Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 2480703, 15 pages
https://doi.org/10.1155/2020/2480703

Research Article

Nontarget-Based Global Registration for Unorganized Point Clouds Obtained in the Dynamic Shipyard Environment

Jinho Song and Kwanghee Ko

The School of Mechanical Engineering, Gwangju Institute of Science and Technology, Gwangju 61005, Republic of Korea

Correspondence should be addressed to Kwanghee Ko; khko@gist.ac.kr

Received 1 July 2020; Revised 8 October 2020; Accepted 20 October 2020; Published 2 November 2020

Academic Editor: Nazrul Islam

In this paper, we propose a method for registering unorganized point clouds without using targets or markers. Motivated by the 4-points congruent sets (4PCS) algorithm, which is a nontarget-based registration method commonly used in the related fields, we develop a feature-based 4PCS algorithm (F-4PCS). The method combines the basic approach of the 4PCS algorithm with geometric feature information to produce consistent global registration results efficiently. We use the features from the point feature histogram descriptor and the ones that capture the surface curvature. The experimental results show that the proposed method successfully registers point clouds of both the outdoor and indoor scenes and demonstrates better performance than the existing 4PCS-based registration methods.

1. Introduction

In industrial environments such as a megastructure construction site or a shipbuilding yard, scanning is a popular method to obtain the geometric information of various objects that are being assembled or constructed or of the shipyard itself for modelling its current configuration for various purposes. Currently, 3D scanners such as LiDAR's are mostly used to produce 3D point clouds of an object. For a large object, multiple scans at different positions are required to cover its entire shape because of the limited range of the scanner. However, the data points of each scan are defined with respect to the local coordinate system associated with its scan position. Therefore, the entire shape of the target object cannot be represented by moving the scan datasets into the reference coordinate system. Instead, each scanned data should be transformed against the reference frame to represent the shape of the target object while the geometric consistency among the scan datasets is being considered. This process is called *registration*.

There are two types of registration: global and local. The global registration computes the best transformation to align two point sets without considering their initial relative positions or orientations. The local registration, on the other hand, iteratively finds a local optimum solution that registers two point sets, starting from an approximate solution determined by the initial relative positions and orientations of the point sets.

The most widely used local registration algorithm is an iterative closest point (ICP) algorithm [1]. When a source and a target point clouds are given, it finds the correspondence between them based on the closest point pairs, computes the rigid body transformation for registration, and updates the relative position of the source point cloud with respect to the target. This process is repeated until an optimum solution is reached. In the past decades, a large number of ICP variants were introduced to improve the original ICP algorithm. Lim [2] proposed a point-to-plane ICP scheme to improve registration performance. LM-ICP algorithm improves robustness using a nonlinear minimization technique, and various extended versions of ICP algorithm were developed to improve both speed and performance [3–5]. In [6], global optimization was studied based on ICP. In addition, the application of ICP algorithm is extended to nonrigid registration. Nonrigid ICP algorithms were developed, which consider affine transformation by

using a more efficient function or simplifying the affine transformation problem into a quadratic programming problem [7].

ICP-based methods generally solve a local optimization problem. Given a rough initial solution, they iteratively improve the solution to an optimal solution. Therefore, they have the limitation that appropriate initial relative positions and orientations of input point sets should be provided to find the optimum solution that best registers the two point clouds. Moreover, nonrigid registration is not considered in this paper because the 3D scanner is calibrated sophisticated enough to have no scaling or skew problem. In this paper, we focus on the problem of global registration with no scaling effect. Therefore, ICP-based algorithms cannot be used in our application. Instead, they can be considered for a more refined result after the global registration solution is obtained.

The core step of global registration is to find correspondence between two sets of points. Once the correspondence is obtained, a rigid body transformation can be computed from the correspondence to register the two scans. Manual registration is highly time consuming because it requires browsing each point set and looking for points or geometric parts that exist in both of the point sets to establish correspondence. Furthermore, it is not easy to find matching points between two point sets visually. Therefore, a method that can find correspondence automatically is needed.

The correspondence between two datasets can be determined automatically with targets or markers. This approach uses multiple known geometric shapes or markers, such as spheres or distinctive patterns on the paper. The targets or markers are detected, and the corresponding relations between them are established for registration [3, 8–11]. It can produce accurate registration results as long as the targets or markers are detected accurately and robustly; however, it requires that the targets should be carefully installed and maintained in the area of interest. Furthermore, they should remain unmoved during the acquisition step, which is a restricted condition in an actual industrial site. Besides, it is not easy to detect targets or markers robustly.

A nonmarker-based method can be used to solve such problems. Various features and descriptors computed from the point sets are used for computing correspondence. Some of the features are feature histograms (PFH), fast PFH, radius-based surface descriptors, ensembles of shape functions, or compact geometric feature descriptors [12–16]. However, it may take a long time to compute the features because all of the points need to be used in the computation. Filtering the point cloud can decrease the computation time by reducing the number of points in each point set; however, a feature loss can occur, thereby resulting in unsuccessful registration. Moreover, the sampling density, the sampled point distribution, and the noise level sensitively affect the feature computation, and therefore, robust and consistent features are not easy to obtain.

To overcome the drawbacks of feature-dependent methods, feature-independent methods are developed such as the reformative component registration algorithm [17] and the 4-points congruent sets (4PCS) algorithm [18]. In [17], the authors proposed a ship component accuracy evaluation framework. They propose an algorithm that registers the points of a component and the corresponding CAD model by performing parallel transformation and finding a correct registration direction without feature information. However, matching the center of gravity of the point sets using parallel transformation can fail when one input data is part of other data. Moreover, one entity should be expressed as a NURBS surface to find the registration direction, which is not a usual case in practice.

The 4PCS algorithm is an extended version of the RANdom SAmple Consensus (RANSAC) algorithm by [19]. It finds a congruent base in two point clouds P and Q by introducing a 4-point set as a base to dramatically reduce the time for estimating the optimal correspondence. The main advantage of the 4PCS algorithm is that it requires point coordinates only and does not consider all the points during the computation. However, the 4-point set in Q is randomly selected, which means that its registration result can be inconsistent or unstable depending on the choice of the 4-points sets. We may avoid such a problem by considering more points for the choice of the 4-point sets. However, it would significantly extend the computation time when large point sets are considered, which compromises the main advantage of the algorithm.

There are some modified versions of the 4PCS algorithm to boost both its performance and speed: the SUPER 4PCS algorithm [20], the key-point-based 4PCS (K-4PCS) algorithm [21], and the semantic key-point-based 4PCS algorithm (SK-4PCS) [22]. The flow of these algorithms is essentially the same as that of the 4PCS method. The SUPER 4PCS algorithm focuses on how to reduce the time for searching congruent 4-points sets by reducing the search range. The K-4PCS algorithm, which was designed to handle huge 3D terrestrial laser scanning data, introduces a preprocessing stage that reduces the number of points through a voxel grid filtering technique and estimates key points. The key points are used to construct congruent point sets, which the 4PCS algorithm takes as input to produce a matching result. Similar to the K-4PCS algorithm, the SK-4PCS algorithm also applies the voxel grid filter and extracts semantic key points from the nonground points. It shows faster computation speed and better performance compared to the K-4PCS algorithm, although the input point cloud is limited to building scenes. These methods, however, have the same problem as the 4PCS algorithm related to the random base generation step.

In this study, we propose a feature-based 4PCS (F-4PCS) algorithm, a hybrid approach that combines the stochastic selection step of the 4PCS method for a reduced computation time with a deterministic search step of the feature-based algorithm for improved consistency. The algorithm uses features based on the PFH descriptor and an additional one using the surface curvature at each point. The reduction of computation time is achieved by reducing the number of congruent sets based on the feature correspondence information. The consistency of the registration is realized by choosing the congruent 4-point sets using feature

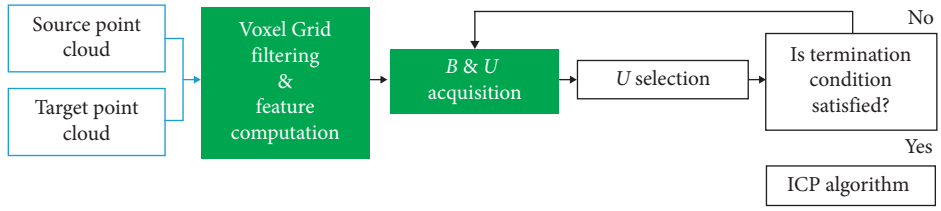


Figure 1: Flow chart of the proposed algorithm, where the boxes in the second and third columns are newly added/modified stages.

information from the input point sets, thereby, eliminating any random selection.

