

A point cloud registration algorithm based on normal vector and particle swarm optimization

Measurement and Control

1–11

© The Author(s) 2019


Article reuse guidelines:

sagepub.com/journals-permissions

DOI: 10.1177/0020294019858217

journals.sagepub.com/home/mac



Xu Zhan^{1,5}, Yong Cai¹, Heng Li³, Yangmin Li⁴ and Ping He^{2,3} 

Abstract

Based on normal vector and particle swarm optimization (NVP), a point cloud registration algorithm is proposed by searching the corresponding points. It provides a new method for point cloud registration using feature point registration. First, in order to find the nearest eight neighbor nodes, the k -d tree is employed to build the relationship between points. Then, the normal vector and the distance between the point and the center gravity of eight neighbor points can be calculated. Second, the particle swarm optimization is used to search the corresponding points. There are two conditions to terminate the search in particle swarm optimization: one is that the normal vector of node in the original point cloud is the most similar to that in the target point cloud, and the other is that the distance between the point and the center gravity of eight neighbor points of node is the most similar to that in the target point cloud. Third, after obtaining the corresponding points, they are tested by random sample consensus in order to obtain the right corresponding points. Fourth, the right corresponding points are registered by the quaternion method. The experiments demonstrate that this algorithm is effective. Even in the case of point cloud data lost, it also has high registration accuracy.

Keywords

K -d tree, normal vector, quaternion method

Date received: 20 April 2019; accepted: 28 May 2019

Introduction

In recent years, signal analysis and image processing have been booming. In particular, laser scanning technology has made great progress in obtaining three-dimensional (3D) point cloud model of physical objects and made point cloud become the mainstream data in 3D processing. High-precision scanning equipment can effectively obtain the detailed data of the physical objects. Due to the limitations of measuring instruments and environment, the data cannot be completed in one laser scan. Point cloud data should be obtained by scanning separately from different perspectives, and then point cloud data from multiple perspectives should be registered to obtain a complete point cloud model of physical objects. Therefore, point cloud registration is widely used in 3D reconstruction, reverse engineering, target recognition and other fields, and it is an important part in point cloud data processing. The accuracy of point cloud registration directly affects the quality of subsequent processing technology.¹

The existing point cloud registration methods can be divided into two types: one is the accurate positioning of the navigation system in the scanning process and

the other is the accurate alignment of point clouds from different perspectives. The algorithm in this paper belongs to the second type.

The most classical algorithms in the automatic registration of point cloud model are iterative closest point (ICP) and its improved algorithms. They are methods based on point-to-point or point-to-surface search technology, and point cloud registration is completed by

¹School of Information Engineering, Southwest University of Science and Technology, Mianyang, People's Republic of China

²School of Intelligent Systems Science and Engineering, Jinan University, Zhuhai, People's Republic of China

³Department of Building and Real Estate, The Hong Kong Polytechnic University, Kowloon, Hong Kong

⁴Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong

⁵Department of Automation, Sichuan University of Science & Engineering, Yibin, People's Republic of China

Corresponding author:

Ping He, Department of Industrial Engineering, School of Intelligent Systems Science and Engineering, Jinan University, No. 206, Qianshan Road, Xiangzhou District, Zhuhai 519070, Guangdong, People's Republic of China.

Email: pinghecn@qq.com; p.he@polyu.edu.hk



Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License

(<http://www.creativecommons.org/licenses/by/4.0/>) which permits any use, reproduction and distribution of the work without

further permission provided the original work is attributed as specified on the SAGE and Open Access pages (<https://us.sagepub.com/en-us/nam/open-access-at-sage>).

minimizing the distance between point clouds. ICP is easy to implement, but it requires that there is a constraint between two point clouds, and the positions of the two point clouds are relatively close. The algorithm results depend on the initial position of the point cloud, which is easy to cause the problem of rapid convergence to local optimal. Therefore, there are a lot of works to do to improve the ICP algorithm.²⁻⁴

Non-feature registration

One of the improvements in the ICP algorithm is non-feature registration. The non-feature registration does not need to extract the feature data. It aims at improving registration efficiency. Many scholars have proposed non-feature registration algorithm. Some have studied the key points or key areas how to detect to reduce the mismatch points,⁵⁻⁷ some have studied the corresponding points how to search and some speed up the operation of ICP.⁸

Feature registration

The other improvement in the ICP algorithm is to feature registration. They usually study the rough registration first and then study the non-rough registration. Rough registration is a registration method without any initial position information at all. The main purpose of this method is to quickly estimate a general point cloud registration matrix with unknown initial conditions. The whole calculation process requires a high calculation speed, and the accuracy of the registration is not too high, such as the method based on local feature description, based on global search strategy, through statistical probability and so on.

The method based on local feature description is to extract the neighborhood geometric features of source and target point clouds. The corresponding points between them can be quickly determined by geometric features to achieve registration.

The representative algorithm based on global search strategy is sampling consistency algorithm. The algorithm randomly selects corresponding points with the same geometric features between source and target point clouds and obtains the optimal solution by calculating the transformation relationship of corresponding points.

Statistical probability method based on normal distribution is that according to the normal distribution of point cloud, the corresponding points are determined so as to calculate the transformation relationship between target and source point clouds.

In above strategy, the method based on feature point matching is popular. There are a lot of research achievements in feature registration.^{9,10} The method which was proposed by Kase et al.¹¹ is to extract effective feature correspondence points, determine the initial position of two point clouds and then use ICP algorithm to non-rough registration. However, feature

points are often defined as points with large-scale surface gradient. A single feature point often has similar geometric features, so using a single feature point to rough registration ignores the local structural information of feature points, which is prone to mismatching and affects the precision of non-rough registration.

In order to solve the problem of the single feature point, a point cloud registration algorithm based on normal vector and particle swarm optimization (NVP) is proposed. This method is the continuation of the previous article.¹² The previous article mainly focused on the noise of input point cloud and the accuracy of registration. The particle swarm optimization (PSO) in the previous article was used to search the rotation matrix. It was with randomness. In order to solve the randomness, the method of finding the corresponding points is proposed in this paper and then the registration is achieved by quaternion method. The main contributions are shown as shown follows:

- In order to extract multiple feature data, the k -d tree is employed to find the nearest eight neighbor nodes. Then, the normal vector of the node and the distance between the point and the center gravity of the eight neighbor nodes are defined. The feature data will not change by the geometric transformation; they are regarded as the feature data of the node.
- In order to search the corresponding points with the same feature data between the two point clouds, PSO is used. The fitness function is that the normal vector of the node in the original point cloud is the most similar to that in the target point cloud and the distance between the point and the center gravity of eight points of the node is the most similar to that in the target point cloud.
- After obtaining the corresponding points, they are tested by random sample consensus (RANSAC) and then the registration is employed by quaternion method.
- The experiments demonstrate that this algorithm is effective. Even in the case of point cloud data lost, it also has high registration accuracy.

