Research Article

Advances in Mechanical Engineering

Advances in Mechanical Engineering
2018, Vol. 10(12) 1–13
© The Author(s) 2018
DOI: 10.1177/1687814018814330
journals.sagepub.com/home/ade
SAGE

A three-dimensional point cloud registration based on entropy and particle swarm optimization

Xu Zhan[1,2], Yong Cai[1] and Ping He[2,3,4]

Abstract

A three-dimensional (3D) point cloud registration based on entropy and particle swarm algorithm (EPSA) is proposed in the paper. The algorithm can effectively suppress noise and improve registration accuracy. Firstly, in order to find the k-nearest neighbor of point, the relationship of points is established by k-d tree. The noise is suppressed by the mean of neighbor points. Secondly, the gravity center of two point clouds is calculated to find the translation matrix *T*. Thirdly, the rotation matrix *R* is gotten through particle swarm optimization (PSO). While performing the PSO, the entropy information is selected as the fitness function. Lastly, the experiment results are presented. They demonstrate that the algorithm is valuable and robust. It can effectively improve the accuracy of rigid registration.

Keywords
k-d tree, entropy, robust

Date received: 24 May 2018; accepted: 26 October 2018

Handling Editor: David R Salgado

Introduction

3D point cloud registration is widely applied in computer vision, computer graphics and so on. It is always a research hotspot in the computer field. In recent years, information science and technology have developed rapidly.[1–11] The field represented by artificial intelligence has achieved fruitful scientific research results.[12,13] Numerous descriptions have been proposed from rigid registration to nonrigid registration.[14] Rigid registration is a challenging work. It is because there is noise and unwanted points in the data and the original positions affect algorithm performance. Therefore, there is still much work to perform, such as removing the noise[15–17] and improving the registration accuracy and so on.

Rigid registration

For the original point cloud P and the target one Q, where there are a lot of overlapping between the two, the rigid registration is to find the rotation matrix R and the translation matrix T to transform the original point cloud to the target one. The equation is defined as

[1] School of Information Engineering, Southwest University of Science and Technology, Mianyang, People's Republic of China
[2] Cooperative Control & System Optimization Laboratory, Department of Automation, School of Automation and Information Engineering, Sichuan University of Science & Engineering, Yibin, People's Republic of China
[3] Smart Construction Laboratory, The Hong Kong Polytechnic University, Kowloon, Hong Kong
[4] Emerging Technologies Institute, Faculty of Engineering, The University of Hong Kong, Hong Kong

Corresponding author:
Ping He, Cooperative Control & System Optimization Laboratory, Department of Automation, School of Automation and Information Engineering, Sichuan University of Science & Engineering, Room 437, Block A8, No. 188, University Town, Lingang Economic-Technological Development Area, Yibin, Sichuan 644000, People's Republic of China.
Email: pinghecn@qq.com; pinghe@hku.hk

$$q_i = R \times p_i + T \tag{1}$$

where q_i and p_i are the corresponding points, R is the rotation matrix, and T is the translation matrix. For all kinds of rigid registration methods, most of them build registration algorithm models through various constraints.

Transform constraint

The nearest point method can be solved by establishing constraint potential consistency correspondence. It is to complete the final registration by selecting the nearest point as consistent correspondence point. The iterative closest point (ICP) is solved by establishing the following constraint condition

$$E(R, T) = \min \sum_i \|q_i - (R \times p_i + T)\|^2 \tag{2}$$

Due to the good performance of ICP, many scholars have proposed many improved algorithms to improve the computation speed and robustness.[18–20]

Feature constraint

The geometric properties of point cloud, such as curvature and normal vectors, remain unchanged in rigid transformation.[21,22] Vary features can form an eigenvector. For the higher the eigenvector dimension, the lower the probability of all the features which can be matched, the high-dimensional eigenvectors can simplify data. Many surveys propose registration methods of feature constraints.[23–25] However, such method needs that the features of point cloud are obvious and easy to be extracted, and more time will be spent in the process of feature extraction. Kase et al.[26] proposed the extended Gaussian curvature, and a matching rate equation is used to determine the difference between the corresponding point sets. The extended Gaussian curvature is defined as

$$\varepsilon = |(k_1(p_i) - k_1(q_i)) \cdot (k_2(p_i) - k_2(q_i))| \tag{3}$$

where q_i and p_i are the corresponding points, and k_1 and k_2 are the main curvatures. The data can be greatly simplified and the registration efficiency is improved by extracting features.

Significance constraint

The significant areas are different from the surrounding areas. Significance can be used to measure local information to detect the key points or key areas.[27–29] The significance is usually used to reduce the potential mismatch points. The significance methods are usually geometric scale spatial analysis, significant scale and curvature based on vision, multi-scale sliding, maximum stability region extremum, and so on.

Regularization constraint

Regularization is constrained by adding penalty terms to the target function. Regularization constraint contains prior information, which avoids the occurrence of local minimum value and improves the search efficiency. Gold and Rangarajan[30] proposed that the rigid registration is regarded as a continuous optimization problem and dealt with problem between rigid transformation and consistency correspondence. The regularization term based on entropy is defined as

$$-\sum_i \sum_j M_{ij} \log M_{ij} \tag{4}$$

where M is a consistency correspondence matrix. According to the definition of entropy, when all points achieve registration, the entropy reaches the maximum.

Search constraint

Search constraint is mainly aimed at improving registration efficiency, including localization method and hierarchical search method. Jost and Hugli[31] proposed a method that speeds up the iteration of ICP with the above coarse to fine search technology and is refined gradually to obtain a more reliable consistency correspondence.

Contribution of this article

From the descriptions above, it is shown that 3D point cloud registration is the fundamental in computer field and still has many problems to further study and discussion. Our survey aims to solve the two rigid problems: (1) getting rid of the noise and (2) improving the accuracy. The challenge work in our article is how to improve the accuracy using the search constraint. The contributions of the article are introduced in the following:

- The k-d tree is used to build the relationship in point cloud to find the k-nearest neighbor (KNN) of the point, where the mean filter is employed to optimize the neighbor of the point. The noise of the point cloud is removed.
- The method of calculating the entropy of the point cloud is introduced in this article. It is used as the fitness function in particle swarm optimization (PSO) to search the best R. The results of the experiment show that our method effectively improved the accuracy.

Problem formulation

Point cloud registration is used to seek consistent correspondence between different datasets and to transform the different coordinate systems into the same coordinate system to gather the full data of object. Due to the limitation of 3D scanning technology, different datasets are usually obtained from different observation points. Each observation point is in a different coordinate system, so point cloud registration is an important aspect in 3D data acquisition. Point cloud registration is divided into rigid registration and nonrigid registration. Our method is the rigid registration and starts with formula (1) to find R and T. How to efficiently find R and T and improve the accuracy are the main problems solved by us. However, how to remove the noise is also a challenge work which we have to face.

Related work and main results

Definition of entropy and particle swarm algorithm

We propose entropy and particle swarm algorithm (EPSA) to achieve rigid registration. The process of our algorithm is shown in Figure 1. First, the original and target point clouds are the input data. Then, the EPSA descriptor uses k-d tree to find the KNN of the point, where the mean filter is used to get rid of the noise. Second, the EPSA descriptor calculates the center of two point clouds to find the translation matrix T; the information entropy is defined to calculate the entropy of the two point clouds; and PSO, which is the search constraint, is employed to search the rotation matrix R at random. The stopping criterion in PSO is that the entropy of the original point cloud is the closest to the target one. Once R and T are found, we can achieve the registration.

k-d tree

k-d tree is also named as k-dimensional tree. It is a high-dimensional data structure and applied in searching the KNN such as the KNN matching in high-dimensional image feature vectors in the image retrieval, that is, optimization of KNN algorithm. The k-d tree is a high-dimensional binary search tree. Unlike the common binary search tree, the tree stores the k-dimensional data.

