IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 10, 2018, accepted December 10, 2018, date of publication December 24, 2018,
date of current version February 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2889304

Efficient Laser-Based 3D SLAM for Coal Mine Rescue Robots

MENGGANG LI, HUA ZHU, SHAOZE YOU, LEI WANG, AND CHAOQUAN TANG
School of Mechanical and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China
Jiangsu Collaborative Innovation Center of Intelligent Mining Equipment, China University of Mining and Technology, Xuzhou 221008, China

Corresponding author: Hua Zhu (zhuhua83591917@163.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC0808000, in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD), China, in part by the National Natural Science Foundation of China under Grant 61603394, and in part by the Natural Science Foundation of Jiangsu Province, China, under Grant BK20150185.

ABSTRACT An accurate description of laneway space with self-localization is a key issue when coal mine rescue robots (CMRRs) perform post-disaster exploration and rescue missions. The 3D simultaneous localization and mapping (SLAM) is an effective but time-critical and highly challenging task in complex laneway scenarios, especially after disasters. In this paper, we propose a novel real-time 3D SLAM based on normally distributed transform (NDT) that employs pose graph optimization and loop closure to further improve mapping consistency. We innovatively extract floors and walls in the laneway as plane nodes to construct landmark constraints, in addition to applying pose nodes from the lidar odometry via NDT. A lightweight and effective loop detection method is conducted using odometry with an appearance-based approach to building a globally consistent map. The proposed method was evaluated on a public dataset, and field tests in an underground coal mine were performed. Results indicate that our algorithm can achieve lower computational complexity and drift, which can provide pose estimation and environment description for CMRRs to realize remote control assistance and automatic navigation in coal mine rescue missions.

INDEX TERMS CMRRs, SLAM, NDT, pose graph optimization.

I. INTRODUCTION

Disasters and accidents in coal mines are highly risky for miners and rescuers. Coal mine rescue robots (CMRRs) equipped with multiple sensors can inspect accident sites and re-establish communication with disaster regions. Sounds, images, temperature, gas concentrations, and other data can be transmitted to the command center for formulating a rescue strategy [1]. However, available robot positioning methods are limited in coal mine laneways, which are complex environments with unpredictable scenarios (Fig. 1). Coal mine disasters lead to priori signposts and maps no longer reliable. The scene is also impossible to reach with GPS, and other wireless positioning methods that rely on base stations are unavailable in the event of post-disaster power interruption. Robotic pose estimation based on dead reckoning cannot provide precise long-term estimation due to the cumulative error on rough terrain, especially in the case of rotation. At present, most CMRRs are based on video for remote operation and positioning, which limits the robot's range of motion and safety [2], [3]. 3D simultaneous localization and mapping (SLAM) is an effective method to provide robot pose estimation and construct a laneway spatial model simultaneously, serving as an important reference for rescue missions. SLAM build maps and provide an intuitive visualization for rescue responders. CMRR path planning and autonomous navigation can also be realized. Yet despite remarkable progress throughout the last 30 years, SLAM algorithms are often unable to cope with overly challenging motions from either the robot or the environment, nor can these algorithms consistently meet strict performance requirements [4]. The application of SLAM technology to CMRRs has faced many obstacles. Coal mines are a hostile environment because the grounds are rugged and laneways are typically degenerated, thus posing obstacles to laser-based SLAM. Due to power interruption, the CMRR must use its own light source, which has low illumination and changes dramatically, leading to difficulty in the application of visual-based SLAM. CMRR sensors must also meet explosion-proof requirements, thereby limiting the available sensor types.

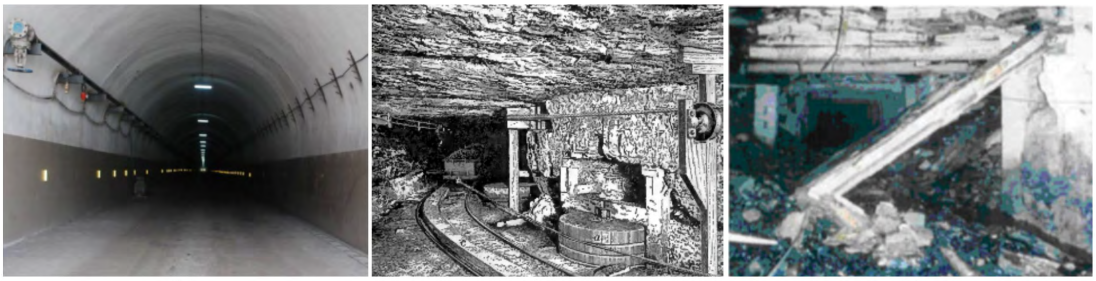

FIGURE 1. Typical environments in underground coal mine. (a) is a tunnel-like main roadway which is a structure environment; (b) is the area near the coal mining face which is a semi-structure environment; (c) is a messy scene after a disaster which is unstructured.

Overall, 2D laser-based SLAM has lower computational requirements and can build a map in the scanning plane in real time. However, it cannot estimate a robot pose that moves in an uneven 3D space with six degrees of freedom (DOF) on uneven ground. The other drawback is that due to too few available features in one scan plane, the long laneway with high similarity is hardly adaptable. Some researchers adopted the method of 2-axis lidar moving in 6-DOF to produce 3D point clouds. Thrun *et al.* [5] used two orthogonal lidars for 3D mapping in underground mine. The horizontal mounted lidar was used to build a 2D map and restore the robot's pose. The upward scan was converted to a point cloud. Zhang and Singh [6] utilized a motor-driven rotating laser scanner and implemented two parallel threads to perform motion estimation and mapping. In these methods, a distance measurement that generated a 3D model based on 2D lidar was received from different times. Hence, the configuration of point cloud assembly was relatively complicated. In addition, inaccurate motion estimation could easily cause distort the point cloud, resulting in mis-registration.

Huber and Vandapel [7] used a cart to carry a 3D laser range scanner to draw a high-precision 3D geological model of the mine laneway. This method used a stop-scan-go method to convert a 3D scan obtained by a 3D laser range scanner into a unified coordinate system. Yet high-precision laser scanners are expensive. Because of the large amount of data in laser point clouds, they need to be stitched offline, which cannot meet real-time mapping and localization requirements for CMRRs. Reference [8] proposed a 3D mapping approach based on a Velodyne HDL-64 laser scanner, focusing on handling measurement noise and building precise model using an off-line map-refinement step.

Many SLAM systems have combined a laser scanner with other sensors to obtain higher accuracy and robustness. López *et al.* [9] integrated a 2D laser, vision, altimeter and Inertial measurement unit (IMU) to improve the 6D pose estimation of an aerial robot in GPS-denied environments. Bosse *et al.* [10] utilized a 2D laser coupled with IMU mounted on a spring, which demonstrated good adaptability to severe motion. Their findings were suitable for mapping but not for robot positioning. Zhang's subsequent work [11] merged data from 3D lidar, camera and IMU to handle highly dynamic motion and in dark, texture-less and structure-less environments, which seemed the most suitable approach for coal mine application.

Current 3D SLAM methods usually require many calculations and cannot realize real-time operation under limited computing resources [7], [8], [12]. Without a method to reduce cumulative errors in real time, most laser-based methods have only focused on the scan registration process [13]; or were used solely in lidar odometry [6], [11], leading to poor adaptability in large scale and rough ground environments. This paper focuses on using 3D lidar to achieve real-time state estimation and mapping for CMRR applications. We believe that a reasonable SLAM algorithm should consider environmental features and be application-oriented. For example, a CMRR may encounter planar surfaces such as walls beside the tunnel and ground plane in the main road. But when it enters a cluttered scene, planar surfaces may not be detected, and other features would need to be used. Cumulative error must also be eliminated because the walking ground is uneven and far away, but we cannot expect a loop to necessarily appear, in which case the complex calculation of loop closure would not be conducive to real-time processing.

The main objective of this work is to provide accurate pose estimation and environmental description for CMRRs, therefore providing a reference for coal mine rescue missions. We propose a real-time and graph-optimized 3D SLAM for mapping and pose estimation of CMRRs in complex environments with variable scenarios. The rest of paper is organized as follows. Section 2 introduces related work. Section 3 describes the proposed approach in detail. Section 4 presents a set of experiments over a public dataset and field tests in an underground coal mine. Finally, we provide a conclusion and suggestions for future work.

