

Article

A Point Cloud Registration Algorithm Based on Weighting Strategy for 3D Indoor Spaces

Wenshan Lv ¹, Haifeng Zhang ^{1,2,*} , Weiren Chen ¹, Xiaoming Li ¹ and Shengtian Sang ¹

¹ MEMS Center, Harbin Institute of Technology, Harbin 150001, China; 22s021045@stu.hit.edu.cn (W.L.); chen_weir@126.com (W.C.); lixiaoming@hit.edu.cn (X.L.); stsang@hit.edu.cn (S.S.)

² Key Laboratory of Micro-Systems and Micro-Structures Manufacturing, Ministry of Education, Harbin 150001, China

* Correspondence: zhanghf@hit.edu.cn

Abstract: Point cloud registration is a technology that aligns point cloud data from different view-points by computing coordinate transformations to integrate them into a specified coordinate system. Many cutting-edge fields, including autonomous driving, industrial automation, and augmented reality, require the registration of point cloud data generated by millimeter-wave radar for map reconstruction and path planning in 3D environments. This paper proposes a novel point cloud registration algorithm based on a weighting strategy to enhance the accuracy and efficiency of point cloud registration in 3D environments. This method combines a statistical weighting strategy with a point cloud registration algorithm, which can improve registration accuracy while also increasing computational efficiency. First, in 3D indoor spaces, we apply PointNet to the semantic segmentation of the point cloud. We then propose an objective weighting strategy to assign different weights to the segmented parts of the point cloud. The Iterative Closest Point (ICP) algorithm uses these weights as reference values to register the entire 3D indoor space's point cloud. We also show a new way to perform nonlinear calculations that yield exact closed-form answers for the ICP algorithm in generalized 3D measurements. We test the proposed algorithm's accuracy and efficiency by registering point clouds on public datasets of 3D indoor spaces. The results show that it works better in both qualitative and quantitative assessments.

Keywords: point cloud registration; millimeter-wave radar; semantic segmentation; PointNet; ICP; objective weighting strategy



Citation: Lv, W.; Zhang, H.; Chen, W.; Li, X.; Sang, S. A Point Cloud Registration Algorithm Based on Weighting Strategy for 3D Indoor Spaces. *Appl. Sci.* **2024**, *14*, 5240. <https://doi.org/10.3390/app14125240>

Academic Editor: Eui-Nam Huh

Received: 13 May 2024

Revised: 3 June 2024

Accepted: 13 June 2024

Published: 17 June 2024



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

1. Introduction

Various registration algorithms have significantly improved accuracy and efficiency, but their primary application is in the registration of small components. When facing complex 3D indoor spaces, existing registration algorithms struggle to attain a good balance between accuracy and efficiency. A 3D indoor space, like a room, contains objects of different shapes and sizes with large quantities of point clouds, which pose a challenge for both accuracy and efficiency. The idea behind this paper is to use a statistical weighting strategy to come up with a point cloud registration algorithm that is both accurate and quick for 3D indoor places. Point cloud registration typically serves 3D map reconstruction or navigation in 3D indoor spaces, necessitating higher weights for significant targets that impact navigation, tracking, or localization, such as chairs on the ground, tables in the room, or walls obstructing the path. We can appropriately reduce weights for other types of targets to save computational resources during registration, thereby enhancing the overall efficiency of the algorithm and the registration performance of key targets. This paper uses PointNet [1] for semantic segmentation, marking different targets in the point cloud of 3D indoor spaces. A statistical weighting strategy is introduced where the segmented parts of the point cloud are treated as evaluation criteria and various attributes of the point

cloud are set as samples for evaluation. We then use the weights of different parts of the point cloud as reference values to determine the minimum metric of the nonlinear objective function in the Iterative Closest Point (ICP) registration algorithm. We use a new technique to obtain closed-form answers to the nonlinear objective function in a 3D generalized metric. This makes the registration more accurate. Finally, experiments with 3D indoor space datasets verify the accuracy and efficiency of the proposed algorithm, demonstrating significant improvements over the traditional point cloud registration algorithms.

We organize the remainder of this paper as follows: Section 2 introduces the weighting strategy, including the basic principles of objective weighting methods and a novel improvement to the calculation steps proposed in this paper. Section 3 describes the process of obtaining the ICP minimum metric solution with assigned weights. Section 4 analyzes the accuracy and efficiency of the proposed algorithm with registration experiments. Section 5 ultimately provides the conclusion and outlook.

Point cloud registration is a vital technique in computer vision. It introduces a spatial transformation that maps one image onto another and ensures a one-to-one correspondence between points in both images of the same spatial location to achieve information fusion [2]. Over the years, research into point cloud registration has made great progress. In 1992, Besl et al. proposed the Iterative Closest Point (ICP) algorithm [3]. By calculating the distances between corresponding points in the source point cloud and the target point cloud, a rotation–translation matrix RT is constructed. Then, the RT can be applied to transform the source point cloud and calculate the mean squared difference of the transformed points. If the mean squared difference satisfies a threshold condition, then the algorithm terminates. Otherwise, it continues iterating until the error meets the threshold condition or the maximum iteration count is reached. However, ICP is usually sensitive to initial values and may become stuck in local optima [4]. In 2009, Magnusson proposed the normal distributions transform (NDT) as a great 3D surface representation with applications in point cloud registration [5]. The NDT utilizes statistical probability methods to determine corresponding point pairs based on the normal distribution of point clouds, thereby calculating the transformation relationship between the source point cloud and the target point cloud. It is a discrete global registration algorithm that, in discretized coordinate space, maximizes the score of the probability density of the normal distribution calculated for the source points voxelized in the target point cloud. It employs probability density functions and optimal solving methods to compute the transformation relationship between the point clouds to be registered. Compared to ICP, the NDT significantly decreases consumption of hardware resources and enhances registration accuracy. However, since all point cloud data need to be discretized, the NDT has longer running times than ICP, leading to lower efficiency. In 2019, Yasuhiro et al. proposed the PointNetLK, a deep-learning-based point cloud registration algorithm [6]. In PointNetLK, PointNet is treated as a learnable “imaging” function and is utilized for point cloud registration by modifying the classic Lucas–Kanade (LK) algorithm used for image alignment to integrate with PointNet [7]. Unlike ICP and its variants, PointNetLK does not require large amounts of computation resources, offering advantages in terms of registration accuracy, robustness to initialization, and computational efficiency. On the other hand, due to the explicit encoding of the alignment process in the network architecture, PointNetLK shows significant generalization performance for unseen objects, which implies that the network only needs to learn point cloud representations without the necessity to learn the alignment task. In summary, existing point cloud registration algorithms can be broadly categorized into several types: iterative, optimization-based algorithms such as ICP and its variants Point-to-plane ICP, NICP, Go-ICP, and IMLS-ICP, etc. [8–11]; statistical, Gaussian distribution-based algorithms such as NDT and its variant MSKM-NDT [12]; and deep-learning-based algorithms such as PointNetLK, DeepVCP, and PCGAN [13,14].

In the last few years there have been other papers, such as [15], which provides an overview of traditional and learning-based methods for point cloud registration and outlier rejection. It discusses the limitations of existing approaches and sets the stage for the pro-

posed PointDSC model. Ref. [15] discusses traditional point cloud registration algorithms, such as RANSAC and its variants, which are popular outlier rejection methods but have drawbacks in terms of slow convergence and low accuracy in cases with large outlier ratios. It also mentions the limitations of 3D descriptors in traditional methods due to irregular density and a lack of useful texture, highlighting the importance of geometric consistency, such as length constraints under rigid transformation, which are commonly utilized by traditional outlier rejection algorithms. In addition, it presents learning-based outlier rejection methods, such as DGR and 3DRegNet, which formulate outlier rejection as an inlier/outlier classification problem. These methods use operators like sparse convolution and pointwise MLP for feature embedding, but they overlook the spatial compatibility between inliers. The section also highlights the limitations of existing outlier rejection methods, as well as the need for spatial consistency in outlier rejection for robust 3D point cloud registration. In the related work section of [15], certain algorithms are suggested, including RANSAC [16], spectral matching [17], DGR [18], and 3DRegNet [19]. This section gives a full picture of all the current methods used for point cloud registration and outlier rejection.

Ref. [20] focuses on the recent developments in point cloud registration, including same-source and cross-source methods, as well as the connections between optimization-based and deep learning techniques. The authors emphasize the need for a comprehensive survey to summarize recent advances in point cloud registration, as existing reviews mainly focus on conventional point cloud registration and deep learning techniques, with limited coverage of cross-source point cloud registration. The [20] paper's contributions include a comprehensive review of same-source point cloud registration, a review of cross-source registration, the development of a new cross-source point cloud benchmark, and the exploration of potential applications and future research directions in point cloud registration. It also discusses the challenges in cross-source point cloud registration, such as noise, outliers, density differences, partial overlap, and scale differences. It provides an evaluation of state-of-the-art registration algorithms on a cross-source dataset, highlighting the performance of optimization-based and deep learning methods. Refs. [17,21–25] provide a foundation for the survey and support the authors' claims and contributions in the field of point cloud registration. Ref. [20] introduces new cross-source datasets, such as 3DMatch, KITTI, ETHdata, and 3DCSR, for comparison experiments and benchmarking.

