LoGDesc: Local geometric features aggregation for robust point cloud registration

Karim Slimani, Brahim Tamadazte, Catherine Achard

LoGDesc: Local geometric features aggregation for robust point cloud registration

Karim Slimani[1 [0009−0007−4791−7173]], Brahim Tamadazte[1 [0000−0002−4668−3092]], and Catherine Achard[1 [0000−0002−5790−0830]]

ISIR, Sorbonne Université, CNRS UMR 7222, INSERM U1150, 4 place Jussieu, Paris, France
Corresponding Author: karim.slimani@isir.upmc.fr

Abstract. This paper introduces a new hybrid descriptor for 3D point matching and point cloud registration, combining local geometrical properties and learning-based feature propagation for each point's neighborhood structure description. The proposed architecture first extracts prior geometrical information by computing each point's planarity, anisotropy, and omnivariance using a Principal Components Analysis (PCA). This prior information is completed by a descriptor based on the normal vectors estimated thanks to constructing a neighborhood based on triangles. The final geometrical descriptor is propagated between the points using local graph convolutions and attention mechanisms. The new feature extractor is evaluated on *ModelNet40*, *Bunny Stanford* dataset, *KITTI* and MVP (Multi-View Partial)-RG for point cloud registration and shows interesting results, particularly on noisy and low overlapping point clouds. *The code will be released after publication.*

Keywords: Geometric registration · Point cloud learning · Feature extraction · Pose estimation

1 Introduction

The recent development of new generations of three-dimensional sensors, such as LIDAR and Kinect, offers many applications in computer vision, robotics, 3D printing, graphics, etc. Unlike traditional visual sensors (cameras), these sensors provide a 3D representation of the observed scene in the form of a 3D point set called a point cloud, providing important cues for analyzing objects and environments [8, 26]. These technological advances have enabled us to overcome the recurrent problem of directly estimating scene depth that characterizes traditional visual sensors. The field of view of 3D sensors is relatively limited in many applications, such as robotic navigation, autonomous vehicles, and manipulation tasks. As a result, several scientific questions have emerged relating to 3D reconstruction [35], mapping [10], object or scene recognition [28], pose estimation [1], medical imaging [2], etc. Nevertheless, *3D point registration*, also known as scan matching or point cloud alignment, is undoubtedly the problem that has received the most interest from the robotics, graphics, and computer

vision communities [19]. A typical registration problem involves aligning two or more point clouds in a three-dimensional coordinate system acquired from different viewpoints into a unified coordinate system. This means finding the optimal spatial transformation (rotation, translation, and potentially scale) that aligns the given point clouds. The registration problem can be tackled from the optimization angle, opening the way to so-called iterative methods. Most of the existing registration methods are formulated by minimizing a geometric projection error through two processes: correspondence searching and transformation estimation, which are repeated until convergence is reached; this is the case in the two most used registration methods, ICP (Iterative Closest Point) [4] and RANSAC for instance. For a long time, these methods were the benchmark in point cloud registration, thanks to their efficiency and simplicity of implementation. However, their performances can deteriorate under unfavorable conditions, often leading them to converge on a local minimum due to non-convexity.

To overcome these limitations, features-based methods attempt to extract locally significant and robust geometric descriptors. These descriptors can be either local (extracted from the interesting part of the point cloud) [42], global (generated by encoding the geometric information of the whole point cloud) [48] or hybrid, combining local and global descriptors) [3]. Each descriptor has advantages and disadvantages, depending on the point clouds to be aligned, but none is precise, robust, and versatile.

2 Related Work and Contributions

The emergence of deep learning across multiple disciplines has likewise enhanced 3D registration techniques in several aspects. Among the contributions of deep learning, the development of learned descriptors has brought significant progress in point cloud registration [5]. The feature learning approach, PRNET [37], is designed to extract features invariant to rigid transformations, addressing the unordered nature of point clouds and leveraging local surface characteristics. Similarly, R-PointHop [13] extracts the descriptors by utilizing various neighborhood sizes facilitated by a Local Reference Frame (LRF) established for each point through its nearest neighbors. To achieve rotation invariance, 3DSmoothNet [9] projects a voxelized smoothed density value representation to the LRF before feeding it to 3D CNNs. These approaches have demonstrated promising outcomes on geometric registration benchmarks. However, transforming points to voxels can incur substantial computational expenses, particularly with dense point clouds. Besides, PointNet [20] processes the point cloud by mapping it into a learned canonical space and employing a symmetric function to ensure the output remains unaffected by the input points' order. The popular DGCNN [38] method enhances local property captured between points by implementing successive convolutional layers. It constructs input graphs based on each point's K Nearest Neighbors that are dynamically refreshed at each layer.

Likewise, learning hybrid descriptors (combining local and global information) is possible. Recent approaches incorporate local and global features into

a transformer module, as in DCP [36]. These models enhance traditional transformers by integrating additional geometric information, including pairwise distances and triplet angles, as exemplified by GeoTransformer [21].

RoCNet++ [30] investigated a new descriptor that encodes the local geometric properties of the surface, *i.e.*, each point is characterized by all the triangles formed by itself and its nearest neighbors. Casspr [40]investigated cross-attention transformers fusing point-wise and sparse voxel features to capture information at low and fine resolutions. Soe-net [41] introduced the *PointOE* module to capture local structures by analyzing patterns from multiple spatial orientations.

In this paper, we present a new hybrid descriptor *LoGDesc* for point cloud registration, exploiting geometric properties given the very local structure of the points and enhancing their robustness to noise by feeding them to learning modules. The first main contribution is using the normals to the planes containing the triangles formed by each point and its nearest neighbors to estimate a single normal vector on each point by weighting each normal's support by the triangle area's function. Secondly, a local PCA (Principal Component Analysis) allows computing a Local Reference Frame (LRF), the anisotropy (A), the omnivariance (O), and the planarity (P) for each point. The last three functions complete the 3D coordinates to make a robust first vector of features. Then, the previously estimated normals are projected in the LRFs to ensure rotation-invariant descriptors. Finally, the information in each point descriptor is propagated locally to globally thanks to KNN-based graphs followed by a *self-attention* mechanism.

