



Article

GraM: Geometric Structure Embedding into Attention Mechanisms for 3D Point Cloud Registration

Pin Liu ¹, Lin Zhong ¹, Rui Wang ^{2,*} , Jianyong Zhu ³, Xiang Zhai ⁴ and Juan Zhang ⁵ 

¹ School of Information Engineering, China University of Geosciences, Beijing 100083, China; liupin@cugb.edu.cn (P.L.); zhonglin@email.cugb.edu.cn (L.Z.)

² School of Computer Science and Engineering, Beihang University, Beijing 100191, China

³ Department of Computer, North China Electric Power University, Beijing 102206, China; zhuji@ncepu.edu.cn

⁴ China Reasset Management Ltd., Beijing 100033, China; zhaixiang@cramc.cn

⁵ Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne NE1 8ST, UK; juan.zhang@northumbria.ac.uk

* Correspondence: ruiking@buaa.edu.cn

Abstract: 3D point cloud registration is a crucial technology for 3D scene reconstruction and has been successfully applied in various domains, such as smart healthcare and intelligent transportation. With theoretical analysis, we find that geometric structural relationships are essential for 3D point cloud registration. The 3D point cloud registration method achieves excellent performance only when fusing local and global features with geometric structure information. Based on these discoveries, we propose a 3D point cloud registration method based on geometric structure embedding into the attention mechanism (GraM), which can extract the local features of the non-critical point and global features of the corresponding point containing geometric structure information. According to the local and global features, the simple regression operation can obtain the transformation matrix of point cloud pairs, thereby eliminating the semantics that ignores the geometric structure relationship. GraM surpasses the state-of-the-art results by 0.548° and 0.915° regarding the relative rotation error on ModelNet40 and LowModelNet40, respectively.

Keywords: 3D point cloud registration; deep learning; attention mechanism; geometric structure



Citation: Liu, P.; Zhong, L.; Wang, R.; Zhu, J.; Zhai, X.; Zhang, J. GraM: Geometric Structure Embedding into Attention Mechanisms for 3D Point Cloud Registration. *Electronics* **2024**, *13*, 1995. <https://doi.org/10.3390/electronics13101995>

Academic Editor: Claus Pahl

Received: 18 February 2024

Revised: 24 April 2024

Accepted: 25 April 2024

Published: 20 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Three-dimensional reconstruction provides digital 3D models by presenting a real-world scene, promoting the development of augmented reality, such as autonomous driving and digital twins [1–5]. On the other hand, point cloud data is fundamental for building a digital 3D model because 3D point cloud registration is the core technology for achieving 3D reconstruction by providing stereoscopic models [6,7]. In recent years, the rapid development of sensor technologies enables 3D point cloud data to visualize the world, further promoting technology innovation in practice.

Three-dimensional point cloud registration is a crucial step in the 3D reconstruction process. It aims to learn the local information of an object from multiple perspectives and integrate it into a unified perspective to obtain objects with rich global information. Specifically, it evaluates the correlation of each corresponding point in the point cloud sample by learning the characteristics of each point cloud sample under multiple visual angles.

Then, it estimates the rigid transformation parameters that ensure all the corresponding points can be directly transformed in 3D space. Subsequently, it obtains the converting matrix of the point cloud sample, thereby achieving the goal of 3D point cloud registration.

In the latest research, many methods utilize attention mechanisms for learning point cloud features [8,9]. For instance, the Transformer can extract the local and global features of the corresponding points between the clouds to obtain the transformation matrix for point clouds [10–12]. However, the features extracted by the traditional Transformer do not

contain the critical geometric information for point cloud registration, making it impossible to achieve better performance. Therefore, for 3D point cloud registration technology, extracting features containing geometric structure information becomes a solid challenge.

To solve the above challenges, we propose a 3D point cloud registration method that embeds geometric structure into the attention mechanism, forming an end-to-end registration framework. More specifically, REGTR [13] is a classic 3D point cloud registration model. Its famous innovation is to adopt the attention mechanism in the Transformer, which can effectively obtain the global and local features, to replace traditional feature matching.

Nonetheless, the extracted features do not include information on geometric structure, which is extremely important for registering point cloud data. To this end, we take REGTR as the primary architecture and introduce two embedded modules to extract geometric structures. The two structures are bound with the original two self-attention structures of REGTR, respectively, to learn more rich features that contain information on geometric structure. The fusion of global and local features that include geometric structure information can improve the accuracy of point cloud registration. Extensive experimental validation demonstrates that the proposed method can significantly outperform the state-of-the-art mechanisms.

The main contributions of this paper are summarized as follows:

- As far as authors know, this is the first proposal to embed the geometric structure into an improved REGTR network. The proposed GraM effectively promotes the local features integrated with information on geometric structure and global features.
- We introduce the attention mechanism to the point cloud registration task and optimize the feature extraction on the REGTR network, significantly improving the accuracy and efficiency of the low-overlap point cloud registration task.
- Comprehensive experiments on the reconstructed ModelNet40 and KITTI datasets show that GraM obtains better accuracy than state-of-the-art methods.

The remainder of this paper is organized as follows. Section 2 illustrates the related work on 3D point cloud registration technology. Descriptions of the problem definition and the core technology used are in Section 3. Our research methodology and specific implementation steps are introduced in detail in Section 4. Section 5 evaluates the performance of our proposed 3D point cloud registration method. Finally, we summarize the paper and present future research in Section 6.

2. Related Work

Extensive research has been conducted on 3D point cloud registration technologies, which include optimization-based registration, feature learning-based registration, and end-to-end learning-based registration.

Optimization-based point cloud registration. Besl et al. [14] proposed the classic Iterative Closest Point (ICP) algorithm, which iteratively estimates corresponding points between two point clouds and their transformation matrix to achieve registration. IMLP was proposed in [15] to improve the corresponding point estimation of ICP by incorporating measurement noise into the transformation estimation. Segal et al. [16] proposed a generalized version of ICP that allows for the inclusion of arbitrary covariance matrices in ICP variants using point-to-plane metrics. Zhu et al. [17] proposed a graph registration method that simultaneously considers vertices and edges to find point-to-point correspondences between two graphs. Huang et al. [18] introduced a novel pruning module to enhance deep learning-based point cloud registration in low overlap scenarios (Predator), resulting in significant performance improvements. However, the computational efficiency and registration accuracy were significantly decreased when these methods were used to deal with large-scale datasets and low-overlap scenarios.

