Composite clustering normal distribution transform algorithm

Tian Liu, Jiongshi Zheng, Zhenting Wang, Zhengdong Huang and Yongfu Chen

Abstract
Scan registration is a fundamental step for the simultaneous localization and mapping of mobile robot. The accuracy of scan registration is critical for the quality of mapping and the accuracy of robot navigation. During all of the scan registration methods, normal distribution transform is an efficient and wild-using one. But normal distribution transform will lead to the unreasonable interruption when splitting the grid and can’t express the points’ local geometric feature by prefixed grid. In this article, we propose a novel method, composite clustering normal distribution transform, which comprises the density-based clustering and \( k \)-means clustering to aggregate the points with similar local distributing feature. It takes singular value decomposition to judge the suitable degree of one cluster for further division. Meanwhile, to avoid the radiating phenomenon of LIDAR in measuring the points’ distance, we propose a method based on trigonometric to measure the internal distance. The clustering method in composite clustering normal distribution transform could ensure the expression of LIDAR’s local distribution and matching accuracy. The experimental result demonstrates that our method is more accurate and more stable than the normal distribution transform and iterative closest point methods.

Keywords
Composite clustering, scan registration, normal distribution transform, SLAM, wheeled mobile robot

Introduction
In unknown environment, simultaneous localization and mapping technology (SLAM)\(^1\) is the cornerstone for the autonomous and intelligence of mobile robot.\(^2\) Scan registration is a fundamental step for SLAM and autonomous navigating.\(^3\) Surrounding information of robot is obtained by distance sensors, such as LIDAR or depth camera. Scan registration algorithm first establishes the association of two near LIDAR frames, then constructing the objective function. Finally, it will be optimized to get the transform between two LIDAR frames. With continuing scan registration, the environment model around the robot is built.

At present, there are two mostly widespread scan registration algorithms. The first one is based on iterating the closest points. The typical algorithm is iterative closest point (ICP).\(^4\) The second one is based on the probability density function to build the relation. The objective function is built by the probability distributing model of current frame in reference frame. Its typical algorithm is normal distribution geometric constraints of points, lines, and surfaces between two frames. The typical algorithm is iterative closest point (ICP).\(^4\) The second one is based on the probability density function to build the relation. The objective function is built by the probability distributing model of current frame in reference frame. Its typical algorithm is normal distribution

Corresponding author:
Yongfu Chen, National CAD Support Software Engineering Research Center, Huazhong University of Science and Technology, Room A610, Wuhan 430074, China.
Email: chenyf@mail.hust.edu.cn

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://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).
transform (NDT). The main idea of this article is based on NDT method.

In this article, by analyzing the characteristic of NDT, we find that when fitting the local region of LIDAR data, NDT method will lead to unreasonable interruption while splitting points by normal distribution. Also NDT can’t express the local geometric features by prefixed grid cells. A new scan registration algorithm based on composite clustering is proposed in this article, in which the points with similar local feature in the reference frame are aggregated. The points are clustered by density-based spatial clustering of applications with noise (DBSCAN) and k-means method. DBSCAN is used to aggregate the region with continuous point distribution in reference frame. Then, the unsuitable clusters are selected by singular value decomposition (SVD)-based method and subdivided again by k-means if this cluster is unsuitable for normal fitting. The bounding box of each cluster is built for search and index. Finally, objective function of all matches is formed by normal distribution function and then optimized by trust region method.

The main contributions of this article are as follows:

1. We propose a novel clustering method to aggregate LIDAR points by their local features, which is a combination of density-based clustering and k-means clustering.
2. To judge a cluster’s suitable degree for NDT matching after DBSCAN, we propose a ratio factor based on SVD decomposition to get a cluster’s singular values.
3. We propose a novel distance measuring method based on trigonometric to evaluate the neighbor LIDAR points, which eliminate the LIDAR’s radiation phenomenon.

This article is organized as follows. In “Related work” section, we discuss the related work about scan registration. In “Problem statement” section, the basic procedure of NDT is narrated, and the disadvantage of original NDT method is discussed. Section “Composite clustering scan registration” is the main text of this article, and the overall procedure of composite clustering NDT (CCNDT) is narrated in detail. In “Experiments” section, the comparing experiments among CCNDT, NDT, and ICP are designed and discussed. The last section is a brief conclusion of this article.

Related work

LIDAR is a main exteroceptive sensor in mobile robot when acquiring distance data from the robot’s surrounding. In contrast with wheel odometer and IMU, LIDAR information represent the correlation between robot and environment. With the map built by LIDAR data, robot can realize where it locates and where it is the destination and path. Scan registration is a method to get the transform between LIDAR frame, which is critical for robot.

Scan registration aims to construct the relation of two LIDAR frames by matching their points, then building the objective function and optimizing. The most representable scan registration methods are ICP and NDT methods.