2. Weighting Strategy of Targets in Semantic Segmentation

2.1. The Objective Weighting Method

The objective weighting method utilizes objective and measurable metrics, rather than subjective judgments, for weight assignment to ensure fairness and rationality. In our task of point cloud registration, the coordinates, color information, and category labels of each point cloud data point are totally quantifiable. Therefore, using the objective weighting method for weight assignment aligns well with the requirements of the application background. Common objective weighting methods include the standard deviation method, the entropy weight method, and the criteria importance through intercriteria correlation (CRITIC) method [26–28]. The standard deviation method's basic idea is to calculate the variance S_i of the data under each indicator. We obtain the weight value for the i th indicator by dividing S_i by the sum of all S_i values. Larger variances result in higher weights. This method is relatively simple and easy to implement, effectively distinguishing between different indicators. However, it assumes that all indicators are equally important, and a larger variance only indicates higher discrimination ability for different scenarios, not necessarily the importance of the indicator [29]. The entropy weight method, on the other hand, calculates entropy weights based on the dispersion of data for each indicator using information entropy. Afterward, we apply certain corrections to these entropy weights, leading to more objective indicator weights. This method allows for weight adjustments and exhibits adaptability, but it cannot handle strong correlations between indicators; thus, there is a risk of unreasonable weight assignments [30]. The CRITIC method comprehensively assesses the objective weights of indicators based on the comparative intensity of

evaluation indicators and the conflict between indicators. It takes into account both the variability in indicator values and the correlation between indicators. Larger numerical values do not necessarily indicate greater importance; instead, the method utilizes the inherent objectivity of the data for scientific evaluation. Comparative intensity is defined as the magnitude of the differences in values between various evaluation schemes for the same indicator, expressed in the form of standard deviation. A larger standard deviation indicates greater fluctuation, implying larger differences in values between schemes, leading to higher weights. Correlation coefficients represent the conflict between indicators. If two indicators exhibit a strong positive correlation, indicating lower conflict, then the weights will be lower. The CRITIC method is used in this paper to give semantic segmentation targets weights. It takes into account the strong correlation between point cloud sets of the same target as well as positional relationships like occlusion and overlap between different targets that affect the registration process.

2.2. Definition of Alternatives and Criteria

In the classic CRITIC method, objects of different categories are defined as alternatives and performance metrics of different aspects of objects are defined as criteria [31]. This paper proposes a novel definition of alternatives and criteria, considering the practical significance of labels in semantic segmentation. The individual segmentation performance of each label influences the overall segmentation performance, and these influences positively correlate. A better local segmentation performance contributes to the improvement in overall segmentation quality. On the other hand, the labels have relative independence, allowing them to serve as different criteria for the same performance metric. Therefore, in the following sections of this paper, different performance metrics in semantic segmentation will be defined as alternatives, and different labels will be defined as criteria.

Based on the format of point cloud data in semantic segmentation, we define each point cloud as a vector,

$$\vec{p}_i = \{x_i, y_i, z_i, R_i, G_i, B_i, L_i\}, 0 \leq i \leq n \quad (1)$$

where x , y , and z are the 3D coordinates of point cloud, R , G , and B are the color information of each point cloud. L represents the label of each point cloud, indicating which class of target it belongs to, and n is the number of all the point clouds in the space.

We can divide the labels of point clouds used in this paper into the following categories based on the Stanford Large-Scale 3D Indoor Spaces Dataset (S3DIS) classification [32], as shown in Table 1.

Table 1. The labels of point clouds in 3D indoor spaces.

Class	Ceiling	Floor	Wall	Beam	Column	Window	Door	Table	Chair	Sofa	Case	Board	Clutter
No.	0	1	2	3	4	5	6	7	8	9	10	11	12

All the labels refer to common furniture found in indoor rooms, with the exception of the last one, “clutter”, which is added to categorize point clouds that cannot be reliably classified into a specific class.

We assume three performance metrics as alternatives for the weighting strategy, which include the importance (IM) of labels, the intersection-over-union (IoU) of labels, and the Class-Pixel-Accuracy (CPA) of labels [33,34]. We subjectively assign and quantitatively express the importance of labels in the following ways, as shown in Table 2:

Table 2. The importance of labels in 3D indoor spaces.

Class	Ceiling	Floor	Wall	Beam	Column	Window	Door	Table	Chair	Sofa	Case	Board	Clutter
IM	0.2	0.2	0.2	0.2	0.6	0.6	0.4	1	1	0.8	0.6	0.6	0.1

We set the maximum score of importance at 1. We assign a lower importance to objects such as ceilings, floors, and walls that serve as backgrounds. We assign a higher importance to furniture like tables, sofas, and chairs, which are numerous and highly utilized. Other types of furniture, with relatively infrequent use, fall between the two in terms of importance. We set the importance of clutter objects to only 0.1 in order to exclude them from the radar detection range.

We define the IoU and CPA based on a confusion matrix, a specific matrix that visualizes algorithm performance, where each column represents predicted values, and each row represents the actual categories [35]. In the semantic segmentation results of point clouds, we evaluate the segmentation performance for each class using a binary confusion matrix.

2.3. Steps of Calculation

A decision matrix is defined as follows:

$$M = \begin{pmatrix} M_{11} & \cdots & M_{1m} \\ \vdots & \ddots & \vdots \\ M_{n1} & \cdots & M_{nm} \end{pmatrix} \quad (2)$$

The variables n and m represent the quantity of alternatives and evaluation criteria, respectively. M_{ij} is the value of the j th criterion for the i th alternative, $1 \leq i \leq n$, $1 \leq j \leq m$. We specify n as 3 and m as 13 for our 3D indoor point cloud semantic segmentation task.

Because each criterion has different dimensions, normalizing the elements in the decision matrix is necessary to unify the units. This paper's criteria anticipate higher values, leading to the adoption of a forward normalization approach:

$$D_{ij} = \frac{M_{ij} - M_j^-}{M_j^+ - M_j^-}; i = 1, \dots, n; j = 1, \dots, m \quad (3)$$

where $M_j^- = \min(M_{1j}, M_{2j}, \dots, M_{nj})$, $M_j^+ = \max(M_{1j}, M_{2j}, \dots, M_{nj})$, and D_{ij} is the element M_{ij} of the decision matrix after normalization by dividing its distance from the maximum value by the length of the range.

The Pearson correlation coefficient between two criteria j and k is defined as follows:

$$r_{jk} = \frac{\sum_{i=1}^n (D_{ij} - \overline{D_j})(D_{ik} - \overline{D_k})}{\sqrt{\sum_{i=1}^n (D_{ij} - \overline{D_j})^2 \sum_{i=1}^n (D_{ik} - \overline{D_k})^2}} \quad (4)$$

where $\overline{D_j}$ and $\overline{D_k}$ are mean values of the j th and k th criteria after normalization by Equation (5):

$$\overline{D_j} = \frac{1}{m} \sum_{i=1}^n D_{ij}; j = 1, \dots, m \quad (5)$$

$$\overline{D_k} = \frac{1}{m} \sum_{i=1}^n D_{ik}; k = 1, \dots, m \quad (6)$$

The standard deviation of each criterion is defined:

$$\sigma_j = \sqrt{\frac{1}{m-1} \sum_{i=1}^n (D_{ij} - \overline{D_j})^2}; j = 1, \dots, m \quad (7)$$

where σ is the standard deviation, which serves to contrast the intensity among each criterion of the same alternative.

The measure of conflict among each criterion is defined:

$$p_j = \sum_{k=1}^m (1 - r_{jk}); j = 1, \dots, m \quad (8)$$

where p_j is the measure of conflict between the j th criterion and the others, which indicates the correlation among the criteria. The more positive the correlation is, the smaller the value of the inconsistency will be.

The amount of information of each criterion is defined:

$$Q_j = \sigma_j p_j \quad (9)$$

where Q_j is the amount of information of the j th criterion, which equals the product of standard deviation and measure of conflict. The greater Q_j is, the more information transmitted by the corresponding criterion, and the relatively higher importance it will have.

Finally, the weights of the criteria are defined as follows:

$$\omega_j = \frac{Q_j}{\sum_{j=1}^m Q_j}; j = 1, \dots, m \quad (10)$$

where ω_j indicates the weight of the j th criterion and will be applied in the subsequent ICP registration process.

2.4. Improvement of Steps

The criteria for labels in the semantic segmentation application background do not strictly adhere to a linear correlation. For instance, there is no inherent positive or negative correlation between the importance of a particular label and its CPA. The classical CRITIC method, unfortunately, fails to adequately consider this kind of fuzzy nonlinearity [36]. Therefore, we make some improvements to CRITIC based on the methodology proposed by Sharkasi and Rezakhah [37]. In the normalization procedure, we include fuzzy set theory and Hamming distance as similarity measures. Additionally, we use the distance correlation instead of the Pearson correlation to describe both the linear and nonlinear relationships between the criteria.

The following procedures allow for improvements: First, the elements of decision matrix D are converted into percentages.

$$M'_{ij} = \frac{M_{ij}}{\sum_{i=1}^n M_{ij}} \quad (11)$$

Then, the fuzzy form of elements of decision matrix D is defined as follows:

$$\begin{cases} D'_{ij}{}^L = M'_{ij} - \Delta d \\ D'_{ij}{}^H = M'_{ij} + \Delta d \end{cases} \quad (12)$$

where Δd is a very small positive number for all i and j values: $0 < D'_{ij} + \Delta d < 1$ and $0 < D'_{ij} - \Delta d < 1$. The boundaries of fuzzy RP points are defined as:

$$\begin{cases} L_j = \sum_{i=1}^n \frac{D'_{ij}{}^L}{n} \\ H_j = \sum_{i=1}^n \frac{D'_{ij}{}^H}{n} \end{cases} \quad (13)$$

where L_j is the lower bound of the fuzzy RP points and H_j is the upper bound of the fuzzy RP points. The normalization procedure can be described as:

$$D'_{ij} = \begin{cases} 1, & [D'_{ij}^L, D'_{ij}^H] \subseteq [L_j, H_j] \\ 1 - \frac{|D'_{ij}^L - L_j| + |D'_{ij}^H - H_j|}{2(D'_{ij}^{H+} - D'_{ij}^{L-})}, & [D'_{ij}^L, D'_{ij}^H] \not\subseteq [L_j, H_j] \end{cases} \quad (14)$$

where D'_{ij} is the element after normalization, which belongs to fuzzy set $[D'_{ij}^L, D'_{ij}^H]$ and $D'_{ij}^{L-} = \min(D'_{1j}^L, D'_{2j}^L, \dots, D'_{nj}^L)$, $D'_{ij}^{H+} = \max(D'_{1j}^H, D'_{2j}^H, \dots, D'_{nj}^H)$.