Related work and results

Definition of NVP

We propose NVP to find the corresponding points to achieve registration. The main process is demonstrated in Figure 1. First, it builds k -d tree in the point cloud to search the nearest eight neighbor nodes and uses the neighbor nodes to calculate the normal vector of the point node and the distance between the point and the center gravity of the eight neighbor nodes, which are as two feature data of each node. Second, PSO is used to search the corresponding points. The fitness function is that the normal vector of the node in the original point

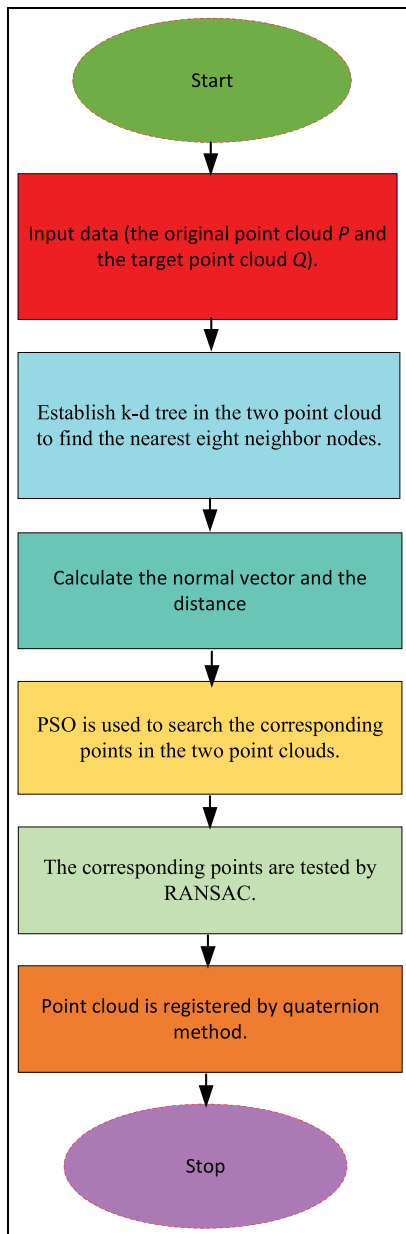


Figure 1. The process of NVP.

cloud is the most similar to that in the target and the distance between the point and the center gravity of eight points of the node is the most similar to that in the target point cloud. After obtaining the corresponding points, they are tested by RANSAC in order to obtain the right corresponding points. Finally, point cloud registration is realized by quaternion method.

K-d tree

K-d tree is short for *k*-dimensional tree. It is a kind of data structure that divides *k*-dimensional data. It is mainly used in the key data search of multidimensional space, such as range search and nearest neighbor search.^{13,14} *K-d tree* is used by scale-invariant feature transform (SIFT) when matching feature points.

Feature point matching is actually a problem of similarity retrieval between high-dimensional vectors by distance function. How to find the nearest neighbor of the query point quickly and accurately is a hot issue. Many scholars have proposed their algorithms. *K-d tree* is one of the algorithms which is applied widely. There are two ways to search the similarity in an index structure: one is the range search and the other is the nearest neighbor search. The range search is finding all the data in the data set whose distance from the query point is less than the threshold value. The nearest neighbor search is finding *k* data from the data set closest to the query point. If *k* is set to 1, it is the nearest neighbor search.

There are two types of feature matching algorithms. One is the linear scanning method. The points in the data set are compared with the query points one by one. It does not establish any data structures and the search efficiency is very low. The other is to establish data index and then to do the feature match quickly. Because the actual data generally present a cluster-like clustering form, the retrieval speed can be greatly accelerated by the effective index structure. *K-d tree* belongs to the second type. The main idea is to divide the search space into different levels.

The search space is to find the nodes in axis-aligned rectangle by performing the following three steps: the first step is to check whether point in node lies in given rectangle, the second is to recursively search left/bottom (if any could fall in rectangle) and the third is to recursively search right/top (if any could fall in rectangle).

K-d tree is used in this paper to establish the data structure in point cloud to search the nearest eight neighbor nodes. After finding the nearest eight neighbor nodes, the normal vector of the point node, the distance between the point and the center gravity of the eight neighbor nodes can be calculated.

The normal vector

One of the essential features of the node in point cloud is the normal vector. The accurate and high-quality point-based drawing methods not only depend on the normal vectors, but the precise reconstruction results also need the precise normal vectors. The reconstruction algorithm especially needs normal vector aggregation, such as multi-level partition of unity (MPU), implicit surface reconstruction algorithm and detection and recovery of sharp features.

The point cloud normal vector calculation methods can be divided into three types. They are methods based on local surface fitting, Delaunay or Voronoi and robust statistics. The point curvature calculation in this paper belongs to the first type. The method was first proposed by Hoppe¹⁵ when he studied the surface reconstruction algorithm based on signed distance function. Assume that the plane of the point cloud is smooth everywhere. The steps of calculating curvature are shown as follows.

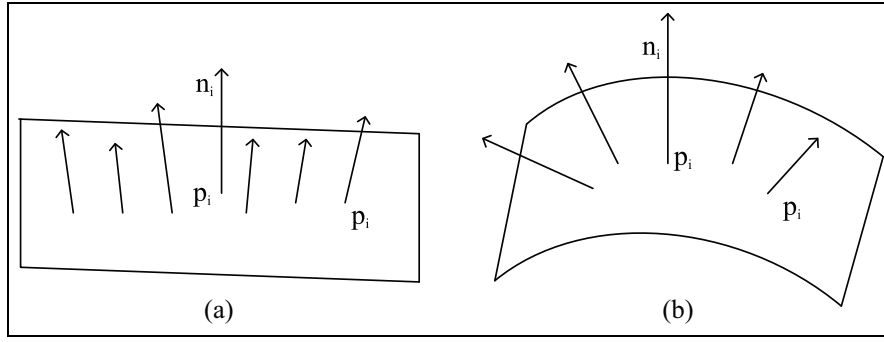


Figure 2. Normal vectors from different surfaces: (a) the flat surface and (b) the non-flat surface.