ICP is first proposed by Besl and Mckay, in which the frame relation is built by matching the closest point pairs. But without additional information such as geometric characteristics, the matching is worse when there is a large transform. Then, Segal et al. propose a general ICP method using the local inner points to get the face structure, which makes the result more accurate. Serafin and Gri- setti use the surface characteristics around the points to search for ICP corresponds. Cox and colleagues extract line features from LIDAR data and match the points with relevant lines. Iterative dual correspondence method is used to improve matching accuracy by maintaining two relevant sets. Wang et al. come up with a multilayer matching method to deal with the data-association problem, in which the uncertainty of the matching results is described by Fisher information matrix. MbICP is designed to improve convergence with large initial orientation errors, which explicitly put a measure of rotational error as part of the distance metric. Those ICP methods are easy to deploy and use, but their two weaknesses are considerable: the nearest match is not correct matching in many conditions and the computing cost is very heavy when searching the nearest points.

To avoid the heavy matching of ICP, Biber and Strasser come up with the probability-based method named as NDT (normal distribution method). It divides the reference frame into grid cells. Then, the normal distributing parameter is calculated for each cell. Next, each point of current frame is matched with a grid by their coordinate to get the probability density. With this method, it needn’t match each point but the grid cells to reduce computational cost. Moreover, normal distribution can provide the geometric representation of points, which gives the higher order derivatives for optimizing.

Based on NDT, Gutmann and Konolige registrant current frame with serval previous frame to improve precision. Mitra et al. use quadratic approximants to get the distance from target surface, describing the model surface implicitly. Ulas and Temeltas come up with a feature-based NDT method. Kunjin Ryu et al. use ND-to-ND method to get global matching, which can not only provide the internal frames transform but also be used for loop closure detection for graph optimization. Gouveia et al. use the multithreading to parallelize the data fusion step. Even more, Bosse et al. match all the LIDAR frames by registering the subregions and global map. And in kinect fusion by Newcombe et al., the relationship is built by a global optimizing model, but it is too heavy and slow for most calculating platforms. There are also many other scan-matching methods such as expectation-maximization (EM) matching method by Thrun et al. and the graph model method by Feng and Milios.
administration aims to get the coordinate transform $D_m$ points of surrounding in each pose. And NDT method will build the relationship between points and grid cells of two neighbor LIDAR frames, such as $P^{m-1}$ and $P^m$. Then, the transform relation will be calculated by objective function. NDT: normal distribution transform.

**Problem statement**

Compared with other scan registration methods, NDT has the following two advantages when building the correlation:

1. The normal distributing model gives the points in one cluster a common attribute, which reduces the quantity in matching.
2. Compared with the error function of point distance, the normal distributing error function has higher-order derivation, which makes it possible to use the optimizing method with the second-order gradient.

The scheme of NDT includes three parts, as shown in Figure 1. Firstly, the points in reference frame are divided into grid cell and constructed the normal distribution of the points in each cell. Then, the points of current frame are linked with those cells by their relevant coordinate. Finally, the error of those links is minimized to get the transform. Next, we’ll have an elaborate narration of the inference.

When the robot moving, distance points are obtained from LIDAR. Let $P^m = \{p^m_i \mid i \in \{1, \ldots, n_m\}\}$ and $P^{m-1} = \{p^{m-1}_i = [x \ y] \mid i \in \{1, \ldots, n_{m-1}\}\}$ individually represent the current and previous frames indexed by $m$, $m - 1$, and the point number is denoted by $i$. Scan registration aims to get the coordinate transform $\Delta^{m-m-1} = [x_\Delta \ y_\Delta \ \theta_\Delta]^T$, where $x_\Delta, y_\Delta$ is a plane transform and $\theta_\Delta$ is rotation. In this article, we use “reference frame” to represent the previous frame $P^{m-1}$.

In NDT method, the reference frame is first divided into grids, as shown in Figure 2. In advance, it needs to get the grid span in vertical and horizontal. Then, the frame is divided into span distance and cell interval. The values of span distance and cell interval are manually tuned.

![Figure 1. Schematic illustration of NDT method.](image)

![Figure 2. Grid splitting illustration with LIDAR data from ACES building at the University of Texas.](image)
frame is linked to a relevant grid cell by coordinate. The point’s error function is obtained by normal distributing density function

\[
(i) f_k^m = \eta_{\text{norm}} \ast \exp \left( \left( (i) \mathbf{z}_k^{m-1} \right)^T \Sigma^{-1}_i \left( (i) \mathbf{z}_k^{m-1} \right) \right)
\]

where \( f \) is a normal distributing probability density from points \( k \) to grid cell \( (i, j) \).

And the overall objective function is

\[
\log f^m = \sum_{k \in P^m} \log f_k^m
\]

The gradient and Hessian of objective function are calculated to optimize the displacement of the two robot poses, which will be deduced in “Range-based decomposition” section. Then, Newton method is used to optimize the transform parameter

\[
\Delta^m = -(H^m)^{-1} g^m
\]

Here, \( g^m \) is the gradient of overall objective function, and \( H^m \) is its Hessian matrix.

The inference above is NDT’s scan-matching procedure. It is ingenious to use normal distribution to represent the reference LIDAR frame. Thus, the optimizing step could utilize the normal probability density function to get the first-order and second-order derivatives.

But while using NDT method in real LIDAR data, it is found that the points should belong to one cluster according to their local feature, but they are split unsuitable by prefixed grid boundary, which make points aggregate to the wrong cluster.

As shown in Figure 2, the LIDAR frame should be divided into 20 local sets according to its local shape but resulted into 31 by the fixed interval division of NDT, where the unsuitable split is 35.5%.