In the searching process, the k-dimensional data need to decide from which dimension the data are selected. Then, the data are compared with the root node in the selected dimension. How to select the dimension and ensure the number of nodes in the left subtrees is as equal as possible to the right one after selecting dimension are two challenge works to face. In order to select the dimension, the variance in each dimension is calculated and the dimension of the largest variance is selected; for the larger the variance, the more the data disperse. The right and left subtrees are divided according to the pivot value to ensure the number of nodes in the left subtrees is as equal as possible to the right one. Table 1 shows the data structures in each node of the k-d tree.

From Table 1, it can be seen that building k-d tree is a hierarchical recursive process. The pseudo-code is shown as follows:

```
Name: CreateKDTree.
Input: Data_set.
Output: K_d tree.
Step 1. If Data_set is empty, returns an empty K_d tree.
Step 2. Call the node generating function.
    (a) Split is determined: the data variance in each dimension is calculated. Then, the corresponding dimension in which data variance is the maximum is the value of Split. The large data variance indicates that the data along the coordinate axis are scattered widely, and the data segmentation in this direction has a better resolution.
    (b) Node_data is determined: the data_set is sorted by Split. The data in the middle is selected as Node_ data, Data_set' = Data_set\Node_ data, where it means getting riding of Node_data in Data_set.
Step 3. $dataleft = (d \in Data\_set(\ )) \&\& (d[split] \leqslant Node\_data[split])$.
Step 4. $Left\_Range = Range \&\& dataleft$.
Step 5. $dataright = (d \in Data\_set(\ )) \&\& (d[split] > Node\_data[split])$.
Step 6. $Right\_Range = Range \&\& dataright$.
Step 7. $left = CreateKDTree(dataleft,Left\_Rang)$, the parent of the left is set to K_d tree.
Step 8. $right = CreateKDTree(dataright,Right\_Rang)$, the parent of the right is set to K_d tree.
```

In this article, k-d tree is used to establish the relationship between points to find the KNN of the point. Once the KNN of the point is found, we use mean filter to get rid of the noise. For KNNs of point $p_{i1}, p_{i2}, \ldots, p_{ik}, k = 1, 2, \ldots,$ of p_i, the value q_i by mean filtering is defined as

$$q_i = \frac{p_{i1} + p_{i2} + \cdots + p_{ik}}{k} \tag{5}$$

Therefore, $p_i = q_i$.

Information entropy

The definition of information entropy is proposed by Shannon.[32] It is applied in calculating the average amount of the information. The information associated with the probability of data. The lower the probability,

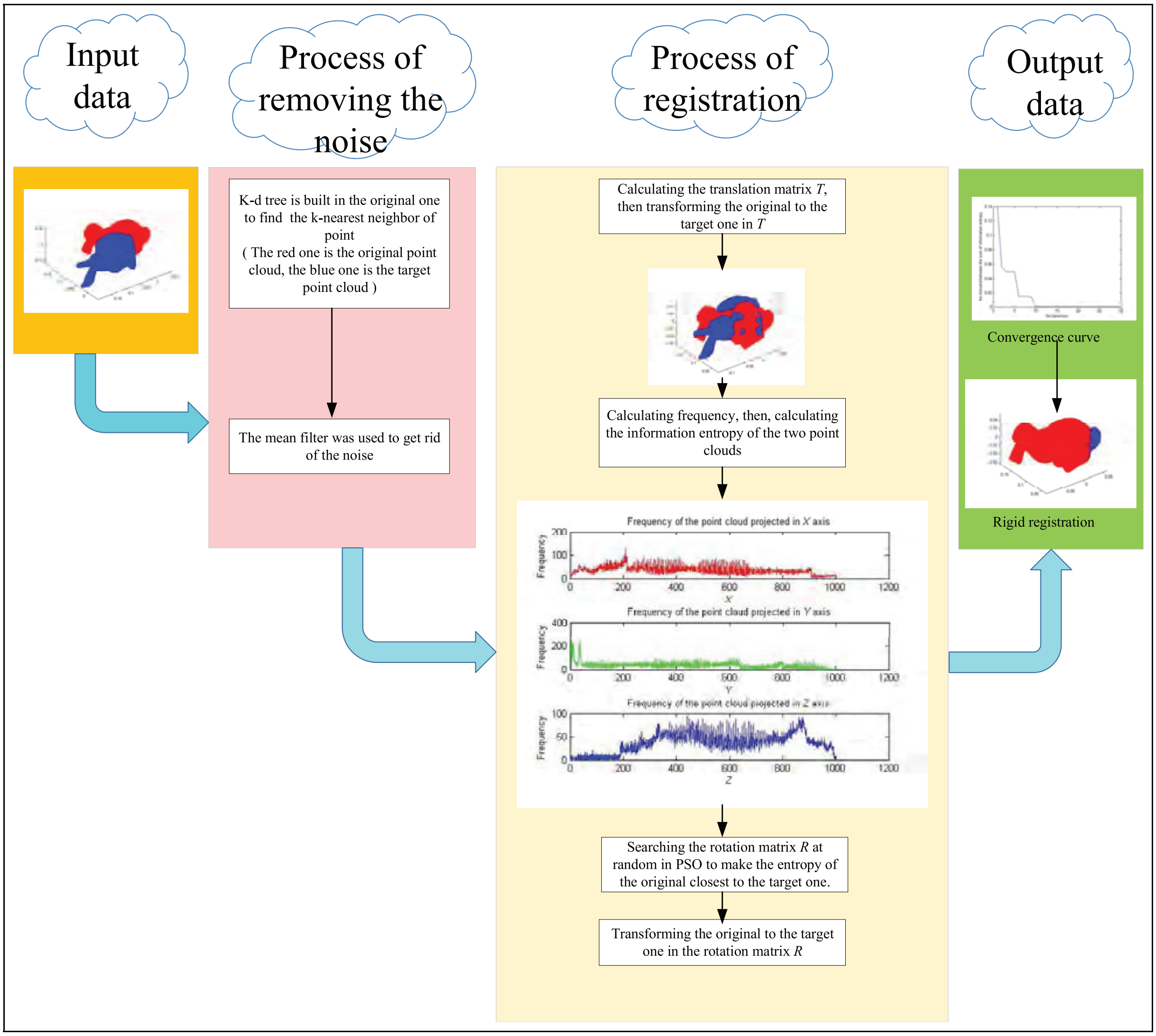
Input data
Process of removing the noise
Process of registration
Output data
K-d tree is built in the original one to find the k-nearest neighbor of point
(The red one is the original point cloud, the blue one is the target point cloud)
The mean filter was used to get rid of the noise
Calculating the translation matrix T, then transforming the original to the target one in T
Calculating frequency, then, calculating the information entropy of the two point clouds
Frequency of the point cloud projected in X axis
Frequency
200
100
0
0
200
400
600
800
1000
1200
X
Frequency of the point cloud projected in Y axis
Frequency
400
200
0
Y
Frequency of the point cloud projected in Z axis
Frequency
100
50
0
Z
Searching the rotation matrix R at random in PSO to make the entropy of the original closest to the target one.
Transforming the original to the target one in the rotation matrix R
Convergence curve
Rigid registration

Figure 1. Illustration of EPSA.

Table 1. Data structures of k-d tree.

Name	Data type	Description
Node data	Data vector	The node in the dataset is a *k*-dimensional vector
Range	Space vector	Space range of the node
Split	Integer	The directional axis number perpendicular to the segmentation hyperplane
Left	K_d tree	The k-d tree consisted of all data points in the left subspace of the hyperplane segmented by the node
Right	K_d tree	The k-d tree consisted of all data points in the right subspace of the hyperplane segmented by the node
Parent	K_d tree	Parent node

the more is the information which the event carries. The amount of the information conveyed by each event depends on the random variable's value. The value is the information entropy.

