sensors

MDPI

Article

A Structure-Based Iterative Closest Point Using Anderson Acceleration for Point Clouds with Low Overlap

Chao Zeng [1,2], Xiaomei Chen [1,2,*], Yongtian Zhang [1,2] and Kun Gao [1,2]

[1] School of Optics and Photonics, Beijing Institute of Technology, Beijing 100081, China
[2] MOE Key Laboratory of Optoelectronic Imaging Technology and System, Beijing Institute of Technology, Beijing 100081, China
* Correspondence: cxiaomei@bit.edu.cn; Tel.: +86-10-68912560

Abstract: The traditional point-cloud registration algorithms require large overlap between scans, which imposes strict constrains on data acquisition. To facilitate registration, the user has to strategically position or move the scanner to ensure proper overlap. In this work, we design a method of feature extraction based on high-level information to establish structure correspondences and an optimization problem. And we rewrite it as a fixed-point problem and apply the Lie algebra to parameterize the transform matrix. To speed up convergence, we introduce Anderson acceleration, an approach enhanced by heuristics. Our model attends to the structural features of the region of overlap instead of the correspondence between points. The experimental results show the proposed ICP method is robust, has a high accuracy of registration on point clouds with low overlap on a laser datasets, and achieves a computational time that is competitive with that of prevalent methods.

Keywords: registration; point clouds; overlap; Anderson acceleration

check for updates

Citation: Zeng, C.; Chen, X.; Zhang, Y.; Gao, K. A Structure-Based Iterative Closest Point Using Anderson Acceleration for Point Clouds with Low Overlap. *Sensors* **2023**, *23*, 2049. https://doi.org/10.3390/s23042049

Academic Editor: Jason Rambach

Received: 29 December 2022
Revised: 7 February 2023
Accepted: 8 February 2023
Published: 11 February 2023

Copyright: © 2023 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

Rigid point-cloud registration consists of finding the best transformation between point clouds, and plays a key role in robotics and computer vision. Its wide application has rendered it ubiquitous, such as in simultaneous localization and mapping (SLAM) [1,2], 3D reconstruction [3], and motion estimation [4]. At first glance, its performance on various benchmarks gives the impression that it can solve the problem of registration. However, it has now been noted that this is not the case owing to the leniency of the evaluation protocols used, where the benchmarks used in recent literature have considered the overlap between only pairs of point clouds $\geq$80% and reached nearly $\geq$90% matching recall [5,6]. The registration-related performance of prevalent methods deteriorate rapidly in practice when the input data contain a region with a low overlap with the template data [7]. Moreover, many approaches are not capable of being generalized to large-scale empirical 3D point clouds owing to their different densities and the characteristics of LiDAR sensors [8].

The classic Iterative Closest Points (ICP) [9] algorithm alternates between querying the closest point in the target set and minimizing the distance between corresponding points, and guarantees convergence to an optimal alignment. However, the major shortcoming of the ICP when used for registration is that it can converge to only a local optimization near the initialization. The search for corresponding points is time consuming when large-scale point clouds need to be processed, and the differences in their densities, missing data, and regions of low overlap in laser scans lead to poor registration-related performance [10], as illustrated in the example shown in Figure 1.

In this work, we address the registration of pairs of point clouds with a low overlap as captured by LiDAR from real-world scenes. Point clouds scanned by different LiDAR-based methods vary greatly. Thus, different approaches that rely on a sufficient amount of overlapping and descriptive features cannot guarantee the performance expected of them. In fact, man-made scenes like building interiors and exteriors comprising mainly planar

and edge-related structures are universal in the real-world. We think that edge-related and planar features along with their inter-relations reveal high-level global characteristics that provide a sufficient amount of information for point cloud registration. However, currently used plane-/line-based methods have unsatisfactory robustness and a high time cost [6,10,11].



(**a**) (**b**) (**c**)

Figure 1. When registering point clouds with regions of low overlap, the iterative closest point method is prone to becoming trapped in a local minimum. (**a**) Input point clouds. (**b**) Ground-truth registration. (**c**) The result of the ICP.

Key to our approach here is a combination of planar and edge-related features as well as the solution of a fixed-point problem. Specifically, we extract planar and edge-related features from a raw point cloud captured by the mechanical LiDAR and the solid-state LiDAR. And then we establish structure-level correspondences between them and formulate the optimization problem. Note that the non-linear optimization problem can be rewritten as a fixed-point equation, which can be solved by Anderson acceleration. Instead of standard iteration usually used in ICP, this method is an iterative procedure for finding a fixed point of contractive mapping. Nevertheless, the value computed by Anderson Acceleration is an affine combination of rotation matrices. To address it, we apply the Lie algebra to parameterize the transformation.

We provide benchmark datasets in this paper that are obtained through scans of a set of indoor and outdoor scenes with a low overlap. We use RealSense L515 to scan indoor scenes as it has a high angular resolution and a high density of points [12]. The outdoor dataset used here consisted of KITTI data captured by using the Volodyne HDL-64E LiDAR sensor [13]. The performance of the method of registration was simply measured based on the percentage of successful registration scans, and the differences between the estimated transformation and the ground truth.

2. Related Work

Current methods used to align two input point clouds can be classified according to the approaches that they apply to find the correspondences that are used for motion estimation.

2.1. Correspondences Based on 3D Features

Algorithms in this category are prevalent in point cloud registration. They focus on using/defining different features of local or global points to establish putative correspondences between the key points extracted from raw point clouds [14], and then use closed-form solutions or robust optimization algorithms to realize the transformation [15].

SIFT3D [16] is an extension of the original 2D SIFT detector in which the difference between Gaussian pyramids is used to find the key points. Steder et al. [17] exploit the NARF detector to select points at which the surface is stable and that features a sufficient amount of change in the local neighborhood. It is also important to identify general and robust 3D feature descriptors. The point feature histogram (PFH) [18] and the fast point feature histogram (FPFH) [19] represent relationships between a given point and its neighbors. Similarly, some global shape descriptors have been proposed to align point clouds, such as Hough transform descriptors [20], the spherical entropy image [21], and viewpoint descriptors [22].

By contrast, recent studies have used data-driven deep neural networks to learn feature descriptors from large-scale datasets. Zeng et al. [5] propose a 3D convolution neural network (3DMatch) that uses the local volumetric region around a given key point to compute feature descriptors for it. Building on sparse convolutions, FCGF [23] achieves a performance similar to the best patch-based descriptors [24], while being orders of magnitude faster. D3feat [25] leverages a 3D fully convolution network to predict both a detection score and a description feature for each point. SpinNet [26] introduces a Spatial Point Transformer and 3D cylindrical convolution to extract local features which are rotationally invariant. Recently, Predator [7] utilizes attention mechanisms to aggregate contextual information to learn more discriminative feature descriptor. To simplify the architecture, REGTR [27] and JAR-Net [28] directly predict clean correspondences using multiple transformer layers.

