Exploring the Impact of Different Registration Methods and Noise Removal on the Registration Quality of Point Cloud Models in the Built Environment: A Case Study on Dickabrma Bridge

Zicheng Zhu 1,2,*, Steve Rowlinson 2,3,*, Tianzhuo Chen 1,2 and Alan Patching 1

1 Faculty of Society & Design, Bond University, Gold Coast, QLD 4226, Australia
2 Centre for Comparative Construction Research, Bond University, Gold Coast, QLD 4226, Australia
3 Department of Real Estate and Construction, The University of Hong Kong, Pokfulam, Hong Kong
4 Correspondence: zicheng.zhu@student.bond.edu.au (Z.Z.); hrecsmr@hku.hk (S.R.)

Abstract: Point cloud models are prevalently utilized in the architectural and civil engineering sectors. The registration of point clouds can invariably introduce registration errors, adversely impacting the accuracy of point cloud models. While the domain of computer vision has delved profoundly into point cloud registration, limited research in the construction domain has explored these registration algorithms in the built environment, despite their inception in the field of computer vision.

The primary objective of this study is to investigate the impact of mainstream point cloud registration algorithms—originally introduced in the computer vision domain—on point cloud models, specifically within the context of bridge engineering as a category of civil engineering data. Concurrently, this study examines the influence of noise removal on varying point cloud registration algorithms. Our research quantifies potential variables for registration quality based on two metrics: registration error (RE) and time consumption (TC). Statistical methods were employed for significance analysis and value engineering assessment. The experimental outcomes indicate that the GRICP algorithm exhibits the highest precision, with RE values of 3.02 mm and 2.79 mm under non-noise removal and noise removal conditions, respectively. The most efficient algorithm is PLICP, yielding TC values of 3.86 min and 2.70 min under the aforementioned conditions. The algorithm with the optimal cost-benefit ratio is CICP, presenting value scores of 3.57 and 4.26 for non-noise removal and noise removal conditions, respectively. Under noise removal conditions, a majority of point cloud algorithms witnessed a notable enhancement in registration accuracy and a decrease in time consumption. Specifically, the POICP algorithm experienced a 32% reduction in RE and a 34% decline in TC after noise removal. Similarly, PLICP observed a 34% and 30% reduction in RE and TC, respectively. KICP showcased a decline of 23% in RE and 28% in TC, CICP manifested a 27% and 31% drop in RE and TC, respectively, GRICP observed an 8% reduction in RE and a 40% decline in TC, and for FGRICP, RE and TC decreased by 8% and 52%, respectively, subsequent to noise removal.

Keywords: laser scanning; point cloud; ICP; registration quality; open3d

1. Introduction

Point cloud models have an extensive range of applications within the Architecture, Engineering, and Construction (AEC) industry, such as building maintenance management, building refurbishment and modification, as well as structural health monitoring [1–5]. The term “point cloud model” refers to a collection of 3D points, each of which is characterized by x, y, z coordinates, used to represent the information on various objects...
or components [6–8]. Typically, these 3D points also encompass RGB data, providing color information for the depicted component [9].

Point cloud data can be gathered using various reality capture technologies, including laser scanning and digital photogrammetry. Laser scanning technology can be further subdivided into Terrestrial Laser Scanning (TLS) and Mobile Laser Scanning (MLS) technologies. TLS, which involves stationary scanning, is characterized primarily by offering superior precision, capturing point cloud data with higher resolution and density [7,10–13]. While MLS may not match TLS in terms of precision and density, it offers greater flexibility, enabling the scanning of complex architectural environments [14]. It provides excellent solutions for addressing architectural blind spots that TLS might not be able to cover [15–19]. Digital photogrammetry, in contrast to laser scanning techniques, is known as a passive reality capture technology. It does not actively emit any laser beams or signal waves. Instead, it passively captures reflected light from natural or artificial light sources on objects through photodetectors to form images. Then, algorithms such as Structure from Motion (SFM) are employed to reconstruct the three-dimensional information corresponding to the pixel values in the photographs [20–23]. Compared to laser scanning techniques for point cloud data collection, digital photogrammetry offers lower accuracy and point cloud density, and it cannot perform reality capture in dark environments [24]. However, its hardware equipment is considerably more cost-effective than that utilized in laser scanning methods [23].

Upon the completion of point cloud data collection, the raw data must undergo a series of processing steps to form a point cloud model. These processes include noise removal, subsampling, and registration, among others [7,13]. Of these, registration is arguably the most crucial step. Typically, a project is large, and it is unlikely that a single scan can cover all the data required. Additionally, multiple devices may be used together, each performing its own point cloud data collection, with the data existing in an independent coordinate system [25,26]. To generate a comprehensive point cloud model, it is necessary to accurately synthesize the data from different parts of the project into a common coordinate system using appropriate point cloud registration methods. However, the registration process may not always be straightforward. Significant registration errors can lead to a decrease in the overall accuracy of the point cloud model. Even minor errors can gradually accumulate into larger ones throughout multiple continuous scans and iterative registration processes, thereby impacting the overall accuracy [27,28]. This is especially pertinent in bridge engineering, which often involves many successive and adjacent scans in the same direction. Even the slightest errors can iterate into substantial ones, potentially causing significant discrepancies.

Data in the built environment possesses distinct characteristics. Conducting in-depth research on the point cloud registration algorithms proposed in the field of computer vision, specifically within a built environment context, facilitates for construction professionals the crafting of superior point cloud models, thereby better aligning with application scenarios in the construction industry. However, in the current construction industry, few teams have explored point cloud registration algorithms, proposed in the computer vision domain, in specific architectural contexts. The primary objective of this research is to investigate the impact of different registration methods and the role of noise removal on the quality of point cloud registration. This study quantifies registration quality through two metrics: Registration Error (RE) and Time Consumption (TC). Furthermore, all registration methods tested in this experiment are evaluated using the value engineering method, with the goal of identifying the registration method that offers the highest cost-effectiveness. Through this experimental study, the authors aim to provide empirical data support to scholars and practitioners in the construction industry when establishing high-precision point cloud models within the built environment. Based on the results of this investigation, readers can gain a comprehensive understanding of the performance of mainstream point cloud registration methods in bridging construction, as well as the extent to which noise removal can impact registration quality.
2. Related Work

2.1. Registration Method

Registration methods can generally be subdivided into two distinct categories: coarse registration and fine registration [28–31]. The primary objective of coarse registration is to establish an approximate initial alignment of the source and target point cloud that are to be registered. This alignment, which ideally should result in a general overlap of the two point clouds, consequently influences the precision of the subsequent fine registration process. There are primarily three strategies employed in coarse registration. The first strategy is the manual method, which necessitates human intervention. In this method, corresponding points or surfaces are selected from both the source and target point clouds, after which point pairs are formed and used for registration [32]. The second strategy is the target-based method. During the registration process, markers are placed, which are then identified in both the source and target point clouds. This method facilitates the formation of point pairs from corresponding markers [8]. The third strategy is descriptor-based. In this method, descriptors of both the source and target point clouds are computed, and point pairs are formed from points with identical descriptor features for registration [26,33,34].

Fine registration, on the other hand, is conducted after the completion of the coarse registration process. Its purpose is to fine-tune the alignment of the source and target point clouds to achieve an optimal overlap. The accuracy of the point cloud ensemble formed from the two point clouds can be affected by error in the fine registration [28]. The methodologies for fine registration are primarily Iterative Closest Point (ICP) and its derivative algorithms [35,36]. These algorithms are characterized by initial position, thresholds, loss function, and iteration. Throughout the iterative process, the loss function is manipulated to meet the threshold requirements, ultimately leading to the final transformation matrix [25].