3 Problem Statement and Proposed Method

3.1 Problem Statement

Let us define two point clouds $\boldsymbol{X}$ and $\boldsymbol{Y}$ such that $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_i, ..., \boldsymbol{x}_M\} \subset \mathbb{R}^{3\times M}$ and $\boldsymbol{Y} = \{\boldsymbol{y}_1, ..., \boldsymbol{y}_j, ..., \boldsymbol{y}_N\} \subset \mathbb{R}^{3\times N}$, where each $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ are the 3D coordinates of the i^{th} and j^{th} points in the source and destination point clouds, respectively. Suppose the two point clouds are at least partially overlapping, *i.e.*, there are C pairs of matches between $\boldsymbol{X}$ and $\boldsymbol{Y}$, forming two sets of matches $\bar{\boldsymbol{X}} \subset \mathbb{R}^{3\times C}$ and $\bar{\boldsymbol{Y}} \subset \mathbb{R}^{3\times C}$, where $C \leq min(M, N)$.

The purpose of point cloud registration is to estimate the rigid transformation matrix $\mathbf{T}_{XY} = \begin{bmatrix} \mathbf{R}_{xy} & \mathbf{t}_{xy} \\ 0_{1\times 3} & 1 \end{bmatrix}$ which minimizes the Euclidean distance between $\bar{\boldsymbol{X}}$ and the transformed $\bar{\boldsymbol{Y}}$:

$$\mathbf{R}_{xy}, \mathbf{t}_{xy} = \underset{\mathbf{R}^*{}_{xy}, \mathbf{t}^*{}_{xy}}{argmin} \left(\mathrm{d}\big(\bar{\boldsymbol{X}}, \mathbf{R}^*{}_{XY}\bar{\boldsymbol{Y}} + \mathbf{t}^*{}_{xy}\big) \right) \tag{1}$$

with $\mathbf{R}^*{}_{xy} \in SO(3)$ being all admissible 3×3 matrices, $\mathbf{t}^*{}_{xy} \in \mathbb{R}^3$ all admissible 3D vectors and d the Euclidean distance between the two sets of paired points which can be formulated by: $\mathrm{d}\big(\bar{\boldsymbol{X}}, \mathbf{R}^*{}_{XY}\bar{\boldsymbol{Y}} + \mathbf{t}^*{}_{xy}\big) = \sum_{c=1}^{C} \|(x_c - \mathbf{R}^*{}_{xy} \cdot y_c + \mathbf{t}^*{}_{xy})\|_2$.

3.2 Registration overall architecture

The hybrid feature extractor proposed in this paper is assessed in the point cloud registration challenge by integrating it in an architecture inspired by the state-of-the-art methods: [31, 30, 27], as depicted in Figure 1. The overall architecture is described in the following subsections.

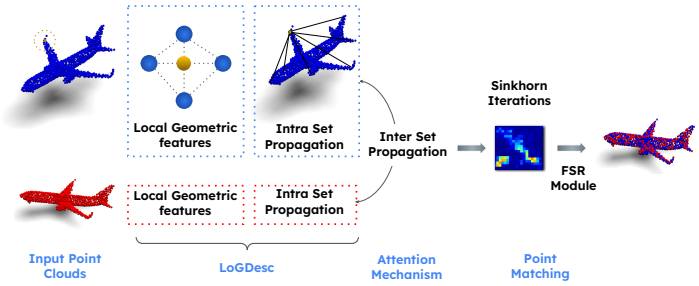

Fig. 1. Summary of the point cloud registration method.

Feature extraction The first step of the algorithm is to extract the feature vectors $\boldsymbol{f}_i \in \mathbb{R}^d$ and $\boldsymbol{h}_i \in \mathbb{R}^d$ for each source and target point, respectively. This is a crucial step in learning point cloud registration methods since the accuracy of the estimated rigid transformation explicitly depends on the accuracy of the matching process, which in turn depends on the similarity between the corresponding pairs of points from the two point clouds. To this end, we propose a new descriptor called *LoGDesc*. The main principle of *LoGDesc* is to use local geometric properties to capture the patterns of the local structure of each point and to learn to propagate this local information globally within each point cloud using attention mechanisms. The section 3.3 details the proposed feature extraction and aggregation.

Normal encoder attention mechanism Once the features are extracted using *LoGDesc*, we propose propagating their information within each point cloud and between the two point clouds using a geometric transformer. This allows the algorithm to learn an inter-sets contextual understanding of these features. To do so, we propose to adapt the normal encoder transformer reported in RoCNet [31] which enhances the feature vectors by successive *self-attention* and *cross-attention* layers [34] to get the final features $\tilde{\boldsymbol{f}}_i \in \mathbb{R}^d$ and $\tilde{\boldsymbol{h}}_i \in \mathbb{R}^d$. As vanilla transformers used in 3D point clouds may omit geometric structure encoding, authors proposed to feed the transformer with transformation-invariant geometric information. To this aim, for each point, the relative orientation of its surface normal and all the other points of the same set is encoded thanks to sinusoidal functions inspired by the positional encoding from [34] and [21]. Noting

$\boldsymbol{n}_i$ the surface normal estimated on the position $\boldsymbol{x}_i$, the geometric information between each pair of points is embedded in a vector $\boldsymbol{r}_{i,j} \in \mathbb{R}^d$ as follows:

$$\boldsymbol{r}_{i,j}^{2p} = \sin\left(\frac{\angle(\boldsymbol{n}_i, \boldsymbol{n}_j)}{\sigma \times u^{2p/d}}\right), \qquad \boldsymbol{r}_{i,j}^{2p+1} = \cos\left(\frac{\angle(\boldsymbol{n}_i, \boldsymbol{n}_j)}{\sigma \times u^{2p/d}}\right) \tag{2}$$

where p is the dimension index in $\boldsymbol{r}_{i,j}$, $\sigma = \frac{15 \times \pi}{180}$ is a normalization coefficient to limit the sensitivity of normals orientation, and u a constant empirically defined as 10000. The embedding $\boldsymbol{r}_{i,j}$ is used in the *self-attention* scores computation next to point features vectors, as detailed in [31].

Matching module Once the final features $\tilde{\boldsymbol{f}}_i \in \mathbb{R}^d$ and $\tilde{\boldsymbol{h}}_i \in \mathbb{R}^d$ built, pairwise correspondences must be estimated between the two point clouds. To this aim, a similarity matrix $\mathbf{S} \in \mathbb{R}^{M \times N}$ is obtained by a dot product between the feature vectors, with:

$$\mathbf{S}_{i,j} = \left\langle \tilde{\boldsymbol{f}}_i, \tilde{\boldsymbol{h}}_j \right\rangle \text{ with } 1 \leq i \leq M,\ 1 \leq j \leq N \tag{3}$$

This last matrix is incrementally optimized by a differentiable transport algorithm [29]. Collecting the mutual best scores along each row and each column from the final score matrix $\tilde{\mathbf{S}} \in \mathbb{R}^{M \times N}$ allows constructing a binary assignment between the point clouds [27].

