



Room-level localization method in industrial workshops using LiDAR-based point cloud registration and object recognition

Yunzhi Li¹ · Libin Tan¹ · Xiangrong Xu¹ · Zequn Zhang²

Accepted: 30 December 2024 / Published online: 24 January 2025

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2025

Abstract

In this work, we aim to achieve room-level localization for mobile robots in industrial workshops. It is difficult to obtain precise localization information via common methods because of the complexity of the industrial environment. Our findings show that precise room-level localization can be achieved via LiDAR-based point cloud registration and object recognition. For this purpose, we formulate room-level localization as a classification problem. Registration and object recognition are used to extract features from point clouds. After the data enhancement algorithm, called Stacked Auto Encoder is employed to overcome the issue of limited feature data, the neural network algorithm is leveraged to address the classification problem. To this end, we collected point cloud data from industrial workshops and performed experimental validation. We evaluated the recognition performance of the algorithm in a metallurgical workshop and achieved good accuracy.

Keywords Mobile robot · Room-level localization · Registration · Object recognition · Data enhancement · Neural network algorithm

1 Introduction

Autonomous localization is an active field in the realm of mobile robots and has garnered significant attention from researchers [1]. The objective of localization is to accurately determine the position of the mobile robots [2]. Matching recent sensory information with prior information about the environment is a common approach for robust localization [3]. Odometry based on machine vision [4] and LiDAR [5] is a popular way to address localization problems.

In various scenarios, such as apartments, offices, or industrial workshops, the implementation of room-level indoor positioning enables robots to offer customized services. The room-level position refers to the specific position of the robot in one or more enclosed areas determined by sensors, control systems, and external positioning devices. Several key points are listed to illustrate the necessity of room-level location.

First, room-level localization can serve as a foundation for achieving precise localization. Then, the occurrence of errors when the robot is located in similar rooms in the given scenario is reduced. Additionally, various navigation strategies can be employed based on the specific environment in which the robot is located.

The room-level localization method has been investigated by various scholars. A room-level fingerprint localization method based on the RSSI, which is a threshold algorithm based on receiver operating characteristic analysis, was proposed [6]. The actual location of the user is estimated by the exchange of people and devices. A navigation and room-level localization system called BFound was proposed [7]. Room-level localization is realized by wavelet transformation across the unique signature of the room generated by the beacon sensor. A new occupancy schedule modeling system in a building information modeling environment called stations-oriented indoor localization was presented, which was able to detect indoor occupants' room-level locations with high accuracy [8]. A counterpart indoor tag trajectory reasoning data structure OPTR-tree is proposed, which provides an approach for multi-tag track fast recognition in an indoor environment at room-level accuracy [9]. These localization methods are based on preset devices in the environment. Positioning is not possible in environments where there

✉ Libin Tan
alexltb@ahut.edu.cn

¹ Anhui University of Technology, Maxiang Road,
243032 Ma'anshan, Anhui Province, China

² Nanjing University of Aeronautics and Astronautics, 29
Yudao Street, 246001 Nanjing, Jiangsu Province, China

are no interacting devices. In simple environments such as apartments or offices, a sonar sensor with low cost and low power consumption is an option [3]. An automatic room-level localization approach resolves three key problems:

- The collection of fingerprints.
- Determination of fingerprints' location information.
- Energy consumption [10].

However, these methods are not suitable for places where many rooms have complex characteristics.

To date, robot localization methods based on cameras and radar have also been extensively studied. Image-based localization methods [4, 11, 12] have been extensively investigated and implemented in a wide range of fields. Nevertheless, visual sensors are prone to malfunction under varying light or shadow conditions. Recent studies on LiDAR-based localization can be classified into two categories:

- Retrieval-based methods.
- Registration-based methods

Retrieval-based methods [13, 14] conceptualize localization as a dedicated task of searching for a specific location. The general process involves obtaining global descriptors through certain rules and searching for the corresponding point clouds in global databases. Registration-based methods [15, 16] solve localization problems through the matching of point clouds. Once point clouds successfully match a corresponding part in the database, the LiDAR position can be accurately estimated. However, in a large environment, it may not position accurately, as the matching process demands extensive computations to compare the query point cloud with every part of the database.

Owing to the inadequacy of traditional positioning methods, in this paper, a robotic localization method is proposed based on LiDAR-based point cloud registration and object recognition. We turn the room location problem into a classification problem. The feature data of point cloud registration and recognition are extracted, and the amount of data is increased through data enhancement. Then, a neural network algorithm is adopted for classification according to the feature data. Once the room is determined, the foundation is laid for further accurate positioning.

The subsequent sections of this paper are structured as follows. Section 2 provides an overview of the relevant literature on localization. Section 3 presents the proposed model and method for room-level localization. This method includes two methods for extracting point cloud features and a decision method based on a neural network. In Section 4, the acquisition of datasets and subsequent experimental validations are conducted. Section 5 presents the conclusion and prospects.

2 Related work

This method is closely related to the point cloud localization of mobile robots. Matching real-time sensor information with prior information about the environment is a common method for robot positioning. Previous works on point cloud localization can be divided into retrieval-based localization and registration-based localization. In this section, the representative works for point cloud localization are briefly reviewed and discussed.

2.1 Retrieval-based localization

Lidar-based localization is frequently regarded as point cloud retrieval, where the location of the point cloud is estimated from existing data. Object recognition is one of the core tasks in machine vision. Considerable progress has been made in 2D object detection on the basis of images in recent years [17]. Hence, object recognition based on flat cameras has difficulty playing a role in poor lighting conditions. Unlike 2D detection, which is solely based on flat camera image data, sensors with spatial sensing ability are beneficial for recognizing objects in 3D space. The 3D point cloud data are precise and have high spatial resolution because of the directivity and energy concentration of the laser [18]. A hierarchical registration algorithm [19] is presented to align unordered point clouds on the basis of a context descriptor. A new type of feature descriptor that contains a local viewpoint feature histogram (LVFH) and a local ensemble of shape function (LESF), was proposed in [20]. However, these methods are not appropriate for complex environments.

With the amazing achievements of deep learning, some studies have utilized neural networks to replace traditional descriptors. ShapeNet [21] is designed for object recognition, but it can work only on small-scale datasets. An object classification method that is based on a convolutional neural network (CNN) and image upsampling theory was presented in [22] for vision and light detection and ranging (LiDAR) fusion. PointNet [23] was proposed to process the point cloud directly and has been the network architecture basis for many subsequent methods. Existing leading approaches, such as PointNetVLAD [24] and PCAN [25], have obtained superior results. A Multi-view Semantic Learning Network (MVSLN) for 3D object detection was proposed, which considered the feature discrimination for LiDAR point cloud [26]. An advanced object detection model building on YOLOv5, called YOLO-TLA, was proposed [27]. The YOLO-TLA model has efficient and effective performance in small object detection, achieving high accuracy with fewer

parameters and computational demands. A monocular 3D object detection method that formulates the 3D object localization as a paired keypoints regression problem was presented [28]. Two modules for object detection works by more detailed voxel initial information extraction and full fusion of context information was proposed [29]. The retrieval methods can detect coarse location recognition when the point cloud feature is extracted.

2.2 Registration-based localization

Point cloud registration is one of the most important topics in mobile laser scanning (MLS) and terrestrial laser scanning (TLS) [30, 31]. The aim is to obtain the transformation parameters to transform captured point clouds to their correct positions. LiDAR-based localization often requires a precise 6-DoF pose, which motivated the development of localization methods based on point cloud registration. Most point cloud registration methods employ a coarse-to-fine strategy [32]. The structure-based methods employ feature descriptors such as spin images [33], point feature histograms [34], and fast point feature histograms [35]. A lightweight and fast point cloud registration method was proposed [36]. It achieved initial point cloud registration by preserving the curvature characteristics of the internal point cloud and utilizing the spatial relationship between grid cells and the quantitative relationship between the internal point cloud. A registration strategy [37] was presented on the basis of a planar primitive group for indoor environments between a large-scale point cloud and a small-scale point cloud, providing a localization solution that is fully independent of prior information about the initial positions of the two point cloud coordinate systems. A registration method [38] using a general feature extraction operator in the spectral domain was proposed, in which the spectrum used a variety of wavelet transforms. A place recognition and navigation method [39] was presented based on the random forest algorithm for outdoor mobile robots with three-dimensional (3D) laser point cloud data.

On the basis of the obtained feature vectors, a 3D–3D correspondence relationship was established via feature matching in RANSAC. Coarse registration algorithms were first applied to calculate the orientation of approximately six degrees of freedom and the translation parameters between adjacent point clouds [40]. Instead of the features, the iterative closest-point algorithm [41] was used to estimate point correspondences via least-squares optimization, which is the most commonly used method for pairwise fine registration. In recent years, methods based on deep learning have achieved superior performance. A deep feature correlation learning network (CNet) for multimodal remote sensing image registration was proposed [42]. Reference [43] proposed a novel remote sensing image registration method

based on a deep learning regression network. Recently, some studies have combined these two approaches for localization. PR–NET described a coarse-to-fine workflow to localize tampered regions progressively [44].

We review research on point cloud localization from two aspects:

Retrieval-based localization.

Registration-based localization.

However, most of these methods are based on research conducted on one of these aspects, and based on depth images, 2D point clouds are more extensively studied than 3D point clouds. The LiDAR-based localization method proposed in this paper combines two-directional algorithm-based and registration-based features from 3D point clouds for positioning.

3 Proposed model

In this paper, a robotic localization method is proposed based on LiDAR-based point cloud registration and object recognition. The primary illustration is depicted in Fig. 1. The robot is located in several positions and acquires multiple sets of original point cloud data. The features extracted from the data are used to determine their location.

In this section, the framework of the method is shown in Fig. 2. It contains three main components:

Point cloud registration.

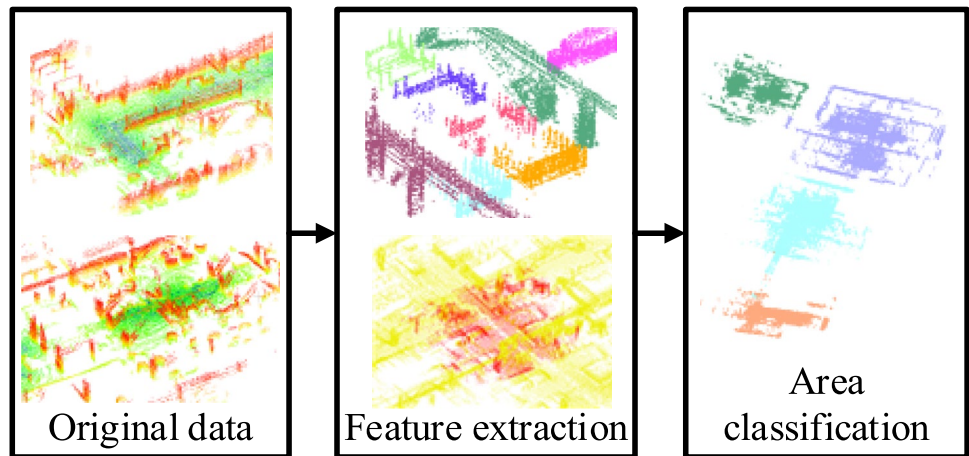
Point cloud object recognition.