Many effective strategies have been developed to reject outliers in the correspondences. RANSAC [29] and its variants are commonly used to this end. In the recent, graph-based methods have been widely applied to model fitting. Lin et al. [30] propose a method based on co-clustering on bipartite graphs to estimate model in data contaminated with outliers and noise.

The above-mentioned methods have focused on individual local patches, where this significantly increases computational cost. Moreover, they are usually tailored to specific tasks, and thus are not sufficiently flexible and descriptive for complicated and novel scenarios.

2.2. Handcrafted Registration Methods

The well-known iterative closest point [9] method is considered a milestone in point cloud registration. It uses iterative methods to establish correspondences by using the point-to-point distance and updating the transformation. However, motion estimation by it is sub-optimal owing to a poor initial transformation and the large amount of requisite calculation [22].

Several studies have focused on overcoming the shortcomings of ICP. Lecoy et al. [31] propose a method to compare variants of ICP, and use it to examine a combination of variants optimized for speed. Billings et al. [32] introduce the IMLP to robustly solve the problem of rigid-body alignment. The PLICP [10] uses the point-to-line metric and an exact closed form for minimizing it. It is more precise and requires fewer iterations than the IMLP. The point-to-plane ICP [33] minimizes the orthogonal distance between points in one point set and the corresponding local planes in the other to estimate the motion transform. Motion estimation by using the plane-to-plane distance is more efficient for human-made scenes than point-to-point and point-to-plane correspondences [11]. Nevertheless, most ICP variant still require relatively good initialization to avoid converging to bad local minima. A notable exception, the Go-ICP [34] method is based on a branch-and-bound scheme that searches the entire 3D motion space.

2.3. Deep Point-Cloud Registration

Instead of combining learned feature descriptors with robust optimization at inference time, some works incorporate the entire pose estimation into the training pipeline. PointNetLK [35] combines a PointNet-based global feature descriptor [36] with the the Lucas & Kanade (LK) algorithm and estimates relative transformation. DeepVCP [37] innovatively generates corresponding points based on learned matching probabilities among a group of candidates. Deep Closest Point (DCP) [38] proposes a learned version of Iterative Closest Point (ICP) [9] and utilizes soft correspondences on learned pointwise features to compute the rigid transform. Pointdsc [39] formulates a nonlocal feature aggregation module and a differentiable spectral matching module to prune outlier correspondences. OMNet [40] learns overlapping masks to reject non-overlapping regions, which converts the partial-to-partial registration to the registration of the same shape.

Many methods achieve the expected score on the training set, but them have relatively poor generalization ability across the unseen dataset. Moreover, a few recent studies have proposed methods that either require computing the density of points or rely on the region of a high overlap between point clouds, and thus are time consuming.

3. Materials and Methods

3.1. Problem Formulation and Classic ICP Revisited

Let the two sets of points $P = \{\mathbf{p}_1, \cdots, \mathbf{p}_M\}$ and $Q = \{\mathbf{q}_1, \cdots, \mathbf{q}_N\}$, which represent point coordinates in $\mathbb{R}^3$, be the source point set and the target point set, respectively. The goal of point cloud registration is to estimate a rigid transformation $\mathbf{T}$ (represented by using a rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$) on P to align it with Q. This minimizes the following l_2-norm error E:

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{M} e_i(\mathbf{R}, \mathbf{t}) + I_{SO(d)}(\mathbf{R}), \tag{1}$$

where $e_i(\mathbf{R}, \mathbf{t}) = \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|^2$ is the per-point residual error between the transformed source point $\mathbf{R}\mathbf{p}_i + \mathrm{t}$ and the closest target point $\mathbf{q}_i$, and $I_{SO(d)}(\cdot)$ is an indicator function that requires that $\mathbf{R}$ be a rotation matrix:

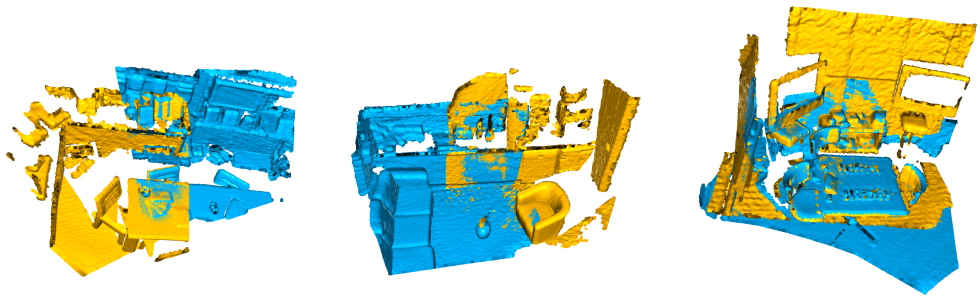
$$I_{SO(d)}(\mathbf{R}) = \begin{cases} 0, & \text{if } \mathbf{R}^T\mathbf{R} = \mathbf{I} \text{ and } \det(\mathbf{R}) = 1 \\ +\infty & \text{others.} \end{cases} \tag{2}$$

Equations (1) and (2) represent a well-known chicken-and-egg problem that cannot be solved trivially. The ICP algorithm is frequently used to solve the problem of registration.

Given an initial guess for a rigid transformation T^0, the ICP algorithm repeats the following two steps:

1. Find the corresponding closest point $\mathbf{q}_i^{(k)} \in Q$ for each point $\mathbf{p}_i^{(k)} \in P$ based on the last iterative transformation $\mathbf{T}^{(k)}$.
2. Update the transformation by minimizing the l_2-norm error E between the corresponding points, and render the result as the transformation $\mathbf{T}^{(k+1)}$.

Such an iterative method can ensure that the transformation matrix can converge to a local optimum. Despite its simplicity, the classical ICP converges slowly to a local minimum because of its linear convergence and search speed. In addition, in case of a low overlap between the source and target point sets, ICP generates an erroneous alignment. Point clouds with different ratios of overlap are shown in Figure 2.



(**a**) (**b**) (**c**)

Figure 2. Point clouds with different overlap ratios. The overlap is computed relative to the source and target fragments. When matching point clouds with a low overlap, it is difficult to find the correct corresponding points. (**a**) overlap ratio = 0.2. (**b**) overlap ratio = 0.5. (**c**) overlap ratio = 0.8.