2.2. Noise and Outlier Removal

Noise and outliers both represent data characteristics that negatively impact the accuracy of point cloud models [37]. Outliers are defined as points with exceptionally large error values, far exceeding systematic errors and caused by complex and special factors. The presence of these points is determined by numerous factors, such as the material, color, roughness of the scanned object, and environmental conditions, such as temperature and lighting intensity [10,38,39]. Outlier detection and removal in data has been extensively studied, employing diverse methodologies. Prominent approaches predominantly utilize local statistics to determine anomalies, referencing metrics like local density [40], proximity to nearest neighbors [41], and eigenvalues of the local covariance matrix [42], with further methods catalogued by Papadimitriou et al. [43]. While some methods opt for direct hard thresholding to identify outliers [41], others prefer a distributional approach, commonly assuming a normal distribution, to highlight deviant points [40]. Moreover, there is a distinction in literature between sparse outliers and temporal artifact outliers, the latter often associated with scene movements. For instance, Kanzok et al. [44] focused on filtering out such temporal artifacts by identifying transient clusters.

On the other hand, noise refers to a uniform, systematic error inherent in the process of data collection via laser scanning technology. This type of error is ubiquitous and cannot be compensated for through hardware optimization. Certain non-target objects can also be considered a specific type of noise. During the scanning process, the resultant point cloud data often encapsulates objects irrelevant to the target object. For instance, when scanning a building structure, the collected point cloud data may also incorporate flora, fauna, pedestrians, and birds [45]. These elements typically exhibit non-static properties and can obstruct the target objects to a certain extent [7]. In the realm of point cloud data processing, noise removal stands out as a pivotal challenge, given the intricate balance
between filtering out extraneous noise and preserving crucial, sharp features. Foundational techniques, like local neighborhood smoothing, provide a basis, but often risk oversmoothing. The kernel density estimator, as noted by Schall et al. [40], showed potential, but faltered near sharp regions. Iterative sampling, explored by Liu et al. [46], filled data holes adeptly but struggled with feature restoration. The literature revealed a trend towards anisotropic filtering, which adjusts smoothing based on directionality, with Lange and Polthier [47] introducing a method using mean curvature flows, though it is computationally demanding. An evolution of this, presented by Wang et al. [48], targeted point normal for more efficient results. Interestingly, some other researchers, such as Ahmed et al. [49] and Aiger et al. [31], circumvented filtering, aiming for inherent noise robustness. Tools like PCL, Open3d, CloudCompare, and MeshLab encapsulate these advancements, offering practical solutions for researchers and practitioners.

2.3. The Importance of High-Quality Point Cloud Models

The escalating integration of point cloud models into modern engineering and architectural applications accentuates the pivotal role of their accuracy and precision. Most research concerning the accuracy of point cloud models predominantly elucidates the implications of varying precision levels from an application-oriented perspective.

In the evolving landscape of construction quality inspection, point cloud models act as a mirror, reflecting the real-world status of ongoing construction projects. They offer an unparalleled medium to juxtapose the on-ground construction with as-designed BIM models. This comparative analysis facilitates the evaluation of adherence to design standards, detection of discrepancies, and ensures that the on-site work aligns with the preconceived design blueprints [50]. A compromise in the fidelity of point cloud models can result in erroneous evaluations, thereby potentially jeopardizing the integrity of the entire construction project [51,52].

The Scan-to-BIM methodology underscores the significance of point cloud models. A large swathe of extant structures, with legacy documentation or even lacking detailed plans, necessitates the creation of high-precision point cloud models. These models serve as the foundational layer for developing digital twin replicas, which aid in visual asset management, structural health assessments, and future retrofitting plans. An imprecise point cloud model might lead to the formulation of a digital twin that veers away from reality, limiting its applicability in real-time simulations and analysis [53–55].

Point cloud models play a linchpin role in contemporary renovation and remodelling tasks. The ability to digitize an existing structure with high precision provides architects and designers with an accurate canvas, which becomes the basis for their redesign initiatives. A lackluster accuracy in point cloud models could stymie the planning phase, culminating in redesigns that might be unfeasible or in dissonance with the actual structure [1,56].

High-quality point cloud models are not a luxury but a quintessential need in today's technologically-driven architecture and engineering landscapes. Their precision dictates the success of a myriad of applications, making their accuracy a non-negotiable attribute for professionals in the field.

3. Methodology and Experiment

3.1. Framework

The primary aim of this study is to investigate the impact of different point cloud registration strategies and the potential influence of noise removal on the quality of point cloud registration. The entire exploration process encompasses five steps (Figure 1).

Firstly, we collected data on the target bridge structure using a terrestrial laser scanner. Secondly, the original point clouds, or scans, were divided into several scan pairs according to the rule of pairing adjacent scans. Each scan pair consists of a source point cloud and a target point cloud. Thirdly, we established two scenarios. Scenario 1 involves raw
scan pairs, meaning no additional processing is performed. Scenario 2 involves noise removal processing on the basis of raw scan pairs, deleting all point clouds irrelevant to the main body of the bridge, primarily including trees, plants, pedestrians, and vehicles. Fourthly, we applied six different point cloud registration algorithms to register the scan pairs in the two scenarios, and then obtained the RE data and the time TC data for each scan pair. Finally, we employed significance analysis and value analysis to process and analyze the RE data and TC data.

Figure 1. Research Framework. Note: The illustration elucidates the process of collecting data from point clouds, subsequently testing various registration methods under distinct conditions, leading to the quantification of point cloud registration quality in terms of RE and TC.

3.2. Project and Equipment

The Dickabram Bridge (Figure 2), a significant historical road-and-rail bridge over Australia’s Mary River, connects the towns of Miva and Theebine in the Gympie Region, Queensland. The bridge, known also as Mary River Bridge (Miva), is notable for being a primary structure on the Kingaroy railway line. Its construction, from 1885 to 1886, was overseen by Henry Charles Stanley, with building duties carried out by Michael McDermott, Owens & Co. This structure was recognized for its national importance and was added to the former Register of the National Estate in 1988. Being one of only three such road-and-rail bridges left in Australia, it is the sole representative of its kind in Southeast Queensland, especially after the Burdekin Bridge was completed in 1957. It holds the distinction of being the oldest large steel truss bridge that remains in Queensland [57].

In this experiment, the primary equipment we employed was the Faro Focus S70 (Figure 3a), a terrestrial laser scanner designed by Faro Corporation. This scanner offers point cloud collection speeds of up to 614 m at 0.5 million points per second and 307 m at 1 million points per second. The overall accuracy of the scanner can generally reach within ±2 mm [27,54]. We implemented a total of six PCD registration algorithms and computed two types of metrics. All algorithms and metric computations were primarily executed using Python, utilizing libraries such as Open3D, Numpy, and Time. Given that each registration process and the calculation of the RE values and TC values were required for every pair of consecutive scans in each setting, and for every algorithm, the computational
load was substantial. Therefore, our team simultaneously deployed 12 computers in the Bond BIM lab for executing these algorithms and computations (Figure 3b). Each machine was equipped with an Intel i9-11900 @2.5GHZ CPU, an RTX 3080 GPU, and 64 GB of RAM.

Figure 2. Dickabram Bridge.

Figure 3. (a) Faro focus S70. (b) BIM lab’s high-performance computer.

3.3. Point Cloud Data Collection and Noise Removal

A total of 29 scanning points were selected on the Dickabrma Bridge for point cloud collection (Figure 4), resulting in 29 point clouds under independent coordinate systems. During the scanning process, the Faro S70’s built-in GPS, inclinometer, compass, and altimeter recorded the corresponding scanning position information. The data, harnessed from the built-in sensors of the FS70, can be employed for the coarse registration between raw point clouds. The purpose of this process is to provide these point clouds, initially existing in independent coordinate systems, with relatively good initial positioning within a unified coordinate system. Ultimately, we duplicate the initially configured point cloud data into two sets. The first set (Scenario 1) is left without any subsequent processing (Figure 5a). For the second set (Scenario 2), based on the original data, non-bridge components, such as residual images of flowers, trees, and pedestrians, are identified and eliminated as noise points (Figure 5b). Throughout the entire process of point cloud processing,
no subsampling operations are involved to prevent the introduction of additional experimental variables that could affect the reliability of the final results.