Area classification.

In detail, through coarse and fine registration between current point clouds and global point clouds, the overlap rates of the point cloud can be acquired. Next, the current point clouds are segmented and clustered, and the number of identified objects is determined. Data enhancement is performed sequentially after the data from the two methods mentioned above are combined. Finally, the obtained data are classified via a neural network algorithm to determine the position of the robot. A detailed description of these methods is provided in Fig. 2.

Several key points are listed to illustrate the necessity of the proposed framework. On the one hand, robots can match real-time maps with global maps for localization. Matching recent data of point clouds with prior data of the environment is an approach for finding connections between parts and the whole. On the other hand, robots determine where they are based on objects in the environment. Matching recent object information of point clouds with prior object information of the environment is another way to find

Fig. 1 Principle illustration of the problem. Raw point clouds of robots at different positions are collected, and their features are extracted to determine the room-level locations of the robots



connections between parts and the whole. The information obtained by combining the two methods is used as the basis for localization. The neural network algorithm may be a suitable method for classifying the above mentioned data. Considering that neural network algorithms require a large amount of raw data, data enhancement is employed to augment the amount of available data.

3.1 Point cloud registration

To acquire location information, it is necessary to establish a correlation between current point clouds and global point clouds.

The current point clouds are 3D point clouds of the current location collected by the robot. The global point clouds are the overall 3D point clouds within which the robot can move. Point cloud registration facilitates the determination of positional relationships between these two entities.

3.1.1 Coarse registration

After the current point clouds are acquired from the LiDAR system, the first step is to extract feature points for registration. For a large number of point cloud data points, the correlation between current point clouds and global point clouds is established according to the features extracted, which improves efficiency and accuracy.

Intrinsic shape signature (ISS) A method called the intrinsic shape signature (ISS) describes the geometry of point clouds. The algorithm is a regional point cloud feature extraction method that measures the importance of points by calculating the curvature and normal change in point cloud data. This method enables the acquisition of various geometric features, which can be used for registration. The point cloud P contains i points $p_i(x_i, y_i, z_i)$. The coordinate system is established for each point p_i , and the search radius

r_{frame} is set. p_i is considered the centre to select all the points with radius R , and the weight w_{ij} is calculated.

$$w_{ij} = 1 / \left(\|p_i - p_j\|, |p_i - p_j| < r_{frame} \right) \quad (1)$$

Then, the covariance matrix for each point in the range

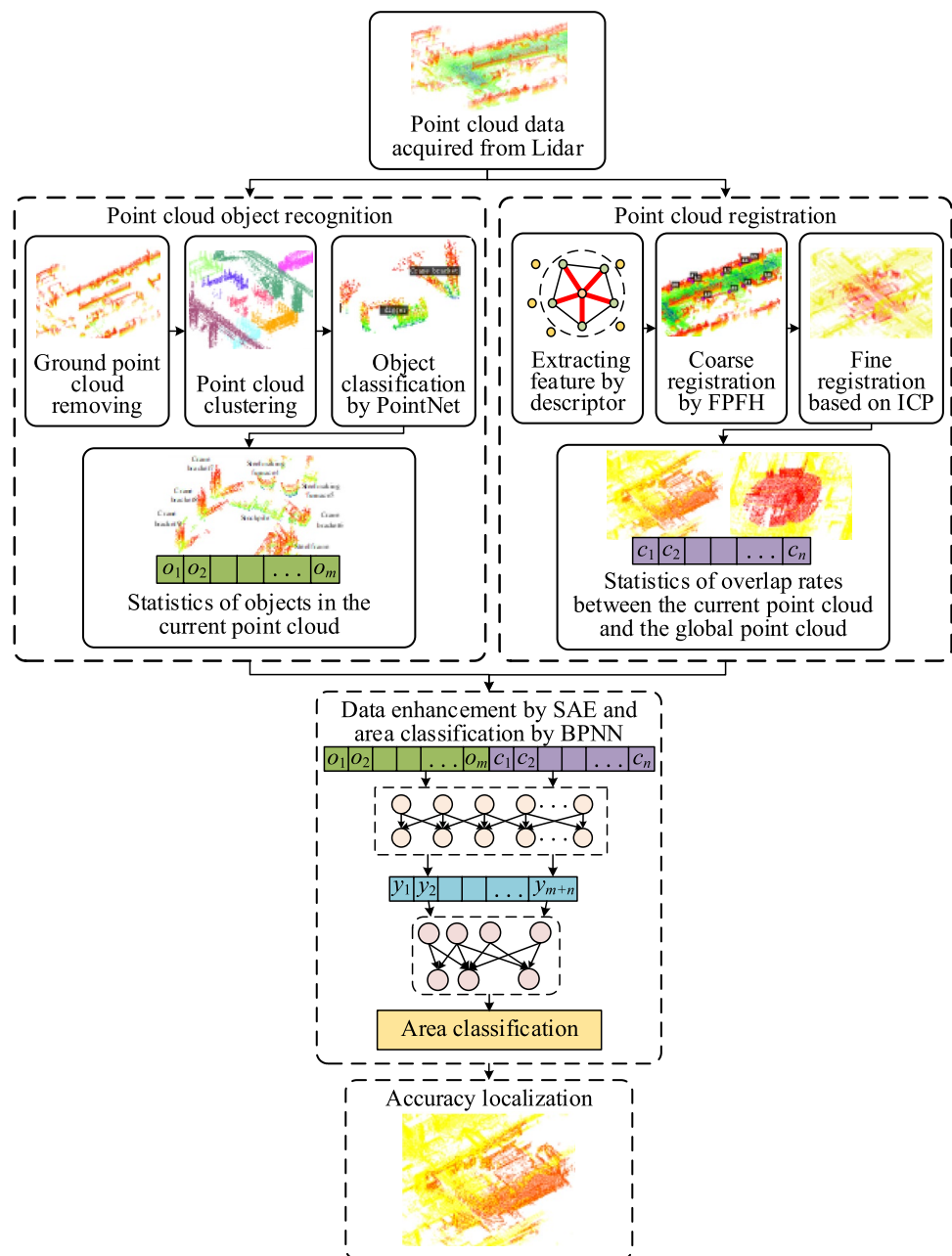
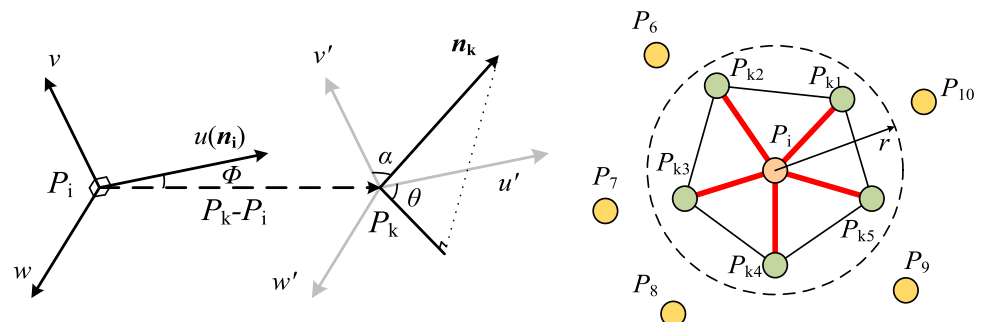
$$\text{cov}(p_i) = \sum_{|p_i - p_j| < r_{frame}} w_{ij} (p_i - p_j)(p_i - p_j)^T / \sum_{|p_i - p_j| < r_{frame}} w_{ij} \quad (2)$$

The eigenvalues of $\text{cov}(p_i)$ are calculated from large to small as $\{\lambda_i^1, \lambda_i^2, \lambda_i^3\}$. Thresholds ε_1 and ε_2 are set. The points in the point cloud P that meet the conditions $\lambda_i^2 / \lambda_i^1 \leq \varepsilon_1$ and $\lambda_i^3 / \lambda_i^2 \leq \varepsilon_2$ are the required key points.

Fast point feature histograms Fast point feature histograms (FPFHs), which are generally used to describe the geometric features around a point, are subsequently employed to acquire feature point descriptors. By parameterizing the spatial differences between feature points and adjacent points, point feature histograms (PFHs) form a multidimensional histogram to geometrically describe the nearest neighbours of points. Domain points within radius r are searched for feature point P_i , which is defined as P_k . The local coordinate system of P_i is established as follows:

$$\begin{cases} d = \|p_k - p_i\|_2 \\ \mathbf{u} = \mathbf{n}_i \\ \mathbf{v} = \mathbf{u} \times (p_k - p_i) / d \\ \mathbf{w} = \mathbf{u} \times \mathbf{v} \end{cases} \quad (3)$$

As shown in Fig. 3, d is the Euclidean distance between two points. Both $\mathbf{n}_i$ and $\mathbf{n}_k$ are unit normal vectors of points P_i and P_k .

Fig. 2 Framework of the position recognition method**Fig. 3** Coordinate system parameters of feature points

The element characteristics of P_i and each of its domain points are calculated.

$$\begin{cases} \alpha = \mathbf{v} \cdot \mathbf{n}_k \\ \phi = \mathbf{u} \cdot (\mathbf{p}_k - \mathbf{p}_i) / d \\ \theta = \arctan(\mathbf{w} \cdot \mathbf{n}_k, \mathbf{u} \cdot \mathbf{n}_k) \end{cases} \quad (4)$$

The above feature element values of point P_i and every point P_k in its k neighbourhood are calculated and statistically obtained as a simplified point feature histogram (SPFH), which can be denoted $\text{SPFH}(P_i)$.

Then, the SPFH of each point in the k neighbourhood of P_k is calculated separately, and all the SPFHs are weighted statistically. The FPFH reduces the computational complexity of the algorithm but still retains most of the identification characteristics of the PFH. Therefore, the FPFH consists of two parts, and the formula is as follows.

$$\text{FPFH}(P_i) = \text{SPFH}(P_i) + \frac{1}{k} \sum_{m=1}^k \frac{1}{d_k} \cdot \text{SPFH}(P_k) \quad (5)$$

$\text{SPFH}(P_k)$ is the feature of P_k and each of its domain points. d_k is the distance between P_i and P_k . As shown in Fig. 4, the FPFH involves not only the relationships between feature points and points within radius r but also the relationships between points within radius r and additional point pairs outside it.

Three sets of features are statistically input into one histogram. All three features are angles and are related to normal vectors, which can be placed in the same interval. The value of each feature is divided into several subintervals according to the range. The number of points falling in each subinterval is counted, which is the histogram of the point feature.