The distance correlation is defined as:

$$\rho_{jk} = \frac{\sum_{i=1}^n (D'_{ij} - \overline{D'_j})(D'_{ik} - \overline{D'_k})}{\sqrt{\frac{\sum_{i=1}^n (D'_{ij} - \overline{D'_j})^2}{n} * \frac{\sum_{i=1}^n (D'_{ik} - \overline{D'_k})^2}{n}}} \quad (15)$$

where $\overline{D'_j}$ and $\overline{D'_k}$ are mean values of the j th and k th criteria after normalization by Equation (16), the numerator refers to the covariance between the j th and k th criteria, $cov(j, k)$, and the denominator refers to the square root of the product of the variances of the two criteria, $var(j)$ and $var(k)$.

Finally, Equation (14) is used for normalization instead of Equation (3), and Equation (15) is used for the distance correlation coefficient ρ_{jk} instead of the Pearson correlation coefficient r_{jk} from Equation (6). In Equation (10), r_{jk} is replaced with ρ_{jk} to calculate the measure of conflict.

3. The ICP Algorithm with Weighting Strategy

3.1. Selection of Error Metric

In the ICP algorithm, an objective function is defined to minimize the measurement error. Generally, there are two metrics for error measurement: point-to-point and point-to-plane.

$$e_i = \sum_{i=1}^N \|(\mathbf{R}\mathbf{p}_i + \mathbf{t}_k - \mathbf{q}_i)^T\|^2 \quad (16)$$

$$e_i = \sum_{i=1}^N \|(\mathbf{R}\mathbf{p}_i + \mathbf{t}_k - \mathbf{q}_i)^T \mathbf{n}_i\|^2 \quad (17)$$

Equation (16) describes the point-to-point metric, which simply finds the minimum distance between the target point and the corresponding point on the source surface, where $\mathbf{R}$ is the rotational matrix, $\mathbf{p}_i$ is the source point cloud, $\mathbf{t}_k$ is the translation matrix, and $\mathbf{q}_i$ is the target point cloud. Equation (17) shows the point-to-plane metric. Its goal is to find the shortest path between the target point and the tangent plane of the source surface's corresponding point, where $\mathbf{n}_i$ is the vector that goes through the tangent plane of the source surface at $\mathbf{p}_i$.

As shown in Figure 1b, $\mathbf{p}_i'$ is a point on the destination surface and $\mathbf{p}_i$ is the corresponding point on the source surface. The vector $\mathbf{e}_i$ is the normal vector to the tangent plane of the source surface at $\mathbf{p}_i$, which takes the distance from $\mathbf{p}_i'$ to the plane tangent to $\mathbf{p}_i$ as the point-to-plane error metric. The objective function is to find a suitable rigid transformation that satisfies the minimum sum of all these distances.

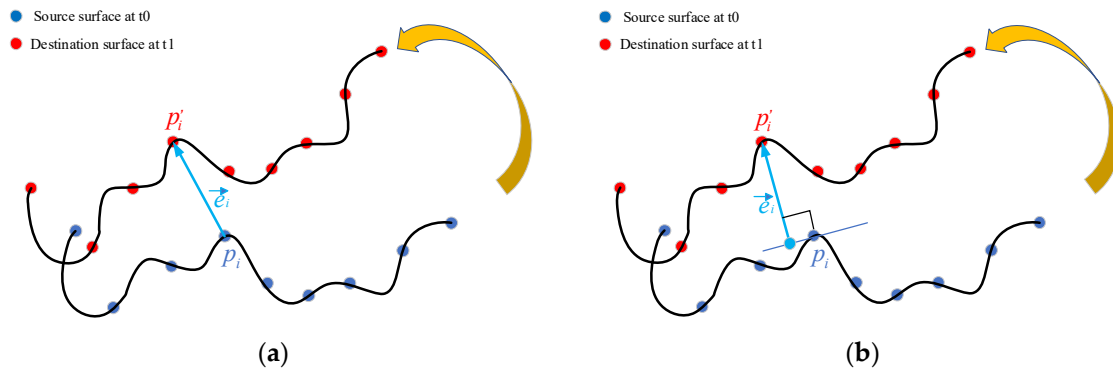


Figure 1. Error metrics: (a) point-to-point; (b) point-to-plane.

3.2. Derivation of Solution

The rigid transformation process of a point clouds is defined as:

$$q_i = R(\alpha_k, \beta_k, \gamma_k)p_i + t_k \quad (18)$$

where p_i is the i th point in the source point cloud and q_i is the corresponding point after rigid transformation. $R(\alpha_k, \beta_k, \gamma_k)$ is the rotation matrix with rotation angle α_k , β_k , and γ_k , and t_k is the corresponding translation value. k refers to the iterations of the ICP algorithm.

Based on Equation (18), the objective function can be written as:

$$F_{\min}(\alpha_{k+1}, \beta_{k+1}, \gamma_{k+1}, t_{k+1}) = \sum_{i=1} (n_i^T [R(\alpha_{k+1}, \beta_{k+1}, \gamma_{k+1})p_i + t_{k+1} - p'_i])^2 \quad (19)$$

Considering that $n_i, p_i, t_{k+1}, p'_i \in R^3$, and that $R(\alpha, \beta, \gamma)$ is a 3×3 matrix, the objective function can be denoted as:

$$\begin{aligned} F_{\min}(\alpha_{k+1}, \beta_{k+1}, \gamma_{k+1}, t_{k+1}) &= \sum_{i=1} (n_i^T [R(\alpha_{k+1}, \beta_{k+1}, \gamma_{k+1})p_i + t_{k+1} - p'_i])^T (n_i^T [R(\alpha_{k+1}, \beta_{k+1}, \gamma_{k+1})p_i + t_{k+1} - p'_i]) \\ &= \sum_{i=1} [R(\alpha_{k+1}, \beta_{k+1}, \gamma_{k+1})p_i + t_{k+1} - p'_i]^T n_i n_i^T [R(\alpha_{k+1}, \beta_{k+1}, \gamma_{k+1})p_i + t_{k+1} - p'_i] \end{aligned} \quad (20)$$

Then, the weight coefficient is introduced into Equation (18):

$$F_{\min}(\alpha_{k+1}, \beta_{k+1}, \gamma_{k+1}, t_{k+1}) = \sum_{i=1} [R(\alpha_{k+1}, \beta_{k+1}, \gamma_{k+1})p_i + t_{k+1} - p'_i]^T \omega_i n_i n_i^T [R(\alpha_{k+1}, \beta_{k+1}, \gamma_{k+1})p_i + t_{k+1} - p'_i] \quad (21)$$

where ω_i is the weight coefficient of the i th point whose value is calculated according to the weighting strategy in Section 3.

The traditional method for solving this nonlinear objective function is to linearize it, then use the least squares method to obtain a linearly approximate solution. These methods often have trouble obtaining exact closed-form solutions because of the nonlinearity introduced by the reference plane and the rigid transformation operator. This makes convergence times slower when working with large point cloud datasets. This paper proposes an exact closed-form solution to improve computational efficiency and enhance registration accuracy, by reducing the objective function into a matrix form with constraints and using Lagrange's multipliers.

In weighted PP-ICP, the rigid transformation is decomposed into many small transformations, each producing a minimal rotation angle along the rotation axis, called θ , which theoretically should be an infinitesimal value that satisfies:

$$0 < \theta \leq 1^\circ \quad (22)$$

For each small transformation, the rotation angles α_k , β_k , and γ_k have small values. Therefore, the rotation matrix $R(\alpha_k, \beta_k, \gamma_k)$ can be simplified based on the approximation rule proposed in [8]:

$$\cos \theta \approx 1, \sin \theta \approx 0, \theta^2 \approx 0 \text{ if } \theta \approx 0 \quad (23)$$

Then, the rotation matrix $R(\alpha_k, \beta_k, \gamma_k)$ can be simplified to

$$R(\alpha_k, \beta_k, \gamma_k) \approx \begin{bmatrix} 1 & \alpha_k \beta_k - \gamma_k & \alpha_k \gamma_k + \beta_k \\ \gamma_k & \alpha_k \beta_k \gamma_k + 1 & \beta_k \gamma_k - \alpha_k \\ -\beta_k & \alpha_k & 1 \end{bmatrix} \approx \begin{bmatrix} 1 & -\gamma_k & \beta_k \\ \gamma_k & 1 & -\alpha_k \\ -\beta_k & \alpha_k & 1 \end{bmatrix} \quad (24)$$

A six-dimensional vector is defined

$$\mathbf{x} = [t_x, t_y, t_z, \alpha, \beta, \gamma] \quad (25)$$

as the solution of the objective function. Given the i th point in the source point cloud

$$\mathbf{p}_i = [p_{i0}, p_{i1}, p_{i2}] \quad (26)$$

an auxiliary matrix is defined

$$\mathbf{A}_i = \begin{bmatrix} 1 & 0 & 0 & 0 & p_{i2} & -p_{i1} \\ 0 & 1 & 0 & -p_{i2} & 0 & p_{i0} \\ 0 & 0 & 1 & p_{i1} & -p_{i0} & 0 \end{bmatrix} \quad (27)$$

Then, Equation (21) can be written as

$$F_{\min}(\mathbf{x}) = \sum_i (\mathbf{A}_i \mathbf{x} - \mathbf{p}_{i1})^T \omega_i \mathbf{n}_i \mathbf{n}_i^T (\mathbf{A}_i \mathbf{x} - \mathbf{p}_{i1}) \quad (28)$$

by defining

$$\mathbf{B}_i = \omega_i \mathbf{n}_i \mathbf{n}_i^T \quad (29)$$

Ignoring constant terms, Equation (29) can be expanded as

$$F_{\min}(\mathbf{x}) = \mathbf{x}^T \left(\sum_i \mathbf{A}_i^T \mathbf{B}_i \mathbf{A}_i \right) \mathbf{x} + \left(\sum_i -2\mathbf{p}_i^T \mathbf{B}_i \mathbf{A}_i \right) \mathbf{x} \quad (30)$$

by defining

$$\mathbf{G} = \sum_i \mathbf{A}_i^T \mathbf{B}_i \mathbf{A}_i \quad (31)$$

$$\mathbf{H} = \sum_i -2\mathbf{p}_i^T \mathbf{B}_i \mathbf{A}_i \quad (32)$$

$$\mathbf{J} = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (33)$$

Finally, the objective function can be written as

$$F_{\min}(\mathbf{x}) = \mathbf{x}^T \mathbf{G} \mathbf{x} + \mathbf{H} \mathbf{x}, \text{ s.t. } \mathbf{x}^T \mathbf{J} \mathbf{x} = 0 \quad (34)$$

where the constraint is the approximate matrix form of $\alpha^2 + \beta^2 + \gamma^2 \approx 0$.