Figure 4. Scan planning. Note: The red markers represent scanning locations on the bridge, while the blue markers denote scanning positions beneath the bridge.

Figure 5. (a) Scenario 1. (b) Scenario 2. Note: Both sets were formed following a coarse registration based on data from built-in sensors. However, for scenario 2, a noise removal procedure was implemented post-coarse registration in comparison to scenario 1.
3.4. Registration Strategies Used in Experiment

In this experiment, a total of six different point cloud registration strategies or algorithms were compared. These strategies are primarily used for fine registration and include point-to-point ICP (POICP), point-to-plane ICP (PLICP), colored registration combined with P2PL (CICP), robust kernels with Iterative Closest Point (KICP), global registration in conjunction with P2PL (GRICP), and fast global registration with P2PL (FGRICP). All algorithms were primarily executed using Python 3.8.4, utilizing libraries such as Open3D and Numpy.

3.4.1. Point-to-Point ICP (POICP)

POICP is the most fundamental ICP algorithm. Its objective is to finely register two adjacent point clouds—source and target—that are initially in relatively good alignment. This alignment is achieved by iteratively finding correspondence sets and updating transformation matrix $T$ to minimize an objective function [35,57]. This process is defined over three main steps.

The first step involves transforming the position of the source point cloud to the expected position. This transformation is subject to different conditions. If the two adjacent point clouds are already well-placed, meaning they are substantially overlapping, the transformation matrix adopts the identity matrix $T_0$. However, if the two adjacent point clouds are relatively distant, an initial coarse registration is required to attain the initial transformation matrix $T_1$. In this study, as each point cloud is already well-positioned due to the FS70 built-in sensor, we adopt the approach of using the identity matrix $T_0$:

$$T_0 = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix} \quad (1)$$

The second step is to perform the nearest point search within a certain range in the target point cloud, forming initial point pairs. The search range used in this experiment is 0.02 m. This step involves finding the correspondence set $K = \{(p_i, q_i)\}$ from the target point cloud $P$ and the source point cloud $Q$:

$$p_i \in P = \{p_1, p_2, \ldots, p_n\} \quad (2)$$

$$q_i \in Q = \{q_1, q_2, \ldots, q_n\} \quad (3)$$

$$(p_i, q_i) \in K = \{(p_1, q_1), (p_2, q_2), (p_3, q_3), \ldots, (p_n, q_n)\} \quad (4)$$

After forming the initial point pairs, the algorithm enters the phase of minimizing an objective function $E(T)$:

$$E(T) = \sum_{i=1}^{n} \left| |p_i - Tq_i| \right|^2 \quad (5)$$

This objective function represents the sum of the squared differences between each pair of corresponding points $(p_i, q_i)$, where $p_i$ is a point in the target point cloud and $Tq_i$ is a point in the transformed source point cloud. The transformation matrix $T$ is then updated by finding the $T$ that minimizes $E(T)$.

3.4.2. Point-to-Plane ICP (PLICP)

PLICP is also one of the fundamental algorithms in ICP. The principle of registration in P2PL is very similar to that in P2PO. However, P2PL, while considering the spatial relationship of point pairs as P2PO does, also takes into account the influence of the normal vector of the points [35]. The calculation of a normal vector for a point within a point...
cloud is typically a two-step process. The first step involves conducting a nearest neighbor search for a specific point in the point cloud. The second step involves determining the normal vector for that point, which is normally achieved either through methods such as principal component analysis (PCA) or plane fitting techniques. In the context of this research, we derive the normal vector and compute the objective function by leveraging the functionalities provided by the TransformationEstimationPointToPlane method within the Open3D library. This class offers efficient implementations for calculating the normal vectors for each point in a point cloud and for formulating the objective function. The objective function can be expressed as follows:

\[ E(T) = \sum_{i=1}^{n} \left( (p_i - Tq_i) \cdot n_{pi} \right)^2 \]  

(6)

where \( E(T) \) is the objective function that we aim to minimize. It represents the sum of squared distances from each point in the transformed source point cloud (\( q_i \)) to the corresponding point in the target point cloud (\( p_i \)), weighted by the dot product with the normal of the source point. \( T \) is the transformation matrix that we are solving for. It represents the rotation and translation that best aligns the \( q_i \) to the target \( p_i \). \( p_i \) denotes a point in the target point cloud \( P \). \( q_i \) represents a corresponding point in the source point cloud \( Q \), transformed by \( T \). \( n_{pi} \) is the normal vector at point \( p_i \) in the target point cloud.

3.4.3. Colored ICP (CICP)

Colored ICP is a variant based on the vanilla ICP (often referred to as POICP or PLICP). This method was proposed by Park at the ICCV in 2017 [58]. Unlike the vanilla ICP, which only considers the optimization of the geometric objective, Colored ICP additionally incorporates the optimization of color differences into the objective function. As a result, its objective function is a composite of two different objective functions, each with its own weight:

\[ E(T) = (1 - \delta)E_G(T) + \delta E_C(T) \]  

(7)

where \( T \) denotes the transformation matrix that we aim to estimate. \( E_C \) and \( E_G \) symbolize the photometric and geometric components, respectively. The weight factor \( \delta \), which lies in the range of \((0, 1)\), is established based on empirical data.

The geometric term \( E_G \) is, in fact, consistent with the objective function of the vanilla ICP, and the objective function based on PLICP is adopted in this paper. ([Please see Equation (6)].)

The color term \( E_C \) quantifies the disparity in color between point \( q_i \), represented as \( C(q_i) \), and the color of its projected counterpart on the tangent plane of point \( p_i \):

\[ E_C(T) = \sum_{i=1}^{n} \left( C_{pi} \left( f(Tq_i) \right) - C(q_i) \right)^2 \]  

(8)

where, \( Tq_i \) represents the i-th indexed point in the source point cloud after transformation, and \( f(.) \) is a projection function that maps \( Tq_i \) to the tangent plane of the i-th indexed point \( p_i \) in the target point cloud. \( C_{pi}(.) \) is a continuously varying function based on the color of \( p_i \) used to quantify the color of the projected point of \( q_i \) on the tangent plane of \( p_i \).
\[
    w_i = \frac{1}{(p_i - T q_i) \cdot n_{p_i}} \rho'( (p_i - T q_i) \cdot n_{p_i}) \tag{9}
\]

where \( q_i \) represents the source point in the point pair, \( p_i \) represents the corresponding target point in the point pair, \( T \) is the transformation matrix, \( n_{p_i} \) represents the norm used and calculated based on \( p_i \), and \( \rho' \) is the derivative of the Robust Kernel function applied to the squared residual between \( p_i \) and \( q_i \).

With these weights \( (w) \), the objective function of KICP can be expressed as:

\[
    E(T) = \sum_{i=1}^{n} w_i \left( (p_i - T q_i) \cdot n_{p_i} \right)^2 \tag{10}
\]

In this objective function, the summation extends over all point pairs, each contributing a term that is the squared residual weighted by the corresponding \( w \). This ensures that the well-matched point pairs (i.e., pairs with smaller residuals) have a more substantial influence on the final transformation estimate, while the outliers (i.e., pairs with larger residuals) have their influence substantially diminished. This mechanism underlies the robustness of the KICP algorithm.

By iteratively minimizing this weighted objective function using techniques like Gauss-Newton, we can estimate the optimal transformation matrix that best aligns the source point cloud to the target. This results in an enhanced point-to-plane ICP algorithm that is robust to outliers and can lead to more accurate point cloud registration.