Feature learning-based point cloud registration. Zeng et al. [19] introduced 3DMatch, a parallel network trained from RGBD images, to extract features by combining the local structure around critical points and further capture the local characteristics of the 3D point cloud. The network 3DFeatNet [20] uses a weakly supervised approach to learn feature

correspondences from 3D point clouds. RPMNet [21] can obtain soft correspondences of points in partially overlapping point clouds from a mixture of features learned from spatial coordinates and local geometry. A dynamic graph convolutional neural network is employed in Deep Closest Point (DCP) [22] for feature extraction, which then uses an attention module to learn the correspondence between two point clouds. It still utilizes an SVD module to calculate the rotation matrix and translation vector required for the transformation. These algorithms cannot optimize post-processing operations through learning methods during training, resulting in significant limitations in performance. Wang et al. [23] proposed a novel local descriptor-based framework (YOHO). It employs rotation-equivariant descriptors to achieve robust and efficient point cloud registration with superior performance compared to conventional methods. Recently, Zhang et al. [24] presented a novel approach utilizing rotation-invariant features and spatial geometric consistency for robust partial-to-partial point cloud registration, outperforming existing methods, particularly in handling large rotations. Liu et al. [25] proposed a group-wise contrastive learning (GCL) scheme to extract density-invariant geometric features.

End-to-end learning-based point cloud registration. The core idea of these methods is to add the transformation matrix to the learning network to avoid the impact of post-processing operations on the algorithm's performance. Deng et al. [26] proposed a relative pose regression network that can directly estimate the relative pose of point clouds based on features learned from local descriptors. Yasuhiro et al. [27] proposed PointNetLK by combining PointNet with the Lucas–Kanade (LK) algorithm [28]. DGR [29] utilized fully convolutional geometric features (FCGFs) [30] for feature extraction from point clouds. The six-dimensional convolutional network structure was employed to estimate point correspondences, and a weighted Procrustes model was used to estimate the transformation. Yu et al. [31] employed rotation-invariant and globally aware descriptors for robust point cloud registration (RIGA), surpassing state-of-the-art methods, especially in managing large rotations across diverse datasets. Recently, Yew et al. [13] applied Transformer for the first time in the point cloud registration. It extracted global and local features through a multi-headed attention mechanism (REGTR), alleviating the problem of point cloud registration at low overlap. These methods can achieve higher accuracy but require more complex network structures and greater computational power.

Although the above methods can complete the point cloud data registration at a certain level, they ignore the most critical geometric structure information. Only the extracted features include geometric structure information, and the 3D coordinates obtained will be more accurate, further improving the registration accuracy.

3. Preliminary

3.1. Problem Definition

Three-dimensional point cloud registration task can be described as follows: there are two point clouds to be registered, $\mathbf{X} \in \mathbb{R}^{M \times 3}$ and $\mathbf{Y} \in \mathbb{R}^{N \times 3}$. \mathbf{X} represents the source point cloud. \mathbf{Y} represents the target point cloud. M and N are the number of points in the source and target point clouds, respectively. The task of 3D point cloud registration involves utilizing a rigid transformation composed of a rotation matrix $\mathbf{R} \in SO(3)$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$ to align the source point cloud \mathbf{X} with the target point cloud \mathbf{Y} . Therefore, the process of 3D point cloud registration is the process of finding the optimal rotation matrix \mathbf{R} and translation vector \mathbf{t} .

3.2. Transformer Model

The Transformer represents a paradigm shift in sequence modeling within the domain of deep learning. Its core concept lies in utilizing self-attention mechanisms for processing sequence data. This mechanism enables the model to dynamically allocate attention weights to various elements within the sequence without needing fixed window sizes or recurrent structures. Such flexibility allows the Transformer model to better capture global and local data information. Additionally, its effectiveness has been demonstrated in

the field of computer vision. The self-attention mechanism comprises scaled dot-product attention and multi-head attention.

Scaled dot-product attention. Within the self-attention mechanism, attention weights are computed by scaling the dot product of the query and key vectors and applying the result to the value vector.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

where Q , K , and V represent the query, key, and value vectors, respectively, with d_k denoting the dimensionality of the keys.

Multi-head attention. Multi-head attention augments the model's representative capacity by parallelly applying multiple queries, key, and value projection sets. The results from multiple attention heads are concatenated and linearly transformed to obtain the final output.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (2)$$

where each attention head head_i is computed as $\text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$ and W_i^Q , W_i^K , W_i^V , and W^O represent the weight matrices for linear transformations.

4. The Proposed Method

Transformation estimation from the corresponding points in two point cloud samples is crucial in point cloud registration. Corresponding point estimation involves identifying the correspondence between points in two point cloud frames corresponding to the same object or scene in the same location. Transformation estimation determines the rotation and translation operations required to align the corresponding positions of two point cloud frames seamlessly. The problem definition, the network architecture, and the loss function are presented and discussed in detail accordingly.

4.1. The Overall Network Architecture

REGTR is an end-to-end network based on Transformer, which can predict the probability of being in the overlapping region for each point in the source point cloud and their corresponding positions in the target point cloud cloud [13]. It could effectively extract the global and local features and perform well in predicting the rigid transformation. To overcome the problem of REGTR not obtaining the essential geometric structure, we propose GraM's overall network architecture that optimizes the REGTR. Specifically, we bind two geometric structure embedded modules on the two self-attention layers in the Transformer cross-coding, respectively. Consequently, they can learn about local features, including information on geometric structure.

The architecture of GraM is devised by embedding the end-to-end point cloud registration network with a geometric structure, as shown in Figure 1. The architecture contains three core modules: (1) Feature extraction module. We utilize the kernel point convolution backbone [32] to extract critical points' features from the source and target point clouds while downsampling the input point cloud. (2) Cross-encoder module. A cross-encoder embedded with a geometric structure receives the features. It utilizes a multi-head self-attention layer to learn features of non-critical points within the point cloud itself and a multi-head cross-attention layer to learn features corresponding to points to be registered. (3) Output module. The output decoder obtains the predicted corresponding critical point positions and transformation matrices between the two point clouds using simple regression operations.

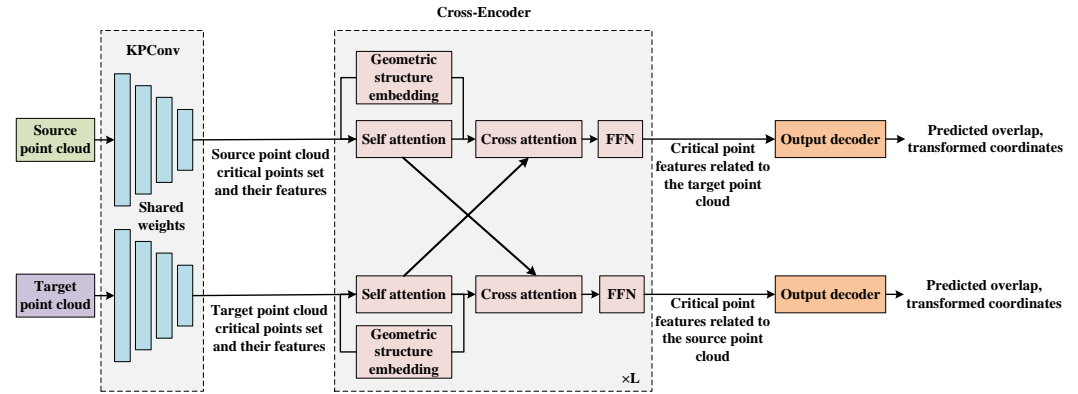


Figure 1. The overall architecture of the proposed GraM. $\times L$ represents multiple cross-encoder network layers.