II. RELATED WORK

The SLAM method, based on a laser range finder, has been the cornerstone of research on mobile robot mapping and navigation in the past 20 years. 6-DOF localization with 3D mapping remains a focal point of research. Most work on 3D lasers has centered on scan registration, which is used

to estimate the relative transformation between two scans, (i.e., the rigid body transformation matrix representing translation and rotation). Extensive research has focused on the iterative closet point (ICP) method and its variants [14]. The standard ICP method iterates over the nearest point pair and optimizes the solution by minimizing the sum of the point-to-point square Euclidean distance. ICP performance is mainly influenced by factors such as the environment type, motion trajectory, and uncertainty of the initial pose. It is sensitive to the accuracy of the initial value and has poor adaptability to dynamic environments. Armesto *et al.* [15] proposed a metric ICP method to improve robustness to adapt to the uncertainty of mobile robot platform motion. Generalized ICP [16] uses an improved cost function to reduce the effects of incorrectly associated point pairs. In addition to point-to-point, correspondences can be point to line, point to plane, normal vectors, and curvature [17]. References [18] and [19] proposed 6D SLAM, which combines ICP scan registration with a heuristic loop detection and a global relaxation method. This approach results in a highly precise mapping system, but cannot run in real time. LOAM [6] uses edge points and planar points as features, computing point-to-line and point to plane distance. It is considered state-of-the-art laser SLAM, although LOAM does not involve loop closure, and thus cannot correct motion drift in the long-term. Recently, a light-weight and ground-optimized SLAM algorithm, LeGO-LOAM [20], was proposed and achieved similar accuracy with reduced computational costs compared to LOAM. LeGO-LOAM adopted the same features as LOAM but was conducted in a graph optimization framework, performing loop closure simply using ICP. Besides point and line features, planar features [21], [22] can also be employed in SLAM. Weingarten [23] proposed an approach involves using planar features in an EKF-based framework. Trevor *et al.* [24] took the same planar feature extraction method and employed a graph-based approach, which is similar to our work. However, their method relied on a large number of planes and could only build a sparse plane feature map, which is not suitable for complex occasions where there are no planar surfaces.

NDT is a novel scan registration algorithm, which was first applied to 2D laser scan matching [25]. ICP and NDT were compared with regard to registration robustness and speed in [26], revealing NDT to achieve better performance in these aspects than ICP. Reference [27] evaluated two types of scan registration: NDT for local registration and MUMC for global matching on a challenging benchmark, which again performed more robust registration than ICP.

Inevitably, that the cumulative error will occur when pose estimation is only considered in the adjacent time, which makes it impossible to obtain globally consistent trajectories and maps. Scan registration usually applies in the post-processing stage of stitching to eliminate errors, requiring a stop-scan-go strategy and offline processing [7], [18], [19], [28]. Graph optimization is an effective method for reducing cumulative error, and is commonly used in the SLAM back-end [29]. Graph-based SLAM often uses robot poses and landmarks in the environment as state variables. Nodes in the graph are the variables to be optimized, and the edges are the observation constraints between the two interconnected variables. Bundle adjustment (BA) [30] is a classical optimization method for visual-based SLAM. It takes the sensor poses and positions of features in the environment as optimization variables and decreases dimensions based on the sparse characteristic. Pose graph optimization [31] only considers keyframe nodes formed by sensor poses, thus saving optimization time of for many features and resulting in more efficient calculation.

Loop closure produces a correlation between current data and all historical data, providing more constraints for graph optimization to build globally consistent maps, and relocating after scan matching fails due to intense motion. Because of minor differences in textures and features, loop detection under degenerate scenes such as tunnels and laneways in underground coal mines remains difficult and time consuming. Much work on loop detection has been based on visual and 2D scan data. Bag-of-words [32] is a common method in visual SLAM, which uses feature clustering to construct a dictionary and detect similarities between two image frames. Google's Cartographer [33] applies scan-to-submap matching via the brance-and-bound approach to create loop closure constraints. Magnusson [34] proposed an appearance-based loop closure method, created a feature histogram of surface topography to effectively match a two-frame scan, and achieved good detection results.

The approach most similar to ours is that of Mendes *et al.* [35] and Einhorn and Gross [36]. In [35], an ICP layer was used as the front-end to construct and concatenate local maps. Further, a graph layer with loop closure was constructed as a back -end. In essence, this work applied ICP registration which is time consuming in a nearest neighborhood search. A constraint in the pose graph was only added by the transformation computed by ICP along with an estimation of its covariance or by loop closing. An error function was taken to judge a loop closing when it lay below a threshold, which was too tolerant and susceptible to failure when encountering similar scenes. In [36], NDT mapping was combined with occupancy mapping to handle dynamic objects, which was also based on a graph approach for lifelong SLAM usage.

This paper proposes an efficient and robust 3D-SLAM method that allows CMRRs to solve state estimation and mapping problems in complex various environments. This paper's contribution is threefold: First, point cloud segmentation is performed to extract planar surfaces as landmarks to construct measurement constraints, which is suitable for graph-SLAM for CMRRs in unbroken structured roadways. Second, NDT is used to construct lidar odometry constraints, which plays a major role in unstructured environments. Third, a lightweight but effective loop closure is designed using an odometry and appearance-based approach and applied to correct motion drift once the loop appears. In addition, our

results indicate a complete graph-based framework for 3D lidar SLAM, which provides a demonstration for practical applications in the industrial and rescue fields.

III. APPROACH

Scan registration is the core of laser-based SLAM approaches, solving the problems of feature extraction, data correlation and motion estimation. Our work is based on the NDT method to construct lidar odometry. Unlike ICP and its variants which look for associated features such as points, lines, and planes, NDT uses range scans to describe the environment model. The space is partitioned into cells and normal distributions are assigned to represent the probability model of the measured space. Points in the data scan are registered with a series of distributions in the model scan. The advantage of NDT scan registration is that it does not need to compute the complex search of the nearest neighbor. The point cluster represented by a normal distribution (ND) does not need to store a large amount of point cloud data in the original scan frame, which takes up low storage. In addition, the gradient vector and Hessian matrix used to compute the score function each has analytical solutions, allowing the standardized nonlinear optimization method to be used to determine optimal motion estimation. These factors improve the computation speed and robustness of the algorithm when facing complex environments.

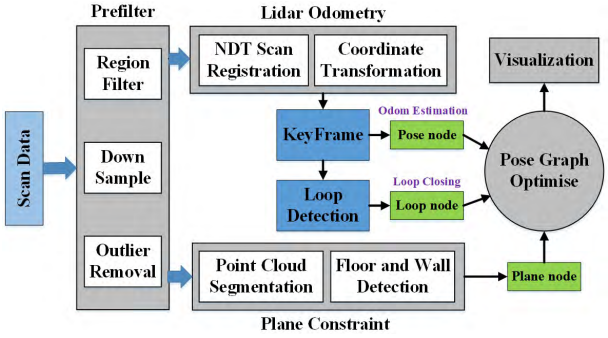

FIGURE 2. System overview, showing all the steps performed by the prefilter, lidar odometry, floor constraint, and loop closure with pose graph optimization threads.

An overview of the proposed algorithm framework is presented in Figure 2. The approach is based on four main parallel threads: prefiltering, lidar odometry, plane constraint, and pose graph optimization. The original point cloud contains many redundant points, whereas the data acquisition process has many noisy points due to unsteady robot movement. To improve calculation speed and registration precision, filtering is used to compress point clouds and remove outliers. NDT scan registration is used in the process of lidar odometry for pose estimation, as well as in loop detection for environmental similarity detection. Lidar odometry is in charge of creating keyframes by pose changes. The keyframes are used to optimize the graph and detect loops. These steps save computational time compared to optimizing all scans. The point cloud segment approach of RANdom Sample Consensus (RANSAC) is taken to extract planar surfaces of the laneway floor and side wall to construct the plane node. Edges are constructed by using the constraint relationship between the pose nodes of the keyframe and corresponding plane nodes. These steps are explained in detail in the following sections.

A. 3D-NDT

An important innovation of this paper is applying NDT to pose estimation, loop detection, graph optimization and mapping respectively. Therefore, the NDT algorithm and the factors affecting performance are analyzed first.

1) ND DESCRIPTION

The nearest neighbor search is time-consuming and requires a lot of storage space to compute the point-to-point registration method, which has low registration accuracy in complex and dynamic environments. ND describes the model scanning without using the point's position; instead, it is divided into voxels (squares for 2D, cubes for 3D), using the probability density function (PDF) constructed by all points within the voxel to represent the distribution characteristics, (e.g., the location, direction and smoothness of the point cloud surface). We assume that the reference frame surface points are generated by a 3D random process. The point $\boldsymbol{x}_i$ of the scanned frame is filtered from these distributions. Model scan Y is divided into K individual cells and k is the nearest cell to $\mathbf{x}_i$. The inner point set is expressed as $Y_k = \{\mathbf{y}_1^{(k)} \dots \mathbf{y}_M^{(k)}\}$, where element $\mathbf{y}_i^{(k)}$ represents the position of point i within cell k. The likelihood function where there is a point at position $\mathbf{x}_i$ in cell Y_k can be formulated as

$$p(\mathbf{x}_i) = \frac{1}{(2\pi)^{3/2}\sqrt{\det \boldsymbol{\Sigma}_k}} \exp\left(-\frac{(\mathbf{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_k)}{2}\right) \tag{1}$$

The mean and covariance of all points' positions within cell k can be expressed as

$$\boldsymbol{\mu}_k = \frac{1}{M_k}\sum_{i=1}^{M_k} \mathbf{y}_i^{(k)} \tag{2}$$

$$\sum\nolimits_k = \frac{1}{M_k - 1}\sum_{i=1}^{M_k} (\mathbf{y}_i^{(k)} - \boldsymbol{\mu}_k)(\mathbf{y}_i^{(k)} - \boldsymbol{\mu}_k)^T \tag{3}$$

The eigenvalues and eigenvectors of the covariance matrix reflect the surface orientation and smoothness. Three orthogonal eigenvectors reflect the shape of an ellipsoid, which can be linear (one eigenvalue is much larger than the other two), planar (one eigenvalue is significantly smaller than the other two), and point or spherical (three eigenvalues are similar). The covariance matrix is singular when there are fewer than 4 points in a cell. Cells with fewer than 4 points will be considered unoccupied.