In this article, the information entropy is developed to calculate the entropy of point cloud. First, we need to project the point cloud to the X, Y, Z axis, respectively. The information entropy of X, Y, Z is the average amount of information. The formula is defined in the following

$$\begin{aligned} H(x) &= -\sum p_x \times \log(p_x) \\ H(y) &= -\sum p_y \times \log(p_y) \\ H(z) &= -\sum p_z \times \log(p_z) \end{aligned} \tag{6}$$

where p_x is the probability of the 3D point clouds which are projected to X-axis, and it is related to frequency. We need to standardize data in order to calculate the frequency. The standardizing formula is defined in the following

$$\begin{aligned} S(x_i) &= \frac{x_i - \min(x)}{\max(x) - \min(x)} \times 1000 \bmod 1000 \\ S(y_i) &= \frac{y_i - \min(y)}{\max(y) - \min(y)} \times 1000 \bmod 1000 \\ S(z_i) &= \frac{z_i - \min(z)}{\max(z) - \min(z)} \times 1000 \bmod 1000 \end{aligned} \tag{7}$$

where $S(x_i)$ is the standardized value within a region (0–1000) in X-axis when the point cloud is projected to the X-axis. max() is a function to find the maximum, and min() is a function to find the minimum. Similarly, we have $S(y_i)$ and $S(z_i)$. The frequency is obtained (See Figure 2), and it is to count the number of times which the value appears between 0 and 1000. Then, p_x is obtained, which is related to frequency and calculated as

$$\begin{aligned} p_x &= \sum_{i=0}^{1000} \frac{n_{xi}}{N} \\ p_y &= \sum_{i=0}^{1000} \frac{n_{yi}}{N} \\ p_z &= \sum_{i=0}^{1000} \frac{n_{zi}}{N} \end{aligned} \tag{8}$$

where n_{xi}, n_{yi}, n_{zi} is the number of point cloud which is projected to the X, Y, Z axis and N is the total number of point clouds. The same procedure may be easily adapted to get $H(y)$ and $H(z)$. H_{sum} is defined as the sum of information entropy when the 3D point clouds are projected to the X, Y, Z axis

$$H_{\text{sum}} = H(x) + H(y) + H(z) \tag{9}$$

PSO

To find the rotation angle to generate point-to-point correspondence, we study the optimization algorithm[33–35] and we research PSO. PSO is a populate algorithm to solve the optimization problems.[36] The particles move in the whole search space. A candidate solution is represented by a particle. According to the rules, each particle searches the better position. The basic formula of PSO is defined in the following

$$\begin{aligned} V_{iD}^{k+1} = V_{iD}^{k} + c_1 r_1 (p_{iD}^{k} - X_{iD}^{k}) \\ + c_2 r_2 (p_{gD}^{k} - X_{iD}^{k}) \end{aligned} \tag{10}$$

$$X_{iD}^{k+1} = X_{iD}^{k} + V_{iD}^{k+1} \tag{11}$$

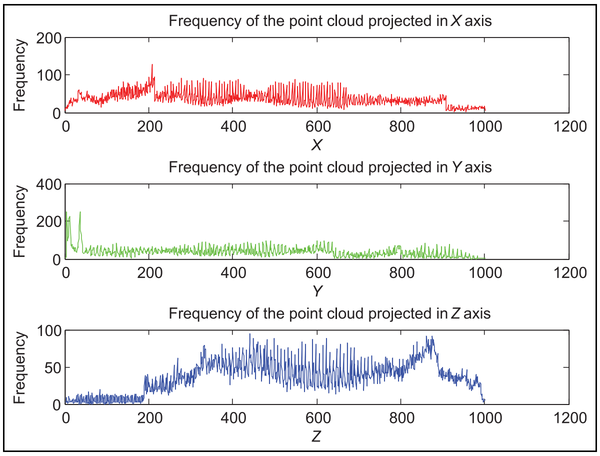

Figure 2. The frequency.

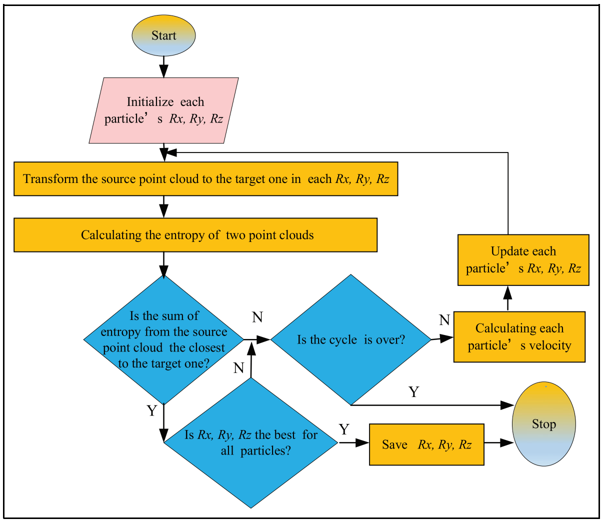
Start
Initialize each particle' s Rx, Ry, Rz
Transform the source point cloud to the target one in each Rx, Ry, Rz
Calculating the entropy of two point clouds
Is the sum of entropy from the source point cloud the closest to the target one?
N
Is the cycle is over?
N
Calculating each particle' s velocity
Update each particle' s Rx, Ry, Rz
Y
N
Y
Is Rx, Ry, Rz the best for all particles?
Y
Save Rx, Ry, Rz
Stop

Figure 3. Flow diagram of PSO.

where D is the spatial dimension; X is the particle i's position; V is the particle i's velocity; p_i is the particle i's best position in history; p_g is the best position for all particles; c_1, c_2 are the learning factors or accelerators, which are usually set to 2; and r_1, r_2 are the pseudo-random numbers with a region (0–1).

The particle of PSO in this article starts by generating random rotation angle within an initialization region (0°–360°). The particle's velocity is usually set to zero or to small random values in order to make the particle move in the search space during the first iterations. Once the stopping criterion is met, the velocities and rotation angles of the particles are no more updated in the algorithm. The stopping criterion in the algorithm is that the H_{sum} from the source point cloud by rotating is closest to the H_{sum} from the target one. The flow diagram is shown in Figure 3, where R_x is a rotation angle around the X-axis. Similarly, we have R_y and R_z.