4.1.1. Feature Extraction

KPCConv (Kernel Point Convolution) exhibits excellent spatial preservation in feature extraction in 3D point cloud data. Based on the ability of spatial preservation, it can handle point cloud data of varying densities and shapes, showcasing outstanding performance across multiple tasks. With the features of preprocessed point cloud data, we optimize the feature extraction process using KPCConv networks. Specifically, we build a double-input KPCConv network with shared weights to extract the same homogeneous characteristics from the original and target point cloud data. It can meet the requirements that they come from the same dataset and can allow swaps. The network iteratively applies a series of residual blocks, including convolution, kernel point convolution, stridden convolution, normalization modules, and LeakyReLU activation functions. Specifically, multiple times of feature extraction and downsampling are firstly performed on the input source point cloud $X \in \mathbb{R}^{M \times 3}$ and target point cloud $Y \in \mathbb{R}^{N \times 3}$, and then, they are transformed into critical point sets $\tilde{X} \in \mathbb{R}^{M' \times 3}$ and $\tilde{Y} \in \mathbb{R}^{N' \times 3}$, along with their features $F_{\tilde{X}} \in \mathbb{R}^{M' \times D}$ and $F_{\tilde{Y}} \in \mathbb{R}^{N' \times D}$.

With the above thought, we apply this feature extraction network separately to the source and target point clouds, obtaining critical points' features for both the source and target point clouds. This process supports the subsequent learning of features in the cross-encoder.

4.1.2. Cross-Encoder

We employ the transformer cross-encoder network to learn features of points in the point cloud and their correspondences with points in the target point cloud. To solve the problem that the output dimensions of different depths from the feature extraction network are diverse, we introduce linear feature projection functions to reduce the dimensions of the outputs before passing them into the cross-encoder. Figure 2 shows the cross-encoder structure, which can extract local and global features. One is the local feature of points outside the key point of a self-point cloud, and the other is the global features that describe the correlation of the two point clouds.

Although the classical Transformer performs sine positional encoding to embed the coordinate information, the coordinate-based encoding is unfixed or invariant. As a result, when executing point cloud registration, the point cloud coordinates change accordingly if different initial poses are used for the same point cloud pair. In this case, coordinate-based coding does not work [33].

In this paper, we replace the sine positional encoding of point clouds in the cross-encoder with geometric structure position encoding. This modification enables the cross-encoder to learn the geometric structure features between critical points before learning self-features and relevant features. It can further improve point cloud registration accuracy. Geometric structure position encoding includes pairwise distance embedding and triangu-

lar embedding. The former represents the distance of the pair of critical points, and the latter is the angle of the triple critical points.

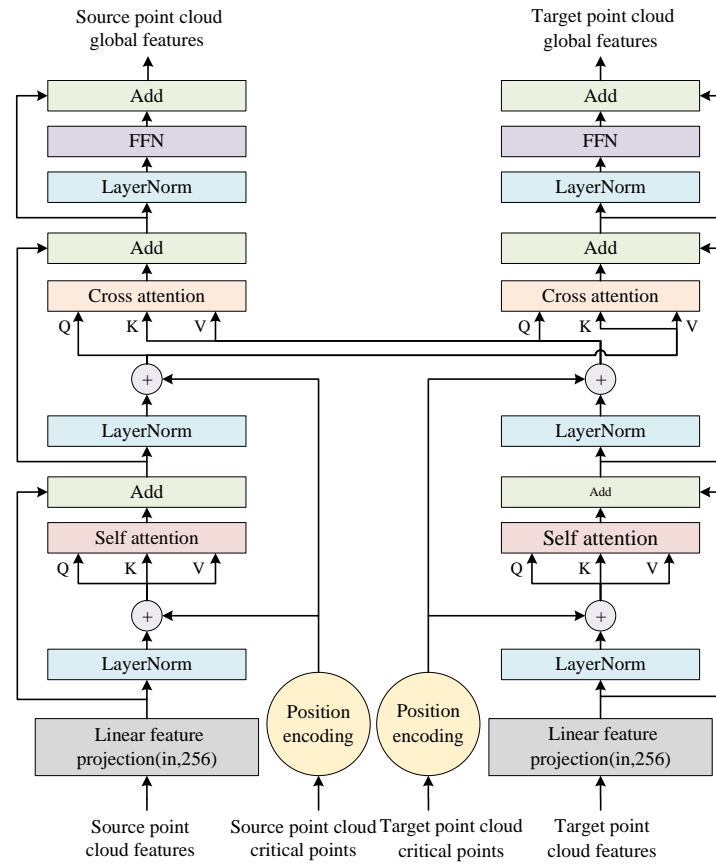


Figure 2. The network architecture of cross-encoder based on geometric structure embedding.

Pairwise distance embedding. We assume that \hat{P}_i and \hat{P}_j are given points and the distance between the two points $d_{i,j} = \|\hat{P}_i - \hat{P}_j\|_2$. The distance formula for pairwise distance embedding $r_{i,j}^D$ should satisfy

$$\begin{cases} r_{i,j,2k}^D = \sin\left(\frac{d_{i,j}/\sigma_d}{10000^{2k/d_t}}\right) \\ r_{i,j,2k+1}^D = \cos\left(\frac{d_{i,j}/\sigma_d}{10000^{2k/d_t}}\right) \end{cases} \quad (3)$$

where d_t is the feature dimension and σ_d is the temperature coefficient controlling the sensitivity to distance changes.

Triangular embedding. The triangular embedding can be calculated using the same method. Assuming that the given angles is $\alpha_{i,j}^k$, the triangular embedding $r_{i,j,k}^A$ can be calculated based on

$$\begin{cases} r_{i,j,k,2x}^A = \sin\left(\frac{\alpha_{i,j}^k/\sigma_a}{10000^{2x/d_t}}\right) \\ r_{i,j,k,2x+1}^A = \cos\left(\frac{\alpha_{i,j}^k/\sigma_a}{10000^{2x/d_t}}\right) \end{cases} \quad (4)$$

where σ_a is another temperature coefficient controlling the sensitivity to angle variations.

4.1.3. Output Decoder

This paper's output decoder differs from the original Transformer's decoder in architecture. Since the cross-encoder has already learned the local and global features, there is no need to use attention mechanisms for decoding. Instead, simple regres-

sion operations are sufficient to estimate the corresponding positional coordinates and transformation matrices.