As shown in Figure 3, the left figure illustrates the points are divided based on their local distributing feature, the set’s center is same as the points’ centroid. Then, as shown in the arrow of Figure 3(a), the new point will get good match with the cluster. But the ineligible division in Figure 3(b) will lead to unreasonable matching, where points are divided by the prefixed boundary, like NDT method, and the point after matching can’t link with the target cluster’s centroid. So, the precision of NDT is limited by the prefixed grid boundary, which can’t adapt to the local distribution’s variety.

The core thought of our method is motivated by the disadvantage of ineligible grid splitting of NDT. To deal with this problem, we compose a new point dividing method based on composite clustering, which can adapt to the local distribution of LIDAR points. In this way, the accuracy of scan registration is improved as shown in the “Experiment” section.

![Figure 3. Example of suitable (left) and unsuitable (right) splitting results. The orange dots are the LIDAR points in current frame, and the green and blue dots are the points in reference frame. Each arrow represents a matching result. And the boundary of grid cell is denoted as blue line. (a) Left: the points are divided based on their local distributing shape. (b) Right: the points are divided by prefixed boundaries, like NDT method, which can’t link to the centroid of cluster after matching. NDT: normal distribution transform.](image)

**Composite clustering scan registration**

**Algorithm overview**

The scheme of CCNDT method is introduced summarily, of which the core goal is to get the transform between two robot poses by their LIDAR and Odometer data. As shown in Figure 4, the whole process includes five steps:

1. It is started from the reference frame, where the inputted reference LIDAR data are first processed by the density clustering (see “Local distributing-based clustering” section). And the trigonometric-based distance method (see “Distance measuring method-based on trigonometric function” section) is used to measure the interval between each point. The reference LIDAR points are clustered by their local distributing shape.
2. The outputted clusters from (1) are filtered by SVD method to judge their eligible degree. Otherwise, the ineligible clusters will be further divided (see “Range-based decomposition” section).
3. After the filter process, the ineligible cluster will be divided into \( k \)’s small cluster by \( k \)-means method. The value of \( k \) is proportional to the ratio scale of this cluster’s singular values (see “Range-based decomposition” section).
4. Each clustering result will be given a bounding box for matching. Those points in current frame can be mapped into the bounding box by their relative coordination (see “Registration and objective function” section).
With the links from current frame’s points to reference frames’ clusters, the objective function is obtained by gathering each link’s normal distribution error. The transform between these two LIDAR frames is calculated by the objective function (see “Optimization and solving” section).

In the next several sections, we will walk through each of those steps in detail.

Local distributing-based clustering

The characteristic of LIDAR points should be analyzed firstly. LIDAR is a surveying sensor, as shown in Figure 5, which measures distance to a target by pulsing laser light and measuring the flying time from launching to receiving the reflected pulses by a laser receiver. LIDAR will record each measurement’s angle and distance when rotating. So, the obstacle in the laser illuminating path will be captured.

As shown in Figure 5, LIDAR points captured from the wall-like shape will form a straightline LIDAR distribution and from the cylinder shape will form an arc distribution. To make those points from same region clustered together and from different regions separated, it needs to get a method to identify the similarity and otherness among each object.

As the continuous of object boundary, the LIDAR point from the same obstacle is more continuous and has a higher reachable density than other areas. This condition is suitable for the density-based clustering named DBSCAN, which the points from the same cluster have a higher density than from the different clusters. DBSCAN provides a model named “Density Reachability,” which selects points satisfied the distance thresholds into one cluster. It is good at clustering the points in arbitrary distributing shapes.

In DBSCAN, $D_{m-1}$ is an initial cluster, and put into a father point $p_{x_{m-1}}$, which is randomly fetched from the reference frame $P_{m-1}$. Then it is going to find out all the points in $P_{m-1}$ satisfied with the limitation:

$$||p_{i_{m-1}} - p_{x_{m-1}} < \beta||$$

where $\beta$ is the distance threshold. In “Distance measuring method based on trigonometric function” section, we will introduce a new method to initialize and calculate this threshold.

And the points satisfied with the limitation are put into set $D_{m-1}$, meanwhile, removed from $P_{m-1}$. After ending a traversal round, the next turn starts with all the points from the last turn as the father points of this turn. Then, it continues searching until no more new point added in during a
new turn, as shown in Figure 6. So far, a density cluster \( D_{m-1}^n \) is formed.

For the left points in \( P_{m-1} \), the above iterating procedure will be repeated until emptying \( P_{m-1} \). Finally, a series of sets \( \{ D_{m-1} \} \) are constructed

\[
\{ D_{m-1} \} = \{ D_{1}^{m-1} D_{2}^{m-1} D_{3}^{m-1} \ldots D_{n}^{m-1} \} \quad (10)
\]

Each of those clusters has two characteristics. (1) The points in one cluster are density connected within threshold \( \beta \). (2) All of the density reachable points from one cluster must belong to this cluster.