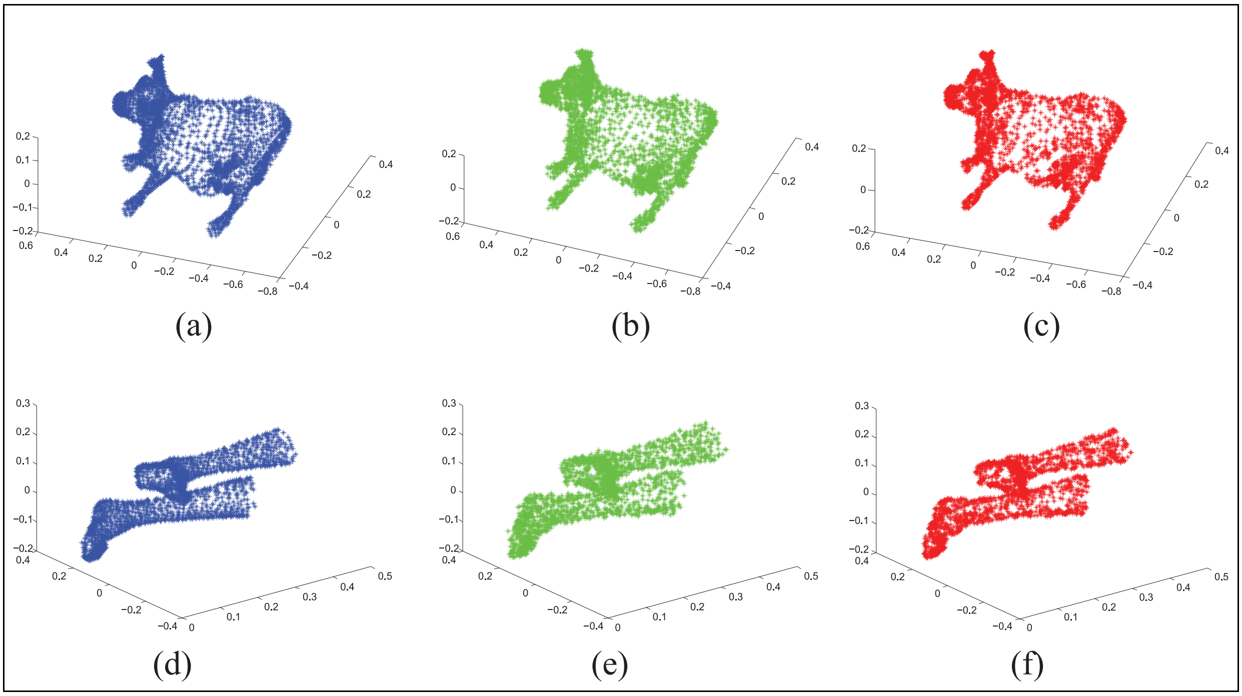

Figure 4. Assessing the robustness of the noise: (a) cow, (b) cow adding noise, (c) cow removing noise, (d) feet, (e) feet adding noise, and (f) feet removing noise.

Experimental simulation

This experiment is performed in MATLAB R2012b and the datasets are from Stanford's experimental database.

Robustness

In order to assess the robustness of the noise, the experimental dataset contains two models ("cow" and "feet of man"). In the test, we use eight nearest neighbors of each point and add the random noise to the point cloud. The result is shown in Figure 4. From the results, we find that the descriptor is workable when facing such a challenging case.

Registration accuracy test

In order to test the registration accuracy, the experimental dataset contains three models ("bunny,""cow," and "man"). The dataset used in complete overlapping is shown in Figure 5 and in partial overlapping is shown in Figure 6. The red one is the original point cloud and the blue one is the target point cloud.

Performance of EPSA. We calculate the mean square error (MSE) to further test the performance of EPSA. MSE is defined in the following

$$MSE = \frac{1}{n}\sum_{k=1}^{n} \min\left[(x_{ok}-x_{tk})^2 + (y_{ok}-y_{tk})^2 + (z_{ok}-z_{tk})^2\right] \quad (12)$$

where MSE is the mean square error; x_{ok}, y_{ok}, z_{ok} is the source point cloud; x_{tk}, y_{tk}, z_{tk} is the target one; and n is the smallest size of the point clouds.

In order to make the data in experiment, first, we calculate the center of gravity of two point clouds. For the point cloud $P = \{p_i(x_i, y_i, z_i) | \, i = 1, 2, 3, \ldots, N\}$, the gravity is defined as

$$O_i = \begin{bmatrix} \text{mean}(x_i) \\ \text{mean}(y_i) \\ \text{mean}(z_i) \end{bmatrix} \quad (13)$$

where mean() is the mean function. Then, T, which is the difference in the gravity center between two clouds, is obtained. Second, the entropy of the two point clouds is calculated, and PSO is used to search the R. The experimental flow is shown in Figure 7 and the steps are shown as follows:

Input: The original point cloud and the target point cloud.
Output: The rotation matrix R_g.

Step 1. Set the number of iterations, M, and the number of randomly generated rotation angles, N.

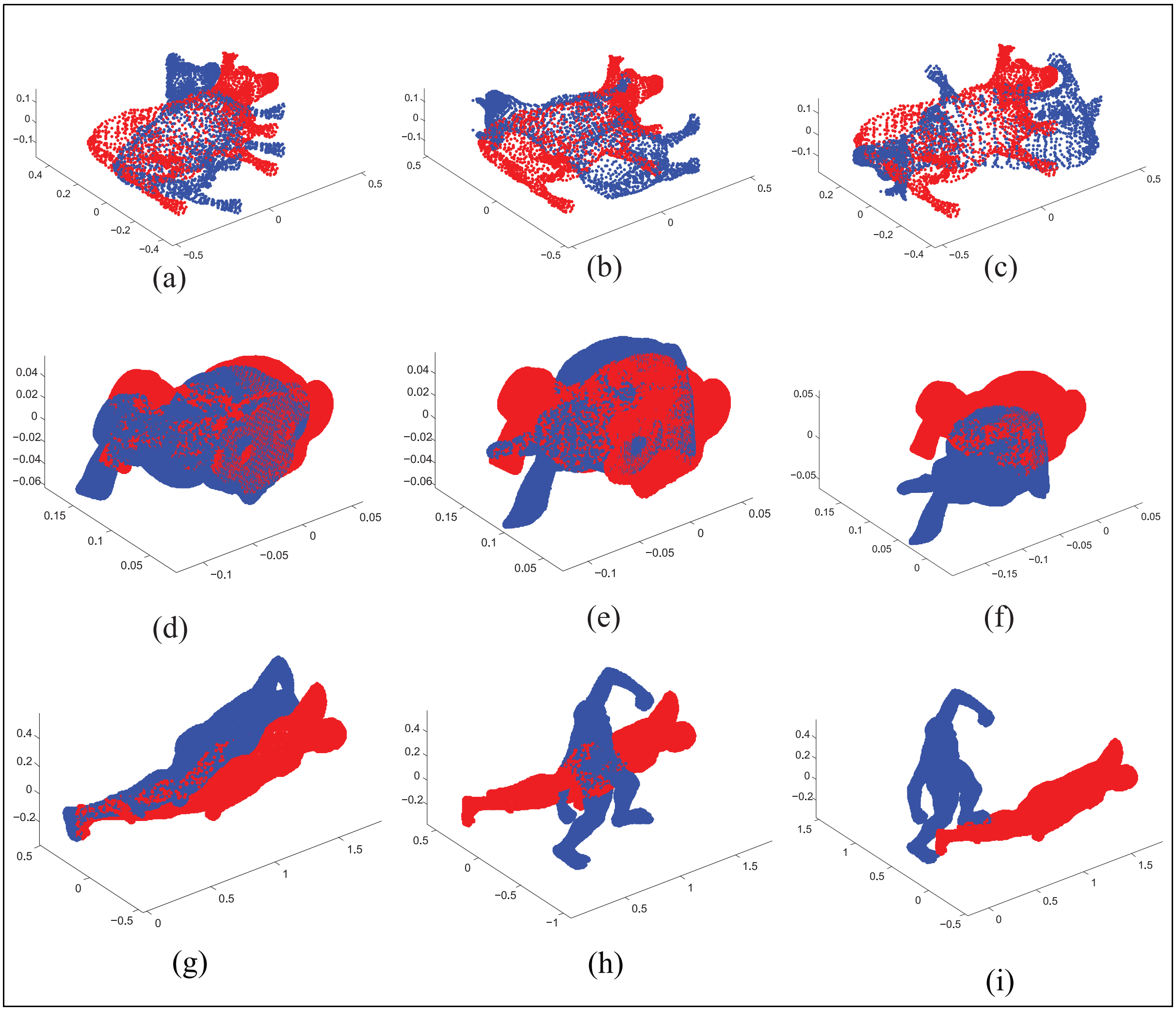

Figure 5. Dataset used in complete overlapping: (a) cow-view1, (b) cow-view2, (c) cow-view3, (d) bunny-view1, (e) bunny-view2, (f) bunny-view3, (g) man-view1, (h) man-view2, and (i) man-view3.