The main contribution of the proposed method is as follows. The method overcomes the problems of the existing 4PCS algorithm and its variants. When a source and a target point clouds are given, 4PCS algorithm and its variants generate a random base or a 4-point set, with which the registration computation is performed. This means that a different registration result could occur even though the same input data is provided because of the random generation of the base. Namely, it often happens that registration can either succeed or fail, even for the same input data. To solve this inconsistency problem, we propose an approach that generates bases using feature information. We newly introduce an improved PFH by using a new feature that can capture the curvature property of the underlying geometry and fuse it with the 4PCS method to overcome the limitation of the 4PCS method while maintaining the computation time. We checked that it always produces the same registration results if the same inputs and parameters are given, while the computation time for registration is maintained similar to that of 4PCS algorithm and its variants.

2. Technical Approach

2.1. Overview. In this section, we present the schematic diagram of the proposed algorithm, as shown in Figure 1. We assume that the target point cloud Q and the source point cloud P are given as input. First, we preprocess P and Q using a voxel grid filter to obtain filtered point clouds P_v, Q_v and compute features. Next, we compute the 4-point sets $B = \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}$, where $B \in Q_v$ and $U' = \{U_1, U_2, \ldots, U_k\}$, where U_i is the ith set of 4-points in P_v that is assumed to be aligned with B when a proper transformation matrix is applied to P_v up to a certain threshold δ. The goal of the proposed algorithm is to find the best corresponding base $U \in U'$ when the base B is chosen in Q_v. The general structure of the proposed algorithm is equal to the structure of 4PCS algorithm, but two important steps of 4PCS algorithm are modified: base B acquisition step and U' acquisition step. In B acquisition step, the first main step of our approach, fixed bases B are generated using the proposed feature information. The second step is to find the set U' using the feature information. The third step is the selection step that chooses the best correspondent set U to B from U'. After these three steps are performed, a transformation matrix $\mathbf{T}_{\text{opt}}$ is computed using the relationship between B and U. If the transformed P_v accurately aligns with Q_v up to δ after $\mathbf{T}_{\text{opt}}$ is applied, the algorithm is terminated. If not, the whole procedure is repeated until the termination condition is satisfied or all the bases are used. After the algorithm is terminated, the general point-to-point iterative closest point (ICP) algorithm is used to refine the registration result [1]. In the following sections, the preprocessing steps and the main process of the proposed algorithm will be explained in detail. The notations used in this study are summarized in Table 1.

2.2. Voxel Grid Filtering. Massive 3D scan point clouds, P and Q, are obtained by a 3D scanner. Therefore, we cannot directly use them because the computation time taken to process them would grow almost exponentially. We avoid this problem by reducing the numbers of points of P and Q with a voxel grid filter [23]. During the voxel grid filtering procedure, we compute and save points $\mathbf{p}_k^l, k = 1, \ldots, L_i$ inside each subdivided 3D box and also compute the centroid or the mean vector $\mathbf{c}_l$ for each points set $\mathbf{p}_k^l$ of an input point cloud using equation

$$\mathbf{c}_l = \frac{1}{L_i} \sum_{k=1}^{L_i} \mathbf{p}_k, \tag{1}$$

where the centroids $\mathbf{c}_l$ become the new points of the input point cloud. We apply this procedure to P and Q and obtain filtered point clouds P_v and Q_v.

2.3. Feature Computation. The F-4PCS algorithm finds 4-point sets B and U in P_v and Q_v using features. The features used in the procedure are based on the PFH descriptor that extracts unique features from the input point cloud using only the Euclidean coordinates. We choose PFH descriptor because it shows reliable registration results of laser scans [12–16, 24]. The feature histogram acquisition stage for a given query point $\mathbf{p}_c$ in the input point cloud is as follows [24]. First, we search the K neighbourhood point set of $\mathbf{p}_c$. Then, three geometric features f_1, f_2, f_3 from the PFH descriptor are computed:

$$\begin{aligned} f_1 &= \mathbf{v} \cdot \mathbf{n}_j, \\ f_2 &= \mathbf{u} \cdot \frac{(\mathbf{p}_i - \mathbf{p}_j)}{\mathbf{p}_i - \mathbf{p}_j}, \\ f_3 &= \arctan\left(\mathbf{w} \cdot \mathbf{n}_j, \mathbf{u} \cdot \mathbf{n}_j\right), \end{aligned} \tag{2}$$

where $\| \cdot \|$ is the Euclidean L^2 norm [25], $\mathbf{p}_i$ and $\mathbf{p}_j$ are from the K neighborhood point set of a given query point $\mathbf{p}_c$, and

Table 1: List of notations.

Variable	Explanation
General	
B	4-point set $\{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}$ of the target point cloud Q
U'	Congruent 4-point sets to B of the source point cloud P
U	Best correspondent set among U'
Voxel grid filtering and feature computation steps	
L	Number of points in a 3D box
$\mathbf{c}$	Centroid of the points in a 3D box
N	Number of 3D boxes or filtered points in the voxel grid filtering step
$\mathbf{p}_k$	Original point before the voxel grid filtering step
C	Covariance matrix for the points in 3D box
$\mathbf{p}_i, \mathbf{p}_j$	Points after the voxel grid filtering step
K	Number of the used neighbourhood points to each query point $\mathbf{p}_k$
FH	Proposed feature histogram created by the features f_1, f_2, f_3, f_4
P_v, Q_v	Filtered point clouds by the voxel grid filter
F-4PCS algorithm steps	
$\mathbf{p}, \mathbf{q}$	Points of P_v and Q_v
$\mathbf{e}$	Intersection point of the line $\mathbf{ab}$ and $\mathbf{cd}$ of B
S	Number of the nearest points to points of B
H	Histogram to generate point pairs R_1, R_2
R	Point pair which is composed of two points α_i^1, α_j^2
β	Number of the bases of B' in F-4PCS algorithm
o	Overlap ratio of the transformed P and Q
δ	Main threshold to control the range of congruent 4-points sets U'
$\mathbf{T}$	Set of transformation matrixes computed from U' and B'
s	Score threshold, a termination condition of F-4PCS algorithm
$\mathbf{T}_{\text{opt}}$	Best transformation matrix among $\mathbf{T}$

$\mathbf{n}_j$ is a normal of $\mathbf{p}_j$.$\mathbf{u}, \mathbf{v}, \mathbf{w}$ are axes of the Darboux frame defined by [15]. In the F-4PCS procedure, we use four features f_1, f_2, f_3, and f_4, three of which are the same as those of the PFH descriptor. The main key difference from the PFH descriptor is that we use one more feature f_4. To compute f_4, we first obtain covariance matrices C^i using equation (3) for all the filtered points of the input point cloud.

$$C^i = \frac{1}{L_i} \sum_{k=1}^{L_i} \left(\mathbf{p}_k^i - c_i\right)\left(\mathbf{p}_k^i - \mathbf{c}_i\right)^T, \tag{3}$$

where $\mathbf{p}_k^i$ are the removed points in each 3D box and $\mathbf{c}_i$ is a centroid obtained from equation (1). Once C^i is computed, we compute the eigenvalues λ_m^i, where $m = 1, 2, 3$ and $i = 1, \ldots, N^*$ and N^* is the number of the subdivided boxes with L_l and $l = 1, \ldots, N$ using the symmetric QR algorithm [26]. Then, finally we compute the feature f_4 using equation (4) and surface curvature σ_i based on the eigenvalues of the covariance matrix C:

$$\begin{aligned} \sigma_i &= \frac{\lambda_1^i}{\lambda_1^i + \lambda_2^i + \lambda_3^i}, \quad \lambda_1^i < \lambda_2^i < \lambda_3^i, \\ f_4 &= \frac{\sigma_i}{\sigma_j}, \quad \sigma_i < \sigma_j. \end{aligned} \tag{4}$$

The surface curvature depicts how much the underlying shape is bent at a given point. In this case, the surface curvature close to zero implies that the local geometric structure is close to a planar surface. Note that the computed eigenvalues λ_m^i and λ_m^j for the two different points $\mathbf{p}_i$ and $\mathbf{p}_j$ are used to construct f_4. The main role is to recover the lost geometric information before the voxel grid filtering because λ_m^i and λ_m^j are computed from the removed points within each 3D box. Then, we can form a new feature set f_1, f_2, f_3, f_4.

The procedure is repeated until the feature sets for all the points in the K-neighborhood are computed. After the feature sets are obtained, we perform a histogram binning process that transforms the feature sets into the bins of a histogram and creates a feature histogram FH with $81\,(3^4)$ bins. For a more detailed instruction of the binning process, refer to [27].

2.4. F-4PCS Algorithm. The essence of the F-4PCS algorithm includes how to acquire B and U', and how to select U that yields the best result. In this section, each step of the F-4PCS algorithm is presented in detail.