To solve the nonlinear minimization problem, Lagrange's multipliers are used for derivation. The Lagrange function is defined

$$L(\mathbf{x}) = \mathbf{x}^T \mathbf{G} \mathbf{x} + \mathbf{H} \mathbf{x} + \lambda \mathbf{x}^T \mathbf{J} \mathbf{x} \quad (35)$$

The necessary condition for optimization is derived as

$$\frac{\partial L}{\partial x} = 2x^T G + H + 2\lambda x^T J = 0^T \quad (36)$$

which is equivalent to

$$x = -(2G + 2\lambda J)^{-T} H^T \quad (37)$$

Equation (37) is substituted into the constraint of Equation (34) to obtain

$$H(2G + 2\lambda J)^{-1} J(2G + 2\lambda J)^{-T} H^T = 0 \quad (38)$$

where λ is the only unknown parameter. According to [38], we can find the solutions in closed form for Equation (38), a sixth-order polynomial. Finally, Equation (37) yields x .

In conclusion, we can summarize the weighted PP-ICP process in this paper as follows: Firstly, PointNet semantically segments the input point cloud data, generating labels for different types of items along with corresponding parameters: importance, IoU, and CPA. Next, we assign weights to these category labels using an objective weighting method, which employs an improved CRITIC as the weighting strategy to calculate the corresponding weight values. Finally, we use the calculated weight values as the weight parameters for the point-to-plane ICP, registering the input point cloud effectively. Figure 2 illustrates the algorithm's process of weighted PP-ICP.

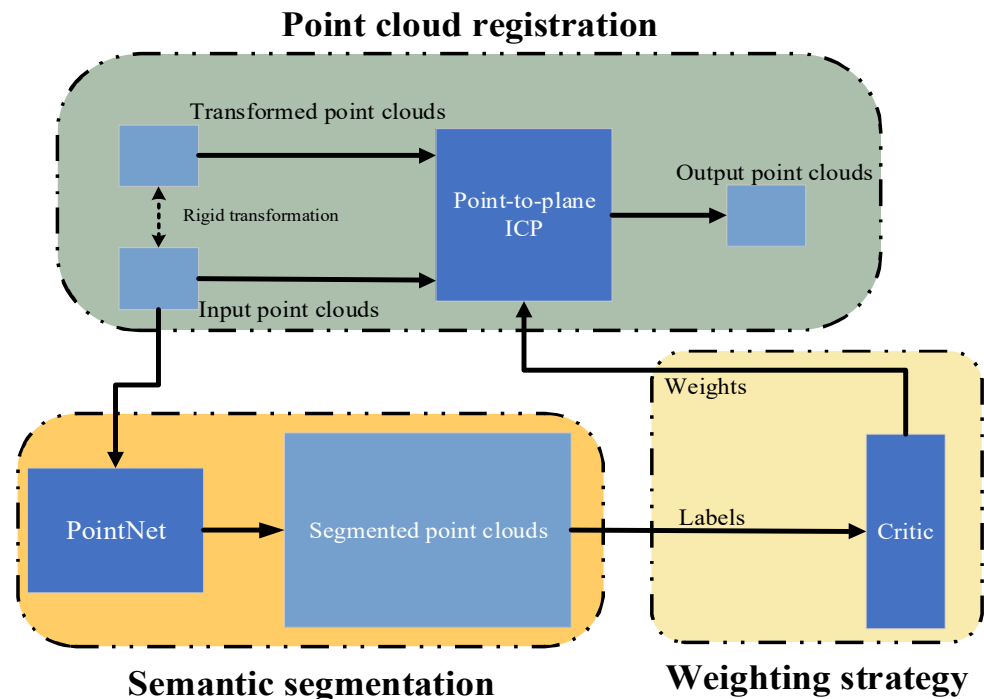


Figure 2. Process of weighted PP-ICP.

4. Experiment and Analysis

4.1. Data Preparation

The experimental datasets are sourced from public datasets, S3DIS [32], which consist of six different indoor spaces from six areas, respectively: conference room for Area_1, hallway for Area_2, office for Area_3, lobby for Area_4, storage for Area_5, and open space for Area_6. PointNet infers each label's IoU and CPA, calculates their average values from the six areas, and summarizes them along with the label's IM in Table 3.

Table 3. The average values of alternatives and criteria for six areas.

	Ceiling	Floor	Wall	Beam	Column	Window	Door	Table	Chair	Sofa	Case	Board	Clutter
IM	0.2	0.2	0.2	0.2	0.6	0.6	0.4	1	1	0.8	0.6	0.6	0.1
IoU	0.958	0.977	0.843	0.818	0.514	0.825	0.758	0.786	0.859	0.757	0.799	0.808	0.728
CPA	0.977	0.987	0.934	0.893	0.587	0.893	0.824	0.862	0.901	0.786	0.887	0.888	0.848

4.2. Weighting Strategy Steps

Based on Table 3, the decision matrix can be described as

$$M = \begin{bmatrix} 0.200 & 0.200 & 0.200 & 0.200 & 0.600 & 0.600 & 0.400 & 1.000 & 1.000 & 0.800 & 0.600 & 0.600 & 0.000 \\ 0.977 & 0.979 & 0.870 & 0.918 & 0.685 & 0.887 & 0.896 & 0.832 & 0.922 & 0.836 & 0.841 & 0.888 & 0.804 \\ 0.989 & 0.987 & 0.957 & 0.932 & 0.735 & 0.902 & 0.919 & 0.890 & 0.946 & 0.844 & 0.909 & 0.930 & 0.918 \end{bmatrix} \quad (39)$$

Equations (11) and (12) are applied to transform the decision matrix into fuzzy form, as shown in Table 4.

Table 4. Decision matrix in fuzzy form.

Criteria		IM	IoU	CPA
Ceiling	D'_{ij}^L	0.0887	0.4437	0.4526
	D'_{ij}^H	0.0987	0.4537	0.4626
Floor	D'_{ij}^L	0.0874	0.4465	0.4511
	D'_{ij}^H	0.0974	0.4565	0.4611
Wall	D'_{ij}^L	0.0962	0.4214	0.4674
	D'_{ij}^H	0.1062	0.4314	0.4774
Beam	D'_{ij}^L	0.0997	0.4230	0.4623
	D'_{ij}^H	0.1097	0.4330	0.4723
Column	D'_{ij}^L	0.3477	0.2972	0.3401
	D'_{ij}^H	0.3577	0.3072	0.3501
Window	D'_{ij}^L	0.2538	0.3509	0.3802
	D'_{ij}^H	0.2638	0.3609	0.3902
Door	D'_{ij}^L	0.1968	0.3774	0.4107
	D'_{ij}^H	0.2068	0.3874	0.4207
Table	D'_{ij}^L	0.3726	0.2918	0.3205
	D'_{ij}^H	0.3826	0.3018	0.3305
Chair	D'_{ij}^L	0.3573	0.3062	0.3214
	D'_{ij}^H	0.3673	0.3162	0.3314
Sofa	D'_{ij}^L	0.3364	0.3181	0.3305
	D'_{ij}^H	0.3464	0.3281	0.3405
Case	D'_{ij}^L	0.2575	0.3445	0.3830
	D'_{ij}^H	0.2675	0.3545	0.3930
Board	D'_{ij}^L	0.2563	0.3469	0.3818
	D'_{ij}^H	0.2663	0.3569	0.3918
Clutter	D'_{ij}^L	0.0547	0.4294	0.5010
	D'_{ij}^H	0.0647	0.4394	0.5110

The small positive number Δd in Equation (12) is set to be 0.005.

The boundaries of each criterion and corresponding fuzzy RP points are calculated according to Equation (13), and the results are shown in Table 5.

Table 5. Boundaries of each criterion and corresponding fuzzy RP points.

	Ceiling	Floor	Wall	Beam	Column	Window	Door	Table	Chair	Sofa	Case	Board	Clutter
L_j	0.328	0.328	0.328	0.328	0.328	0.328	0.328	0.328	0.328	0.328	0.328	0.328	0.328
H_j	0.338	0.338	0.338	0.338	0.338	0.338	0.338	0.338	0.338	0.338	0.338	0.338	0.338
D_j^{L-}	0.089	0.087	0.096	0.100	0.297	0.254	0.197	0.292	0.306	0.318	0.258	0.256	0.055
D_j^{H+}	0.463	0.461	0.477	0.472	0.358	0.390	0.421	0.383	0.367	0.346	0.393	0.392	0.511

The decision matrix is normalized according to Equation (14), and the results are shown in Table 6.

Table 6. Normalized decision matrix.

	Ceiling	Floor	Wall	Beam	Column	Window	Door	Table	Chair	Sofa	Case	Board	Clutter
IM	0.911	0.910	0.912	0.915	0.999	0.990	0.971	0.996	0.998	0.759	0.989	0.976	0.875
IoU	0.957	0.956	0.964	0.965	0.998	0.997	0.989	0.997	0.999	0.862	0.998	0.997	0.954
CPA	0.094	0.087	0.086	0.079	0.268	0.252	0.176	0.316	0.309	0.318	0.174	0.218	0.045

The standard variances of criteria are calculated according to Equation (7), and the results are shown as Table 7.

Table 7. Standard variances of criteria.

	Ceiling	Floor	Wall	Beam	Column	Window	Door	Table	Chair	Sofa	Case	Board	Clutter
σ_j	0.029	0.024	0.030	0.001	0.0008	0.0045	0.013	0.035	0.039	0.031	0.007	0.003	0.0001

The distance correlations between different pairs of criteria are calculated according to Equation (15), and the results are shown in Table 8.