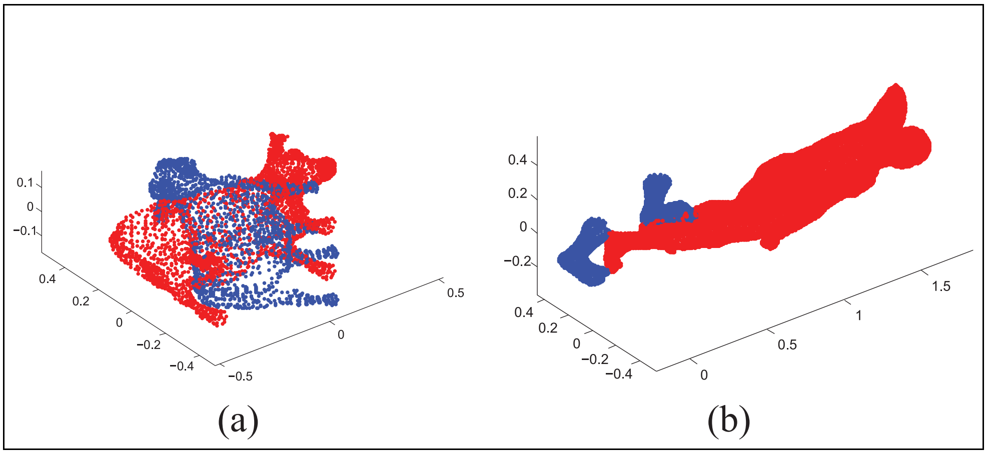

Figure 6. Dataset used in partial overlapping: (a) cow and (b) man.

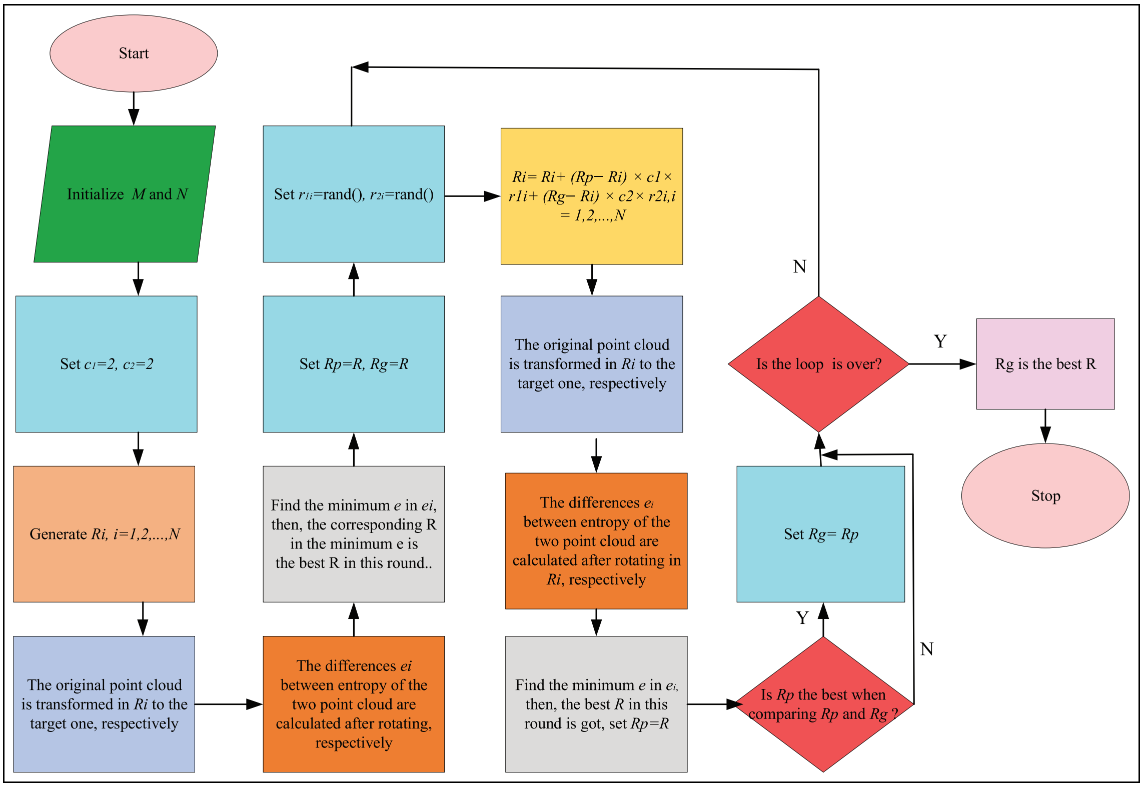
Start
Initialize M and N
Set c1=2, c2=2
Generate Ri, i=1,2,...,N
The original point cloud is transformed in Ri to the target one, respectively
The differences ei between entropy of the two point cloud are calculated after rotating, respectively
Find the minimum e in ei, then, the corresponding R in the minimum e is the best R in this round..
Set Rp=R, Rg=R
Set r1i=rand(), r2i=rand()
Ri= Ri+ (Rp− Ri) × c1 × r1i+ (Rg− Ri) × c2× r2i,i = 1,2,...,N
The original point cloud is transformed in Ri to the target one, respectively
The differences ei between entropy of the two point cloud are calculated after rotating in Ri, respectively
Find the minimum e in ei, then, the best R in this round is got, set Rp=R
Is Rp the best when comparing Rp and Rg ?
Y
N
Set Rg= Rp
Is the loop is over?
N
Y
Rg is the best R
Stop

Figure 7. Experimental flow.

Step 2. Set $c_1 = 2, c_2 = 2$, where c_1, c_2 are the learning factors or accelerators.

Step 3. The rotation matrices $R_i, i = 1, 2, \ldots, N$ is randomly generated.

Step 4. The original point cloud is transformed in $R_i, i = 1, 2, \ldots, N$, to the target one, respectively.

Step 5. The differences $e_i, i = 1, 2, \ldots, N$, between the entropy of the two point clouds are calculated after rotation, respectively.

Step 6. Find the minimum e in $e_i, i = 1, 2, \ldots, N$, then the corresponding R in the minimum e is the best R in this round.

Step 7. Set $R_p = R, R_g = R$, where R_p is the particle's best position in history, and R_g is the best position for all particles.

Step 8. Enter the loop.

(a) Set $r_{1i} = \text{rand}(), r_{2i} = \text{rand}(),\ i = 1, 2, \ldots, N$, where r_{1i}, r_{2i} are the pseudo-random numbers and rand() is the random function, which generates a random number between 0 and 1 from a uniform distribution.

(b) $R_i = R_i + (R_p - R_i) \times c_1 \times r_{1i} + (R_g - R_i) \times c_2 \times r_{2i}$, $i = 1, 2, \ldots, N$.

(c) The original point cloud is transformed in $R_i, i = 1, 2, \ldots, N$, to the target one, respectively.

(d) The differences $e_i, i = 1, 2, \ldots, N$, between the entropy of the two point clouds are calculated after rotation, respectively.

(e) Find the minimum e in $e_i, i = 1, 2, \ldots, N$, then the best R in this round is obtained, and set $R_p = R$.

(f) Compare R_p and R_g. If R_p is the best, set $R_g = R_p$.

(g) If the number of iterations, M, ran out, exit the loop. If not, repeat steps (a) to (f).

Case of the point cloud completely overlapped. Case of the point cloud completely overlapped is that the data of the point cloud belong to the complete object model and reflect the complete shape of the object. The results in complete overlapping are shown in Figures 8–10, and Table 2 shows the size of the experimental datasets and the *MSE* by comparing it with the descriptor coherent point drift (CPD).[37]

Case of the point cloud partially overlapped. Case of the point cloud partially overlapped is that the data of the

Table 2. MSE in complete overlapping.