3.4.5. Global Registration + ICP (GRICP)

The Iterative Closest Point (ICP) algorithm and its variants belong to the category of fine registration techniques, which implies the necessity for a relatively decent initial position to establish point correspondences for alignment. However, in many scenarios, even after processing with coarse registration-related methods, this decent initial position is still far from ideal. Global registration algorithms are designed to automatically optimize this initial position, facilitating the ICP algorithm to generate a more accurate set of point correspondences at the very beginning of its operation. The procedure for global registration predominantly unfolds in two phases. During the initial phase, the Fast Point Feature Histogram (FPFH) descriptors are computed for each individual point in both the source and target point clouds [60]. Subsequently, the Random Sample Consensus (RANSAC) method [61] was applied for global registration. With each RANSAC iteration, a subset of points is randomly picked from the source point cloud. By querying the nearest neighbor in the 33-dimensional FPFH feature space, we locate the corresponding points in the target point cloud.

In the subsequent phase, rapid pruning algorithms are employed early on to swiftly discard mismatches during the pruning step. Open3D library offers various pruning algorithms:

1. CorrespondenceCheckerBasedOnDistance confirms whether the distances between aligned point clouds are within a specific threshold.
2. CorrespondenceCheckerBasedOnEdgeLength assesses whether the lengths of two arbitrary edges (lines formed by two points), derived individually from source and target correspondences, exhibit similarity. This check verifies the conditions \( \| \text{source edge length} \| > 0.9 * \| \text{target edge length} \| \) and \( \| \text{target edge length} \| > 0.9 * \| \text{source edge length} \| \).
3. CorrespondenceCheckerBasedOnNormal evaluates the affinity of vertex normal for any given correspondences. This is performed by computing the dot product of two normal vectors, using a radian value as the threshold.
Only those matches that successfully pass through the pruning phase are leveraged to compute a transformation. This transformation is subsequently validated across the entirety of the point cloud. The crucial function deployed in this procedure is `registration_ransac_based_on_feature_matching`. The most crucial hyperparameter for this function is the `RANSACConvergenceCriteria`, which stipulates the maximum iterations for RANSAC and the confidence probability. The higher these parameters, the greater the accuracy of the results, albeit at the cost of a longer algorithm execution time.

Upon the completion of global registration, we are able to optimize an initial transformation matrix, denoted as $T_1$. The source point cloud, when transformed by this initial matrix, can align with the target point cloud at a more precise initial position. This alignment serves as the basis for executing the ICP algorithm.

### 3.4.6. Fast Global Registration + ICP (FGRICP)

In the process of global registration, RANSAC represents an iterative model estimation method, involving a substantial number of point pair proposals and evaluations. This constitutes a highly time-consuming algorithmic process. Fast Global Registration, however, is an alternative method for global registration, outperforming the RANSAC-based approach in terms of computational efficiency. This method was introduced by Qian-Yi Zhou in his 2016 paper titled Fast Global Registration [62].

The crux of the Fast Global Registration lies in the optimization of linear process weights to identify the best correspondences, thereby achieving model registration. In contrast to RANSAC, this method eliminates the need for copious model proposals and evaluations, resulting in significantly enhanced computational efficiency.

The Fast Global Registration algorithm proceeds as follows:

1. Feature Extraction and Matching: Initially, features are extracted from the two adjacent point clouds that need to be registered. Based on these features, matching is performed to identify potential point pairs. In the experiment, the feature is the same as the global registration, which is FPFH.
2. Graph Model Construction: Using the matched point pairs, a graph model is constructed where each edge represents a point pair.
3. Linear Process Weight Optimization: The next step involves the optimization of the linear process weights to determine the significance of each point pair. This step is central to the Fast Global Registration algorithm and the reason for its efficient performance. The aim of optimization is to achieve as much consistency as possible in the model after registration.
4. Iterative Optimization: Iterative optimization continues until the model converges or a predetermined number of iterations are reached.
5. Final Registration Result: Finally, the optimized linear process weights are used to obtain the final registration result.

Upon completion of the fast global registration, a similar effect to that of the global registration can be achieved, optimizing the initial positions of the source and target point clouds. Subsequently, superior precision can be realized through the utilization of the ICP algorithm.

### 3.5. Registration Error and Time Consumption

In previous studies, it has been substantiated that the quality of point cloud registration can significantly affect the accuracy of the resultant point cloud model [27]. The current research primarily employs two metrics, namely RE and TC, to assess the quality of point cloud registration tasks.

The computation of RE data primarily relies on the transformation matrix $T$, derived from each registration operation conducted by a specific registration algorithm. Utilizing
this transformation matrix $T$, we can translocate the current source point cloud. Subsequently, the data structure of the target point cloud is optimized based on the kd-tree algorithm. Further, using the nearest-neighbor search method within a 0.02 m range ($d$), each point in the target point cloud is matched with the closest point in the source point cloud, resulting in pairs. Subsequently, the Euclidean distances for each pair of points are computed, and the median of these distances is selected as the RE value for the current pair of point clouds. The process of calculating TC data predominantly relies on the time function from the time library. Before each execution of the related point cloud registration algorithm, a start time is established. Following the computation of the transformation matrix $T$ using the point cloud registration algorithm, an end time is set. The difference between the end time and the start time allows us to ascertain the time consumption required for the current point cloud registration algorithm to process the given source and target point clouds (Algorithm 1).

Algorithm 1. The pseudocode for the computational procedure of RE and TC.

Input: source point cloud $Q$, target point cloud $P$, distance threshold $d$
Output: registration error RE, time consumption TC

1: Initialize start time as now ()
2: For each registration operation in the specific registration algorithm:
3:       Calculate transformation matrix $T$
4:       Transform source point cloud $Q$ using $T$ to get $Q$
5: Set end time as now ()
6: Calculate TC as the difference between end time and start time
7: Initialize kd-tree based on target point cloud $P$
8: Initialize pairs as an empty list
9: For each point $q_i$ in $Q$:
10:       Find the closest point $p_i$ in $P$ using kd-tree within range $d$
11:       Add the pair ($p_i, q_i$) to pairs
12: Initialize distances as an empty list
13: For each pair ($p_i, q_i$) in pairs:
14:       Calculate the Euclidean distance between $p_i$ and $q_i$
15:       Add the distance to distances
16: Calculate RE as the median of distances
17: Return TC, RE

3.6. Significance and Value Analysis

This experiment primarily employs ANOVA (Analysis of Variance) plus Tukey’s HSD test and the Value Engineering Method to, respectively, 1. ascertain whether the differences among various registration methods under the same environmental conditions (with or without noise removal) possess statistical significance, and 2. quantify the value of specific registration algorithms under a particular environment.

ANOVA and Tukey’s HSD test are relatively straightforward statistical analysis methods, the specifics of which can be referred to in Abdi and Williams [63]. Following this, the emphasis of this article is to elaborate on the utilization of the Value Engineering method employed here to quantify the value of each registration strategy under the same environmental conditions.

The first step involves the calculation of the mean RE and TC for the same registration method under the same environmental conditions:
\[ \text{Ave}(R_{Ed}) = \frac{\sum_{i=1}^{n} R_{Edi}}{n} \]  
(11)

where \( \text{Ave}(R_{Ed}) \) denotes the average RE value under the \( d \)-th point cloud registration algorithm. \( R_{Ed} \) refers to the \( i \)-th RE value under the same \( d \)-th point cloud registration algorithm.

\[ \text{Ave}(T_{Cd}) = \frac{\sum_{i=1}^{n} T_{Cd i}}{n} \]  
(12)

where \( \text{Ave}(T_{Cd}) \) denotes the average TC value under the \( d \)-th point cloud registration algorithm. \( T_{Cd} \) refers to the \( i \)-th RE value under the same \( d \)-th point cloud registration algorithm.