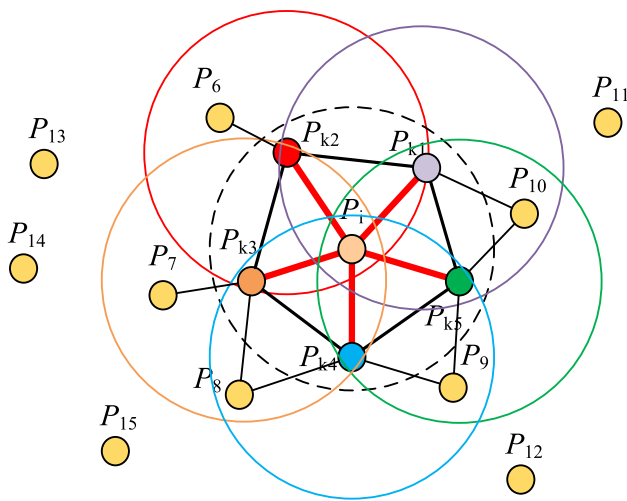


Fig. 4 FPFH of the feature points

The feature points of the current point clouds are matched with the points of the global point clouds to find the closest point pair via the FPFH, which is the corresponding point.

Random sample consensus All the above methods are aimed at finding the corresponding point between two clouds. Next, random sample consensus (RANSAC) can achieve the coarse registration of point clouds. The algorithm is an iterative algorithm for robustly estimating model parameters from a dataset. In terms of point cloud processing, RANSAC can be used to find matching points for a pair of point clouds.

Three non-collinear points are randomly selected from the feature points of the current point cloud and denoted (p_1, p_2, p_3) . The global point cloud is defined as Q , and the corresponding points of (p_1, p_2, p_3) , called (q_1, q_2, q_3) , are found. This set of point pairs is used to calculate the spatial transformation matrix R and T for the current points and global point clouds.

R and T are used to transform the remaining points in P , and the distance error between the transformed points in P and the corresponding points in Q is calculated. If the error of the point pair is less than the threshold δ , it is put into point set I . This process is repeated until the number of points in I reaches the threshold or the number of iterations reaches the maximum. The transformation matrix R and T with the largest number of points in I is selected as the optimal transformation to achieve coarse registration.

3.1.2 Fine registration

The spatial transformation matrix derived from the RANSAC algorithm is computed via three pairs of feature points, which may contain errors within the overall point cloud. If the original point cloud is registered directly with coarse registration, the accuracy of the results is unsatisfactory. Consequently, further refinement is necessary to achieve precise registration.

The iterative closest point (ICP) algorithm [45, 46] is an accurate and efficient approach that is proposed to solve this problem, and it can solve the rigid registration problem. The ICP aims to find the optimal space transformation of two sets of data points. In essence, the algorithm is based on the least squares method to solve the optimal transformation matrix. In the current point cloud P , each point is defined as $\{P_m, m = 1, 2, \dots, N_p\}$, where N_p is the number of current points. In the global point cloud Q , each point is defined as $\{Q_m, m = 1, 2, \dots, N_q\}$, where N_q is the number of global points. The corresponding point set of point set Q and point set P in the k iteration calculation is $\{Q_{km}, m = 1, 2, \dots, N_p\}$. The objective of the alignment transformation is to minimize the objective function of the following Eq.

$$E(R, t) = \sum_{m=1}^{N_p} \left\| Q_m - (RP_m + t) \right\|^2 \quad (6)$$

Each point in the current point set P_m and the corresponding points in point set Q are taken, which minimizes $\|Q_m^k - P_m^k\|$. The rotation matrix R_k and translation vector t_k are calculated, which minimizes $E(R, t) = \sum_{m=1}^{N_p} \left\| Q_m^k - (R_m^k P_m^k + t_m^k) \right\|^2$. Then, the transformation coordinates P_m^{k+1} are defined as $P_m^{k+1} = R_m^k P_m^k + t_m^k$. If $d^{k+1} = 1/N_p \sum_{m=1}^{N_p} \|P_m^{k+1} - Q_m^k\|^2$ is less than the defined dm , the next iteration is started.

The exact coordinate transformation matrix of the current point cloud and the global point cloud is obtained after the iteration is completed. The registration process is shown in Fig. 5. To further enhance the extraction of the proximity between the current point cloud and the global point cloud, the current point cloud and the global point cloud are compared. The overlap rate of the current point cloud is subsequently computed following fine registration.

The overlap rate is defined as c , which reflects the degree of overlap between the current point clouds and global point clouds. The distance between each point in the current point cloud and the nearest point of the global point cloud is calculated and denoted d_p . If d_p is less than the set value d_{min} , it is marked as a coincidence point.

The overlap rate is as follows.

$$c = \frac{N_{pc}}{N_p} \quad (7)$$

where N_p is the number of current point clouds. N_{pc} is the number of points that satisfy the distance requirement.

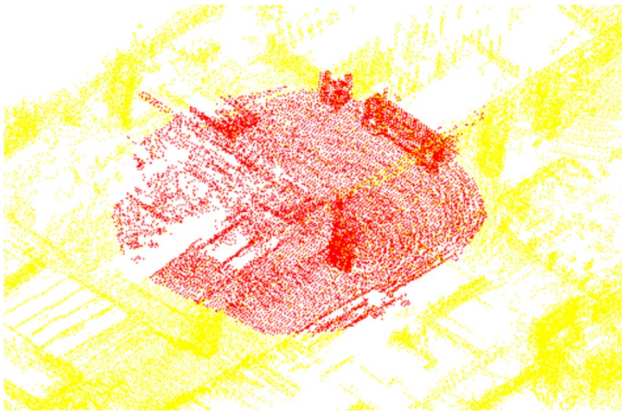


Fig. 5 Registration of the current and global point clouds

3.2 Point cloud object recognition

3.2.1 Preprocessing for recognition

Preprocessing is indispensable for acquiring object information from a point cloud. Among the current point clouds, obtaining effective information from ground point clouds is difficult. Environmental features are usually focused on objects on the ground. Filtering the point cloud of the ground is a basic preprocessing operation in LiDAR and easily affects the clustering of objects.

Cloth simulation filtering To alleviate the influence of noise in the field environment, an initial filtration process is applied to the point cloud data. The ground filtering algorithm, known as cloth simulation filtering (CSF) [47], is derived from the physical model simulation of “cloth”. In computational graphics, a cloth can be conceptualized as a grid comprising interconnected particles with mass, forming what is known as a mass–spring model. The position and shape of the cloth are determined by the position of the particle in three-dimensional space. According to Hooke’s law and Newton’s second law, the relationship between position and force is as follows:

$$m \frac{\partial^2 X(t)}{\partial t^2} = F_{ext}(x, t) + F_{int}(x, t) \quad (8)$$

where x represents the position of the particle at time t . $F_{ext}(x, t)$ represents the external forces, which are composed of gravity and collision forces, and $F_{int}(x, t)$ represents the internal forces of the particles at position x and time t .

A flowchart of the CSF algorithm is depicted in Fig. 6, which can be comprehended as follows:

- (1) The original point cloud is initially inverted.

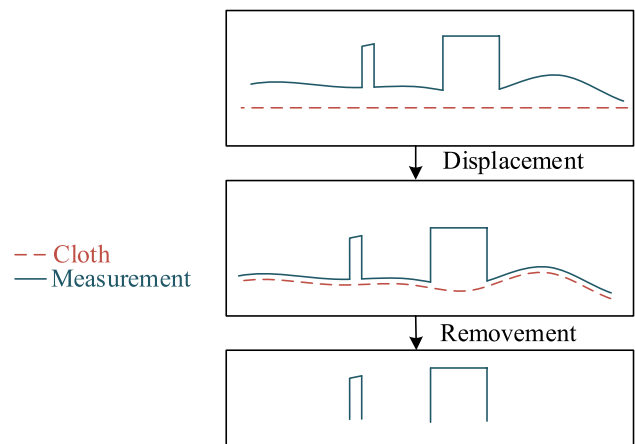


Fig. 6 Flowchart of the CSF algorithm

- (2) Subsequently, the cloth model is superimposed onto the inverted point cloud.
- (3) Then, the position constraints for the fabric nodes are analyzed to simulate and determine the ground shape and position.

Density-based spatial clustering of applications with nodes The ground point cloud is removed, and the remaining point cloud is clustered. Density-based spatial clustering of applications with nodes (DBSCAN) is a type of density-based clustering algorithm. The algorithm assumes that the clustering structure can be determined by the tightness of the sample distribution. According to the perspective of sample density, the continuity between samples is investigated, and the clustering is continuously expanded to obtain the final result.

The radius (EPS) and density threshold (MinPts) of the DBSCAN algorithm are defined. A circle with each point P_m as the centre and EPS as the radius is drawn. This circle is called the P_m neighbourhood. If the number of points in a circle exceeds the density threshold MinPts, the centre of the circle is denoted the core point. All the points in the neighbourhood of the core point P_M are called the density direct of P_M . If both P_i and P_j are direct points of P_m density, P_i is connected to P_j density. A cluster is formed by connecting these points of connected density. In this process, the current point cloud is clustered into several distinct parts.

3.2.2 Object recognition via PointNet

Before PointNet [23] was proposed, point clouds could not be dealt with directly because of the characteristics of 3D point clouds. If the order of the points in the set is disturbed, the same point cloud is still represented. After the rigid transformation (rotation and translation) of the point cloud, the classification result of the output remains unchanged. Features are extracted from the point cloud dataset by solving the classification task for point clouds. The structure of PointNet is shown in Fig. 7.

For each $N \times 3$ input dataset of point clouds, T-Net is used to learn the rotation transformation of the point clouds for calibration and align it in this dimension. Next, a multilayer perceptron (MLP) is used to map the image to 64 dimensions. Each 1×3 vector point is raised into a 1×64 vector through two MLP layers. $N \times 3$ input data becomes $N \times 64$ data and is aligned a second time in this dimension by T-net. Then, the MLP is used to map the data to 1024 dimensions in the same way. Each point is currently represented by a vector of 1024 dimensions, which is redundant for the 3-dimensional point cloud data. The maximum pooling operation is introduced to retain the maximum value of all channels in 1024 dimensions. The 1×1024 vector obtained is the global feature of N point clouds.

Using global features as input, the probability of each object is obtained through a trained multilayer MLP. The point cloud is labelled with the object corresponding to the maximum probability value. The number of objects corresponding to each subpoint cloud after clustering is counted and defined as $\{o_{kn}, n = 1, 2, \dots, N_o\}$.

3.3 Positioning based on the SAE and BPNN

In this section, we propose a method to recognize the current position on the basis of the data obtained. First, the registration and object recognition data are combined. The process of using a stacked autoencoder (SAE) to enhance data is described below. Finally, the structure of the back propagation neural network (BPNN) and the whole classification process are given.