Name	Size	MSE (EPSA)	MSE (CPD)
Cow-view1	2904×3	8.98×10^{-16}	9.21×10^{-5}
Cow-view2	2904×3	1.43×10^{-4}	1.3×10^{-3}
Cow-view3	2904×3	5.45×10^{-5}	1.3×10^{-3}
Bunny-view1	35, 947×3	2.28×10^{-17}	3.51×10^{-6}
Bunny-view2	35, 947×3	4.95×10^{-18}	3.51×10^{-6}
Bunny-view3	35, 947×3	2.64×10^{-13}	3.51×10^{-6}
Man-view1	12, 500×3	2.63×10^{-16}	1.37×10^{-4}
Man-view2	12, 500×3	2.75×10^{-8}	1.37×10^{-4}
Man-view3	12, 500×3	2.60×10^{-16}	1.37×10^{-4}

MSE: mean square error; EPSA: entropy and particle swarm algorithm; CPD: coherent point drift.

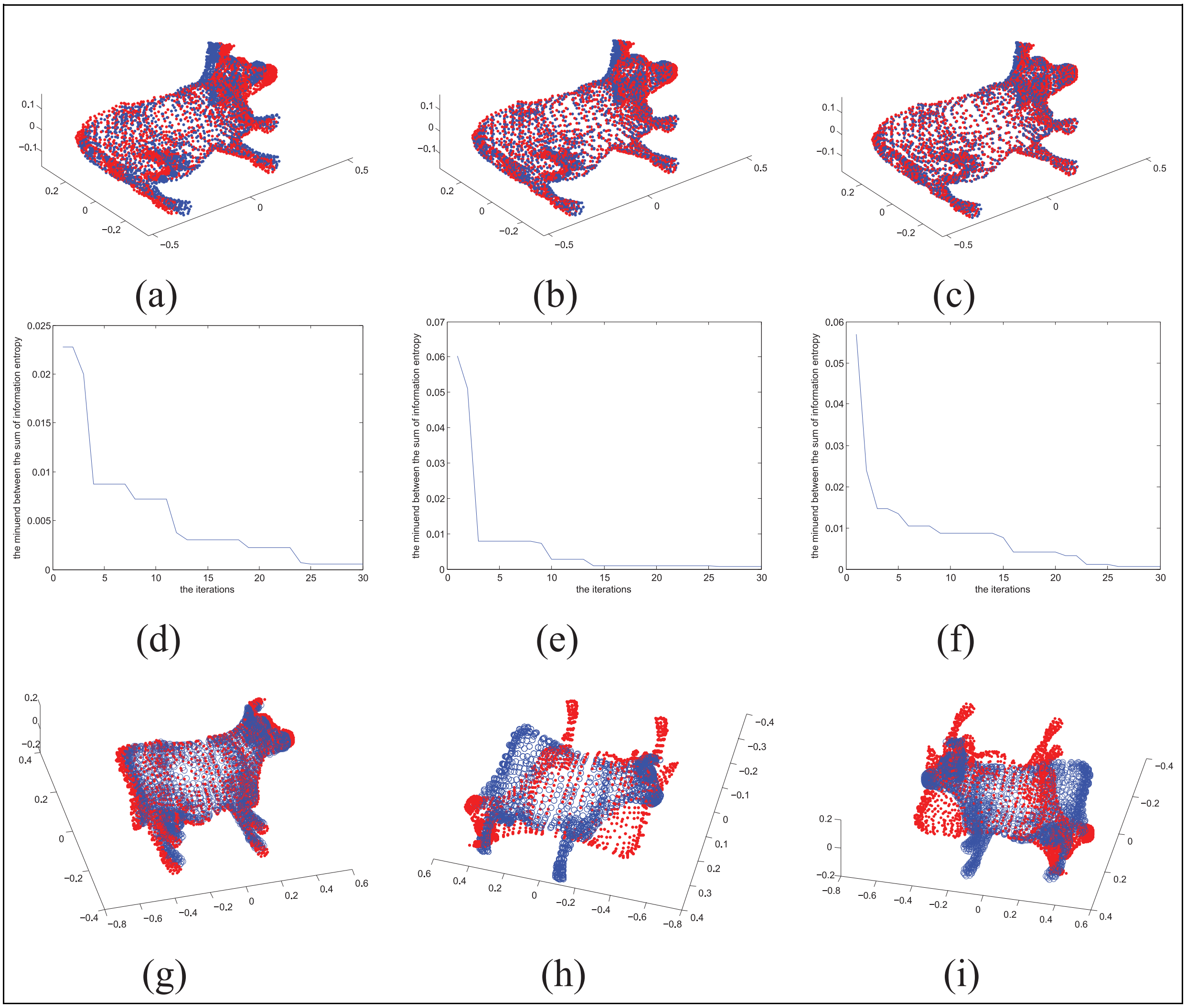

Figure 8. Experimental results of cow: (a) result, cow-view1 (EPSA); (b) result, cow-view2 (EPSA); (c) result, cow-view3 (EPSA); (d) curve, cow-view1 (EPSA); (e) curve, cow-view2 (EPSA); (f) curve, cow-view3 (EPSA); (g) result, cow-view1 (CPD); (h) result, cow-view2 (CPD); and (i) result, cow-view3 (CPD).

point cloud are only part of the object. The results in partial overlapping are shown in Figure 11, and Table 3 shows the *MSE* of the experimental datasets which are compared with the descriptor CPD.

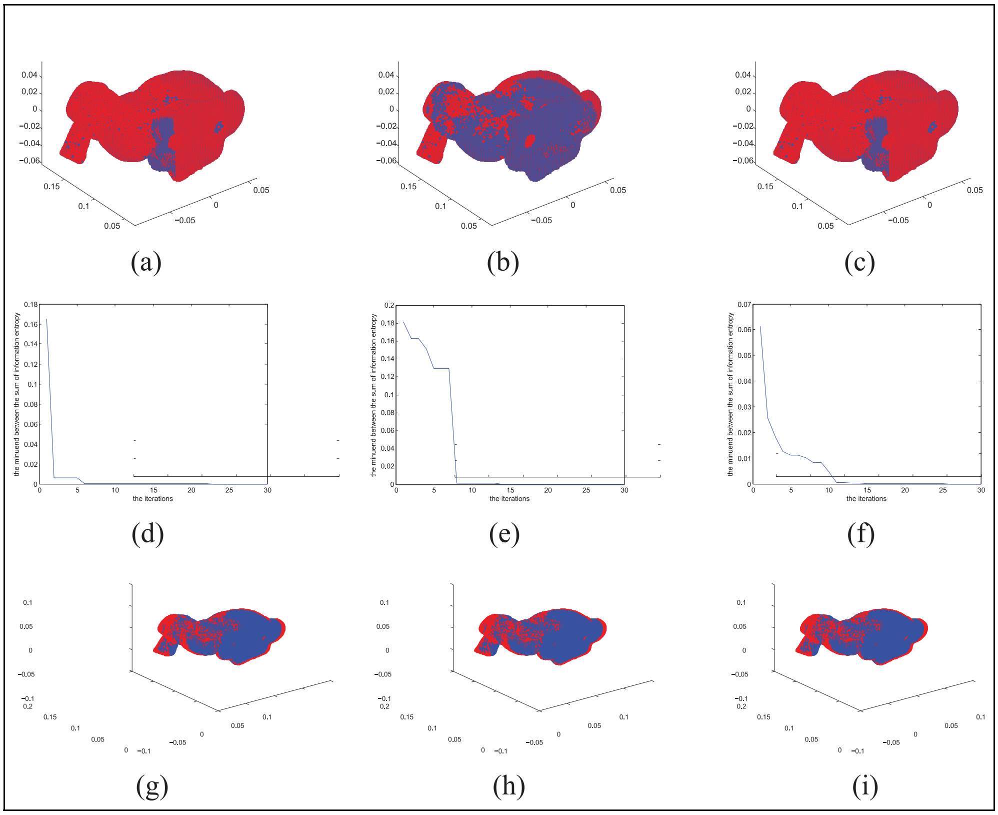

Figure 9. Experimental results of bunny: (a) result, bunny-view1 (EPSA); (b) result, bunny-view2 (EPSA); (c) result, bunny-view3 (EPSA); (d) curve, bunny-view1 (EPSA); (e) curve, bunny-view2 (EPSA); (f) curve, bunny-view3 (EPSA); (g) result, bunny-view1 (CPD); (h) result, bunny-view2 (CPD); and (i) result, bunny-view3 (CPD).