3.2. Feature Point Extraction

LiDAR measures the distance to a given object by using a laser and calculating the distance traveled by the beam. Current sensors based on LiDAR can be classified into mechanical multi-line spinning LiDAR sensors and solid-state LiDAR sensors. The scanning of point clouds by using different LiDAR sensors yields discrepancies in the results. Solid-state LiDAR sensors usually have a small field of view (FoV) and an irregular scanning pattern that lead to a high angular resolution and dense point clouds in the scene. On the contrary, mechanical LiDAR has a line-scan pattern that enables it to obtain a sparse point cloud by spinning the laser array at a high frequency.

In general, registering the raw point clouds captured by LiDAR sensors is too computationally burdensome. In particular in case of scenes with a small overlap, the iterations are prone to converging to the local optimum. The real-world environment typically consists of a line–plane geometry. Therefore, we select feature points that are along sharp edges and on planar surface patches to reduce the amount of requisite computation and eliminate noise. The distortion in motion is eliminated from the data by using linear interpolation.

For the scanning of point clouds by multi-line spinning LiDAR, let $\mathbf{p}_i$ be a point, $\mathbf{p}_i \in \mathcal{S}$, and let $\mathcal{S}$ be the set of consecutive points of $\mathbf{p}_i$ on the same scanning line. To differentiate between edge and planar points, we compute the local smoothness of the candidate point $\mathbf{p}_i$ by searching its neighboring points:

$$\varrho_i = \frac{1}{\lambda^2} \sum_{\mathbf{p}_j \in \mathcal{S}_i, j \neq i} \left(\|\mathbf{p}_j\| - \|\mathbf{p}_i\| \right) \tag{3}$$

with

$$\mathcal{S}_i = \left\{ \mathbf{p}_j \in \mathcal{S} \mid j \in [i - \lambda, i + \lambda] \right\} \tag{4}$$

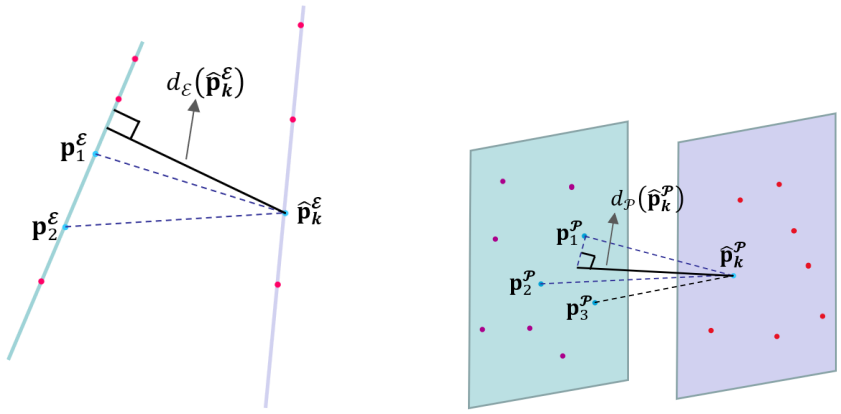
where λ is the radius of search around point $\mathbf{p}_i$. The points in $\mathcal{S}$ are sorted based on their local smoothness. The value of ϱ_i of points along the edges is higher than that of points on the plane.

To select feature point clouds from the evenly distributed raw point set, we separate a scan line into four sub-regions that can provide a maximum of two edge points and five planar points. A point with local smoothness ϱ_i can be selected as an edge or a planar point only if ϱ_i is larger or smaller than a pre-defined threshold and the number of points selected is smaller than the maximum number possible.

To scan the point clouds by using the solid-state LiDAR, we need different strategies to extract the feature points owing to the irregular pattern and density of the point clouds. Random sample consensus (RANSAC) can be applied to extract the features. The classic RANSAC collects point clouds P by scanning a scene, including edge-related and planar features. We select a random sample subset S of the original data, and choose n points to form an initial model M with the objective function C. The points in the complementary set fit model M, with the initial points forming the set of inliers. If the number of inliers exceeds the standard value, we use the set of inliers to construct the new model M^* by using the least-squares method. This process is continued until the maximum interior point set has been obtained, at which time the point set is considered to satisfy the requirements of the model.

Because random point sampling reduces the efficiency of the algorithm, the subset of random samples S is initially judged. In case the initial points can form a planar or an edge-based model, the number of point clouds in the sphere of radius r_0, centered at the initial point, that are needed to form the model are assessed. If this number is insufficient, the initial point is randomly selected again.

The edge-based and planar points are selected from the point cloud as shown in Figure 3.



(**a**) (**b**)

Figure 3. A feature in current scan and the corresponding feature points in the last scan: (**a**) edge-related feature; (**b**) planar feature.

3.3. Error Model

As mentioned above, registering raw point clouds is inefficient, and is sensitive to noise and outliers. We thus use matching edge-related and planar points from raw point clouds in the feature space. We assume that $\boldsymbol{\mathcal{E}}_s$ and $\boldsymbol{\mathcal{E}}_t$ are the sets of all edge-related points in the source and the target point sets, respectively. To increase the search speed, the target edge-related point set $\boldsymbol{\mathcal{E}}_t$ is built by using the KD tree.

Figure 3a shows the correspondence between edge-related features in the source and target point clouds. For each edge-related point $\mathbf{p}_k^{\mathcal{E}} \in \boldsymbol{\mathcal{E}}_s$, we obtain the transform $\widehat{\mathbf{p}}_k^{\mathcal{E}} = \mathbf{R}^0\mathbf{p}_k^{\varepsilon} + \mathbf{t}^0$ based on a guess regarding the initial transformation $\mathbf{t}^0$. We can search two nearest edge-related feature points $\mathbf{p}_1^{\mathcal{E}}$ and $\mathbf{p}_1^{\mathcal{E}}$ from the target edge-related feature set $\boldsymbol{\mathcal{E}}_t$. Thus, the point-to-edge distance $d_{\varepsilon}(\widehat{\mathbf{p}}_k^{\mathcal{E}})$ between $\widehat{\mathbf{p}}_k^{\mathcal{E}}$, and the edge crossing $\mathbf{p}_1^{\mathcal{E}}$ and $\mathbf{p}_2^{\mathcal{E}}$ is computed as:

$$d_{\varepsilon}\left(\widehat{\mathbf{p}}_k^{\mathcal{E}}\right) = \frac{\left|\left(\widehat{\mathbf{p}}_k^{\mathcal{E}} - \mathbf{p}_1^{\mathcal{E}}\right) \times \left(\widehat{\mathbf{p}}_k^{\mathcal{E}} - \mathbf{p}_2^{\mathcal{E}}\right)\right|}{\left|\left(\mathbf{p}_1^{\mathcal{E}} - \mathbf{p}_2^{\mathcal{E}}\right)\right|} \tag{5}$$

where $|\cdot|$ represents the modules of the vector and $\times$ is the cross-product of two vectors. Only when the point-to-edge distance is shorter than the pre-set threshold is the model optimized.