In estimating corresponding positional coordinates, we use a two-layer MLP to regress the required coordinates. So, the corresponding position of the critical point $\hat{\mathbf{X}}$ of the source point cloud in the target point cloud $\hat{\mathbf{Y}} \in \mathbb{R}^{M' \times 3}$ is

$$\hat{\mathbf{Y}} = \text{ReLU}(\bar{\mathbf{F}}_{\hat{\mathbf{X}}} \mathbf{W}_1 + b_1) \mathbf{W}_2 + b_2 \quad (5)$$

where \mathbf{W}_1 , \mathbf{W}_2 , b_1 , and b_2 are learnable weights and biases, respectively. Similar methods can be employed to obtain the predicted positions $\hat{\mathbf{X}} \in \mathbb{R}^{N' \times 3}$ after receiving the critical points $\hat{\mathbf{Y}}$ of the target point cloud. Simultaneously, we utilize a fully connected layer with the sigmoid activation function to predict overlap confidences $\hat{\mathbf{O}}_{\mathbf{X}} \in \mathbb{R}^{M' \times 1}$ and $\hat{\mathbf{O}}_{\mathbf{Y}} \in \mathbb{R}^{N' \times 1}$. This design eliminates interference from points outside the overlapping region that cannot accurately predict corresponding relationships, significantly improving the accuracy of the transformation matrix estimation. After obtaining the predicted transformation coordinates, the estimation of the transformation matrix can be performed. Connecting the predicted transformation positions for the two point clouds yields a $M' + N'$ dimensional corresponding point set, as shown in Equation (6):

$$\hat{\mathbf{X}}_{corr} = \begin{bmatrix} \tilde{\mathbf{X}} \\ \hat{\mathbf{X}} \end{bmatrix}, \hat{\mathbf{Y}}_{corr} = \begin{bmatrix} \tilde{\mathbf{Y}} \\ \hat{\mathbf{Y}} \end{bmatrix}, \hat{\mathbf{O}}_{corr} = \begin{bmatrix} \hat{\mathbf{O}}_{\mathbf{X}} \\ \hat{\mathbf{O}}_{\mathbf{Y}} \end{bmatrix} \quad (6)$$

where $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ represent sets of critical points and $\hat{\mathbf{X}}$ is the predicted value for the critical points of the target point cloud \mathbf{Y} corresponding to the source point cloud \mathbf{X} , while $\hat{\mathbf{Y}}$ is the predicted value for the critical points of the source point cloud \mathbf{X} corresponding to the target point cloud \mathbf{Y} .

$$\hat{R}, \hat{t} = \underset{R, t}{\operatorname{argmin}} \sum_i^{M'+N'} \hat{o}_i \|R\hat{x}_i + t - \hat{y}_i\|^2 \quad (7)$$

where \hat{x}_i , \hat{y}_i , \hat{o}_i represent i -th rows of matrices $\hat{\mathbf{X}}_{corr}$, $\hat{\mathbf{Y}}_{corr}$, and $\hat{\mathbf{O}}_{corr}$, respectively. R is the rotation transformation matrix, t is the translation transformation vector, and \hat{R} and \hat{t} are the minimum predicted values for R and t satisfying the Equation (7). In this paper, we follow the approach proposed in [21,34] by using a differentiable weighted Kabsch–Umeyama [35,36] algorithm to solve the equation and obtain the rotation matrix and translation vector.

4.2. The Loss Function

Three loss functions for the supervised training of an end-to-end network incorporating attention mechanisms are used in this paper: the feature loss function, the overlap loss function, and the correspondence loss function.

The feature loss. To obtain the geometric structure when calculating the correspondence of the critical point, we apply InfoNCE loss [37] on the features related to both the current point cloud and another point cloud. Considering the correspondence between the critical point set $x \in \tilde{\mathbf{X}}$ of the source point cloud and the critical point set $\tilde{\mathbf{Y}}$ of the target point cloud, the InfoNCE loss for the source point cloud can be described as follows:

$$\mathcal{L}_f^{\mathbf{X}} = -\mathbb{E}_{x \in \tilde{\mathbf{X}}} \left[\log \frac{f(x, p_x)}{f(x, p_x) + \sum_{n_x} f(x, n_x)} \right] \quad (8)$$

We follow the work of Oord et al. [37], where the function $f(\cdot)$ in the Equation (8) is a log-linear model, expressed as follows:

$$f(x, c) = \exp(\bar{f}_x^T \mathbf{W}_f \bar{f}_c) \quad (9)$$

where \bar{f}_x denotes the conditional feature of point x . p_x and n_x denote the sets of critical points in the target point cloud critical point set $\tilde{\mathbf{Y}}$ that match and do not match with x ,

respective, that is, the positive and negative sample sets. These two sets are determined by the margin values of positive and negative samples (r_p, r_n) , where the values of (r_p, r_n) are set to $(m, 2m)$ and m is the voxel distance used in the final downsampling layer of KPConv. All negative sample points falling outside the negative margin are included in the set n_x .

The overlap loss. To calculate the overlap rate of the point cloud and predict the corresponding point relationships, avoiding some redundant work of critical point extraction, we use the binary cross-entropy loss to calculate overlap loss. The expression for the overlap loss function of the source point cloud \mathbf{X} is depicted as follows:

$$\mathcal{L}_o^{\mathbf{X}} = -\frac{1}{M'} \sum_i^{M'} o_{\tilde{x}_i}^* \cdot \log \hat{o}_{\tilde{x}_i} + (1 - o_{\tilde{x}_i}^*) \cdot \log(1 - \hat{o}_{\tilde{x}_i}) \quad (10)$$

To obtain the true value for the overlap labels $o_{\tilde{x}_i}^*$, we employ the approach proposed by Huang et al. [18] to calculate the truth labels for the original dense point cloud. Thus, the truth label for point $\mathbf{X}_i \in X$ is defined as follows:

$$o_{\tilde{x}_i}^* = \begin{cases} 1, & ||\mathcal{T}^*(x_i) - NN(\mathcal{T}^*(x_i), \mathbf{Y})|| < r_o \\ 0, & otherwise \end{cases} \quad (11)$$

where $\mathcal{T}^*(x_i)$ represents the truth transformation matrix $\{R^*, t^*\}$, $NN(\cdot)$ denotes spatial nearest neighbors, and r_o is a predefined overlap threshold. Subsequently, average pooling is employed to obtain the truth overlap labels $o_{\tilde{x}_i}^*$ for the downsampled critical points by using the same parameters as the pooling operation in the downsampling step of KPConv.

