IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 25, 2021, accepted May 14, 2021, date of publication May 17, 2021, date of current version May 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3081252

Indoor Multi-Robot Cooperative Mapping Based on Geometric Features

ZHIYUAN ZHU[1], **WENSONG JIANG**[1,2], **LI YANG**[3], **AND ZAI LUO**[1]
[1]College of Metrology and Measurement Engineering, China Jiliang University, Hangzhou 310018, China
[2]Anhui Province Key Laboratory of Measuring Theory and Precision Instrument, Hefei University of Technology, Hefei 230009, China
[3]College of Information Engineering, China Jiliang University, Hangzhou 310018, China

Corresponding authors: Wensong Jiang (jwensong@cjlu.edu.cn) and Zai Luo (luozai@cjlu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 52005471 and Grant 52075511.

ABSTRACT To improve the efficiency and accuracy of the indoor multi-robot cooperative mapping, the extraction method of overlapping regions between each local map based on the geometric features is proposed. First, the geometric features of local maps are extracted and fitted to represent the semantic information of the indoor building structure. Second, the attribute similarity evaluation, the topology similarity evaluation, and the ICP evaluation are used to extract the most similar frame pair between each local map. Third, the local map is merged by overlapping area registration to construct a global map. The proposed method is verified in three different real-world scenarios, the experimental result shows that the efficiency and accuracy of the cooperative mapping method are improved compared with the single-robot mapping method and the existing cooperative mapping method.

INDEX TERMS LIDAR SLAM, cooperative mapping, similarity evaluation, map merging.

I. INTRODUCTION

With the rapid development of LIDAR technology, the simultaneous localization and mapping (SLAM) [1] is widely used in the fields of unmanned driving [2] and environmental surveying [3], [4]. The focus of SLAM is to estimate the pose transformation and construct the map by point cloud registration, which is commonly realized by the method of iterative closest point (ICP) [5]. Since the ICP method relies on point cloud matching, it lacks feature information and is expensive to compute. The LIDAR odometry and mapping (LOAM) [6] is an optimized ICP method based on point-line and point-plane, which estimates the LIDAR motion by matching corner points and corner lines, plane points and plane blocks. The fusion of inertial measurement unit (IMU) [7], [8] can improve the localization accuracy of the LIDAR in the case of severe motion. To further reduce the computational complexity of the LIDAR SLAM, the point cloud features are segmented based on LOAM. As well, the point cloud matching is performed by using the segmentation information [9].

To improve the accuracy of indoor 3D mapping, geometric features are often extracted as reliable information of location and mapping. In [10], feature points of initial structural lines are extracted and embedded into the optimization models by different semantic constraints. In [11], arc/line segments-based SLAM is presented by updating accumulated LIDAR data with a mobile robot moving in an unknown environment, which show the good performance in an irregular indoor scenario. References [12] presented an approach to SLAM using the feature-based model, which can efficiently manage the edge feature. In [13], the two-dimensional environment map based on geometric features information was built by LIDAR data when a mobile robot worked in an indoor structured environment.

Those mapping methods described above are applied in simple working scenarios with a single robot. However, the large-scale LIDAR SLAM scenes will increase the storage pressure and accumulate the pose errors [14], especially in large indoor spaces without GNSS/INS [15]. For example, for disaster rescue and post-disaster exploration in indoor environments, the single-robot SLAM will traverse the entire scene to complete the global mapping. Therefore, it is significant to develop an efficient large-scale indoor mapping technology. The representative method to improve the accuracy of mapping is loop detection [16]–[18]. Although loop detection can improve the mapping accuracy, it needs to traverse the whole scene and its computational complexity is increased, therefore, cooperative mapping is usually adopted.

Cooperative mapping [19]–[24] refers to the application of multiple robots to build local maps in different areas of

The associate editor coordinating the review of this manuscript and approving it for publication was Azwirman Gusrialdi.

the scene, then local maps are merged to build a globally consistent map. Since the trajectories of each robot are independent, the difficulty of cooperative mapping lies in how to determine the pose transformation between each local map based on the communication of the robots. In the traditional multi-robot cooperation scheme, the path of each robot is set in advance, in which, the pose transformation between local maps can be obtained directly. However, pre-setting the robot paths is over-idealized, which is difficult to be deployed in many unknown scenarios [25]. Another scheme is to pre-place the landmarks in the scene as the marks of the common areas, in which the local maps are merged by the poses of the robots in the common area. This scheme is inappropriate in the scenes which lack of relevant equipment [26].

This paper proposes a 3D cooperative mapping method for indoor scenarios, which is based on the geometric features of building structures. The proposed method consists of four steps to construct the global map. First, each robot generates its corresponding local map by the LIDAR SLAM algorithm. Second, the geometric features of each LIDAR frame are identified. Third, the most similar LIDAR frame pair between each local map is extracted according to the geometric feature similarity. Fourth, the map transformation matrix between each robot is optimized by the frame pairs in the vicinity of the most similar frame pair, and the global map is constructed.

The main contribution of this paper is the second to fourth steps of the proposed method. A novel feature description of the geometric shape is designed, which is specifically designed to simplify the description of building structure features and enhance the efficiency of the method. The most similar frame pair is extracted by the similarity evaluation according to the geometric features to represent the overlapping region between each local map. The transformation matrix is optimized by the similar frame pairs within the over-lapping region. The proposed method not only reduces the computational complexity of the over-lapping region extraction, but also improve the accuracy of the global map.

II. RELATED WORK

Generally, the cooperative mapping mainly focuses on the map merging, which is divided into two categories: direct map merging and indirect map merging [19].

The direct map merging is to directly compute the pose transformation (PT) between each local map. In the case that the relative locations of the robots are known, such as the robots meet each other at a rendezvous, the PT is obtained directly according to the relative location. References [27] proposed a map merging framework by using the measurements between each robot, in which the PT is obtained when the robots meet each other successfully at the estimated location. References [28] adopted the range measurement to realize the probabilistic map merging when the robots meet each other. The above two methods merge the local maps on the premise that the robots could detect each other, which is not available in all the scenes. References [29] examined the application of multiple robots in SLAM, in which each robot must estimate the poses of all robots in the system, along with the positions of all pre-placed landmarks. In [30], the cameras are adopted to compute the PT by matching the visual objects. As the same visual objects are recognized, their relative locations and orientations are estimated by each robot and used to compute the PT. The direct map merging methods depend heavily on the quality of the sensors and are difficult to be deployed in the real-world scene.

The indirect map merging is to compute the PT by extracting and matching the common area of each local map. References [31] proposed that the local maps are merged with small overlapping areas in relation to the global map based on the maximal empty rectangles. References [32] proposed an approach for the map merging, which can simultaneously merge multi-robot grid maps at different resolutions. References [33] proposed a matching algorithm for the outdoor mapping, in which the GPS location is used as the rendezvous indication and the further extraction of common areas are performed based on the LIDAR. In [34], the LOAM algorithm is first used to generate the local maps. Then, with a three-step matching of ground points, min-z points, and all points in the key frame, the detection on overlapping area and registration of local maps can be realized. This method only uses lidar to realize global map merging with a good mapping effect. However, these methods consume huge computation cost when search for the overlapping areas. In all, the problem of how to improve the efficiency of the cooperative mapping without reduce the mapping accuracy needs to be addressed.

In general, the main mission is to improve the efficiency and precision of the key-frame extraction in overlapping areas. As well, it is especially important to simplify the description of the original point cloud. The detection of the overlapping regions between each local map is to find the most similar frame pairs in overlapping areas. Compared with the loop detection, the overlapped area detection contains more data without any rule to follow. References [35] suggested that general large-scale buildings are consisted of corridors, lobbies, public foyers, and other areas. By analyzing the indoor environment characteristics, the building structure is mainly consisted of walls, corners, columns, and other building structures. A description method based on the simplified structure of the indoor point cloud is proposed by [35] to extract the salient features of the original point cloud. Inspired by the simplified structure loop detection, the 3D cooperative mapping method based on the geometric features of building structures is proposed.

III. COOPERATIVE MAPPING

There are four parts for this method: First, the local map scanned by each robot is constructed by the LEGO-LOAM method. Second, the main feature structure of each frame is extracted by the method of down-sampling, dimensionality reduction, and clustering. The clusters of each frame are described by geometric shapes based on the fitting method. Third, three steps of attribute similarity evaluation, topology similarity evaluation, and ICP evaluation are used to extract the most similar frame pair between local maps. Fourth, a submap pair of the overlapping area is constructed based on the most similar frame pair, which is used to match each local map and generate a global map.

In our cooperative mapping method, we made the following assumptions:

1). Robots are applied to scan different areas of the building.

2). Every two robots meet at most once.

3). All tested scenarios are indoor building environments with smooth grounds, rich building structures, and no stairs.

The flow chart of the cooperative mapping method is shown in Fig. 1.

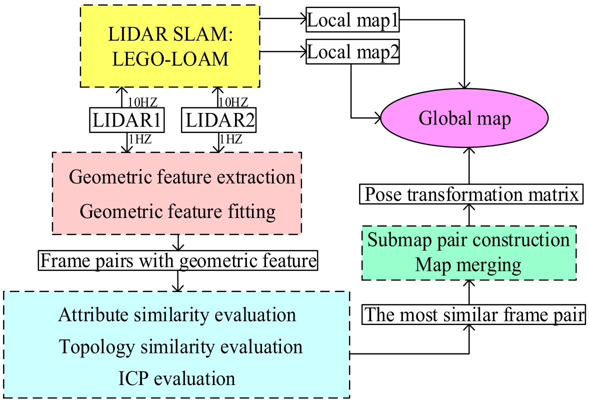

LIDAR SLAM:
LEGO-LOAM
Local map1
Local map2
Global map
10HZ
10HZ
LIDAR1
LIDAR2
1HZ
1HZ
Geometric feature extraction
Geometric feature fitting
Pose transformation matrix
Submap pair construction
Map merging
Frame pairs with geometric feature
Attribute similarity evaluation
Topology similarity evaluation
ICP evaluation
The most similar frame pair

FIGURE 1. The flow chart of the cooperative mapping method.