Similarly, we denote by $\boldsymbol{\mathcal{P}}_s$ and $\boldsymbol{\mathcal{P}}_t$ the set of all planar features in the source and the target point sets, respectively, in Figure 3b. For each planar feature $\mathbf{p}_k^{\mathcal{P}} \in \boldsymbol{\mathcal{P}}_s$, it is necessary to search three planar points nearest to its transform $\widehat{\mathbf{p}}_k^{\mathcal{P}} = \mathbf{R}^0\mathbf{p}_k^{\mathcal{P}} + \mathbf{t}^0$ by estimating a plane in 3D space. We select the points $\mathbf{p}_1^{\mathcal{P}}$, $\mathbf{p}_2^{\mathcal{P}}$, and $\mathbf{p}_3^{\mathcal{P}}$ from the target planar feature set. The point-to-plane distance $d_{\mathcal{P}}\left(\widehat{\mathbf{p}}_k^{\mathcal{P}}\right)$ between $\mathbf{p}_k^{\mathcal{P}}$, and the edge crossing $\mathbf{p}_1^{\mathcal{P}}$, $\mathbf{p}_2^{\mathcal{P}}$ and $\mathbf{p}_3^{\mathcal{P}}$ is then computed as:

$$d_{\mathcal{P}}\left(\widehat{\mathbf{p}}_k^{\mathcal{P}}\right) = \frac{\left|\left(\widehat{\mathbf{p}}_k^{\mathcal{P}} - \mathbf{p}_2^{\mathcal{P}}\right)^{\mathrm{T}} \cdot \left(\mathbf{p}_1^{\mathcal{P}} - \mathbf{p}_2^{\mathcal{P}}\right) \times \left(\mathbf{p}_1^{\mathcal{P}} - \mathbf{p}_3^{\mathcal{P}}\right)\right|}{\left|\left(\mathbf{p}_1^{\mathcal{P}} - \mathbf{p}_2^{\mathcal{P}}\right) \times \left(\mathbf{p}_1^{\mathcal{P}} - \mathbf{p}_3^{\mathcal{P}}\right)\right|} \tag{6}$$

Only when the distance is shorter than the pre-set threshold is it considered. The final rigid transformation can be estimated by minimizing the point-to-edge distance and the point-to-plane distance:

$$\arg\max_{\mathbf{R},\mathbf{t}} \frac{1}{N_{\mathcal{E}}} \sum_{i=1}^{N_{\mathcal{E}}} d_{\varepsilon}(\widehat{\mathbf{p}}_k^{\varepsilon}) + \frac{1}{N_{\mathcal{P}}} \sum_{i=1}^{N_{\mathcal{P}}} d_{\mathcal{P}}\left(\widehat{\mathbf{p}}_k^{\mathcal{P}}\right) + I_{SO(d)}(\mathbf{R}) \tag{7}$$

As with ICP, we can use an iterative approach and the Gauss–Newton optimization to solve the non-linear optimization problem. However, the cost and efficiency of the calculations need to be considered.

3.4. Anderson Acceleration for Fixed-Point Problem

Anderson Acceleration: By assuming a fixed-point iteration $g^{(k+1)} = G\left(g^{(k)}\right)$, we define its residual function $f(g) = G(g) - g$, and $f^{(k)} = G\left(g^{(k)}\right) - g^{(k)}$. According to its definition, a fixed point g^* of the mapping $G(\cdot)$ satisfies $f(g^*) = 0$, and can be solved by fixed-point iteration. To accelerate convergence, Anderson acceleration utilizes the latest iteration $g^{(k)}$ as well as the preceding m iterations $g^{(k-m)}, g^{(k-m+1)}, \cdots, g^{(k-1)}$ to derive a new iteration $g^{(k+1)}$ that converges more quickly to a fixed point [41]:

$$g^{(k+1)} = G\left(g^{(k)}\right) - \sum_{j=1}^{m} \theta_j^* \left(G\left(g^{(k-j+1)}\right) - G\left(g^{(k-j)}\right)\right) \tag{8}$$

where $\left(\theta_1^*, \cdots, \theta_m^*\right)$ is solution to the following linear least-squares problem:

$$\left(\theta_1^*, \cdots, \theta_m^*\right) = \arg\min \left\| f^{(k)} - \sum_{j=1}^{m} \theta_j \left(f^{(k-j+1)} - f^{(k-j)}\right) \right\|^2 \tag{9}$$

We note several properties of the ICP that make it computationally expensive and challenging to find the rigid transformation of Equation (7) in order to accelerate convergence by using higher-order methods.

We write our target function, explained in Section 2.3, as a fixed-point iteration of the transformation variables $\mathbf{R}$ and $\mathbf{t}$:

$$\left(\mathbf{R}^{(k+1)}, \mathbf{t}^{(k+1)}\right) = G\left(\mathbf{R}^{(k)}, \mathbf{t}^{(k)}\right) \tag{10}$$

where

$$G\left(\mathbf{R}^{(k)}, \mathbf{t}^{(k)}\right) = \arg\max_{\mathbf{R},\mathbf{t}} \frac{1}{N_\varepsilon} \sum_{i=1}^{N_\varepsilon} d_\varepsilon \left(\mathbf{R}^{(k)} \mathbf{p}_k^{\mathcal{E}} + \mathbf{t}^{(k)}\right) + \frac{1}{N_\mathcal{P}} \sum_{i=1}^{N_\mathcal{P}} d_\mathcal{P} \left(\mathbf{R}^{(k)} \mathbf{p}_k^{\mathcal{P}} + \mathbf{t}^{(k)}\right) + I_{SO(d)}(\mathbf{R}) \tag{11}$$

d_ε denotes the distance between $\mathbf{R}^{(k)} \mathbf{p}_k^{\mathcal{E}}$ and the closest edge to it, and $d_\mathcal{P}$ denotes the distance between $\mathbf{R}^{(k)} \mathbf{p}_k^{\mathcal{P}}$ and the plane closest to it.