The BPNN is a machine learning model based on neural networks that can automatically extract features from patterns and address nonlinear problems. The advantages of the BPNN model are that the process is relatively simple and that the number of computations is small. Neural network algorithms always need large amounts of data for training. Nevertheless, a finite amount of point cloud data are obtained from metallurgical workshops. In this study, an SAE is used to increase the one-dimensional original training data. The decoding and encoding of the SAE extract the original data features in the process of lossy

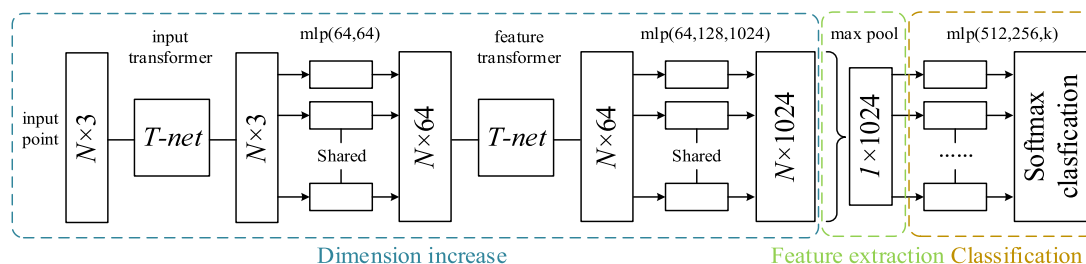


Fig. 7 Structure of PointNet

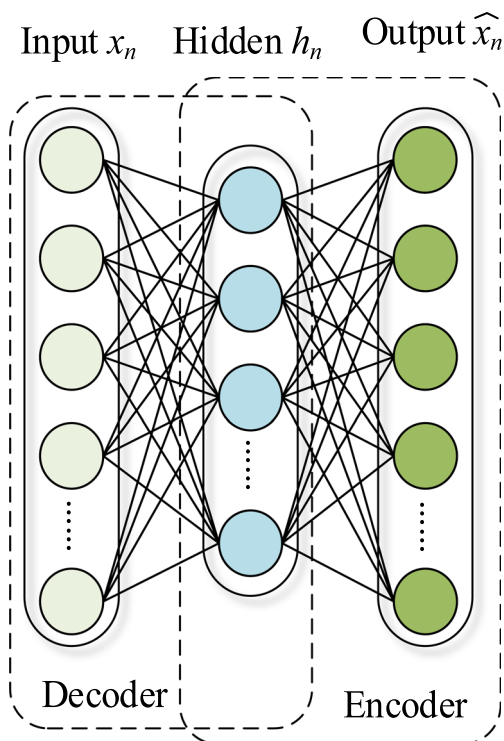


Fig. 8 Structure of the AE

data compression and generate a new dataset on the basis of the retained data features. The SAE can directly train

raw data to generate similar data, which is more efficient than the GAN.

3.3.1 Data enhancement via the SAE

Autoencoders (AEs), which are unsupervised learning algorithms, constitute the basic unit of the SAE. For an AE, there is no need to label the data. It is a feedforward neural network consisting of an input layer, an output layer, and a hidden layer. The AE consists of two components, namely, the encoder and decoder, as shown in Fig. 8. The encoder maps the input data to the hidden layer to extract the data features. The decoder restores the data in the hidden layer to the original data and reconstructs the data.

The AE is essentially the same as that of normal neural networks. Since the output data are a reconstruction based on the input data, ideally, the output should be equal to the input. Therefore, the output of the AE is not the labelled data but the input data itself.

The training datasets are defined as $\{x = x_1, x_2, \dots, x_n\}$, where $x_n \in R^{m \times 1}$. The process of encoding is as follows.

$$\mathbf{h}_n = \sigma(\mathbf{W}_1 \mathbf{x}_n + \mathbf{b}_1) \quad (9)$$

The process of decoding is as follows.

$$\hat{\mathbf{x}}_n = \sigma(\mathbf{W}_2 \mathbf{h}_n + \mathbf{b}_2) \quad (10)$$

Fig. 9 Structure of the SAE

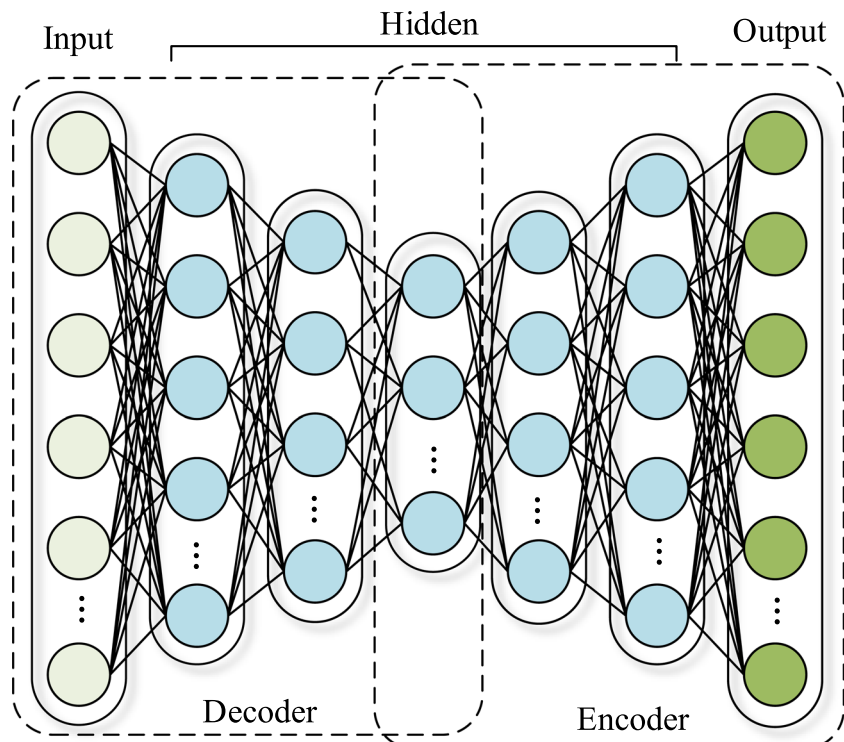
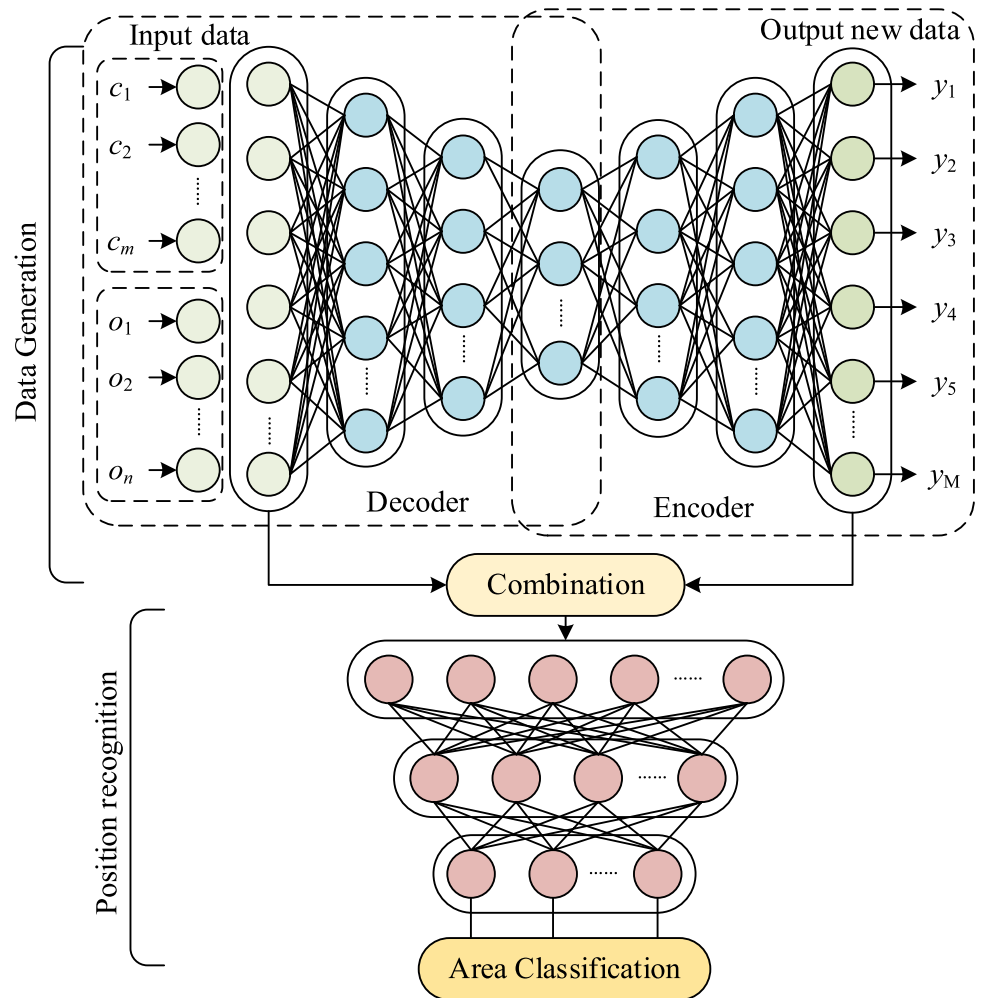


Fig. 10 Framework of position recognition. Combining raw and generated data as a dataset for neural network algorithms



h_n is the hidden layer vector corresponding to x_n , and $\hat{x}_n$ is the output layer vector corresponding to h_n . σ is the activation function, and W and b are the weight and bias, respectively.

The error function is as follows and is usually expressed in terms of the second-order norm.

$$L(x_n, \hat{x}_n) = \|x_n - \hat{x}_n\|^2 \quad (11)$$

The mean square error function is used to calculate the reconstruction error of the SAE.

$$\text{MSE} = \frac{1}{N} \sum_{n=1}^N L(x_n, \hat{x}_n) \quad (12)$$

To minimize the mean square error function, stochastic gradient descent (SGD) is used to find the optimal parameter configuration.

As shown in Fig. 9, the SAE is a stack of two or more AEs. It learns abstract vectors of different dimensions from complex input data by increasing the number of hidden layers. With the increase in the number of neural network

layers, deeper features are obtained, but the training time increases rapidly. The three-layer SAE can extract more representative features while ensuring the calculation efficiency.