2.5. **B** *Acquisition Step.* The first step of the F-4PCS algorithm is to obtain the initial bases $\{B_1, B_2, \ldots, B_m\}$. We construct the bases using the computed feature histograms for all the points in Q_v by the following procedure. When one query point $\mathbf{a} = \mathbf{p}_q$ is selected in Q_v, two most nearest points, b and c are chosen using the similarity measure $\|\text{FH}_a - \text{FH}_i\|$, where the lower value indicates higher similarity and FH_a represents a feature histogram of $\mathbf{a}$ in Q_v. FH_i is a feature histogram of the point $\mathbf{w}_i$, $\mathbf{w}_i \in Q_v - \{\mathbf{a}\}$. Then, a

triplet $\{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ is obtained. We choose the fourth point $\mathbf{d}$ that is the closest to the triangle created by the triplet to build a 4-point set like the 4PCS algorithm does. We repeat this procedure until m bases are obtained, where m is the number of the points in Q_v.

The next step is to filter the m bases to reduce the number of candidates because using all the bases is time consuming. The filtering procedure is to directly use the method by [28], called a reciprocity test, which works as follows. First, for each $\mathbf{q} \in Q_v$, we find the most similar or nearest neighbour of $\mathbf{q}$ among $\mathbf{p} \in P_v$ using the computed feature histograms and the similarity measure, and we similarly find the nearest neighbour of $\mathbf{p}$ among Q_v for each $\mathbf{p} \in P_v$.

Then, we build correspondence sets κ_1^i from these correspondence results. After computing κ_1^i, we now select correspondence pairs $(\mathbf{p}, \mathbf{q})$ from κ_1^i if and only if $\mathbf{p}$ is the nearest point from $\mathbf{q}$ and $\mathbf{q}$ is the nearest point of $\mathbf{p}$ from $(\mathbf{p}, \mathbf{q})$. The selected correspondence set is defined as κ_2.

In B acquisition step, we use the reciprocity test to select the reliable bases. The base B is a 4-point set. So, we perform the reciprocity test for all the four points. And, if one of the points passes the test, we assume that the base is reliable. We repeat this procedure for all the bases and build the base set B' that contains the total β bases. We move to the next step after the base B is selected from B'. Algorithm 1 summarizes the entire B acquisition step, where FH^p indicates the feature histogram of P_v.

2.6. $\mathbf{U}'$ *Acquisition Step.* After the 4-point set B is acquired, we need to find the best congruent 4-points set U from P_v. In order to build U, we construct the congruent 4-point sets U'. First, the affine invariant ratios r_1 and r_2 are computed from B using equation

$$r_1 = \frac{\|\mathbf{a} - \mathbf{e}\|}{\|\mathbf{a} - \mathbf{b}\|}, \qquad r_2 = \frac{\|\mathbf{c} - \mathbf{e}\|}{\|\mathbf{c} - \mathbf{d}\|}, \tag{5}$$

where $\mathbf{e}$ is the intersection point of the lines $\mathbf{ab}$ and $\mathbf{cd}$. Next, we move to the point pair generation step to build point pairs R_1^g, R_2^l, which will be used to construct U'.

Point pairs R_1, R_2 are 2-point sets from P_v, which are constructed based on the Euclidean distances $\|\mathbf{a} - \mathbf{b}\|$ and $\|\mathbf{c} - \mathbf{d}\|$. In this step, we use a new input parameter S of F-4PCS algorithm, the nearest points in P_v of base B which is used to filter unreliable point pairs using the computed feature histograms FH and FH^p. The procedure of the point pair generation step is composed of two steps: R_1 and R_2 generation steps.

In R_1 generation step, we build point pairs based on the distance $\|\mathbf{b} - \mathbf{a}\|$, meaning we construct 2-point sets congruent to the first point $\mathbf{a}$ and the second point $\mathbf{b}$ of the base B in Q_v. For $\mathbf{a}$, we search the S nearest points in P_v using $\|\mathrm{FH}_a - \mathrm{FH}_i^p\|$, where FH_a is the feature histogram of $\mathbf{a}$ and FH_i^p is the feature histograms of the ith point in P_v. Then, we collect these nearest points and define them as the point set α^1, which is the set of congruent point candidates to $\mathbf{a}$. In the next step, we similarly find S nearest points of $\mathbf{b}$ in P_v using $\|\mathrm{FH}_b - \mathrm{FH}_i^p\|$, where FH_b is the feature histogram of $\mathbf{b}$, and define these points as the another point set α^2. Finally, we build point pairs R_1^g using α^1 and α^2. In order to build R_1^g, we construct two histogram sets $H_h = \{\alpha_1^h, \ldots, \alpha_S^h | h = 1, 2\}$ with the length S for each histogram. Then, the point pairs R_1^g are created by H_h and the constraint that $\|\alpha_j^2 - \alpha_i^1\| \le \|\mathbf{b} - \mathbf{a}\|$ from equation (6), which describes the general form of one point pair. Note that the computation complexity of the point pair generation step is $\mathrm{O}(S^2 + k)$ since two histograms H_h with length S are extensively compared:

$$R_1 = \left\{\left(\alpha_i^1, \alpha_j^2\right) | \alpha_i^1 \neq \alpha_j^2\right\}, \quad \left\|\alpha_j^2 - \alpha_i^1\right\| \le \|\mathbf{b} - \mathbf{a}\|. \tag{6}$$

In R_2 generation step, a similar procedure is repeated to build point pairs R_2^l, except that we find S nearest points of $\mathbf{c}$ and $\mathbf{d}$ to obtain the point sets α^1, α^2. After obtaining the point pairs R_1^g, R_2^l, the point pair generation step ends and moves to the final step of U' acquisition step.

In the final step, the immediate points $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4\}$ are estimated using all possible combinations of the point pairs R_1, R_2 and the ratios r_1, r_2, are defined in equation

$$\mathbf{e}_1 = \alpha_i^1 + r_1\left(\alpha_j^2 - \alpha_i^1\right), \qquad \mathbf{e}_1 = \alpha_i^1 + r_2\left(\alpha_j^2 - \alpha_i^1\right). \tag{7}$$

Here, for simplicity, only two immediate points corresponding to R_1 are described. Finally, we compare the computed immediate points $(\mathbf{e}_1, \mathbf{e}_2)$ from R_1 with $(\mathbf{e}_3, \mathbf{e}_4)$ from R_2 and construct $U_i \in U'$ when $\|\mathbf{e}_3 - \mathbf{e}_1\|$, $\|\mathbf{e}_4 - \mathbf{e}_1\|$, $\|\mathbf{e}_3 - \mathbf{e}_2\|$, and $\|\mathbf{e}_4 - \mathbf{e}_2\| < 2\delta$; thus, the set U_i that is composed of two point pairs are approximately congruent to B [18]. The complexity of the F-4PCS algorithm is $\mathrm{O}(S^2 + k)$ because generating point pairs takes most of the computation time. After U' is obtained, the initial transformation matrices $\mathbf{T} = \{\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_k\}$ between B and U', and the overlap ratios o are computed. To estimate an overlap ratio o, we apply the transformation matrix $\mathbf{T}_i \in \mathbf{T}$ to the source point cloud P_v so that P_v aligns to the target point cloud Q_v.

For each point of P_v, we estimate the distance between the point and its closest point in Q_v, and if the distance is less

Input: point clouds P_v, Q_v and feature histograms FH, FH^p
Output: Selected base set B'
for $\mathbf{q} \in Q_v$ do
  $\mathbf{a} \longleftarrow \mathbf{q}$;
  $\mathbf{b}, \mathbf{c} \longleftarrow$ two nearest points of $\mathbf{a}$ in Q_v using FH;
  $\mathbf{d} \longleftarrow \mathrm{MIN}_{q_i \in Q_v} \|\mathbf{q}_i - \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}\|$ where $i = 1, \ldots, m$;
  $B \longleftarrow \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}$;
Set Initial bases B_i, where $i = 1, \ldots, m$;
Construct correspondence sets κ_1^i using FH, FH^p;
for B_i do
  Apply reciprocity tests to $\{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}\} \in B$;
  if $\mathbf{a}$ or $\mathbf{b}$ or $\mathbf{c}$ or $\mathbf{d}$ passes the reciprocity test **then**
  $B \in B'$;
return B'

Algorithm 1: B acquisition step.

than an uncertainty measure ($\delta > 0$), we conclude that the points align and count them to compute the overlap ratio. This process continues for all values of $U_i \in U'$ to compute the overlap ratio set $o_i = \{o_1, o_2, \ldots, o_k\}$, and both $\mathbf{T}$ and o_i are saved for the next main step. Algorithm 2 summarizes the entire U' acquisition step.

2.7. **U** *Selection Step.* U selection step is identical to the step of 4PCS procedure. We choose the congruent base with the highest overlap ratio $o_{\mathbf{best}}$ among o_i as the best congruent base $U = \{\mathbf{a}', \mathbf{b}', \mathbf{c}', \mathbf{d}'\}$ from $U^{'}$, as shown in Figure 2. Then, we check whether the overlap ratio $o_{\mathbf{best}}$ satisfies a termination threshold s and terminates if $o_{\mathbf{best}} \geq s$, choosing $\mathbf{T}_i$ of U as an final initial transformation matrix $\mathbf{T}_{\text{opt}}$, the final output of F-4PCS algorithm. Otherwise, the algorithm goes back to the B acquisition step, selecting the next base B from $B^{'}$ and repeats the whole procedure until s is satisfied or all the bases are used.