A. LOCAL MAP CONSTRUCTION

A lightweight LIDAR SLAM method [8] based on ground point optimization is adopted to realize local map construction. The IMU data is applied to remove the motion distortion of the point cloud. Local map construction consists of four parts: point cloud preprocessing, feature extraction, motion estimation, and map construction. The parameters collected by the LIDAR are defined as follows: S represents a LIDAR scan, L_i represents the scan line i in a frame, $P_{(i,j)}$ represents the point from line i and column j in a frame, r represents the Euclidean distance from the point to the LIDAR. The 16-line LIDAR is composed of 16 scan lines in the vertical direction, and the horizontal resolution is set to 1800. The points of each frame are stored in a 16 × 1800 array I_S.

$$I_S = \begin{bmatrix} P_{(1,1)} & \cdots & P_{(1,\mathrm{j})} & \cdots & P_{(1,1800)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ P_{(\mathrm{i},1)} & \cdots & P_{(\mathrm{i},\mathrm{j})} & \cdots & P_{(\mathrm{i},1800)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ P_{(16,1)} & \cdots & P_{(16,\mathrm{j})} & \cdots & P_{(16,1800)} \end{bmatrix} \quad (1)$$

Point cloud preprocessing: points with different attributes are labeled regularly. The empty points are deleted, and the remaining points are divided into ground points and non-ground points by calculating the pitch angle between adjacent scan line, which is

$$\theta_{pitch} = arctan\left(\Delta z, \sqrt{\Delta x^2 + \Delta y^2}\right) \quad (2)$$

where $(\Delta x, \Delta y, \Delta z)$ represents coordinate difference between point pair $P_{(i,j)}$ and $P_{(i+1,j)}$. For the point pair whose θ_{pitch} is less than 10° is marked as ground point G. Then, the non-ground points are segmented to different clusters and labeled as F. The clusters with less than 30 points are regarded as noise points and removed. Each point gets three label attributes after preprocessing: A. category (G or F), B. index value (i, j), C. distance $r_{(i,j)}$.

Feature extraction: the smoothness of a point is obtained by

$$C = \frac{1}{|L_i| \cdot \left\| r_{(i,j)} \right\|} \sum_{k \in L_i, k \neq j} \left\| r_{(i,j)} - r_{(i,k)} \right\| \quad (3)$$

where C is the smoothness of a point $P_{(i,j)}$ in L_i. Through smoothness values, all points are divided into two feature groups: edge points with maximum C values and planer points with minimum C values. I_S is divided into six 16×300 subarrays to evenly extract features from all directions. Then, the points in each line of the subarray are sorted based on their C values. f_e, f_p, F_e and F_p are set to store different feature points. n_{f_e} edge points which do not belong to G and have maximum C values are stored in f_e, n_{f_p} planer points with minimum C values are stored in f_p. For the current frame, n_{F_e} edge points which do not belong to G and have maximum C values are stored in F_e, n_{F_p} planer points which only belong to G and have minimum c values are stored in F_p. These four storage containers satisfy the following relationship: $F_e \subset f_e, F_p \subset f_p$.

Motion estimation: the vector $T_t = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]$ is applied to represent the pose transformation between S_{t-1} and S_t. First, the corresponding feature points of two consecutive frames are extracted from $\left\{f_e^{t-1}, f_p^{t-1}\right\}$ and $\left\{F_e^t, F_p^t\right\}$. For the planer points in F_p^t, only those marked as ground points in f_p^{t-1} will be used to find the corresponding plane blocks. For the edge points in F_e^t, the corresponding edge lines are extracted from f_e^{t-1}. Then, T_t is estimated by a two-step Levenberg-Marquardt (L-M) method. The first step L-M is to estimate $[t_z, \theta_z, \theta_x]$ by matching the planer points in F_p^t with the corresponding plane blocks in f_p^{t-1}. The second step L-M is to estimate $[t_x, t_y, \theta_y]$ with $[t_z, \theta_z, \theta_x]$ as the constraint condition, by matching the edge points in F_e^t with the corresponding edge lines in f_e^{t-1}. T_t is obtained by fusing $[t_x, t_y, \theta_y]$ and $[t_z, \theta_z, \theta_x]$.

Map construction: the feature sets within 50 meters from the current LIDAR position are extracted and fused into a single surrounding map Q^{t-1}.The pose transformation is optimized simultaneously at a low frequency. The transform between the frames is estimated and the local map is constructed based on the original coordinate of robot traversing.

B. EXTRACTION AND FITTING OF GEOMETRIC FEATURES

To improve the extraction efficiency and accuracy of similar frame pairs between local maps, the frames are simplified by geometric features. The relationship between the building structure and geometric features is shown in Fig. 2. Fig. 2(a) shows the real scene of building structures, Fig. 2(b) shows the geometric features corresponding to the building structures.

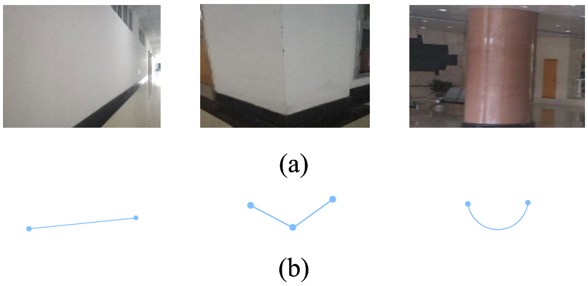


FIGURE 2. The description of geometric features. (a) Real scene. (b) Geometric feature.

Geometric features are mainly divided into three categories: the straight walls are represented by line segments, the wall corners and the rectangular columns are represented by polylines, the arc-shaped structures are represented by arcs.

The extraction and fitting of geometric features are performed simultaneously with local map construction. Since the robot moves slowly and the LIDAR works at a frequency of 10HZ, the scenes collected by LIDAR in one second are almost the same. To reduce the computational load, one frame is extracted from the newly scanned ten frames for the extraction and fitting of geometric features.

1) EXTRACTION OF GEOMETRIC FEATURES

The extraction of geometric features, including z-axis correction, filtering, dimensionality reduction, down-sampling, and segmentation.

First, the ground point G in original frame is extracted and the plane equation $ax + by + cz + d = 0$ is obtained by plane fitting, where (a, b, c) is the normal vector of the plane, and $|d|$ represents the theoretical distance from the LIDAR to the ground. The z-axis of the original point cloud is corrected by

$$P_{calib} = R_{calib}P \tag{4}$$

where P is the original points of a frame and P_{calib} is the points after z-axis correction. R_{calib} indicates the rotation matrix between (a, b, c) and $(0, 0, 1)$.

Second, the point cloud filtering of P_{calib} is performed. Environmental structures such as the floors, ceilings, and beams are deleted by height threshold filtering as

$$P_{z-filter} = \{P_{calib}.z| \, Z_{min} < P_{calib}.z < Z_{max}\} \tag{5}$$

where $Z_{\min}$ and $Z_{\max}$ are height thresholds. $P_{z-filter}$ indicates the points after height threshold filtering. The number of points in the frame is reduced to thousands by

$$P_f = P_{z-filter}{}^{voxel\_grid1(15\times15\times15)} \tag{6}$$

where $voxel\_grid1$ is a voxel grid of size 15cm×15cm×15cm. P_f indicates the points after voxel filtering.

Third, the dimensionality reduction is performed on P_f by orthographically projecting the points onto the XOY plane. The projected points P^{2d} are obtained by

$$P^{2d} = \begin{cases} P^{2d}.x = P_f.x, P^{2d}.x \in [-10, 10] \\ P^{2d}.y = P_f.y, P^{2d}.y \in [-10, 10] \\ P^{2d}.z = 0 \end{cases} \tag{7}$$

where only the points within the area of 20m×20m around the LIDAR are used for projection.

Then, the number of points are reduced into hundreds by down-sampling method. The down-sampled points p are obtained by

$$p = P^{2d^{voxel\_grid2(15\times15)}} \tag{8}$$

Last, the point cloud segmentation is performed by the Euclidean clustering method. There is a two-step filtering to eliminate the clusters which have no contribution to the building structure. First, the clusters that have fewer than 5 points are eliminated. Second, the clusters whose point cloud elevation in the original point cloud is lower than a height threshold $H_{\min}$, such as movable structures like trash cans and public seats are eliminated. The process of point cloud processing above is shown in Fig. 3.

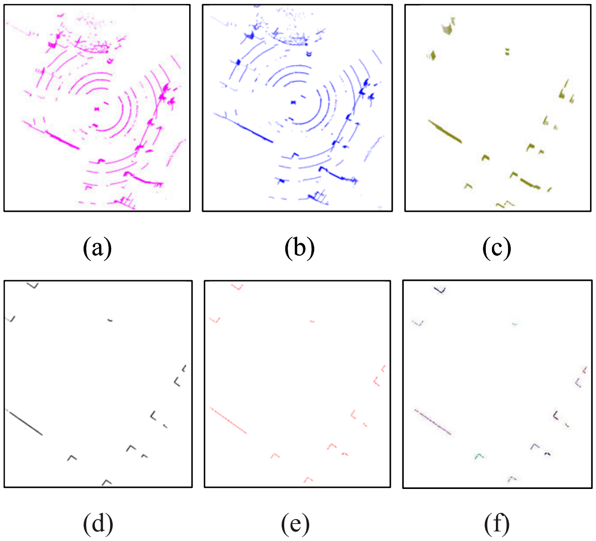


FIGURE 3. The process of point cloud processing. (a) Original points. (b) Z-axis correction. (c) Filtering. (d) Dimensionality reduction. (e) Down-sampling. (f) Segmentation.

