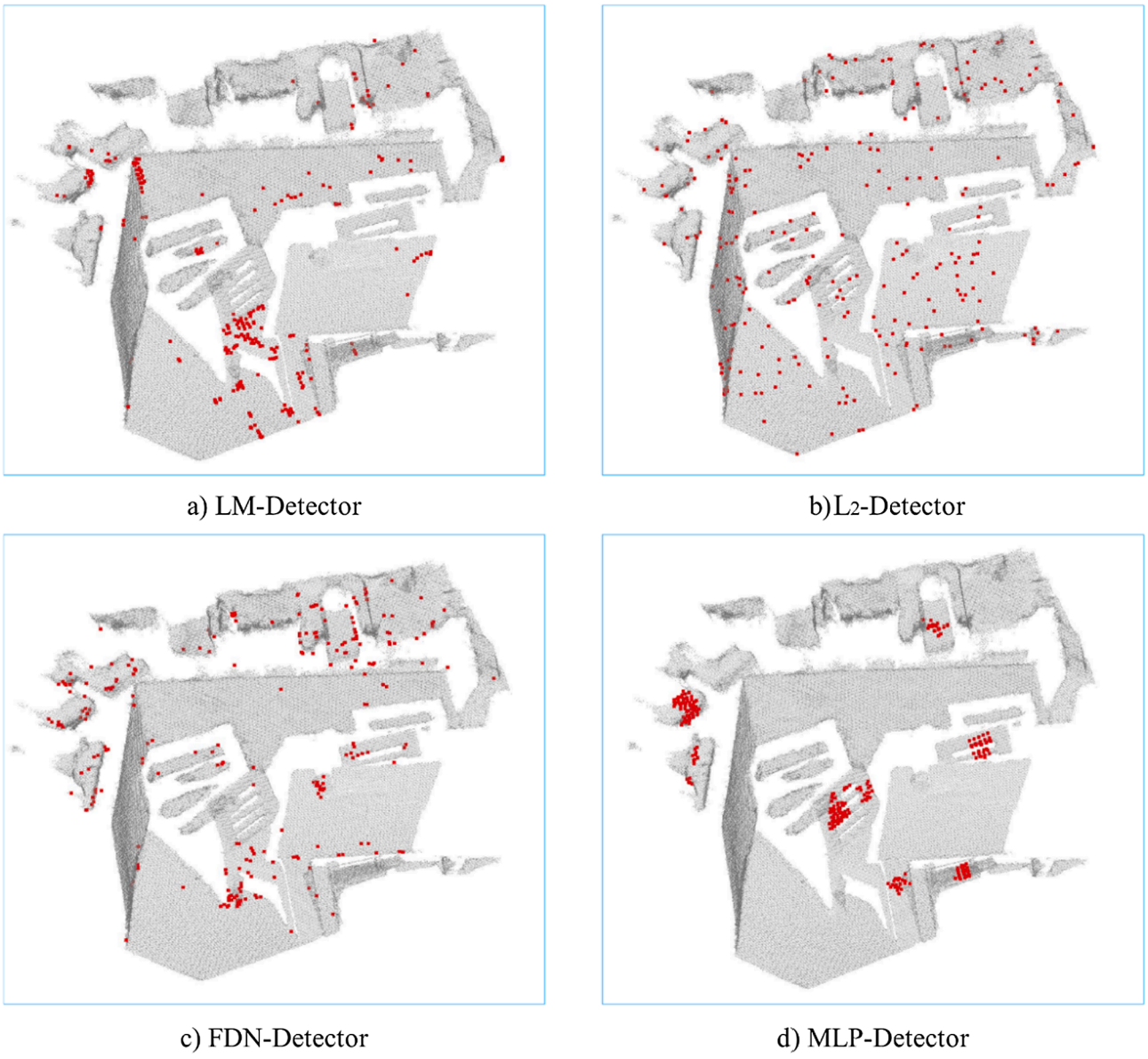


Fig. 4. The visualization of extracted keypoints on 3DMatch.

Table 2
The result of FMR on 3DMatch.

	Origin		Rotated	
	FMR (%)	STD	FMR (%)	STD
FPFH (Rusu et al., 2009)	35.9	13.4	36.4	13.6
SHOT (Salti et al., 2014)	23.8	10.9	23.4	9.5
3DMatch (Zeng and Xiao, 2017)	59.6	8.8	1.1	1.2
CGF (Khoury et al., 2017)	58.2	14.2	58.5	14
PPFNet (Deng et al., 2018a)	62.3	10.8	0.3	**0.5**
PPF-FoldNet (Deng et al., 2018b)	71.8	10.5	73.1	10.4
PerfectMatch (Gojcic et al., 2019)	94.7	2.7	94.9	2.5
FCGF (Choy et al., 2019b)	95.2	2.9	95.3	3.3
LM-Detector	95.8	2.9	**95.5**	3.5
L2-Detector	94.93	3.01	94.49	3.25
FDN-Detector	95.15	2.18	94.05	3.54
MLP-Detector	**95.9**	**2.02**	**95.5**	2.39

Table 3
The inlier ratio on KITTI Odometry.