The second step is to normalize the average RE and TC values obtained for each type of registration strategy under the same environmental conditions. This normalization operation scales these values proportionally within the range of (0, 1). The specific calculation method is as follows:

\[ \text{Ave}(R_{Ed})' = \frac{\text{Ave}(R_{Ed})}{\sum_{d=1}^{n} \text{Ave}(R_{Ed})} \]  
(13)

where \( \text{Ave}(R_{Ed})' \) represents the normalized \( R_{Ed} \), which is calculated by dividing any given \( R_{Ed} \) by the sum of all \( R_{Ed} \) under the same environmental conditions.

\[ \text{Ave}(T_{Cd})' = \frac{\text{Ave}(T_{Cd})}{\sum_{d=1}^{n} \text{Ave}(T_{Cd})} \]  
(14)

where \( \text{Ave}(T_{Cd})' \) represents the normalized \( T_{Cd} \), which is calculated by dividing any given \( T_{Cd} \) by the sum of all \( T_{Cd} \) under the same environmental conditions.

The third step involves a special treatment of \( \text{Ave}(R_{Ed})' \). Each \( \text{Ave}(R_{Ed})' \) calculated under the same environmental conditions for every registration algorithm needs to be subtracted from (1). This is because \( \text{Ave}(R_{Ed})' \) is intended to represent utility value in subsequent computations, where the logic is that the smaller the error, the larger the utility value. Thus, it is necessary to specially handle \( \text{Ave}(R_{Ed})' \) to achieve this effect:

\[ \text{Ave}(R_{Ed})'' = \frac{1 - \text{Ave}(R_{Ed})'}{5} \]  
(15)

where \( \text{Ave}(R_{Ed})'' \) represents the specially processed \( \text{Ave}(R_{Ed})' \) and, in subsequent calculations, it denotes the utility value of the current registration algorithm.

The final step involves calculating the value of each point cloud registration strategy under the same environmental conditions:

\[ V_{d} = \frac{\text{Ave}(R_{Ed})''}{\text{Ave}(T_{Cd})'} \]  
(16)

where \( V_{d} \) represents the value corresponding to the \( d \)-th point cloud registration algorithm under the same environmental conditions.

4. Result

4.1. Scenario 1 (Non-Noise Removal)

In this experiment, the quality of six different point cloud registration strategies was tested under non-noise-removal environmental conditions. Each point cloud registration strategy was applied to 28 pairs of point clouds, resulting in 28 RE data and 28 TC data for each strategy. In total, all registration strategies produced 168 RE data and 168 TC data.

4.1.1. Registration Error
According to the descriptive analysis of these 168 RE data (Table 1), the average error is ranked from largest to smallest as follows: POICP (7.32 mm), PLICP (6.38 mm), KICP (4.71 mm), CICP (3.93 mm), FGRICP (3.36 mm), GRICP (3.02 mm). We found that GRICP has the smallest average RE, indicating the highest precision in registration. Concurrently, the variance of the registration error of the GRICP algorithm is also the smallest, which signifies that this method exhibits excellent stability in performing registration tasks under a non-noise removal environment. Conversely, the POICP algorithm exhibits the highest average RE values as well as the largest variance, implying that the current algorithm has a lower overall accuracy and exhibits instability in its error range (Figure 6).

**Table 1.** Descriptive analysis for RE under non-noise removal condition.

<table>
<thead>
<tr>
<th>Method</th>
<th>Count</th>
<th>Sum</th>
<th>Average</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>POICP</td>
<td>28</td>
<td>204.87</td>
<td>7.32</td>
<td>6.49</td>
</tr>
<tr>
<td>PLICP</td>
<td>28</td>
<td>178.51</td>
<td>6.38</td>
<td>3.88</td>
</tr>
<tr>
<td>KICP</td>
<td>28</td>
<td>131.95</td>
<td>4.71</td>
<td>0.82</td>
</tr>
<tr>
<td>CICP</td>
<td>28</td>
<td>108.96</td>
<td>3.93</td>
<td>0.56</td>
</tr>
<tr>
<td>GRICP</td>
<td>28</td>
<td>107.97</td>
<td>3.02</td>
<td>0.42</td>
</tr>
<tr>
<td>FGRICP</td>
<td>28</td>
<td>117.39</td>
<td>3.36</td>
<td>0.46</td>
</tr>
</tbody>
</table>

**Figure 6.** Bar chart depicting registration errors from various registration methods under the non-noise removal condition. Note: The top of the bar represents the mean, and the error bar indicates the standard deviation.

Based on the results of the ANOVA analysis, we observe a $p$-value of 0.000 (Table 2). This denotes that, among the six groups, each employing a distinct point cloud registration algorithm to obtain RE data, there is at least one pair with a significant difference in their means. Further interpretation suggests that the registration method is a meaningful variable under conditions of non-noise removal. In a subsequent analysis utilizing the Tukey HSD method (Figure 7), we find that there is no statistical significance in the comparisons between POICP and PLICP ($p = 0.153$), KICP and CICP ($p = 0.333$), CICP and GRICP ($p = 0.185$), CICP and FGRICP ($p = 0.659$), and GRICP and FGRICP ($p = 0.900$). This infers that the point cloud accuracy is very close when these groups are compared individually.
Figure 7. Matrix plot based on the results from the Tukey HSD test. Note: The numbers represent adjusted p-values; blue indicates p-value less than 0.01, while red signifies p-value greater than 0.1.

Table 2. Summary of ANOVA analysis for RE under non-noise removal condition.

<table>
<thead>
<tr>
<th>Source of Variation</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>F</th>
<th>p-Value</th>
<th>F Crit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between Groups</td>
<td>415.51</td>
<td>5</td>
<td>83.10</td>
<td>39.43</td>
<td>0.000</td>
<td>2.27</td>
</tr>
<tr>
<td>Within Groups</td>
<td>341.45</td>
<td>162</td>
<td>2.11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>756.96</td>
<td>167</td>
<td></td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: SS = Sum of Square; df = Degree of Freedom; MS = Mean Square.

4.1.2. Time Consumption

In an analysis of 168 TC data (Table 3), the average TC values, ordered from highest to lowest, are as follows: GRICP (54.53 min), FGRICP (32.53 min), CICP (4.19 min), KICP (4.18 min), POICP (3.98 min), and PLICP (3.86 min). Of these, the PLICP algorithm has the smallest mean value. This implies that, based on the overall sample, the PLICP algorithm can compute the required results in the shortest amount of time. Additionally, it also has the smallest variance, demonstrating a high level of stability in its time consumption. Conversely, GRICP has the highest mean value, indicating a high time cost for running the algorithm and a relatively low level of stability in its execution time (Figure 8).

Based on the ANOVA analysis of the total TC data, I obtained a p-value of 0.000 (Table 4), which indicates that the registration method is a significant influencing factor in the time consumption of point cloud registration. Further, based on the Tukey HSD test for intergroup data analysis (Figure 9), we found that POICP vs. PLICP (p = 0.900), POICP vs. KICP (p = 0.900), POICP vs. CICP (p = 0.900), PLICP vs. KICP (p = 0.900), PLICP vs. CICP (p = 0.900), and KICP vs. CICP (p = 0.900), and there is no significant difference in time consumption for the five groups of registration methods.

Table 3. Descriptive analysis for TC under non-noise removal condition.

<table>
<thead>
<tr>
<th>Groups</th>
<th>Count</th>
<th>Sum</th>
<th>Average</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>POICP</td>
<td>28</td>
<td>111.48</td>
<td>3.98</td>
<td>0.42</td>
</tr>
<tr>
<td>PLICP</td>
<td>28</td>
<td>108.21</td>
<td>3.86</td>
<td>0.37</td>
</tr>
<tr>
<td>KICP</td>
<td>28</td>
<td>116.95</td>
<td>4.18</td>
<td>0.43</td>
</tr>
<tr>
<td>CICP</td>
<td>28</td>
<td>117.31</td>
<td>4.19</td>
<td>0.45</td>
</tr>
<tr>
<td>GRICP</td>
<td>28</td>
<td>1526.87</td>
<td>54.53</td>
<td>51.86</td>
</tr>
<tr>
<td>FGRICP</td>
<td>28</td>
<td>910.94</td>
<td>32.53</td>
<td>19.34</td>
</tr>
</tbody>
</table>
Table 4. Summary of ANOVA analysis for TC under non-noise removal condition.