However, we cannot directly apply Anderson acceleration to the mapping $G(\cdot)$ because the new value of $\mathbf{R}$ computed by Anderson acceleration is an affine combination of rotation matrices, where $\mathbf{R}^T\mathbf{R} \neq \mathbf{I}$ and $\det(\mathbf{R}) \neq 1$. We can parameterize a rigid transformation $(\mathbf{R}, \mathbf{t})$ by using another set of variables $\mathbf{X}$. Equation (9) can then be re-written as:

$$\mathbf{X}^{(k+1)} = \bar{G}\left(\mathbf{X}^{(k)}\right) \tag{12}$$

When applying Anderson acceleration, we can then parameterize the transformation to obtain $\mathbf{X}^{(k)}$. When computing the optimization function, we can recover the rotation matrix $\mathbf{R}^{(k)}$ and the translation vector $\mathbf{t}^{(k)}$ from the variable $\mathbf{X}^{(k)}$.

The parameterization of a rigid transformation involves concatenating the translation vector and the parameterized rotation matrix, where this can be represented as Euler angles or unit quaternions in $\mathbf{R}^4$ [42]. However, the Euler angle has singularities called gimbal lock and the affine combination of unit quaternions does not result in a unit quaternion. Nonetheless, all rigid transformations in $\mathbf{R}^4$ form the special Euclidean group $SE(3)$, which

is a Lie group that gives rise to a Lie algebra $\mathfrak{se}(3)$ [43], and Lie algebras are closed under addition. Thus, we define $\mathbf{X} = [\rho, \phi] \in \mathfrak{se}(3)$ and recover the transformation matrix from it:

$$\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0^T & 1 \end{bmatrix} = \exp(\mathbf{X}^\wedge) \tag{13}$$

where $\mathbf{X}^\wedge$ converts a 6D vector into a 4×4 matrix by

$$\mathbf{X}^\wedge = \begin{bmatrix} [\rho]_\times & \phi \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{14}$$

where $[\cdot]_\times$ is the skew matrix of a 3D vector. We can then execute Anderson acceleration on the Lie algebra of the transformation matrices. To avoid the instability and stagnation of Anderson acceleration, we accept the accelerated value as the new value only if it reduces the target function compared with the value in the previous iteration [44].

We can use the left perturbation scheme and apply increment on the Lie group to solve the non-linear optimization problem (Equation (7)) by using the Gauss–Newton algorithm. The left perturbation model can be formulated as follows:

$$\mathbf{J}_p = \frac{\partial \mathbf{T}\mathbf{p}}{\partial \delta \mathbf{X}} = \lim_{\delta \mathbf{X} \to \mathbf{0}} \frac{(\exp(\delta \mathbf{X}^\wedge) \cdot \mathbf{T}\mathbf{p} - \mathbf{T}\mathbf{p})}{\delta \mathbf{X}} = \begin{bmatrix} \mathbf{I}_{3\times 3} & -[\mathbf{T}\mathbf{p}]_\times \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \end{bmatrix} \tag{15}$$

Note that $[\mathbf{T}\mathbf{p}]_\times$ transforms the 4D point expression $\{x, y, z, 1\}$ into the 3D point expression $\{x, y, z\}$ before calculating the skew matrix. Thus, the Jacobian matrix of the point-to-edge distance $d_\varepsilon(\mathbf{p}_k^{\mathcal{E}})$ is defined by

$$\mathbf{J}_\varepsilon = \frac{\partial d_\varepsilon(\widehat{\mathbf{p}}_k^{\mathcal{E}})}{\partial \delta \mathrm{X}} = \frac{\partial d_\varepsilon(\widehat{\mathbf{p}}_k^{\mathcal{E}})}{\partial \widehat{\mathbf{p}}_k^{\mathcal{E}}} \frac{\partial \widehat{\mathbf{p}}_k^{\mathcal{E}}}{\partial \delta \mathrm{X}} = \frac{\partial d_\varepsilon(\widehat{\mathbf{p}}_k^{\mathcal{E}})}{\partial \widehat{\mathbf{p}}_k^{\mathcal{E}}} \mathbf{J}_p \tag{16}$$

Similarly, the Jacobian matrix of the point-to-edge distance $d_\mathcal{P}(\widehat{\mathbf{p}}_k^{\mathcal{P}})$ can be written as:

$$\mathbf{J}_\mathcal{P} = \frac{\partial d_\mathcal{P}(\widehat{\mathbf{p}}_k^{\mathcal{P}})}{\partial \widehat{\mathbf{p}}_k^{\mathcal{P}}} \mathbf{J}_p \tag{17}$$

This formulation has several advantages, such as the sorting of the rotation in a singularity-free format and unconstrained optimization.

4. Experiments and Results

In this section, we evaluate the proposed method and justify our design-related choices on empirically acquired point clouds. We used RealSense L515 to scan indoor scenes. It had a solid-state LiDAR with a small FoV and a viewing angle of $70° \times 55°$. The outdoor dataset was assembled from outdoor scenes in the KITTI odometry dataset. Our method was coded in C++, and implemented on Ubuntu 18.04 and ROS Melodic.

4.1. RealSense L515 Data

In our experiments, we used the hand-held Intel RealSense L515 with a frequency of update of 30 Hz to scan the indoor scenes. The ground truth was provided by a VICON system.

We collect the point clouds of indoor scenes. And by means of segmentation and transformation, we add its counterpart in which we consider scan pairs with overlaps between 100 and 10%. For fairness, we made sure that the same pair of source and template point clouds with different overlapping regions were used to test. We compare our methods to recent registration methods: handcrafted methods including ICP [9], GICP [33], NDT [45] and HMRF-ICP [46], feature-based method including FPFH [19], deep point-cloud registration methods including PointNetLK [35] and Predator [7]. The dataset built from raw scans contained 1000 point cloud pairs uniformly distributed over the rates of overlap.

The relative translational error (RTE), relative rotational error (RRE), and run time were used as evaluation metrics, as shown in Table 1. For registration methods that are not end-to-end, the run time includes feature extraction and transformation matrix calculation.

Our method outperformed most of the other optimization-based approaches tested, including the ICP, GICP, NDT and HMRF-ICP. And FPFH focused on the feature of local point cloud, and this led to many inaccurate correspondence pairs, especially in case of a low rate of overlap. For deep learning-based approaches, Predator had achieved satisfactory performance except runtime. Due to the use of edge-related and planar features, our method was able to register all the scanned pairs. Although some of the indoor scenes considered here contained curved surfaces, planar and edge structures still dominated them such that our method successfully registered these dense point clouds. Even with low overlap, we still established line-plane geometry correspondence accurately and easily as obvious structure features. And then use optimization methods to get registration transformation with Anderson Acceleration.

Table 1. Average evaluation of our method on the indoor dataset.