Table 8. Distance correlations between different pairs of criteria.

	Ceiling	Floor	Wall	Beam	Column	Window	Door	Table	Chair	Sofa	Case	Board	Clutter
Ceiling	1.000	0.199	0.529	0.131	0.965	0.466	0.822	0.180	0.249	0.473	0.673	0.092	0.213
Floor	0.199	1.000	0.263	0.722	0.323	0.480	0.124	0.110	0.857	0.029	0.433	0.518	0.304
Wall	0.529	0.263	1.000	0.428	0.406	0.489	0.342	0.422	0.346	0.510	0.759	0.008	0.652
Beam	0.131	0.722	0.428	1.000	0.304	0.175	0.408	0.391	0.128	0.701	0.452	0.947	0.558
Column	0.965	0.323	0.406	0.304	1.000	0.728	0.287	0.776	0.255	0.918	0.687	0.511	0.826
Window	0.466	0.48	0.489	0.175	0.728	1.000	0.492	0.836	0.729	0.050	0.356	0.114	0.005
Door	0.822	0.124	0.342	0.408	0.287	0.492	1.000	0.088	0.848	0.979	0.867	−0.095	0.539
Table	0.180	0.110	0.422	0.391	0.776	0.836	0.088	1.000	0.086	0.972	0.010	0.601	0.591
Chair	0.249	0.857	0.346	0.128	0.255	0.729	0.848	0.086	1.000	0.581	0.450	0.712	0.556
Sofa	0.473	0.029	0.510	0.701	0.918	0.050	0.979	0.972	0.581	1.000	0.647	0.531	0.737
Case	0.673	0.433	0.759	0.452	0.687	0.356	0.867	0.010	0.450	0.647	1.000	0.437	0.404
Board	0.092	0.518	0.008	0.947	0.511	0.114	0.095	0.601	0.712	0.531	0.437	1.000	0.482
Clutter	0.213	0.304	0.652	0.558	0.826	0.005	0.539	0.591	0.556	0.737	0.404	0.482	1.000

The amounts of information and weights of criteria are calculated according to Equations (8)–(10) and corresponding ranks are assigned to the criteria based on the weights. The results are shown in Table 9.

Table 9. Information amounts, weights, and ranks of criteria.

	Ceiling	Floor	Wall	Beam	Column	Window	Door	Table	Chair	Sofa	Case	Board	Clutter
Q_j	0.072	0.077	0.112	0.008	0.004	0.011	0.027	0.147	0.116	0.098	0.011	0.017	0.001
ω_j	0.103	0.110	0.160	0.011	0.006	0.016	0.039	0.210	0.165	0.140	0.015	0.024	0.001
Ranks	6	5	3	12	11	9	7	1	2	4	10	8	13

Table 9 indicates that among all the labels, table, chair, and wall hold the top three rankings in terms of weights, which aligns well with the practical requirements of point cloud registration in 3D indoor spaces. In the Stanford Large-Scale 3D Indoor Spaces Dataset, chairs, walls, and tables are indispensable objects that occupy the central positions in the indoor space and the majority of the spatial volume. Improving the accuracy of point cloud registration for these objects will correspondingly enhance the registration accuracy for the entire 3D indoor space, as registration transforms the entire indoor space into a rigid structure. On the other hand, the weight assigned to clutter is minimized to a value close to zero, effectively minimizing the negative impact of noise and incorrectly matched points that will reduce registration accuracy.

4.3. Point Cloud Registration Experiments

We use the entire S3DIS dataset in the experiments, from Area_1 to Area_6 [32]. When the S3DIS datasets are used as a whole, all the test data are the averaged results of the six indoor spaces in the datasets, as shown in Figure 3. PointNet is used for semantic segmentation of the point cloud.

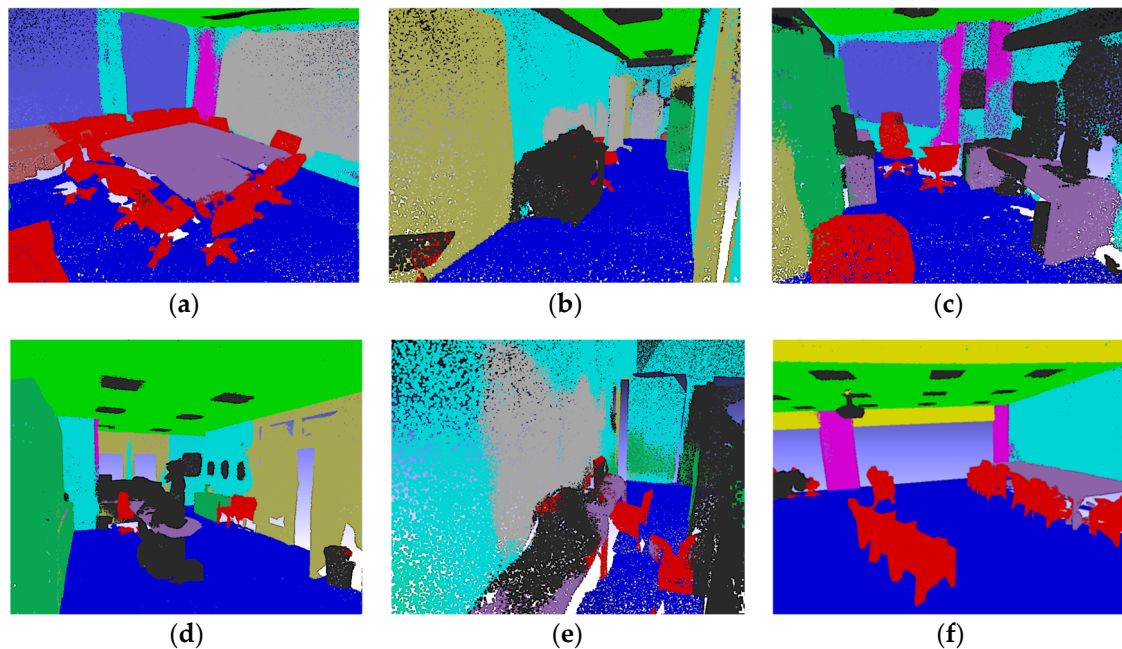


Figure 3. Point cloud semantic segmentation of S3DIS: (a) Area_1 (conference room); (b) Area_2 (hallway); (c) Area_3 (office); (d) Area_4 (lobby); (e) Area_5 (storage); (f) Area_6 (open space).

We divide the registration experiment into two parts. In the first part, we conduct experiments with weighted PP-ICP under different parameters, using the corresponding experimental data to determine the parameters that enable the algorithm to perform optimally. We also conduct other experiments to study the influence of semantic segmentation quality on the registration results. The second part is a comparison of weighted PP-ICP with three other registration algorithms: ICP, the NDT, and PointNetLK. We assess the advantages and disadvantages of weighted PP-ICP relative to the other algorithms by comparing the algorithms in terms of root mean square (RMS) error, runtime, and hardware resource utilization across six test scenarios in the datasets, using the method of controlled variables. In these experiments, we conduct both single-axis and multi-axis angle tests on the entire S3DIS dataset. In the single-axis angle test, we rotate the datasets around one axis, keeping the other two axes fixed. The range of rotation angles varies from 15° to 120° , in steps of 15° . The multi-axis angle tests maintain a range of angles between 15° and 120° , with a step of 15° in between. For each type of rigid transformation, we rotate all three axes clockwise by the same angle value, for example, $[30^\circ, 30^\circ, 30^\circ]$ and $[60^\circ, 60^\circ, 60^\circ]$.

Section 3 uses subjective assignment to determine each category's importance in the semantic segmentation results. The following tests will verify the validity of these assignments. According to Table 2, we reduce the importance of all labels above 0.5 and increase those below 0.5, resulting in a change magnitude of 0.2 for each group, which decreases to 0.1 when starting from 0.2. This process generates four groups of data regarding label importance, as shown in Table 10.

Table 10. Groups of importance of labels in 3D indoor spaces.

Group	Ceiling	Floor	Wall	Beam	Column	Window	Door	Table	Chair	Sofa	Case	Board	Clutter
1	0.2	0.2	0.2	0.2	0.6	0.6	0.4	1.0	1.0	0.8	0.6	0.6	0.1
2	0.4	0.4	0.4	0.4	0.4	0.4	0.6	0.8	0.8	0.6	0.4	0.4	0.3
3	0.6	0.6	0.6	0.6	0.2	0.2	0.8	0.6	0.6	0.4	0.2	0.2	0.5
4	0.8	0.8	0.8	0.8	0.1	0.1	1.0	0.4	0.4	0.2	0.1	0.1	0.7

We calculate and integrate the corresponding weight values into the proposed registration algorithm based on these four sets of data. We perform validation using multi-axis angle tests, as illustrated in Figure 4 and Table 11.

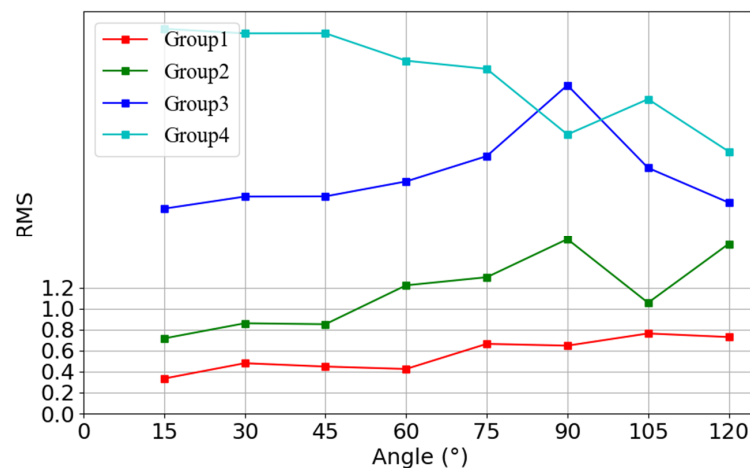


Figure 4. RMS of weighted PP-ICP for different groups of label importance.