Inlier Ratio (%)					
# Keypoints	**250**	**500**	**1000**	**2500**	**5000**
LM-Detector	15.97	19.46	23.17	28.32	30.29
L2-Detector	9.66	14.3	19.58	26.29	30.14
FDN-Detector	15.85	19.8	24.12	30.19	34.28
MLP-Detector	**46.59**	**47.49**	**47.17**	**46.21**	**44.62**

$$E_{RMSE} = \sqrt{\frac{1}{|\Omega_s|} \sum_{(x_i, y_j) \in \Omega_s} \|(Rx_i + T) - y_j\|^2}$$

Based on this equation, two overlapping point clouds are assumed be successfully aligned if the E_{RMSE} is lower than 0.2 m.

The FMR measures the matching quality during pairwise registration and is obtained by calculating the percentage of successful alignment with the inlier ratio higher than a certain threshold (i.e., $\tau = 5$ %).

Comparison of the inlier ratio: To evaluate the inlier ratio performance of four different kinds of 3D keypoints detection methods, we

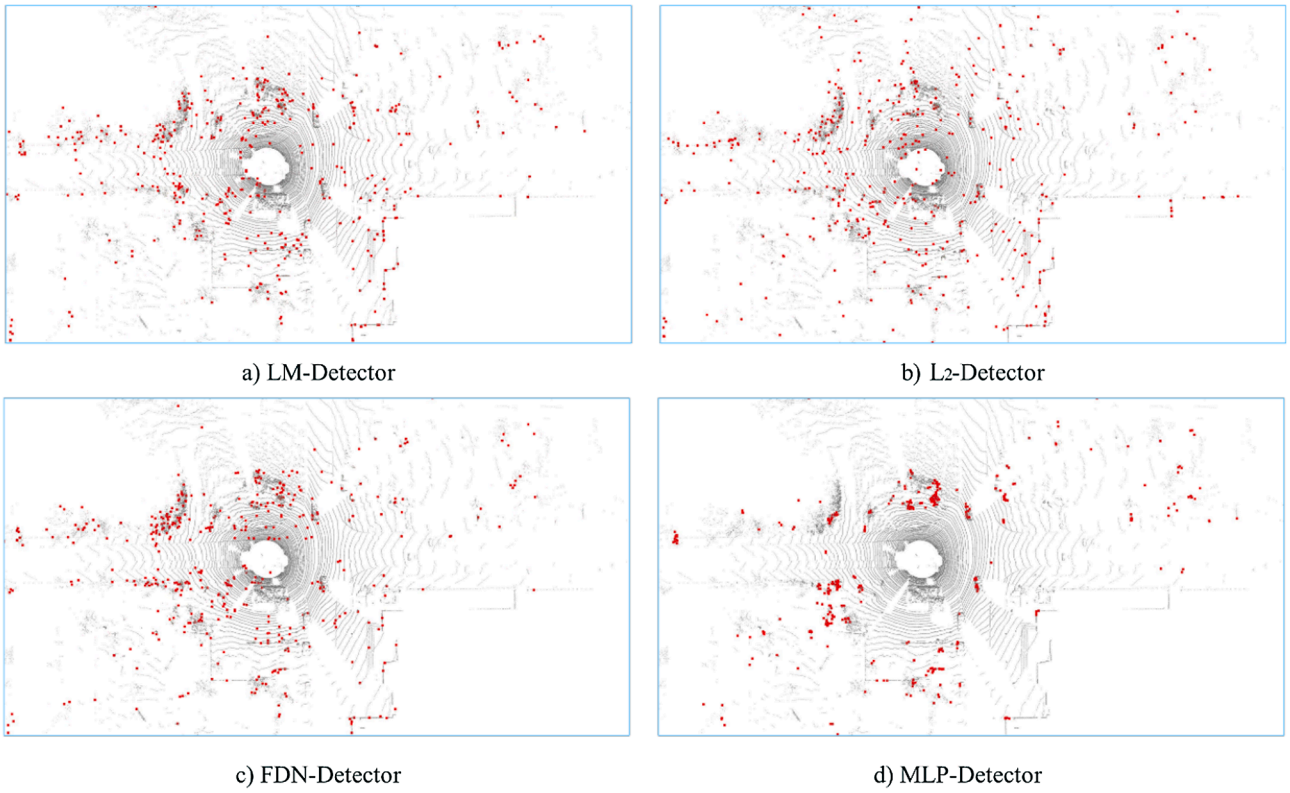


Fig. 5. The visualization of extracted keypoints on KITTI Odometry.

Table 4
Quantitative comparisons on KITTI Odometry.