2) NDT REGISTRATION

The objective of scan registration is to find the optimal pose transformation, which can maximize the likelihood of the current data scan falling on the surface of the reference scan. The current data scan is represented as $X = \{\mathbf{x}_1 \dots \mathbf{x}_N\}$, where element $\mathbf{x}_i$ is the location of each scan point. Rotation and translation parameters for pose estimation can be represented by $\mathbf{p} = [t_x \ t_y \ t_z \ \theta_x \ \theta_y \ \theta_z]^T$. Using 3D rigid body space transformation, the transformed poses of scan points can be represented as

$$\mathbf{x}_i' = \mathbf{T}(\mathbf{p}) \cdot \mathbf{x}_i = \mathbf{R}_z(\theta_z)\mathbf{R}_y(\theta_y)\mathbf{R}_x(\theta_x)\mathbf{x}_i + \mathbf{t} \tag{4}$$

where $\mathbf{R}(\theta)$ is the rotation matrix of each axis, and $\mathbf{t}$ is the translation matrix. ICP directly maps the transformed point to the closest point in the reference scan by using brute force search iteratively. NDT assumes another strategy, which uses point-to-distribution correspondence to calculate the optimal transformation relationship when all points fall into these distributions to obtain the maximum likelihood

$$\bar{\psi}(\mathbf{p}) = \prod_{i=1}^{N} p(T(\mathbf{p}, \mathbf{x}_i)) = \prod_{i=1}^{N} p(\mathbf{x}_i') \tag{5}$$

Then the following score function is used to establish an optimization problem

$$\begin{aligned} score(\mathbf{p}, \mathbf{x}_i, k) &= -\log \bar{\psi}(\mathbf{p}) = -\sum_{i=1}^{N} \log(p(\mathbf{x}_i')) \\ &= \sum_{i=1}^{N} \frac{(\mathbf{x}_n' - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\mathbf{x}_n' - \boldsymbol{\mu}_k)}{2} + const \end{aligned} \tag{6}$$

We can use an arbitrary probability density function (PDF) that describes the local surface structure and is robust to outliners. To reduce the influences of outliers on the results, a mixed Gaussian distribution is employed instead of the standard ND function (1), which is expressed as

$$p(\mathbf{x}_i') = c_1 \exp\left(-\frac{(\mathbf{x}_i' - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_i' - \boldsymbol{\mu}_k)}{2}\right) + c_2 p_0 \tag{7}$$

This function limits the influences of outliers in the bounded scope. c_1 and c_2 are constants that integrate the kth cell to one, and p_0 is the ratio of outliers. The negative log-likelihood can be approximated by

$$\begin{aligned} \bar{p}(\mathbf{x}_i') &= -\log(c_1 \exp\left(-\frac{(\mathbf{x}_i' - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_i' - \boldsymbol{\mu}_k)}{2}\right) + c_2 p_0) \\ &\approx -d_1 \exp(-d_2 \frac{(\mathbf{x}_i' - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_i' - \boldsymbol{\mu}_k)}{2}) \end{aligned} \tag{8}$$

Thus the score function can be represented as

$$score = -\sum_{i=1}^{N} \bar{p}(\mathbf{x}_i') = -\sum_{i=1}^{N} \bar{p}(T(\mathbf{p}, \mathbf{x}_i)) \tag{9}$$

The optimal parameter $\mathbf{p}$ can be calculated by minimizing Eq.(9). Because the first and second derivatives of this score function have analytic forms, Newton's method can be directly used to search the optimal $\mathbf{p}$ iteratively:

$$\begin{aligned} \mathbf{H}\Delta\mathbf{p} &= -\mathbf{g} \\ \mathbf{p} &\leftarrow \mathbf{p} + \Delta\mathbf{p} \end{aligned} \tag{10}$$

The distance from the converted position of a point in the data scan to the center of its nearest cell is defined as $\mathbf{x}_i' \equiv \mathbf{T}(\mathbf{p}, \mathbf{x}_i)$-$\boldsymbol{\mu}_k$. Thus the gradient can be written as

$$g_j = \frac{\partial s}{\partial \mathbf{p}_j} = \sum_{k=1}^{n} d_1 d_2 \mathbf{x}_i'^T \boldsymbol{\Sigma}_i^{-1} \frac{\delta \mathbf{x}_i'}{\delta \mathbf{p}_j} \exp(-\frac{d_2}{2} \mathbf{x}_i'^T \boldsymbol{\Sigma}_i^{-1} \mathbf{x}_i') \tag{11}$$

H_{jl} corresponding to Hessian matrix $\mathbf{H}$, can be calculated by

$$\begin{aligned} H_{jl} &= \frac{\partial^2 s}{\partial \mathbf{p}_j \partial \mathbf{p}_l} \\ &= \sum_{i=1}^{n} d_1 d_2 \exp(\frac{d_2}{2} \mathbf{x}_i'^T \boldsymbol{\Sigma}_i^{-1} \mathbf{x}_i') \\ &\quad \times \left(-d_2 (\mathbf{x}_i'^T \boldsymbol{\Sigma}_i^{-1} \frac{\delta \mathbf{x}_i'}{\delta \mathbf{p}_j})(\mathbf{x}_i'^T \boldsymbol{\Sigma}_i^{-1} \frac{\delta \mathbf{x}_i'}{\delta \mathbf{p}_l})\right) \\ &\quad + \mathbf{x}_i'^T \boldsymbol{\Sigma}_i^{-1} \frac{\partial^2 \mathbf{x}'}{\partial \mathbf{p}_j \partial \mathbf{p}_l} + \frac{\delta \mathbf{x}_i'^T}{\delta \mathbf{p}_l} \boldsymbol{\Sigma}_i^{-1} \frac{\delta \mathbf{x}_i'}{\delta \mathbf{p}_l} \end{aligned} \tag{12}$$

If convergence is found or the limited iteration number is met, the nonlinear optimization will terminate to obtain the optimal relative transform $\mathbf{T}$. NDT registration can be accelerated with a given initial estimate, which can be derived from estimation of other sensors such as an odometer. In this work, the last keyframe's pose will be set as the initial estimation of the new pair-wise registration to accelerate the convergence process.

Many factors affect the performance of the NDT algorithm in actual environments. A discussion about the influence of the sample method, cell partitioning and valley of convergence for NDT in different scenes can be found in [13]. Due to the inefficiency of the NDT algorithm in the PCL library [37], we use multiple threads to accelerate the algorithm. The system relies on several heuristics that require fine tuning to obtain good results in the selected environment; therefore, we provide adjustable interfaces for some parameters to deal with different environmental characteristics.

B. PREFILTER

The point cloud feature of laser scanning indicates that a farther distance leads to sparser points and more outliers in the point cloud; hence, a region filter approach is used to extract an effective point cloud from the distance between 20cm to 100m. We only keep the point cloud within the specified distance for subsequent processing. The filter is achieved by constructing a voxel grid and down-sampling. The number of point clouds can be reduced substantially by

approximating all points in voxel with the center of their gravity to achieve point cloud compression. We use the RADIUS filter to eliminate outliers by giving the search radius and the number threshold of the points in the circle, which is faster than the statistical-based filter. The above three filtering processes are achieved based on the PCL library.