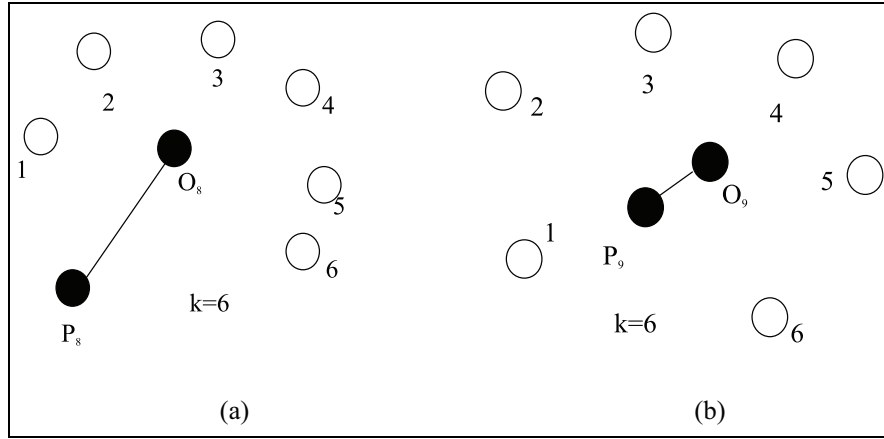


Figure 3. Different distances from the point to its neighbor center of gravity: (a) boundary point and its neighbor gravity center and (b) interior point and its neighbor gravity center.

Suppose $P = \{p_i(x_i, y_i, z_i) | i = 1, 2, 3, \dots, N\}$ is the input point cloud data and $\{p_{ij}(x_{ij}, y_{ij}, z_{ij}) | j = 1, 2, 3, \dots, k\}$ is the k -nearest neighbors of point cloud p_i . Then, the neighbor center of gravity is defined as follows

$$O_i = \frac{1}{k} \sum_{j=1}^k p_{ij} \quad (1)$$

The covariance matrix of p_i is calculated as follows

$$T_i = \begin{bmatrix} p_{i1} - O_i \\ p_{i2} - O_i \\ \dots \\ p_{ik} - O_i \end{bmatrix}^T \begin{bmatrix} p_{i1} - O_i \\ p_{i2} - O_i \\ \dots \\ p_{ik} - O_i \end{bmatrix} \quad (2)$$

where T_i is a semi-positive symmetric matrix with geometric information of surface and solved by Jacobian method. The three eigenvalues ($\lambda_1, \lambda_2, \lambda_3$) and their feature vectors (n_1, n_2, n_3) are obtained. If $\lambda_3 \geq \lambda_2 \geq \lambda_1$, the point curvature C_i and the surface normal vector n_i are defined as follows

$$C_i = \frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3}, n_i = n_1 i^1 \quad (3)$$

By definition, it is shown that greater the change of surface is, the larger the point curvature is (see Figure 2).

Distance between the point and its neighbor center of gravity

Another feature of the node in point cloud is the distance between the point and its neighbor center of gravity. It is defined as follows (see Figure 3)

$$D_{li} = |p_i - O_i| \quad (4)$$

PSO

After obtaining the two feature data of the node, we need to find corresponding points with the same feature data between the two point clouds. Therefore, we resort to PSO.¹⁶ PSO and genetic algorithm (GA) are widely used in many fields.¹⁷⁻¹⁹ PSO is a kind of evolutionary algorithm and is similar to the simulated annealing algorithm. It starts from the random solution and seeks the optimal solution through iteration. It evaluates the quality of the solution through fitness and is simpler than GA. Crossover and mutation operation are not needed in PSO. This algorithm has attracted academic

attention because of its advantages such as easy implementation, high accuracy and fast convergence. The formula of PSO is as follows

$$V_{iD}^{k+1} = V_{iD}^k + c_1 r_1 (p_{pD}^k - X_{iD}^k) + c_2 r_2 (p_{gD}^k - X_{iD}^k) \quad (5)$$

$$X_{iD}^{k+1} = X_{iD}^k + V_{iD}^{k+1} \quad (6)$$

where D is the number of dimension, X is the current position of particle i , V is the velocity of particle i , p_p is the best position of the particle i in one iteration, p_g is the best position for all particles, c_1, c_2 are accelerators or learning factors, which are set to 2. r_1 and r_2 are pseudo random numbers, which are set to (0–1).

The particle of PSO in this paper starts by calculating the two feature data (the point curvature, the distance from the point to its neighbor center of gravity). The particle from the target point cloud moves in the search space by setting to zero or to small random values. If the search criterion meets, the particles' velocities are no more updated. The search criterion is that the normal vector of the node in the original point cloud is the most similar to that in the target point cloud, and the distance between the point and the center gravity of eight neighbor points of the node is the most similar to that in the target point cloud. The main flow is shown in Figure 4.

RANSAC

There are, usually, outlier data among the corresponding points. Therefore, the RANSAC is employed in searching the right corresponding points. The algorithm was first proposed by Fischler and Bolles²⁰ in 1981. RANSAC is based on the following assumptions:

- Data are composed of reasonable points, for example, the distribution of data can be explained by some model parameters.
- Outlier data do not fit the model.
- Other data are noise.

There are several reasons to generate the outlier data, such as the extreme value of noise, wrong measurement method and incorrect assumptions about data. RANSAC is used in searching the right corresponding points in this paper; the main steps are shown in Algorithm 1.

Quaternion method

After obtaining the right corresponding points, point cloud registration is realized by quaternion method. It was proposed by Horn²¹ in 1987. He used a quaternion-based least square method to solve the motion parameters between adjacent point clouds. ICP, which is the most widely used in point cloud registration, adopts this method too. The algorithm flow is as follows.