Table 11. RMS of weighted PP-ICP for different groups of label importance.

Angles (°)	Group 1	Group 2	Group 3	Group 4
15	0.333	0.715	1.959	3.674
30	0.479	0.859	2.074	3.627
45	0.447	0.85	2.076	3.628
60	0.424	1.227	2.218	3.367
75	0.663	1.297	2.459	3.289
90	0.646	1.662	3.134	2.665
105	0.762	1.054	2.347	3.001
120	0.728	1.615	2.017	2.503

Figure 4 and Table 11 demonstrate that group 1 consistently has the smallest RMS across all rotational angles. The maximum value in group 3, 3.134, is 3.11 times higher than the maximum value in group 1, which is 0.762; the maximum value in group 4, 3.674, exceeds the maximum in group 1 by 3.82 times. These results validate that the subjective assignment of label importance in Section 3 is reasonable.

Given that this paper proposes an algorithm based on weights generated from point cloud semantic segmentation results, we expect the quality of segmentation to significantly influence the algorithm's registration performance. However, we notice differences in the

point cloud segmentation results among the six areas of the datasets during the experimental process, prompting us to conduct further tests to investigate the impact of semantic segmentation on the registration algorithm in this paper.

PointNet calculates the whole-scene point accuracy for each area in the datasets to assess the quality of semantic segmentation for the entire area. Table 12 displays the whole-scene point accuracy values for the six areas.

Table 12. Whole-scene point accuracy for six areas.

Area_1	Area_2	Area_3	Area_4	Area_5	Area_6
0.946	0.951	0.935	0.923	0.902	0.974

Table 12 shows that Area_6 has the best quality of semantic segmentation, while Area_5 has the worst. Three tests for the six areas are conducted with the rotational angles of the three axes being $[30^\circ, 30^\circ, 30^\circ]$, $[60^\circ, 60^\circ, 60^\circ]$, and $[90^\circ, 90^\circ, 90^\circ]$ respectively. The visualizations of the registration results when the rotational angle is equal to 30° are shown in Figure 5, and the test data are shown in Figure 6 and Table 13.

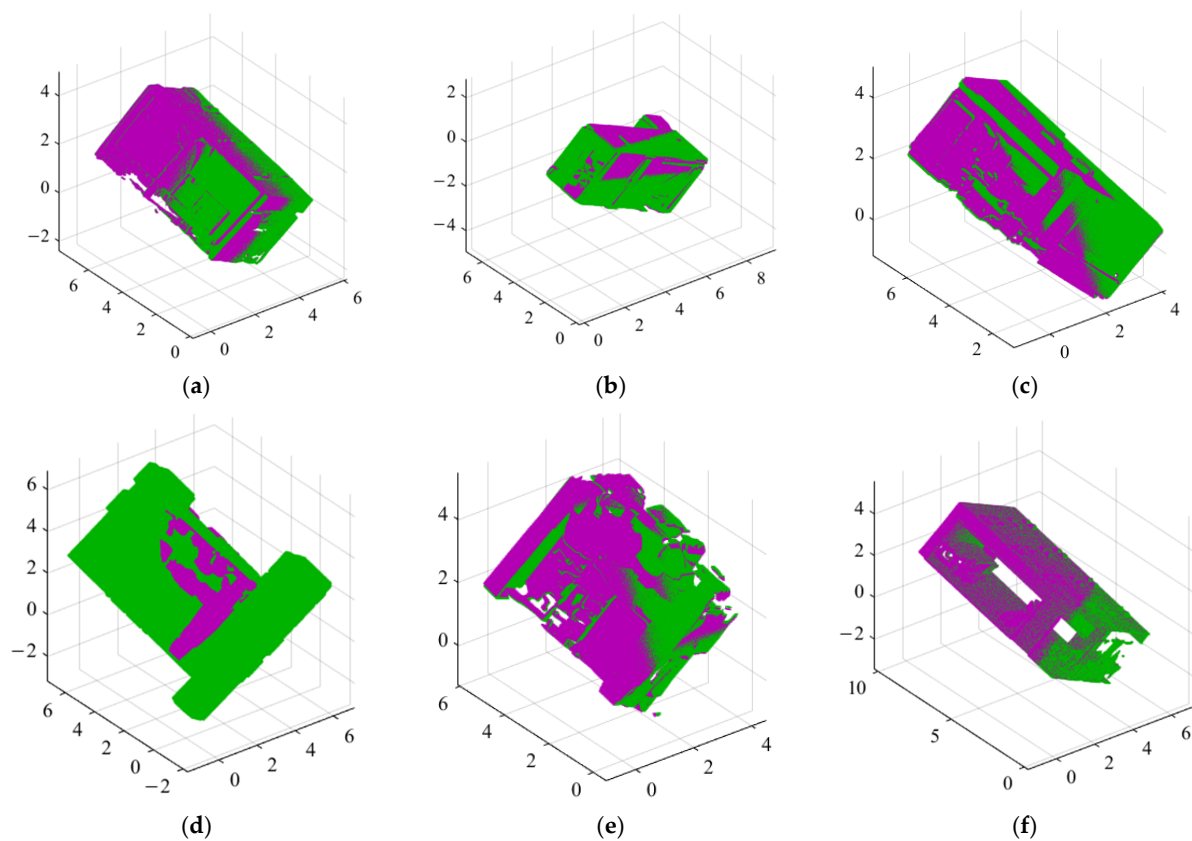


Figure 5. Registration results of S3DIS datasets: (a) Area_1; (b) Area_2; (c) Area_3; (d) Area_4; (e) Area_5; (f) Area_6. Rotational angle = $[30^\circ, 30^\circ, 30^\circ]$.

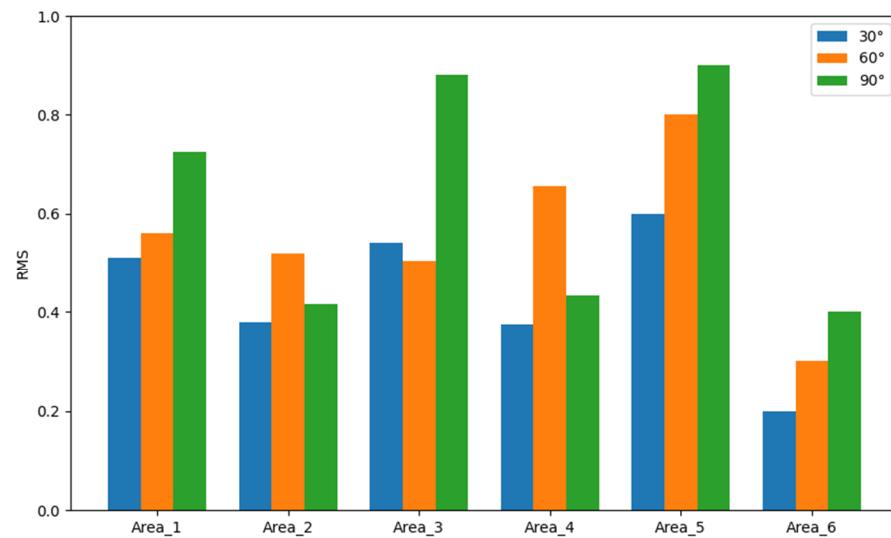


Figure 6. RMS for six areas.

Table 13. RMS for six areas.

Angles	Area_1	Area_2	Area_3	Area_4	Area_5	Area_6
30°	0.511	0.369	0.541	0.384	0.602	0.197
60°	0.561	0.520	0.504	0.656	0.796	0.305
90°	0.726	0.415	0.881	0.433	0.904	0.397

Figure 6 and Table 13 reveal that Area_6 has the smallest RMS under all three rotation angles, whereas Area_5 has the largest. This corresponds with the results of the PointNet semantic segmentation, where the whole-scene point accuracy is the highest in Area_6 and the lowest in Area_5. Similarly, the whole-scene point accuracy in Area_2 is higher than in Area_4, and the RMS in Area_2 is also smaller than in Area_4 under all three rotation angles. Based on what we have seen in practice, we can say that the proposed algorithm in this paper has a positive relationship between how well it registers and how well it semantically segments point clouds. Correct classifications in semantic segmentation are the only way to assign the correct weights in the registration algorithm's weight distribution process, thereby enhancing the algorithm's accuracy.

We conduct comparison experiments after determining the importance of labels in semantic segmentation. We set the translation to [0.1, 0.2, 0.15] for all tests. We set the change in angle in the rotational axis for each small rigid transformation in the weighted PP-ICP to 1°. Figure 7 and Table 14 display the root mean square error (RMS).

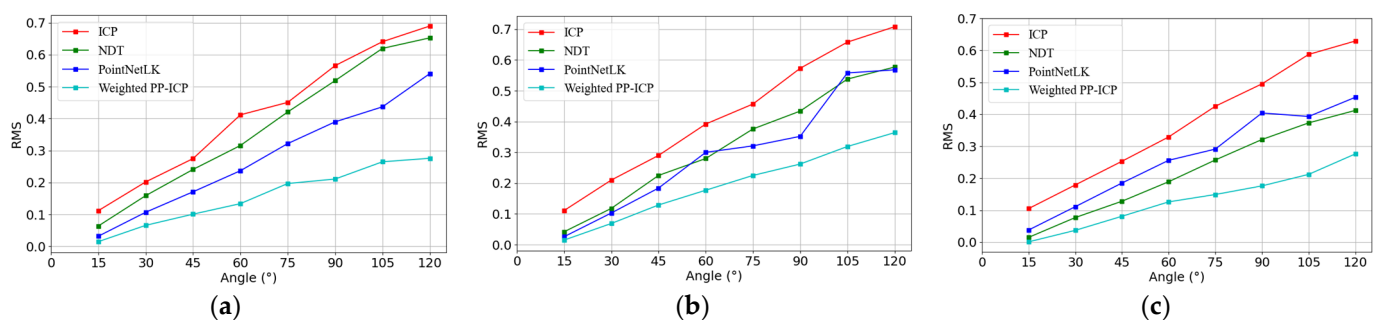


Figure 7. RMS of single-axis angle tests: (a) x-axis; (b) y-axis; (c) z-axis.