	RTE (cm)		RRE (°)	
	AVG	STD	AVG	STD
3DFeat-Net (Yew and Lee, 2018)	25.9	26.2	0.57	0.46
FCGF (Choy et al., 2019b)	9.52	1.3	0.3	0.28
LM-Detector	6.9	0.3	0.24	**0.06**
L2-Detector	5.24	0.04	0.143	0.25
FDN-Detector	4.92	0.04	**0.141**	0.25
MLP-Detector	**4.46**	**0.04**	0.19	0.24

conduct experiments with different numbers of selected keypoints in the last step of the testing stage, that is 250, 500, 1000, 2500 and 5000 respectively in the experiments.

As shown in Table 1, when using L_2-Detector, the modified D3Feat network achieves lower inlier ratios than the LM-Detector proposed in the original D3Feat method when the numbers of sampled points are 250, 500 and 1000, and achieves relative higher inlier ratios than the LM-Detector when the numbers of sampled points are 2500 and 5000. The performance of using FDN-Detector is better than that of L_2-Detector. The reason for the poor performance of L_2-Detector with the small numbers of sampled points is that the L_2 norm of feature vectors has no obvious geometric meaning, and the repeatable probability of extracted keypoints is low, as described in section 4.4. The MLP-Detector obtains the best performance at all the sampling strategies. Especially, the inlier ratio of MLP-Detector is 59.28 % when the number of sampled points is 250. This is because MLP can fit high dimensional non-linear functions and also is optimized according to the training data. Thus, compared with other three keypoints detectors, MLP-Detector is more suitable for point clouds with various kinds of geometric structures, and can extract specific kinds of keypoints adaptively according to the input point clouds. To better understand the keypoints detection performance on the 3DMatch dataset, Fig. 4 visualizes the distribution of 250 keypoints extracted by these four methods. The reason for the concentrative distribution of keypoints extracted by the MLP-Detector is that, points with similar features should have similar saliency scores and the detector loss also encourages to detect easily matchable correspondences, which is explained in detail in the original D3feat network (Bai et al., 2020).

Comparison of the registration recall: Instead of selecting 250 keypoints according to the saliency scores directly, we utilize the downsampling and hard selection strategy to improve the registration recall. Firstly, the original point clouds are downsampled to 5000 points according to the saliency scores. Secondly, by using the hard selection strategy, the corresponding points between two point clouds can be obtained. Finally, we select 250 keypoints with highest saliency scores to calculate the registration recall. As shown in Table 1, the L_2-Detector, FDN-Detector and MLP-Detector achieve higher registration recall than LM-Detector, and the FDN-Detector and MLP-Detector achieve the best performance of 85.8 % on the registration recall.

Comparison of the FMR: To evaluate the robustness of these four methods, we also compare these four methods with other methods on the FMR metric. As shown in Table 2, the MLP-Detector obtains the highest FMR compared with all other methods, regardless of whether the rotation transformation is added to the point clouds. The MLP-Detector achieves 95.9 % on the origin point cloud, which is higher than 95.8 % using the LM-Detector. The results of FMR using the L_2-Detector and FDN-Detector fall short of expectations, which are both lower than the LM-Detector.

4.3. Evaluation on KITTI Odometry

The KITTI Odometry contains outside point clouds acquired by the Velodyne HDL64 and ground truth poses provided by the GPS/INS system. Following the protocol of D3Feat, the training set contains 14,136 pairs of point clouds from 0 to 5 sequences, the validation set contains 2202 pairs of point clouds from 6 to 7 sequences, and the