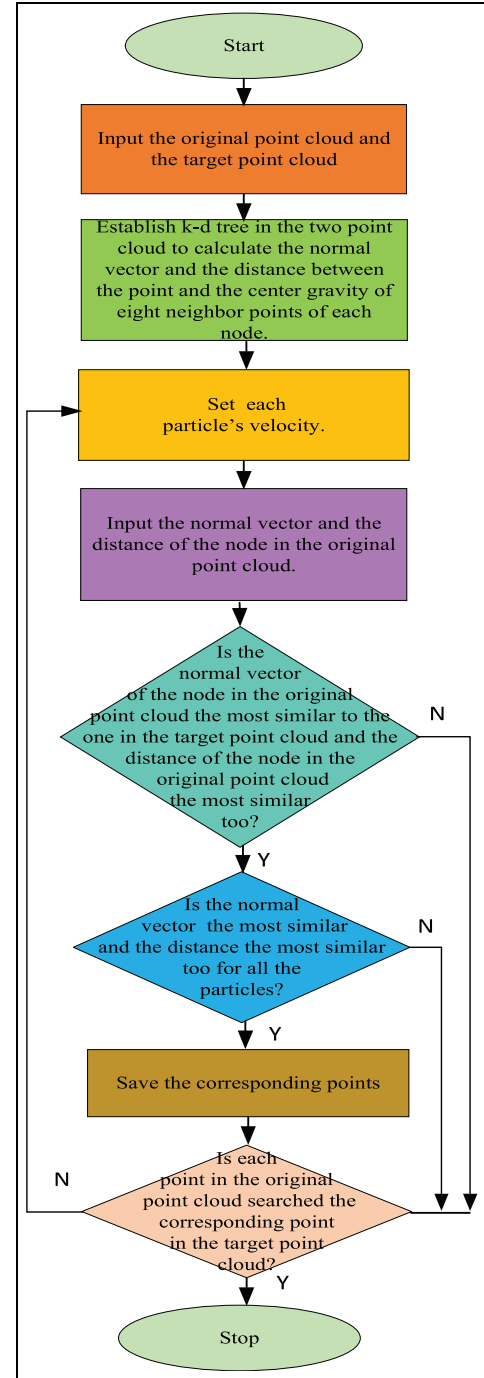


Figure 4. The PSO flow diagram.

Suppose $P = \{p_i(x_i, y_i, z_i) | i = 1, 2, 3, \dots, N\}$ is the original point cloud, and $Q = \{q_j(x_j, y_j, z_j) | j = 1, 2, 3, \dots, N\}$ is the target point cloud.

The center gravity of the two point clouds is defined as follows

$$\begin{cases} G_p = \frac{1}{N} \sum_{i=1}^N p_i \\ G_q = \frac{1}{N} \sum_{j=1}^N q_j \end{cases} \quad (7)$$

Algorithm 1. The steps of RANSAC.

Input: The corresponding points $(s_1^1, s_1^2), (s_2^1, s_2^2), \dots, (s_n^1, s_n^2)$

Output: The right corresponding points

1: Set the number M of iterations

2: **for** each $j \in [1, M]$ **do**

3: Set the number $N_j = 0$ of iterations

4: Randomly select three pairs of points as a sample

5: Proposing the three pairs of points are right corresponding points, calculate the transformation matrix T

6: Set the threshold value $\varepsilon = 2 \times \frac{1}{n} \sum_{i=1}^n \sqrt{(T(s_i^1) - s_i^2)^2}$

7: Calculate the distance between other $n - 3$ corresponding points $d_{jj} = \sqrt{T((s_{jj}^1) - s_{jj}^2)^2}$, $jj = 1, 2, \dots, n - 3$

8: If $d_{jj} < \varepsilon$, the corresponding point (s_{jj}^1, s_{jj}^2) are the right corresponding point, save them and count the correct number N_j of corresponding points. Otherwise, they are outlier data.

9: **end for**

10: Find the right corresponding points whose number N from N_j , $j = 1, 2, \dots, M$ is the maximum.

Gravity-centralize the two point clouds.

$$\begin{cases} P' = P - G_p \\ Q' = Q - G_q \end{cases} \quad (8)$$

Construct the covariance matrix.

$$M = \frac{1}{N} \sum_{i=1}^N [P'Q'^T] = \frac{1}{N} \sum_{i=1}^N [PQ^T] - G_p G_q^T \quad (9)$$

Construct the symmetric matrix by the covariance matrix

$$P_q = \begin{bmatrix} M_{11} + M_{22} + M_{33} & M_{23} - M_{32} & M_{31} - M_{13} & M_{12} - M_{21} \\ M_{23} - M_{32} & M_{11} - M_{22} - M_{33} & M_{12} + M_{21} & M_{13} + M_{31} \\ M_{31} - M_{13} & M_{12} + M_{21} & M_{22} - M_{11} - M_{33} & M_{23} + M_{32} \\ M_{12} - M_{21} & M_{13} + M_{31} & M_{23} + M_{32} & M_{33} - M_{11} - M_{22} \end{bmatrix} \quad (10)$$

- Find and calculate the maximum eigenvalue of P_q ; the corresponding eigenvectors can form quaternions, which is defined as follows

$$q_R = [q_0 \quad q_1 \quad q_2 \quad q_3] \quad (11)$$

where $q_0 \geq 0$ and $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$.

- Construct the rotation matrix R by the quaternions

$$R = \begin{cases} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2 \times (q_1 \times q_2 - q_0 \times q_3) & 2 \times (q_1 \times q_3 + q_0 \times q_2) \\ 2 \times (q_1 \times q_2 + q_0 \times q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2 \times (q_2 \times q_3 - q_0 \times q_1) \\ 2 \times (q_1 \times q_3 - q_0 \times q_2) & 2 \times (q_2 \times q_3 + q_0 \times q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{cases} \quad (12)$$

- Calculate the translation matrix T

$$T = G_p - R G_q \quad (13)$$

The experimental simulation

The performance of this algorithm is made in MATLAB and the data sets are used from Stanford's

experimental database <http://graphics.stanford.edu/data/3Dscanrep/>.

The registration accuracy test

Two models ("cow," "feet" of "man") are adopted in this experiment (see Figure 5): the red one is the original point cloud and the blue one is the target point cloud.

In order to further test the performance of NVP, the mean square error (MSE) is used in the experiment. The formula is as follows

$$MSE = \frac{1}{n} \sum_{k=1}^n \min \left[(x_{ok} - x_{tk})^2 + (y_{ok} - y_{tk})^2 + (z_{ok} - z_{tk})^2 \right] \quad (14)$$

where MSE is the mean square error, x_{oi}, y_{oi}, z_{oi} is the source point cloud and x_{ti}, y_{ti}, z_{ti} is the target. n is the minimum size in the two point clouds. The experiment steps are shown in Algorithm 2.

After obtaining the corresponding points, they are tested by RANSAC (see section "RANSAC") and then point cloud is registered by quaternion method (see section "Quaternion method").