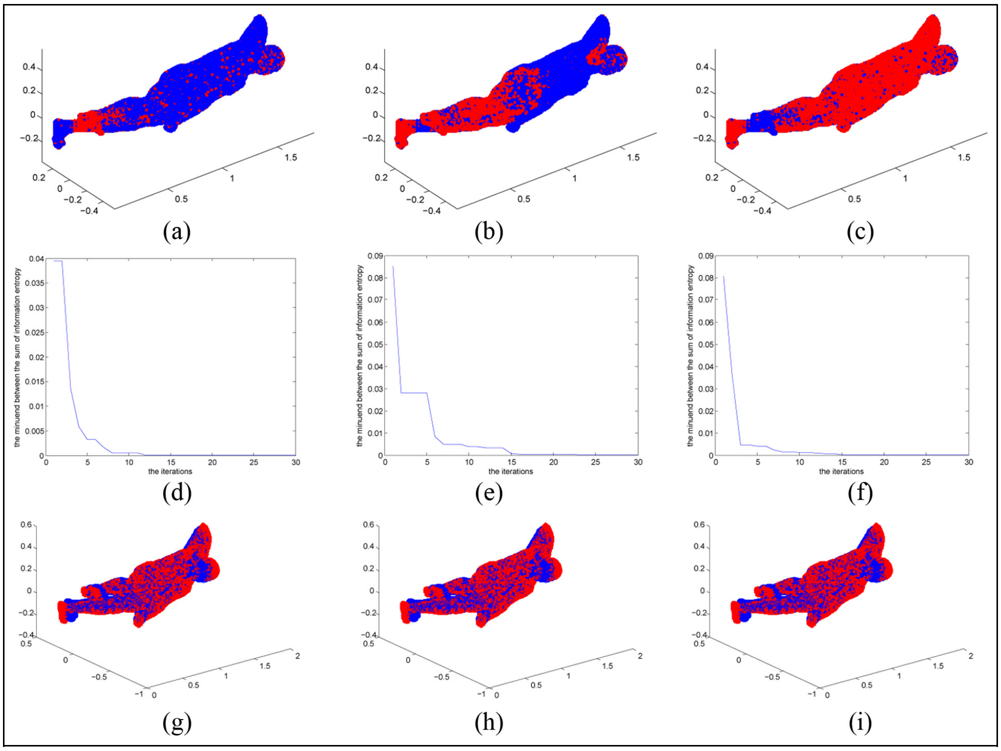

Figure 10. Experimental results of man: (a) result, man-view1 (EPSA); (b) result, man-view2 (EPSA); (c) result, man-view3 (EPSA); (d) curve, man-view1 (EPSA); (e) curve, man-view2 (EPSA); (f) curve, man-view3 (EPSA); (g) result, man-view1 (CPD); (h) result, man-view2 (CPD); and (i) result, man-view3 (CPD).

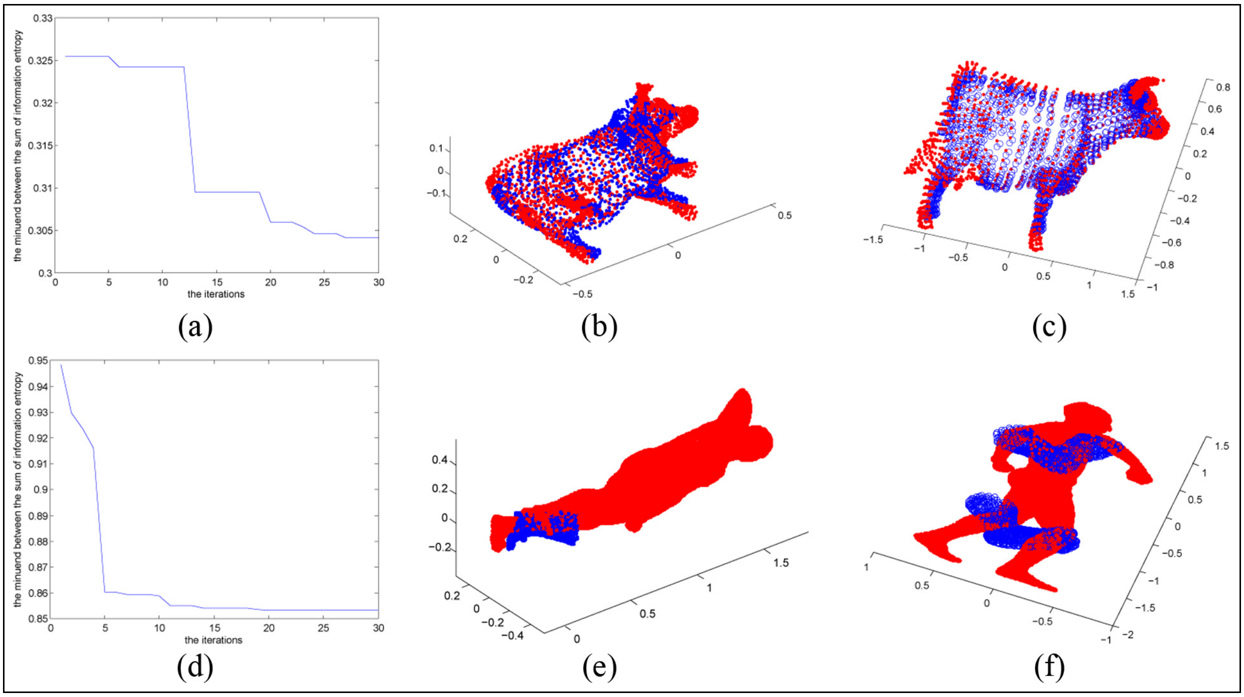

Figure 11. Experimental results in partial overlapping: (a) curve-cow (EPSA), (b) result-cow (EPSA), (c) result-cow (CPD), (d) curve-man (EPSA), (e) result-cow (EPSA), and (f) result-cow (CPD).

Table 3. MSE in partial overlapping.

Name	MSE (EPSA)	MSE (CPD)
Cow	2.51×10^{-4}	1.04×10^{-4}
Man	2.1×10^{-3}	2.3×10^{-3}

MSE: mean square error; EPSA: entropy and particle swarm algorithm; CPD: coherent point drift.

From Figures 8–11 and Tables 2 and 3, we find that the experimental datasets achieved the high correct rate followed by the proposed descriptor.

Conclusion

In this paper, a 3D point cloud registration based on entropy and particle swarm algorithm is proposed. Firstly, to find the k-nearest neighbor, the k-d tree is employed in the point cloud to establish the relationship in the points. The noise is suppressed by the mean of neighbor points. Secondly, in order to improve the registration accuracy, the gravity center of two point clouds is calculated to find T and the search constraint is used to find the best R. Once the R and T are found, the original point cloud can be transformed to the target one. Lastly, the experiment results are presented. They demonstrate that the algorithm is workable.