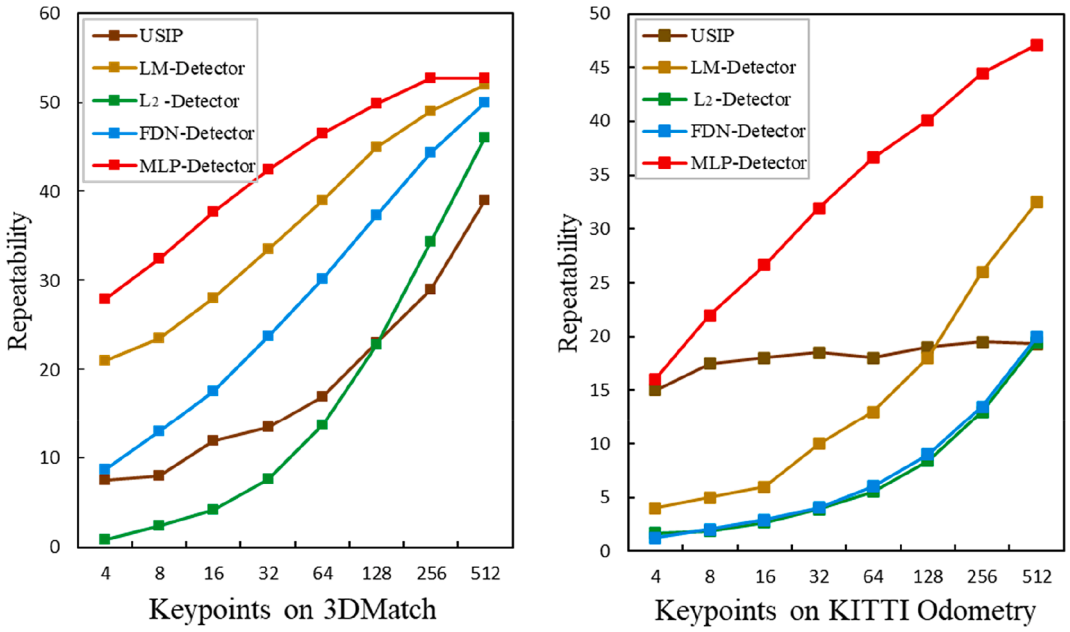


Fig. 6. Repeatability of keypoints on the 3DMatch and KITTI Odometry.

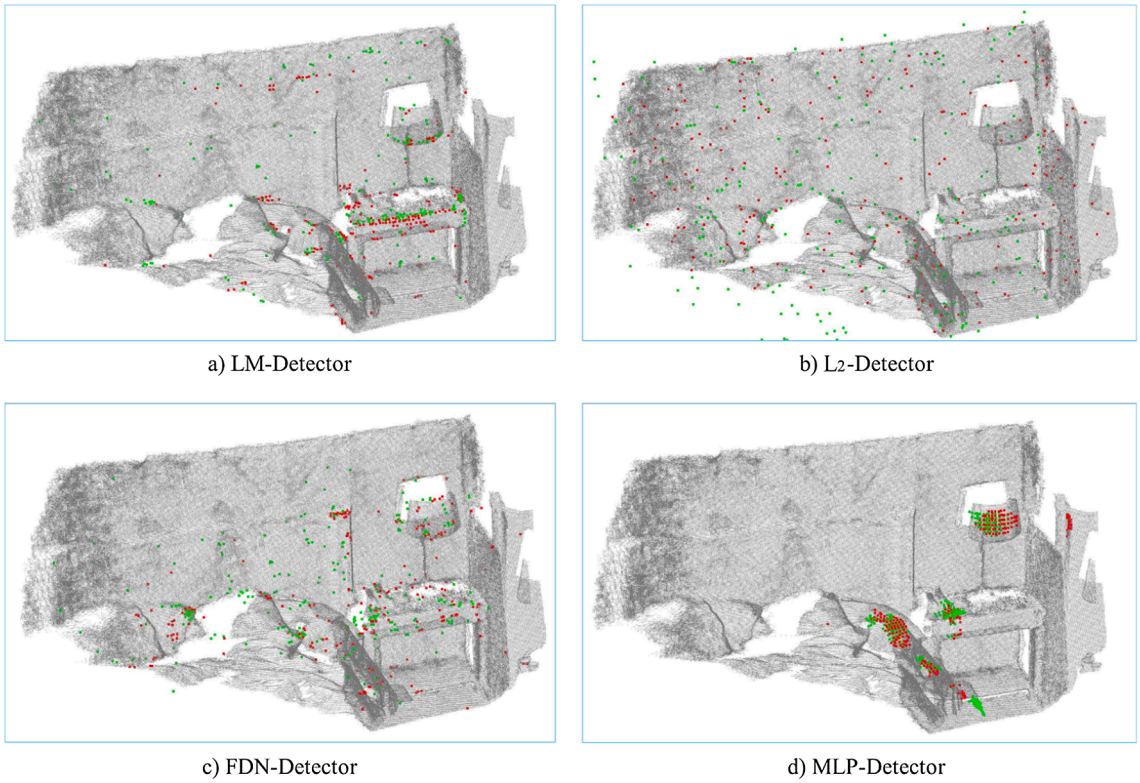


Fig. 7. The distribution of extracted 250 keypoints on the 3DMatch dataset.

testing set contains 6876 pairs of point clouds from 8 to 10 sequences. The point clouds are also downsampled with a sampling interval of 0.3 m for preprocessing.

Evaluation metric: Three evaluation metrics including *inlier ratio*, *relative translation error (RTE)* and r*elative rotation error (RRE)* proposed in (Elbaz et al., 2017) are adopted in this section.

Comparison of the inlier ratio: We select 250, 500, 1000, 2500 and 5000 keypoints to evaluate the inlier ratio performance. As shown in Table 3, the inlier ratio gradually decreases with the reduction of the number of sampled keypoints using the LM-Detector, which is different from that in 3DMatch dataset. This indicates that the LM-Detector is not applicable in large-scale outdoor scenes. The L_2-Detector achieves the lowest inlier ratios than the other three methods on all the different numbers of sampled keypoints. The FDN-Detector achieves similar inlier ratios to the LM-Detector when the numbers of sampled points are 250, and achieves higher inlier ratios when the numbers of sampled points are 500, 1000, 2500 and 5000. The MLP-Detector obtains the best performance in all situations compared with the other three methods.