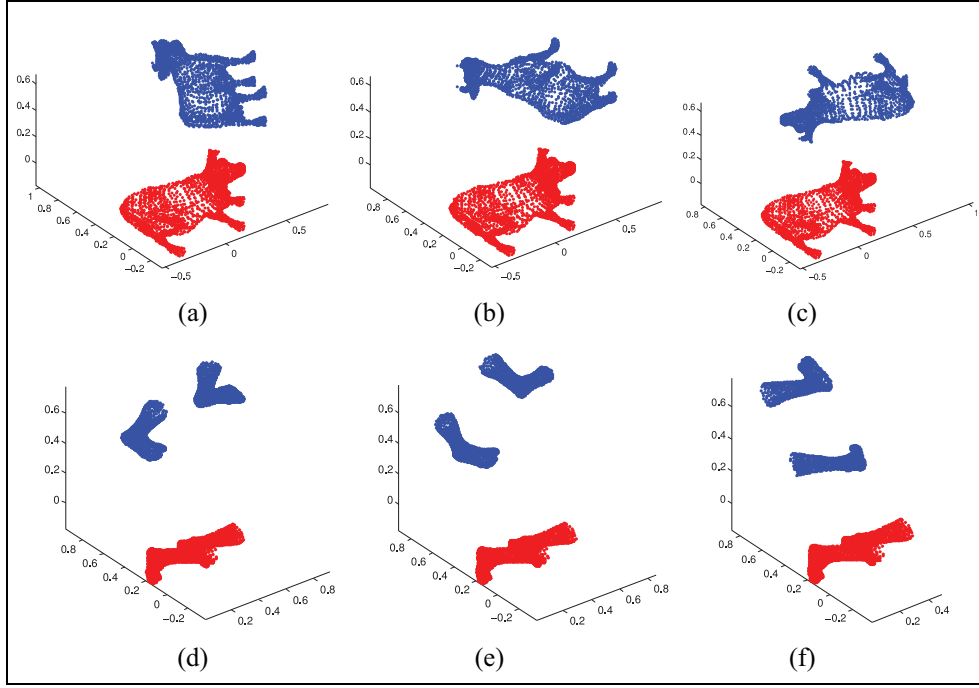


Figure 5. Data sets: (a) Cowview1, (b) Cowview2, (c) Cowview3, (d) Feetview1, (e) Feetview2 and (f) Feetview3.

Algorithm 2. The experiment steps.

Input: The original point cloud P

The target point cloud Q

Output: The corresponding points

- 1: Establish k -d tree in two point clouds
- 2: Find the nearest eight neighbor nodes of each points
- 3: Calculate the normal vector and the distance from the point to the center gravity of eight neighbor points of each point
- 4: Obtain the number X of the original point cloud
- 5: Set the number $Y = X$ of iterations
- 6: **for** each $j \in [1, Y]$ **do**
- 7: Input the normal vector and the distance of the point p_j from the original point cloud, which needs to find the corresponding point in the target point cloud
- 8: Set the number M of iterations, the number N of randomly searching the points in the target point cloud
- 9: Generate randomly N index numbers $ld_i, i = 1, 2, \dots, N$
- 10: Get the points $pp_{ld_i}, i = 1, 2, \dots, N$ of the corresponding index numbers
- 11: Set $c_1 = 2, c_2 = 2$
- 12: Obtain the normal vectors and the distances of the points $pp_{ld_i}, i = 1, 2, \dots, N$ in the target point cloud, which need to compare with the input point
- 13: Compare the normal vectors of the points $pp_{ld_i}, i = 1, 2, \dots, N$ in target point cloud with the one of the input points and find the closest normal vector, save the index number NV_j
- 14: Compare the distances of the points $pp_{ld_i}, i = 1, 2, \dots, N$ in target point cloud with the one of the input points and find the closest distance, save the index number DI_j
- 15: If NV_j is equal to DI_j , the corresponding point is found and set $ld_p = NV_j, ld_g = NV_j$. Otherwise, $ld_p = 0, ld_g = 0$, where ld_p is the particles' best position in one iteration, ld_g is the best position for all particles
- 16: **for** each $jj \in [1, M]$ **do**
- 17: set $r_{1i} = rand(), r_{2i} = rand(), i = 1, 2, \dots, N$, where r_{1i}, r_{2i} are pseudo random numbers, $rand()$ is the random function, which generates a random number (0 – 1) in the uniform distribution.
- 18: $ld_i = ld_i + (ld_p - ld_i) \times c_1 \times r_{1i} + (ld_g - ld_i) \times c_2 \times r_{2i}, i = 1, 2, \dots, N$.
- 19: Get the points $pp_{ld_i}, i = 1, 2, \dots, N$ of the corresponding index numbers.
- 20: Obtain the normal vectors and the distances of the points $pp_{ld_i}, i = 1, 2, \dots, N$ in the target point cloud, which need to compare with the input point.
- 21: Compare the normal vectors of the points $pp_{ld_i}, i = 1, 2, \dots, N$ in target point cloud with the one of the input points and find the closest normal vector, save the index number NV_{jj} .
- 22: Compare the distances of the points $pp_{ld_i}, i = 1, 2, \dots, N$ in target point cloud with the one of the input points and find the closest distance, save the index number DI_{jj} .
- 23: If NV_{jj} is equal to DI_{jj} , the corresponding point is find and set $ld_p = NV_{jj}$.
- 24: Compare ld_p and ld_g . If ld_p is the best, set $ld_g = ld_p$.
- 25: **end for**
- 26: **end for**

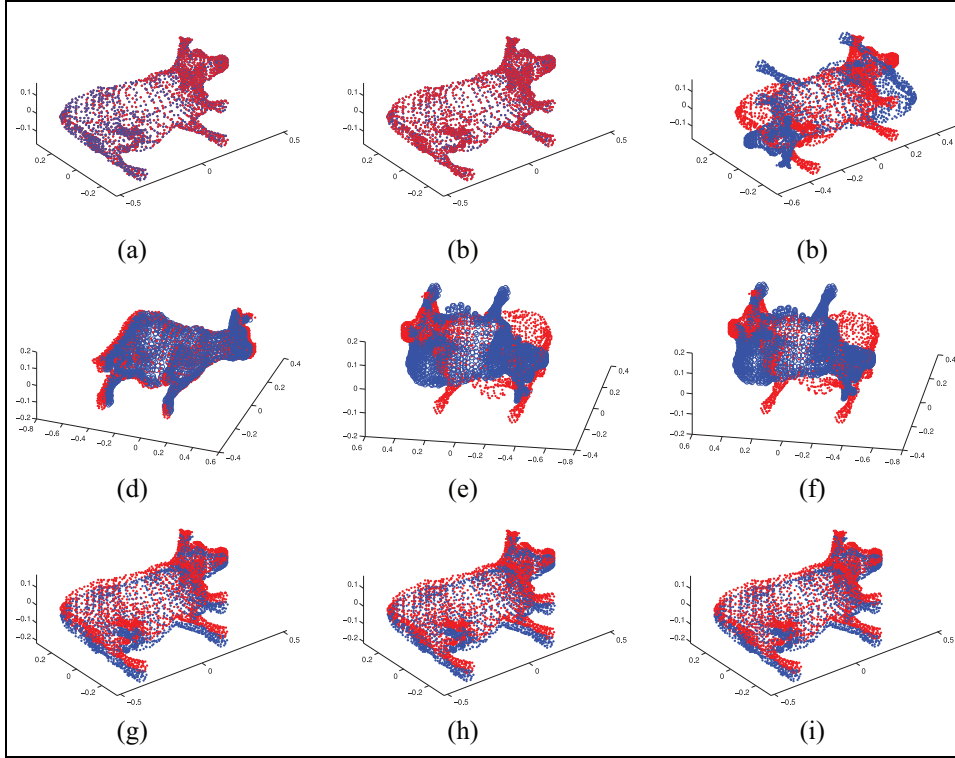


Figure 6. Point cloud registration without data lost: (a) Cow1 (Scale-ICP), (b) Cow2 (Scale-ICP), (c) Cow3 (Scale-ICP), (d) Cowview1 (CPD), (e) Cowview2, (f) Cowview3 (CPD), (g) Cowview1 (NVP), (h) Cowview2 (NVP) and (i) Cowview3 (NVP).