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 the National Natural Science Foundation of China (grant nos 11705122, 61803098, and 61803305), the Research Foundation of the Department of Education of Sichuan Province (grant nos 17ZA0271, 16ZA0263, 17ZB0194, 18ZB0418, 18ZA0357, and 18ZA0357), the Open Foundation of Enterprise Informatization and Internet of Things Key Laboratory of Sichuan Province (grant nos 2016WYJ03 and 2017WZY01), the Open Foundation of Artificial Intelligence Key Laboratory of Sichuan Province (grant nos 2016RYJ04, 2015RYJ03, 2016RYY01, 2017RZJ02, and 2017RYJ01), the Open Foundation of Sichuan Provincial Key Lab of Process Equipment and Control (grant no. GK201612), the Open Foundation of Material Corrosion and Protection Key Laboratory of Sichuan province (grant no. 2017CL09), the Innovation Platform & Key Project of International Cooperation of General Institutes of Higher Education of Guangdong Province & Overseas (including Hong Kong, Macao, and Taiwan; grant no. 2015KGJHZ025, Natural Science), Research & Innovation Foundation of General University Graduate of Jiangsu Province (grant no. CXZ Z13_0659), Sichuan Science and Technology Program (grant nos 19ZDZX0037, 2018JY0197, and 2016SZ0074), the Sichuan Key Provincial Research Base of Intelligent Tourism Foundation (grant no. ZHZJ18-01), and the Natural Science Foundation of Sichuan University of Science & Engineering (grant nos 2018RCL17, 2018RCL18, 2017RCL10, 2017RCL52, and 2017RCL12).

ORCID iD

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

References

1. Zhou C, Li Q, Li F, et al. Finite-time bounded functional observer design for a class of nonlinear systems. In: *Proceedings of the 37th Chinese control conference*, Wuhan, China, 25–27 July 2018, pp.36–41. New York: IEEE.
2. Cheng Y, Wang Y, Yu H, et al. Solenoid model for visualizing magnetic flux leakage testing of complex defects. *NDT&E Int* 2018; 100: 166–174.
3. Geng H, Liang Y and Cheng Y. Target state and Markovian jump ionospheric height bias estimation for OTHR tracking systems. *IEEE T Syst Man Cyb*. Epub ahead of print 4 May 2018. DOI: 10.1109/TSMC.2018.2822819
4. Cheng Y, Zhang Y, Shi L, et al. Consensus seeking in heterogeneous second-order multi-agent systems with switching topologies and random link failures. *Neurocomputing* 2018; 319: 188–195.
5. Sheng H, Xiao J, Cheng Y, et al. Short-term solar power forecasting based on weighted Gaussian process regression. *IEEE T Ind Electron* 2018; 65: 300–308.
6. Cheng Y, Tian L, Yin C, et al. A magnetic domain spots filtering method with self-adapting threshold value selecting for crack detection based on the MOI. *Nonlinear Dynam* 2016; 86: 741–750.
7. He P. Consensus of uncertain parabolic PDE agents via adaptive unit-vector control scheme. *IET Control Theory A*. Epub ahead of print 15 September 2018. DOI: 10.1049/iet-cta.2018.5202
8. He P. Pinning control and adaptive control for synchronization of linearly coupled reaction-diffusion neural networks with mixed delays. *Int J Adapt Control* 2018; 32: 1103–1123.
9. Zhao YP, He P, Saberi Nik H, et al. Robust adaptive synchronization of uncertain complex networks with multiple time-varying coupled delays. *Complexity* 2015; 20: 62–73.
10. He P, Jing CG, Fan T, et al. Robust decentralized adaptive synchronization of general complex networks with coupling delayed and uncertainties. *Complexity* 2014; 19: 10–26.
11. He P, Ma SH and Fan T. Finite-time mixed outer synchronization of complex networks with coupling time-varying delay. *Chaos* 2012; 22: 043151.
12. He P and Fan T. Distributed fault-tolerance consensus filtering in wireless sensor networks—part I: communication failure. *Int J Sens Netw* 2016; 22: 127–142.
13. Yue YG and He P. A comprehensive survey on the reliability of mobile wireless sensor networks: taxonomy, challenges, and future directions. *Inform Fusion* 2018; 44: 188–204.
14. 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: 1199–1217.
15. Wang K, He P, Tang J, et al. Static output feedback $\mathcal{H}^\infty$ control for active suspension system with input delay and parameter uncertainty. *Adv Mech Eng* 2018; 10: 1–14.
16. He P and Li Y. $\mathcal{H}^\infty$ synchronization of coupled reaction-diffusion neural networks with mixed delays. *Complexity* 2016; 21: 42–53.
17. He P, Li Y and Park JH. Noise tolerance leader-following of high-order nonlinear dynamical multi-agent systems with switching topology and communication delay. *J Frankl Inst* 2016; 353: 108–143.
18. 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, IEEE Computer Society*, Quebec, QC, Canada, 28 May–1 June 2001, pp.145–152. New York: IEEE.
19. Rodrigues M, Fisher R and Liu Y. Special issue on registration and fusion of range images. *Comput Vis Image Und* 2002; 87: 1–7.
20. Liu Y. A mean field annealing approach to accurate free form shape matching. *Pattern Recogn* 2007; 40: 2418–2436.
21. Rusu RB, Blodow N and Beetz M. Fast point feature histograms (FPFH) for 3D registration. In: *Proceedings of the IEEE international conference on robotics and automation, IEEE Robotics and Automation Society*, Kobe, Japan, 12–17 May 2009, pp.3212–3217. New York: IEEE.
22. Rusu RB, Blodow N, Marton ZC, et al. Aligning point cloud views using persistent feature histograms. In : *IEEE/RSJ international conference on intelligent robots and systems, Acropolis Convention Center, Nice*, 22–26 September 2008, pp.3384–3391. New York: IEEE.
23. Rusu RB, Marton ZC, Blodow N, et al. Persistent point feature histograms for 3D point clouds. In: *Proceedings of 10th international conference intelligent autonomous system, Acropolis Convention Center, Nice*, 22–26 September 2008, pp.119–128. New York: IEEE.
24. Tombari F, Salti S and Di Stefano L. Unique signatures of histograms for local surface description. In: *European conference on computer vision, Foundation for Research and Technology—Hellas*, Crete, 5–11 September 2010, pp.356–369. Berlin: Springer-Verlag.
25. Salvi J, Matabosch C, Fofi D, et al. A review of recent range image registration methods with accuracy evaluation. *Image Vision Comput* 2007; 25: 578–596.
26. Kase K, Makinouchi A, Nakagawa T, et al. Shape error evaluation method of free-form surfaces. *Comput Aided Design* 1999; 31: 495–505.
27. Zaharescu A, Boyer E and Horaud R. Keypoints and local descriptors of scalar functions on 2D manifolds. *Int J Comput Vision* 2012; 100: 78–98.
28. Bueno M, Bosché F, González-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: 120–134.

29. Hidaka N, Michikawa T, Motamedi A, et al. Polygonization of point clouds of repetitive components in civil infrastructure based on geometric similarities. *Automat Constr* 2018; 86: 99–117.
30. Gold S and Rangarajan A. A graduated assignment algorithm for graph matching. *IEEE T Pattern Anal* 1996; 18: 377–388.
31. 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, IEEE Computer Society*, Quebec, QC, Canada, 6–10 October 2003, pp.427–433. New York: IEEE.
32. Shannon CE. A mathematical theory of communication. *Bell Sys Tech J* 1948; 27: 379–423.
33. Cao B, Zhao J, Lv Z, et al. Distributed parallel particle swarm optimization for multi-objective and many-objective large-scale optimization. *IEEE Access* 2017; 5: 8214–8221.
34. Cao B, Zhao J, Lv Z, et al. Deployment optimization for 3D industrial wireless sensor networks based on particle swarm optimizers with distributed parallelism. *J Netw Comput Appl* 2018; 103: 225–238.
35. Cao B, Zhao J, Lv Z, et al. A distributed parallel cooperative coevolutionary multiobjective evolutionary algorithm for large-scale optimization. *IEEE T Ind Inform* 2017; 13: 2030–2038.
36. Clerc M. Handbook of Swarm Intelligence. In: Panigrahi BK, Shi Y and Lim MH (eds) *From Theory to Practice in Particle Swarm Optimization*. Berlin Heidelberg: Springer, 2011, pp. 3–36.
37. Myronenko A and Song X. Point set registration: coherent point drift. *IEEE T Pattern Anal* 2010; 32: 2262–2275.