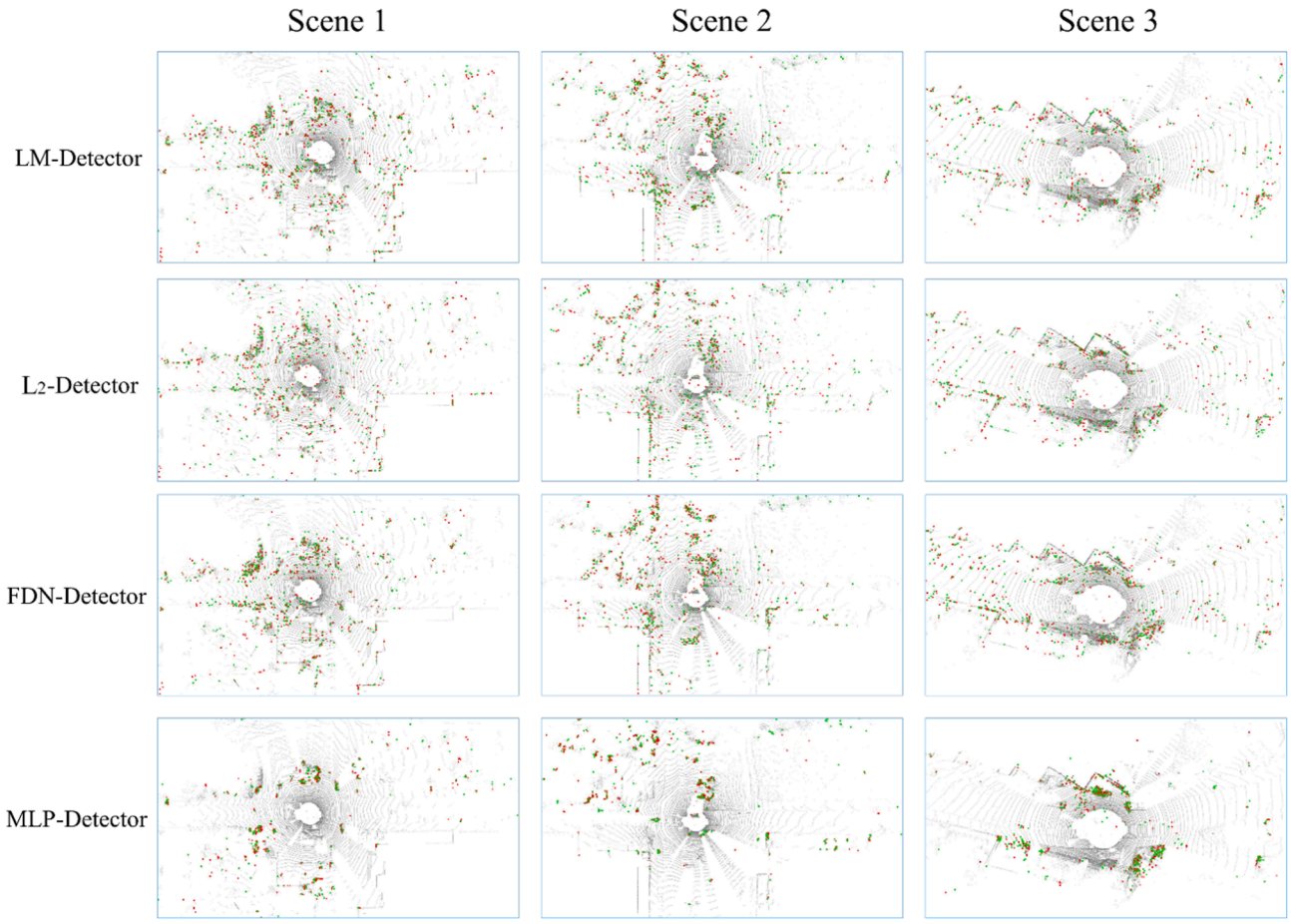


Fig. 8. The distribution of extracted 512 keypoints on the KITTI Odometry dataset.

Especially when the number of selected keypoints is 250, the MLP-Detector method still achieves the inlier ratio of 46.59 %, while that of the LM-Detector is only 15.97 %. In addition, although the inlier ratio decreases with the increasing number of sampled keypoints, the MLP-Detector still achieves 44.62 % when the number of sampled points is 5000. This demonstrates that the MLP-Detector is still applicable in large-scale outdoor scenes. To better understand the keypoint detection performance on the KITTI Odometry dataset, Fig. 5 visualizes the distributions of 250 keypoints extracted by these four methods. Similar to 3DMatch dataset, keypoints extracted by the MLP-Detector distribute concentrative in the outdoor scenes.

Comparison of the registration performance: The RTE and RRE are adopted to evaluate the registration performance of these four methods. First, the transformation matrix is calculated according to the two sets of matched keypoints based on the RANSAC algorithm. Then, the errors of the rotation matrix and translation vector are calculated with respect to the provided ground truth transformation matrix. In detail, we extract 5000 keypoints for each point cloud, and compare these four keypoints detection methods as well as 3DFeatNet (Yew and Lee, 2018), and FCGF (Choy et al., 2019b).