Transformation estimation The final step of the algorithm is the rigid transformation estimation, which can be computed using a simple Singular Values Decomposition (SVD) of the matched points covariance matrix or by a RANSAC. In this paper, to compare *LoGDesc* performance with the state-of-the-art methods, we use the farthest sampling-guided registration (FSR) module introduced in RoCNet++ [30] as our pose estimator, its pseudocode is detailed in the pseudocode 1. This is mainly motivated by the fact that it offers a good compromise between robustness to noise and computation cost [30]. Besides this, to highlight that the proposed feature contributes to the accuracy independently of the pose estimator used, we also compute the rigid transformation using FGR [48] and RANSAC [6], both with their version based on feature matching (*i.e.*, without using the matching module 3.2). We conduct this experiment on the challenging *MVP-RG* [18] dataset and report the results on Table 6.

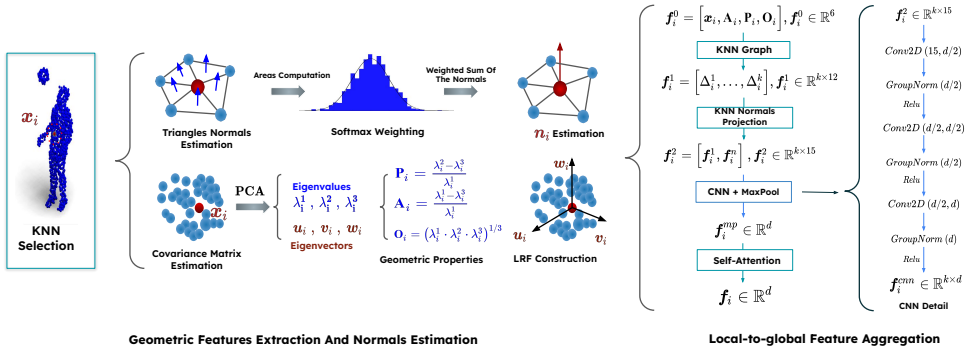

Fig. 2. Overview of the proposed geometrical descriptor. KNN of each point are collected to build local sub-samples. A PCA is applied on these subsamples to extract geometric properties **P**, **A** and **O**, build the LRF and compute the normal of each point (*left*). Once the geometric feature vectors built, they are fed into a learning module for convolution and attention features aggregation (*right*)

3.3 Proposed Method

Algorithm 1: Pseudo-code of the FSR module

Input N: Number of iterations, k: Number of sampled points, τ: Inlier threshold, $\bar{\boldsymbol{X}}$, $\bar{\boldsymbol{Y}}$: Paired source and target point clouds

Output $\boldsymbol{T}$ # Estimated transformation matrix [4,4]

$\hat{\boldsymbol{T}} = zeros(4,4)$ # Initialization with 4x4 null matrix
$\hat{\boldsymbol{T}}[3,3] \leftarrow 1.0$ # Set the last cell value
$best \leftarrow 0$ # Initialize the best number of inliers
for $i = 1$ **to** N **do**
  $ind = FPS(\bar{\boldsymbol{X}}, k)$ # $[k, 1]$, Farthest points indices
  $\tilde{\boldsymbol{X}} = \bar{\boldsymbol{X}}[ind, :]$ # $[k, 3]$, Farthest source points gathering
  $\tilde{\boldsymbol{Y}} = \bar{\boldsymbol{Y}}[ind, :]$ # $[k, 3]$, Farthest target points gathering
  $\tilde{\boldsymbol{x}}_{\boldsymbol{m}} = \text{mean}(\tilde{\boldsymbol{X}}, \text{axis} = 0)$ # $[1, 3]$, Source points centroid
  $\tilde{\boldsymbol{y}}_{\boldsymbol{m}} = \text{mean}(\tilde{\boldsymbol{Y}}, \text{axis} = 0)$ # $[1, 3]$, Target points centroid
  $\tilde{\boldsymbol{X}}_c \leftarrow \tilde{\boldsymbol{X}} - \tilde{\boldsymbol{x}}_{\boldsymbol{m}}$ # Center source points
  $\tilde{\boldsymbol{Y}}_c \leftarrow \tilde{\boldsymbol{Y}} - \tilde{\boldsymbol{y}}_{\boldsymbol{m}}$ # Center target points
  $[\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V}] \leftarrow \text{SVD}(\tilde{\boldsymbol{X}}_c^T \cdot \tilde{\boldsymbol{Y}}_c)$ # SVD of the covariance matrix
  $\boldsymbol{R} \leftarrow \boldsymbol{U} \cdot \boldsymbol{V}^T$ # Compute rotation matrix
  $\boldsymbol{t} \leftarrow \tilde{\boldsymbol{y}}_{\boldsymbol{m}} - \boldsymbol{R} \cdot \tilde{\boldsymbol{x}}_{\boldsymbol{m}}$ # Compute translation vector
  $\hat{\boldsymbol{T}}[0:3, 0:3] \leftarrow \boldsymbol{R}$ # Set rotation part
  $\hat{\boldsymbol{T}}[0:3, 3] \leftarrow \boldsymbol{t}$ # Set translation part
  $\bar{\boldsymbol{X}}_y \leftarrow Apply\_transform(\bar{\boldsymbol{X}}, \hat{\boldsymbol{T}})$ # Transform source points
  $n_i \leftarrow 0$ # Reset the inlier count for this iteration
  for $j = 1$ **to** $length(\bar{\boldsymbol{X}}_y)$ **do**
    $d \leftarrow \|\bar{\boldsymbol{X}}_y[j, :] - \bar{\boldsymbol{Y}}[j, :]\|$ # Pairwise Euclidean distance
    if $d < \tau$ **then**
      $n_i \leftarrow n_i + 1$ # Count as inlier if within threshold
    end
  end
  if $n_i > best$ **then**
    $best \leftarrow n_i$ # Update best inliers number
    $\boldsymbol{T} \leftarrow \hat{\boldsymbol{T}}$ # Update best transformation
  end
end
Return $\boldsymbol{T}$ # Best transformation matrix [4,4]