Fig. 11 Robot platform with LiDAR

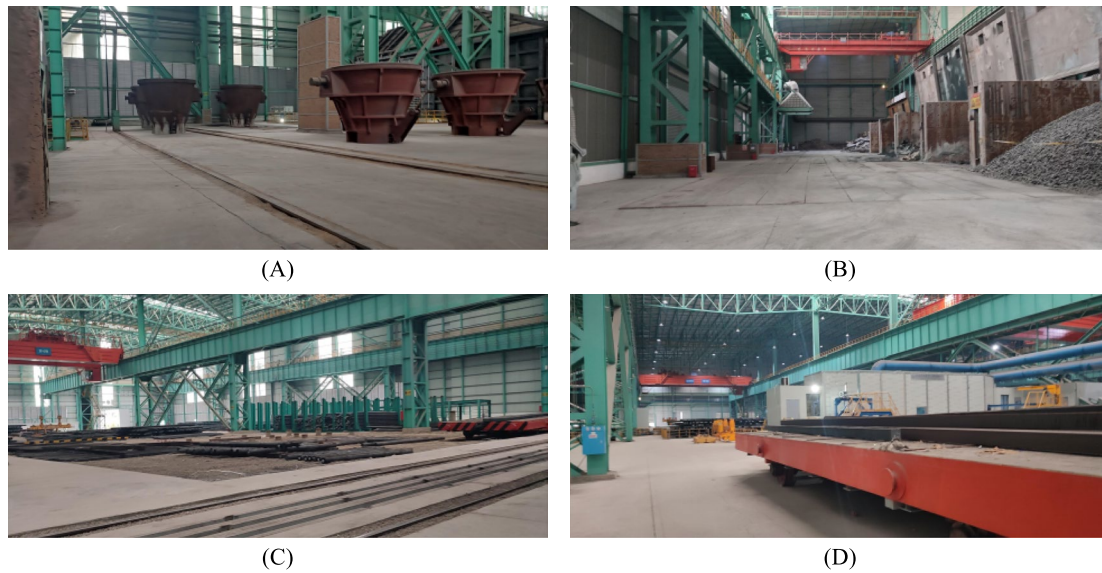


Fig. 12 Several different areas in the steel plant

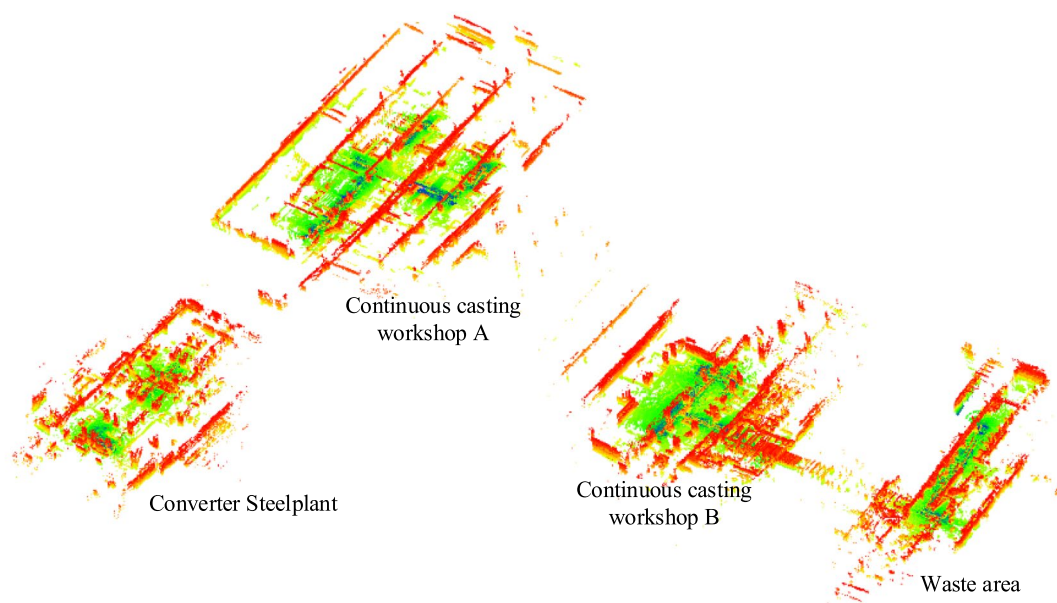


Fig. 13 Global point cloud. It is divided into the steel plant, the waste area, continuous casting workshop A, and continuous casting workshop B

Fig. 14 Current point clouds

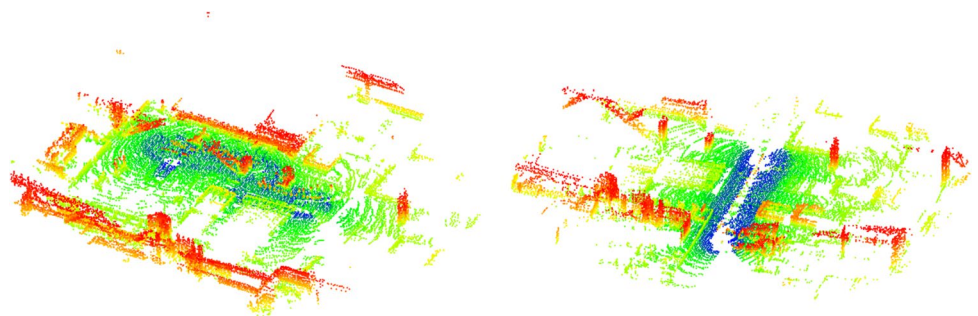


Fig. 15 Registration of the current and global point clouds

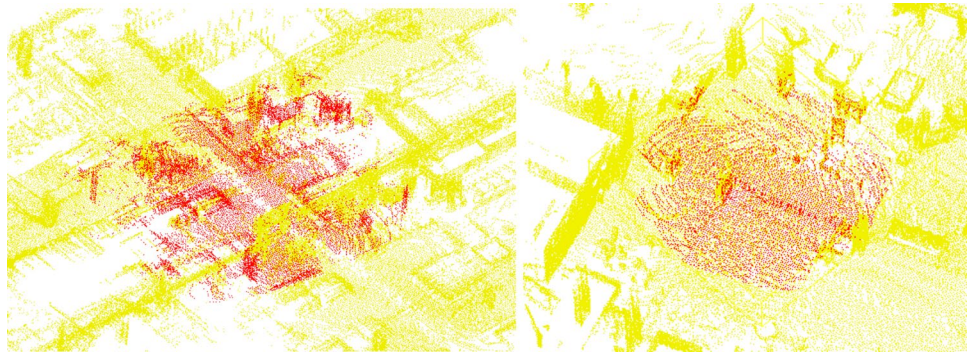


Table 1 Overlap rates of the current point cloud

No.	Area 1	Area 2	Area 3	Area 4
$c^{(1)}$	86.29%	7.34%	14.60%	17.45%
$c^{(2)}$	79.51%	16.46%	14.02%	19.45%
$c^{(3)}$	85.55%	13.07%	10.98%	9.92%
$c^{(4)}$	92.37%	8.08%	11.00%	4.53%
$c^{(5)}$	80.53%	18.39%	19.23%	13.61%
...	...	...	...	...

By training stacked autoencoders, the input original data pass through multiple layers of encoders to extract features and decoders to generate new data. The generated data are different from the original data when the features are retained. It can be used to compensate for an insufficient sample set.

3.3.2 Position recognition by the BPNN

The framework of position recognition based on the BPNN is shown in Fig. 10.

The overlap rate c is obtained via point cloud registration. Considering the error in the registration process and the unification of the dimensions of the neural network input data, the results of registration are defined as $\{c_{km}, m = 1, 2, \dots, N_w\}$. This array evaluates the level of the point cloud registration results for the current point cloud and each workshop's point cloud. N_w is defined as the number of workshops, and k is the number of current point clouds. We obtain the list of the objects in the current points, which is defined as $\{o_{kn}, n = 1, 2, \dots, N_o\}$. N_o is defined as the number of types of objects.

The input of a BPNN is usually in the form of a one-dimensional array. To retain the registration and object recognition features, we combine two arrays and define them as $\{c_1, \dots, c_{km}, o_1, \dots, o_{kn}\}$, which can be replaced by $\{F_k = f_1, f_2, \dots, f_{kM}, M = N_w + N_o\}$.

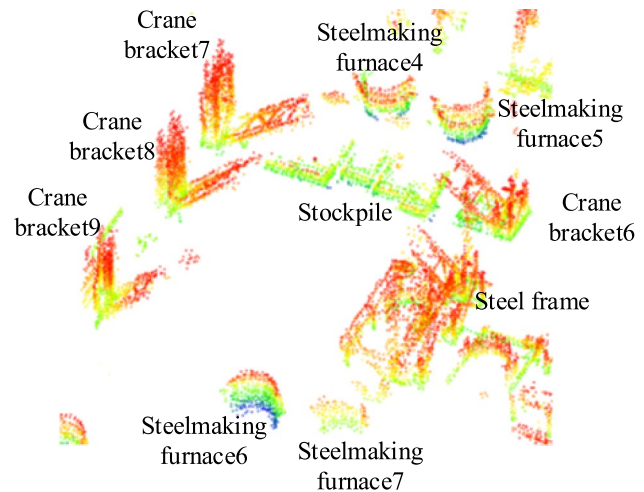


Fig. 16 Labelling of the point cloud. Most of the point cloud data involve the construction of structures and massive equipment

The trained encoder and decoder generate new data using F_k as the original input data. The generated data are defined as $\{y_k = y_{k1}, y_{k2}, \dots, y_{kM}, M = N_w + N_o\}$. F_k and y_k are merged as the training dataset for the BPNN.

According to the training data, the position recognition problem can be transformed into a place classification problem. The different workshops are defined as numbers 1 to N_w , and the training data are labeled with the number of workshops. Since the present study is a classification problem, it is imperative to employ cross-entropy loss as the designated loss function. The formula for this purpose can be expressed as follows.

$$E = - \sum_{i=1}^M \hat{y}_i \ln y_i \quad (13)$$

In each iteration, to minimize L , the backpropagation algorithm is used to update the network parameters (w, b). The formula is as follows.

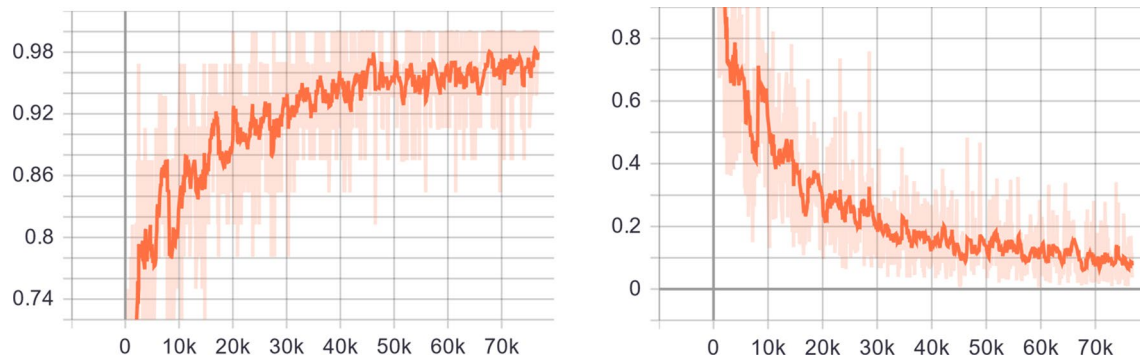


Fig. 17 Accuracy and loss of PointNet

Table 2 Quantification of object recognition

No.	Steel furnace	Crane bracket	Wall	Steel frame	Rail car	Digger	Rigging	Semi-finished product	...
$o^{(1)}$	3	15	6	0	0	0	0	0	...
$o^{(2)}$	3	13	4	2	0	0	0	0	...
$o^{(3)}$	3	6	1	0	0	0	0	0	...
$o^{(4)}$	3	8	1	1	0	0	0	0	...
$o^{(5)}$	5	6	1	1	0	0	0	0	...
...	...	...	...	...	...	...	...	...	...