As shown in Table 4, the L_2-Detector and FDN-Detector achieve better performance than LM-Detector. The average relative translation errors are respectively 5.24 cm and 4.92 cm and the standard deviations of relative translation errors are both 0.04 cm for these two methods. The MLP-Detector obtains the best performance, where the average relative translation error is 4.46 cm and the standard deviation of relative translation error is 0.04 cm.

From Table 4, we can also observe that the average relative rotation errors of L_2-Detector, FDN-Detector and MLP-Detector are all lower than the LM-Detector, which are 0.143°, 0.141° and 0.19° respectively, but the standard deviation of relative rotation errors of them are higher than the LM-Detector. In addition, the FDN-Detector achieves the lowest average relative rotation errors.

4.4. Keypoints repeatability

Repeatability is also a very important metric to measure the robustness of keypoints detection methods. After transforming the two input point clouds into the same coordinate system, repeatability means that the distance between the keypoint extracted from the source point cloud and its nearest keypoint extracted from the target point cloud is smaller than the predefined threshold. Similar to D3Feat, the keypoints repeatability is evaluated on the 3DMatch and KITTI Odometry. The distance threshold is set to 0.1 m on the 3DMatch and 0.5 m on the KITTI Odometry respectively.

Following the protocol of D3Feat, we compare these four keypoints detection methods with USIP (Li and Lee, 2019). Specifically, we extract 4, 8, 16, 32, 64, 128, 256 and 512 keypoints in the 3DMatch and KITTI Odometry datasets respectively, and calculate the ratios of points that are qualified as repeatable keypoints. As shown in Fig. 6, we can see that (a) the MLP-Detector performs best on both the 3DMatch and KITTI Odometry datasets and the repeatability of the MLP-Detector is better than all other methods on all different numbers of keypoints; (b) the L_2-Detector and FDN-Detector perform worse on both the 3DMatch and KITTI Odometry datasets. The repeatability of the L_2-Detector and FDN-Detector are lower than the LM-Detector in all different numbers of keypoints.

To evaluate the repeatability of keypoints, we also visualize the extracted keypoints from two different frames with different colors for the 3DMatch dataset and KITTI Odometry dataset. Specifically, the two

sets of keypoints are first extracted and then transformed into the same coordinate system according to the given ground truth transformation matrix. As shown in Fig. 7 and Fig. 8, the positions of most of the keypoints extracted by the MLP-Detector are almost the same, which demonstrates that the MLP-Detector achieves robust repeatable performance on both the large-scale indoor and outdoor point clouds datasets.

5. Conclusion

In this study, four kinds of 3D keypoints detection methods including LM-Detector, FDN-Detector, L_2-Detector and MLP-Detector are discussed based on the D3Feat network. The evaluation experiments are carried out on the 3DMatch and KITTI Odometry datasets. Four metrics for each dataset are utilized to evaluate the performances of these four kinds of methods. Experimental results demonstrate that MLP-based keypoints detection method achieves the best performances on the metrics of inlier ratio, FMR, registration recall, and repeatability on the 3DMatch dataset, and achieves the best performances on metrics of inlier ratio, RTE, RRE, and repeatability on the KITTI Odometry dataset. Our future work is to construct an end-to-end deep neural network for large-scale point clouds registration with MLP-based keypoints detection module.

CRediT authorship contribution statement

ShaoCong Liu: Conceptualization, Methodology, Investigation, Writing – original draft, Writing – review & editing. **Tao Wang:** Conceptualization, Writing – review & editing. **Yan Zhang:** Formal analysis, Resources, Writing – review & editing. **Ruqin Zhou:** Methodology, Writing – review & editing. **Chenguang Dai:** Investigation, Writing – review & editing. **Yongsheng Zhang:** Resources, Supervision, Project administration. **Haozhen Lei:** Formal analysis. **Hanyun Wang:** Conceptualization, Methodology, Investigation, Writing – original draft, Supervision, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

This research is supported by a grant from State Key Laboratory of Resources and Environmental Information System.