Geometric features using the 3D covariance matrix First, inspired by [43], the local structure of each point $\boldsymbol{x}_i$ is described using the three scalar values: the anisotropy $\mathbf{A}_i$, the planarity $\mathbf{P}_i$ and the omnivariance $\mathbf{O}_i$. These functions are extracted from the 3D covariance matrix $\boldsymbol{\Sigma}_i$ of the k nearest neighbors of $\boldsymbol{x}_i$ by applying an eigenvalue decomposition on $\boldsymbol{\Sigma}_i$ to get $\lambda_i^1 \geq \lambda_i^2 \geq \lambda_i^3$. Then, $\mathbf{A}_i$, $\mathbf{P}_i$ and $\mathbf{O}_i$ are given by:

$$\mathbf{A}_i = \frac{\lambda_i^1 - \lambda_i^3}{\lambda_i^1}, \quad \mathbf{P}_i = \frac{\lambda_i^2 - \lambda_i^3}{\lambda_i^1}, \text{ and } \mathbf{O}_i = \left(\lambda_i^1 \cdot \lambda_i^2 \cdot \lambda_i^3\right)^{1/3} \tag{4}$$

Besides this, the eigenvectors $\boldsymbol{u}_i, \boldsymbol{v}_i$ and $\boldsymbol{w}_i$ associated to λ_i^1, λ_i^2 and λ_i^3 respectively are used to build a Local Reference Frame (LRF) for each point by stacking them as column vectors in a matrix $\boldsymbol{R}_i^{lrf} \in \mathbb{R}^{3\times3}$. The latter is used to project the estimated normal vectors detailed in the Section above (3.3)

Normal estimation using local triangles Inspired by the umbrella surface representation [22], we compute the surface normal vector of each point using the $k-1$ triangles formed by its k nearest neighbors. To do so, the normal $\boldsymbol{z}_j$ to the plan containing each triangle j is computed using a simple cross product between its two edges. The final normal vector $\boldsymbol{n}_i$ associated to the point $\boldsymbol{x}_i$ is computed by a weighted sum of the $k-1$ triangle normals:

$$\boldsymbol{n}_i = \sum_{j=1}^{k-1} \left(\omega_j \boldsymbol{z}_j\right) \tag{5}$$

where the weights ω_j are the result of a *SoftMax* operation over the areas of each triangle to give a bigger contribution for the largest triangles and reduce the smallest triangles contribution since experiments showed they are more sensitive to the presence of noise.

Local-to-global feature aggregation Once the geometrical properties $\mathbf{A}_i$, $\mathbf{P}_i$ and $\mathbf{O}_i$ and the normal $\boldsymbol{n}_i$ are estimated for each single point, we propose to construct an initial descriptor vector $\boldsymbol{f}_i^0$ by concatenating the 3D coordinates of each $\boldsymbol{x}_i$ and its associated $\mathbf{A}_i$, $\mathbf{P}_i$ and $\mathbf{O}_i$:

$$\boldsymbol{f}_i^0 = \left[\boldsymbol{x}_i, \mathbf{A}_i, \mathbf{P}_i, \mathbf{O}_i\right], \boldsymbol{f}_i^0 \in \mathbb{R}^6 \tag{6}$$

Inspired by [38], we propose to propagate this local information by comparing it to the feature vectors of the k nearest neighbors of $\boldsymbol{x}_i$ to get a new intermediate geometrical descriptor:

$$\boldsymbol{f}_i^1 = \left[\Delta_i^1, ..., \Delta_i^k\right], \boldsymbol{f}_i^1 \in \mathbb{R}^{k\times12} \tag{7}$$

where Δ_i^j is the vector capturing the information between $\boldsymbol{x}_i$, and its j^{th} neighbour. It is obtained by the following:

$$\Delta_i^j = \left[\boldsymbol{f}_i^0, (\boldsymbol{f}_j^0 - \boldsymbol{f}_j^0)\right], \Delta_i^j \in \mathbb{R}^{1\times12} \tag{8}$$

To better exploit the very local structure around each point, the normal vectors $\boldsymbol{n}_i$ complete the last vector and get the final geometrical descriptor $\boldsymbol{f}_i^2$. To do so, we propose to collect the normals of the same *knn* and project them in the *LRF* estimated previously. The projected normal of the j^{th} neighbour noted $\boldsymbol{n}_j^{lrf}$ is thus obtained by the following:

$$\boldsymbol{n}_j^{lrf} = \boldsymbol{R}_i^{lrf} \cdot \boldsymbol{n}_j \tag{9}$$

The projected vectors of the *knn* are then stacked, resulting in a new vector $\boldsymbol{f}_i^n \in \mathbb{R}^{k\times 3}$. The final descriptor $\boldsymbol{f}_i^2$ is the concatenation of $\boldsymbol{f}_i^n$ and $\boldsymbol{f}_i^1$:

$$\boldsymbol{f}_i^2 = \left[\boldsymbol{f}_i^1, \boldsymbol{f}_i^n\right], \boldsymbol{f}_i^2 \in \mathbb{R}^{k\times 15} \tag{10}$$

The so-obtained descriptor encodes the variation of anisotropy, planarity, and omnivariance between the point and its neighbors, as well as the variation of the normals. The same procedure is repeated for each point from the source and the target point clouds. To improve the robustness of the points representation, the features are fed to successive learning modules. Inside each point cloud, all the points and their knn form a graph feature tensor $\mathrm{R}^{N\times k\times 15}$ which is fed into a CNN composed of three successive 2D convolutions with 1×1 kernel size to increase the features dimension from 15 to d resulting in a new feature tensor $\mathrm{R}^{N\times k\times d}$. Each layer is followed by a group normalisation operation and a *ReLU* activation function to get updated features $\boldsymbol{f}_i^{cnn} \in \mathbb{R}^{k\times d}$ for each point. A *MaxPool* operation over the k neighbours is applied to obtain a single dimensional features vector $\boldsymbol{f}_i^{mp} = MaxPool(\boldsymbol{f}_i^{cnn}) \in \mathbb{R}^d$. Finally, to propagate a global intra-point cloud representation for the two input sets, a *self-attention* mechanism is introduced. It contains three layers incorporating the 3D rotatory position encoding from [17] and [32] to build an efficient and translation-invariant representation within the *self-attention* module. This is achieved by rotating the descriptor of each point with a matrix $\boldsymbol{R}^{sa} \in \mathbb{R}^{d\times d}$ defined as detailed below:

$$\boldsymbol{R}^{sa} = Diag\big[\boldsymbol{R}^1, \boldsymbol{R}^2, \ldots, \boldsymbol{R}^{d/6}\big]$$

where each $\boldsymbol{R}^j$ is a $\mathbb{R}^{6\times 6}$ matrix defined as follows:

$$\boldsymbol{R}^j = \begin{bmatrix} \cos x\theta_j & -\sin x\theta_j & 0 & 0 & 0 & 0 \\ \sin x\theta_j & \cos x\theta_j & 0 & 0 & 0 & 0 \\ 0 & 0 & \cos y\theta_j & -\sin y\theta_j & 0 & 0 \\ 0 & 0 & \sin y\theta_j & \cos y\theta_j & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos z\theta_j & -\sin z\theta_j \\ 0 & 0 & 0 & 0 & \sin z\theta_j & \cos z\theta_j \end{bmatrix} \tag{11}$$