The loss $\mathcal{L}_o^{\mathbf{Y}}$ for the target point cloud \mathbf{Y} can be obtained in a similar manner. Thus, the total overlap loss is given by $\mathcal{L}_o = \mathcal{L}_o^{\mathbf{X}} + \mathcal{L}_o^{\mathbf{Y}}$.

The correspondence loss. Matching the main points in the overlapping area is used to calculate the overlapping rate. Therefore, we apply an $\mathcal{L}_c^{\mathbf{X}}$ loss on the predicted transformation matrix for critical points in the overlapping region. The $\mathcal{L}_c^{\mathbf{X}}$ loss for the source point cloud \mathbf{X} is defined as follows:

$$\mathcal{L}_c^{\mathbf{X}} = \frac{1}{\sum_i o_{\tilde{x}_i}^*} \sum_i^{M'} o_{\tilde{x}_i}^* |\mathcal{T}^*(\tilde{x}_i) - \hat{y}_i| \quad (12)$$

The $\mathcal{L}_c^{\mathbf{Y}}$ loss on the target point cloud is similar to $\mathcal{L}_c^{\mathbf{X}}$, and the overall loss for the correspondence is $\mathcal{L}_c = \mathcal{L}_c^{\mathbf{X}} + \mathcal{L}_c^{\mathbf{Y}}$. Therefore, the final loss in this paper is a weighted sum of these three components: $\mathcal{L} = \mathcal{L}_c + \lambda_o \mathcal{L}_o + \lambda_f \mathcal{L}_f$, where $\lambda_o = 1.0$ and $\lambda_f = 0.1$.

5. Experiments

This section presents the dataset, metrics, baselines, experimental setup, main results, and ablation studies. The code is available at <https://github.com/liupin-source/CSR-RegTR> (accessed on 11 May 2024).

5.1. Dataset

We conducted extensive experiments on the representative ModelNet40 and KITTI datasets. To address issues such as insufficient data volume and information in the ModelNet40 dataset and inaccuracies in some truth labels in the KITTI dataset, we reconstructed a dataset more suitable for 3D point cloud registration tasks.

ModelNet40. ModelNet40 is a subset of the ModelNet dataset built by Princeton University and includes 40 types of point cloud data. We directly sampled the initial ModelNet40 point cloud dataset [38] twice at complete random to generate the source and target point clouds, which do not have precisely corresponding points. Then, we selected 4096 points in each sampling. Finally, we applied segmentation operations to the point cloud data, which can generate datasets with overlap rates of 70% and 50%. These two datasets are named ModelNet40 and LowModelNet40, respectively, in which sample point cloud data are shown in Figures 3 and 4.

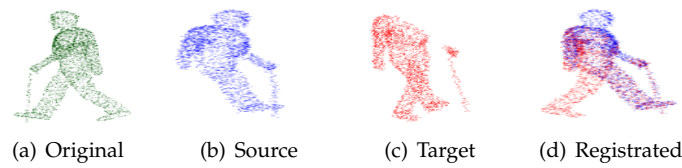


Figure 3. Example from the ModelNet40 dataset.

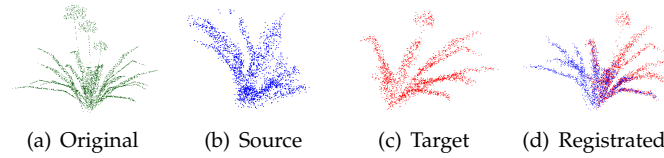


Figure 4. Example from the LowModelNet40 dataset.

KITTI. KIT (Karlsruhe Institute of Technology) and TTI-C (Toyota Technological Institute at Chicago) jointly founded the dataset and obtained data from the collection vehicle equipped with a Velodyne lidar with 0.09 degrees resolution. KITTI contains multiple datasets, such as 3D object detection and visual ranging. We only use point cloud data for 3D registration tasks. To address the problem that some truth labels in the KITTI dataset [39] are not accurate, we perform a manual matching to calibrate the truth labels. Because of the large scale of the KITTI dataset, we employed voxel filtering with a grid size of 0.3 m for downsampling to preserve the density of the 3D point cloud after subsampling. An example of the preprocessed KITTI point cloud dataset is shown in Figure 5.

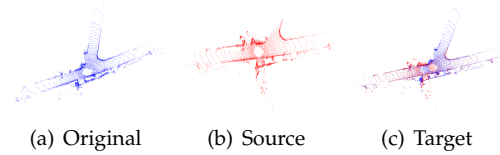


Figure 5. Example from the KITTI dataset.

5.2. Evaluation Metrics

Relative rotation error (RRE). RRE is the degree difference between the predicted rotation matrix and the actual rotation matrix used to measure the error of the rotation matrix.

$$RRE = \cos^{-1} \left(\frac{1}{2} \text{trace}(R^T \bar{R} - 1) \right) \quad (13)$$

where R represents the true rotation matrix and \bar{R} represents the predicted rotation matrix.

Relative translation error (RTE). RTE refers to the Euclidean distance between the predicted translation vector and the actual translation vector, serving to quantify the error in the translation vector.

$$RTE = \|t - \bar{t}\|_2 \quad (14)$$

where t represents the true translation matrix and \bar{t} represents the predicted translation matrix.

Registration recall (RR). RR measures the accuracy of point cloud registration algorithms predicting the transformation matrix. The larger the value, the higher the accuracy of the transformation matrix. RR refers to the average ratio of correspondences correctly matched in the overlapping region to the total ones. This correct match occurs when the source point cloud is registered with the target point cloud using the predicted transformation matrix.

$$RR = \frac{1}{M} \sum_{i=1}^M \left[\sqrt{\frac{1}{|e^*|} \sum_{(p_{x_i}^*, q_{y_i}^*) \in e^*} \|\hat{T} p_{x_i}^* - q_{y_i}^*\|_2^2} < \tau_3 \right] \quad (15)$$

where e^* represents point pairs in the true labels with corresponding relationships and $(p_{x_i}^*, q_{y_i}^*)$ denotes a pair of true corresponding points. $\hat{T} \in SE(3)$ represents the predicted transformation matrix. Additionally, τ_3 represents the error threshold between the predicted and true values.

5.3. Baselines

3DFeatNet [20]: A network for learning feature correspondences from 3D point clouds using weak supervision methods.

RPMNet [21]: A deep learning-based point cloud registration method that is less sensitive to initialization and more robust.

DCP [22]: A learning-based approach that includes a point cloud feature extraction network, point cloud matching prediction based on attention mechanisms, and a differentiable singular value decomposition layer.

PointNetLK [27]: A 3D point cloud registration method that combines PointNet with the LK algorithm.

REGTR [13]: An end-to-end 3D point cloud registration network that utilizes attention mechanisms.

DGR [29]: A differentiable network architecture designed for actual point cloud data.

Predator [18]: A point cloud registration method explicitly designed to handle low overlap scenarios.

