Fast and coarse registration of point cloud in ICP three-dimensional space

Xiaokang Ren 1, Mai Zhang 2, Wenqiao Wang 3, Xuetao Mao 4 and Jie Ren 5

12345 School of Computer Science and Engineering, Northwest Normal University, Lanzhou, Gansu 730070, China

*Corresponding author’s e-mail: 2424487218@qq.com

Abstract. The traditional ICP algorithm has a large amount of computation in the process of searching for corresponding points [1]. This is the bottleneck of the traditional ICP algorithm [2], and the standard ICP is looking for the corresponding point [3]. The point where the Euclidean distance is closest is the corresponding point [4]. Unreasonable, there will be a certain number of wrong points [5]. In order to improve the accuracy and speed of 3D point cloud data registration, this paper combines the coarse registration of the center of the box and the circular domain search method of the rasterized small cube to further improve and optimize the traditional ICP algorithm [6].

1. Introduction
Point cloud data can obtain accurate topological structure and geometry of objects with a small storage cost [7], and thus has gained more and more attention [8]. Among the registration algorithms, the most used by researchers is the ICP algorithm [9]. The ICP (Iterative Closest Point Iterative Point) algorithm is a point set to point set registration method [10]. The ICP algorithm has the following conditions [11]:

1. The registration result is highly accurate and is an accurate registration algorithm;
2. The initial matrix is strictly required, and the poor initial matrix seriously affects the performance of the algorithm, and even causes a local optimum [12].

Therefore, in view of the above limitation of traditional ICP, this paper uses the rounded domain search method of rasterized small cubes to establish a large-scale point cloud search method to improve the search efficiency and select the rectangular bounding box. The method of center coincidence coarsely registers the point cloud dataset, provides a good initial iteration value for subsequent ICP fine registration, and optimizes the traditional ICP algorithm to further improve the accuracy and speed of 3D point cloud data registration.

2. Improved 3D point cloud registration algorithm

2.1. Rasterized small cube circular domain search for massive data point pairs
This paper improves on the basis of traditional rasterization, uses the circular domain to limit the search space range of the point cloud, and ensures the scanned point cloud data in the effective point cloud search space through the cell matrix, especially for the massive point cloud data search. It can further speed up search efficiency. To establish a topological relationship of point cloud search, when performing a K-nearest neighbor search on a point \( p_i \) of the source point cloud, the search range is first locked in the grid where the \( p_i \) is located and in the grid above and below, left and right, and adjacent.
the pi is near the edge of the grid, the point closest to the pi is more likely to be in the adjacent grid, and the search is done in the adjacent grid. When the target grid is searched, if there are still less than \( K \), then the search range is expanded by one adjacent circle, until \( K \) is satisfied, and the search ends. Rasterized search and circular domain partial search process as shown below:

![Figure 1. 3D grid method circular domain search box](image1)

![Figure 2. 3D grid method circle domain step search center point](image2)

The minimum coordinate point in the point cloud data set is the center point \( O \) of the initial cell circle, and the rasterized circle domain step search of the three dimensions of \( X, Y, \) and \( Z \) is performed, and the positive and negative \( radius \) are used as the maximum search radius of the circle domain. To limit the search range of the 3D spatial point cloud data points, and sequentially scan the point cloud data points in the small grid.

### 2.2. Cuboid Bounding Box Center Heavy Law

The three-dimensional space coordinate limitation is used to limit the boundary of the point cloud data set point to ensure that the scanned point cloud data is within the effective point cloud space, reducing the invalid search and improving the search efficiency of the point cloud data set. The cuboid center weighting method uses the point cloud \( A \) to coincide with the center of the point cloud \( B \) to be registered in the point cloud coarse registration, thereby reducing the translation difference between the two sets of point cloud data, and increasing the coincidence rate between the point clouds, and obtaining a better initial registration matrix.

The method for determining the minimum cuboid bounding box center is as follows:

1. Determine the minimum cuboid bounding box of point cloud data
   - Suppose \( in \) is a three-dimensional matrix in space. Use \([N,b] = \text{size}(in)\) in matlab to find \( N \) as the number of rows, the total number of point clouds, and \( b \) is set to 3.
   - Find the maximum and minimum element values of the first, second and third column in the matrix \( in \) of the \( x \)-axis, \( y \)-axis and \( z \)-axis directions:
     
     \[
     \begin{align*}
     \text{max}_x &= \max(in(:,1)); \\
     \text{min}_x &= \min(in(:,1)); \\
     \text{max}_y &= \max(in(:,2)); \\
     \text{min}_y &= \min(in(:,2)); \\
     \text{max}_z &= \max(in(:,3)); \\
     \text{min}_z &= \min(in(:,3));
     \end{align*}
     \]

     Therefore, the side length of the box enclosing box of point cloud data is:

     \[
     \begin{align*}
     L_x &= \text{max}_x - \text{min}_x; \\
     L_y &= \text{max}_y - \text{min}_y; \\
     L_z &= \text{max}_z - \text{min}_z;
     \end{align*}
     \]

   \( L_x, L_y, \) and \( L_z \) are the side lengths of the bounding box in the \( x \)-axis, \( y \)-axis and \( z \)-axis directions.