If the F-4PCS algorithm fails to reach the score s after using all the bases, we collect the best transformation matrices $\mathbf{T}_i^{\beta}$ from U_i, where β is the total number of $B^{'}$, and choose $\mathbf{T}_i^{\beta}$ with the highest overlap ratio among $o_{\mathbf{best}}^{\beta}$ from $\mathbf{T}_i^{\beta}$ as the final initial transformation matrix $\mathbf{T}_{\text{opt}}$. After the F-4PCS algorithm is terminated, the point-to-point ICP algorithm is applied to refine the registration result for further optimization. Algorithm 3 summarizes the proposed F-4PCS algorithm, including the B and U' acquisition steps, which are unique from the 4PCS-based algorithms.

2.8. Problems of 4PCS and SUPER 4PCS Algorithms. The F-4PCS algorithm shares the same overall structure as the 4PCS algorithm; however, it can handle the problems of the latter effectively.

First, the base B is randomly generated in the 4PCS algorithm by randomly sampling the points from Q. Thus, the registration result of the 4PCS algorithm can be different for each run, causing the inconsistent registration results. However, the F-4PCS algorithm produces consistent registration results because the F-4PCS algorithm does not randomly generate a base but a fixed base for each run. Second, in the $U^{'}$ acquisition step, generation of R_1 and R_2 is different between the 4PCS algorithm and the F-4PCS algorithm. The 4PCS algorithm goes over all the points in P to build R_1 and R_2 by only allowing point pairs the distance of which is less than $\|\mathbf{a} - \mathbf{b}\|$ or $\|\mathbf{c} - \mathbf{d}\|$ of B, resulting in its time complexity of $O(n^2 + k)$. Thus, the problem is that the total number of possible point pair combinations becomes extremely large when the size of the input data is huge, leading to a long computation time. In the F-4PCS algorithm, on the other hand, two histograms H_1 and H_2 with the length $S \ll n$ obtained from the method in $U^{'}$ acquisition step are used to compute the corresponding point pairs, and so its complexity becomes $O(S^2 + k)$, which is lower than that of the 4PCS algorithm.

The SUPER 4PCS algorithm shares the structure of the F-4PCS algorithm because it is an extended version of the 4PCS algorithm. The time complexity of the SUPER 4PCS algorithm is $O(n + k_1 + k_2)$, which can be lower than the complexity of the F-4PCS algorithm depending on the choice of the parameter S. However, the SUPER 4PCS algorithm randomly generates bases in the similar way to the 4PCS algorithm; thus, it also produces different registration results, even though the same input data is used. Therefore, it produces inconsistent registration results, which is the same problem we face with the 4PCS algorithm.

3. Results and Discussion

To demonstrate the performance of the F-4PCS algorithm, three actual 3D datasets acquired from different scanners are used. The algorithm is compared with the 4PCS, K-4PCS, and SUPER 4PCS algorithms with the same voxel grid filter size. SK-4PCS algorithm is not used for experiments because datasets do not have major ground planes, which is necessary for SK-4PCS algorithm. The common input parameters are an approximation level δ that affects the range of possible matched bases U', a score s that is a termination condition, an overlap fraction that is a ratio showing how much two point clouds overlap, and the voxel grid filter size that is the cube dimension size used to filter the points in meter. For the F-4PCS, 4PCS, and K-4PCS algorithms, the parameters δ, s and the overlap fraction are fixed to 0.1, 0.9, and 0.5. However, $\delta = 0.5$ was used for the SUPER 4PCS algorithm with Dataset 1 experiments because, with $\delta = 0.1$, the algorithm always failed to find a registration result. The voxel grid filter size is set to 2.5 for Dataset 1, 0.4 for Dataset 2, and 2.0 for Dataset 3, where 1.0 corresponds to one meter. In the F-4PCS algorithm, we set the number of neighbourhood points to be 25 ($K = 25$) for each point in the feature computation step and set S to 150 for Dataset 1, 350 for Dataset 2, and 200 for Dataset 3. For a fair comparison of the existing methods with the proposed one, we used the same parameter values, except one case. The selected parameter values were chosen empirically by considering the properties of the input data points and the performance of each algorithm. The original implementation of the SUPER 4PCS algorithm is modified to use OpenGR library [29] and a voxel grid filter for a fair comparison. To boost the speed of computation of the algorithms, OpenMP library [30] is applied to all the algorithms. The F-4PCS, 4PCS, and K-4PCS algorithms are compiled and tested in a C++ environment with Visual Studio 2015 while the SUPER 4PCS algorithm is compiled and tested with Visual Studio 2017 because OpenGR library is not compatible with Visual Studio 2015. A hardware system used in the tests has an Intel core i7-6700k CPU and 32 GB RAM on the 64-bit Windows 10 platform.

To test the performance of the algorithms, the root mean square error (RMSE) from equation (8) for the transformed P and Q is used, where $\mathbf{q}_i \in Q$ is the closest point to $\mathbf{p}_i \in P$, and n is the total number of points in P.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (\mathbf{p}_i - \mathbf{q}_i)^2}{n}}. \tag{8}$$

In addition, we estimate RMSE_{ICP}, the RMSE after the general point-to-point ICP algorithm. We estimate the

Input: base B, parameter S, point clouds P_v, Q_v and feature histograms FH, FH^p
Output: Congruent base set U', Transformation matrix set $\mathbf{T}$ and overlap ratio set o_i
Compute ratios r_1, r_2;
$R_1^g, R_2^l \longleftarrow$ Point pairs generation step;
for $R_1 \in R_1^g$ **do**
  for $R_2 \in R_2^l$ **do**
    Compute immediate points $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4\}$;
    $U_i \longleftarrow \{R_1, R_2\}$;
    if $\|\mathbf{e}_i - \mathbf{e}_j\| < 2\delta$, where $i = 3, 4$ and $j = 1, 2$ **then**
      Compute transformation matrix $\mathbf{T}_i$ and overlap ratio o;
      $U_i \in U'$;
      $\mathbf{T}_i \in \mathbf{T}$;
      $o \in o_i$;
return $U', \mathbf{T}, o_i$

Algorithm 2: U' acquisition step.

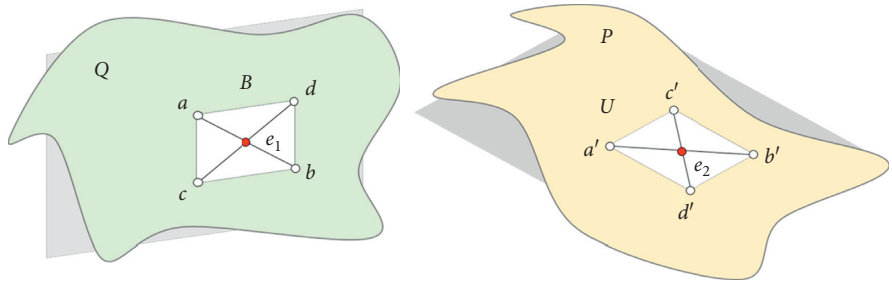


Figure 2: Selected base $B = \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}$ in Q and its congruent base $U = \{\mathbf{a}', \mathbf{b}', \mathbf{c}', \mathbf{d}'\}$ in P.

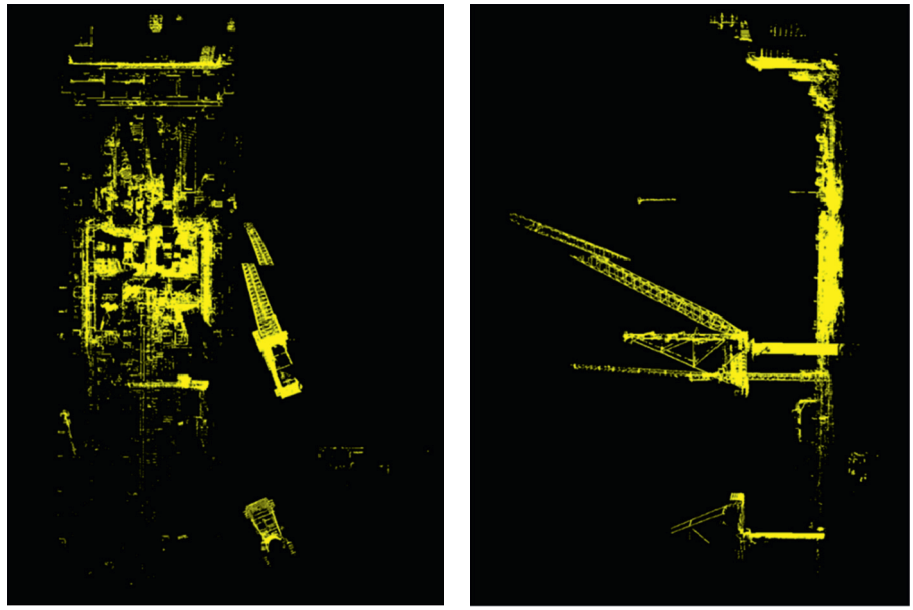
Input: point clouds P, Q
Output: Initial transformation matrix set $\mathbf{T}_{\text{opt}}$ that aligns P to Q
Apply a Voxel grid to P and Q to reduce their size;
Compute feature histograms FH$_i$, FH$_j^p$ from P_v, Q_v;
Base set $B' \longleftarrow$ Algorithm 1: B acquisition step;
for $B \in B'$ **do**
  $U', \mathbf{T}, o_i \longleftarrow$ Algorithm 2: U' acquisition step;
  $U, \mathbf{T}_i, o_{\mathbf{best}} \longleftarrow U$ selection step;
  if $o_{\mathbf{best}} \geq s$ **then**
    $\mathbf{T}_{\text{opt}} \longleftarrow \mathbf{T}_i$;
    return $\mathbf{T}_{\text{opt}}$;
Set Transformation matrices $\mathbf{T}_i^\beta$ and overlap ratios $o_{\mathbf{best}}^\beta$;
Set $\mathbf{T}_i \in \mathbf{T}_i^\beta$, where its $o^i = \text{MAX}o_{\mathbf{best}}^i$;
$\mathbf{T}_{\text{opt}} \longleftarrow \mathbf{T}_i$;
return $\mathbf{T}_{\text{opt}}$