where x, y and z are the spatial coordinates of the point $\boldsymbol{x}_i$ and θ_j is the embedding associated to the j^{th} dimension given by $\theta_j = 1/\big(10000^{6(j-1)/d}\big)$. The final feature vector is then updated thanks to the following equations:

$$\boldsymbol{f}_i^{mp} \leftarrow \boldsymbol{f}_i^{mp} + MLP\big[\boldsymbol{R}^{sa}\mathbf{W}^Q\boldsymbol{f}_i^{mp}, \sum_{j=1}^{M} \alpha_{ij}\big(\boldsymbol{R}^{sa}\mathbf{W}^V\boldsymbol{f}_j^{mp}\big)\big] \tag{12}$$

with $[\cdot,\cdot]$ is the concatenation operation, j covering all the points contained in the source point cloud $\boldsymbol{X} \in \mathbb{R}^{M\times 3}$, and:

$$\alpha_{ij} = \underset{j}{softmax}\left(\frac{(\boldsymbol{R}^{sa}\mathbf{W}^{Q}\boldsymbol{f}_i^{mp})(\boldsymbol{R}^{sa}\mathbf{W}^{K}\boldsymbol{f}_j^{mp})^{T}}{\sqrt{d}}\right) \tag{13}$$

where $\mathbf{W}^{Q} \in \mathbb{R}^{d\times d}$, $\mathbf{W}^{K} \in \mathbb{R}^{d\times d}$ and $\mathbf{W}^{V} \in \mathbb{R}^{d\times d}$ are the learned projection matrices for the query, the key and the value points in the attention message passing. The exact same operations are applied for all the points $\boldsymbol{y}_i$ from the target point cloud $\boldsymbol{Y} \in \mathbb{R}^{N\times 3}$. The final descriptor of each source point $\boldsymbol{x}_i \in \boldsymbol{X}$ will be referred to as $\boldsymbol{f}_i \in \mathbb{R}^d$. At the same time, we call $\boldsymbol{h}_i \in \mathbb{R}^d$, the final descriptor of each target point $\boldsymbol{y}_i \in \boldsymbol{Y}$. We then follow the algorithm described in Section 3.2 for point matching, and we adopt the Farthest Sampling-guided Registration [30] for the rigid transformation estimation.

4 EXPERIMENTS

The method is implemented in PyTorch and trained on a Nvidia Tesla V100-32G GPU using the Adam optimiser [14] with a learning rate of 10^{-4} for 100 epochs for the *ModelNet40* dataset and 200 epochs for the *MVP-RG* dataset. A number of $k = 30$ nearest neighbors are used to construct the local triangles and graphs, while the feature dimension d is set to 132. The kernel size of all the convolutions is set to 1×1, and the *GroupNorm* operations are followed by a *ReLU* activation function. The output features of the CNN $\boldsymbol{f}_i^{mp} \in \mathbb{R}^{132}$ are fed into 4 self-attention layers. For the LRF and the geometric features $\mathbf{A}$, $\mathbf{P}$ and $\mathbf{O}$ estimation, we collect a maximum of 128 neighbors and keep only those that are less than $r = 0.3$ away from the point on which the PCA is applied.

4.1 Datasets

To evaluate *LoGDesc* performances, we first test it on the synthetic *ModelNet40* dataset [39]. This last proposes 9,843 point clouds for training and 2,468 for testing, containing 2048 points sampled from CAD models. As in [37], the target point clouds are created by randomly rotating each set by an angle between 0 and 45 degrees and translating them by a displacement between -0.5 and 0.5 along each axis. This is followed by a random permutation of the points before 1024 points are selected for each object as input. Besides this, we follow [37] and test the robustness to noise and to outliers of *LoGDesc*, first, by adding a Gaussian noise sampled from N (0, 0.01) and clipped to [0.05, 0.05] to each point. Finally, occlusions are simulated by selecting the 768 nearest neighbors from the source and the target point clouds of a random point in space. For a second time, we propose to follow [13] and assess the ability of *LoGDesc* to generalize to unseen objects and real point clouds by testing the model trained on ModelNet using the 10 range scanner point clouds from *Stanford Bunny* dataset [33]. The root mean squared error (RMSE) and mean absolute error (MAE) between

ground truth values and estimated values for rotation and translation are reported in the following section for registration evaluation. At the same time, precision (P), accuracy (A), and recall (R) are used to analyze the estimation of point correspondence. Finally, we assess our method on MVP (Multi-View Partial)-RG dataset [18], which contains 7600 partial point clouds representing 16 different objects categories generated by virtual cameras from diverse viewpoints leading to inconsistent local point densities and thus making this dataset extremely challenging. We perform the exact same study than [18] by splitting the dataset into 6400 training samples and 1200 testing samples and using the same metrics: isotropic rotation errors L_R, translation errors L_t, and the root mean square error L_{RMSE}. We propose to recreate the USIP [15] and MDGAT [27] experiments on *KITTI* where 256 keypoints are used for registration to address the computational cost of our method on large point clouds, as noted in the paper's conclusion. We follow the same training, testing procedures, and metrics as MDGAT (Failure Rate **FR** and the Inlier Ratio **IR**).

Finally, we aim to explore another potential application of *LoGDesc*, i.e., medicine as outlined by [11] dealing with knee arthroplasty surgery. Specifically, we plan to evaluate the model trained on *ModelNet40* using a set of point clouds representing a human femur. This dataset includes point clouds captured with a 12 MPx TrueDepth smartphone camera and a Kinect containing approximately $10,000$ points. Additionally, we will randomly sample $5,000$ points from the CAD model to generate an input point cloud for the model. For evaluation, $2,048$ points will be randomly sampled from the initial point clouds, following the same procedure as *ModelNet40* to simulate partial overlap and noisy input.