5.4. Experimental Setup

For the ModelNet40 and LowModelNet40 datasets, their training sets included 6316 pairs of point cloud data. The former's test set contained 5995 pairs of point cloud data, and the latter's test set contained 12,311 pairs of point cloud data. The convolution radius of KPConv is 2.75, and the initial sampling radius is 0.0375. During the training process, we trained the network using the AdamW [40] optimizer with an initial learning rate of 0.0001 and weight decay of 0.0001. The training epochs were 80, and each training iteration was verified in the validation set. For the KITTI dataset, the training and test sets contained 1358 and 555 pairs of point cloud data, respectively. The convolution radius of KPConv was 4.5, and the initial sampling radius was 0.3. The optimizer settings were consistent with the ModelNet40 dataset. The training consisted of 200 epochs, with a learning rate decay of 0.5 every 50 epochs. After each training iteration, the network was tested over the validation set.

5.5. Convergence Analysis

The core idea of GraM is to extract the global and local features containing geometric structure information to improve the importance of the 3D point cloud. To analyze whether it can achieve the above goal, we recorded the losses, creating feature-loss, overlap-loss, and correspondence-loss curves on the ModelNet40 and KITTI datasets, as shown in Figure 6. On the one hand, all loss curves quickly converge (i.e., 10 epochs on the ModelNet40 and 40 epochs on the KITTI), which shows that GraM can quickly position and learn the features related to geometric structure information. On the other hand, all the loss curves are relatively stable, without oscillation. This indicates that the loss function we designed can accurately restrict the limited conditions for each feature learning.

5.6. Comparison with State-of-the-Art Methods

To verify the superiority of GraM's performance, we compare it with those of feature learning-based methods RPMNet and DCP, as well as end-to-end methods PointNetLK, basic REGTR, etc. Table 1 shows the experimental results of the ModelNet40, LowModelNet40, and KITTI datasets on RRE, RTE, and RR evaluation metrics. The results in Table 1 indicate that our method, GraM, is slightly advantageous compared to basic REGTR. However, GraM has a significant advantage over DCP and PointNetLK. The main reason is that processing the ModelNet40 and LowModelNet40 datasets involves only quantitative

changes. Significantly, the number of point clouds increases, allowing the network to capture as much sufficient information as possible. In conclusion, our algorithm shows a certain degree of performance improvement effect, as indicated by the registration results shown in Figures 7–9.

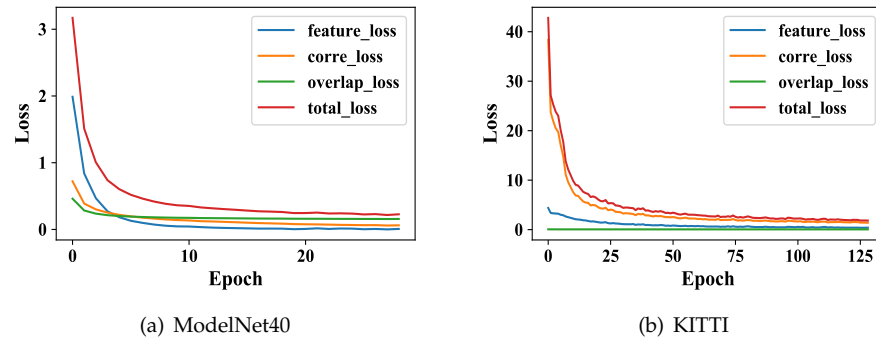


Figure 6. The convergence situation of our GraM on ModelNet40 and KITTI datasets; feature_loss, corre_loss, overlap_loss, and total_loss represent feature loss, overlap loss, correspondence loss, and their total loss, respectively.

Table 1. The comparison of GraM to the state-of-the-art approaches, where the best results are in bold.

Method	ModelNet40		LowModelNet40		KITTI	
	RRE (°)	RTE (m)	RRE (°)	RTE (m)	RRE (°)	RTE (m)
RPMNet	1.712	0.018	7.342	0.124	1.021	0.633
DCP	11.975	0.171	16.501	0.300	0.965	0.583
PointNetLK	29.725	0.297	48.567	0.507	2.352	0.936
REGTR	1.473	0.014	3.930	0.087	0.482	0.425
3DFeatNet	2.057	0.039	4.026	0.073	0.254	0.259
Predator	1.948	0.026	3.568	0.072	0.277	0.068
DGR	2.004	0.024	3.627	0.069	0.373	0.320
GraM	0.925	0.010	2.653	0.049	0.270	0.110

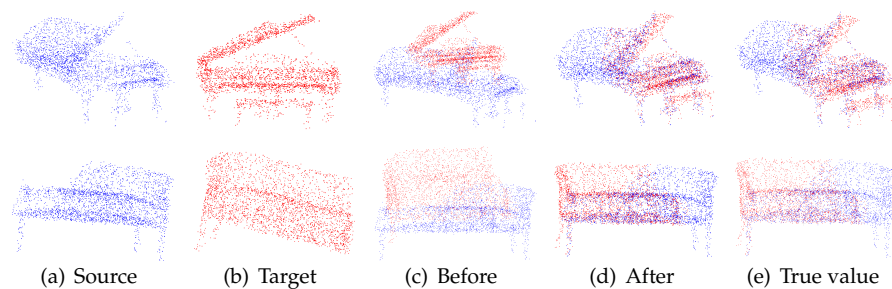


Figure 7. Example of a registration result using GraM on the ModelNet40 dataset.

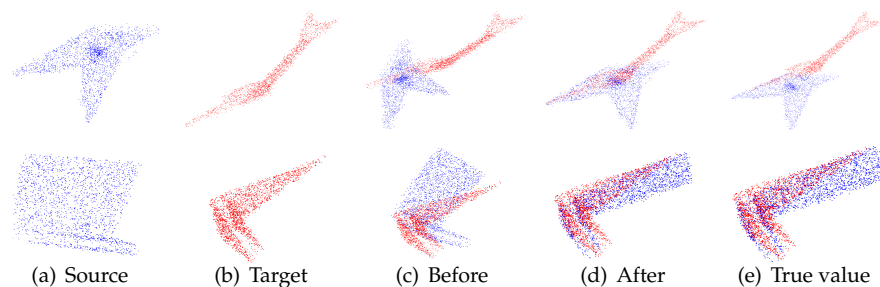


Figure 8. Example of a registration result using GraM on the LowModelNet40 dataset.

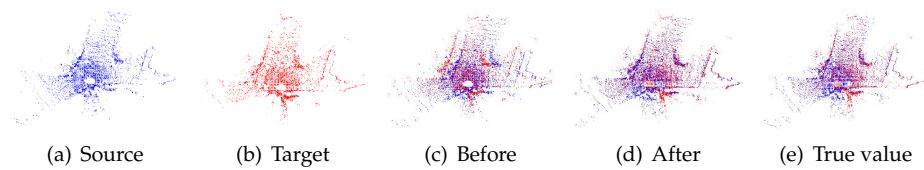


Figure 9. Example of a registration result using GraM on the KITTI dataset.