If the points in one cluster are too few \( N_{D_{m-1}^i} < \zeta \), it will be improper for normal distributing fitting, where \( N_{D_{m-1}^i} \) is the number of points in cluster \( D_{m-1}^i \), and \( \zeta \) is an adjusting parameter depend on the resolution of LIDAR. When the LIDAR’s angular resolution is 0.25\(^\circ\), we set \( \zeta = 3 \). So, these clusters will be deleted and their points will be collected into a special cluster named \( D_{m-1}^0 \). Then the series of clusters become \( \{ D_{m-1} \} = \{ D_{0}^{m-1} D_{1}^{m-1} D_{2}^{m-1} \ldots D_{n}^{m-1} \} \).

Distance measuring method based on trigonometric function

After clustering LIDAR points by DBSCAN, we find a disappointed weakness. As the radiating phenomenon of LIDAR, the LIDAR points from the straight facet would not be uniformly distributed in a line. The LIDAR point captured on the same object from different distance will be sampled in different interval, as shown in Figure 7. The nearby interval is close and the further interval is distant. So, the threshold \( \beta \) is unsuitable to be a constant when measuring the points interval.

Here, we use a new method based on trigonometric function to measure the density reachability. When starting a new cluster, there is only one initial father point in the

\[
\text{Figure 6. Searching for the points belonging to one set with DBSCAN. The dash lines are the searching boundary. The red points belong to one density cluster. DBSCAN: density-based spatial clustering of applications with noise.}
\]

\[
\text{Figure 7. Radiation of the laser beam from a straight obstacle. The interval will be formed as the nearly points density and the further points sparsity.}
\]

\[
\text{Figure 8. The measurement after initializing, where} \ a \ \text{is the LIDAR center and} \ o' \ \text{is the pedal from LIDAR to the object's facet. Points} \ p_0, p_1 \ \text{are the father points obtained from initial step.} \ p_2 \ \text{is the next candidate point to be measured.} \ a \ \text{is the angle of} \ \angle p_0 p_1 a.
\]

\[
\beta = \rho_0 \sin \Delta \theta \quad \text{if initial} \quad \beta = \frac{\rho_1 \sin a}{\cos (\Delta \theta - a)} - \rho_1 \cos a \quad \text{if not initial}
\]

where \( \Delta \theta \) is the LIDAR’s angular resolution, and \( \rho_0 \) is the distance of father point, as shown in Figure 8.

After the initial procedure, the value of angle \( a \) can be obtained by the law of cosines with the initial points

\[
\cos a = \frac{\rho_0^2 + \rho_1^2 - \rho_{QP_1}^2}{2\rho_0 \rho_1} \quad (12)
\]

where the formula is based on the law of cosines, and \( \rho_{QP_1} = \rho_0, \rho_{QP_1} = \rho_1 \) is the edge of triangle \( \Delta p_0 p_1 p_2 \). Then, in triangle \( \Delta p_0 p_1 p_2 \) with two angles and one edge known, it is easy to calculate other parameters with the trigonometric function. We can get the distance method based on the trigonometric relation

\[
\beta = l_{p_1 p_2} = \frac{\rho_1 \sin a}{\cos (\Delta \theta - a)} - \rho_1 \cos a \quad (13)
\]

where \( \beta \) means that if the third point is collinear with the two initial points, the distance of \( p_1 p_2 \) should be \( l_{p_1 p_2} \) in theory, as shown in Figure 8.

Finally, the combined distance measuring threshold is

\[
\beta = \left\{ \begin{array}{ll}
\lambda \rho_0 \sin \Delta \theta & \text{if not initial} \\
\lambda \rho_1 \sin a - \rho_1 \cos a & \text{if initial}
\end{array} \right.
\]

\[
\lambda = 1 + \| Q_{\text{LIDAR}} \|_2
\]

where \( Q_{\text{LIDAR}} \) is the covariance of LIDAR’s accuracy, and the covariance can be obtained by calibrating the
Range-based decomposition

After DBSCAN processing, the continual regions of LIDAR data have been aggregated together, as shown in Figure 9. But on the one hand, as the larger cluster has a larger bounding box, so a point may be included by this bounding box but far from the cluster’s center, which will lead to heavy distorted links. On the other hand, the larger cluster has a strong attraction for its nearby points than a smaller cluster, which is because the weight of one cluster is in direct proportion to its quantity of points. And it will make the relaxing iteration unbalance. So, the result of DBSCAN is unsuitable for matching till now.

It needs to further decompose the larger cluster. The first question is how to select out the unsuitable clusters. It is noticed that such kind of clusters not only have a wider boundary but also have a big disparity of primary and secondary eigenvalues. So, each cluster in \( \{ \Sigma^{m-1} \} \) first be performed with SVD to get its singular values

\[
U \Sigma_{\text{singular}} V^T = \text{svd} \left( \Sigma^{m-1} \right)
\]

where \( \Sigma^{m-1} \) is the covariance of \( D^{m-1}_{i} \). And \( \Sigma_{\text{singular}} = \text{diag} \left( \sigma_1, \sigma_2, \sigma_3 \right) \) is the singular values of covariance \( \Sigma^{m-1} \).