4.2 Results

In the matching challenge, Table 1 shows that *LoGDesc* outperforms the state-the-art descriptors DGCNN [38], FPFH [24] and the triangle-based descriptor [30] in all the metrics on the noisy point clouds, thus demonstrating the robustness of the proposed method to noise. Concerning the transformation estimation, experiments show that the good results in the matching challenge highlighted previously are reflected in the estimation of the rigid transformation as shown in Tables 2 and 3. Indeed, *LoGDesc* outperforms all tested methods in three of the four metrics when applied to occlusion-free point clouds, is second in translation RMSE behind R-PointHop [13], and is first in all metrics on partially overlapping point clouds, ex-œquo with RoCNet++ in translation, both methods outperforming the second best (RoCNet [31] and GeoTransformer [21]) by 50% in RMSE and by 67% in MAE. On this challenging case of noisy partially overlapping point clouds, we also report in Table 4 the results under the relative rotation L_R and relative translation L_t errors and the RMSE on the rigid transformation L_{RMSE} metrics, as proposed in [46]. Table 4 confirms the strong performances seen in the previous table, since it matches the best performance in L_t and L_{RMSE} while ranking second in L_R. Our method shows an interesting generalization ability since the model trained on *ModelNet40* achieved, on the *Stanford Bunny* dataset, the best result in the RMSE($\mathbf{t}$), the second in the

MAE(**R**) while being third in RMSE(**R**) right behind RoCNet (with a 2% gap) and fourth in the MAE(**t**) as shown in Table 5.

Table 1. Matching performances on *ModelNet40* with noisy and partially overlapping point clouds

Descriptor	**P**(↑)	**A**(↑)	**R**(↑)
FPFH [23]	72.1	71.0	71.2
DGCNN [38]	85.8	85.9	85.5
RoCNet++ [30]	<u>89.4</u>	<u>89.4</u>	<u>89.2</u>
Ours	**92.0**	**92.1**	**91.9**

Moreover, Table 6 highlights that our method can handle very challenging configurations with varying local densities, low overlapping point clouds, and unrestricted rotations $[0°, 360°]$ since it outperforms the other methods in all the used metrics by a 55% relative gap at least. The last two rows of the same Table also highlight that *LoGDesc* can perform well when combined with other pose estimators than FSR since the registration performances still make our method the best in three metrics when the matching process detailed in Section 3.2 is skipped. The transformation is estimated using RANSAC or FGR based on feature matching. Table 7 indicates strong abilities of *LoGDesc* to handle low overlapping and real world sources data (*i.e* Lidar) which may be corrupted with sensor noise, since it comes first for **FR** and second for **IR** on the *KITTI* keypoints registration.

An additional experiment was carried out using real data from a human femur as described in section 4.1. Figure 4 shows a sample of the registration results. The results indicate that *LoGDesc* can effectively generalise to unseen and real data, provided that the desired transformation matrix is within the range seen during training. The performance of *LoGDesc* is promising, especially given its ability to generalise to other data, especially in certain applications where annotated data is difficult to obtain, such as medical applications like neurosurgery and orthopaedics.

4.3 Ablation Study

To highlight the impact of using the geometric properties proposed in LoGDesc, an ablation study is proposed here under noisy, partially overlapping point clouds from *ModelNet40*. In Table 8, different versions of *LoGDesc* are tested by removing each geometric variable -**A**,**P**, **O** and the normals (**N**) - one by one, leaving the rest of the architecture unchanged. The results show that each variable has its contribution to the descriptor performance, with in particular the removal of **A**, **O**, and **P** leading to a 7% drop in matching metrics.

Table 2. Performances on noisy and fully overlapping data from *ModelNet40*.

Method	RMSE(**R**)	MAE(**R**)	RMSE(**t**)	MAE(**t**)
DCP-V2 [36]	8.417	5.685	0.03183	0.02337
PRNET [37]	3.218	1.446	0.11178	0.00837
R-PointHop [13]	2.780	0.980	**0.00087**	0.00375
VRNet [47]	2.558	1.016	0.00570	0.00289
GeoTransf [21]	0.692	0.267	0.00519	0.00200
RoCNet [31]	1.920	0.555	0.00260	0.00180
RoCNet++ [30]	1.004	0.249	0.00133	0.00092
Ours	**0.618**	**0.187**	0.00121	**0.00089**

Table 3. Performances on noisy and partially overlapping data from *ModelNet40*.

Method	RMSE(**R**)	MAE(**R**)	RMSE(**t**)	MAE(**t**)
DCP-V2 [36]	6.883	4.534	0.028	0.021
PRNET [37]	4.323	2.051	0.017	0.012
VRNet [47]	3.615	1.637	0.010	0.006
GeoTransf [21]	0.915	0.386	0.007	0.003
RoCNet [31]	1.810	0.620	0.004	0.003
RoCNet++ [30]	1.278	0.318	**0.002**	**0.001**
Ours	**0.774**	**0.266**	**0.002**	**0.001**

Table 4. Performances on noisy and partial data from *ModelNet40* using the metrics from [46]

	PRNet [37]	DCP [36]	Predator [12]	GMCNet [18]	RIGA [46]	**Ours**
L_R	4.37°	9.33°	3.33°	**0.94°**	1.15°	0.95°
L_t	0.034	0.070	0.018	0.007	**0.006**	**0.006**
L_{RMSE}	0.045	0.018	0.025	**0.008**	0.009	**0.008**

Table 5. Performances on the *Bunny Stanford* dataset

Method	RMSE(**R**)	MAE(**R**)	RMSE(**t**)	MAE(**t**)
FGR [48]	1.99	1.49	0.1993	0.1658
DCP [36]	6.44	4.78	0.0406	0.0374
R-PointHop [13]	1.49	1.09	0.0361	**0.0269**
RoCNet [31]	0.99	0.83	0.0338	0.0288
RoCNet++ [30]	**0.91**	**0.50**	0.0316	0.0301
Ours	1.01	0.73	**0.0272**	0.0310

Table 6. Performances on Multi-View Partial virtual scan ReGistration (*MVP-RG*). The indication $^{(\dagger)}$ indicates a pose estimation based on *LoGDesc* features matching, without the matching module 3.2

Method	L_R	L_t	L_{RMSE}
IDAM [16]	24.35°	0.280	0.344
RGM [7]	41.27°	0.425	0.583
DCP [36]	30.37°	0.273	0.634
Predator [12]	10.58°	0.067	0.125
RPMNet [44]	22.20°	0.174	0.327
GMCNet [18]	16.57°	0.174	0.246
Ours	7.33°	**0.043**	0.099
Ours$^{(\dagger)}$ - RANSAC (1k)	**6.64°**	0.053	**0.082**
Ours$^{(\dagger)}$ - FGR	9.08°	0.061	0.099

Table 7. Registration performance on *KITTI* using 256 USIP keypoints.