	Relative Translation Error [cm]	**Relative Rotation Error [°]**	**Run Time for Processing [s]**
ICP [9]	7.32	0.42	28.71
GICP [33]	3.24	0.31	142.2
NDT [45]	38.92	12.83	23.21
HMRF-ICP [46]	6.31	0.89	13.41
FPFH [19]	9.12	1.39	16.96 + 3.22
PointNetLK [35]	12.51	2.31	21.41
Predator [7]	3.18	0.22	7.19
Our methods	3.25	0.21	2.25 + 2.31

Figure 4 shows that we registered the two point sets scanned by L515 with overlaps of 20%, 60%, and 90%. Thanks to the line-plane geometry, our registration method managed to register all these scan pairs. Though some scenes partially consist of curved surfaces, the extractive feature points still provide sufficient information for a reliable registration.

Our method is capable of registering scans with different overlap. In Figure 5a, We found that filtering would reduce the accuracy of registration, so we tried our best to increase the proportion of sampling while ensuring the efficiency of the algorithm. We also tested our method to determine whether it focused on structural correspondences when the rate of overlap between the point sets decreased. We extracted test pairs by varying the completeness of the input point clouds from 100% to 10%. Remarkably, about 70% of the test pairs contained a line–plane geometry. Figure 5b shows that our method maintained a higher robustness in terms of registration than the other methods.

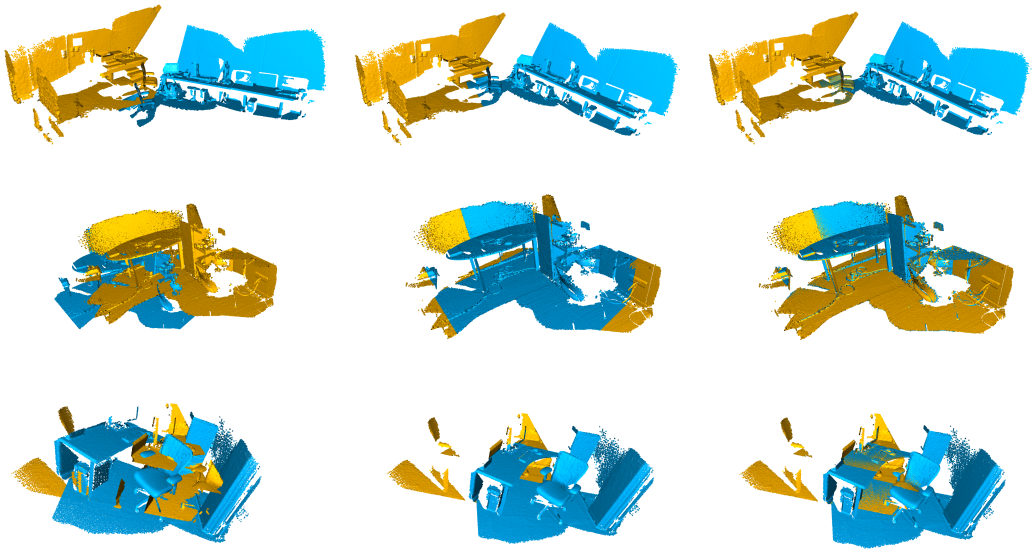
4.2. KITTI Odometry Dataset

The KITTI odometry dataset, acquired by using Velodyne-64 3D LiDAR scanners, is an outdoor dataset of sparse point clouds. It consists of 11 sequences of outdoors scans. We used the same evaluation metrics for it as for the RealSense L515 dataset. However, the KITTI odometry dataset does not have a low rate of overlap during the acquisition of the point cloud. Moreover, the point clouds were gravitationally aligned in this dataset such that the reference axis was aligned.

It is clear that the point clouds in the KITTI odometry dataset were significantly different from those in the RealSense L515 dataset because the former was mainly composed of large-scale, sparse, and partial LiDAR scans. In addition, its overlap ratio was approximately 100%, and its rotation around the x- and y-axes was roughly 0°.

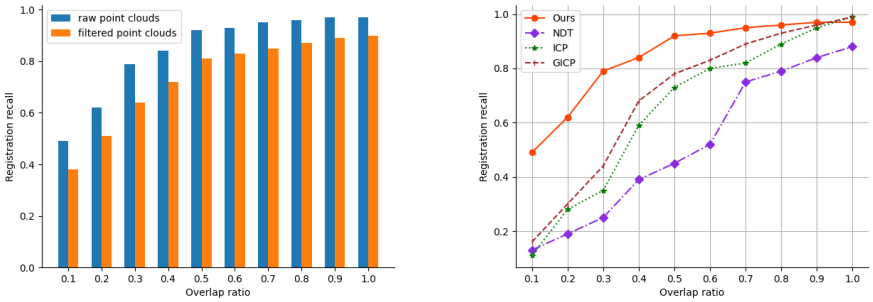
As shown in Table 2 and Figure 6, compared with the performance of the ICP and Predator algorithm, our method still achieved satisfactory results of registration on it.

Especially, the feature extractor and Anderson acceleration used for the fixed-point problem ensured its adequate performance in terms of computation time.



(**a**) (**b**) (**c**)

Figure 4. Example of results on the RealSense L515 dataset. (**a**) Input point clouds. (**b**) Ground-truth registration. (**c**) Our method.