C. LIDAR ODOMETRY

There is little new information and few accumulated linear errors when processing continuous scans. The keyframe can be utilized in odometry estimation, pose graph optimization, loop detection, and mapping. It can improve computational efficiency substantially, particularly for a large map size, and can make the algorithm run in real-time. The concept of a keyframe was first used in visual SLAM [38]. When establishing keyframes, steps should be taken to decrease redundant keyframes and save computation while reducing registration error; however, overly sparse keyframes lead to increased uncertainty in the observation between the frames and are highly unfavorable in optimization, thus compromising mapping quality. Referring to the visual SLAM method, we use a simple approach to establish keyframes and eliminate redundant selections. Poses lie in the special Euclidean group of $SE(3)$, and we are interested in translation and rotation. The first filtered point cloud is always treated as a keyframe. The input point cloud will be registered in the nearest keyframe to calculate the lidar odometry. The relative pose $\Delta\mathbf{T}_{k-1,i}$ from a new scan frame i to the last keyframe $k-1$ can be obtained by matching the input point cloud to the previous keyframe through NDT registration. Accordingly, the pose of the current input point cloud i and the lidar odometry can be calculated by:

$$\mathbf{T}_i = \mathbf{T}_{k-1}\Delta\mathbf{T}_{k-1,i} \tag{13}$$

The keyframe poses can be expressed as 4×4 homogeneous matrices. The NDT registration can run faster when using the previous relative transformation $\Delta\mathbf{T}_{k-1,i-1}$ as an initial guess, assuming the robot does not change the motion state drastically. Once the change is larger than a threshold, which causes the NDT to no longer converg, the previous keyframe $\mathbf{T}_{k-1}$ and relative transformation $\Delta\mathbf{T}_{k-1,i-1}$ will be used to calculate the odometry of the current input point cloud. According to Formula (13), the transformation between adjacent frames can be expressed as

$$\Delta\mathbf{T}_{k-1,i} = \mathbf{T}_{k-1}^{-1}\mathbf{T}_i = \begin{bmatrix} \Delta\mathbf{R} & \Delta\mathbf{t} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{14}$$

The Euclidean norm of translation vector $\Delta\mathbf{t}$, which represents the translation of current frame and the last keyframe, is

$$\|\Delta\mathbf{t}\| = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \tag{15}$$

The rotation of these adjacent frames Δr can be defined as the Geodesic distance directly from the rotation matrix $\Delta\mathbf{R}$.

$$\Delta r = \arccos\left(\frac{trace(\Delta\mathbf{R}) - 1}{2}\right) \tag{16}$$

The keyframe pose will be updated when the following condition applies. Set the threshold of minimum translation distance threshold $t_{\min}$, the rotation angle threshold $r_{\min}$ and the running time threshold $h_{\min}$. Once one of the three threshold is exceeded, a keyframe pose will be updated by the pose of current lidar odometry. The keyframe will be registered in the graph when one of the former two conditions is met. Thus, every input point cloud will be used in odometry estimation, but only those satisfying the relative transformation condition can be considered keyframes and participate in graph optimization. This method eliminates redundant information, especially when the robot is not moving for a long time.

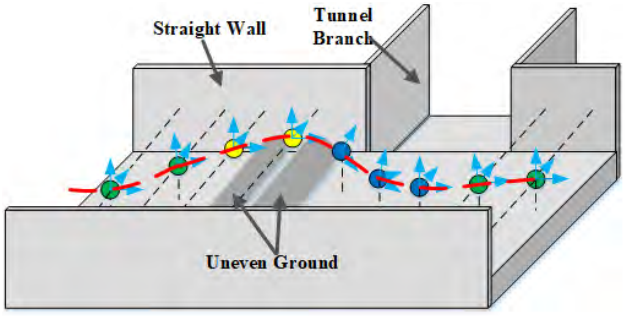

FIGURE 3. The plane nodes, the keyframe pose nodes, and their relationship of geometric constraints. The red curve represents the trajectory of the robot, and the black dotted line is the constraint provided by wall or ground. The spheres and the corresponding coordinate systems represent the poses of different keyframes. The green sphere indicates that the pose is subjected to both wall and ground constraints in most cases. The yellow sphere indicates a condition in which only the walls are constrained while the robot moves on the uneven road. The blue sphere indicates that the movement is only constrained by the ground when the robot is at the junction of the laneways.

D. LOOP CLOSURE AND POSE GRAPH OPTIMIZATION

1) PLANE CONSTRAINT

The coal mine environment is usually dominated by laneways of equal width over a long distance, as shown in Fig. 1-a. The large area of ground surface and side wall plane can provide additional constraints for motion estimation of the keyframes. The purpose of establishing plane constraints is to provide additional vertices for optimization of the pose graph while decreasing accumulated errors by using more constraint information. Although additional constraints increase computational complexity, the 'stiffness' of the pose graph can increase, thus improving the accuracy of pose estimation. Fig. 3 presents the plane nodes, keyframe pose nodes, and their relationship in geometric constraints. When the robot is running in different of space regions, the plane of the ground or wall can provide additional constraints on the pose of the current keyframe. The same plane node may be observed under different pose nodes. In mutative conditions with uneven or forked laneways, different plane nodes can be constructed to provide keyframes constraints.

The key to establishing ground and wall constraints is the extraction of horizontal and vertical planes in space, which can be regarded as a problem of point cloud segmentation.

An iterative RANSAC is used for plane detection. The plane with the most and sufficient inlier points will be extracted and saved. After removing all inliers from the last detection, the next largest plane will be searched as long as a sufficient number of points can be found. After each plane extraction, inliers will be clustered to remove small clusters of points and separate multiple surfaces that are coplanar but correspond to different locations. Clusters with a sufficient number of points will be used as plane landmarks for graph optimization. Detected planes must track continually to strengthen the graph constraints as the same surfaces can be seen in different poses.

To deal with multiple models in one dataset, feature filtering will be performed before iterative RANSAC by using the normal direction of the point cloud surface. The target point clouds used to detect the plane by RANSAC are limited to those whose normal directions are within a certain threshold range from the vertical or horizontal normal. If there are sufficient points after filtering, the iterative RANSAC will be further used to detect the plane. In real environments, especially structured mines, the planar landmarks we want to detect are always either side walls or ground, which are often perpendicular to each other. Feature filter will decrease the points processed when performing RANSAC and accelerate extraction of planar landmarks. Fig. 4 presents an intuitive explanation for segmented planar points using the above approach.

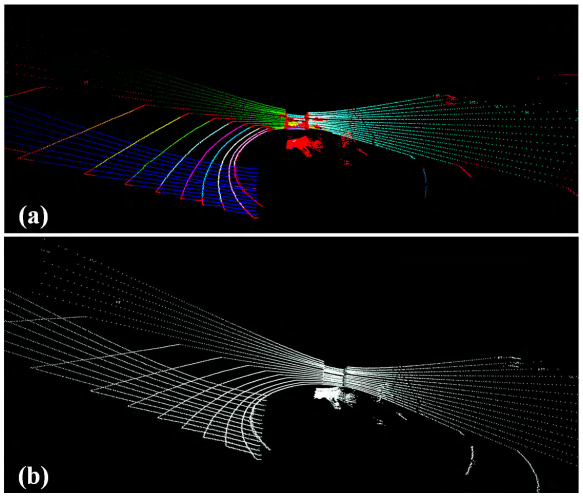

FIGURE 4. The segmented planar points in one scan by the iterative RANSAC. (a) The blue, green and cyan points on three planes will be modeled as planar features, as well as the ground points in different colors. Points in red are filtered by the feature filter. (b) The original point cloud in one scan.

After planar extraction, the model of a plane can be represented as: $ax + by + cz + d = 0$. The segmented point cloud of wall and ground, as well as the model coefficient, will be released separately and registered in the pose graph. Constraints from planar features reduce accumulated error, particularly when orientated in environments with large flat planes.

2) LOOP CLOSURE

Because the laser odometry in the front end only obtains the local pose constraint, errors generated before will inevitably accumulate to the next moment, making it difficult to construct a globally consistent trajectory and map. Loop detection can identify the revisit area, after which accumulated errors can be greatly decreased by graph optimization. We propose a simple method that combines an odometry-based with an appearance-based approach to detect loop candidates.

First, it is impossible for the robot to reach a loop in a few scene sequences in a real scenario. The transformation between any two keyframes is estimated in the real world by lidar odometry. If the distance between two non-contiguous keyframes is less than a threshold, then a loop is highly likely to emerge. To avoid redundant loops caused by contiguous keyframes, the minimum accumulated distance threshold between any two keyframes is set. Using the odometry-based approach, loop candidates are spatially constrained within a set range.

Second, during the appearance-based stage, the same method to describe scene similarity by histogram is taken as in [34]. As mentioned earlier, three orthogonal eigenvectors ($\lambda_1 < \lambda_2 < \lambda_3$) can reflect the distribution shapes of a fixed voxel:

- If $\lambda_2/\lambda_3 \leq 0.1$, then the distribution tends to be linear.
- If it is non-linear and $\lambda_1/\lambda_2 \leq 0.1$, then the distribution is deemed planar.
- If neither of the above applies, then the distribution is considered spherical.

The histogram based on the above three types of shapes will be counted within a certain range of the robot. The higher the similarity of the histogram, the greater the likelihood of the loop appearing. Loop candidates are further decreased when using an appearance-based approach.

Finally, NDT is applied to confirm the loop following the above two stages through the fitness score. The fitness score, which reflects appearance similarity, can be calculated by the root mean squared error (RMSE) of the Euclidean distance of the nearest point pairs:

$$fitness\_score = \frac{1}{N}\sum_{k=1}^{N}\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \qquad (17)$$

When the fitness score is less than the threshold, it can be considered the same scene. The smaller the threshold is set, the more severely the detected similarity. At the same time, the loop edges that are too close to each other will be eliminated to save the calculation.