Metric	**FPFH** [23]	**SHOT** [28]	**3DFeatNet** [45]	**USIP** [15]	**DCP** [36]	**MDGAT+ SuperGlue** [27, 25]	**MDGAT** [27]	**Ours**
FR	8.37	5.40	1.55	1.41	4.01	0.58	0.67	**0.52**
IR	18.77	18.21	22.48	32.20	35.37	36.19	**42.23**	37.7

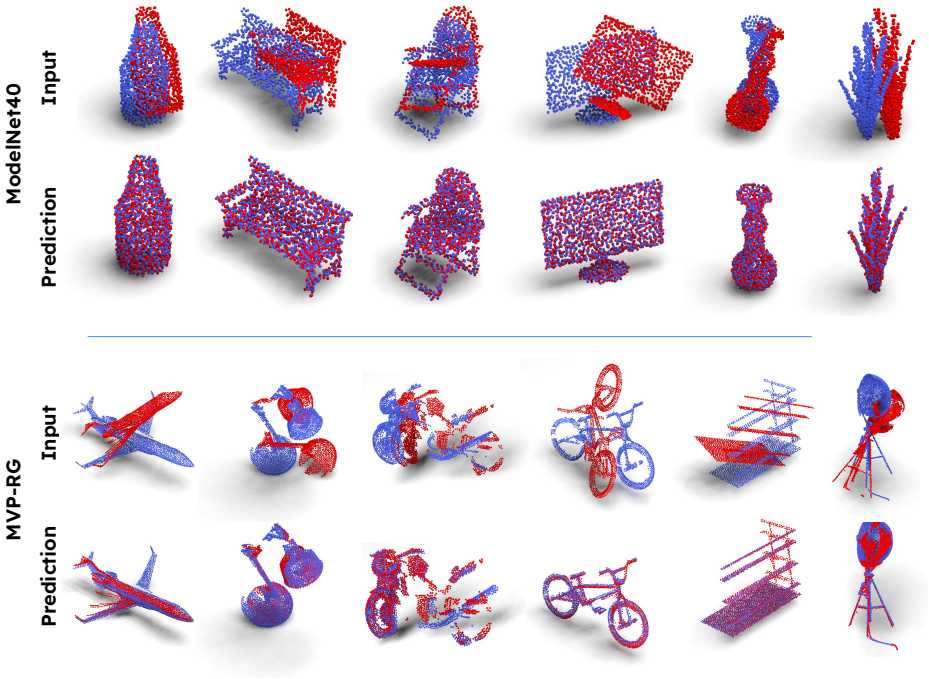

Fig. 3. Performed registrations on *ModelNet40* (top) and *MVP-RG* (bottom).

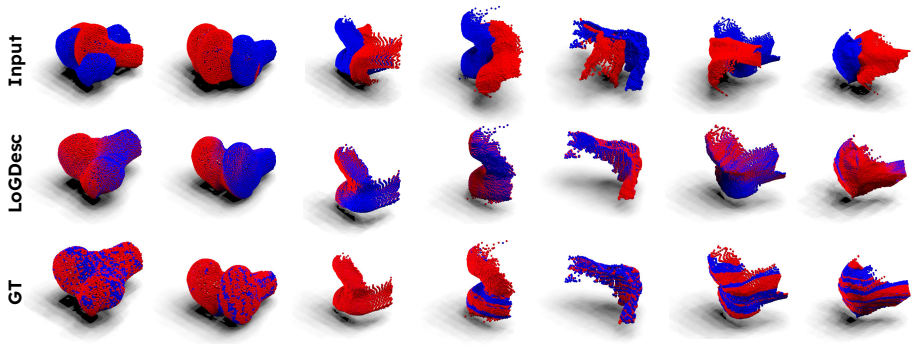

Fig. 4. Performed registrations on a 3D printed femur.

Table 8. Matching performances on noisy data (*ModelNet40*)

Metric	w/o **A,O,P**	w/o **A**	w/o **O**	w/o **P**	w/o **N**	**Ours**
P(↑)	85.2	88.7	91.6	88.8	91.6	**92.0**
A(↑)	85.4	88.8	91.7	89.0	91.7	**92.1**
R(↑)	85.0	88.4	91.5	88.7	91.5	**91.9**

5 CONCLUSION

This paper presented a new robust descriptor called *LoGDesc* and a deep learning architecture for point cloud registration. This descriptor exploits local and global geometric properties of 3D points and machine learning techniques for robust feature extraction and point matching. By aggregating local geometric information such as flatness, anisotropy, and omnivariance, as well as features learned through graph convolutions and attention mechanisms, *LoGDesc* demonstrates superior performance, especially in handling noisy point clouds and challenging registration scenarios where most methods in the literature show limitations in terms of robustness and precision. An evaluation of our method was carried out on the *ModelNet40*, *Stanford Bunny*, *MVP-RG* and *KITTI* datasets, highlighting the effectiveness and robustness of *LoGDesc*, which surpasses the most advanced and recent methods in the literature, especially on noisy data.

For future work, we plan to extend our method to other robotics tasks such as object recognition, visual servoing, and 6 DoF multi-object pose estimation. We will also focus on improving the scalability of *LoGDesc* to handle larger cloud points, which currently pose a computational problem due to the use of attention mechanisms in the pipeline using, for instance, the superpoints concept.

Acknowledgments. This work was supported by the French ANR program MARSurg (ANR-21-CE19-0026).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Aldoma, A., Marton, Z.C., Tombari, F., et al.: Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation. IEEE Rob. & Auto. Mag. **19**(3), 80–91 (2012)
2. Audette, M.A., Ferrie, F.P., Peters, T.M.: An algorithmic overview of surface registration techniques for medical imaging. Med. Imag. Anal. **4**(3), 201–217 (2000)
3. Avidar, D., Malah, D., Barzohar, M.: Local-to-global point cloud registration using a dictionary of viewpoint descriptors. In: IEEE Int. Conf. on Comput. Vision. pp. 891–899 (2017)
4. Besl, P.J., McKay, N.D.: Method for registration of 3-d shapes. In: Sensor fusion IV: control paradigms and data structures. vol. 1611, pp. 586–606 (1992)
5. Dong, Z., Liang, F., et al.: Registration of large-scale terrestrial laser scanner point clouds: A review and benchmark. J. of Photog. and Remote Sens. **163**, 327–342 (2020)
6. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**, 381–395 (1981)
7. Fu, K., Liu, S., Luo, X., Wang, M.: Robust point cloud registration framework based on deep graph matching. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8893–8902 (2021)
8. Geng, J.: Adv. Opt. Photon. **3**(2), 128–160 (2011)
9. Gojcic, Z., Zhou, C., Wegner, J.D., Wieser, A.: The perfect match: 3d point cloud matching with smoothed densities. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5545–5554 (2019)
10. Haala, N., Peter, M., Kremer, J., et al.: Mobile lidar mapping for 3d point cloud collection in urban areas—a performance test. Int. Arch. Photo. Remote Sens. Spat. Inf. Sci **37**, 1119–1127 (2008)
11. Hu, X., Liu, H., Baena, F.R.Y.: Markerless navigation system for orthopaedic knee surgery: A proof of concept study. IEEE Access **9**, 64708–64718 (2021)
12. Huang, S., Gojcic, Z., Usvyatsov, M., et al.: Predator: Registration of 3d point clouds with low overlap. In: IEEE/CVF Conf. Comput. Vision Pattern Recognit. pp. 4267–4276 (2021)
13. Kadam, P., Zhang, M., Liu, S., et al.: R-pointhop: A green, accurate, and unsupervised point cloud registration method. IEEE Trans. on Ima. Process. **31**, 2710–2725 (2022)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
15. Li, J., Lee, G.H.: Usip: Unsupervised stable interest point detection from 3d point clouds. arXiv preprint arXiv:1904.00229 (2019)
16. Li, J., Zhang, C., Xu, Z., Zhou, H., Zhang, C.: Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16. pp. 378–394. Springer (2020)
17. Li, Y., Harada, T.: Lepard: Learning partial point cloud matching in rigid and deformable scenes. In: IEEE/CVF Conf. on Comput. Vis. and Pattern Recogn. pp. 5554–5564 (2022)
18. Pan, L., Cai, Z., Liu, Z.: Robust partial-to-partial point cloud registration in a full range. IEEE Rob. and Auto. Lett. (2024)