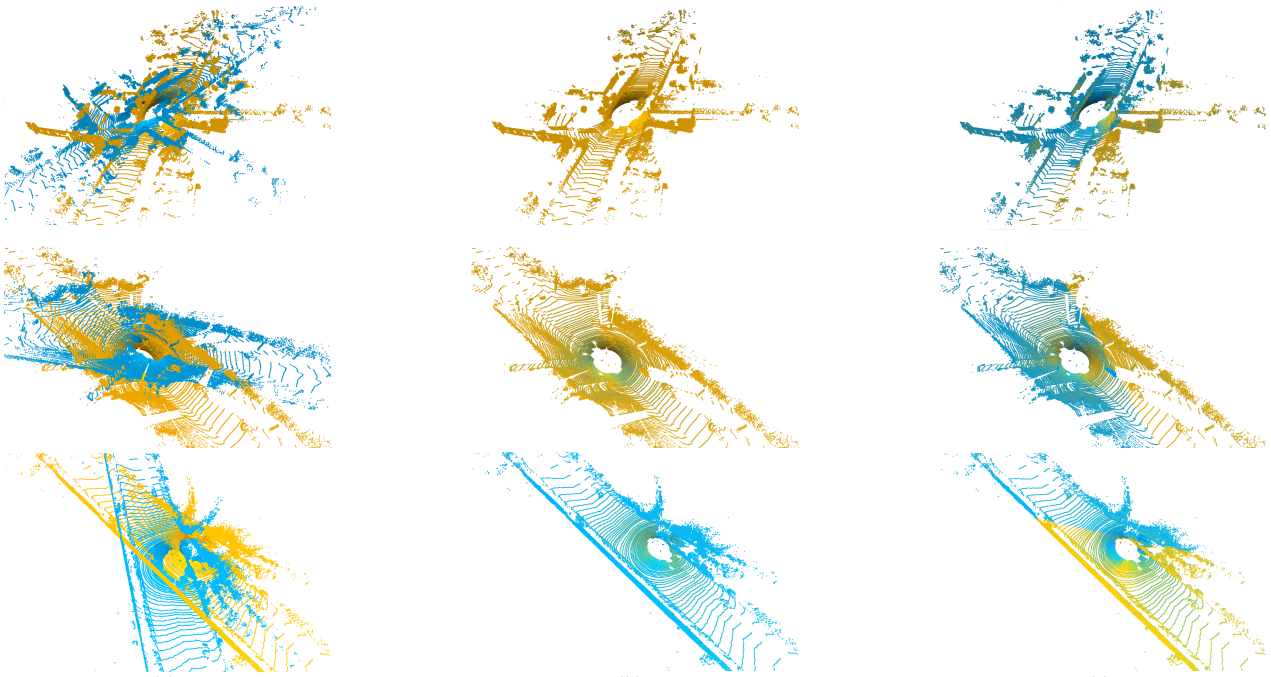


(**a**) (**b**)

Figure 5. (**a**) Our method avoided failure by increasing the fraction of sampled points because dense points are beneficial for extracting structural features. (**b**) As the rate of overlap decreased, the recall of registration decreased to a certain extent, but our method was more robust than the others at a low rate of overlap.

Table 2. Average values of the performance metrics of our method on the KITTI odometry dataset.

	Relative Translation Error [cm]	**Relative Rotation Error [°]**	**Run Time for Processing [s]**
ICP [9]	6.29	0.16	26.81
GICP [33]	4.92	0.31	172.6
NDT [45]	29.67	12.83	32.71
HMRF-ICP [46]	8.54	1.01	11.98
FPFH [19]	11.12	1.77	14.26 + 4.51
PointNetLK [35]	14.32	1.97	18.49
Predator [7]	5.12	0.19	5.29
Our methods	5.01	0.21	1.03 + 1.37



(**a**) (**b**) (**c**)

Figure 6. Example of results on the KITTI odometry dataset. (**a**) Input point clouds. (**b**) Ground-truth registration. (**c**) Our method.

5. Discussion

Since the traditional correspondence-based registration algorithm requires the local feature calculation, the feature contains the information of the local point cloud, which is indeed advantageous. However, experiments show that the edge area of the point cloud and the non-overlapping region of the scans causes inaccurate correspondence, which affects the quality of registration. At the same time, calculating the local features increases the time complexity of the algorithm. This paper uses high-level structural information of raw laser scans for point correspondence, and formulate a fixed-point problem by examining correspondences between feature points, which can be solved by the Lie derivative and Anderson Acceleration.

Actually, our high-level structural feature extraction is dedicated to registering point clouds of scenes that at least partially consist of planar and edge structures. Thus, the proposed method is especially suitable for registration scans of man-made environments. For scans of vegetation and scans of individual objects that consist of curved surfaces, the extraction of plane and edge feature will probably not perform as expected.

Our work can be extended by addressing its shortcomings discussed in detail. In the one hand, our approach relies strongly on the feature detection algorithm. These steps not only influence accuracy of results of our proposed method of transformation search, but also make up to $\geq$50% of algorithm runtime, suggesting that using a more sophisticated method for feature detection in terms of speed and accuracy is a key step to make our method effective and applicable. In the other hand, despite the Anderson acceleration is applied, we still need iteration to converge. In future work, the deep-learning-based method can be utilized to solve the fixed-point problem.

6. Conclusions

In this study, we propose an approach for the pairwise registration of point clouds with a low overlap as captured by mechanical LiDAR and solid-state LiDAR. The core of our method is a combination of planar features and edge-related features as well as the formulation of a fixed-point problem, with the aim of finding the relations between the point

clouds to identify their high-level characteristics and improve the speed of convergence. The main contributions of this paper are as follows:

- We propose a method to extract planar and edge-related feature points in data obtained from mechanical LiDAR and solid-state LiDAR.
- We formulate a non-linear optimization problem by examining structural correspondences between the planar and the edge-related points.
- Rewritten the problem as a fixed-point equation, we apply Anderson acceleration to speed up convergence and use the Lie algebra to represent a rigid transformation when solving the optimization function.

The results of simulations showed that our method achieves accurate results on scans of indoor and outdoor scenes. In future work, we plan to investigate the descriptors of feature points for scans without a line–plane structure and the solution of fixed-point problem.

Author Contributions: Methodology and software, C.Z.; validation and formal analysis, C.Z. and Y.Z.; writing—original draft preparation, C.Z.; writing—review and editing, X.C.; visualization, C.Z. and Y.Z.; supervision and project administration, X.C. and K.G.; All authors have read and agreed to the published version of the version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: The data presented in this study are openly available from KITTI at https://www.cvlibs.net/datasets/kitti/ (accessed on 15 September 2022), reference number [13].

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

Abbreviations

The following abbreviations are used in this manuscript:

ICP	Iterative Closest Point
LiDAR	Light Detection and Ranging
NDT	Normal Distribution Transform
SLAM	Simultaneous Localization and Mapping

References

1. Zhang, J.; Singh, S. LOAM: Lidar odometry and mapping in real-time. In Proceedings of the Robotics: Science and Systems, Berkeley, CA, USA, 12–16 July 2014; Volume 2, pp. 1–9.
2. Shan, T.; Englot, B. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4758–4765.
3. Monji-Azad, S.; Hesser, J.; Löw, N. A review of non-rigid transformations and learning-based 3D point cloud registration methods. *ISPRS J. Photogramm. Remote Sens.* **2023**, *196*, 58–72. [CrossRef]
4. Choi, C.; Taguchi, Y.; Tuzel, O.; Liu, M.Y.; Ramalingam, S. Voting-based pose estimation for robotic assembly using a 3D sensor. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, St. Paul, MN, USA, 14–18 May 2012; pp. 1724–1731.
5. 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, Honolulu, HI, USA, 21–26 July 2017; pp. 1802–1811.
6. Makovetskii, A.; Voronin, S.; Kober, V.; Tihonkih, D. Affine registration of point clouds based on point-to-plane approach. *Procedia Eng.* **2017**, *201*, 322–330. [CrossRef]
7. 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, Nashville, TN, USA, 20–25 June 2021; pp. 4267–4276.
8. Liu, Z.; Zhang, F.; Hong, X. Low-cost retina-like robotic lidars based on incommensurable scanning. *IEEE/ASME Trans. Mechatronics* **2021**, *27*, 58–68. [CrossRef]
9. Besl, P.J.; McKay, N.D. Method for registration of 3-D shapes. In Proceedings of the Sensor Fusion IV: Control Paradigms and Data Structures, Boston, MA, USA, 12–15 November 1991; Volume 1611, pp. 586–606.
10. Censi, A. An ICP variant using a point-to-line metric. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 19–25.