Table 1. Point cloud registration without data lost.

Name	Size	MSE (Scale-ICP)	MSE (CPD)	MSE (NVP)
Cow-view1	2904×3	2.83×10^{-31}	9.21×10^{-5}	1.60×10^{-2}
Cow-view2	2904×3	4.25×10^{-31}	1.30×10^{-3}	1.46×10^{-2}
Cow-view3	2904×3	1.50×10^{-3}	1.30×10^{-3}	1.43×10^{-2}
Feet-view1	2000×3	6.79×10^{-4}	2.93×10^{-5}	3.78×10^{-16}
Feet-view2	2000×3	3.14×10^{-4}	2.93×10^{-5}	4.83×10^{-16}
Feet-view3	2000×3	6.79×10^{-4}	1.00×10^{-3}	5.09×10^{-16}

MSE: mean square error; ICP: Iterative Closest Point; CPD: Coherent Point Drift.

Point cloud registration without data lost. The point cloud registration without data lost is that the size of original point cloud is the same as the target. The experiment is made by comparing NVP with the descriptor Scale-ICP²² and CPD (Coherent Point Drift).²³

The results of the experimental data set are shown in Figures 6 and 7, and the size of the experimental data sets and the MSE are shown in Table 1. It can be seen from the results that the Scale-ICP registration accuracy depends on the initial position, but the accuracy is very high if the good initial position is provided. The CPD does not depend on the initial position, but it is sometimes not workable. The description which we proposed can work well and sometimes show high registration accuracy.

Point cloud registration with data lost. The point cloud registration with data lost is that the size of original point

cloud is not same as the target. A random data loss experiment is tested by comparing NVP with the descriptor Scale-ICP²² and CPD.²³ The results of the experimental data set are shown in Figures 8 and 9 and Table 2. It can be seen from the results that the Scale-ICP registration and CPD are not workable, but the NVP still works well when facing such challenging case.

Conclusion

Point cloud registration algorithm based on normal vector and particle swarm optimization is proposed in this paper. It provides a new method for point cloud registration using feature point registration. First, it is given a method on how to extract the feature data of each point in the point cloud, and the feature data are used to search the corresponding points in two point clouds in PSO. Second, the corresponding points are

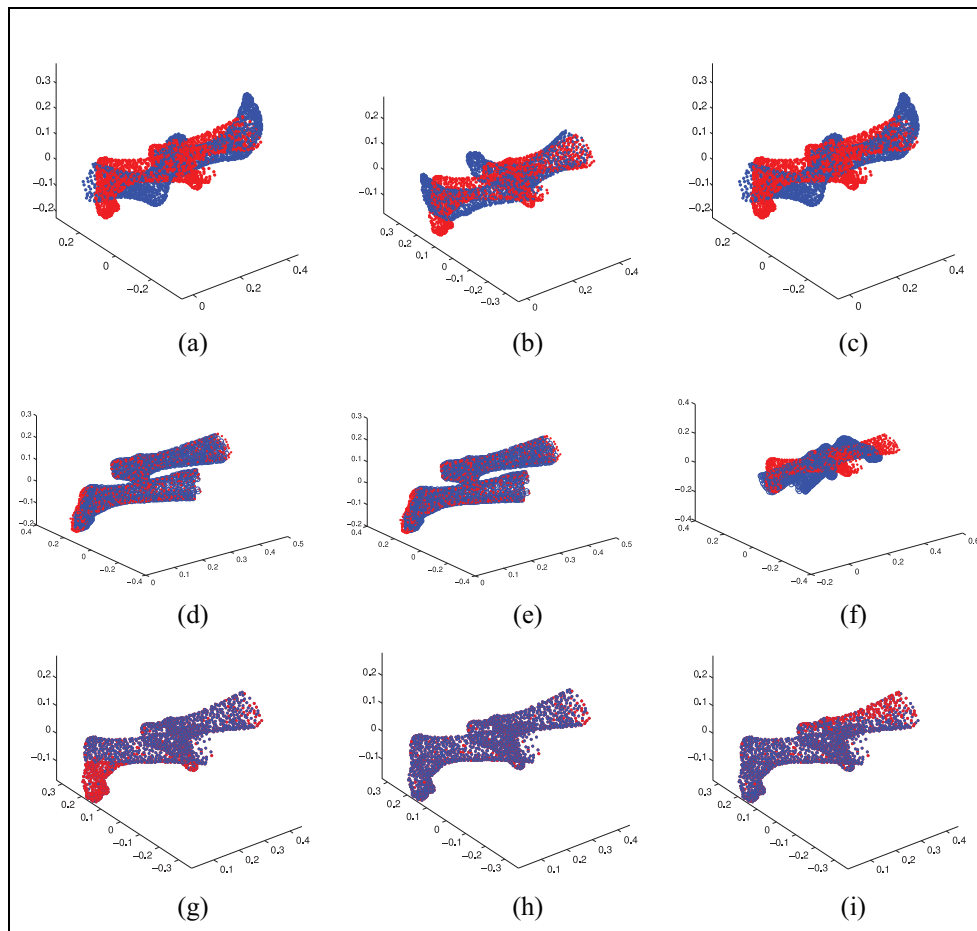


Figure 7. Point cloud registration without data lost: (a) Feet1 (Scale-ICP), (b) Feet2 (Scale-ICP), (c) Feet3 (Scale-ICP), (d) Feetview1 (CPD), (e) Feetview2 (CPD), (f) Feetview3 (CPD), (g) Feetview1 (NVP), (h) Feetview2 (NVP) and (i) Feetview3 (NVP).