5.7. Analysis Sensitivity of Sampling Radius

Downsampling is a critical procedure of 3D point cloud data processing in KPConv. The sampling radius is an essential parameter of the process. The appropriate sampling radius can effectively reduce the scale of the point cloud data to facilitate feature learning of subsequent network structures. To analyze the sensitivity and effectiveness of the sampling radius, we conducted experiments using GraM with different sampling radii. Table 2 displays three metrics' performance results and the model training's time consumption on the KITTI dataset. From the performance results in Table 2, we can observe that a large sampling radius does not achieve the high accuracy of our geometrically embedded 3D point cloud registration algorithm because a massive sampling radius may overlook critical point information, preventing the cross-encoding network from learning sufficient features. In particular, the model training time decreases as the sampling radius increases. The core reason is that the small sampling radius can make the data too large, which causes many network parameters and eventually increases model training time.

Table 2. Experiment results using GraM with different sampling radii on the KITTI dataset. TC represents the time consumption (hours) of model training. The best results are in bold.

Method	KITTI			
	RRE (°)	RTE (m)	RR (%)	TC (h)
radius-0.4	0.352	0.214	97.3	4.27
radius-0.5	0.413	0.325	96.1	6.24
radius-0.3	0.270	0.110	99.8	3.51

5.8. Ablation Studies

The following subsections discuss the ablation studies, including the effectiveness of each component of GraM and performances with different loss functions.

5.8.1. Effectiveness of GraM's Each Component

GraM takes the architecture of the REGTR as a carrier and introduces the feature extraction module (KPConv) and geometric structure embedding (GSE) module of the shared weight. To analyze the effectiveness of these two modules on GraM's performance, we carried out an ablation study and recorded the results on ModelNet40 and KITTI datasets, as shown in Table 3. Overall, our final GraM (REGTR+KPConv+GSE) achieved optimal performance on all metrics. The table also shows that the two modules we introduce can effectively improve the performance of 3D point cloud registration. From the perspective of individual modules on performance effects, compared with GSE-based (i.e., ‡ relative to †) performance improvement and KPConv-based (i.e., † relative to *) performance improvement, three of the four metrics achieved the optimal result. This illustrates that the contribution of geometric structure embedding rather than KPConv to performance improvement is more significant. The results further verify that geometric structure embedding can effectively extract valuable geometric structure information to 3D points cloud registration.

Table 3. Registration result using different components on ModelNet40 and KITTI datasets. GSE indicates the geometric structure embedding network used to extract geometric structure information. The best results are in bold. *, †, and ‡ refer to the baseline, GraM, and our final GraM. ↓ means the RRE and RTE are reduced.

Method	ModelNet40		KITTI	
	RRE (°)	RTE (m)	RRE (°)	RTE (m)
Baseline (REGTR) *	1.473	0.014	0.482	0.425
Our GraM (REGTR+KPConv) †	1.248	0.013	0.324	0.301
Our Final GraM (REGTR+KPConv+GSE) ‡	0.925	0.010	0.270	0.110
‡ relative to *	0.225↓	0.001↓	0.158↓	0.124↓
‡ relative to †	0.323↓	0.003↓	0.054↓	0.191↓

5.8.2. Effectiveness of GraM with Different Loss Functions

We conducted experiments using GraM with different combinations of loss functions and recorded the results in Table 4. We can observe from Table 4 that any one or any two of the three loss functions cannot achieve satisfactory accuracy. It is worth noting that the algorithm can achieve the best performance only when all three loss functions are used simultaneously for network training. Judging from the analysis of a single loss function, comparing GraM with $(\mathcal{L}_c + \mathcal{L}_f)$ to GraM with $(\mathcal{L}_c + \mathcal{L}_o)$, the former achieves four maximum values in the four values, which shows that A is more advantageous than performance improvement (i.e., $\mathcal{L}_f > \mathcal{L}_o$). Similarly, $\mathcal{L}_c > \mathcal{L}_o$ and $\mathcal{L}_c > \mathcal{L}_o$. Therefore, we can sort the contribution of the three loss functions to the point cloud distribution performance as follows: $\mathcal{L}_f > \mathcal{L}_c > \mathcal{L}_o$.

Table 4. Registration result of GraM using different loss functions on ModelNet40 and KITTI dataset. The best results are in bold.

Method	ModelNet40		KITTI		
	RRE (°)	RTE (m)	RRE (°)	RTE (m)	RR(%)
Baseline (\mathcal{L}_c loss in Equation (12))	2.442	0.020	0.302	0.174	98.9
Our GraM ($\mathcal{L}_o + \mathcal{L}_f$)	2.206	0.016	0.345	0.142	98.6
Our GraM ($\mathcal{L}_c + \mathcal{L}_f$)	2.125	0.015	0.342	0.139	99.1
Our GraM ($\mathcal{L}_c + \mathcal{L}_o$)	2.241	0.017	0.351	0.153	98.0
Our Final GraM ($\mathcal{L}_c + \mathcal{L}_o + \mathcal{L}_f$)	1.623	0.013	0.270	0.110	99.8

6. Conclusions

This paper proposes a 3D point cloud registration method, GraM, which embeds the geometric structure into the attention mechanism to form an end-to-end registration framework. The framework can effectively extract local and global features containing the geometric structure information. With this feature, simple regression is enough to obtain the corresponding position coordinates and transformation matrix, thereby improving the registration accuracy of the point cloud. Extensive experiments show that the proposed method is far superior to existing state-of-the-art methods. In the future, we will explore techniques to achieve better registration accuracy for large-scale point cloud datasets with low overlap rates using lightweight models.

Author Contributions: Methodology, P.L.; Software, X.Z.; Formal analysis, L.Z.; Writing—review and editing, R.W. and J.Z. (Juan Zhang); Visualization, J.Z. (Jianyong Zhu); Supervision, R.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Fundamental Research Funds for the Central Universities (No. 2-9-2022-062).

Data Availability Statement: Data not available due to commercial restrictions. Due to the nature of this research, participants of this study did not agree for their data to be shared publicly, so supporting data is not available.