Algorithm 3: F-4PCS algorithm.

computation time t in seconds. The tested algorithms except F-4PCS produce different results for the same inputs because of their randomly selected bases; thus, we run each test five times for each two input point clouds to compute the average of the computation time and their RMSEs.

3.1. Dataset 1. Dataset 1 contains 13 point clouds of an offshore plant with multiple structures and cranes. They were obtained using a FARO scanner moving around the offshore plant. The numbers of points in each scanned point set are in the range of two to four millions, and the average number of the filtered points by the voxel grid filter is 2,375. We consecutively perform a pairwise registration for Dataset 1, so there are a total of 12 pairwise registrations. Notice that a huge crane in the right side was continuously moving during the 3D data acquisition step, so the positions of the crane are different in each point cloud or part of the crane was cut off in some data as we can observe in Figure 3.

Table 2 shows the average evaluation results for the 4PCS, K-4PCS, SUPER 4PCS, and F-4PCS algorithms with Dataset 1. The 4PCS algorithm shows reasonable performance with



(a) (b)

Figure 3: Source input data of case 1 with (a) a front view and (b) a side view.

Table 2: Dataset 1 registration results of 4PCS, K4PCS, SUPER 4PCS, and F-4PCS algorithms.

	4PCS			K-4PCS		
Avg.	t (s)	RMSE (m)	RMSE_{ICP} (m)	t (s)	RMSE (m)	RMSE_{ICP} (m)
	14.6	**1.39**	**0.94**	**8.35**	**4.19**	**1.49**
	Super 4PCS			F-4PCS (proposed)		
Avg.	t (s)	RMSE (m)	RMSE_{ICP} (m)	t (s)	RMSE (m)	RMSE_{ICP} (m)
	3.85	**2.23**	**1.15**	**6.73**	**1.23**	**0.84**

the RMSE of 1.49 m and the average computation time of 14.6 s. However, the 4PCS algorithm sometimes fails to find correct results for Cases 5, 6, 7, and 9. The K-4PCS algorithm is faster than the 4PCS algorithm with an average computation time of 8.35 s. However, the K-4PCS algorithm fails more often than the 4PCS algorithm for a repetitive test of each case, resulting in the worst average RMSE of 4.19 m. The SUPER 4PCS algorithm achieves the fastest computation time among the tested algorithms with only 3.85 s, and its performance is the best in some registrations such as Case 3 with RMSE 0.99 m. However, it sometimes fails to find correct registration results just like the 4PCS and K-4PCS algorithms for Cases 5, 6, 7 and 9, where the 4PCS algorithm also fails, resulting in the average RMSE of 2.23 m. The F-4PCS algorithm fails to find a correct registration of Case 5 but succeeds to register more input cases than the others. Figure 4 shows the registration results of Cases 6 and 9 by 4PCS algorithm and the F-4PCS algorithm, where the red point cloud is a target and the green point cloud is a source. While 4PCS algorithm fails to register the top parts of the main structure in Figures 4(b) and 4(e) for both cases, the F-4PCS successfully registers the main structures, producing the average RMSE 1.23 m and the computation time of 6.73 s.

The inconsistent results of the 4PCS, K-4PCS, and SUPER 4PCS algorithms are shown in Figure 5. All the three algorithms produce different registration results for each run, even though the same input data is used. On the contrary, the F-4PCS algorithm achieves a consistent registration result.

3.2. Dataset 2. Dataset 2 is composed of indoor scenes around pipes in the shipyard. It includes various pipes, conveyor belts, wrapped structures, and workers, as we can see in Figure 6, where a black circle indicates the location of a 3D scanner during the 3D data acquisition step. We can see that a scan range of some data such as Figure 6(a)–6(c) are smaller than the range of the other data such as Figure 6(d). This is because the pipes or structures blocked the scanning range during the scanning step. The data sequence is composed of 11 point clouds that were consecutively taken by moving the laser scanner, and the density of each point set is far lower than that in Dataset 1. We set S to 350, which is higher than that for Dataset 1 because the acquired point clouds of Dataset 2 contain more noise than the point clouds of Dataset 1. The total numbers of the points in each point cloud are in the range of about seventy thousands to one million, and the average number of the filtered points by the voxel grid filter is 1,230. Overall, ten pairwise cases of registration are performed.

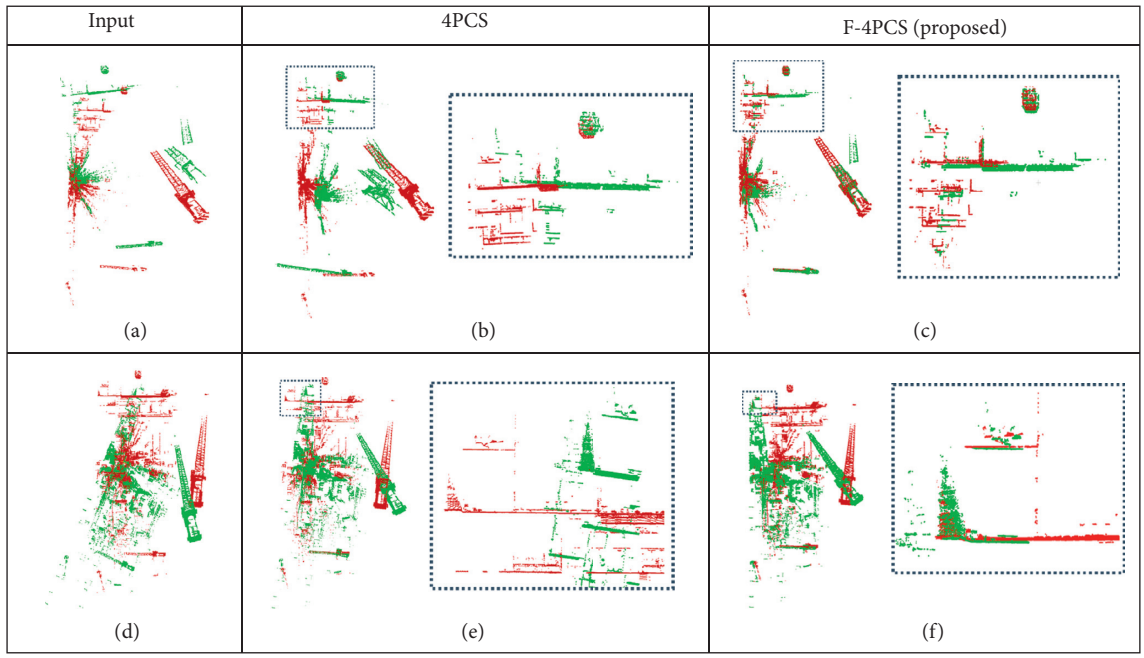


Figure 4: Input data sequences of case 6 (first row) and case 9 (second row) and the initial registration results. (a) and (d) are the input datasets of case 6 and case 9 before registration, (b) and (e) are the results by 4PCS, and (c) and (f) by the F-4PCS algorithms. The dashed rectangular areas are magnified for better visualization.