Table 14. RMS of single-axis angle tests for the algorithms.

Angles (°)	Axis	ICP	NDT	PointNetLK	Weighted PP-ICP
15	<i>x</i>	0.112	0.064	0.032	0.015
	<i>y</i>	0.111	0.042	0.026	0.015
	<i>z</i>	0.105	0.015	0.038	0.001
30	<i>x</i>	0.202	0.159	0.107	0.066
	<i>y</i>	0.214	0.118	0.103	0.069
	<i>z</i>	0.179	0.077	0.111	0.037
45	<i>x</i>	0.275	0.241	0.171	0.101
	<i>y</i>	0.271	0.225	0.184	0.129
	<i>z</i>	0.253	0.128	0.185	0.081
60	<i>x</i>	0.412	0.316	0.237	0.134
	<i>y</i>	0.392	0.285	0.314	0.177
	<i>z</i>	0.329	0.189	0.256	0.126
75	<i>x</i>	0.451	0.421	0.322	0.197
	<i>y</i>	0.457	0.376	0.321	0.225
	<i>z</i>	0.425	0.257	0.291	0.149
90	<i>x</i>	0.566	0.519	0.398	0.211
	<i>y</i>	0.573	0.434	0.352	0.262
	<i>z</i>	0.495	0.321	0.404	0.176
105	<i>x</i>	0.641	0.621	0.437	0.265
	<i>y</i>	0.658	0.538	0.558	0.319
	<i>z</i>	0.587	0.373	0.393	0.212
120	<i>x</i>	0.692	0.653	0.541	0.273
	<i>y</i>	0.708	0.577	0.568	0.364
	<i>z</i>	0.629	0.412	0.453	0.276

Figure 7 and Table 14 demonstrate that the proposed algorithm in this paper consistently achieves the smallest RMS in all tests, while the original ICP consistently demonstrates the highest RMS. This demonstrates the significant improvement in registration accuracy achieved by the algorithm proposed in this paper. On the other hand, as the rotation angle increases, the RMS of all the algorithms increases, but the weighted PP-ICP always has the smallest increase. This means that the algorithm is very stable when rigid bodies change shape at different rotation angles.

The multi-axis angle tests maintain the angle's range between 15° and 120°, with a 15° step. For each type of rigid transformation, we rotate all three axes clockwise by the same angle value, such as [30°, 30°, 30°] and [60°, 60°, 60°]. Figure 8 and Table 15 display the tests.

Table 15. RMS of multi-axis angle tests for the algorithms.

Angles (°)	ICP	NDT	PointNetLK	Weighted PP-ICP
15	0.358	0.423	0.242	0.375
30	0.473	0.512	0.359	0.434
45	0.589	0.597	0.493	0.491
60	0.645	0.677	0.576	0.557
75	0.832	0.733	0.698	0.651
90	0.986	0.829	0.791	0.626
105	1.129	0.894	0.954	0.717
120	1.177	0.973	1.088	0.787

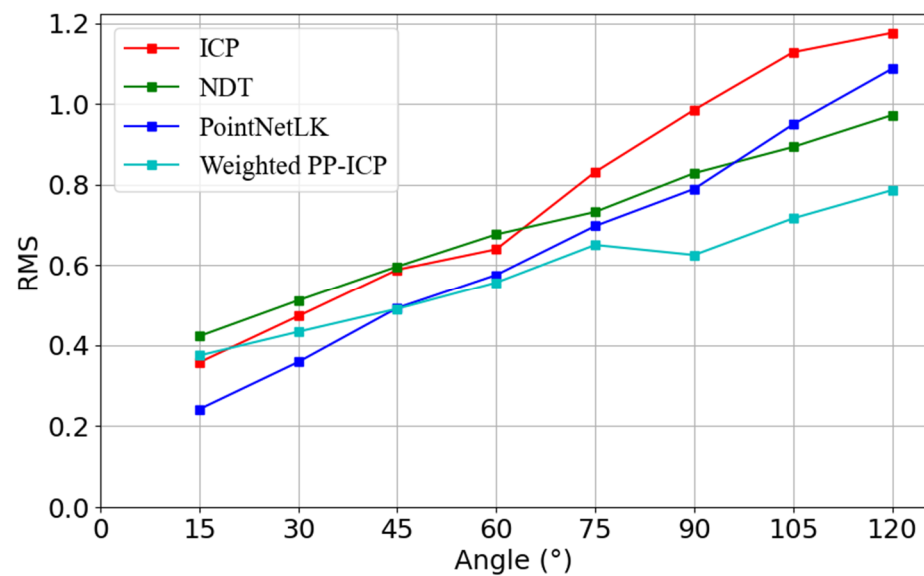


Figure 8. RMS of multi-axis angle tests.

Figure 8 and Table 15 demonstrate that in the multi-axis angle tests, the weighted PP-ICP consistently exhibits the smallest RMS, with the exception of the cases at 15° and 30°. Similar to the single-axis angle tests, the RMS increases as the rotation angle increases. However, we observe that in the multi-axis angle tests, the RMS of all the algorithms surpasses that of the single-axis angle tests, suggesting an increase in registration errors as the complexity of the rigid-body transformations increases.

We evaluate the runtime and computation resources used during the multi-axis tests for the four algorithms on a personal computer with a CPU processor (AMD Ryzen 7 4800H) and 16.0 GB of RAM to validate the efficiency. Figure 9 and Table 16 display the results.

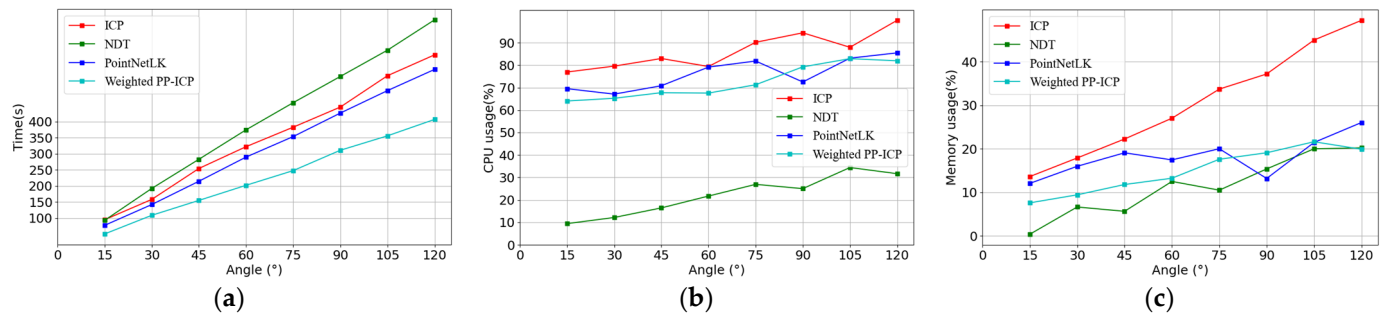


Figure 9. Runtime and computation resource costs during multi-axis tests: (a) runtime; (b) CPU usage; (c) memory usage.

Figure 9 and Table 16 conclude that the weighted PP-ICP algorithm in this paper exhibits significantly shorter runtimes than the original ICP, NDT, and PointNetLK. However, in comparison to the NDT, the proposed algorithm does not perform better in terms of CPU or memory usage. The main reason for this is that the NDT breaks up the original point cloud into a voxel grid to obtain Gaussian distributions through feature extraction. It then changes these distributions to the reference point cloud's coordinate system [4]. Compared to ICP, which directly matches the original point cloud, the NDT significantly reduces computational complexity, thereby enhancing the hardware efficiency of the algorithm.

Table 16. Runtime and computation resource costs during multi-axis tests.

Indicators		ICP	NDT	PointNetLK	Weighted PP-ICP
Time (s)	15°	94.561	92.656	77.072	50.233
	30°	157.958	192.108	141.939	108.381
	45°	254.401	283.051	214.551	154.845
	60°	322.314	374.898	289.913	201.934
	75°	383.113	459.502	353.857	247.719
	90°	445.865	541.271	427.142	311.205
	105°	543.583	623.247	497.156	356.182
	120°	608.738	717.924	563.291	407.393
CPU Usage (%)	15°	76.979	9.433	69.557	64.061
	30°	79.595	12.166	67.123	65.274
	45°	82.988	16.437	70.858	67.763
	60°	79.431	21.714	79.166	67.589
	75°	90.211	26.924	81.871	71.298
	90°	94.439	25.022	72.539	79.271
	105°	87.991	34.421	83.143	82.918
	120°	99.996	31.688	85.503	81.985
Memory Usage (%)	15°	13.616	0.357	12.051	7.574
	30°	17.889	6.623	15.977	9.417
	45°	22.252	5.628	19.077	11.785
	60°	27.025	12.498	17.448	13.237
	75°	33.719	10.491	20.011	17.593
	90°	37.199	15.371	13.174	19.088
	105°	44.998	19.999	21.456	21.618
	120°	49.523	20.215	26.034	19.893

5. Discussion

We decompose a complete rigid-body transformation into a superposition of many tiny rigid-body transformations. For instance, a 120-degree rigid-body transformation is essentially a combination of numerous short-time, small-angle rigid-body transformations. By breaking down complex transformations into smaller, simpler components, we can better understand and analyze the overall motion of the rigid body. This decomposition method allows us to study the effects of each individual transformation and how they contribute to the overall movement of the object. By studying these small-angle transformations, we can gain insights into the mechanics and dynamics of rigid-body motion. Within each of these tiny rigid-body transformations, the angle is extremely small, enabling the application of approximation theory. This decomposition allows us to break down complex transformations into simpler, more manageable steps that can be analyzed and understood more easily. By applying approximation theory to each small angle transformation, we can accurately model and predict the behavior of the overall rigid-body transformation. This approach is particularly useful in robotics, computer graphics, and other fields where precise control and understanding of rigid-body movements are crucial. This small angle allows the algorithm to quickly iterate to the optimal solution. During the subsequent experimental validation phase, we discover that the algorithm's running time increases with the size of the transformation angle, as it superimposes more tiny, rigid-body transformations. However, this decomposition fully utilizes the benefits of the new method for solving the iterative problem suggested in Section 3. Each small transformation exhibits high accuracy, which enhances the accuracy of the final superimposed changes. The original paper provides a theoretical derivation, and we set the amount of angular change in the rotation axis to 1° for each iteration of the algorithm in the subsequent experiments. This incremental approach allows us to achieve precise results with each iteration, ultimately leading to a highly accurate final transformation. By setting the rotation axis change to 1° for each iteration, we are able to control the granularity of the transformations and ensure

that the algorithm runs efficiently. This method proves to be highly effective in solving the iterative problem and provides a solid foundation for future experiments and applications.