2) FITTING OF GEOMETRIC FEATURES

The fitting of geometric features is performed on the segmented 2D point cloud, and the geometric features are divided into six categories, as shown in Fig. 4. Fig. 4(a) shows the line feature represented as l_w. Fig. 4(b) shows the

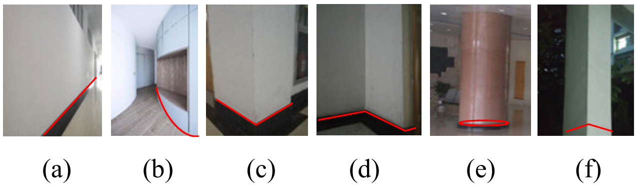


FIGURE 4. Geometric feature classification. (a) Straight line of wall. (b) Arc of wall. (c) One foot of corner. (d) Multiple feet of corner. (e) Arc of cylinder. (f) Rectangle of cylinder.

arc feature of the wall represented as a_w. Fig. 4(c) shows the polyline feature with one foot point, represented as p_f. Fig. 4(d) shows the polyline feature with multiple foot points, represented as p_F. Fig. 4(e) shows the arc feature of the cylinder represented as a_c. Fig. 4(f) shows the rectangle feature of the cylinder represented as r_c. The cluster to be fitted is represented as $C_{fitting}$.

The least-square method is used to fit the indoor geometric features, which is divided into 4 steps.

Step 1. The linear fitting is performed on all clustered point clouds of one frame by

$$E_l = \sum_{i=1}^{n} (f(x_i) - y_i)^2 \tag{9}$$

where (x_i, y_i) are the point coordinates of $C_{fitting}$, and $f(x)$ is the linear equation. The linear fitting error E_l is minimized by optimizing $f(x_i)$. If E_l is less than the error threshold $error_l$, the geometric feature represented by $C_{fitting}$ is regarded as l_w.

Step 2. The clusters are classified according to the number of points, where the clusters with less than num_{thr} points are defined as $cluster_c$, and the clusters with more than num_{thr} points are defined as $cluster_{w,f}$. $C_{fitting}$ of $cluster_c$ is assumed to be a cylinder structure. The cylinder fitting consists of two parts: the arc fitting and the vertical line fitting. First, the arc fitting is performed by

$$E_c = \sum_{i=1}^{n} \left| (x_i - x_c)^2 + (y_i - y_c)^2 - R^2 \right| \tag{10}$$

where (x_c, y_c) is the center of $C_{fitting}$ and R is the radius of $C_{fitting}$. The arc fitting error E_c is minimized by optimizing (x_c, y_c) and R. If E_c is less than the error threshold $error_c$, the geometric feature represented by $C_{fitting}$ is regarded as a_c. Then, the vertical line fitting is performed by extracting the foot point q_f, upper boundary point q_{up} and down boundary point q_{down} for $C_{fitting}$.

As shown in Fig. 5, for each vertical line, the point closest to LIDAR is the foot point. Besides, the foot point is also the maximum or minimum point in the vertical line. The extraction strategy of boundary points is shown in Table 1.

The fitting of the vertical line is regarded as the merging of two straight lines. l_1 is the line determined by q_f and q_{up} whose slope is $k_{(f,up)}$. l_2 is the line determined by q_f and q_{down} whose slope is $k_{(f,down)}$. The angle θ_f between l_1 and l_2 is verified by

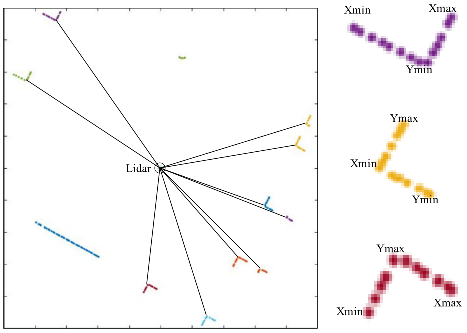


FIGURE 5. The foot point of vertical lines.

TABLE 1. The extraction of boundary points according to feet points.

Number	q_f	q_{up}	q_{down}
1	$q.x_{\min}$	$q.y_{\max}$	$q.y_{\min}$
2	$q.x_{\max}$	$q.y_{\max}$	$q.y_{\min}$
3	$q.y_{\min}$	$q.x_{\max}$	$q.x_{\min}$
4	$q.y_{\max}$	$q.x_{\max}$	$q.x_{\min}$

$$\theta_f = arctan \left| \frac{k_{(f,up)} - k_{(f,down)}}{1 + k_{(f,up)} k_{(f,down)}} \right| \tag{11}$$

The $C_{fitting}$ whose θ_f is within the error threshold θ_{fthr} is regarded as r_c.

Step 3. To sort and fit the follow-up cluster points, the posture of the frame is adjusted. The main line structure of the frame is extracted and its slope k_s is used as the posture adjustment parameter. The main line feature refers to the line feature with the most points, in which, if the frame does not contain the line feature, the vertical line segment with the most points is chosen as the main line feature.

The adjustment is first made by rotation around the center of the LIDAR corresponding to k_s, in which, the rotation direction depends on the positive and negative polarity of k_s. Then, all the features are included in the 10 × 10 area by translation. The results of frame posture adjustment are shown in Fig. 6.

Step 4. It is assumed that the clustering sets with large fitting errors above, and $cluster_{w,f}$ are arc of wall or polyline of corner. First, arc fitting is performed. The $C_{fitting}$ whose arc fitting error is less than the error threshold $error_c$ is regarded as a_w.

Then, the polyline of corner is fitted. As shown in Fig. 7, the original unordered cluster points are sorted according to their coordinate values. The sharpness s_i of the point i in $C_{fitting}$ is calculated by

$$s_i = \arccos \left[\frac{(L_i^-)^2 + (L_i^+)^2 - (L_i^*)^2}{2L_i^- \times L_i^+} \right] \tag{12}$$

where L_i^-, L_i^+, and L_i^* are three edges composed of three points in Fig. 7. If s_i is within the sharpness threshold s_{thr}, point i is considered as a foot point initially. Line segments are fitted by all candidate foot points. If the angle between

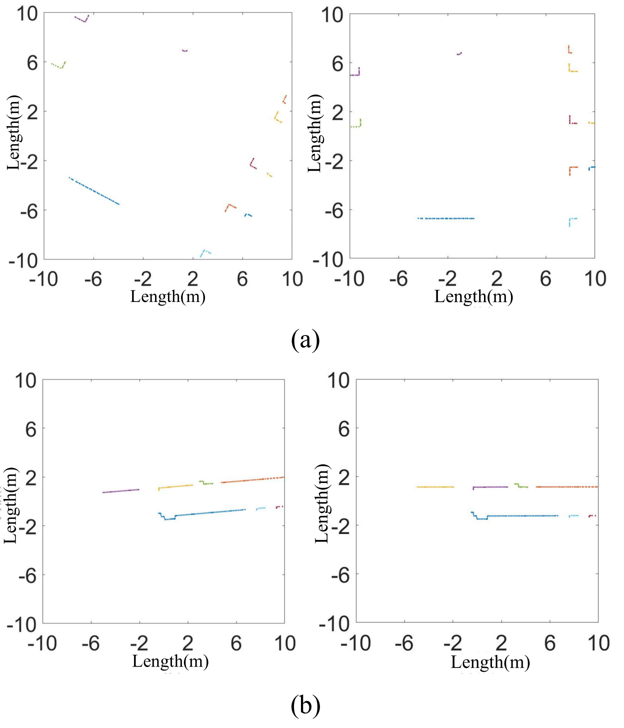


FIGURE 6. Posture adjustment of frames. (a) Posture adjustment of frame 1. (b) Posture adjustment of frame 2.

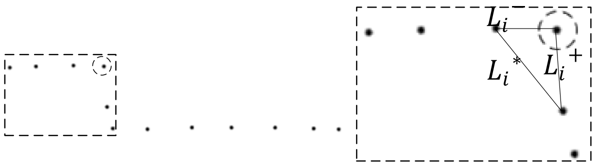


FIGURE 7. The foot point of polyline.

two adjacent line segments does not exceed the polyline angle threshold θ_c, the point of this angle is determined as a foot point. If the number of the foot point is one, the $C_{fitting}$ that satisfies the polyline fitting standard is regarded as p_f. If the number of the foot point is more than one, the $C_{fitting}$ that satisfies the polyline fitting standard is regarded as p_F.

The $C_{fitting}$ with poor error from all the fitting steps is regarded as noise points and eliminated.

C. SIMILARITY EVALUATION

After the previous section, the geometric features of each frame are obtained. In this section, the geometric features are applied to extract the most similar frame pairs between the two local maps by the similarity evaluation.

1) ATTRIBUTE SIMILARITY EVALUATION

The attribute vector $G_{vector} = [a, b, c, d, e, f]$ consisted of geometric features is applied to evaluate the attribute similarity of two frames. In this vector, a represents the number of l_w, b represents the number of a_w, c represents the number of p_f, d represents the number of p_F, e represents the number of a_c, f represents the number of r_c. For the two frames to be evaluated, the geometric feature attribute vectors

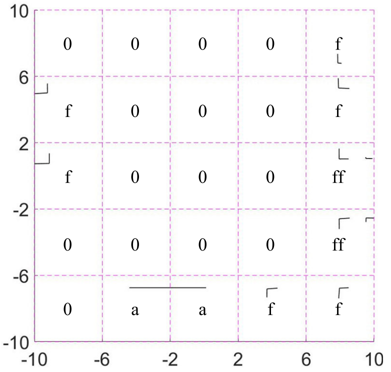


FIGURE 8. Distribution matrix E_d.

are constructed according to their geometric feature fitting results by

$$X = [x_1, x_2, x_3, x_4, x_5, x_6], Y = [y_1, y_2, y_3, y_4, y_5, y_6] \quad (13)$$

where X is the G_{vector} of the frame from robot1, Y is the G_{vector} of the frame from robot2. The attribute similarity of X and Y is evaluated by

$$\rho_{X,Y} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - E^2(X)}\sqrt{E(Y^2) - E^2(Y)}} \quad (14)$$

$$\sigma_{X,Y} = \frac{\sum x_i \cap y_i}{6} \quad (15)$$

$$P_{av} = \lambda \rho_{X,Y} + \mu \sigma_{X,Y} \quad (16)$$

where $\rho_{X,Y}$ is the Pearson correlation analysis between X and Y, $\sigma_{X,Y}$ is the geometric feature existence analysis between X and Y, P_{av} is the result of attribute similarity evaluation. The frame pairs whose scene structures are quite different or the number of geometric features is quite different are eliminated by P_{av}.