The ratio from the biggest to the second singular value can denote the ineligible degree of one cluster.

\[
\xi = \frac{\sigma_1^2}{\sigma_2^2}
\]

where \( \sigma_1 \) and \( \sigma_2 \) are the primary and secondary singular values of the cluster’s covariance, respectively. \( \xi \) is the ratio factor. If the ratio factor is bigger than 2, it means the larger cluster should be divided into two smaller clusters at least. Otherwise, it doesn’t need to be divided. So, if \( \xi > 2 \), the cluster’s shape is disproportion and unsuitable for normal fitting. Then, it is necessary to get a further division for those unsuitable clusters.

The second question is how to further divide the unsuitable clusters. Those larger clusters can be divided by \( k\)-means method into \( k \) smaller clusters. \( k\)-Means makes their inner distance of one cluster minimum and their outer distance maximum. The unsupervised learning method of \( k\)-means can adaptively select the coordinate of initial points. And with the Euclidean distance, the shapes of clusters outputted from \( k\)-means are suitable for normal fitting.

And its most important parameter is the clustering quantity \( k \). Here, we use the previous SVD result to calculate the value of \( k \)

\[
k_i^{m-1} = \text{round}(\xi)
\]

where \( k_i^{m-1} \) is the clustering number of the \( i \)th cluster in the \( m-1 \) LIDAR frame. \( \xi \) is the ratio factor.

The result of \( k\)-means is

\[
\{ D^{m-1}_{i,1}, D^{m-1}_{i,2}, \ldots, D^{m-1}_{i,k} \} = \text{k-means}(D^{m-1}_{i}, k_i^{m-1}, \text{dis})
\]

where \( D^{m-1}_{i} \) is decomposed into \( k \) clusters \( \{ D^{m-1}_{i,1}, D^{m-1}_{i,2}, \ldots, D^{m-1}_{i,k} \} \) with Euclidean distance method \( \text{dis} = ||p_i - p_j|| \).

The cluster with few numbers of points also needs to be removed and puts into \( \{ D_{O}^{m-1} \} \) as mentioned in “Local distributing-based clustering” section. The cluster \( \{ D_{O}^{m-1} \} \) is useful when a point can’t match with any cluster, then the points in \( \{ D_{O}^{m-1} \} \) can be an optional candidate to build the matching relation.

As an illustration, we select the unsuitable clusters in Figure 9 by SVD ratio factor, then applying \( k\)-means method to divide those unsuitable clusters. The change from Figure 9 to Figure 10 demonstrates that the unsuitable clusters are decomposed into smaller clique by \( k\)-means. Each of the clusters as the box marked in Figure 10 is more balance in each direction, which is more suitable for normal distribution fitting.

Till now, there are two groups of the cluster: the first group is the suitable results of DBSCAN and the second group is those unsuitable clusters decomposed by \( k\)-means. Then reorder them as...
unqualified clusters are further divided by k-means. Those clusters are first selected by SVD-based method, then the unqualified clusters are further divided by k-means. SVD: singular value decomposition.

\[ \{D_{m}^{m-1}, D_{m+1}^{m-1}, \ldots, D_{c+(n-1)}^{m-1}\} \]

\[ = \text{reorder}\{D_{i,1}^{m-1}, D_{i,2}^{m-1}, \ldots, D_{i,n}^{m-1}\} \]

(19)

where \( c \) is the number of elements in \( \{D^{m-1}\} \), which means the first cluster \( D_{i,1}^{m-1} \) replaces its parent index \( D_{i}^{m-1} \), and the rest clusters \( [2, 3 \ldots n] \) are added after the latest index \( c \).

For uniform, those two groups of clusters are integrated together and denoted as

\[ \{\tilde{D}^{m-1}\} = \{D_{1}^{m-1}, D_{2}^{m-1}, D_{3}^{m-1}, \ldots, D_{n}^{m-1}\} \]

(20)

This step just changes the index of the clusters for their lower right corner.

**Registration and objective function**

After obtaining the proper clusters, here it is going to infer their matching parameters. As the origin NDT method inferred in “Problem statement” section, the normal distributing parameter of each cluster includes its mean and covariance. The mean of cluster \( D_{i}^{m-1} \) is

\[ i_{z}^{m-1} = \frac{1}{n} \sum_{k=1}^{n} i_{k}^{m-1} \]

(21)

where \( \{i_{k}^{m-1}\}_{k=1,2,\ldots,n} \in D_{i}^{m-1} \), and \( i \) is the index of cluster. And its covariance is

\[ i_{z}^{m} = \frac{1}{n} \sum_{k=1}^{n} i_{k}^{m}(i_{k}^{m})^{T} \]

(22)

When building the matching relation, the original NDT method uses grid cells to link the points and the clusters. But in CCNDT, the cluster’s size and position are not regular, so here we use the bounding box as shown in Figure 11, for searching.

\[ \alpha = \frac{2 \times \text{area}(\tilde{D}_{i}^{m-1})}{\text{quantity}(\tilde{D}_{i}^{m-1})} \]

(23)

\[ \text{area}(\tilde{D}_{i}^{m-1}) = \left( \max_{y} \tilde{D}_{i}^{m-1} - \min_{y} \tilde{D}_{i}^{m-1} \right) \]

\[ \times \left( \max_{x} \tilde{D}_{i}^{m-1} - \min_{x} \tilde{D}_{i}^{m-1} \right) \]

(24)

where area \( (\tilde{D}_{i}^{m-1}) \) is the area of this cluster and quantity \( (\tilde{D}_{i}^{m-1}) \) is the number of points in this cluster.