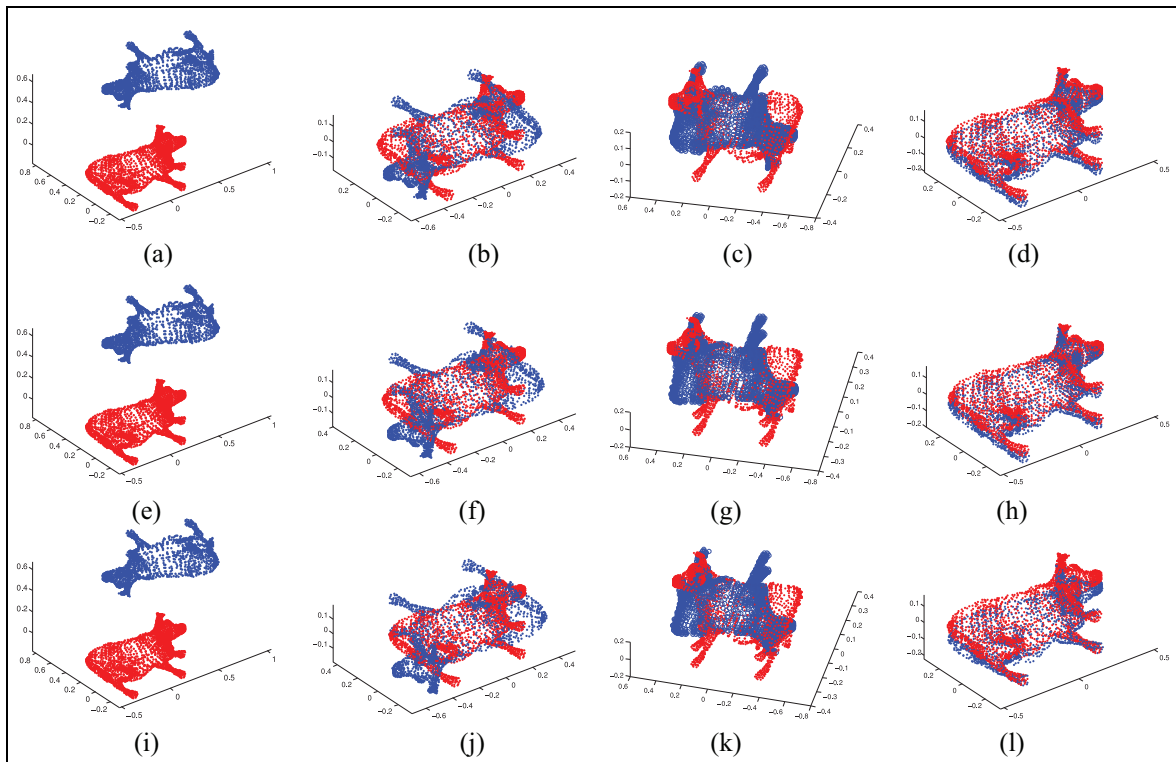


Figure 8. Point cloud registration with data lost: (a) Cow (data lost 5%), (b) Result (Scale-ICP), (c) Result (CPD), (d) Result (NVP), (e) Cow (data lost 15%), (f) Result (Scale-ICP), (g) Result (CPD), (h) Result (NVP), (i) Cow (data lost 25%), (j) Result (Scale-ICP), (k) Result (CPD) and (l) Result (NVP).

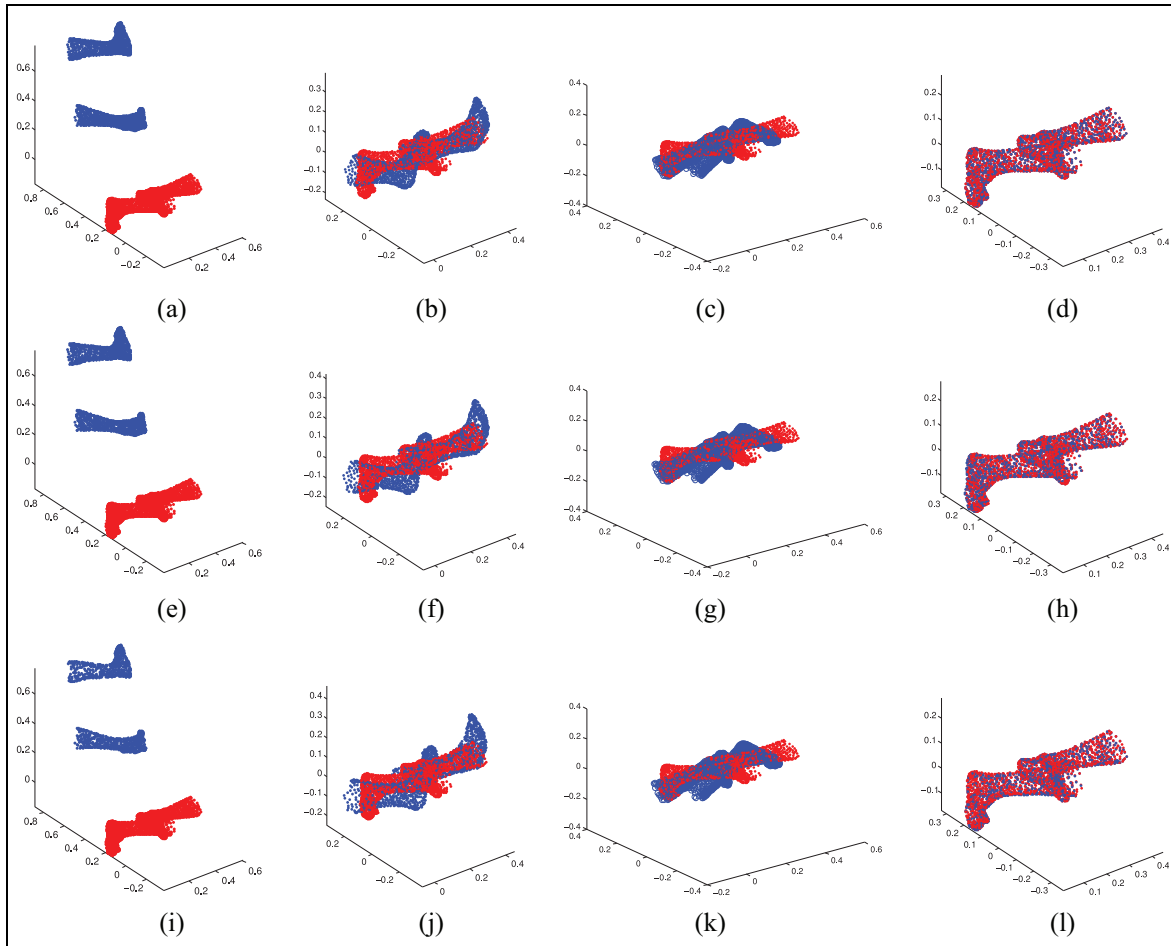


Figure 9. Point cloud registration with data lost: (a) Feet (data lost 5%), (b) Result (Scale-ICP), (c) Result (CPD), (d) Result (NVP), (e) Feet (data lost 15%), (f) Result (Scale-ICP), (g) Result (CPD), (h) Result (NVP), (i) Feet (data lost 25%), (j) Result (Scale-ICP), (k) Result (CPD) and (l) Result (NVP).