<table>
<thead>
<tr>
<th>Source of Variation</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>F</th>
<th>p-Value</th>
<th>F Crit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between Groups</td>
<td>64,964.54</td>
<td>5</td>
<td>12,992.91</td>
<td>1069.61</td>
<td>0.000</td>
<td>2.27</td>
</tr>
<tr>
<td>Within Groups</td>
<td>1967.87</td>
<td>162</td>
<td>12.15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>66,932.41</td>
<td>167</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: SS = Sum of Square; df = Degree of Freedom; MS = Mean Square.

Figure 8. Bar chart depicting time consumption from various registration methods under the non-noise removal condition. Note: The top of the bar represents the average, and the error bar indicates the standard deviation.

Figure 9. Matrix plot based on the results from the Tukey HSD test. Note: The numbers represent adjusted p-values; blue indicates p-value less than 0.01, while red signifies p-value greater than 0.1.

4.1.3. Value Comparison

According to the calculations using the value engineering method, we obtained the value scores ($V_d$) for six different registration methods in a non-noise removal environment (Table 5). From highest to lowest, they are: CICP (4.26), PLICP (4.16), KICP (4.13),
POICP (3.87), FGRICP (0.56), and GRICP (0.34). The final data indicates that the CICP algorithm offers the highest cost-performance ratio, while the GRICP has the lowest (Figure 10).

![Figure 10](image)

**Figure 10.** The bar chart illustrates the final value scores associated with various registration methods under non-noise removal conditions. Note: Ave (REd) is directly proportional to Value (Vd), while Ave (TCd) is inversely proportional to Vd.

**Table 5.** Value comparison between different registration methods under non-noise removal condition.

<table>
<thead>
<tr>
<th>Registration Method</th>
<th>Ave (REd)''</th>
<th>Ave (TCd)''</th>
<th>Vd</th>
</tr>
</thead>
<tbody>
<tr>
<td>POICP</td>
<td>0.15</td>
<td>0.04</td>
<td>3.87</td>
</tr>
<tr>
<td>PLICP</td>
<td>0.16</td>
<td>0.04</td>
<td>4.16</td>
</tr>
<tr>
<td>KICP</td>
<td>0.17</td>
<td>0.04</td>
<td>4.13</td>
</tr>
<tr>
<td>CICP</td>
<td>0.17</td>
<td>0.04</td>
<td>4.26</td>
</tr>
<tr>
<td>GRICP</td>
<td>0.18</td>
<td>0.53</td>
<td>0.34</td>
</tr>
<tr>
<td>FGRICP</td>
<td>0.18</td>
<td>0.32</td>
<td>0.56</td>
</tr>
<tr>
<td>Total</td>
<td>1.00</td>
<td>1.00</td>
<td>17.31</td>
</tr>
</tbody>
</table>

4.2. Scenario 2 (Noise Removal)

In this experiment, the quality of six different point cloud registration strategies was tested under noise removal conditions. Each point cloud registration strategy was applied to 28 pairs of point clouds, resulting in 28 RE data and 28 TC data for each strategy. In total, all registration strategies produced 168 RE data and 168 TC data.

4.2.1. Registration Error

According to the descriptive analysis (Table 6), the six different point cloud registration methods produce an average point cloud error in a noise-removal environment, ranked from highest to lowest as follows: POICP (4.98 mm), PLICP (4.20 mm), KICP (3.61 mm), FGRICP (3.10 mm), CICP (2.86 mm), and GRICP (2.79 mm). Of these, GRICP remains the algorithm with the highest precision in point cloud registration, and its variance of 1.23 mm also indicates good stability. Conversely, POICP has the lowest precision in...
point cloud registration and, with a variance of 3.30 mm, it is relatively unstable (Figure 11).

Table 6. Descriptive analysis for RE under noise removal condition.

<table>
<thead>
<tr>
<th>Method</th>
<th>Count</th>
<th>Sum</th>
<th>Average</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>POICP</td>
<td>28</td>
<td>139.34</td>
<td>4.98</td>
<td>3.30</td>
</tr>
<tr>
<td>PLICP</td>
<td>28</td>
<td>117.59</td>
<td>4.20</td>
<td>2.14</td>
</tr>
<tr>
<td>KICP</td>
<td>28</td>
<td>101.09</td>
<td>3.61</td>
<td>1.64</td>
</tr>
<tr>
<td>CICP</td>
<td>28</td>
<td>79.94</td>
<td>2.86</td>
<td>1.30</td>
</tr>
<tr>
<td>GRICP</td>
<td>28</td>
<td>78.18</td>
<td>2.79</td>
<td>1.23</td>
</tr>
<tr>
<td>FGRICP</td>
<td>28</td>
<td>86.73</td>
<td>3.10</td>
<td>1.24</td>
</tr>
</tbody>
</table>

Figure 11. Bar chart depicting registration errors from various registration methods under the noise removal condition. Note: The top of the bar represents the mean, and the error bar indicates the standard deviation.

Based on the ANOVA analysis (Table 7), a final \( p \)-value of 0.000 was obtained, indicating that the registration method remains a statistically significant factor influencing registration error in a noise removal environment. Further analysis through the Tukey HSD test resulted in the following findings (Figure 12): POICP vs. PLICP \( (p = 0.262) \), PLICP vs. KICP \( (p = 0.564) \), KICP vs. CICP \( (p = 0.292) \), KICP vs. GRICP \( (p = 0.210) \), KICP vs. FGRICP (0.685), CICP vs. GRICP \( (p = 0.900) \), CICP vs. FGRICP \( (p = 0.900) \), and GRICP vs. FGRICP \( (p = 0.900) \). The \( p \)-values of all these eight combinations exceed 0.1, indicating that there is no significant difference in RE produced between the registration methods within these combinations. Apart from these eight combinations, significant differences were displayed when comparing the remaining combinations.

Table 7. Summary of ANOVA analysis for RE under noise removal condition.

<table>
<thead>
<tr>
<th>Source of Variation</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>F</th>
<th>( p )-Value</th>
<th>F Crit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between Groups</td>
<td>103.98</td>
<td>5</td>
<td>20.80</td>
<td>11.49</td>
<td>0.000</td>
<td>2.27</td>
</tr>
<tr>
<td>Within Groups</td>
<td>293.16</td>
<td>162</td>
<td>1.81</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>397.14</td>
<td>167</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: SS = Sum of Square; df = Degree of Freedom; MS = Mean Square.
4.2.2. Time Consumption

Based on the descriptive analysis of TC (Table 8), we can rank the different point cloud registration methods in terms of average time consumption from highest to lowest in a noise-removal environment: GRICP (32.76 min), FGRICP (15.68 min), KICP (3.03 min), CICP (2.90 min), PLICP (2.70 min), and POICP (2.61 min). Among these, POICP has the least runtime and an intra-group variance of only 0.16 min, indicating the stability of this algorithm in terms of time consumption. In contrast, GRICP takes the longest time, with a variance as high as 54.91. This implies that the algorithm’s performance in terms of time consumption is highly unstable, with a large range, making it hard to predict (Figure 13).

Based on the ANOVA analysis (Table 9), we obtained a p-value of 0.000, indicating that the registration method has a significant impact on time consumption in a noise-removal environment. Further analysis using the Tukey HSD test (Figure 14) resulted in POICP vs. PLICP (p = 0.900), POICP vs. KICP (p = 0.900), POICP vs. CICP (p = 0.900), PLICP vs. KICP (p = 0.900), PLICP vs. CICP (p = 0.900), and KICP vs. CICP (p = 0.900). A total of five intra-group comparisons of registration methods revealed no significant differences, and significant differences were found in the remaining intra-group comparisons. This result is almost completely consistent with the results obtained in the Noise-removal test.