For each cluster \( \tilde{D}_{i}^{m-1} \) in \( \{\tilde{D}^{m-1}\} \), its bounding box is denoted as

\[ \mathbb{B}_{j} = \begin{bmatrix} \min_{x} \tilde{D}_{i}^{m-1} - \alpha, \min_{y} \tilde{D}_{i}^{m-1} - \alpha \ 
\max_{x} \tilde{D}_{i}^{m-1} + \alpha, \max_{y} \tilde{D}_{i}^{m-1} + \alpha \end{bmatrix} \]

(25)

The bounding box \( \mathbb{B}_{j} \) is denoted by two points’ coordinates, the upper left corner and the lower right corner.

To judge whether a point belongs to this cluster, it just needs to compare the point’s coordinate with the bounding box’s two corners.

Till now, the relation between points in current frame and the clusters with bounding box in reference frame are constructed.

When the new LIDAR frame \( P_{m} \) obtained, named current frame, we need to match each point in \( P_{m} \) to their relevant cluster in \( \{\tilde{D}^{m-1}\} \). Each bounding box is traversed to
To solve the objective function, the Jacobin is the error from the Liu et al. sum, the inner summary layer is the errors of one point to its relevant cluster which is the partial derivative of the error function with respect to each element of $\Delta^m$.

The Hessian matrix is the second derivation of objective function

$$H^m_{ij} = \frac{\partial^2 f^m_i}{\partial \Delta_i \partial \Delta_j^m}$$

where $l$ and $r$ are the point indexes of the first-round derivation and second-round derivation, respectively.

With objective function substituted in, Hessian is reshaped as

$$H^m_{ij} = \exp \left(-\frac{(\varepsilon^m_i)^T \Sigma^{-1} (\varepsilon^m_i)}{2} \right)$$

$$\frac{\partial (\varepsilon^m_i)^T \Sigma^{-1} (\varepsilon^m_i)}{\partial \Delta_i} \left(\frac{\partial (\varepsilon^m_i)^T}{\partial \Delta^m} - \frac{\partial \Sigma^{-1}}{\partial r} \frac{\partial r}{\partial \Delta^m} \right)$$

$$\frac{\partial (\varepsilon^m_i)^T \Sigma^{-1} (\varepsilon^m_i)}{\partial \Delta^m} = \frac{\partial (\varepsilon^m_i)^T \Sigma^{-1} (\varepsilon^m_i)}{\partial \Delta_i} \left(\frac{\partial (\varepsilon^m_i)^T}{\partial \Delta^m} - \frac{\partial \Sigma^{-1}}{\partial r} \frac{\partial r}{\partial \Delta^m} \right)$$

In which the second derivative of $\varepsilon^m_i$ for $\Delta$ is equivalent to the derivative of $J^m$ for $\Delta$.

$$\frac{\partial J^m_i}{\partial \Delta^m} = \frac{\partial (\varepsilon^m_i)^T \Sigma^{-1} (\varepsilon^m_i)}{\partial \Delta^m}$$

$$\frac{\partial J^m_i}{\partial \Delta^m} = \left\{ \begin{array}{ll}
m_{(5,3)} & = -x_\Delta \cos \theta_\Delta + y_\Delta \sin \theta_\Delta \\
m_{(6,3)} & = -x_\Delta \cos \theta_\Delta + y_\Delta \sin \theta_\Delta \\
others & = 0
\end{array} \right.$$
where the denominator is the actual improvement, which is calculated by the second-order Taylor expansion of object function $f^m$. The numerator is the expected improvement, where $f^m_{P^m + \Delta^m}$ and $f^m_{P^m}$ are the results of objective function at $P^m + \Delta^m$ and $P^m$, respectively. $J^m_{P^m}$ and $H^m_{P^m}$ are the Jacobin and Hessian of objective function $f^m$, respectively.

With the fitting condition $\rho$, the forward step is larger in the well-fitting place and shorter in the worse. So, with the trust region-based method, the fitting is more flexible and adaptable.

Experiments

To evaluate the result of our method in real robot data, we use public data sets to test the mapping quality of our method. The first data set is from Intel Research Lab in Seattle, which was recorded by a Pioneer II robot equipped with a SICK sensor and an odometer. The area of this data set is about $26 \times 26$ m$^2$ and the robot traveled 506 m. The second data set was from the ACES building at the University of Texas provided by Patrick Beeson, which includes 13,632 LIDAR frames.

To illustrate the final mapping results, we use the Karto SLAM as a framework, in which the scan-matching parts are replaced by our CCNDT method to calculate the transform between two neighbor frames. It first takes the odometer data as the initial guess. Then, the output transform is aggregated into the Karto SLAM to generate the occupied grid map, as shown in Figure 12, which is deployed in ROS Kinetic environment. In addition, as the scale of the environment is very large and including many trajectory loops, the Karto SLAM is also used for loop closure matching and global optimization with sparse pose adjustment (SPA) method.

From the mapping result in Figures 12 and 13, it is clear that the maps built by CCNDT have high-quality and clear edges. Even in large environment, the mapping result still has very little distorted, which partly is contributed by Karto SLAM’s loop closure and SPA optimization.