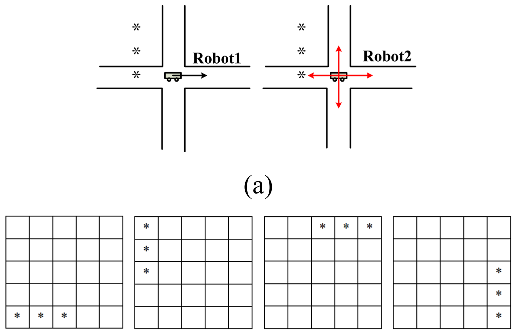
2) TOPOLOGY SIMILARITY EVALUATION

The topology similarity of the frame pairs selected from attribute similarity evaluation is evaluated, which includes direction evaluation and distribution evaluation.

First, the qualitative description method was used to describe the orientation relationship to evaluate the direction similarity. The directional relationship between the line segments of the frame is described as parallel, vertical and intersecting direction. The frame pairs consisted of different orientation properties are eliminated.

Second, the distribution similarity is evaluated. It shows from Fig. 8 that the distribution matrix of 5 × 5 is applied to represent each frame to be evaluated. The geometric feature is filled into the distribution matrix E_d corresponding to the index position. In the same real scene, the distribution matrix detected by the robot has four possibilities as shown in Fig. 9.

For the two distribution matrices E_{d1} and E_{d2} to be evaluated, the matrix postures are adjusted by determining the positions of the geometric features with the most points in E_{d1} and E_{d2}, so that the main geometric features in the two matrices appear at similar matrix index positions. Then,



(a)

(b)

FIGURE 9. Four possibilities of distribution matrix. (a) Robot movement direction. (b) The distribution matrices of different movement direction.

the distribution similarity is calculated by

$$ {}_5^5P_{dv} = \frac{\sum E_{d1} \cap E_{d2}}{25} \tag{17}$$

where ${}_5^5P_{dv}$ is the distribution similarity score between E_{d1} and E_{d2}. Since the positions of the robots do not coincide completely when they pass through the same region, the distribution matrices observed by the robots are not the same. Thus, the submatrices of E_d are used to deal with this problem as shown in Fig. 10.

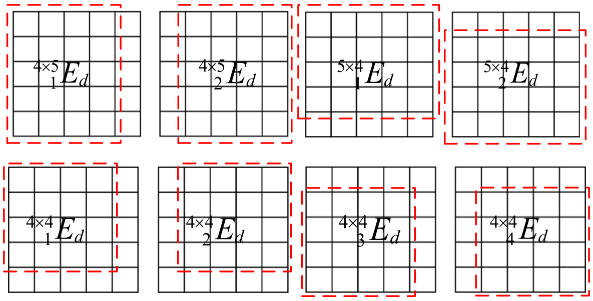


FIGURE 10. Submatrices.

The distribution similarity of the above submatrices was evaluated by

$$ {}_5^4P_{dv} = max(\frac{\sum {}_1^{4\times5}E_{d1} \cap {}_2^{4\times5}E_{d2}}{20}, \frac{\sum {}_2^{4\times5}E_{d1} \cap {}_1^{4\times5}E_{d2}}{20}) \tag{18}$$

$$ {}_4^5P_{dv} = max(\frac{\sum {}_1^{5\times4}E_{d1} \cap {}_2^{5\times4}E_{d2}}{20}, \frac{\sum {}_2^{5\times4}E_{d1} \cap {}_1^{5\times4}E_{d2}}{20}) \tag{19}$$

$$ \begin{aligned} {}_4^4P_{dv} = max(&\frac{\sum {}_1^{4\times4}E_{d1} \cap {}_2^{4\times4}E_{d2}}{16}, \frac{\sum {}_1^{4\times4}E_{d1} \cap {}_3^{4\times4}E_{d2}}{16}, \\ &\frac{\sum {}_1^{4\times4}E_{d1} \cap {}_4^{4\times4}E_{d2}}{16}, \frac{\sum {}_2^{4\times4}E_{d1} \cap {}_1^{4\times4}E_{d2}}{16}, \\ &\frac{\sum {}_2^{4\times4}E_{d1} \cap {}_3^{4\times4}E_{d2}}{16}, \frac{\sum {}_2^{4\times4}E_{d1} \cap {}_4^{4\times4}E_{d2}}{16}, \\ &\frac{\sum {}_3^{4\times4}E_{d1} \cap {}_1^{4\times4}E_{d2}}{16}, \frac{\sum {}_3^{4\times4}E_{d1} \cap {}_2^{4\times4}E_{d2}}{16}, \\ &\frac{\sum {}_3^{4\times4}E_{d1} \cap {}_4^{4\times4}E_{d2}}{16}, \frac{\sum {}_4^{4\times4}E_{d1} \cap {}_1^{4\times4}E_{d2}}{16}, \\ &\frac{\sum {}_4^{4\times4}E_{d1} \cap {}_2^{4\times4}E_{d2}}{16}, \frac{\sum {}_4^{4\times4}E_{d1} \cap {}_3^{4\times4}E_{d2}}{16}) \end{aligned} \tag{20}$$

where ${}_5^4P_{dv}$ is the distribution similarity score of ${}^{4\times5}E_d$, ${}_4^5P_{dv}$ is the distribution similarity score of ${}^{5\times4}E_d$, ${}_4^4P_{dv}$ is the distribution similarity score of ${}^{4\times4}E_d$. The distribution similarity was evaluated by fusing ${}_5^5P_{dv}$, ${}_5^4P_{dv}$, and ${}_4^4P_{dv}$ as follows:

$$ P_{dv} = \alpha {}_5^5P_{dv} + \beta {}_5^4P_{dv} + \gamma {}_4^5P_{dv} + \delta {}_4^4P_{dv}, \alpha + \beta + \gamma + \delta = 1 \tag{21}$$

where P_{dv} is the final score of the distribution similarity evaluation, and α, β, γ, δ are the corresponding weights. The weights are initially set by subjective experience according to the size of the matrix, and then, the feasibility of the weights is verified through the experiments.

3) ICP EVALUATION

The similar frame pairs with the highest scores are selected by utilizing the topology similarity evaluation. Although the ICP evaluation is more computationally expensive than the previous two steps, the number of survived frame pairs is very small and the time cost in ICP evaluation is still little. After that, the most similar frame pair between two robots is obtained to interpret their overlapping region. Finally, the most similar frame pair is selected to merge each local map.

D. MAP MERGING

The global map merging is the last step of the cooperative mapping method. For this, the pose transformation between the local maps is needed, which can be obtained from the most similar frame pair discussed in Section C. In addition to the frame pair with the highest similarity, there are many frame pairs with high similarity scores among the overlapping area that robots pass through. In this section, an accurate map merging method is proposed by using the submap data from the overlapping regions.

Frame pairs with high similarity near the most similar frame pair are selected to construct a submap pair. The size of the submap pair is not limited, in which, each similar pair is extracted to calculate a pose transformation matrix. To minimize the error of the map transformation matrix, the weight of each pose transformation matrix is set by

$$ G_i = \frac{1}{ICP_i^{score}} \tag{22}$$

$$ T^{map} = \sum (\frac{G_i}{\sum G_i} T_i) \tag{23}$$

where ICP_i^{score} is the ICP score of the similar frame pair i, T_i is the pose transformation matrix of the frame pair i, $\frac{G_i}{\sum G_i}$ is the weight of T_i, and T^{map} is the map transformation matrix between the local maps. The number of the similar frame pairs is adjusted to obtain the optimal T^{map}, which can minimize the matching error of the submap pair. After transforming the local maps through T^{map}, all points of two local maps will share the same coordinate system, and the global map is merged.

IV. EXPERIMENT AND ANALYSIS

There are several identical robotic vehicles used in our experiment, as Fig. 11 shows. Each robot is equipped with a VLP-16 LIDAR and a LPMS-IG1 IMU. The robots have the length of 1.00m, width of 0.72m. The installation height of the LIDAR is 0.85m and the vertical FOV of the LIDAR is $-15^\circ \sim 15^\circ$. The robots run at a speed up to $1.3m/s$. The experimental data are processed by the robot operating system (ROS). The robots communicate via multiple machines under the ROS platform. We first analyze our method based on two robots in Sections A to D, and then, the analysis of our method based on more robots is discussed in Section E.

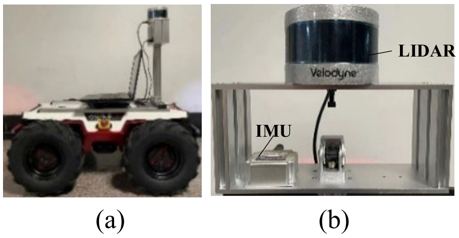


FIGURE 11. The equipment. (a) The robot. (b) LIDAR and IMU.

A. EXPERIMENTAL DISCRIPTION

The experimental parameters are divided into three parts, which are geometric feature extraction module (GFE), geometric feature fitting module (GFF), and similarity evaluation module (SE). The specific parameters are shown in Table 2, where the threshold values of different parameters are obtained by field measurement of the real scenario and data test of the collected datasets. It was verified that these parameters change little for different indoor environments.

Considering the unsolvable problem of environmental degradation, and the necessity to verify the universality of geometric features, it is pointless to experiment in a single-structured scenario, so three scenarios with universal structures are chosen. The experiment is verified in the laboratory building (Scenario 1), the teaching building (Scenario 2), and the parking lot (Scenario 3) with three different scenarios, as shown in Fig. 12.

Dataset 1 is collected in the laboratory building, where robot1 passed through the enclosed corridor, lobby with 502 frames, and robot2 passed through the lobby, open corridor with 537 frames. Dataset 2 is collected in the teaching building, where robot1 passed through the lobby, enclosed corridor, aisle with 1279 frames, and robot2 passed through the lobby, open corridor with 428 frames. Dataset 3 is collected in the parking lot, where robot1 passed through the channel, parking area with 360 frames, and robot2 passed through the parking area with 377 frames.