A false positive in loop detection has an adverse effect because it adds incorrect edges to the graph and results in poor optimization. A good loop detection algorithm must maximize the true positive and minimize the false positive. Features in the coal mine laneway are few and similar, which can easily lead to inaccurate loop detection. In a real application, we can adjust the appearance similarity and

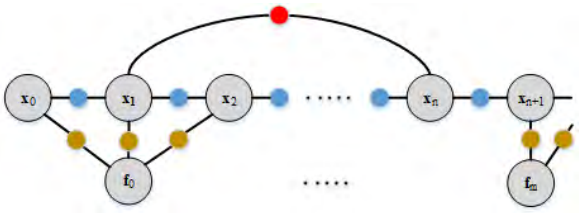

FIGURE 5. The factor graph of the pose graph SLAM. The pose nodes $\{\mathbf{x}_0, \mathbf{x}_1 \ldots \mathbf{x}_n, \mathbf{x}_{n+1} \ldots\}$ of the keyframes and plane nodes $\{\mathbf{f}_1 \ldots \mathbf{f}_m \ldots\}$ of the detected wall or ground plane are represented by gray circles. The blue circles are the constraints (U) constructed by the motion model of lidar odometry. The red circles are the loop closure constraints (Z) constructed by scan matching measurement model. The plane constraints (G) are the yellow circles which are constructed by the measurement model from common detected planes.

distance threshold to increase precision. Although it may reduce recall, a slight overlap is better than incorrect optimization, which may render the map invalid.

3) POSE GRAPH OPTIMIZATION

The cumulative error of laser odometry will continue to increase as the trajectory and map size increase in pairwise registration. Pose graph optimization is a cost-effective approach as the back-end, which can improve the estimation accuracy and obtain a globally consistent trajectory and map. Fig. 5 shows the factor graph of the pose graph SLAM. Nodes in the graph are keyframe poses and common planes detected by different keyframes. The edges are either lidar odometry constraints (U), loop closure constraints (Z), or plane constraints (G) constructed by the ground or wall. The poses and constraints can be modeled as the joint distribution:

$$P(X, U, Z, G) = \prod_{i=1}^{M} P(\mathbf{x}_i|\mathbf{x}_{i-1}, \mathbf{u}_i) \cdot \prod_{j=1}^{N} P(\mathbf{z}_j|\mathbf{x}_{ij}, \mathbf{x}_{lj}) \cdot \prod_{k=1}^{K} P(\mathbf{g}_k|\mathbf{x}_k) \quad (18)$$

Here, assuming there are M poses, N loop closures and, K plane constraints, we can minimize the negative log of the joint probability to find the maximum a posteriori (MAP) estimate over the robot poses:

$$\begin{aligned} X^* &= \arg\min_X P(X, U, Z, G) \\ &= \arg_X \min(-\log P(X, U, Z, G)) \\ &= \arg\min(\sum_{i=1}^{M} \|f_i(\mathbf{x}_{i-1}, \mathbf{u}_i) - \mathbf{x}_i\|^2_{\Sigma_{i,i-1}} \\ &\quad + \sum_{j=1}^{M} \|h_j(\mathbf{x}_{ja}, \mathbf{x}_{jb}) - \mathbf{z}_j\|^2_{\Sigma_j} \\ &\quad + \sum_{k=1}^{M} \|h_k(\mathbf{x}_k) - \mathbf{g}_k\|^2_{\Sigma_k}) \end{aligned} \quad (19)$$

The objective function of the whole problem consists of the sum of squares of multiple errors from different sources. $f_i(\cdot)$ is the lidar odometry motion model, and $h_j(\cdot)$ is the loop closure measurement model. These two models can be represented by the transformation from NDT scan matching. $h_k(\cdot)$ is the plane constraint measurement model. $\mathbf{x}_i$, $\mathbf{z}_j$, $\mathbf{g}_k$ and Σ_i, Σ_j and Σ_k respectively represent the mean and the information matrix. Here, information matrices Σ_i and Σ_j, which reflect the certainty of relative pose estimation can be calculated by:

$$\mathbf{\Sigma} = \begin{pmatrix} \mathbf{\Sigma}^{tra} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}^{rot} \end{pmatrix}, \quad \mathbf{\Sigma}^{tra} = \frac{\mathbf{I}}{\sigma^2_{tra} \cdot e}, \quad \mathbf{\Sigma}^{rot} = \frac{\mathbf{I}}{\sigma^2_{rot} \cdot e} \quad (20)$$

where $\mathbf{I}$ is an identify matrix; σ^2_{tra} and σ^2_{rot} are the variance of translation and rotation errors, respectively; e is the root mean squared error (RMSE) in Eq. (17) of the matching between two poses from keyframe nodes or loop candidates.

Information matrix Σ_k reflects the certainty of registered edges between plane nodes and keyframe nodes, representing the plane normal and perpendicular distance constraints. As planar features are re-observed in adjacent poses, the landmark's convex hull will grow as additional portions are detected. We use the previously observed planar features $(\vec{n}, d)$ in the map frame as the predicted term. Combined with a measured plane $(\vec{n}_m, d_m)$, the error term in (19) can be written as

$$e_{h_j} = \begin{pmatrix} \mathbf{R}^T \cdot \vec{n} - \vec{n}_m \\ \langle \vec{n}, \mathbf{t} \rangle + d - d_m \end{pmatrix} \quad (21)$$

where $(\mathbf{R}, \mathbf{t})$ is the transform between the map frame and the robot frame. Using Formula (21) and the corresponding Jacobian w.r.t the pose and landmark, planar normal and perpendicular distances can be utilized as landmarks in the graph-optimization problem. Lie algebras $se(3)$ is used to represent state variables and solve an unconstrained least-squares optimization problem on manifold [39]. A problem in this form can be effectively solved by the standard optimization methods such as Gauss-Newton or the Levenberg-Marquardt (LM) [40] algorithm, which have been integrated in the optimization library G2O [41].

E. MAPPING

The mapping process runs at a lower frequency than lidar odometry; and is activated only when a new keyframe or loop is detected. The main idea is to convert the point cloud of adjacent keyframes to the world coordinate system for stitching, which is also done by NDT registration. Fig. 6 illustrates the mapping process. When the $k+1$ key frame is built, lidar odometry generates the local point cloud $\mathbf{P}_{k+1}$ in t_{k+1}, as well as the relative pose transformation, $\mathbf{T}^{\mathrm{T}}_{k+1}$. The mapping algorithm utilizes pose transformation $\mathbf{T}^{\mathrm{L}}_{k+1}$ and the last keyframe $\mathbf{T}^{W}_{k}$ to match and register $\boldsymbol{P}_{k+1}$ to the world coordinate system, $\{\mathbf{W}\}$ and constructs the new map of Q_{k+1} based on the existing map of Q_k. When a loop is detected, the map is optimized along with the keyframe pose to obtain a globally consistent map. To save storage space and maintain enough effective point cloud information for loop closure

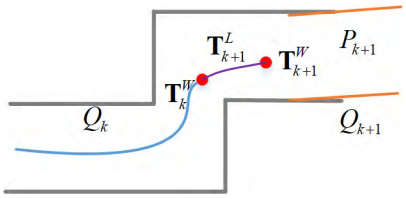

FIGURE 6. Illustration of the mapping process. The blue curve indicates the trajectory of the lidar. The red dots are the poses of the adjacent keyframes, T^W_k and T^W_{k+1}, which are obtained by lidar odometry. The purple curve is the relative pose transformation between adjacent keyframes, T^L_{k+1}. With T^W_k and T^L_{k+1}, the new point cloud P_{k+1} which is published from the new pose T^W_{k+1}, can be projected into the existing cloud on the map Q_k, and construct the new map of Q_{k+1}.

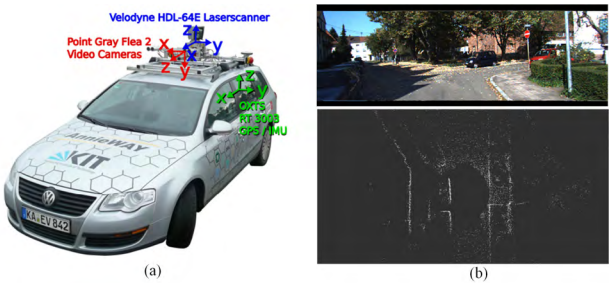

FIGURE 7. (a) Platform used in KITTI benchmark. Our method uses data from the Velodyne HDL-64E only. The ground truth data is provided by the OXTS RT3003 and GPS navigation system. (b) A real scene (upper figure) and its corresponding point cloud.

detection, the octree structure [42] is used to compress point clouds at the same time for rapid searching.