   You can get the bounding box of the smallest box of point cloud data.

2. Determine the number of data points in the unit small cube grid
   - From the length of the smallest cuboid obtained above, the volume of the bounding box is:
     \[
     V = L_x * L_y * L_z;
     \]
   - Then the number of data points in the unit small cube grid is:
     \[
     n = N / (V + \text{eps});
     \]
     where \( N \) is the total number of data points in the point cloud, \( \text{eps} = \text{eps}(1) \), which is the precision of 1.
Then the side length \( L_s \) of the subcube grid is:
\[
L_s = (\lambda \cdot K / (n + \varepsilon))^{(1/3)}
\]
where \( \lambda \) is the scale factor used to adjust the side length of the subcube grid, \( K \) is the number of neighbors, \( (\lambda \cdot K / (n + \varepsilon)) \) (substitute \( n \)) into sub-cube grid volumes.

Finally, the point cloud data is divided into \( mm \times nn \times ll \) small cube grids in the \( x \)-axis, \( y \)-axis, and \( z \)-axis directions:
\[
mm = \text{floor}(L_x / (L_s + \varepsilon)) + 1;
nn = \text{floor}(L_y / (L_s + \varepsilon)) + 1;
ll = \text{floor}(L_z / (L_s + \varepsilon)) + 1;
\]
where \( L_x, L_y \) and \( L_z \) are the side lengths of the bounding box on the \( x \), \( y \) and \( z \) axis.

1. First set an all-zero matrix of 3 rows of \( mm \times nn \times ll \) rows (because the point cloud data is divided into \( mm \times nn \times ll \) small cube grids in the \( x \)-axis, \( y \)-axis and \( z \)-axis directions);
2. Then iteratively update the center coordinate \( \text{center}_x \) in the \( x \)-axis direction, and add the side length of the small grid once per iteration, find the center point in the \( x \)-axis direction one by one. In the \( y \)-axis direction and the \( z \)-axis direction, the center points \( \text{center}_y \) and \( \text{center}_z \) in their respective directions can be obtained in the same manner;
\[
\text{center}_x = \text{min}_x + L_s \cdot (i - 0.5); \quad \text{(minimum \( \text{min}_x \) of the element in the \( x \)-axis direction in the three-dimensional matrix \( in \))}
\]
3. Finally, circulate in 3D space in \((i-1) \times nn \times ll, (j-1) \times ll\) traverse in a two-dimensional plane, \( k \) traverse in one dimension line, and finally each one can be obtained. The center coordinates of the bounding box \( (\text{center}_x, \text{center}_y, \text{center}_z) \).

That is, one data point exists in three dimensions at the same time, the three-axis three-line intersection is the center coordinate of the bounding box \( (\text{center}_x, \text{center}_y, \text{center}_z) \).

![Figure 3. Schematic diagram of the center of the smallest cuboid bounding box](image)

![Figure 4. Schematic diagram of the coincidence of the source point cloud center point and the target point cloud center point](image)
2.3. Improved ICP algorithm overall steps

![Diagram of Improved ICP algorithm](image)

(1) Calculate the initial translation matrix, perform coarse registration, and obtain the coordinates of the point cloud to be registered after the coarse registration.

① Calculate the center of the source point cloud \( P \) and the target point cloud \( Q \), respectively \( T_P, T_Q \);

\[
T_P = [\text{center}_xP, \text{center}_yP, \text{center}_zP],
\]

\[
T_Q = [\text{center}_xQ, \text{center}_yQ, \text{center}_zQ].
\]

② \( T_P, T_Q \) is the difference, the transposed matrix of the difference is used as the translation vector \( T_o \);

\[
T_o = (T_Q - T_P)'
\]

\( T_o \) is the translation vector, and \( T_P \) and \( T_Q \) are the centers of the point set.

③ Using the translation vector \( T_o \) to translate the points of the registration point cloud \( P \);

\[
P_i' = pi - T_o
\]

\( P_i \) is the cluster of points to be registered, \( T_o \) is the translation vector, and \( pi' \) is the point cloud coordinates after the centering. Centering the two sets of point cloud data can reduce the translational misalignment between the point clouds, provide the initial position for the next fine registration, and improve the registration accuracy of the point cloud.

(2) Calculate the nearest neighbors of the target point cloud after the centralization, and the obtained point pairs are \((pi', qi) \ (i = 1, 2, ..., N)\).
(3) When finding the subscript \( j \) when the corresponding point distance is the smallest, the quaternary function is called to find the rotation and translation matrices \( R_m \) and \( T_m \).