Table 8. Descriptive analysis for TC under noise removal condition.

<table>
<thead>
<tr>
<th>Method</th>
<th>Count</th>
<th>Sum</th>
<th>Average</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>POICP</td>
<td>28</td>
<td>73.06</td>
<td>2.61</td>
<td>0.16</td>
</tr>
<tr>
<td>PLICP</td>
<td>28</td>
<td>75.51</td>
<td>2.70</td>
<td>0.20</td>
</tr>
<tr>
<td>KICP</td>
<td>28</td>
<td>84.84</td>
<td>3.03</td>
<td>0.24</td>
</tr>
<tr>
<td>CICP</td>
<td>28</td>
<td>81.23</td>
<td>2.90</td>
<td>0.22</td>
</tr>
<tr>
<td>GRICP</td>
<td>28</td>
<td>917.27</td>
<td>32.76</td>
<td>54.91</td>
</tr>
<tr>
<td>FGRICP</td>
<td>28</td>
<td>439.09</td>
<td>15.68</td>
<td>6.40</td>
</tr>
</tbody>
</table>

Table 9. Summary of ANOVA analysis for TC under noise removal condition.

<table>
<thead>
<tr>
<th>Source of Variation</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>F</th>
<th>p-Value</th>
<th>F Crit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between Groups</td>
<td>21,201.66</td>
<td>5</td>
<td>4240.33</td>
<td>409.46</td>
<td>0.000</td>
<td>2.27</td>
</tr>
<tr>
<td>Within Groups</td>
<td>1677.65</td>
<td>162</td>
<td>10.36</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>22,879.31</td>
<td>167</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: SS = Sum of Square; df = Degree of Freedom; MS = Mean Square.
4.2.3. Value Comparison

According to the calculations using the value engineering method (Table 10), we obtained the value scores ($V_d$) for six different registration methods in a non-noise removal environment. From highest to lowest, they are: CICP (3.57), PLICP (3.56), POICP (3.52), KICP (3.28), FGRICP (0.65), and GRICP (0.32). The final data indicates that the CICP algorithm offers the highest cost-performance ratio, while the GRICP has the lowest (Figure 15).

**Table 10.** Value comparison between different registration method under noise removal condition.

<table>
<thead>
<tr>
<th>Registration Method</th>
<th>Ave (REo)$''$</th>
<th>Ave (TCa)$'$</th>
<th>$V_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>POICP</td>
<td>0.15</td>
<td>0.04</td>
<td>3.52</td>
</tr>
<tr>
<td>PLICP</td>
<td>0.16</td>
<td>0.05</td>
<td>3.56</td>
</tr>
<tr>
<td>KICP</td>
<td>0.17</td>
<td>0.05</td>
<td>3.28</td>
</tr>
<tr>
<td>CICP</td>
<td>0.17</td>
<td>0.05</td>
<td>3.57</td>
</tr>
<tr>
<td>GRICP</td>
<td>0.17</td>
<td>0.55</td>
<td>0.32</td>
</tr>
</tbody>
</table>

**Figure 13.** Bar chart depicting time consumption from various registration methods under the noise removal condition. Note: The top of the bar represents the mean, and the error bar indicates the standard deviation.

**Figure 14.** Matrix plot based on the results from the Tukey HSD test. Note: The numbers represent adjusted $p$-values; blue indicates $p$-value less than 0.01, while red signifies $p$-value greater than 0.1.
Figure 15. The bar chart illustrates the final value scores associated with various registration methods under noise removal conditions. Note: Ave (RE)d” is directly proportional to Vd, while Ave (TC)d’ is inversely proportional to Vd.

4.3. Scenario 1 vs. Scenario 2

In order to investigate the impacts of noise removal and non-noise removal on registration error and time consumption, we conducted t-tests on the same registration method under different conditions. This was performed to ascertain whether varying environments would result in significant effects.

Based on the t-test analysis for RE under the same registration method under different conditions (Table 11), we obtained the p-values for the same registration method under different conditions, which were respectively: POICP (0.000), PLICP (0.000), KICP (0.001), CICP (0.000), GRICP (0.295), and FGRICP (0.359). Among these, with the exception of GRICP and FGRICP, all other algorithms exhibited significant differences between the conditions of noise removal and non-noise removal. This suggests that, in most cases, noise removal serves as a significant influencing factor on registration error, and it can help reduce the registration error and enhance registration accuracy.

Table 11. The result of T-test for RE between noise removal and non-noise removal condition.
According to the $t$-test analysis for TC under the same registration method for different conditions (Table 12), we obtained the $p$-values for the same registration method under different conditions, which were respectively: POICP (0.000), PLICP (0.000), KICP (0.001), CICP (0.000), GRICP (0.000), and FGRICP (0.000). Among these, all point cloud registration algorithms demonstrated significant differences between the conditions of noise removal and non-noise removal. This indicates that noise removal is a universally significant influencing factor on time consumption, and it can very effectively reduce time spent, greatly enhancing operational efficiency.

Table 12. The result of $t$-test for TC between noise removal and non-noise removal condition.

<table>
<thead>
<tr>
<th>Method</th>
<th>$t$-Value</th>
<th>$p$-Value</th>
<th>Average TC (Non-Noise Removal)</th>
<th>Average TC (Noise Removal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>POICP</td>
<td>9.46</td>
<td>0.000</td>
<td>3.98</td>
<td>2.61</td>
</tr>
<tr>
<td>PLICP</td>
<td>8.15</td>
<td>0.000</td>
<td>3.86</td>
<td>2.70</td>
</tr>
<tr>
<td>KICP</td>
<td>7.39</td>
<td>0.000</td>
<td>4.18</td>
<td>3.03</td>
</tr>
<tr>
<td>CICP</td>
<td>8.32</td>
<td>0.000</td>
<td>4.19</td>
<td>2.90</td>
</tr>
<tr>
<td>GRICP</td>
<td>11.15</td>
<td>0.000</td>
<td>54.53</td>
<td>32.76</td>
</tr>
<tr>
<td>FGRICP</td>
<td>17.58</td>
<td>0.000</td>
<td>32.53</td>
<td>15.68</td>
</tr>
</tbody>
</table>

5. Discussion

5.1. Key Findings

The following encapsulates the key findings from our experiment:

- In conditions without noise removal, the average RE for six different point cloud registration methods, listed from highest to lowest, are as follows: POICP (7.32 mm), PLICP (6.38 mm), KICP (4.71 mm), CICP (3.93 mm), FGRICP (3.36 mm), GRICP (3.02 mm). When operating under conditions with noise removal, the average Registration Error (RE) values from highest to lowest are: POICP (4.98 mm), PLICP (4.20 mm), KICP (3.61 mm), FGRICP (3.10 mm), CICP (2.86 mm), and GRICP (2.79 mm).

- Under conditions without noise removal, the TC for six different point cloud registration methods, listed from highest to lowest, are as follows: GRICP (54.53 min), FGRICP (32.53 min), CICP (4.19 min), KICP (4.18 min), POICP (3.98 min), and PLICP (3.86 min). When operating under conditions with noise removal, the Time Consumption (TC) values, from highest to lowest, are: GRICP (32.76 min), FGRICP (15.68 min), KICP (3.03 min), CICP (2.90 min), PLICP (2.70 min), and POICP (2.61 min).

- In the non-noise removal condition, the score of value engineering for CICP was 4.26, while under the noise removal condition, it scored 3.57. In both conditions, CICP achieved the highest score. This suggests that CICP is the most cost-effective point cloud registration algorithm in this experiment. Its overall performance is outstanding, with average registration errors of 3.93 mm (non-noise removal) and 2.86 mm (noise removal), respectively. The average time consumption is 4.19 min (non-noise removal) and 2.90 min (noise removal), respectively.