11. Khoshelham, K. Automated localization of a laser scanner in indoor environments using planar objects. In Proceedings of the 2010 International Conference on Indoor Positioning and Indoor Navigation, Zurich, Switzerland, 15–17 September 2010; pp. 1–7.
12. Van Nam, D.; Gon-Woo, K. Solid-state LiDAR based-SLAM: A concise review and application. In Proceedings of the 2021 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju Island, Republic of Korea, 17–20 January 2021; pp. 302–305.
13. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
14. Ran, H.; Liu, J.; Wang, C. Surface representation for point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 18942–18952.
15. Sheik, N.A.; Deruyter, G.; Veelaert, P. Plane-based robust registration of a building scan with its BIM. *Remote Sens.* **2022**, *14*, 1979. [CrossRef]
16. Stancelova, P.; Sikudova, E.; Cernekova, Z. 3D Feature detector-descriptor pair evaluation on point clouds. In Proceedings of the 2020 28th European Signal Processing Conference (EUSIPCO), Amsterdam, The Netherlands, 18–21 January 2021; pp. 590–594.
17. Steder, B.; Rusu, R.B.; Konolige, K.; Burgard, W. Point feature extraction on 3D range scans taking into account object boundaries. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 2601–2608.
18. Rusu, R.B.; Blodow, N.; Marton, Z.C.; Beetz, M. Aligning point cloud views using persistent feature histograms. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 3384–3391.
19. Rusu, R.B.; Blodow, N.; Beetz, M. Fast point feature histograms (FPFH) for 3D registration. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 3212–3217.
20. Zaharia, T.; Preteux, F.J. Hough transform-based 3D mesh retrieval. In Proceedings of the Vision Geometry X, San Diego, CA, USA, 29–30 July 2001; Volume 4476, pp. 175–185.
21. Sun, B. 3D Global Shape Descriptors Applied in Scan Registration. Ph.D. Thesis, Staats-und Universitätsbibliothek Hamburg Carl von Ossietzky, Hamburg, Germany, 2015.
22. Forstner, W.; Khoshelham, K. Efficient and accurate registration of point clouds with plane to plane correspondences. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 2165–2173.
23. Choy, C.; Park, J.; Koltun, V. Fully convolutional geometric features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8958–8966.
24. 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, Long Beach, CA, USA, 15–20 June 2019; pp. 5545–5554.
25. Bai, X.; Luo, Z.; Zhou, L.; Fu, H.; Quan, L.; Tai, C.L. D3feat: Joint learning of dense detection and description of 3d local features. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6359–6367.
26. Ao, S.; Hu, Q.; Yang, B.; Markham, A.; Guo, Y. Spinnet: Learning a general surface descriptor for 3d point cloud registration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 11753–11762.
27. 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, New Orleans, LA, USA, 18–24 June 2022; pp. 6677–6686.
28. Shi, Z.; Xiao, G.; Zheng, L.; Ma, J.; Chen, R. JRA-Net: Joint representation attention network for correspondence learning. *Pattern Recognit.* **2023**, *135*, 109180. [CrossRef]
29. Martínez-Otzeta, J.M.; Rodríguez-Moreno, I.; Mendialdua, I.; Sierra, B. RANSAC for Robotic Applications: A Survey. *Sensors* **2022**, *23*, 327. [CrossRef] [PubMed]
30. Lin, S.; Luo, H.; Yan, Y.; Xiao, G.; Wang, H. Co-Clustering on Bipartite Graphs for Robust Model Fitting. *IEEE Trans. Image Process.* **2022**, *31*, 6605–6620. [CrossRef] [PubMed]
31. Rusinkiewicz, S.; Levoy, M. Efficient variants of the ICP algorithm. In Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling, Quebec City, QC, Canada, 28 May–1 June 2001; pp. 145–152.
32. Billings, S.D.; Boctor, 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]
33. Ren, Z.; Wang, L.; Bi, L. Robust GICP-based 3D LiDAR SLAM for underground mining environment. *Sensors* **2019**, *19*, 2915. [CrossRef] [PubMed]
34. Yang, J.; Li, H.; Campbell, D.; Jia, Y. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 2241–2254. [CrossRef] [PubMed]
35. 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, Long Beach, CA, USA, 15–20 June 2019; pp. 7163–7172.
36. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.

37. Lu, W.; Wan, G.; Zhou, Y.; Fu, X.; Yuan, P.; Song, S. Deepvcp: An end-to-end deep neural network for point cloud registration. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 12–21.
38. 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, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3523–3532.
39. Bai, X.; Luo, Z.; Zhou, L.; Chen, H.; Li, L.; Hu, Z.; Fu, H.; Tai, C.L. Pointdsc: Robust point cloud registration using deep spatial consistency. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 15859–15869.
40. Xu, H.; Liu, S.; Wang, G.; Liu, G.; Zeng, B. Omnet: Learning overlapping mask for partial-to-partial point cloud registration. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 3132–3141.
41. Walker, H.F.; Ni, P. Anderson acceleration for fixed-point iterations. *SIAM J. Numer. Anal.* **2011**, *49*, 1715–1735. [CrossRef]
42. Diebel, J. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix* **2006**, *58*, 1–35.
43. Iserles, A.; Munthe-Kaas, H.Z.; Nørsett, S.P.; Zanna, A. Lie-group methods. *Acta Numer.* **2000**, *9*, 215–365. [CrossRef]
44. Potra, F.A.; Engler, H. A characterization of the behavior of the Anderson acceleration on linear problems. *Linear Algebra Its Appl.* **2013**, *438*, 1002–1011. [CrossRef]
45. Biber, P.; Straßer, W. The normal distributions transform: A new approach to laser scan matching. In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No. 03CH37453), Las Vegas, NV, USA, 27 October 27–1 November 2003; Volume 3, pp. 2743–2748.
46. Stechschulte, J.; Heckman, C. Hidden Markov random field iterative closest point. *arXiv* **2017**, arXiv:1711.05864.

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.