Table 2. Point cloud registration with data lost.

Name	Size	MSE (Scale-ICP)	MSE (CPD)	MSE (NVP)
Cow (data lost 5%)	2904×3	1.70×10^{-3}	1.30×10^{-3}	1.36×10^{-2}
Cow (data lost 15%)	2904×3	2.20×10^{-3}	1.30×10^{-3}	1.17×10^{-2}
Cow (data lost 25%)	2904×3	3.30×10^{-3}	1.40×10^{-3}	1.85×10^{-2}
Feet (data lost 5%)	2000×3	7.90×10^{-4}	1.00×10^{-3}	7.66×10^{-4}
Feet (data lost 15%)	2000×3	1.20×10^{-3}	1.00×10^{-3}	1.2×10^{-3}
Feet (data lost 25%)	2000×3	2.00×10^{-3}	1.00×10^{-3}	9.48×10^{-4}

MSE: mean square error; ICP: Iterative Closest Point; CPD: Coherent Point Drift.

tested by RANSAC to obtain right corresponding points. After receiving the right corresponding points, the original point cloud is transformed to the target by the quaternion method. The experiments demonstrate that this algorithm is effective. Even if the data are lost, it still works well.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship and/or publication of this article: This work was jointly supported by National Natural Science Foundation of China (grant no. 11705122), Department of Building and Real Estate of The Hong Kong Polytechnic University, the General Research Fund (grant no. BRE/PolyU 152099/18E) entitled “Proactive Monitoring of Work-Related MSD Risk Factors and Fall Risks of Construction Workers Using Wearable Insoles” and Natural Science Foundation of The Hong Kong Polytechnic University (grant no. G-YW3X).

ORCID iD

Ping He  <https://orcid.org/0000-0001-7340-9606>

References

1. Tam GKL, Cheng ZQ, Lai YK, et al. Registration of 3D point clouds and meshes: a survey from rigid to nonrigid. *IEEE T Vis Comput Gr* 2013;19(7): 1199–1217.
2. Rusinkiewicz S and Levoy M. Efficient variants of the ICP algorithm. In: *Proceedings of the third international conference on 3-D digital imaging and modeling*, Quebec City, QC, Canada, 28 May–1 June 2001, pp. 145–152. New York: IEEE.
3. Rodrigues M, Fisher R and Liu Y. Special issue on registration and fusion of range images. *Comput Vis Image Und* 2002; 87(1–3): 1–7.
4. Liu Y. A mean field annealing approach to accurate free form shape matching. *Pattern Recogn* 2007; 40(9): 2418–2436.
5. Zaharescu A, Boyer E and Horaud R. Keypoints and local descriptors of scalar functions on 2D manifolds. *Int J Comput Vision* 2012; 100(1): 78–98.
6. Bueno M, Bosche F, Gonzalez-Jorge H, et al. 4-plane congruent sets for automatic registration of as-is 3D point clouds with 3D BIM models. *Automat Constr* 2018; 89(2): 120–134.
7. Hidaka N, Michikawa T, Motamedi A, et al. Polygonization of point clouds of repetitive components in civil infrastructure based on geometric similarities research article. *Automat Constr* 2018; 86(1): 99–117.
8. Jost T and Hugli H. A multi-resolution ICP with heuristic closest point search for fast and robust 3D registration of range images. In: *Proceedings of the fourth international conference on 3-D digital imaging and modeling*, Quebec City, QC, Canada, 6–10 October 2003, pp. 427–433. New York: IEEE.
9. Rusu BR, Blodow N and Beetz M. Fast point feature histograms (FPFH) for 3D registration. In: *Proceedings of the IEEE international conference on robotics and automation*, Kobe City, Japan, 12–17 May 2009, pp. 3212–3217. New York: IEEE.
10. Rusu BR, Blodow N, Marton CZ, et al. Aligning point cloud views using persistent feature histograms. In: *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, Nice City, 22–26 September 2008, pp. 3384–3391. New York: IEEE.
11. Kase K, Makinouchi A, Nakagawa T, et al. Shape error evaluation method of free-form surfaces. *Comput Aided Design* 1999; 31(12): 495–505.
12. Zhan X, Cai Y and He P. A three-dimensional point cloud registration based on entropy and particle swarm optimization. *Adv Mech Eng*. Epub ahead of print 1 December 2018. DOI: 10.1177/1687814018814330.
13. Aung SS, Nagayama I and Tamaki S. A high-performance classifier from K-dimensional tree-based dual-KNN. *IEIE Trans Smart Proc Comput* 2018; 7(3): 184–194.
14. Jennings E and Madigan M. astroABC: an approximate Bayesian computation sequential Monte Carlo sampler for cosmological parameter estimation. *Astron Comput* 2017; 19(1): 16–22.
15. Hoppe H, DeRose T, Duchamp T, et al. Surface reconstruction from unorganized points. *Comp Graph* 1992; 26(2): 71–78.
16. Clerc M. *Particle swarm optimization*, vol. 93. Hoboken, NJ: John Wiley & Sons, 2010.
17. Castelli M, Cattaneo G, Manzoni L, et al. A distance between populations for n-points crossover in genetic algorithms. *Swarm Evol Comput* 2019; 44: 636–645.
18. Ruberto S, Vanneschi L and Castelli M. Genetic programming with semantic equivalence classes. *Swarm Evol Comput* 2018; 102(3): 1–17.
19. Hajek P, Henriques R, Castelli M, et al. Forecasting performance of regional innovation systems using semantic-based genetic programming with local search optimizer. *Comput Oper Res* 2019; 106: 179–190.
20. Fischler MA and Bolles RC. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun ACM* 1981; 24(6): 381–395.
21. Horn BKP. Closed-form solution of absolute orientation using unit quaternions. *J Opt Soc Am A* 1987; 4(4): 629–642.
22. Ying S, Peng J, Du S, et al. A scale stretch method based on ICP for 3D data registration. *IEEE T Autom Sci Eng* 2009; 6(3): 559–565.
23. Myronenko A and Song X. Point set registration: coherent point drift. *IEEE T Pattern Anal* 2010; 32(12): 2262–2275.