As there is no absolute ground truth in Intel Research Lab and ACES data sets, Burgard provide a manual calibrating method to align the robot path with LIDAR frames, which perform a pair-wise registration to refine the estimates. This calibrated data were accessible online.

Then, we use three different methods, NDT, ICP and CCNDT, to process the LIDAR data set frame by frame. The output results are the transforms of each neighbor frame. They are compared by the metrical benchmark method with the manually calibrated data. As the time stamps of scan matching and manual calibration are not one-to-one matched, so the calibrating series are interpolated to be synchronic with the scan-matching series, which will be regarded as the ground truth

$$\varepsilon = ||f'(\Delta_m, \Delta_{inter})||_2$$

where $\Delta_m$ is the scan-matching result, $\Delta_{inter}$ is the manually calibrated transform sourcing from Kümmerle et al.,
and \( e \) is the error between \( D_{m} \) and \( D_{inter} \). \( T(a, b) \) is a transform operator as equation (4).

The comparing results of each method are shown in Figure 14. After statistical analysis for these error results from Figure 14, we got the mean and covariance of the error results in Table 1.

From Table 1, the mean error of CCNDT is 0.1879 m, which is lower than NDT 0.2726 m and ICP 0.2619 m. The sum error sources from three parts: the manually calibrating error, the scan-matching error, and the linear interpolating error by metrical benchmark method. Besides, the covariance of the CCNDT is 0.0295 m², which is lower than the other two methods. Experimental result demonstrates that our method is more accurate and more stable than NDT and ICP methods. Comparing the speed, CCNDT method processes the total data set costing 2791 s, which is obviously slower than NDT and ICP. This is because CCNDT method spends too much time in clustering procedure.

Comparing the speed, CCNDT method processes the total data set costing 2791 s, which is obviously slower than NDT and ICP. This is because CCNDT method spends too much time in clustering procedure. In terms of matching speed, CCNDT method is not suitable for the robot with lower computing capability, but if the computing capability is sufficient, it is recommended taking CCNDT to get a better matching quality.

**Conclusion**

In this article, we compose a novel scan registration method, CCNDT, which based on normal distributing transforming method. In CCNDT, a composite clustering method, combining DBSCAN and \( k \)-means, is used to get the suitable splitting of LIDAR data. This method can adapt to the local distribution of LIDAR points. Meanwhile, we come up with a new distance measuring method based on trigonometric function, which can avoid the radiating distribution of LIDAR points.

We test CCNDT method with the public data set from Intel Research Lab and ACES building at the University of Texas. The experimental results indicate that CCNDT is more accurate and more stable than NDT and ICP methods. And the point clusters of CCNDT could ensure the continuity of the local region and get the eligible matching from points to clusters.

In our future work, we are going to deploy the proposed CCNDT method on the scan-to-map registration, because the CCNDT can express the feature of local point distribution and decrease the number of reference points by replacing them with clusters.

Table 1. Statistical error of scan-matching result.

| Data source | Matching pairs | Mean (m) | Covariance (m x m) | Total Time (s) |
|-------------|----------------|---------|--------------------|----------------|
| CCNDT Intel | 1774           | 0.1879  | 0.0295             | 2791           |
| NDT Intel   | 1774           | 0.2726  | 0.0634             | 653            |
| ICP Intel   | 1774           | 0.2619  | 0.1049             | 961            |

**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 supported by The General Program of National Natural Science Foundation of China [Grant No. 51775202]. This work is also funded by the Graduates Innovation Fund, Huazhong University of Science and Technology [No. 2019ygsxscy078].

**ORCID iD**

Tian Liu https://orcid.org/0000-0002-0131-4584
Jiongzhli Zheng https://orcid.org/0000-0001-7201-6720

**References**

1. Khairuddin AR, Talib MS, and Haron H. Review on simultaneous localization and mapping (SLAM). In: 2015 IEEE international conference on control system, computing and engineering, George Town, Malaysia, 27–29 November 2015, pp. 85–90.
2. Dunbabin M and Marques L. Robots for environmental monitoring: significant advancements and applications. IEEE Robot Autom Mag 2012; 19(1): 24–39.
3. Chen W and Zhang T. An indoor mobile robot navigation technique using odometry and electronic compass. Int J Adv Robot Syst 2017; 14(3): 1–15.
4. Besl PJ and McKay ND. Method for registration of 3-D shapes. In: Proceedings Volume 1611, Sensor Fusion IV: Control Paradigms and Data Structures, Vol. 1611, Boston, MA, United States, 30 April 1992, pp. 586–607

5. Biber P and Strasser W. The normal distributions transform: a new approach to laser scan matching. In: Proceeding 2003 IEEE/RSJ international conference on intelligent robots and systems (IROS 2003) (Cat. No.03CH37453), Vol. 3, Las Vegas, NV, USA, 27–31 October 2003, pp. 2743–2748.

6. Siegwart R, Nourbakhsh IR, and Scaramuzza D. Introduction to autonomous mobile robots. Cambridge: MIT Press, 2011.

7. Segal A, Haehnel D, and Thrun S. Generalized-ICP. Robot: Sci Syst 2009; 2(4): 435.

8. Serafin J and Grisetti G. Using extended measurements and scene merging for efficient and robust point cloud registration. Rob Auton Syst 2017; 92: 91–106.