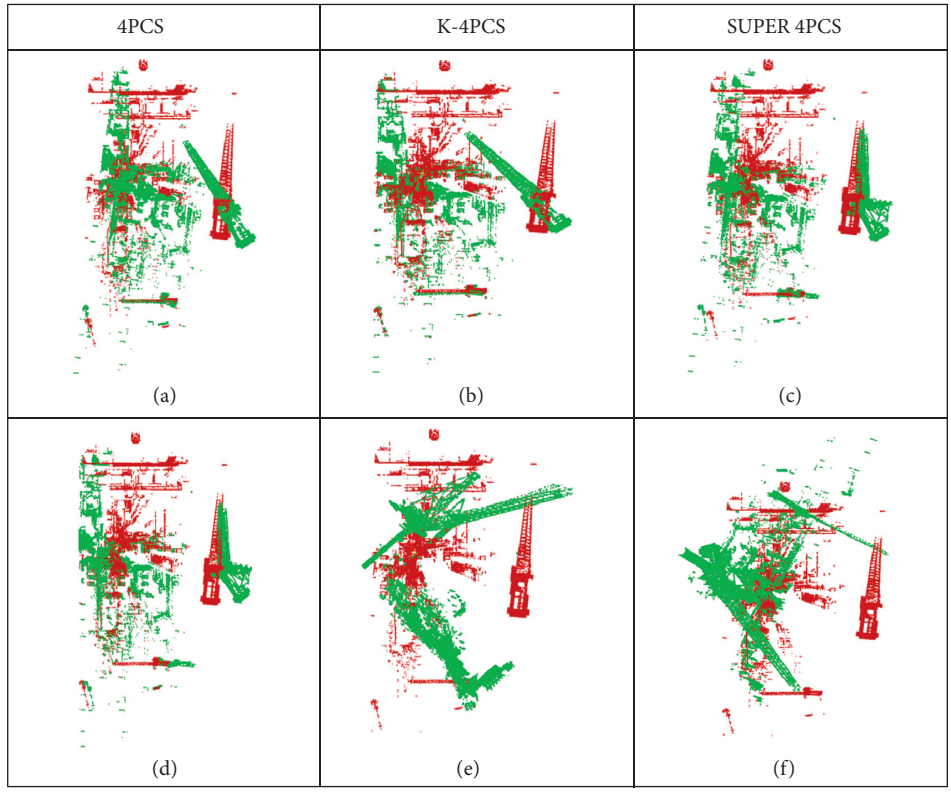
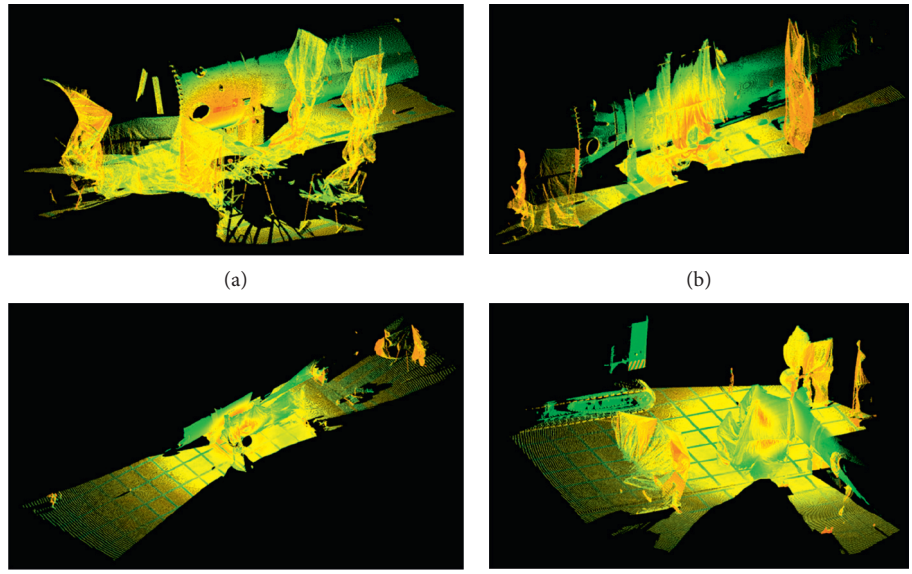


Figure 5: Inconsistent initial registration results of case 9. (a) and (d) are the results by 4PCS, (b) and (e) by K-4PCS, and (c) and (f) by SUPER 4PCS algorithms when the input of case 9 in Figure 4(c) is given.

The registration results of the 4PCS, K-4PCS, SUPER 4PCS, and F-4PCS are summarized in Table 3. The 4PCS algorithm sometimes cannot find appropriate congruent 4-points sets U and fails in 8 out of 10 pairwise registrations. Therefore, the algorithm finishes earlier than expected because it reaches the maximum iteration number without successful registration results. Thus, it ends up with a high RMSE of 0.91 m, but with a low computation time. The



(a) (b) (c) (d)

Figure 6: Raw input data sequence of case 1 and case 7, where (a and b) are of Case 1 and (c and d) are of case 7.

Table 3: Dataset 2 registration results of 4PCS, K4PCS, SUPER 4PCS, and F-4PCS algorithms.

	4PCS			K4PCS		
Avg.	t (s)	RMSE (m)	RMSE_{ICP} (m)	t (s)	RMSE (m)	RMSE_{ICP} (m)
	9.19	**0.91**	**0.63**	**339.43**	**0.59**	**0.42**
	Super 4PCS			F-4PCS (proposed)		
Avg.	t (s)	RMSE (m)	RMSE_{ICP} (m)	t (s)	RMSE (m)	RMSE_{ICP} (m)
	51.76	**0.71**	**0.68**	**16.87**	**0.55**	**0.47**

K-4PCS algorithm successfully registers 8 out of 10 registrations and sometimes fails for 4 and 7 cases, reaching the lowest RMSE of 0.59 m. However, the average computation time reaches up to 1554.33 s, which is inappropriate for actual use. The main cause of the large computation time lies in the keypoint extraction stage because this is the only difference from the 4PCS algorithm. We were unable to run five times for the K-4PCS algorithm since some registrations took more than two to three hours, so we only ran three times per each scan pair. The SUPER 4PCS algorithm mostly fails to align the sequences, registering only 1 out of the ten input cases. In addition, its average computation time, 51.76 s, is higher than the 4PCS and F-4PCS algorithms. F-4PCS successfully registers 8 out of 10 registrations except Cases 5 and 10 with a reasonable average computation time of 16.87 s, achieving the lowest RMSE of 0.55 m, as shown in Figure 7. We can see that the portable ladder and conveyor belts in the scan data do not match when the SUPER 4PCS algorithm is used because it only registers the major ground plane, while the F-4PCS algorithm successfully registers them.

3.3. Dataset 3. Dataset 3 is a construction site scene of an offshore plant, where various pipes and parts are installed. It is composed of one CAD model and ten raw point clouds, as shown in Figure 8. The CAD model is a 3D map data of the completed offshore plant, and the total number of point of which reaches about 80,000. The raw point clouds are taken from the actual scene of the offshore plant by a mobile device, Lenovo Phab 2 Pro for a few seconds. The raw point clouds may contain temporary supports, scaffolds, and wrapped components, and the total numbers of the points in each point cloud are only in the range of about twenty to thirty thousands. We aligned the raw point clouds to the CAD model, and raw point clouds were obtained around the pipe displayed in the dotted box from Figure 8(a). The parameter S is set to 200, which is higher than S of Dataset 1 and lower than S of Dataset 2 to take it into account that the CAD model is accurate, while the raw point clouds contain a high level of noise compared to Dataset 1.

Ten registrations are performed for this experiment, and the registration results are summarized in Table 4. Notice that the total number of the filtered CAD model is 5,122, and the average number of filtered points of source point cloud is 636. All the algorithms except F-4PCS fail to find any transformation matrices for all the data sequences, so the results of them are not included. This is because the point density of the source point sets is much lower than the target point cloud, and the dimension of the target point cloud is much larger than that of the source point cloud. Furthermore, the source point clouds contain temporary supports, scaffolds, and other unexpected parts that may not be included in the CAD model. Therefore, randomly generated

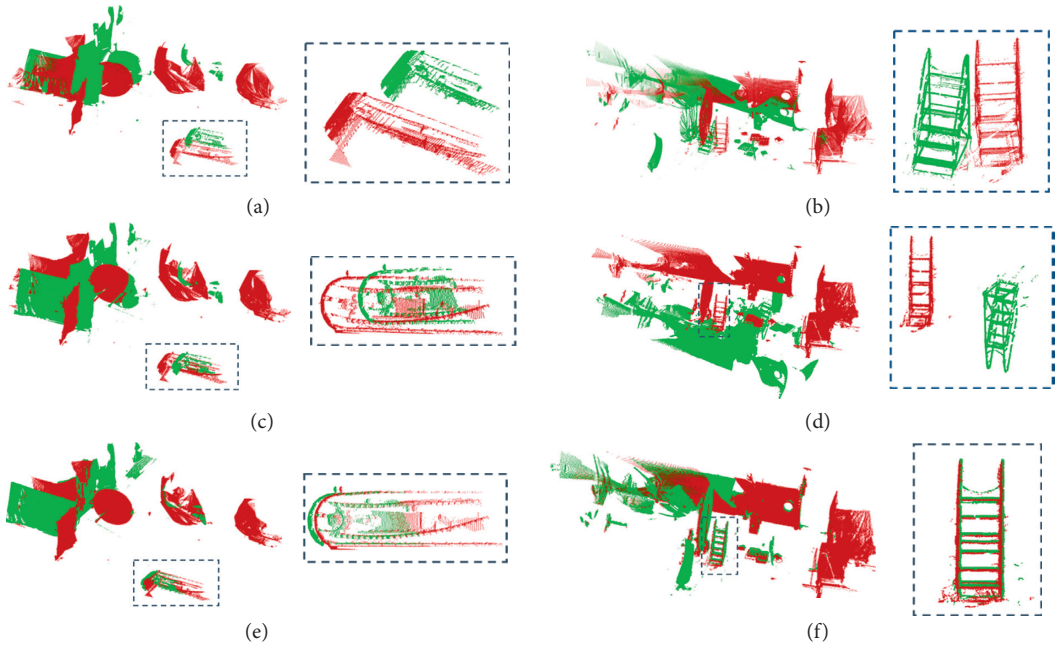


Figure 7: Input data sequences and initial registration results of case 5 (first column) and case 10 (second column). (a and b) are initial poses of Case 5 and Case 10 before registration, (c and d) are the results by the SUPER 4PCS, and (e and f) by the F-4PCS algorithms. The ground planes are removed for better visualization.