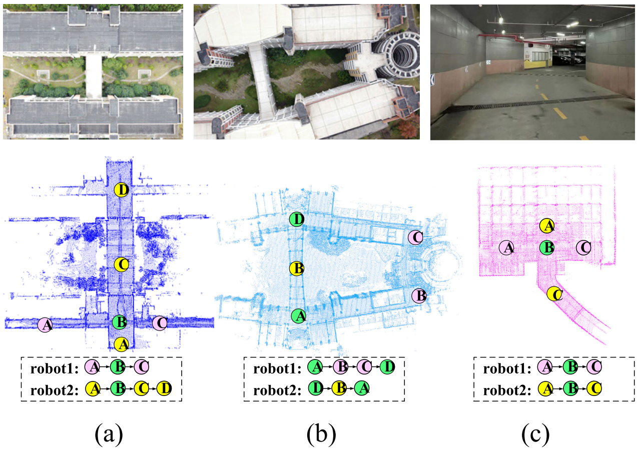


FIGURE 12. (a) Scenario 1: laboratory building. (b) Scenario 2: teaching building. (c) Scenario 3: parking lot. The pink circle is the area which robot1 passed through; the yellow circle is the area which robot2 passed through; the green circle is the overlapping area.

TABLE 2. Experimental parameters.

Module	Parameter	Value
GFE	$Z_{min} \sim Z_{\max}$	$0.1m \sim 2.5m$
	$H_{\min}$	$1m$
	num_{thr}	15
GFF	$error_l = \sqrt{\dfrac{E_l}{n-1}}$	0.02
	$error_c = \sqrt{\dfrac{E_c}{n-1}}$	0.1
	$\theta_{f_{thr}}$	$80^\circ \sim 100^\circ$
	s_{thr}	$60^\circ \sim 120^\circ$
	θ_c	$60^\circ \sim 120^\circ$
SE	Deletion ratio of GFE	$80\% \sim 85\%$
	Deletion ratio of GFF	$\geq 99\%$
	Weights of GFE	$\lambda = 0.5, \mu = 0.5$
	Weights of GFF	$\alpha = 0.4, \beta = 0.2,$ $\gamma = 0.2, \delta = 0.2$

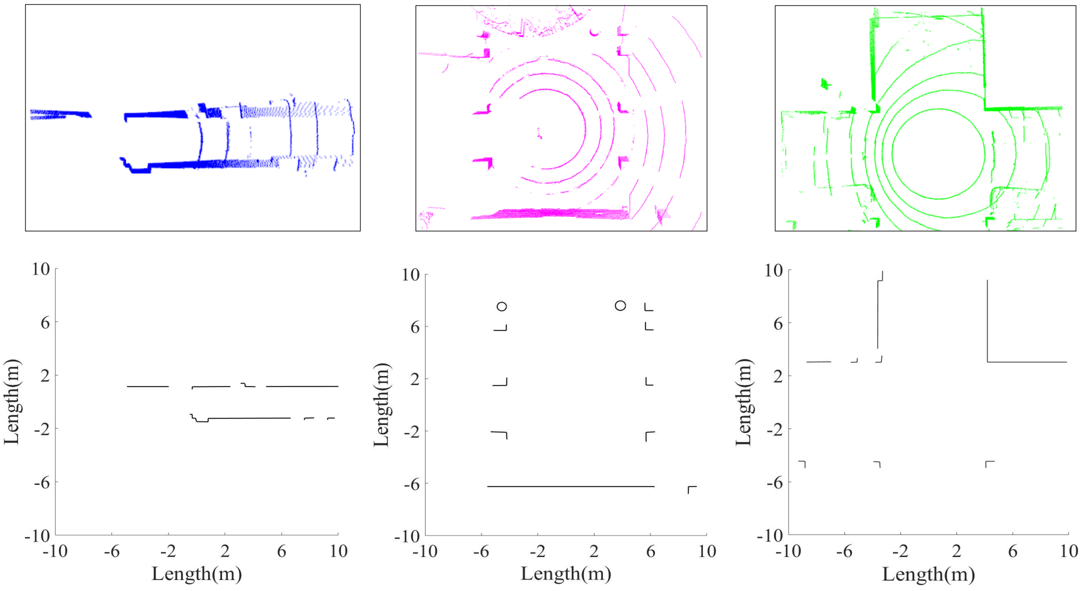
B. ANALYSIS OF GEOMETRIC FEATURE

The geometric feature extraction and fitting (every 10 frames) are performed on the datasets of the above three different scenarios. The number of geometric features extracted from the scenarios is shown in Table 3. Obviously, the number of features extracted in different environments are distinguishable. In scenario 1, the feature numbers of robot1 and robot2 are very different in d and f. In scenario 2, the feature numbers of robot1 and robot2 are very different in a, c, d and e. In scenario 3, the feature numbers of robot1 and robot2 are very different in a, b, c and f.

Then, the geometric features are fitted, and the results are shown in Table 4. Totally 827 line features of the wall are extracted from all scenarios, since the line feature is simple, the fitting accuracy of l_w reached 98.79%. Only 48 arc features of the wall are extracted, the fitting accuracy of a_w is 93.75%. 520 polyline features with one foot are extracted, the fitting accuracy of p_f is 96.54%. 330 polyline features with multiple feet are extracted, the fitting accuracy of p_F is

TABLE 3. The truth number of geometric features.

Scenario		Geometric feature					
		a	*b*	*c*	*d*	*e*	*f*
Scenario 1	Robot1	167	0	114	189	0	0
	Robot2	164	0	132	34	0	160
Scenario 2	Robot1	365	0	208	87	75	266
	Robot2	29	0	22	0	270	76
Scenario 3	Robot1	72	0	29	12	0	126
	Robot2	30	48	15	8	0	24



(a) (b) (c)

FIGURE 13. The visualized results of geometric fitting. (a) Frame 1. (b) Frame 2. (c) Frame 3.

93.94%. 345 arc features of cylinder are extracted, the fitting accuracy of a_c is 97.68%. 652 rectangle features of cylinder are extracted, the fitting accuracy of r_c is 97.55%. Although not all features appear in all scenarios, which causes the system to waste more time on features that do not exist, the accuracy of identification is guaranteed by the fitting method. Very few wrong fitting results are considered that can be ignored.

The visualized result of the geometric fitting is shown in Fig. 13. Among which, Fig. 13(a) is the fitting result of frame 1, which is a frame of the enclosed corridor area in scenario 1. Fig. 13(b) is the fitting result of frame 2, which is a frame of the lobby area in scenario 2. Fig. 13(c) is the fitting result of frame 3, which is a frame of the parking area in scenario 3. The indoor structure detected based on the geometric feature method is relatively complete, and the characteristics of the entire indoor structure can be represented by fewer parameters. It has strong robustness to different indoor environments.

C. ANALYSIS OF SIMILARITY EVALUATION

The analysis of similarity evaluation is carried out from three aspects, which concludes the analysis of attribute similarity evaluation, the analysis of topology similarity evaluation, and the efficiency of similar frame extraction. The frames collected by robot1 is represented as A, the frames collected by robot2 is represented as B.

1) ANALYSIS OF ATTRIBUTE SIMILARITY

The P_{av} score of scenario 1 is listed as an example. For scenario 1, totally 51 × 54 frame pairs are constructed to perform attribute similarity evaluation. As shown in Table 5, the score is of great difference at different part of A and B. The P_{av} score distribution in the [0,1] range of all scenarios is shown in Fig. 14, in which the quantity distinction of different score segments is distinguishable. The percentage of each score segment is shown in Table 6.

The proportion of scores higher than 0.9 is less than 5% in each scene. The score segment with the highest proportion is 0.5∼0.6, which occupies nearly 30% of all scores. The frame pair with the highest P_{av} in scenario 1 is (A_{251}, B_{101}), which scores 0.9925. The frame pair with the highest P_{av} in scenario 2 is (A_{1271}, B_{31}), which scores 0.9801. The frame pair with the highest P_{av} in scenario 3 is (A_{161}, B_{41}), which scores 0.9876.

Since the frames of the wall and corner collected from the corridor has relatively similar scene structures, the attribute similarity score of these two frames is higher. In contrast,

TABLE 4. Fitting result of geometric feature.

Scenario	Result	Geometric feature					
		a	*b*	*c*	*d*	*e*	*f*
Scenario 1	Truth	331	0	246	223	0	160
	Correct fitting	327	0	238	210	0	156
	Wrong fitting	2	2	1	2	3	1
Scenario 2	Truth	394	0	230	87	345	342
	Correct fitting	390	0	222	81	337	334
	Wrong fitting	4	3	1	2	2	2
Scenario 3	Truth	102	48	44	20	0	150
	Correct fitting	100	45	42	19	0	146
	Wrong fitting	2	0	2	0	3	2

TABLE 5. Attribute similarity score of scenario 1.

		Robot 1					
		A_1	A_{101}	A_{201}	A_{301}	A_{401}	A_{501}
Robot 2	B_1	0.5838	0.4037	0.5447	0.5307	0.5513	0.6476
	B_{101}	0.8359	0.5729	0.7823	0.7837	0.5861	0.8283
	B_{201}	0.1468	0.4985	0.3072	0.1811	0.6745	0.3584
	B_{301}	0.1608	0.1120	0.3556	0.1594	0.1858	0.1738
	B_{401}	0.2951	0.4031	0.4651	0.3050	0.6058	0.4942
	B_{501}	0.5906	0.3017	0.5533	0.5192	0.4068	0.5745

TABLE 6. Percentage of score segments.