- In the analysis of value engineering, GRICP scored 0.34 and 0.32 under non-noise removal and noise removal conditions, respectively. These are the lowest scores under both conditions, suggesting a relatively low cost-effectiveness. Although when solely comparing the registration error, its averages of 3.02 mm (non-noise removal) and 2.79 mm (noise removal) are among the best in their respective groups, the time consumption is extremely long, at 54.53 min (non-noise removal) and 32.76 min (noise removal), respectively.

- Noise removal can significantly enhance registration accuracy and reduce computational time for the majority of point cloud registration algorithms. Among the six algorithms involved in this study, the average reduction in registration error (RE) was 1.20 mm, and the average reduction in time consumption (TC) was 7.27 min under
the noise removal condition compared to the non-noise removal condition. Specifically, for the POICP algorithm, the RE and TC were reduced by 32% and 34%, respectively, after noise removal. For PLICP, there was a reduction of 34% in RE and 30% in TC. KICP showed a decrease of 23% in RE and 28% in TC, CICP demonstrated a 27% and 31% decline in RE and TC respectively, GRICP witnessed a reduction of 8% in RE and 40% in TC, and for FGRICP, the RE and TC decreased by 8% and 52%, respectively, after noise removal.

5.2. Future Development Trend

In this study, under the context of building data related to bridging structures, we investigated six different mainstream point cloud registration methods and the impact of noise removal on registration quality. Among them, CICP achieved the best overall score, indicating the highest cost-effectiveness. Unlike other algorithms, CICP incorporates color as an additional dimension in its calculations. This might be a significant reason why it can achieve relatively high accuracy within a reasonable timeframe. However, if high-resolution cameras are not used during point cloud collection or if color information cannot be acquired, this may pose certain limitations to this method. KICP builds upon the traditional ICP by utilizing kernel functions, but when it comes to bridging construction environments, its performance does not exhibit a notably better improvement compared to traditional ICP, like POICP and PLICP. Although the GRICP and FGRICP algorithms underperformed in the aspect of value engineering, they exhibited outstanding performance in registration error metrics under both non-noise removal and noise removal conditions. This can primarily be attributed to these two algorithms conducting more global registration at the initial stage compared to other ICP variants. This puts the ICP algorithm in a comparatively optimal initial state, suggesting that a better starting position contributes to greater accuracy in point pair formation by the ICP. Hence, we can infer that a high-quality initial point pair collection can significantly enhance registration accuracy.

The trend for future point cloud registration methods necessitates continuous improvement in the accuracy of point pair formation. At present, most algorithms depend on the range search for nearest points or certain feature computations, such as FPFH, 3DSC and SHOT, among others. The former has relatively lower accuracy and requires an excellent initial position to circumvent falling into local optima. In contrast, the latter, although potentially more precise, incurs high computational costs in resources and time, impeding the development of iterative algorithms.

The author proposes the exploration of a deep learning framework grounded in big data, which learns point cloud registration features to create pairs. The potential of this approach resides in its capacity to reconcile accuracy with computational costs, two often conflicting aspects in this field. Theoretically, by employing a data-driven approach, the system could generalize and predict more accurate point pairs, thus potentially improving registration performance.

Nonetheless, the transition towards this approach introduces a new set of challenges that researchers need to confront. Firstly, implementing a deep learning model for this task could involve high learning costs. This is due to the computationally expensive nature of these models, especially when dealing with extensive datasets, as well as the expertise required to design and optimize such models.

Secondly, the scarcity of high-quality datasets for training the model poses a significant challenge. While there are public datasets available for point cloud registration, the extent and variety of these datasets might not be sufficient for training a robust deep learning model that can generalize well to different environments and scenarios.

Lastly, the complexities involved in designing an appropriate deep learning architecture and the difficulties in training such a model cannot be overlooked. This process requires careful consideration of various factors such as the choice of layers, the use of activation functions, the optimization of loss functions, and others. The training of the model also needs to be properly managed to avoid issues such as overfitting or underfitting.
5.3. Experimental Statement and Limitation

1. In this study, all point cloud data were not subsampled during the calculation of RE and TC. The only exception was when calculating the FPFH for GRICP and FGRICP, where subsampling was applied due to the extensive duration of the global registration process associated with these methods.

2. The measure of TC in this study only encompasses the period from the commencement of the algorithm to the determination and transformation of the source point cloud via the transformation matrix T. It does not include the subsequent time for Registration Error (RE) calculation. The calculation of RE involves traversing all newly formed point pairs within the RE algorithm and computing the Euclidean distance, a process that is notably time-consuming.

3. The primary objective of the RE and TC metrics in this paper is to aid in evaluating the significance of the registration method and noise removal as influential factors, as well as assisting in assessing the value of each registration method. The results of RE and TC in this study should only be used as a reference under similar environmental conditions to those detailed in this paper, and where no point cloud subsampling is performed. This is because the calculation of RE in this study is highly sensitive to environmental factors and point cloud subsampling. In particular, subsampling that increases the intervals between point clouds can have a substantial impact on the results.

4. The RE calculation method utilized in this study has certain limitations, including its high sensitivity to subsampling, low robustness, and a propensity for local optimal solutions due to the range search for the nearest points. To address these limitations, the author proposes an alternative approach for future researchers. Specifically, the use of markers during scanning could provide a more accurate method for evaluating registration accuracy. By placing markers within the source and target point clouds, these known points can serve as references. Consequently, the Euclidean distance between these markers can be computed as the RE, providing a more accurate reflection of registration accuracy.

6. Conclusions

The main purpose of this study is to investigate the significant impact of registration methods and noise removal on the registration quality of point clouds. The experimental design primarily employed the method of controlled variables to explore the significant influence of registration methods under the same environmental conditions and identical noise removal settings. Additionally, the effect of noise removal was studied by comparing the same registration method under different noise removal conditions. To facilitate the exploration of influencing factors and quantify registration quality, the study utilized RE and TC as quantitative indicators. The final results indicate that the CICP algorithm is the best-performing point cloud registration method in bridging construction environments. Under non-noise removal conditions and noise-removal conditions, its average RE can reach 3.93 mm and 2.86 mm, respectively, while the TC can achieve 4.19 min and 2.86 min, respectively. On the other hand, GRICP stands out as the algorithm with the highest average point cloud registration accuracy in bridging construction contexts, attaining an RE of 3.02 mm under non-noise removal and 2.79 mm under noise removal conditions. However, this algorithm is also the most time-consuming, with time expenditures averaging 54.53 min and 32.76 min under the respective conditions. A marked increase in accuracy and a reduction in processing time were observed for most algorithms following noise removal. Specifically, the POICP algorithm experienced a 32% reduction in RE and a 34% decline in TC after noise removal. Similarly, PLICP observed a 34% and 30% reduction in RE and TC, respectively. KICP showcased a decline of 23% in RE and 28% in TC, CICP manifested a 27% and 31% drop in RE and TC, respectively, GRICP observed an 8%
reduction in RE and a 40% decline in TC, and for FGRICP, RE and TC decreased by 8% and 52%, respectively, subsequent to noise removal.

**Author Contributions:** Conceptualization, Z.Z.; methodology, Z.Z.; writing-original draft and experiment, Z.Z.; writing-review and editing and supervision, S.R.; writing-review and editing, T.C.; writing-review and editing and validation, A.P. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The data presented in this study are available on request from the first author.

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

**References**


2. Bouzas, Ō.; Cabaleiro, M.; Conde, B.; Cruz, Y.; Riveiro, B. Structural health control of historical steel structures using HBIM. *Autom. Constr.* 2022, 140, 104308.


16. Eker, R. Comparative use of PPK-integrated close-range terrestrial photogrammetry and a handheld mobile laser scanner in the measurement of forest road surface deformation. *Measurement* 2023, 206, 112322.


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.