9. Serafin J and Grisetti G. NICP: dense normal based point cloud registration. In: International conference on intelligent robots and systems, Hamburg, Germany, 28 September–2 October 2015, pp. 742–749.

10. Serafin J and Grisetti G. Using augmented measurements to improve the convergence of ICP. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 2014; 8810: 566–577.

11. Kölser J, Simonetti A, Sylvester PJ, et al. Laser-ablation ICP–MS measurements of Re/Os in molybdenite and implications for Re–Os geochronology. Can Mineral 2003; 41(2): 307–320.

12. Feng L and Milios E. Robot pose estimation in unknown environments by matching 2D range scans. In: Proceeding IEEE conference computer vis pattern recognition CVPR-94, Seattle, WA, USA, 21–23 June 1994, pp. 935–938.

13. Wang J, Li L, Zhe L, et al. Multilayer matching SLAM for large-scale and spacious environments. Int J Adv Robot Syst 2015; 12(9): 1–11.

14. Minguéz J, Lamiraux F, and Montesano L. Metric-based scan matching algorithms for mobile robot displacement estimation. In: Proceedings of the 2005 IEEE international conference on robotics and automation, Barcelona, Spain, 18–22 April 2005, pp. 3557–3563.

15. Pomerleau F, Colas F, and Siegwart R. A review of point cloud registration algorithms for mobile robotics. Found Trends® Robot 2015; 4(1): 1–104.

16. Bosse M, Newman P, Leonard J, et al. Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework. Int J Rob Res 2004; 23(12): 1113–1139.

17. Gutmann JS and Konolige K. Incremental mapping of large cyclic environments. In: Proceedings 1999 IEEE international symposium on computational intelligence in robotics and automation. CIRA’99 (Cat. No.99EX3735), Vol. 99, Monterey, CA, USA, 8–9 November 1999, pp. 318–325. IEEE.

18. Mitra NJ, Gelfand N, Pottmann H, et al. Registration of point cloud data from a geometric optimization perspective. In: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on geometry processing, Nice France, July 2004, pp. 22–31.

19. Ulas C and Temeltas H. A fast and robust feature-based scan-matching method in 3D slam and the effect of sampling strategies. Int J Adv Robot Syst 2013; 10: 1–16.

20. Ryu K, Dantaranrayana L, Furukawa T, et al. Grid-based scan-to-map matching for accurate 2D map building. J Adv Robotics 2016; 30: 431–448.

21. Wang A, Li C, Liu Y, et al. Laser-based online sliding-window approach for UAV loop-closure detection in urban environments. Int J Adv Robot Syst 2016; 13(2): 1–11.

22. Gouveia BD, Portugal D, and Marques L. Speeding up Rao–Blackwellized particle filter SLAM with a multi-threaded architecture. In: IEEE international conference on intelligent robots and systems, Chicago, IL, USA, 14–18 September 2014, pp. 1583–1588. IEEE.

23. Newcombe RA, Izadi S, Hilliges O, et al. Kinect fusion: real-time dense surface mapping and tracking. In: 2011 10th IEEE international symposium on mixed and augmented reality (ISMAR), Vol. 11, Basel, Switzerland, 26–29 October 2011, pp. 127–136.

24. Thrun S, Burgard W, and Fox D. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In: Proceedings of the 2000 IEEE international conference on robotics and automation, Vol. 1, San Francisco, CA, USA, 24–28 April 2000, pp. 321–328.

25. Ester M, Kriegel HP, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the second international conference on knowledge discovery and data mining (KDD-96), Portland, Oregon, 2–4 August 1996, Vol. 96, no. 34, pp. 226–231.

26. Celebi ME, Kingravi HA, and Vela PA. A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Syst Appl 2013; 40(1): 200–210.

27. Yuan Y. A review of trust region algorithms for optimization. In: ICIAM, Edinburgh, Scotland, 1999, Vol. 99, no. 1, 2000, pp. 271–282.

28. Stachniss C, Frese U, and Grisetti G. Openslam.Org. 2007.

29. Kleinm A, Steder B, Dornhege C, et al. Slam benchmark. 2009. http://ais.informatik.uni-freiburg.de/slamevaluation/index.php (accessed 3 June 2019).

30. Ferguson M and Bettaieb L. Karto slam. http://wiki.ros.org/slam_karto (12 July 2019).

31. Konolige K, Grisetti G, Kümmel R, et al. Sparse pose adjustment for 2D mapping. In: 2010 IEEE/RSJ international conference on intelligent robots and systems, Taipei, Taiwan, 18–22 October 2010, pp. 22–29.

32. Burgard W, Stachniss C, Grisetti G, et al. A comparison of SLAM algorithms based on a graph of relations. In: 2009 IEEE/RSJ international conference on intelligent robots and systems, St. Louis, MO, USA, 10–15 October 2009, pp. 2089–2095.

33. Kümmel R, Steder B, Dornhege C, et al. On measuring the accuracy of SLAM algorithms. Auton Robots 2009; 27(4): 387–407.