Score	Percentage (%)		
	Scenario 1	Scenario 2	Scenario 3
0.9~1	2.61	3.03	2.92
0.8~1	7.66	8.48	6.87
0.7~1	13.00	15.19	13.82
0.6~1	25.74	28.13	26.61
0.5~1	51.82	53.05	55.85
0.4~1	67.28	69.53	71.13
0.3~1	80.17	82.07	81.36
0.2~1	90.16	89.64	89.55
0.1~1	97.82	97.81	98.03
0~1	100	100	100

For the two frames with different scene structures, such as one frame collected from the corridor while the other frame collected from the lobby, the attribute similarity score of these two frames is low because the cylinder structure of the lobby is not found in the corridor. For two frames collected in similar scenes, the difference in the number of geometric features also affects the attribute similarity evaluation.

The frame pairs from different indoor spaces and the frame pairs with large differences in the number of features are eliminated through the attribute similarity evaluation. The frame pairs whose P_{av} scores ranked top 15% are kept to conduct the following evaluation.

2) ANALYSIS OF TOPOLOGY SIMILARITY

The frame pairs with the different directional attributes are eliminated. The remaining frame pairs are used for distribution similarity evaluation. The result of the distribution similarity evaluation is shown in Table 7-9. In Table 9, the frame pair ranked 125 of scenario 3 is eliminated by direction similarity evaluation. In general, the frame pair with a high P_{av} score obtained a high P_{dv} score, while the frame pair with a low P_{av} score obtained a low P_{dv} score. There are a few frame pairs with a high P_{av} score but a low P_{dv} score.

The frame pair with the highest P_{dv} in scenario 1 is (A_{251}, B_{51}), which scores 0.9850. The frame pair with the highest P_{dv} in scenario 2 is (A_{1271}, B_{11}), which scores 0.9688. The frame pair with the highest P_{dv} in scenario 3 is (A_{161}, B_{41}), which scores 0.9630.

The topology similarity evaluation analyzes the frame pairs from the two aspects of the direction and distribution of geometric features, which can eliminate the frame pairs with high attribute similarity but large difference in feature distribution. This is a supplement based on the attribute similarity evaluation.

In theory, the geometric features of the overlapping regions are identical, but due to the posture difference of the robots through the overlapping regions, there may be small differences in the geometric features they observed. But our similarity evaluation of overlapping regions is quantitative, the result of P_{av} shows that even for two frames in overlapping areas, the attribute similarity score is not necessarily a full score. In general, two frames of similar regions get higher scores in both P_{av} and P_{dv}. To ensure the accuracy of the extraction, the ICP is used to extract the most similar frame pairs. The final extraction results after ICP evaluation for each scenario are (A_{251}, B_{51}), (A_{1271}, B_{11}) and (A_{161}, B_{41}). Although the similarity evaluation has extracted a good result, the ICP evaluation is to increase the robustness of the extraction system.

3) EFFICIENCY OF SIMILAR FRAME EXTRACTION

For scenario 1 to 3, the quantities of remaining frame pairs after each evaluation step are shown in Table 10. Although the ICP evaluation costs about 5 seconds, the similarity evaluation extracts the most similar frame pair with little

TABLE 7. P_{dv} **score of scenario 1.**

P_{av} ranking	1	50	100	150	200	250	300	350	400
P_{av}	0.9925	0.9328	0.8821	0.8531	0.8104	0.7804	0.7423	0.7144	0.6883
P_{dv}	0.9250	0.8062	0.7562	0.5355	0.6132	0.5920	0.6231	0.5987	0.5578

TABLE 8. P_{dv} **score of scenario 2.**

P_{av} ranking	1	100	200	300	400	500	600	700	800
P_{av}	0.9801	0.9304	0.8721	0.8452	0.8215	0.7822	0.7645	0.7347	0.7095
P_{dv}	0.9152	0.8548	0.6542	0.6319	0.5336	0.5741	0.5534	0.5476	0.4741

TABLE 9. P_{dv} **score of scenario 3.**

P_{av} ranking	1	25	50	75	100	125	150	175	200
P_{av}	0.9876	0.9408	0.8537	0.8281	0.7904	-	0.7267	0.7095	0.6875
P_{dv}	0.9630	0.9067	0.7143	0.6132	0.5984	-	0.5765	0.4961	0.5035

TABLE 10. **Quantities of remaining frame pairs.**

		Quantities of remaining frame pairs		
Scenario	Origin	Attribute	Direction	Distribution
Scenario 1	2754	420	408	10
Scenario 2	5504	840	812	10
Scenario 3	1368	210	201	10

time. The efficiency of similar frame extraction is shown in Table 11.

D. ANALYSIS OF MAP MERGING

By adjusting the submap pair, the numbers of similar frames used to calculate the map transformation matrix are 7 for scenario 1, 4 for scenario 2, and 6 for scenario 3. The number of similar frames is affected by the motion of the robots.

The criterion for evaluating the accuracy of mapping is to judge whether the LIDAR positioning is accurate. The single-robot mapping is set to compare the accuracy of single-robot positioning and multi-robot positioning. The single robot traversed the environment with the same starting point of robot1 and the same ending point of robot2. The mapping paths of single-robot and multi-robot are shown in Fig. 15, where the dotted circle area represents the end of the mapping path. The ending points in the dotted circle are shown in Fig. 16, where the ground truth of the ending points is set in advance. The positioning error analysis of ending

TABLE 11. **The efficiency of each step.**

		Attribute evaluation		Topology evaluation		ICP evaluation	
Scenario	pairs	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)
Scenario 1	2754	84.75	0.692	99.64	0.190	99.96	4.988
Scenario 2	5504	84.74	0.920	99.82	0.224	99.98	5.693
Scenario 3	1368	84.65	0.306	99.27	0.159	99.93	5.009

TABLE 12. **The positioning error analysis of end points.**

Scenario	Distance (m)	Ground truth	Our method			Single-robot		
		Positioning (m)	Positioning (m)	Error (m)	Relative accuracy (%)	Positioning (m)	Error (m)	Relative accuracy (%)
Scenario 1	152.276	(30,53,0)	(29.819, 52.782, -0.148)	0.320	0.210	(29.550, 52.666, -0.112)	0.571	0.375
Scenario 2	211.816	(0,0,0)	(-0.115, -0.294, -0.223)	0.387	0.183	(-0.472, -0.492, -0.231)	0.720	0.340
Scenario 3	89.431	(47,-34,1)	(47.170, -34.127, 0.927)	0.224	0.250	(47.396, -34.189, 0.887)	0.453	0.507

TABLE 13. The analysis of relative accuracy.

Scenario	Relative accuracy of our method (%)	Relative accuracy of 3D-CM (%)
Scenario 1	0.210	0.292
Scenario 2	0.183	0.256
Scenario 3	0.250	0.353

TABLE 14. The efficiency of two-robot mapping.

Scenario	Method	Time（s）
Scenario 1	Single-robot mapping	124
	Our method	74
	3D-CM	80
Scenario 2	Single-robot mapping	166
	Our method	145
	3D-CM	150
Scenario 3	Single-robot mapping	72
	Our method	53
	3D-CM	59

TABLE 15. The efficiency and accuracy of multi-robot mapping in scenario 3.

	Case	Time（s）	Relative accuracy (%)
2 robots		53	0.250
3 robots	1	43	0.242
	2	46	
4 robots	1	32	0.237
	2	38	

points is shown in Table 12, where the distance represents the path length of single-robot mapping and the relative accuracy is the ratio of error and distance.

The result in Table 12 shows that the error of the ending point positioning is decreased by multi-robot mapping. Compared with a single-robot positioning, the relative accuracy of our method is approximately doubled.

The existing method 3D-CM [34] realizes global map merging with a good mapping effect based on ground points, min-z points, and all points, which consumes huge computation cost when search for the overlapping areas. Our task is to increase the speed of mapping without sacrificing the accuracy of mapping. Table 13 shows the relative accuracy of our method and 3D-CM, in which, our method gets a better accuracy. The result of two-robot cooperative mapping is shown in Fig. 17, where Fig. 17(a) shows the most similar frame pair. Fig. 17(b) shows the merged map of local maps. The efficiency of our method is shown in Table 14, the speed of mapping has been improved.

E. COOPERATIVE MAPPING WITH MORE ROBOTS

To verify the effect of this method with more robots, three and four robots are invested in scenario 3. Since not all robots share overlapping areas in a multi-robot collaborative system, the efficiency of cooperative mapping depends on whether the robot number corresponding to each overlapping area is pre-known. The experiment is divided into case 1 where the robot index in the overlapping area is pre-known and case 2 where the robot index in the overlapping area is unknown. Case 1 is equivalent to the merging of several two-robot cooperative mapping, while in case 2, it is necessary to establish an overlapping area detection and judgment mechanism between each two-robot. The threshold of the ICP score is set to determine whether there is an overlapping area between the two robots, in which, only the frame pairs that satisfy the ICP score lower than 0.5 are extracted for map merging. The scanning areas of three and four robots in scenario 3 are shown in Fig. 18.

The result of the multi-robot cooperative mapping is shown in Fig. 19. The efficiency of multi-robot cooperative mapping in scenario 3 is shown in Table 15. Although the multi-robot system has completed the mapping in both cases, there is a difference in efficiency, where case 2 costs extra time to determine whether there is an overlapping area between

TABLE 16. The improvement of our method compared with the single-robot mapping and the existing cooperative mapping method.