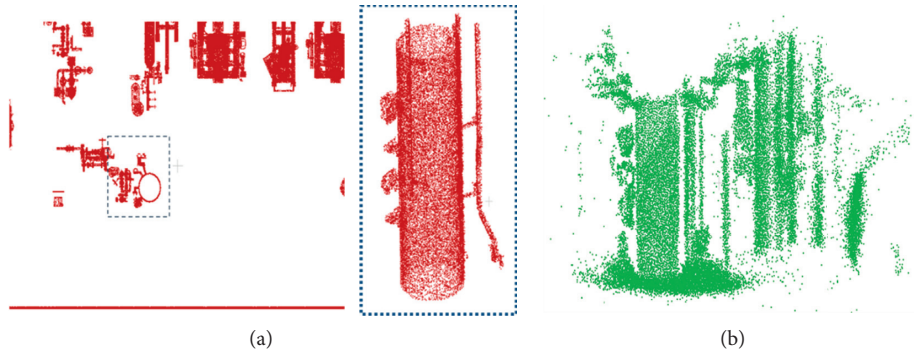


Figure 8: The top view of the raw CAD data (a) wherein the zoomed pipe is displayed in dotted rectangular box and raw point cloud acquired by a mobile device is displayed in (b). The ground plane of CAD data is removed for better visualization.

Table 4: Dataset 3 registration results of F-4PCS algorithms.

	t (s)	RMSE (m)	RMSE_{ICP} (m)
Avg	**4.89**	**0.19**	**0.18**

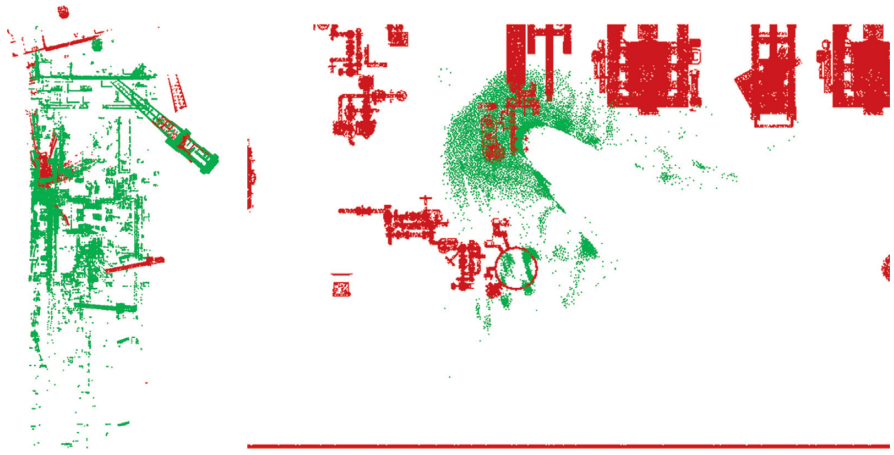
bases from the target point cloud will most likely fail to find congruent bases from the source point cloud, meaning any randomized scheme will not work for this type of datasets. On the contrary, F-4PCS algorithm successfully registers 8 out of 10 registrations as we can observe results in Figure 8, with the average RMSE of 0.18 m. Some of the registration results are shown in Figure 8. In addition, its computation is 4.89 s on average, which is appropriate for actual use. The initial registration results are further refined by the ICP algorithm to produce the RMSE of 0.18 m.

3.4. PFH Descriptor vs. Modified PFH Descriptor. In this section, we analyse the time and performance of the proposed PFH histogram that uses an additional feature f_4. For a fair comparison, all other parameters are set to be equal.

Table 5 presents the evaluation results of the PFH descriptor-based F-4PCS algorithm for all the datasets. In the experiment with Dataset 1, the computation times of the PFH and the modified PFH histograms are almost identical; however, the PFH histogram-based F-4PCS fails to register Case 9, while the modified PFH-based F-4PCS algorithm successfully registers the sequence. The average RMSE and RMSE_{ICP} of PFH-based F-4PCS are 2.42 m and 1.37 m, which are higher than other algorithms except K-4PCS algorithm. In the experiment with Dataset 2, the computation time is lower than the F-4PCS algorithm, and RMSE is slightly lower than the F-4PCS algorithm. However, the

Table 5: Registration results of PFH-based F-4PCS algorithms.

	Dataset 1		
Avg	t (s) **13.53**	RMSE (m) **2.42**	RMSE_{ICP} (m) **1.37**
	Dataset 2		
Avg	t (s) **10.97**	RMSE (m) **0.43**	RMSE_{ICP} (m) **0.38**
	Dataset 3		
Avg	t (s) **11.61**	RMSE (m) **0.18**	RMSE_{ICP} (m) **0.15**



(a) (b)

Figure 9: Registration failure cases for (a) Dataset 1 and (b) Dataset 3 of PFH-based F-4PCS algorithm.

performances of both algorithms are similar to each other since they both successfully register 8 out of 10 registrations for all the same data sequences. For Dataset 3, its average computation time is 11.61 s, which is higher than the proposed F-4PCS algorithm. Although the RMSE of PFH descriptor-based F-4PCS algorithm is similar to the RMSE of the proposed F-4PCS algorithm, the modified PFH-based F-4PCS algorithm actually successfully aligns only 4 out of 10 registrations while the proposed F-4PCS algorithm successfully registers 8 out of 10 registrations. In fact, the main cause of a low RMSE is that the moved source point clouds attach to other pipes or parts of the target CAD data. The failure registration cases of the PFH-based F-4PCS algorithm for Datasets 1 and 3 are shown in Figure 9.

In conclusion, the performance and computation time of the two versions of the F-4PCS algorithm are similar to each other in some datasets; however, the PFH-based F-4PCS algorithm fails in one more case with Dataset 1 and fails to align 6 out of 10 registrations with Dataset 3, thereby showing that the proposed feature descriptor is better than the PFH descriptor.

3.5. Data with Noise and Outliers. In this section, we tested the F-4PCS algorithm with data that includes noise or outliers. We used a hippo model in [20]. The input parameters are equal to the parameters used for the earlier datasets, except S that it is set to 100, and the voxel grid filter size is set to 2.0.

We apply a zero-mean additive random Gaussian noise with variance σ^2 to the input points, similar to that in the 4PCS algorithm tests discussed in [18]. The registration process is successful when $\sigma = 0.05$ and 0.1, as shown in Figure 10. However, sometimes registration fails when $\sigma \geq 0.1$, and the algorithm does not work at all when $\sigma > 0.2$. This is because the F-4PCS algorithm depends on the feature descriptors, which often fail to find correspondence information when the data is highly corrupted with significant noise.

In the experiments using data with outliers, random noise points are added to the original point clouds within their bounding boxes, and the number of outliers are set to ρ. The percentage of the original number of point clouds is exactly the same as that of the 4PCS algorithm experiments in [18]. We tested three cases when ρ are set to 10, 20, and 40%.

The F-4PCS algorithm successfully aligns the hippo models up to 20% without a failure. Even when it reaches up to 40%, which is the maximum percentage that the 4PCS algorithm used in the experiments, it rarely fails to align the point clouds, as shown in Figure 11. This is because the voxel grid filter effectively removes the outliers.