IV. EXPERIMENT

A. TESTS WITH KITTI DATASETS

Initial tests were conducted with the KITTI odometry benchmark [43] which is widely used in the SLAM community for algorithm evaluation. Fig.7 dipicts the platform used in the benchmark and a typical scene in the datasets. The platform contains a Velodyne HDL-64E 3D laser scanner and an OXTS RT3003 inertial and GPS navigation system, etc. Real-world situations include an urban setting with buildings, country setting with surrounding vegetations, and highway with open scene. Because KITTI odometry benchmarks have corrected the point distortion in the data collection from the lidar, the raw dataset was used in our experiment. The algorithm was run on an industrial personal computer (IPC) with CPU: i7-3555LE (2.5 GHz, 4 core) and 8G memory.

Fig. 8 shows the mapping result of our NDT based graph SLAM. We can see that the point cloud is consistency with high mapping quality. Benefit from the loop detection and graph optimization, the map is extremely coincidence with the real scene. The octree map saves storage space of point cloud in the meantime ensuring the map precision and details.

To determine the accuracy of pose estimation, the proposed method was compared with NDT only (NDT-Odometry), ICP-based graph SLAM, state-of-the-art LeGO-LOAM, and the ground truth on four sequences 00, 01, 05, 07. A typically

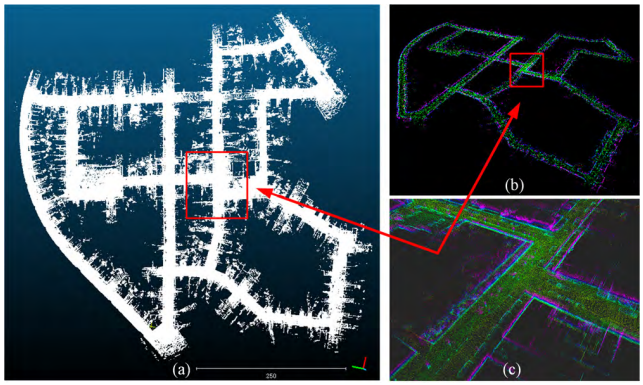

FIGURE 8. (a) and (b) represents the point cloud map built by our method on sequence 00. The color in (b) and (c) reflects the elevation of different points. (c) is a close shot of an intersection in the urban scene. It shows that the mapping quality is pretty good by the proposed approach.

large loop exists in sequences 00, 05, and 07, which benefits global optimization. Compared with ICP-based graph SLAM proposed in [35], we reproduced the algorithm because it is not open source. In the same way, we handled motion estimation and mapping processes based on the keyframe, and constructed the optimization problem. The difference is that we use the keyframe extraction and mapping strategy in this paper without using local maps, and G2O was used for optimization. The trajectory results are presented in Fig.8.

The trajectories in Fig.9 indicate that, LeGO-LOAM performed the best in terms of accuracy and consistency in sequences 00, 05, and 07. These sequences are all in residential areas and city roads, containing many line and plane features in the environment. On all sequences, our algorithm performed better than the ICP-graph-SLAM and NDT-Odometry approaches in consistency and coincidence with ground truth. Compared with NDT-Odometry, benefiting from the loop detection and optimization process, the pose estimation of the proposed approach did not diverge along the way. This is particularly obvious in sequence 00 (Fig. 9-a) in which our algorithm returned to the origin but NDT-Odometry did not. ICP-graph-SLAM also performed well on consistency, but the generated trajectory did not coincide well with the ground truth on each sequence. All trajectories exhibited large shifts on sequence 01 (Fig. 9-b). The real path length was roughly 2450 m. The length of our proposed approach was 2300.6 m, whereas that of NDT-Odometry was only 1297.3 m and that of ICP-graph-SLAM was 1090.1 m. The LeGO-LOAM also demonstrated poor performance, especially in the orientation direction. The large shift was due to the scene is being on a highway where roads were wide and the surrounding environment was structureless, representing a typically degenerate scene that negatively affects scan registration. Another important influencing factor is the point distortion became more obviously when the move speed is faster. Despite this, our algorithm performed well. We suspect the proposed algorithm is superior to the other three algorithms due to extracting the surface of the

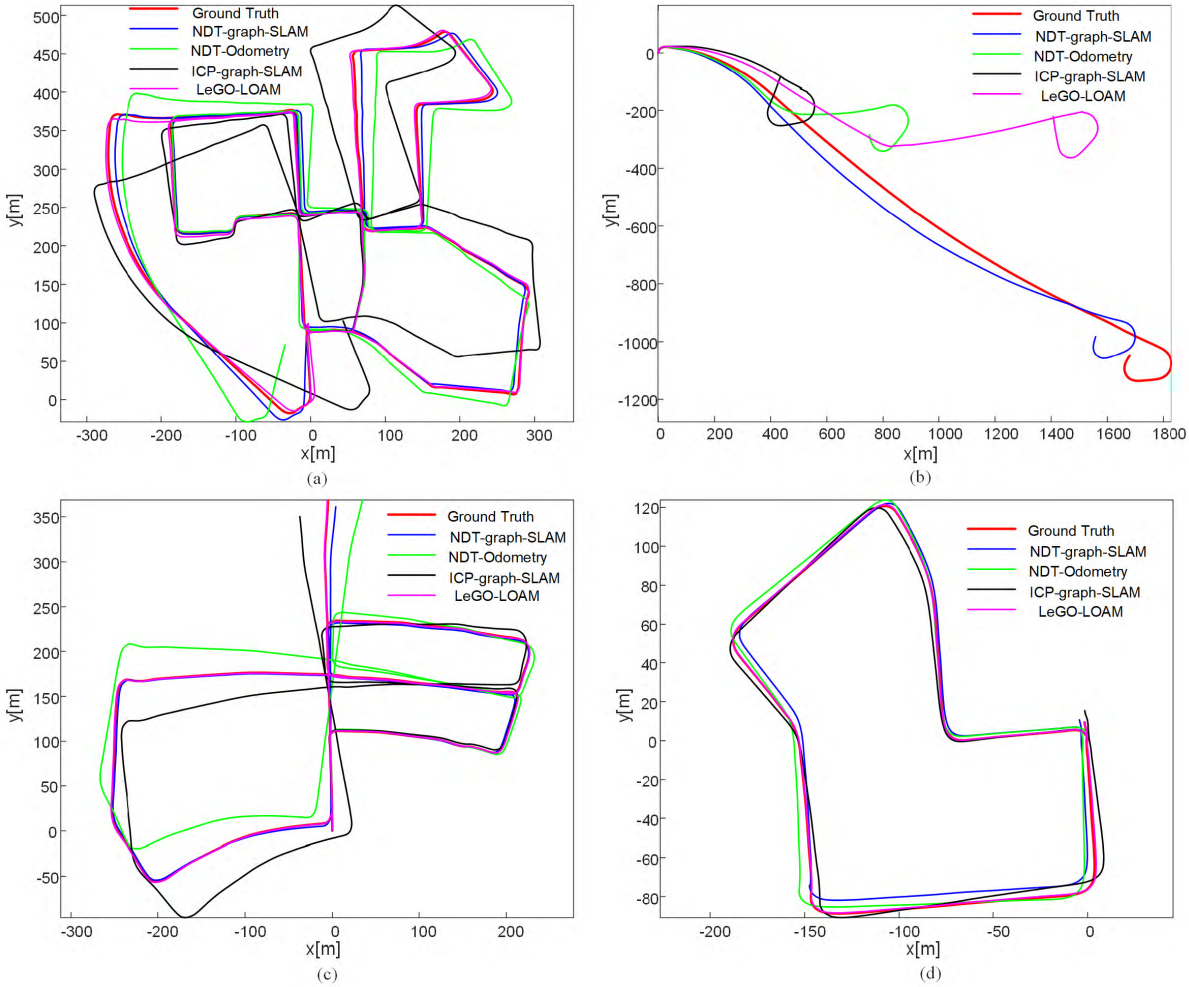

FIGURE 9. The trajectories estimated by NDT-graph-SLAM (blue), NDT-Odometry (green), ICP-graph-SLAM (black), LeGO-LOAM (magenta) and the corresponding ground truth (red) on (a) sequence with loops 00, (b) sequence without loop 01, (c) sequence with loops 05, (d) sequence with loops 07.

coplanar ground as landmarks, which provided additional constraints and achieved better consistenc.

Quantitatively, the relative pose error (RPE) and absolute trajectory error (ATE) are always used for evaluation metrics for in SLAM [44]. The former reflects the local accuracy, and the latter reflects the global consistency. We only focused on the ATE and RPE w.r.t translation portions to simplify analysis because the problem is sequential and rotation errors would show up as errors in translation later. Fig. 10 shows that in most sequences, the proposed algorithm performed similarly to the state-of-the-art LeGO-LOAM on ATE and RPE, especially in sequence 1, which had better results. Even if the performance on sequence 07 was not good enough (Table 1), the difference was small on average but had a more concentrated distribution (Fig.10). The final translation error performed nearly the best or second-best as listed in Table 2. We can also infer that our approach had the best adaptability for degenerate scenes with fast driving and less features because it demonstrated the best results on ATE and RPE in sequence 01.