Scenario	Method for comparison	Improvement of relative accuracy (%)	Improvement of efficiency (%)
Scenario 1	Single-robot mapping	44.0	40.3
	The existing method	28.1	7.5
Scenario 2	Single-robot mapping	46.2	12.7
	The existing method	28.6	3.3
Scenario 3	Single-robot mapping	50.7	26.4
	The existing method	29.1	10.2

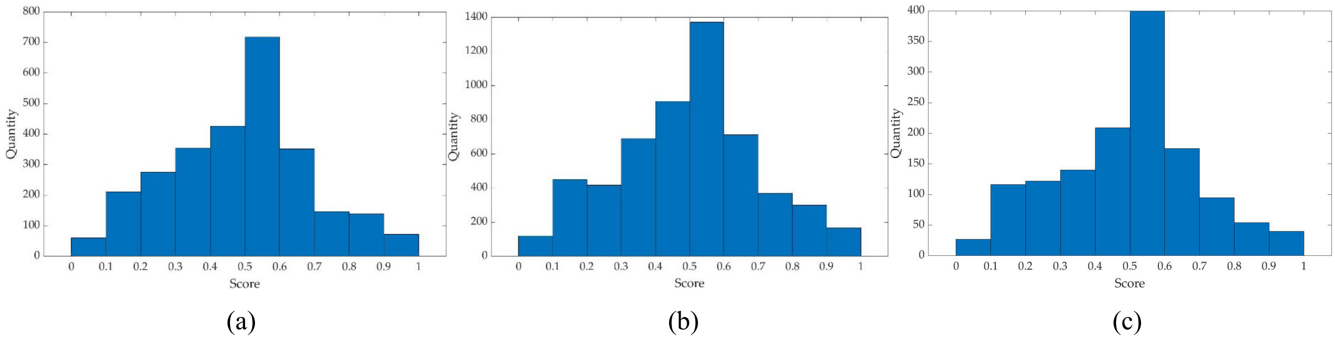


FIGURE 14. P_{av} score distribution: (a) The P_{av} distribution of scenario 1. (b) The P_{av} distribution of scenario 2. (c) The P_{av} distribution of scenario 3.

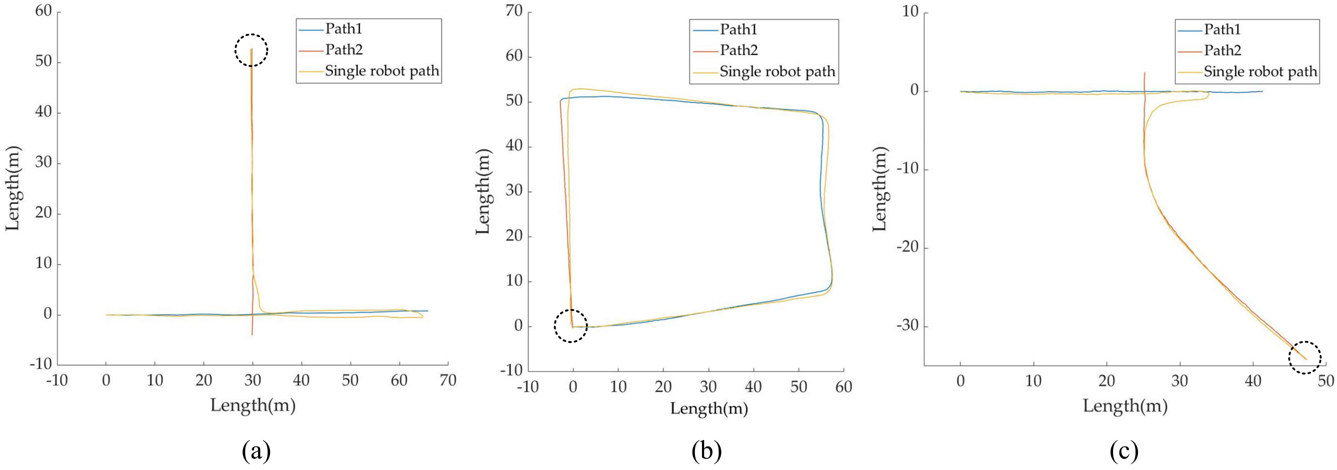


FIGURE 15. The paths of multi-robot system and single-robot system. (a) The path of scenario 1. (b) The path of scenario 2. (c) The path of scenario 3.

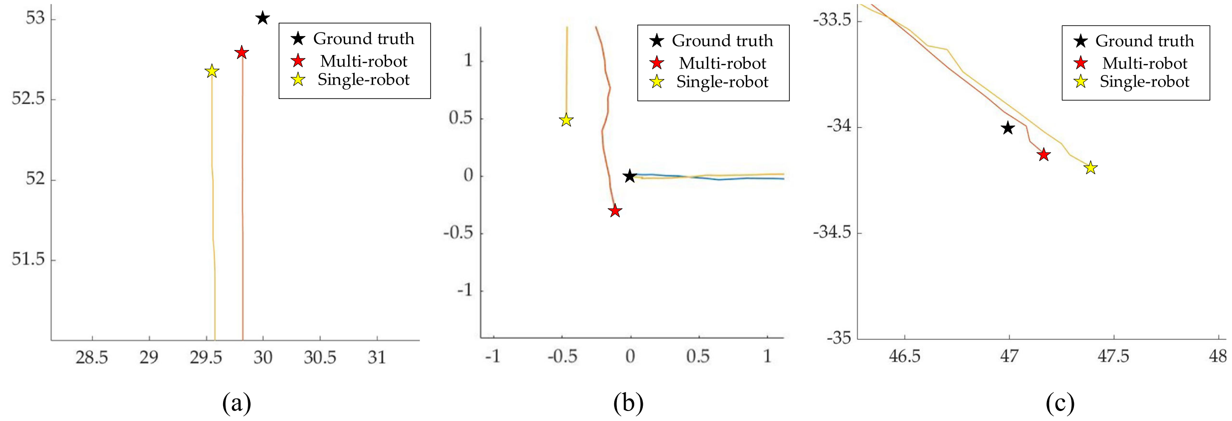


FIGURE 16. The ending points. (a) The ending points of scenario 1. (b) The ending points of scenario 2. (c) The ending points of scenario 3.

the two robots. For case 2, there are $n(n-1)/2$ possible overlapping areas in the n-robot system. With the increase in the number of robots, the number of possible overlapping areas also increases, and the computational pressure caused is greater. Therefore, the efficiency of mapping is not positively correlated with the number of robots, where the optimal number of robots depends on the size of the scenario.

The accuracy of multi-robot cooperative mapping in scenario 3 is shown in Table 15, where the relative accuracy of more robots is improved slightly. Although investing more robots can get better mapping accuracy, the cumulative error is unavoidable. It is the key to investigate the size and structure of the environment and set the number and scanning plan of the robots in advance to get a balance between the mapping efficiency and accuracy.

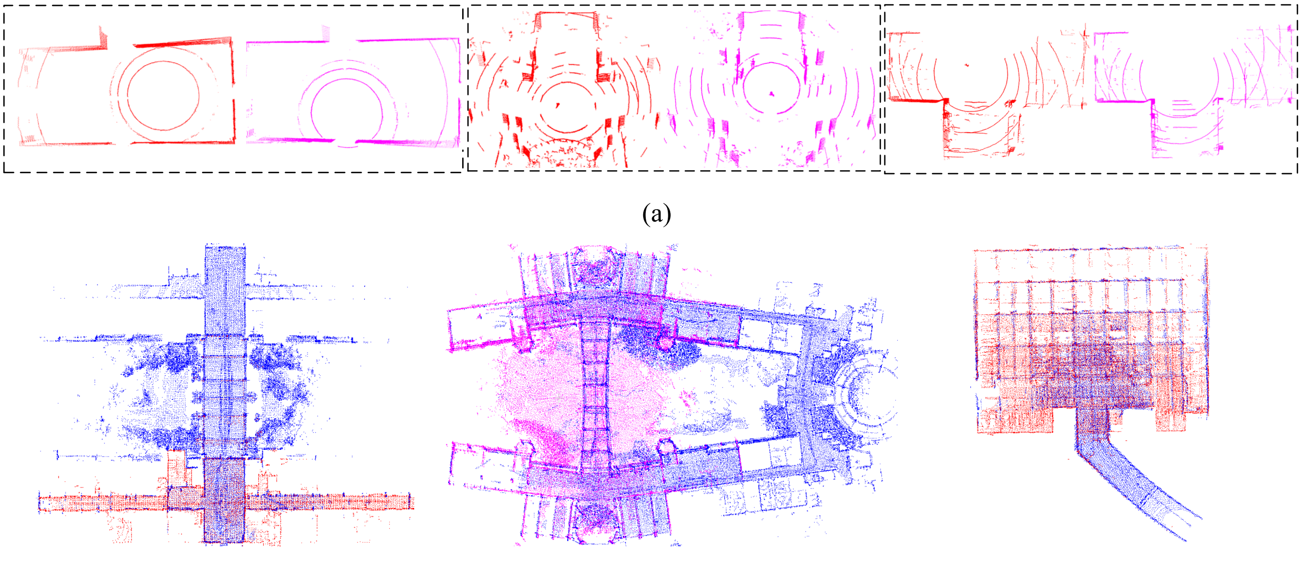


FIGURE 17. Two-robot cooperative mapping. (a) The most similar frame pair. (b) The merged map.

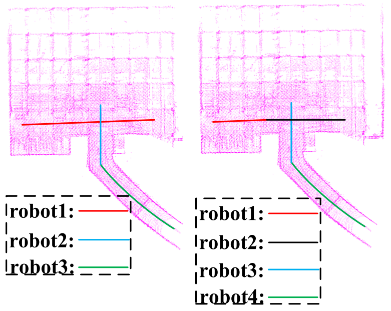


FIGURE 18. The scanning areas. (a) Three robots. (b) Four robots.

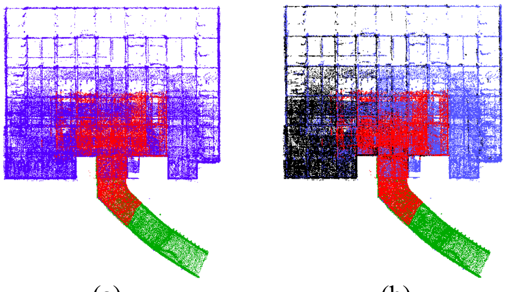


FIGURE 19. Multi-robot cooperative mapping. (a) Three robots. (b) Four robots.

V. CONCLUSION

Aiming at improving the accuracy and efficiency of indoor LIDAR mapping, this paper proposes a multi-robot cooperative mapping method based on geometric features. The local map is constructed by LEGO-LOAM in advance.