Table 3 Dataset after fusion

No.	Overlap rate	Object recognition quantity statistics
$y^{(1)}$	$c^{(1)}$	$o^{(1)}$
$y^{(2)}$	$c^{(2)}$	$o^{(2)}$
$y^{(3)}$	$c^{(3)}$	$o^{(3)}$
$y^{(4)}$	$c^{(4)}$	$o^{(4)}$
$y^{(5)}$	$c^{(5)}$	$o^{(5)}$
.....	...	...

Table 4 Loss rates with the structure

Test No.	Number of neurons in each layer of the SAE			Loss rate
	Layer 1	Layer 2	Layer 3	
1	14	30	60	0.5993
2	14	50	80	0.5883
3	14	20	26	0.2511
4	14	13	12	0.0874
5	14	12	10	0.0744

$$\Delta w = -\eta * \frac{\partial E}{\partial w}, \Delta b = -\eta * \frac{\partial E}{\partial b} \quad (14)$$

3.4 Summary of the methods

The global point cloud is defined as P_g and is divided into several room point clouds P_m according to the enclosed space. The current point cloud at different locations collected in each room is represented as P_i , and the room number is labeled. The coincidence rate c_n between the current point clouds P_i and P_m is calculated after coarse and fine registration, respectively. Moreover, the current point cloud P_i is preprocessed for object recognition through PointNet. The number of objects in the current point cloud is denoted o_m . The dataset y_{m+n} was obtained via data enhancement after combining o_m and c_n . All the datasets are taken as input datasets of the BPNN, which are divided into training sets and test sets. The accuracy of the model parameters obtained by training is verified on the test set.

4 Case study

In this section, a series of experiments are conducted to verify the performance of the model via the point cloud dataset obtained via LsLiDAR -C32. The robot platform equipped



Fig. 18 Grey images of the original and reconstructed data. The resemblance between the original data and the reconstructed data can be visually demonstrated more intuitively

Fig. 19 Results of the object recognition data

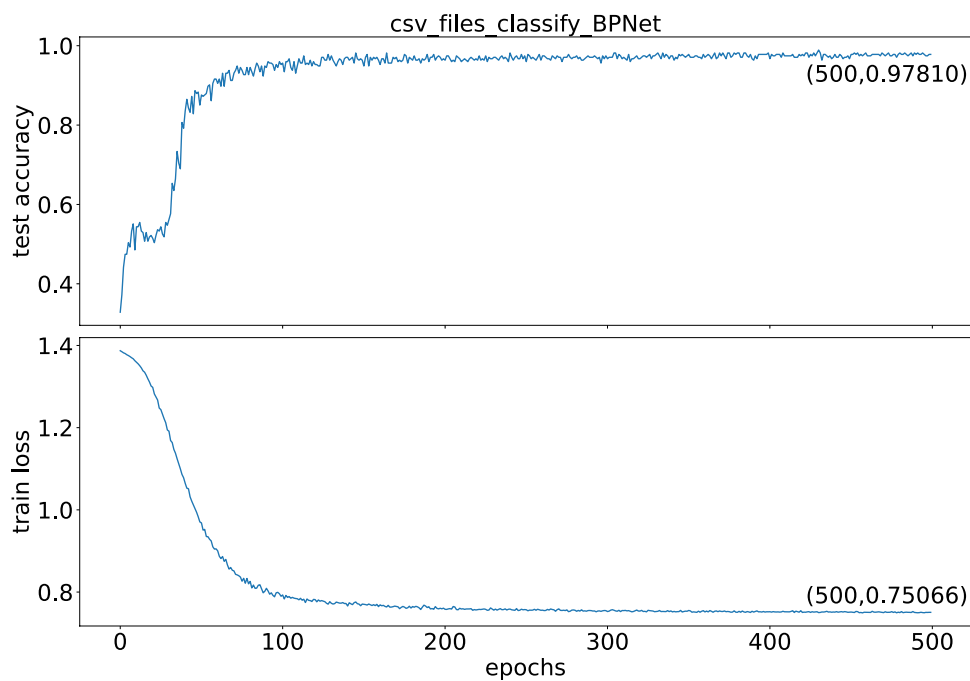


Fig. 20 Results of the overlap rate data

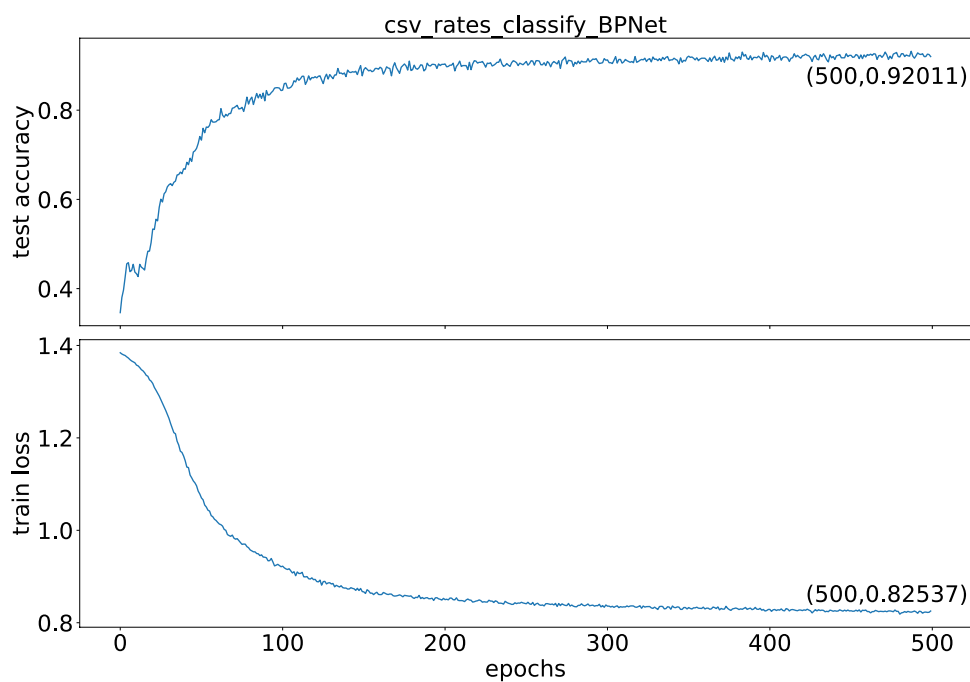
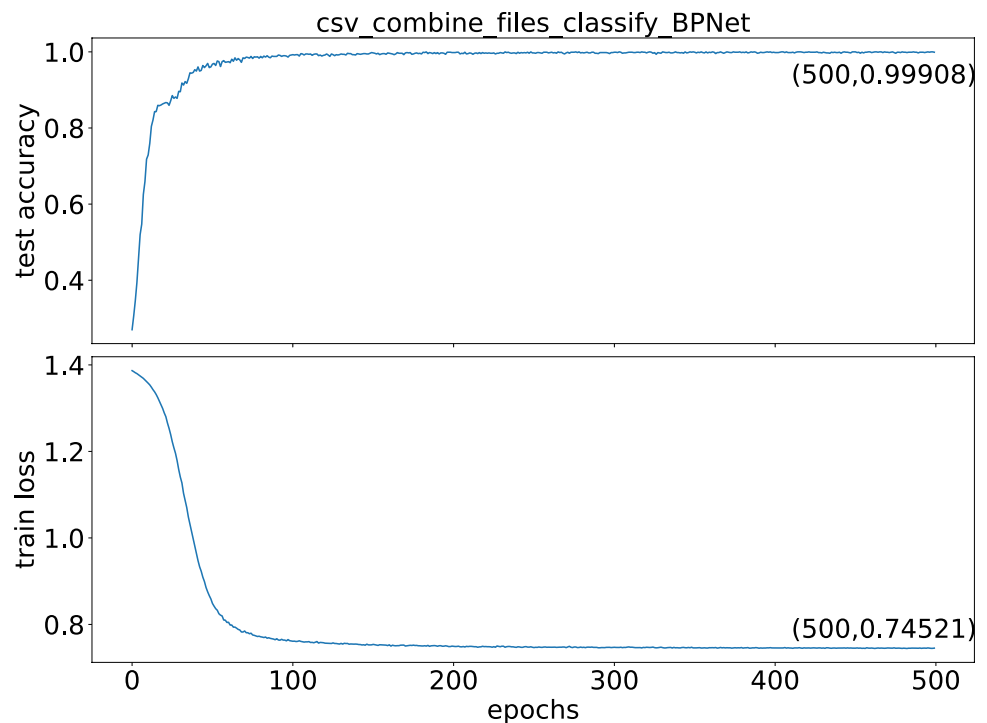


Fig. 21 Results of the combined data



with LiDAR in this experiment is shown in Fig. 11. The model is implemented in Python 3.8 and executed on Windows 10 with a GTX 3060ti GPU.

The effectiveness of the proposed localization method is evaluated on the basis of the dataset. In this section, how to process the original point cloud dataset is introduced first. Subsequently, the details of the neural network and evaluation criterion are described. Finally, the method is evaluated experimentally.

4.1 Point cloud dataset preprocessing

We collected samples from several different functional areas of the Masteel Special Steel Plant. These areas also vary in volume and device, as shown in Fig. 12. First, to obtain global point cloud data, it is necessary to merge scanned point clouds due to physical constraints. The LiDAR device is mounted on a mobile robot platform that systematically traverses the factory in a predetermined sequence. A global point cloud of the metallurgical plant was subsequently constructed. The different workshops in the metallurgical plant can be regarded as larger rooms within a broader scope, each with its function. Therefore, the global point cloud of a metallurgical plant is divided into rooms 1 to 4, which are the converter steel plant, the waste area, continuous casting workshop A, and continuous casting workshop B, respectively (Fig. 13).

When the robot arrives at an area or starts up, it acquires the current location point cloud. The robot rotates once in

place, and the point cloud obtained is the current point cloud. The experimenters collected samples at 15 random locations in each room, once every hour, 3–4 times. If the robots are located in certain rooms, we label their sensor data with the corresponding room number. The current point cloud collected at several locations in the plant is shown in Fig. 14.

To establish the correlation between the current and global point clouds, a progressive coarse-to-fine registration approach is employed, which is shown in Fig. 15. The current point cloud can be considered a submap of the global point cloud. Given the potential similarity among different regions and the inherent registration error, submaps are aligned with each corresponding area on the global map. The overlap rate of the current point cloud is presented in Table 1 as depicted below.

Before point cloud recognition, a preprocessing step is conducted to eliminate outliers and downsample the data to reduce their density. The current point cloud is clustered by removing the ground points, considering each segment of the classified point cloud as an individual object. The point cloud data were labeled manually and are depicted in Fig. 16.

The labeled point cloud obtained after processing the collected current point cloud serves as both the training and test sets for PointNet in deep learning. The accuracy and loss are shown in Fig. 17.

After the completion of PointNet training, each segmented point cloud is evaluated via PointNet calculations. The quantity of partial objects in each group of point clouds is presented in Table 2 below.