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

References

1. Azuma, R.T. A survey of augmented reality. *Presence Teleoperators Virtual Environ.* **1997**, *6*, 355–385. [\[CrossRef\]](#)
2. Carmigniani, J.; Furht, B.; Anisetti, M.; Ceravolo, P.; Damiani, E.; Ivkovic, M. Augmented reality technologies, systems and applications. *Multimed. Tools Appl.* **2011**, *51*, 341–377. [\[CrossRef\]](#)
3. Billinghurst, M.; Clark, A.; Lee, G. A survey of augmented reality. *Now* **2015**, *8*, 73–272.
4. Liu, D.; Long, C.; Zhang, H.; Yu, H.; Dong, X.; Xiao, C. ARShadowGAN: Shadow generative adversarial network for augmented reality in single light scenes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 8139–8148.
5. Popișter, F.; Popescu, D.; Păcurar, A.; Păcurar, R. Mathematical Approach in Complex Surfaces Toolpaths. *Mathematics* **2021**, *9*, 1360. [\[CrossRef\]](#)
6. Luo, K.; Yang, G.; Xian, W.; Haraldsson, H.; Hariharan, B.; Belongie, S. Stay Positive: Non-Negative Image Synthesis for Augmented Reality. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 10050–10060.
7. Joseph, K.; Khan, S.; Khan, F.S.; Balasubramanian, V.N. Towards open world object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 5830–5840.
8. Merickel, M. 3D reconstruction: The registration problem. *Comput. Vis. Graph. Image Process.* **1988**, *42*, 206–219. [\[CrossRef\]](#)
9. Izadi, S.; Kim, D.; Hilliges, O.; Molyneaux, D.; Newcombe, R.; Kohli, P.; Shotton, J.; Hodges, S.; Freeman, D.; Davison, A.; et al. Kinectfusion: Real-time 3D reconstruction and interaction using a moving depth camera. In Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, Santa Barbara, CA, USA, 16–19 October 2011; pp. 559–568.
10. Pan, X.; Xia, Z.; Song, S.; Li, L.E.; Huang, G. 3D Object detection with pointformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 7463–7472.
11. Shi, X.; Ye, Q.; Chen, X.; Chen, C.; Chen, Z.; Kim, T.K. Geometry-based distance decomposition for monocular 3d object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 15172–15181.
12. Zou, Z.; Ye, X.; Du, L.; Cheng, X.; Tan, X.; Zhang, L.; Feng, J.; Xue, X.; Ding, E. The devil is in the task: Exploiting reciprocal appearance-localization features for monocular 3d object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 2713–2722.
13. Yew, Z.J.; Lee, G.H. REGTR: End-to-end point cloud correspondences with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 6677–6686.
14. Besl, P.J.; McKay, N.D. Method for registration of 3-D shapes. *Proc. SPIE* **1992**, *1611*, 586–606. [\[CrossRef\]](#)
15. Billings, S.D.; Bector, E.M.; Taylor, R.H. Iterative most-likely point registration (IMLP): A robust algorithm for computing optimal shape alignment. *PLoS ONE* **2015**, *10*, e0117688. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Segal, A.; Haehnel, D.; Thrun, S. Generalized-ICP. In Proceedings of the Robotics: Science and Systems, Seattle, WA, USA, 28 June–1 July 2009; Volume 2, p. 435.
17. Zhu, H.; Guo, B.; Zou, K.; Li, Y.; Yuen, K.V.; Mihaylova, L.; Leung, H. A review of point set registration: From pairwise registration to groupwise registration. *Sensors* **2019**, *19*, 1191. [\[CrossRef\]](#)
18. Huang, S.; Gojcic, Z.; Usvyatsov, M.; Wieser, A.; Schindler, K. Predator: Registration of 3d point clouds with low overlap. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 4267–4276.
19. Zeng, A.; Song, S.; Nießner, M.; Fisher, M.; Xiao, J.; Funkhouser, T. 3DMatch: Learning local geometric descriptors from rgb-d reconstructions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1802–1811.
20. Yew, Z.J.; Lee, G.H. 3DFeat-Net: Weakly supervised local 3d features for point cloud registration. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 607–623.
21. Yew, Z.J.; Lee, G.H. RPM-Net: Robust point matching using learned features. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11824–11833.
22. Wang, Y.; Solomon, J.M. Deep closest point: Learning representations for point cloud registration. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3523–3532.
23. Wang, H.; Liu, Y.; Dong, Z.; Wang, W. You only hypothesize once: Point cloud registration with rotation-equivariant descriptors. In Proceedings of the 30th ACM International Conference on Multimedia, Lisboa, Portugal, 10–14 October 2022; pp. 1630–1641.
24. Zhang, Y.; Zhang, W.; Li, J. Partial-to-partial point cloud registration by rotation invariant features and spatial geometric consistency. *Remote Sens.* **2023**, *15*, 3054. [\[CrossRef\]](#)

25. Liu, Q.; Zhu, H.; Zhou, Y.; Li, H.; Chang, S.; Guo, M. Density-invariant features for distant point cloud registration. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 18215–18225.
26. Deng, H.; Birdal, T.; Ilic, S. 3D local features for direct pairwise registration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 3244–3253.
27. Aoki, Y.; Goforth, H.; Srivatsan, R.A.; Lucey, S. PointNetLK: Robust & efficient point cloud registration using pointnet. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 7163–7172.
28. Baker, S.; Matthews, I. Lucas-kanade 20 years on: A unifying framework. *Int. J. Comput. Vis.* **2004**, *56*, 221–255. [[CrossRef](#)]
29. Choy, C.; Dong, W.; Koltun, V. Deep global registration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 2514–2523.
30. Choy, C.; Park, J.; Koltun, V. Fully convolutional geometric features. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8958–8966.
31. Yu, H.; Hou, J.; Qin, Z.; Saleh, M.; Shugurov, I.; Wang, K.; Busam, B.; Ilic, S. Riga: Rotation-invariant and globally-aware descriptors for point cloud registration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 3796–3812. [[CrossRef](#)] [[PubMed](#)]
32. Thomas, H.; Qi, C.R.; Deschaud, J.E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. KPConv: Flexible and deformable convolution for point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6411–6420.
33. Qin, Z.; Yu, H.; Wang, C.; Guo, Y.; Peng, Y.; Ilic, S.; Hu, D.; Xu, K. GeoTransformer: Fast and Robust Point Cloud Registration With Geometric Transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 9806–9821. [[CrossRef](#)] [[PubMed](#)]
34. Gojcic, Z.; Zhou, C.; Wegner, J.D.; Guibas, L.J.; Birdal, T. Learning multiview 3d point cloud registration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1759–1769.
35. Kabsch, W. A solution for the best rotation to relate two sets of vectors. *Acta Cryst.* **1976**, *32*, 922–923. [[CrossRef](#)]
36. Umeyama, S. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 376–380. [[CrossRef](#)]
37. van den Oord, A.; Li, Y.; Vinyals, O. Representation learning with contrastive predictive coding. *arXiv* **2018**, arXiv:1807.03748.
38. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3D ShapeNets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1912–1920.
39. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Ind. Robot.* **2013**, *32*, 1231–1237. [[CrossRef](#)]
40. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.