19. Pomerleau, F., Colas, F., Siegwart, R., et al.: A review of point cloud registration algorithms for mobile robotics. Found. and Trends® in Rob. **4**(1), 1–104 (2015)
20. Qi, C.R., Su, H., et al.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Conf. Comput. Vision Pattern Recognit. pp. 652–660 (2017)
21. Qin, Z., Yu, H., Wang, C., et al.: Geometric transformer for fast and robust point cloud registration. In: IEEE/CV Conf. Comput. Vision Pattern Recognit. pp. 11143–11152 (2022)
22. Ran, H., Liu, J., Wang, C.: Surface representation for point clouds. In: IEEE/CVF Conf. on Comp. Vis. and Pattern Recog. pp. 18942–18952 (2022)
23. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: IEEE Int. Conf. Rob. and Auto. pp. 3212–3217 (2009)
24. Rusu, R.B., Blodow, N., et al.: Fast point feature histograms (fpfh) for 3d registration. In: IEEE Int. Conf. on Rob. and Auto. pp. 3212–3217 (2009)
25. Sarlin, P.E., DeTone, D., et al.: SuperGlue: Learning feature matching with graph neural networks. In: Conf. Comput. Vision Pattern Recognit. (2020)
26. Schwarz, B.: Mapping the world in 3d. Nat. Photo. **4**(7), 429–430 (2010)
27. Shi, C., Chen, X., Huang, K., et al.: Keypoint matching for point cloud registration using multiplex dynamic graph attention networks. IEEE Rob. and Auto. Let. **6**, 8221–8228 (2021)
28. Shi, S., Wang, Z., Shi, J., et al.: From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. EEE Trans. Pattern Anal. Mach. Intell. **43**(8), 2647–2664 (2020)
29. Sinkhorn, R., Knopp, P.: Concerning nonnegative matrices and doubly stochastic matrices. Pacific J. of Math.
30. Slimani, K., Achard, C., Tamadazte, B.: Rocnet++: Triangle-based descriptor for accurate and robust point cloud registration. Pattern Recogn. **147**, 110108 (2024)
31. Slimani, K., Tamadazte, B., Achard, C.: Rocnet: 3d robust registration of point-clouds using deep learning. arXiv preprint arXiv:2303.07963 (2023)
32. Su, J., Ahmed, M., Lu, Y., et al.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing **568**, 127063 (2024)
33. Turk, G., Levoy, M.: Zippered polygon meshes from range images. In: Conf. on Comp. Grap. and Inter. Tech. pp. 311–318 (1994)
34. Vaswani, A., Shazeer, N., et al.: Attention is all you need. Adv. Neural. Inf. Process. Syst. **30** (2017)
35. Wang, Q., Kim, M.K.: Applications of 3d point cloud data in the construction industry: A fifteen-year review from 2004 to 2018. Adv. Eng. Info. **39**, 306–319 (2019)
36. Wang, Y., Solomon, J.M.: Deep closest point: Learning representations for point cloud registration. In: IEEE Int. Conf. on Comput. Vision (2019)
37. Wang, Y., Solomon, J.M.: Prnet: Self-supervised learning for partial-to-partial registration. Adv. Neural. Inf. Process. Syst. **32** (2019)
38. Wang, Y., Sun, Y., et al.: Dynamic graph cnn for learning on point clouds. ACM Trans. on Grap. (2019)
39. Wu, Z., Song, S., Khosla, A., et al.: 3d shapenets: A deep representation for volumetric shapes. In: Conf. Comput. Vision Pattern Recognit. pp. 1912–1920 (2015)
40. Xia, Y., Gladkova, M., Wang, R., Li, Q., Stilla, U., Henriques, J.F., Cremers, D.: Casspr: Cross attention single scan place recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8461–8472 (2023)
41. Xia, Y., Xu, Y., Li, S., Wang, R., Du, J., Cremers, D., Stilla, U.: Soe-net: A self-attention and orientation encoding network for point cloud based place recognition.

In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition. pp. 11348–11357 (2021)
42. Yang, J., Cao, Z., Zhang, Q.: A fast and robust local descriptor for 3d point cloud registration. Info. Sci. **346**, 163–179 (2016)
43. Yang, J., Zhao, M., Wu, Y., et al.: Accurate and robust registration of low overlapping point clouds. Comput. & Graph. **118**, 146–160 (2024)
44. Yew, Z.J., Lee, G.H.: Rpm-net: Robust point matching using learned features. In: IEEE/CVF Conf. Comput. Vision Pattern Recognit. pp. 11824–11833 (2020)
45. Yew, Z.J., Lee, G.H.: 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In: Euro. Confe. on Comput. Vision (2018)
46. 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 Transactions on Pattern Analysis and Machine Intelligence (2024)
47. Zhang, Z., Sun, J., Dai, Y., et al.: Vrnet: Learning the rectified virtual corresponding points for 3d point cloud registration. IEEE Trans. on Cir. and Sys. for Video Tech. **32**, 4997–5010 (2022)
48. Zhou, Q.Y., Park, J., Koltun, V.: Fast global registration. In: Europ. Conf. on Computer Vision. pp. 766–782 (2016)