TABLE 1. The ATE-RMSE w.r.t translation part on the 4 sequences (m).

	00	01	05	07
NDT-graph-SLAM	12.75	63.20	6.17	2.77
NDT-Odometry	17.39	474.59	14.49	2.01
ICP-graph-SLAM	26.02	607.95	8.64	2.64
LeGO-LOAM	2.59	231.17	1.46	0.86

TABLE 2. The RPE-RMSE w.r.t translation part on the 4 sequences (%).

	00	01	05	07
NDT-graph-SLAM	1.2238	2.8176	1.0672	1.0094
NDT-Odometry	1.2286	2.9313	1.0781	1.0287
ICP-graph-SLAM	1.2364	3.6935	1.0898	1.0434
LeGO-LOAM	1.1657	3.8175	1.2921	1.0020

To explore the potential for real-time application, we ran the proposed algorithms on KIITI dataset at 10% of the real-time speed (similar to LeGO-LOAM in [20]) and recorded the time cost by each module. The front-end modules of prefilter,

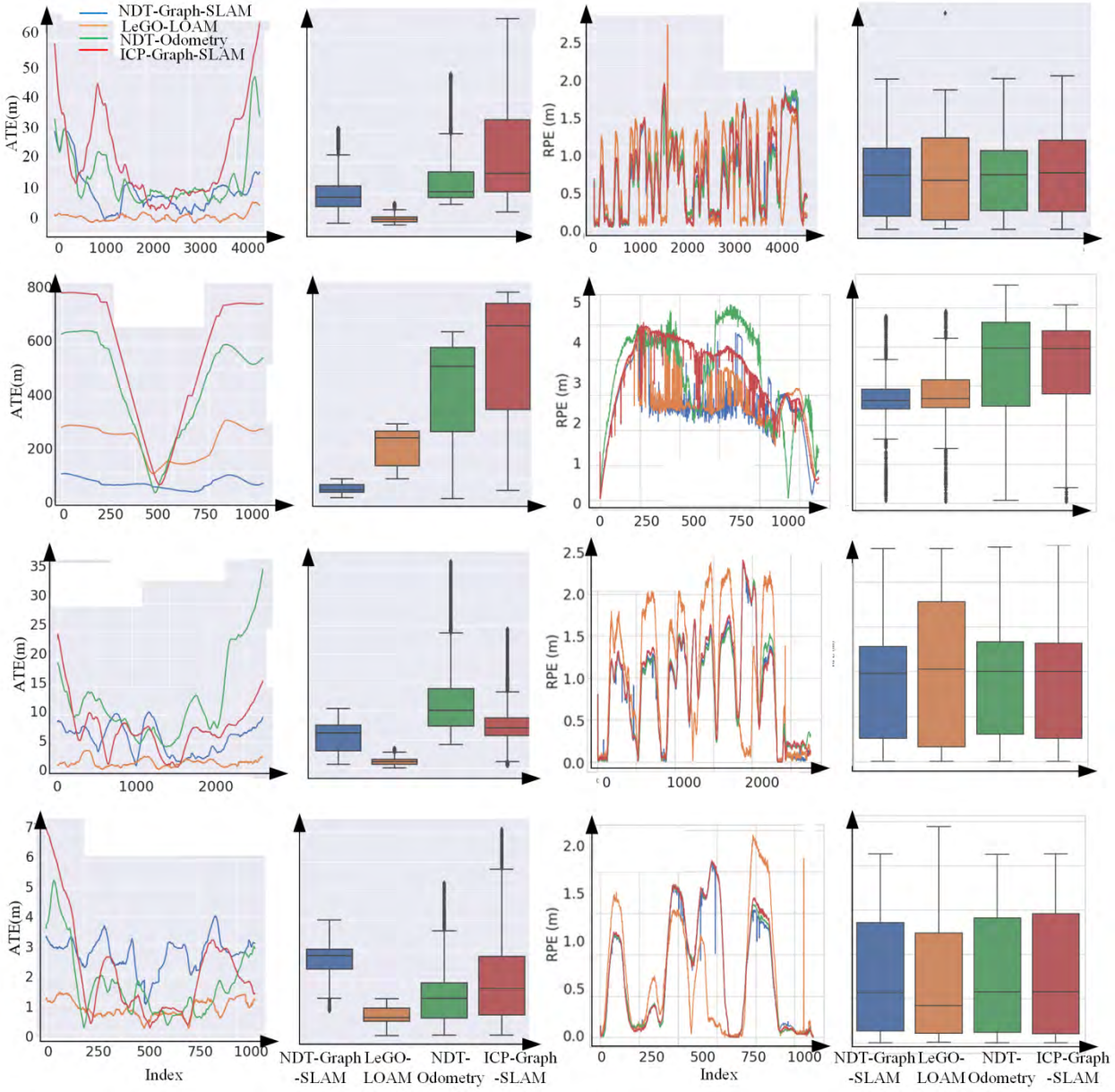

FIGURE 10. The line and box plots corresponding to the four sequences on ATE & RPE. The x-axis of the line-plot represents the poses index. The blue line and box plot result represent our proposed NDT-graph-SLAM. The orange line and box result represent LeGO-LOAM. The green line and box plot result represent NDT-Odometry approach. The red line and box plot result represent ICP-graph-SLAM.

scan registration, and planar segmentation should be performed in real-time to respond to the new scan frame. Loop detection and graph optimization were in another thread and operated at a low frequency of 3 s. The downsample resolution was set to 0.5m in this open environment. Table 3 shows that the front-end ran at 96.65 ms on average and could process a scan at 10Hz on average (Fig. 11). In practical applications, the number of iterations by scan registration will be constrained to ensure real-time performance. The downsample resolution can be adjusted to be larger to reduce the point number processed by scan registration and plane segmentation but should balance effects on accuracy and speed.

TABLE 3. Runtime of modules for processing once ms).

	Mean	Max	Min
Prefilter	25.86	339.03	5.00
Lidar Odometry	63.43	581.75	2.62
Planar segmentation	7.36	25.71	2.31
Loop closure	82.58	921.62	0.001
Graph optimization	1164.89	1683.25	26.08

The map update process by graph optimization together with loop closure required 2604.9ms, which is suitable for real-time mapping applications.

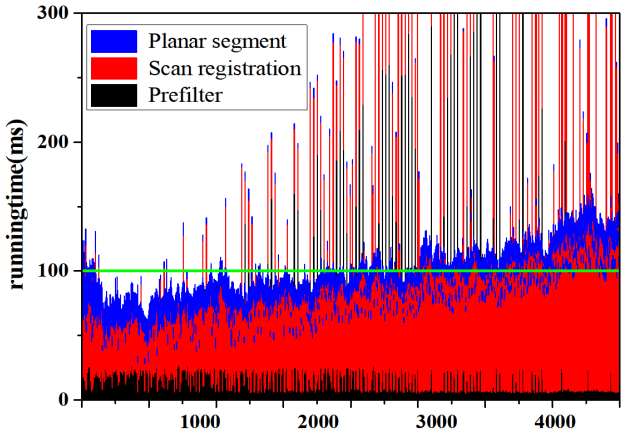

FIGURE 11. The run time needed for prefilter, planar segment and scan registration on KITTI dataset, which must be performed once a new scan came. The proposed algorithm runs at 96.65 ms on average (below the blue line of 100 ms) and have the ability to process a scan at 10Hz on average.

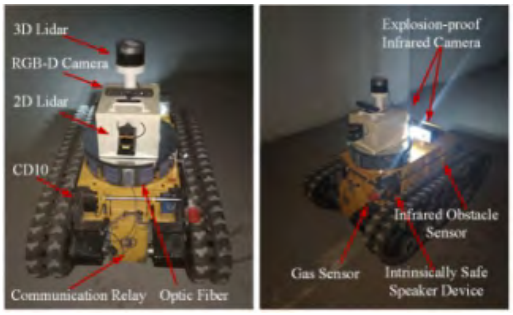

FIGURE 12. The explosion-proof mobile platform with multiple types of perception sensors. 3D lidar is the main sensor used in this paper, but RGB-D camera, 2D lidar and IMU are also deployed in the sensor suite for further study. There are also other sensors like multi-gas sensor (CD10, a kind of integrated sensors which can detect 10 kinds of environmental information), gas sensor (measure gas concentration at different position in vertical direction), infrared obstacle sensor, and intrinsically safe speaker device, which are all used for environmental detection.