Table 5 Results of several datasets

Datasets	No. 1		No. 2		No. 3		No. 4	
	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
$c^{(n)}$	0.9781	0.7535	0.9635	0.7573	0.9745	0.7522	0.9671	0.7522
$o^{(n)}$	0.887	0.8596	0.8953	0.8645	0.8733	0.8661	0.8833	0.8495
$y^{(n)}$	0.9917	0.753	0.9908	0.7466	0.9981	0.7466	0.9963	0.7456
Datasets		No. 5	No. 6		No. 7		No. 8	
	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
$c^{(n)}$	0.9652	0.7421	0.9525	0.7453	0.9865	0.7328	0.9655	0.7472
$o^{(n)}$	0.876	0.8656	0.8825	0.8325	0.8733	0.8551	0.8644	0.8475
$y^{(n)}$	0.9912	0.753	0.991	0.7376	0.9901	0.7467	0.9893	0.7427

Next, the data combined with the overlap rate, the overlap rate, and the object recognition data are used as the training set for localization. Table 3 shows the dataset after fusion, which is expressed as $y^{(n)} = [c^{(n)}, o^{(n)}]$.

4.2 Hyperparameter optimization

To enhance the performance of the SAE and BPNN models and optimize the experimental results, each parameter of the model is meticulously optimized in this study. The testing accuracy and loss rate are usually taken as standards to measure the performance of the model. In this study, the optimal parameter is selected by adjusting the influence of a parameter on the precision. The final results fluctuate slightly in many experiments because the training of the neural network is affected by its random initial value. Therefore, the average results of ten simulation experiments under the same conditions are used for all the listed final data to reduce the influence of experimental error and the uncertainty of the results.

The above datasets of each point cloud group are fused as the training set. Each dataset can be represented as $\{F_k = f_{kl}, f_2, \dots, f_M, M = N_w + N_o\}$. To exert a more pronounced influence on subsequent training using the augmented dataset, a varied combination of neuron numbers is employed, as illustrated in Table 4. By manipulating the various configurations of neurons to increase the diversity of the generated data, the robustness and precision of the neural network model can be improved.

Figure 18 compares the original and the reconstructed data, which are presented in pixel format. By showcasing grayscale images, the resemblance between the original data and the reconstructed data can be visually demonstrated more intuitively.

4.3 Results of different datasets

Several comparative experiments were conducted to validate the model's performance by comparing it with that of other datasets. The training process begins by solely utilizing object recognition data. The overlap rates are subsequently exclusively employed for further training. Finally, the fused data are incorporated into a comprehensive training set. The figure below illustrates the accuracy and loss rate of this meticulous training procedure. (Figs. 19, 20 and 21)

Owing to the sensitivity of neural network algorithms to random initial values, a total of 8 experiments were conducted under identical conditions. Eight sets of experiments were repeated to select the data with low fluctuations and averages to reduce the error. The refined findings are presented in Table 5.

Table 6 Results of several datasets

Datasets	No. 1		No. 2		No. 3		No. 4	
	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
y_0	0.8756	0.8325	0.8953	0.8741	0.8622	0.8774	0.8996	0.8335
y_{i1}	0.9781	0.7911	0.9755	0.7878	0.9645	0.7822	0.9782	0.7822
y_{i2}	0.9852	0.7724	0.9855	0.7573	0.9723	0.7713	0.9871	0.7723
y_{i3}	0.9891	0.7635	0.9843	0.7611	0.9864	0.7656	0.9833	0.7695
y_{i4}	0.9982	0.7624	0.9855	0.7973	0.9865	0.7622	0.9872	0.7638
y_{i5}	0.9881	0.7535	0.9855	0.7673	0.9965	0.7677	0.9982	0.7645
y_{ic}	0.9921	0.7513	0.9914	0.7356	0.9967	0.7652	0.9963	0.7531

The above data indicate that the accuracy of the fused dataset is significantly higher than that of all the datasets before fusion. Moreover, the convergence speed is higher, and the loss rate is lower. Finally, the model achieved an impressive test accuracy of 99% while concurrently maintaining a loss value of 0.75.

Different datasets are obtained according to different combinations of neuron numbers in the previous SAE, which are denoted y_{i1} to y_{i5} . The raw datasets that were not enhanced are denoted y_0 . In addition, the same amount of data is extracted from 5 groups of data and combined, the combination of which is denoted y_{ic} . The accuracy and loss rate are obtained by applying different combinations to the BPNN calculation, as shown in the table.(Table 6)

According to the accuracy and loss rates in the table, data enhancement improves the calculation accuracy. Different combinations of neurons have different computational accuracies. The dataset after the fusion of various combinations further improves the accuracy of the algorithm. In other words, in the case of insufficient sample data, the method presented in this paper provides a model for accurate positioning in industrial workshops.

The authors of another paper aimed to achieve robust and cost-effective room-level localization for an indoor mobile robot [3]. A robotic room-level localization method is proposed, and the performance of the different coding strategies is evaluated on the collected sonar measurement dataset. Several samples were collected from ten different functional areas in a university building.

Different difficulty levels are set, and the performance for the different cases is evaluated independently. The accuracies for the three cases are 0.92, 0.81, and 0.78. However, the introduction of disturbances can have a significant effect on recognition. In this paper, the location accuracy of the proposed method is calculated by using the data collected by LiDAR in different workshops of the metallurgical plant.

Table 7 shows a comparison of the two robot localization methods. CSP, CSWA, CSWB, and WA represent the converter steel plant, continuous casting workshop A, continuous casting workshop B, and the waste area, respectively. HSF, KIT, CAF, INF, STO, HFF, and GYM represent the

hall on the second floor, the kitchen, the café, the information desk, the storage room, the hall on the first floor, and the gymnasium, respectively.

Compared with the methods mentioned above, our method involves the collection of several samples in several functional areas in an industry workshop. Industrial workshops are much larger than office buildings. In addition, the devices and objects inside frequently move and change, and many interference factors exist within them. On this basis, the accuracy of the method proposed in this paper is still superior.

4.4 Discussion

The accuracy of the algorithm is evaluated by the experiments above, but many works are worthy of further investigation. Laser radar needs to collect a certain amount of data when the current position point cloud is acquired before it can be identified and located. Therefore, it takes time to obtain accurate positioning information, and the calculation result has a certain lag. The process of data enhancement requires encoding and decoding of input data, which consumes certain computing resources. When the amount of point cloud data exceeds a certain level, the hardware equipment installed by the robot may not be able to complete the calculation in time, affecting the real-time performance of the algorithm. If the number of samples is too small, overfitting problems may be introduced even with the blessing of data enhancement. Then, in unstructured factory environments where these areas lack clearly defined routes and typical building structures, the application of this method may pose significant challenges. Second, we use only a single LiDAR device to collect point cloud data without considering factors that may affect the results, such as gradient, distance, and density, which may lead to incomplete or incorrect point clouds. In addition, the point cloud data of only 4 metallurgical workshops were collected in this paper, and the identification of more rooms and more variable environments needs further research.

On the basis of the above shortcomings, we will conduct further research on the following aspects in the future. First,

Table 7 Comparison of the two robot localization methods

Rooms	CSP	CSWA	CSWB	WA	HSF	KIT	CAF	INF	STO	HFF	GYM
Accuracy		0.99	0.98	0.98	0.83	0.85	0.83	0.83	0.92	0.80	0.88
Size (m)		100 × 250			4.8 × 21						

we reduce the scanning time for LiDAR to acquire the current location point cloud. We will study how to address the point cloud in the case of insufficient data to ensure accurate positioning under the premise of improving real-time performance. We will study the potential limitations of the algorithm. We will investigate how data enhancement specifically affects the cost of computing and determine an upper limit to the amount of data that can be guaranteed for the algorithm. We will examine how to mitigate or avoid overfitting problems in the case of insufficient data, as well as the minimum amount of data required to ensure the implementation of the algorithm. We will further determine what size and level of chaos in the environment will affect the implementation of the method. Second, we will increase the type and number of sensors used to obtain a variety of environmental information to address the problem of insufficient raw data. Then, we will further improve the existing model and adjust the necessary parameters to improve the operation efficiency so that the method can achieve better real-time positioning. In addition, we will collect data from larger industrial plants in the future that have more workshops and more variable working environments. The proposed method was further improved by increasing the difficulty of the experiment.

5 Conclusion

A room-level location method for mobile robots based on point cloud registration and target recognition is proposed. The proposed method is based on two directions of 3D LiDAR positioning methods, namely, retrieval-based and registration-based methods. The feature data of the two methods are extracted as the basis of positioning, and the original data amount is increased by data enhancement to solve the problem of insufficient raw data. The localization problem is converted into a classification problem, and the backpropagation neural network algorithm is used to identify the problem. On this basis, we collected original point cloud data from a metallurgical plant and carried out experiments to verify the accuracy of the proposed method.

The method proposed in this paper provides a feasible method for robot localization in complex environments. The positioning of mobile robots in complex industrial environments is especially important. Because of the shortcomings of the method, we further study the efficiency and accuracy of the algorithm from the aspects of the timeliness of the algorithm, multisensor information fusion, and the effect of the algorithm in more complex environments.

Data Availability The datasets for this study have been uploaded to GitHub and can be accessed via the following link: https://github.com/lanfengzhiwuZ/Robot_localization.

Declarations

Competing interests The authors declare that they have no competing interests.

Ethical and informed consent for the data used This study was performed by the authors, and no human participants other than the authors were involved. Informed consent was obtained from all the authors.