References

Abubakar, A., Muhammad, F.B., James, Z., 2019. Concrete Autoencoders for Differentiable Feature Selection and Reconstruction. International Conference on Machine Learning (ICML), 97.
Ao, S., Guo, Y., Hu, Q., Yang, B., Markham, A., Chen, Z., 2022. You Only Train Once: Learning General and Distinctive 3D Local Descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence.
Besl, P.J., McKay, N.D., 1992. A Method for Registration of 3-D Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (2), 239–256.
Chen, H., Bhanu, B., 2007. 3D Free-form Object Recognition in Range Images Using Local Surface Patches. Pattern Recognition Letters 28 (10), 1252–1262.
Bai, X., Luo, Z., Zhou, L., Fu, H., Quan, L., Tai, C., 2020. D3Feat: Joint Learning of Dense Detection and Description of 3D Local Features. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 6358–6366.
Chitra D., Anil K., 1997. Cosmos-A Representation Scheme for 3D Free-form Objects. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(10), 1115–1130.
Choi, S., Zhou, Q., Koltun, V., 2015. Robust Reconstruction of Indoor Scenes. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 5556-5565.
Choy, C., Dong, W., Koltun, V., 2020. Deep Global Registration. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2511-2520.
Choy, C., Gwak, J., Savarese, S., 2019. 4D Spatio-temporal Convnets: Minkowski Convolutional Neural Networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3070-3079.
Choy, C., Park, J., Koltun, V., 2019. Fully Convolutional Geometric Features. IEEE International Conference on Computer Vision (ICCV), 8957-8965.
Deng, H., Birdal, T., Ilic, S., 2018. PPFNet: Global Context Aware Local Features for Robust 3D Point Matching. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 195-205.
Deng, H., Birdal, T., Ilic, S., 2018. PPF-FoldNet: Unsupervised Learning of Rotation Invariant 3D Local Descriptors. European Conference on Computer Vision (ECCV), 11209: 620-638.
Dovrat, O., Lang, I., Avidan, S., 2019. Learning to Sample. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2755-2764.
Du, A., Huang, X., Zhang, J., Yao, L., Wu, Q., 2019. Kpsnet: Keypoint Detection and Feature Extraction for Point Cloud Registration. IEEE International Conference on Image Processing (ICIP), 2576-2580.
Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., Sattler, T., 2019. D2-Net: A Trainable CNN for Joint Description and Detection of Local Features. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 8084-8093.
Elbaz, G., Avraham, T., Fischer, A., 2017. 3D Point Cloud Registration for Localization Using a Deep Neural Network Autoencoder. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2472-2481.
Fischler, M.A., Bolles, R.C., 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM 24 (6), 381–395.
Georgakis, G., Karanam, S., Wu, Z., Ernst, J., Košecká, J., 2018. End-to-End Learning of Keypoint Detector and Descriptor for Pose Invariant 3D Matching. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 1965–1973.
Geiger, A., Lenz, P., Urtasun, R., 2012. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3354-3361.
Gil, E,, Tamar, A., Anath, F., 2017. 3d Point Cloud Registration for Localization Using a Deep Neural Network Auto-encoder. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2472-2481.
Gojcic, Z., Zhou, C., Wegner, J.D., Wieser, A., 2019. The Perfect Match: 3D Point Cloud Matching with Smoothed Densities. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 5540-5549.
Groh, F., Wieschollek, P., Lensch, H., 2019. Flex-Convolution Million-Scale Point-Cloud Learning Beyond Grid-Worlds. Asian Conference on Computer Vision (ACCV), 11361, 105-122.
Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M., 2021. Deep Learning for 3D Point Clouds: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (12), 4338–4364.
Hu, Q., Yang, B., Xie, L., Rosa, S., Markham, A., 2020. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 11105-11114.
Iqbal, M.Z., Bobkov, D., Steinbach, E., 2019. Adaptive Fusion-Based 3D Keypoint Detection for RGB Point Clouds. IEEE International Conference on Image Processing (ICIP), 3711-3715.
Jiang, H., Shen, Y., Xie, J., Li, J., Qian, J., Yang, J., 2021. Sampling Network Guided Cross-Entropy Method for Unsupervised Point Cloud Registration. IEEE International Conference on Computer Vision (ICCV), 6108-6117.
Khoury, M., Zhou, Q., Koltun, V., 2017. Learning compact geometric features. IEEE International Conference on Computer Vision (ICCV), 153-161.
Herbert, B., Andreas, E., Tinne, T., Luc, V.G., 2008. Speeded-Up Robust Features (SURF). Computer Vision and Image Understanding (CVIU) 110 (3), 346–359.
Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A., 2021. Learning Semantic Segmentation of Large-Scale Point Clouds with Random Sampling. IEEE Transactions on Pattern Analysis and Machine Intelligence.
Kipf, T.N., Welling, M., 2017. Semi-Supervised Classification with Graph Convolutional Networks. International Conference on Learning Representations (ICLR).
Kong, X., Yang, X., Zhai, G., Zhao, X., Zen,g X., Wang, M., Liu, Y., Li, W., Wen, F., 2020. Semantic Graph Based Place Recognition for 3D Point Clouds. IEEE International Conference on Intelligent Robots and Systems (IROS), 8216-8223.
Li, J., Lee, G.H., 2019. USIP: Unsupervised Stable Interest Point Detection from 3D Point Clouds. IEEE International Conference on Computer Vision (ICCV), 361-370.
Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B., 2018. PointCNN: Convolution on X-transformed Points. In: Conference on Neural Information Processing Systems (NIPS), p. 31.
Lu, W., Wan, G., Zhou, Y., Fu, X., Yuan, P., Song, S., 2019. DeepVCP: An End-to-End Deep Neural Network for Point Cloud Registration. IEEE International Conference on Computer Vision (ICCV), 12-21.
Maturana, D., Scherer, S., 2015. Voxnet: A 3d Convolutional Neural Network for Real-time Object Recognition. IEEE International Conference on Intelligent Robots and Systems (IROS), 922-928.
Mian, A.S., Bennamoun, M., Owens, R.A., 2010. On The Repeatability and Quality of Keypoints for Local Feature-based 3D Object Retrieval from Cluttered Scenes. International Journal of Computer Vision 89 (2–3), 348–361.
Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in A Metric Space. International Conference on Neural Information Processing Systems (NIPS), 5105–5114.
Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 77-85.