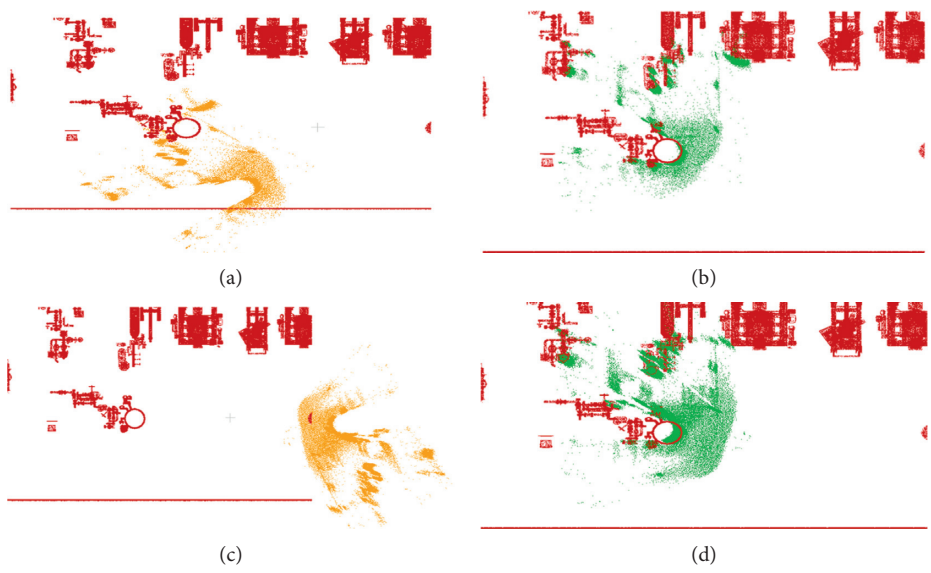
From these experiments, we found that the F-4PCS algorithm is less robust to Gaussian noise as compared to the

(a) (b) (c) (d)

Figure 10: F-4PCS algorithm initial registration results of the hippo models with Gaussian noise of different σ values for (a) original input, (b) $\sigma = 0.05$, (c) $\sigma = 0.1$, and (d) $\sigma = 0.2$.

(a) (b) (c)

Figure 11: F-4PCS algorithm initial registration results of the hippo models with outliers of different ρ values for (a) $\rho = 10\%$, (b) $\rho = 20\%$, and (c) $\rho = 40\%$.



(a) (b) (c) (d)

Figure 12: Initial registration results of the F-4PCS algorithm. (a and c) are initial poses of Cases 1 and 9 and (b and d) are initial registration results.

4PCS algorithm and is as robust as the 4PCS algorithm to outliers.

However, the two input datasets are prepared to have a large overlap area, as shown in Figure 12, which we barely encounter in practice. In many cases, we end up having point sets similar to Datasets 1, 2, and 3, with which the proposed algorithm works much better than the existing ones because the probability of actual datasets possessing outliers is much more than them possessing artificial Gaussian noise. Therefore, the F-4PCS algorithm can be used to align the actual 3D data obtained from a laser scanner in industrial environments.

4. Conclusions

In this study, we propose the F-4PCS algorithm for registering 3D scans used in industrial environments. This algorithm is an extended version of the 4PCS algorithm, which overcomes the drawbacks of the 4PCS algorithm and its variants. Furthermore, it is designed to improve the consistency of the registration results and reduce the computation time.

The proposed algorithm uses scanned points instead of extra markers for registration. Moreover, this method is a global registration scheme that does not require any

approximate initial registration or correspondence, and it automatically performs registration, which are two advantageous features for practical use.

However, this method yields only an approximate registration result, which should be followed by a refining registration method, such as ICP, to achieve high accuracy in registration. Furthermore, the computation time of the proposed registration method is another problem that hinders its real-time application. Thus, enhancing the accuracy and reducing the computation time are two topics that we recommend for future work.

Data Availability

Previously reported point cloud data in Figures 10 and 11 were used to support this study and are available at https://geometry.cs.ucl.ac.uk/projects/2014/super4PCS/. These prior studies and datasets are cited at relevant places within the text [20]. The other point cloud data used to support the findings of this study were supplied by Samsung Heavy Industries under license and so cannot be made freely available. Requests for access to these data can be made to the corresponding author.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Technology Innovation Program (or Industrial Strategic Technology Development Program) (20000208, Development of master data system of lead time for precision enhancement of shipbuilding production management) and funded by the Ministry of Trade, Industry & Energy (MOTIE), Korea.

References

[1] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[2] K. Lim, *Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration*, University of North Carolina, Chapel Hill, NC, USA, 2004.

[3] D. H. Yun, S. I. Choi, S. H. Kim, and K. H. Ko, "Registration of multiview point clouds for application to ship fabrication," *Graphical Models*, vol. 90, pp. 1–12, 2017.

[4] S. Du, G. Xu, S. Zhang, X. Zhang, Y. Gao, and B. Chen, "Robust rigid registration algorithm based on pointwise correspondence and correntropy," *Pattern Recognition Letters*, vol. 132, pp. 91–98, 2020.

[5] A. W. Fitzgibbon, "Robust registration of 2D and 3D point sets," *Image and Vision Computing*, vol. 21, no. 13, pp. 1145–1153, 2003.

[6] J. Yang, H. Li, and Y. Jia, "Go-icp: solving 3d registration efficiently and globally optimally," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1457–1464, Sydney, Australia, Decemeber 2013.

[7] Z. Wu, H. Chen, S. Du, M. Fu, N. Zhou, and N. Zheng, "Correntropy based scale ICP algorithm for robust point set registration," *Pattern Recognition*, vol. 93, pp. 14–24, 2019.

[8] D. Akca, *Full Automatic Registration of Laser Scanner Point Clouds*, ETH Zurich, Zürich, Switzerland, 2003.

[9] B. Becerik-Gerber, F. Jazizadeh, G. Kavulya, and G. Calis, "Assessment of target types and layouts in 3d laser scanning for registration accuracy," *Automation in Construction*, vol. 20, no. 5, pp. 649–658, 2011.

[10] M. Franaszek, G. S. Cheok, and C. Witzgall, "Fast automatic registration of range images from 3d imaging systems using sphere targets," *Automation in Construction*, vol. 18, no. 3, pp. 265–274, 2009.

[11] D. Yun, S. Kim, H. Heo, and K. H. Ko, "Automated registration of multi-view point clouds using sphere targets," *Advanced Engineering Informatics*, vol. 29, no. 4, pp. 930–939, 2015.

[12] M. Khoury, Q. Zhou, and V. Koltun, "Learning compact geometric features," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 153–161, Venice, Italy, October 2017.

[13] Z. Marton, D. Pangercic, N. Blodow, J. Kleinehellefort, and N. Beetz, "General 3D modelling of novel objects from a single view," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3700–3705, Taipei, Taiwan, October 2010.

[14] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3212–3217, Kobe, Japan, May 2009.

[15] R. B. Rusu, Z. C. Marton, and N. Blodow, "Persistent point feature histograms for 3d point clouds," in *Proceedings of the 10th International Conference on Intelligent Autonomous System (IAS-10)*, pp. 119–128, Baden, Germany, 2008.

[16] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3D object classification," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pp. 2987–2992, Phuket, Thailand, December 2011.

[17] J. Sun, K. Hiekata, H. Yamato, N. Nakagaki, and A. Sugawara, "Efficient point cloud data processing in shipbuilding: reformative component extraction method and registration method," *Journal of Computational Design and Engineering*, vol. 1, no. 3, pp. 202–212, 2014.

[18] D. Aiger, N. J. Mitra, and D. Cohen, "4-points congruent sets for robust surface registration," *ACM Transactions on Graphics*, vol. 27, no. 85, pp. 1–10, 2008.

[19] M. A. Fischler and R. C. Bolles, "Random sample consensus," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[20] N. Mellado, D. Aiger, and N. J. Mitra, "Super 4PCS fast global pointcloud registration via smart indexing," *Computer Graphics Forum*, vol. 33, no. 5, pp. 205–215, 2014.

[21] P. W. Theiler, J. D. Wegner, and K. Schindler, "Markerless point cloud registration with keypoint-based 4-points congruent sets," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, no. 2, pp. 283–288, 2013.

[22] X. Ge, "Automatic markerless registration of point clouds with semantic-keypoint-based 4-points congruent sets," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 130, pp. 344–357, 2017.

[23] R. B. Rusu and S. Cousins, "3D is here: point cloud library (PCL)," in *Proceedings of the IEEE International Conference on Robots and Automation*, pp. 1–4, Shanghai, China, May 2011.

[24] R. B. Rusu, *Semantic 3D object maps for everyday manipulation in human living environments*, Ph.D. thesis, Technische Universitaet Muenchen, Munich, Germany, 2009.
[25] E. W. Weisstein, "L2norm from mathworld-a wolfram web resource," 2020, http://mathworld.wolfram.com/L2-Norm.html.
[26] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, USA, 3rd edition, 1996.
[27] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Aligning pointcloud views using persistent feature histograms," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3384–3391, Nice, France, September 2009.
[28] Q. Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *Proceedings of the European Conference on Computer Vision*, pp. 766–782, Amsterdam, Netherlands, October 2016.
[29] N. Mellado, "Open GR: A C++ library for 3D global registration," 2017, https://storm-irit.github.io/OpenGR/.
[30] L. Dagum and R. Menon, "OpenMP: an industry standard api for shared-memory programming," *IEEE Computational Science and Engineering*, vol. 5, no. 1, pp. 46–55, 1998.