6. Conclusions

Many millimeter-wave radar applications, including autonomous driving, robotic navigation, and augmented reality, widely use point cloud registration. During the registration process, accuracy and effectiveness are two key factors that will have a direct impact on performance. This paper proposes a novel algorithm for point cloud registration. First, PointNet uses different category labels in the 3D indoor spaces to semantically segment the point cloud for registration. We introduce a statistical weighting strategy that assigns varying weights to the point cloud corresponding to the category labels. Then, during the registration process, we propose a new method to calculate the closed form of ICP iterative optimal solutions in three-dimensional spaces. This method addresses the issue in traditional registration algorithms where the linear approximation of the objective function reduces accuracy. We successfully calculate the closed-form solutions by applying the weights of the point cloud to the objective function solving process. We validate the accuracy and efficiency of the proposed algorithm on point cloud datasets of 3D indoor spaces, and the registration experiments reveal significant improvements in both accuracy and efficiency after incorporating the weighting strategy into the ICP algorithm. The technique may swiftly iterate to the ideal solution by breaking hard transformations into numerous tiny transformations, each with an infinitesimal angle. Furthermore, the approach to finding closed-form solutions presented in Section 4 may significantly improve the registration's accuracy.

The dataset tests show that the RMS of different types of rigid transformation become much smaller compared to ICP, the NDT, and PointNetLK. On the other hand, compared to the original ICP, the algorithm's runtime and usage of both CPU and memory improve. Overall, this paper's proposed weighting-strategy-based point cloud registration algorithm demonstrates improved accuracy and increased efficiency in large-scale 3D indoor spaces.

While the novel algorithm in this paper, based on a weighting strategy, significantly improves the accuracy and efficiency of point cloud registration in 3D indoor spaces, it still falls short in terms of hardware resource utilization when compared to other registration algorithms like the NDT. Future work will focus on further optimizing the ICP iterative process to reduce computation costs, as well as exploring the integration of other more efficient registration algorithms with the proposed weighting strategy.

Author Contributions: Methodology, W.L., H.Z. and S.S.; Software, W.C. and S.S.; Formal analysis, W.L., H.Z. and X.L.; Writing—original draft, W.L.; Writing—review & editing, H.Z., W.C., X.L. and S.S.; Visualization, W.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Heilongjiang Province's Key Research and Development Project: 'open competition mechanism to select the best candidates', grant number 2023ZXJ03C02; and National Science Foundation of China, grant number 52075117.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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

References

1. Charles, R.Q.; Su, H.; Kaichun, M.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
2. Arun, K.S.; Huang, T.S.; Blostein, S.D. Least-squares fitting of two 3-D point sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **1987**, *5*, 698–700. [[CrossRef](#)] [[PubMed](#)]
3. Besl, P.J.; McKay, N.D. Method for registration of 3-D shapes. *Sens. Fusion IV Control Paradig. Data Struct.* **1992**, *1611*, 586–606. [[CrossRef](#)]

4. 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.
5. Magnusson, M.; Andreasson, H.; Nuchter, A.; Lilienthal, A.J. Appearance-based loop detection from 3D laser data using the normal distributions transform. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009.
6. 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.
7. Chang, C.H.; Chou, C.N.; Chang, E.Y. CLKN: Cascaded Lucas-Kanade Networks for Image Alignment. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
8. Low, K.L. *Linear Least-Squares Optimization for Point-to-Plane Icp Surface Registration*; University of North Carolina: Chapel Hill, NC, USA, 2004.
9. Serafin, J.; Grisetti, G. NICE: Dense normal based point cloud registration. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015.
10. 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](#)]
11. Deschaud, J.E. IMLS-SLAM: Scan-to-model matching based on 3D data. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018.
12. Das, A.; Waslander, S.L. Scan registration with multi-scale k-means normal distributions transform. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012.
13. 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 2019 IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019.
14. Li, T.; Shi, Y.; Sun, X.; Wang, J.; Yin, B. PCGAN: Prediction-Compensation Generative Adversarial Network for Meshes. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 4667–4679. [[CrossRef](#)]
15. 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.
16. Bai, X.; Luo, Z.; Zhou, L.; Fu, H.; Quan, L.; Tai, C.-L. D3Feat: Jointlearning 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.
17. Zhang, Z.; Dai, Y.; Sun, J. Deep learning based point cloud registration: An overview. *Virtual Real. Intell. Hardw.* **2020**, *2*, 222–246. [[CrossRef](#)]
18. Pais, G.D.; Ramalingam, S.; Govindu, V.M.; Nascimento, J.C.; Chellappa, R.; Miraldo, P. 3DRegNet: A deep neural network for 3D point registration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
19. Chen, P.; Li, X. Revised spectral matching algorithm for scenes with mutually inconsistent local transformations. *IET Image Processing*. **2015**, *9*, 916–922. [[CrossRef](#)]
20. Huang, X.; Mei, G.; Zhang, J.; Abbas, R. A comprehensive survey on point cloud registration. *arXiv* **2021**, arXiv:2103.02690.
21. Pomerleau, F.; Colas, F.; Siegwart, R. A review of point cloud registration algorithms for mobile robotics. *Found. Trends[®] Robot.* **2015**, *4*, 1–104. [[CrossRef](#)]
22. Cheng, L.; Chen, S.; Liu, X.; Xu, H.; Wu, Y.; Li, M.; Chen, Y. Registration of laser scanning point clouds: A review. *Sensors* **2018**, *18*, 1641. [[CrossRef](#)] [[PubMed](#)]
23. Saiti, E.; Theoharis, T. An application independent review of multimodal 3d registration methods. *Comput. Graph.* **2020**, *91*, 153–178. [[CrossRef](#)]
24. Zhou, Q.-Y.; Park, J.; Koltun, V. Fast global registration. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 766–782.
25. Huang, X.; Fan, L.; Wu, Q.; Zhang, J.; Yuan, C. Fast registration for cross-source point clouds by using weak regional affinity and pixel-wise refinement. In Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME), Shanghai, China, 8–12 July 2019; pp. 1552–1557.
26. Xin, S.; Wang, Y.; Lv, W. Standard deviation control chart based on weighted standard deviation method. In Proceedings of the 33rd Chinese Control Conference, Nanjing, China, 28–30 July 2014.
27. Liu, Z.; Zhang, Z.; Zhang, X.; Lin, X.; Zhao, H.; Shen, Z. A Path Analysis of Electric Heating Technology Based on Entropy Weight Method. In Proceedings of the 2021 IEEE 5th Conference on Energy Internet and Energy System Integration (EI2), Taiyuan, China, 22–24 October 2021.
28. Diakoulaki, D.; Mavrotas, G.; Papayannakis, L. Determining objective weights in multiple criteria problems: The critic method. *Comput. Oper. Res.* **1995**, *22*, 763–770. [[CrossRef](#)]
29. Xu, Y.; Cai, Z. Standard deviation method for determining the weights of group multiple attribute decision making under uncertain linguistic environment. In Proceedings of the 2008 7th World Congress on Intelligent Control and Automation, Chongqing, China, 25–27 June 2008.

30. Wang, J.; Hao, Y.; Yang, C. An entropy weight-based method for path loss predictions for terrestrial services in the VHF and UHF bands. *Radio Sci.* **2023**, *58*, 1–11. [[CrossRef](#)]
31. Wang, Z.; Yang, X.; Hu, H.; Lou, Y. Actor-critic method-based search strategy for high precision peg-in-hole tasks. In Proceedings of the 2019 IEEE International Conference on Real-Time Computing and Robotics (RCAR), Irkutsk, Russia, 4–9 August 2019.
32. Armeni, I.; Sener, O.; Zamir, A.R.; Jiang, H.; Brilakis, I.; Fischer, M.; Savarese, S. 3D semantic parsing of large-scale indoor spaces. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
33. Zhou, D.; Fang, J.; Song, X.; Guan, C.; Yin, J.; Dai, Y.; Yang, R. Iou loss for 2D/3D object detection. In Proceedings of the 2019 International Conference on 3D Vision (3DV), Quebec City, QC, Canada, 16–19 September 2019.
34. Bressan, P.O.; Junior, J.M.; Martins, J.A.C.; de Melo, M.J.; Gonçalves, D.N.; Freitas, D.M.; Ramos, A.P.M.; Furuya, M.T.G.; Osco, L.P.; de Andrade Silva, J.; et al. Semantic segmentation with labeling uncertainty and class imbalance applied to vegetation mapping. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *108*, 102690.
35. Marom, N.D.; Rokach, L.; Shmilovici, A. Using the confusion matrix for improving ensemble classifiers. In Proceedings of the 2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel, Eilat, Israel, 17–20 November 2010.
36. Baidya, J.; Garg, H.; Saha, A.; Mishra, A.R.; Rani, P.; Dutta, D. Selection of third party reverses logistic providers: An approach of BCF-CRITIC-MULTIMOORA using Archimedean power aggregation operators. *Complex Intell. Syst.* **2021**, *7*, 2503–2530. [[CrossRef](#)]
37. Sharkasi, N.; Rezakhah, S. A modified CRITIC with a reference point based on fuzzy logic and hamming distance. *Knowl. Based Syst.* **2022**, *255*, 109768. [[CrossRef](#)]
38. 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.

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.