Rosten, E., Drummond, T., 2006. Machine Learning for High-speed Corner Detection. European Conference on Computer Vision (ECCV), 430–443.
Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast Point Feature Histograms (FPFH) for 3D Registration. IEEE International Conference on Robotics and Automation (ICRA), 3212-3217.
Saleh, M., Dehghani, S., Busam, B., Navab, N., Tombari, F., 2020. Graphite: Graph-induced Feature Extraction for Point Cloud Registration. International Conference on 3D Vision (3DV), 241-251.
Salti, S., Tombari, F., Di Stefano, L., 2014. SHOT: Unique Signatures of Histograms for Surface and Texture Description. Computer Vision and Image Understanding (CVIU) 125, 251–264.
Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A., 2020. SuperGlue: Learning Feature Matching with Graph Neural Networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4937-4946.
Shen, Y., Feng, C., Yang, Y., Tian, D., 2018. Mining Point Cloud Local Structures by Kernel Correlation and Graph Pooling. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4548-4557.
Shi, R., Xue, Z., You, Y., Lu, C., 2021. Skeleton Merger: An Unsupervised Aligned Keypoint Detector. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 43-52.
Shi, W., Rajkumar, R., 2020. Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1708-1716.
Sipiran, I., Bustos, B., 2011. Harris 3D: A Robust Extension of the Harris Operator for Interest Point Detection on 3D Meshes. Visual Computer 27 (11), 963.
Streiff, D., Bernreiter, L., Tschopp, F., Fehr, M., Siegwart, R., 2021. 3D3L: Deep Learned 3D Keypoint Detection and Description for LiDARs. IEEE International Conference on Robotics and Automation (ICRA). 13064-13070.
Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas L.J., 2019. KPConv: Flexible and Deformable Convolution for Point Clouds. IEEE International Conference on Computer Vision (ICCV), 6410-6419.
Tinchev, G., Penate-Sanchez, A., Fallon, M., 2021. SKD: Keypoint Detection for Point Clouds Using Salient Estimation. IEEE Robotics and Automation Letters (RA-L), 6(2), 3785-3792.
Wang, Y., Solomon, J., 2019. Deep Closest Point: Learning Representations for Point Cloud Registration. IEEE International Conference on Computer Vision (ICCV), 3522-3531.
Wang, Y., Sun, Y., Liu, Z., Sarma, S., Bronstein, M., Solomon, J., 2018. Dynamic graph CNN for Learning on Point Clouds. ACM Transactions on Graphics 38 (5), 1–12.
Wang, Y., Solomon, J., 2019. PRNet: Self-Supervised Learning for Partial-to-Partial Registration. Conference on Neural Information Processing Systems (NIPS), 32.
Wang, Y., Yan, C., Feng, Y., Du, S., Dai, Q., Gao, Y., 2022. STORM: Structure-based Overlap Matching for Partial Point Cloud Registration. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).
Wu, B., Ma, J., Chen, G., An, P., 2021. Feature Interactive Representation for Point Cloud Registration. IEEE International Conference on Computer Vision (ICCV), 5510-5519.
Xu, H., Liu, S., Wang, G., Liu, G., Zeng, B., 2021. OMNet: Learning Overlapping Mask for Partial-to-partial Point Cloud Registration. IEEE International Conference on Computer Vision (ICCV), 3132-3141.
Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y., 2015. Show Attend and Tell: Neural Image Caption Generation with Visual Attention. International Conference on Machine Learning (ICML), 37, 2048-2057.
Yew, Z.J., Lee, G.H., 2018. 3DFeat-Net: Weakly Supervised Local 3D Features for Point Cloud Registration, European Conference on Computer Vision (ECCV), 607-623.
Yu, Z., 2009. Intrinsic Shape Signatures: A Shape Descriptor for 3d Object Recognition. Computer Vision Workshops, 689-696.
Zeng, A., Xiao, J., 2017. 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 199-208.
Zhou, J., Wang, M., Mao, W., Gong, M., 2020. SiamesePointNet: A Siamese Point Network Architecture for Learning 3D Shape Descriptor. Comput. Graphics Forum 39 (1), 309–321.