(1) To simplify the feature description of each frame, our suggested method can extract the main features of the indoor building structure such as wall, corner, and cylinder. The features are classified into six types based on the geometric features of line, polyline, and circle by the least square fitting. The classification accuracy of features exceeds 90% by the multi-step fitting strategy.

(2) To extract the most similar frame pair between two local maps, a three-step gradual extraction strategy is adopted, which consists of the attribution similarity evaluation, the topology similarity evaluation, and the ICP evaluation. The geometric features are used in the strategy to speed up the similar frame extraction. The cost of the most similar frame extraction in the experiment is less than 10 seconds. The ICP evaluation guarantees the accuracy of the extraction. In general, the robustness and efficiency of the extraction system is improved by the three-step strategy.

(3) The local maps are merged through the submap pair constructed around the most similar pair. The positioning error of merged map decreases apparently.

The experiment is first conducted in three scenarios based on two robots, the improvement of our method compared with the single-robot mapping method and the existing cooperative mapping method is shown in Table 16. Then, the experiment is conducted in scenario 3 based on three and four robots, which achieves a degree of improvement in both accuracy and efficiency.

In the future, more robots will be invested to search for the optimum number of robots, which can best balance the mapping efficiency and the mapping accuracy. Furthermore, more robots with heterogeneous sensors will be considered in this method.

REFERENCES

[1] M. Yang, "Overview on issues and solutions of SLAM for mobile robot," *Comput. Syst. Appl.*, vol. 27, no. 7, pp. 1–10, Jul. 2018.

[2] J. Lambert, A. Carballo, A. M. Cano, P. Narksri, D. Wong, E. Takeuchi, and K. Takeda, "Performance analysis of 10 models of 3D LiDARs for automated driving," *IEEE Access*, vol. 8, pp. 131699–131722, 2020.

[3] T. C. Bybee and S. E. Budge, “Method for 3-D scene reconstruction using fused LiDAR and imagery from a texel camera,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 8879–8889, Nov. 2019.

[4] Z. Jiang, J. Zhu, Z. Lin, Z. Li, and R. Guo, “3D mapping of outdoor environments by scan matching and motion averaging,” *Neurocomputing*, vol. 372, pp. 17–32, Jan. 2020.

[5] H. Kim, S. Song, and H. Myung, “GP-ICP: Ground plane ICP for mobile robots,” *IEEE Access*, vol. 7, pp. 76599–76610, 2019.

[6] J. Zhang and S. Singh, “Low-drift and real-time LiDAR odometry and mapping,” *Auton. Robots*, vol. 41, no. 2, pp. 401–416, Feb. 2017.

[7] F. Aghili and C.-Y. Su, “Robust relative navigation by integration of ICP and adaptive Kalman filter using laser scanner and IMU,” *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 4, pp. 2015–2026, Aug. 2016.

[8] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, “A robust and modular multi-sensor fusion approach applied to MAV navigation,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 3923–3929.

[9] T. Shan and B. Englot, “LeGO-LOAM: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 4758–4765.

[10] H. B. Wu, H. Yue, Z. R. Xu, H. M. Yang, C. Liu, and L. Chen, “Automatic structural mapping and semantic optimization from indoor point clouds,” *Autom. Construct.*, vol. 124, Apr. 2021, Art. no. 103640.

[11] R.-J. Yan, Y.-S. Choi, J. Wu, and C.-S. Han, “Arc/line segments-based SLAM by updating accumulated sensor data,” *J. Inst. Control, Robot. Syst.*, vol. 21, no. 10, pp. 936–943, Oct. 2015.

[12] D. Rodriguez-Losada, F. Matia, and R. Galan, “Building geometric feature based maps for indoor service robots,” *Robot. Auton. Syst.*, vol. 54, no. 7, pp. 546–558, Jul. 2006.

[13] L. H. Wu, S. G. Cui, and L. Zhao, “Map building in indoor environment for service robot based on laser sensor,” *Adv. Building Mater. Struct. Eng.*, vol. 461, pp. 671–676, Feb. 2012.

[14] J. Wang, M. Zhao, and W. Chen, “MIM_SLAM: A multi-level ICP matching method for mobile robot in large-scale and sparse scenes,” *Appl. Sci.*, vol. 8, no. 12, p. 2432, Nov. 2018.

[15] J. Tang, Y. Chen, X. Niu, L. Wang, L. Chen, J. Liu, C. Shi, and J. Hyyppä, “LiDAR scan matching aided inertial navigation system in GNSS-denied environments,” *Sensors*, vol. 15, no. 7, pp. 16710–16728, Jul. 2015.

[16] H. Chen, G. Zhang, and Y. Ye, “Semantic loop closure detection with instance-level inconsistency removal in dynamic industrial scenes,” *IEEE Trans. Ind. Informat.*, vol. 17, no. 3, pp. 2030–2040, Mar. 2021.

[17] M. Li, H. Zhu, S. You, L. Wang, and C. Tang, “Efficient laser-based 3D SLAM for coal mine rescue robots,” *IEEE Access*, vol. 7, pp. 14124–14138, 2019.

[18] K. Konolige and M. Agrawal, “FrameSLAM: From bundle adjustment to real-time visual mapping,” *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1066–1077, Oct. 2008.

[19] H.-C. Lee, S.-H. Lee, T.-S. Lee, D.-J. Kim, and B.-H. Lee, “A survey of map merging techniques for cooperative-SLAM,” in *Proc. 9th Int. Conf. Ubiquitous Robots Ambient Intell. (URAI)*, Nov. 2012, pp. 285–287.

[20] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, “Decentralized active information acquisition: Theory and application to multi-robot SLAM,” in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2015, pp. 4775–4782.

[21] J. W. Fenwick, P. M. Newman, and J. J. Leonard, “Cooperative concurrent mapping and localization,” in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2002, pp. 1810–1817.

[22] H. Do, S. Hong, and J. Kim, “Robust loop closure method for multi-robot map fusion by integration of consistency and data similarity,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5701–5708, Oct. 2020.

[23] R. Dube, A. Gawel, H. Sommer, J. Nieto, R. Siegwart, and C. Cadena, “An online multi-robot SLAM system for 3D LiDARs,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1004–1011.

[24] A. Gil, Ó. Reinoso, M. Ballesta, and M. Juliá, “Multi-robot visual SLAM using a rao-blackwellized particle filter,” *Robot. Auton. Syst.*, vol. 58, no. 1, pp. 68–80, Jan. 2010.

[25] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller, “Multiple relative pose graphs for robust cooperative mapping,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 3185–3192.

[26] Q. Li, L. Chen, M. Li, S.-L. Shaw, and A. Nuchter, “A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios,” *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 540–555, Feb. 2014.

[27] K. Konolige, D. Fox, B. Limketkai, J. Ko, and B. Stewart, “Map merging for distributed robot navigation,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2003, pp. 211–217.

[28] H.-C. Lee, S.-H. Lee, M. H. Choi, and B.-H. Lee, “Probabilistic map merging for multi-robot RBPF-SLAM with unknown initial poses,” *Robotica*, vol. 30, no. 2, pp. 205–220, Mar. 2012.

[29] K. Y. K. Leung, T. D. Barfoot, and H. H. T. Liu, “Decentralized cooperative SLAM for sparsely-communicating robot networks: A centralized-equivalent approach,” *J. Intell. Robotic Syst.*, vol. 66, no. 3, pp. 321–342, May 2012.

[30] H. C. Lee, “Implementation of a network-based robot system for cooperative recognition and localization of multiple objects,” in *Proc. IEEK Summer Conf.*, Jun. 2012, pp. 2008–2010.

[31] S. Hadian Jazi, “Map-merging using maximal empty rectangles in a multi-robot SLAM process,” *J. Mech. Sci. Technol.*, vol. 34, no. 6, pp. 2573–2583, Jun. 2020.

[32] Z. Jiang, J. Zhu, C. Jin, S. Xu, Y. Zhou, and S. Pang, “Simultaneously merging multi-robot grid maps at different resolutions,” *Multimedia Tools Appl.*, vol. 79, nos. 21–22, pp. 14553–14572, Jun. 2020.

[33] Z. XuanYuan, B. Li, X. Zhang, L. Chen, and K. Huang, “Online cooperative 3D mapping for autonomous driving,” in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 256–261.

[34] Y. C. Zhang, L. Chen, Z. XuanYuan, and T. Wei, “3D cooperative mapping for connected and automated vehicles,” *IEEE Trans. Ind. Electron.*, vol. 67, no. 8, pp. 6649–6658, Aug. 2020.

[35] Q. Ye, P. Shi, K. Xu, P. Gui, and S. Zhang, “A novel loop closure detection approach using simplified structure for low-cost LiDAR,” *Sensors*, vol. 20, no. 8, p. 2299, Apr. 2020.

ZHIYUAN ZHU received the B.S. degree in instrumentation engineering from the Guilin University of Electronic and Technology, Guilin, China, in 2019. He is currently pursuing the M.S. degree with the College of Metrology and Measurement Engineering, China Jiliang University. His main research interest includes LIDAR simultaneous localization and mapping (LIDAR SLAM).

WENSONG JIANG received the M.S. degree in instrumentation engineering from China Jiliang University, Hangzhou, China, in 2014, and the Ph.D. degree in instrumentation engineering from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2018. He is currently a Master Tutor with the College of Metrology and Measurement Engineering, China Jiliang University. His main research interests include theory of instrumental accuracy, computer vision, and error analysis.

LI YANG received the Ph.D. degree in electronic engineering from Zhejiang University, Hangzhou, China. He has rich experience in artificial intelligence theory research and industry application research. He is currently an Assistant Professor with the College of Information Engineering, China Jiliang University. His main research interests include the computer vision, LIDAR, and drone detection systems.

ZAI LUO received the B.S., M.S., and Ph.D. degrees in instrumentation engineering from the Hefei University of Technology, Hefei, China. He is currently a Professor with the College of Metrology and Measurement Engineering, China Jiliang University. He is mainly engaged in key technology research and equipment development of mobile measurement robot based on computer vision. He is the Standing Committee Member and the Deputy Secretary-General of the China Instrumentation Society.