References

- Chung M-A, Lin C-W (2022) An improved localization of mobile robotic system based on AMCL Algorithm. *IEEE Sens J* 22(1):900–908. <https://doi.org/10.1109/JSEN.2021.3126605>
- Yu S, Wang C, Yu Z, Li X, Cheng M, Zang Y (2021) Deep regression for LiDAR-based localization in dense urban areas. *ISPRS J Photogramm Remote Sens* 172:240–252. <https://doi.org/10.1016/j.isprsjprs.2020.12.013>
- Liu H, Sun F, Fang B, Zhang X (2017) Robotic room-level localization using multiple sets of Sonar measurements. *IEEE Trans Instrum Meas* 66(1):2–13. <https://doi.org/10.1109/TIM.2016.2618978>
- Yang N, von Stumberg L, Wang R, Cremers D (2020) D3VO: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Seattle, WA, USA. pp 1278–1289. <https://doi.org/10.1109/CVPR42600.2020.00136>
- Rozenberszki D, Majdik AL (2020) LOL: Lidar-only odometry and localization in 3D point cloud maps. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, Paris, France, pp 4379–4385. <https://doi.org/10.1109/ICRA40945.2020.9197450>
- Bianchi V, Ciampolini P, De Munari I (2019) RSSI-based indoor localization and identification for zigbee wireless sensor networks in smart homes. *IEEE Trans Instrum Meas* 68(2):566–575. <https://doi.org/10.1109/TIM.2018.2851675>
- Niu J, Wang B, Cheng L, Rodrigues JJPC (2015) WicLoc: An indoor localization system based on WiFi fingerprints and crowd-sourcing. In: 2015 IEEE International Conference on Communications (ICC). IEEE, London, pp 3008–3013. <https://doi.org/10.1109/ICC.2015.7248785>
- Yoo W, Kim H, Shin M (2020) Stations-oriented indoor localization (SOIL): a BIM-Based occupancy schedule modeling system. *Build Environ* 168:106520. <https://doi.org/10.1016/j.buildenv.2019.106520>
- Li S, Lu J, Chen S (2020) A room-level tag trajectory recognition system based on multi-antenna RFID reader. *Comput Commun* 149:350–355. <https://doi.org/10.1016/j.comcom.2019.10.025>
- Liu X, Zhan Y, Cen J (2018) An energy-efficient crowd-sourcing-based indoor automatic localization system. *IEEE Sens J* 18:6009–6022. <https://doi.org/10.1109/JSEN.2018.2842239>
- Wang B, Chen C, Xiaoxuan LuC, Zhao P, Trigoni N, Markham A (2020) AtLoc: Attention guided camera localization. *AAAI* 34(06):10393–10401. <https://doi.org/10.1609/aaai.v34i06.6608>
- Zhang Z, Zhao J, Huang C, Li L (2023) Learning visual semantic map-matching for loosely multi-sensor fusion localization of autonomous vehicles. *IEEE Trans Intell Veh* 8(1):358–367. <https://doi.org/10.1109/ITV.2022.3173662>
- Cheong H, Kim E, Park S-K (2019) Indoor global localization using depth-guided photometric edge descriptor for mobile robot navigation. *IEEE Sensors J* 19(22):10837–10847. <https://doi.org/10.1109/JSEN.2019.2932131>
- Dubé R et al (2020) SegMap: Segment-based mapping and localization using data-driven descriptors. *Int J Robot Research* 39(2–3):339–355. <https://doi.org/10.1177/0278364919863090>
- Li W, Wang C, Lin C, Xiao G, Wen C, Li J (2020) Inlier extraction for point cloud registration via supervoxel guidance and game theory optimization. *ISPRS J Photogramm Remote Sens* 163:284–299. <https://doi.org/10.1016/j.isprsjprs.2020.01.021>
- Li J, Zhan J, Zhou T, Bento VA, Wang Q (2022) Point cloud registration and localization based on voxel plane features. *ISPRS Journal of Photogrammetry and Remote Sensing* 188:363–379. <https://doi.org/10.1016/j.isprsjprs.2022.04.017>
- Yan B, Fan P, Lei X, Liu Z, Yang F (2021) A real-time apple targets detection method for picking Robot based on improved YOLOv5. *Remote Sens* 13(9):1619. <https://doi.org/10.3390/rs13091619>
- Wu Y, Wang Y, Zhang S, Ogai H (2021) Deep 3D object detection networks using LiDAR data: a review. *IEEE Sens J* 21(2):1152–1171. <https://doi.org/10.1109/JSEN.2020.3020626>
- Dong Z, Yang B, Liang F, Huang R, Scherer S (2018) Hierarchical registration of unordered TLS point clouds based on binary shape context descriptor. *ISPRS J Photogrammet Remote Sens* 144:61–79. <https://doi.org/10.1016/j.isprsjprs.2018.06.018>
- Han X-F, Yan X-Y, Sun S-J (2021) Novel methods for noisy 3D point cloud based object recognition. *Multimed Tools Appl* 80(17):26121–26143. <https://doi.org/10.1007/s11042-021-10794-3>
- Zhirong Wu et al (2015) “3D ShapeNets: a deep representation for volumetric shapes”. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Boston, MA, USA. pp 1912–1920. <https://doi.org/10.1109/CVPR.2015.7298801>
- Gao H, Cheng B, Wang J, Li K, Zhao J, Li D (2018) Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment. *IEEE Trans Ind Inf* 14(9):4224–4231. <https://doi.org/10.1109/TII.2018.2822828>
- Charles RQ, Su H, Kaichun M, Guibas LJ (2017) PointNet: deep learning on point sets for 3D classification and segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Honolulu, HI. pp 77–85. <https://doi.org/10.1109/CVPR.2017.16>
- Uy MA, Lee GH (2018) PointNetVLAD: deep point cloud based retrieval for large-scale place recognition. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, Salt Lake City, UT, USA, pp 4470–4479. <https://doi.org/10.1109/CVPR.2018.00470>
- Zhang W, Xiao C (2019) PCAN: 3D attention map learning using contextual information for point cloud based retrieval. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Long Beach, CA, USA, pp 12428–12437. <https://doi.org/10.1109/CVPR.2019.01272>
- Yang Y, Chen F, Wu F, Zeng D, Ji Y, Jing X-Y (2020) Multi-view semantic learning network for point cloud based 3D object detection. *Neurocomputing* 397:477–485. <https://doi.org/10.1016/j.neucom.2019.10.116>
- Ji C-L, Yu T, Gao P, Wang F, Yuan R-Y (2024) Yolo-tla: an efficient and lightweight small object detection model based on YOLOv5. *J Real-Time Image Proc* 21(4):141. <https://doi.org/10.1007/s11554-024-01519-4>
- Ji C, Liu G, Zhao D (2022) Monocular 3D object detection via estimation of paired keypoints for autonomous driving. *Multimed Tools Appl* 81(4):5973–5988. <https://doi.org/10.1007/s11042-021-11801-3>
- Liu B, Tian B, Wang H, Qiao J, Wang Z (2022) FuseNet: 3D object detection network with fused information for lidar point

- clouds. *Neural Process Lett* 54(6):5063–5078. <https://doi.org/10.1007/s11063-022-10848-z>
30. Dong Z et al (2020) Registration of large-scale terrestrial laser scanner point clouds: a review and benchmark. *ISPRS J Photogramme Remote Sens* 163:327–342. <https://doi.org/10.1016/j.isprsjprs.2020.03.013>
 31. Nguyen HL, Belton D, Helmholtz P (2019) Review of mobile laser scanning target-free registration methods for urban areas using improved error metrics. *Photogram Rec* 34(167):282–303. <https://doi.org/10.1111/phor.12293>
 32. Wei P, Yan L, Xie H, Huang M (2022) Automatic coarse registration of point clouds using plane contour shape descriptor and topological graph voting. *Autom Constr* 134:104055. <https://doi.org/10.1016/j.autcon.2021.104055>
 33. Johnson AE, Hebert M (1999) Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans Pattern Anal Mach Intell* 21(5):433–449. <https://doi.org/10.1109/34.765655>
 34. Rusu RB, Blodow N, Marton ZC, Beetz M (2008) Aligning point cloud views using persistent feature histograms. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, Nice, pp 3384–3391. <https://doi.org/10.1109/IROS.2008.4650967>
 35. Rusu RB, Blodow N, Beetz M (2009) Fast point feature histograms (FPFH) for 3D registration. In: 2009 IEEE International Conference on Robotics and Automation. IEEE, Kobe, pp 3212–3217. <https://doi.org/10.1109/ROBOT.2009.5152473>
 36. Zhao Z, Zhang W, Gu J, Yang J, Huang K (2019) Lidar mapping optimization based on lightweight semantic segmentation. *IEEE Trans Intell Veh* 4(3):353–362. <https://doi.org/10.1109/ITV.2019.2919432>
 37. Lin L, Zhang W, Cheng M, Wen C, Wang C (2022) Planar primitive group-based point cloud registration for autonomous vehicle localization in underground parking lots. *IEEE Geosci Remote Sens Lett* 19:1–5. <https://doi.org/10.1109/LGRS.2021.3053252>
 38. Guevara J, Gene-Mola J, Gregorio E, Auat Cheein F (2022) 3D spectral graph wavelet point signatures in pre-processing stage for mobile laser scanning point cloud registration in unstructured orchard environments. *IEEE Sensors J* 22(2):1720–1728. <https://doi.org/10.1109/JSEN.2021.3129340>
 39. Zhou B, He Y, Huang W, Yu X, Fang F, Li X (2022) Place recognition and navigation of outdoor mobile robots based on random Forest learning with a 3D LiDAR. *J Intell Robot Syst* 104(4):72. <https://doi.org/10.1007/s10846-021-01545-5>
 40. Xu Z, Xu E, Zhang Z, Wu L (2019) Multiscale sparse features embedded 4-points congruent sets for global registration of TLS point clouds. *IEEE Geosci Remote Sensing Lett* 16(2):286–290. <https://doi.org/10.1109/LGRS.2018.2872353>
 41. Besl PJ, McKay ND (1992) A method for registration of 3-D shapes. *IEEE Trans Pattern Anal Mach Intell* 14(2):239–256. <https://doi.org/10.1109/34.121791>
 42. Quan D et al (2022) Deep feature correlation learning for multi-modal remote sensing image registration. *IEEE Trans Geosci Remote Sens* 60:1–16. <https://doi.org/10.1109/TGRS.2022.3187015>
 43. Li L, Han L, Ding M, Liu Z, Cao H (2022) Remote sensing image registration based on deep learning regression model. *IEEE Geosci Remote Sens Lett* 19:1–5. <https://doi.org/10.1109/LGRS.2020.3032439>
 44. Shi Z, Chang C, Chen H, Du X, Zhang H (2022) PR-NET: progressively-refined neural network for image manipulation localization. *Int J Intell Sys* 37(5):3166–3188. <https://doi.org/10.1002/int.22822>
 45. Du S, Zheng N, Ying S, Liu J (2010) Affine iterative closest point algorithm for point set registration. *Pattern Recognit Lett* 31(9):791–799. <https://doi.org/10.1016/j.patrec.2010.01.020>
 46. Du S, Zheng N, Xiong L, Ying S, Xue J (2010) Scaling iterative closest point algorithm for registration of m-D point sets. *J Vis Commun Image Represent* 21(5–6):442–452. <https://doi.org/10.1016/j.jvcir.2010.02.005>
 47. Zhang W et al (2016) An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Remote Sens* 8(6):501. <https://doi.org/10.3390/rs8060501>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



Yunzhi Li is a doctoral student at Anhui University of Technology and has participated in multiple national-level research projects. His main research direction is the development of intelligent equipment and industrial applications of artificial intelligence.



Libin Tan is a master's supervisor at Anhui University of Technology. His main research areas are intelligent equipment development, digital design and manufacturing, and industrial applications of artificial intelligence. He has led and participated in multiple national-level scientific research projects, including the National Key Research and Development Program and the National Natural Science Foundation.



Xiangrong Xu is a doctoral supervisor at Anhui University of Technology, a member of the Institute of Electrical and Electronics Engineers (IEEE) and the ASME Society, and serves as a reviewer for multiple international academic journals and conferences. His research interests include bionics, aerial robots, robot trajectory planning and control, biomechanics, and more. He has led and participated in multiple national-level scientific research projects, including the National Key Research and Development Program and the National Natural Science Foundation.



Zequn Zhang is a postdoctoral fellow and master's supervisor at Nanjing University of Aeronautics and Astronautics. His main research areas are Intelligent Manufacturing Systems and Artificial Intelligence. He has led and participated in multiple national-level scientific research projects, including the National Key Research and Development Program, National Natural Science Foundation, and National Defense Basic Research Projects.