(4) Find the point cloud after rotation and translation: \( p = R_m*p + T_m \).

(5) Using the least squares method to minimize the error function,

\[
E(R,T) = \frac{1}{n} \sum_{i=1}^{n} \| q_i - (R \cdot p_i + T) \|^2
\]

If the corresponding point distance \( d \) is less than a given threshold or greater than the preset maximum number of iterations, the iterative calculation is stopped; otherwise, step (1) is returned until the convergence condition is satisfied.

3. Experiment and analysis
The experiment uses the standard point cloud dataset, the frog model frog-deg320, frog-deg340 as the point cloud registration data, and compares the algorithm registration results before the traditional ICP is not improved, in order to verify the algorithm effect.

As shown in the above figure, Figure (a) is the initial point cloud map to be registered (green is the source, red is the target), and the initial two sets of point cloud clusters have larger offset patterns; Figure (b) is the bounding box center point translation vector \( T_0 \). After the initial translation, the difference between the source point cloud and the target point cloud shows that the distance between the pair of point cloud points to be registered has been further reduced. Compared with the graph a, the point cloud cover of the red and green parts is more uniform. The overlap rate is higher.

Table 1. Comparison of frog model frog-deg320 and frog-deg340 point cloud registration

|                | time consuming | number of iterations | error  |
|----------------|----------------|----------------------|--------|
| traditional ICP| 65.614813s     | 80                   | 1.50193|
| improved algorithm | 45.237326s    | 71                   | 1.50193|

Figure 7. Comparison of the number of iterations before and after the improvement of the algorithm

Figure 8. Comparison of iterations and running time
The green source point cloud in Fig. 5(a) contains 10689 data points, and the red target point cloud in Fig. 5(a) contains 10002 data points. From the registration effect diagram, the point cloud data after registration of this method can be seen. The overlap is larger than when it is not registered, and the registration is more uniform. It can be seen from Table 1 that under the premise of ensuring the accuracy of registration, after the improvement, the time-consuming and iterative times of the mass point cloud data registration are further reduced, and the registration efficiency is further improved by 31.06% on the basis of the original, which is for the massive point. Cloud data points are very efficient.

4. Conclusion
In this paper, based on the combination of the circular domain search of the rasterized small cube and the coarse registration of the center of the box, the traditional ICP algorithm is further improved and optimized, which further reduces the search time of the pair, improves the overlap rate, and reduces the mismatching point. The registration accuracy is improved, and the fast and accurate registration is realized especially for massive point cloud data. The future work is based on reducing the registration error of the point cloud, and hopes to obtain a global registration method with low registration error and few registration time.

References
[1] Liu, T., Luo, T.N., Qiao, L.Y. (2016) Global registration method of 3D surface based on branch and bound. Journal of Instrumentation, Commun., 37(8): 1869–1877.
[2] Jin, X.S., Song, Y.Q. (2019) Point cloud coarse registration method in 3D reconstruction technology. Journal of Harbin University of Commerce (Natural Science Edition), Commun., 35(2): 195–198.
[3] Zhou, M.Q., Geng, G.H. (2017) Point cloud registration algorithm based on bounded rotation angle. Microelectronics and Computer, Commun., 34(3): 46–49.
[4] Ren, X.K., Shi, X. (2018) Branch and Bound Point Cloud Registration Method Based on Neighborhood Curvature. Microelectronics and Computer, Commun., 35(6): 7–10.
[5] Wang, Y., Tang, J., Rao, Q.F., Yuan, C.Y. (2015) Point cloud data registration using point cloud center of gravity distance for boundary detection. Small microcomputer system, Commun., 39(9): 2096–2101.
[6] Liu, J., Zhang, X., Zhu, J.W. (2016) An ICP three-dimensional point cloud registration method based on K-D tree optimization. Surveying Engineering, Commun., 25(6): 15–18.
[7] Qiu, S.C., Luo, Y. (2017) Improved point cloud registration of ICP algorithm. Henan Science and Technology, Commun., 2017(7): 40–42.
[8] Huang, Y., Da, F.P., Tao, H.J. (2015) A point cloud automatic registration algorithm based on feature extraction. Chinese laser, Commun., 2015(3): 242–248.
[9] Zou, M. (2017) Multi-view cloud registration method based on ICP algorithm. Technological innovation and productivity, Commun., 2017(2): 74–76.
[10] Xue, Y.H., Liang, X.Z., Ma, T. (2011) An automatic registration method for scanning point clouds. Journal of Computer-Aided Design & Computer Graphics, Commun., 23(2): 223–231.
[11] Liu, Z.J. (2016) Open CL-based ICP point cloud parallel registration algorithm. Computer application and software, Commun., 2016(11): 185–187+192.
[12] Sun, J.Z., Chen, H., Geng, G.H. (2016) 3D artifact point cloud model registration optimization algorithm. Journal of Computer-Aided Design & Computer Graphics, Commun., 28(7): 1068-1074.