B. FIELD TESTS

1) ROBOT PLATFORM

In our previous work [45], we designed a multi-drive crawler walking mechanism to ensure high maneuverability of the CMRR when dealing with complicated terrain. Fig.12 presents the new generation of a CMRR system, an explosion-proof mobile platform with multiple perception sensors. The system is equipped with a separate sensors suite, including a 3D sweep lidar (rslidar-16,12V/9W), motor-driven rotating 2D lidar (UTM-30LX, 12V/8.4W), IMU (Xsens Mti-G-710, in the suite), and RGB-D camera (Xtion Pro Live, powered by USB2.0 interface). All sensors are connected to the mobile platform by physical isolation of a power supply and network cable. The suite is made of flame-retardant anti-static material to meet coal mine application requirements. In this work, we only used the 3D lidar for SLAM work.

Coal mine equipment should be explosion-proof. Either the platform or the carrying sensors must be explosion-proof to ensure they will not become detonation sources for accumulated gas in the laneway. The main control circuits and drivers are deployed in the flameproof chamber. All sensors exposed outside are intrinsically safe devices, which form an explosion-proof and intrinsically safe system with the platform. A laser beam is a possible source of danger, but studies have shown that gas and dust below 150 mW or 20 mW/mm2 pose no threat [5], which can be satisfied by most sweep lidars. Furthermore, multi-gas sensors (CD10) and the gas sensor are used to detect explosive gases in the environment. Once the concentration exceeds the limit, the internal circuit triggers a relay and cuts off power to the external sensor including the 2D, and 3D lidar, as well as the RGB-D camera.

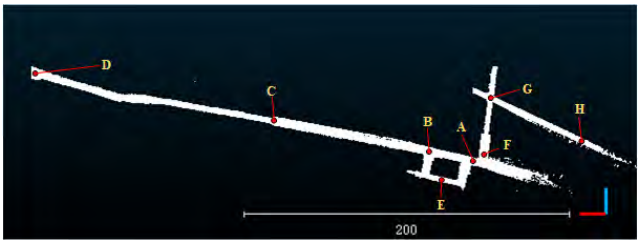

FIGURE 13. The top-view of the map generated by NDT-SLAM with several scan poses. The robot started at point A and followed this sequence A-B-C-B-D-C-B-E-A-F-G-H. Stationary reflective markers were arranged at cross of A, C, D, F, G.

2) EXPERIMENTAL RESULTS

Field tests were carried out in the gas explosion laboratory of CUMT which fully simulates the environment of an underground mine. The top-view map of Fig. 13 was generated by our SLAM algorithm. The traverse started at point A and continued to points B, C, and D. Then the robot returned back from points C, B, E and created a small loop at A. Finally, the robot ran through points F, G and stopped at point H. There were four junctions (A, B, F, and G) and only area from point A to C was illuminated. It constituted a long, narrow laneway nearly 200 m long and 2.5m wide from C to D, including a slight curvature and no lighting. The F-G portion was a variable-width roadway: one had a width of 4.75 m and length of 15 m; the other is a width of 2.5 m and length of 55 m. The region from G to H was a downward sloping roadway. Fig. 14 presents the four local maps and photos respectively generated from B to C, point A, from F to G, and point G in Fig.13.

During all tests, the robot moved at a speed of roughly 0.5 m/s and was controlled by tele-operation. Combined with wired and wireless, packaged data were sent back to the host computer for controlling and displaying. T ROSBAG tool was used to record the original point cloud, tf, map, and other topical information, which were used in subsequent analysis. The robot also used an IPC with CPU: i7-3555LE (2.5 GHz, 4 core), and 8G memory for data processing and algorithm implementation.

It is a challenging to obtain ground-truth poses for a CMRR due to occlusions and the aspect ratio of narrow and

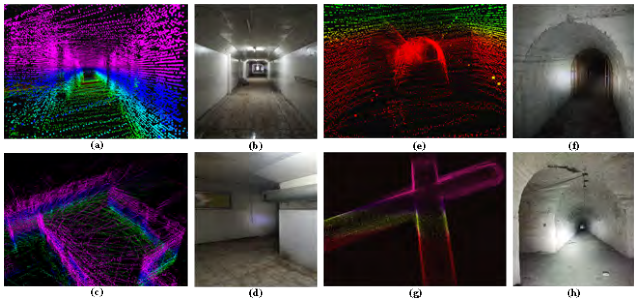

FIGURE 14. Maps generated in the field tests and their actual positions in Fig.13. (a-b) is a long laneway with same width from B to C. (c-d) is a branch in the tunnel, from E to A. (e-f) is a dark tunnel with different width, between F and G. (g-h) is a dark tunnel junction, near G. The robot moved on an average speed of 0.5m/s.

long laneways. We placed reflective markers at several key locations at points A, C, D, F, and G and used a tape ruler to manually measure the distance between reflective markers as the ground truth to reflect modeling accuracy in the forward direction.

TABLE 4. Relative error of mapping between the arranged reflected markers in Figure 13.

	Ground Truth(m)	Measurement(m)	Relative Error
A-C	118.0	112.5	-4.7%
C-D	200.0	152.0	-24%
F-G	70.0	49.7	-29%

Table 4 lists the relative errors of the mapping results. The error in the segment of A-C was only 4.7% to 118m, far less than 24% to 70m in the C-D segment and 29% to 200m in the F-G segment. This result is reasonable because from A to C, A and B were junctions and had many wall surfaces in different directions along with air doors and ventilation pipes, which provided rich environmental features. The C-D and F-G segments were typical degenerate scenarios and only had walls of equal width on each side, which could easily cause registration failure. The robot mistakenly believed it did not move, resulting in mapping a length smaller than that of the actual roadway. There was only a small loop from point B, E to point A. Our algorithm detected the loop and optimized the mapping result successfully. Although loop detection and graph optimization could improve mapping consistency, these processes could only ensure that the map was continuous; they could not eliminate error caused by the registration error of lidar odometry.

In coal mine rescue missions, the topological relationship of environmental characteristics and the robot position relative to the side wall of the laneway are of great help to rescue responders in visual observation of teleoperation control. An accurate environmental model and the position of the robot extending the roadway direction are not of great concern. For autonomous navigation, the position of the robot to the side of the laneway must be accurate to ensure it will not hit the wall. The impact of inaccurate localization caused by degenerate problems in the forward direction on driving safety is not fatal beacuse once irregular obstacles appear, degradation will be eliminated. Nevertheless, we seek to develop SLAM algorithms that are more accurate and meet computational requirements, which will be our future work. LeGO-LOAM overly reliant on line features and surface features in this scenario, which cannot always exist in the degraded scenes of coal mine laneway. In the tunnel-like main roadway, few line features exist because of the parallel structured environment. Lego-LOAM cannot extract sufficient line features, leading to registration failure. In these cases, the NDT algorithm which is a distribution-based registration approach, has a higher matching success rate. Although it also suffers from the degradation problem, the worst result is that the estimated distance may be shorter than the true distance. The utilization of planar landmarks in graph optimization further improves modeling and positioning accuracy.

TABLE 5. Runtime of modules for processing once(ms).

	Mean	Max	Min
Prefilter	3.02	27.24	1.04
Lidar odometry	38.25	130.78	0.91
Planar segmentation	2.21	29.93	0.08
Loop detection	214.23	1392.01	0.00
Graph optimization	92.20	173.29	10.56

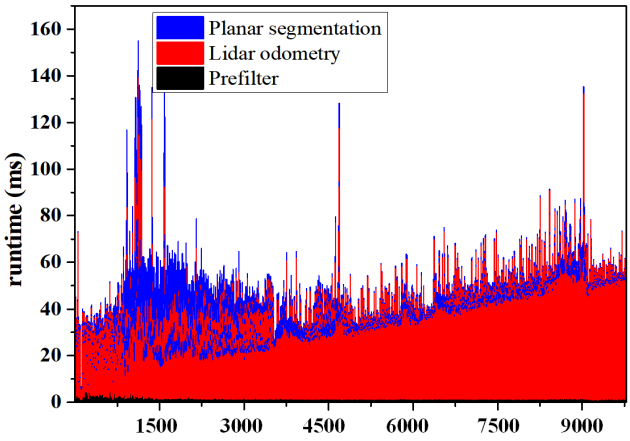

FIGURE 15. The run time needed for prefilter, planar segment and scan registration on the coal mine dataset, which must be performed once a new scan came.

Table 5 shows the time cost of each module to process our coal mine dataset. The downsample resolution was set to 0.1 m in the closed and narrow coal mine laneway. The front-end processing time needed about 43.48 ms on average and 187.95 ms at most to process a scan at 10 Hz on average (Fig.15). The time cost reduction occurred because our lidar (16 beams) contained only 75% of the original point cloud data compared to the lidar (64 beams) used in the KITTI dataset. The map update processed by graph optimization with loop closure requried at most 1565.3 ms at most, which can meet requirement of real-time mapping in 3 s.

V. CONCLUSION AND FUTURE WORK

In this paper, a new 3D laser-based SLAM algorithm with loop closure and graph optimization is proposed. 3D-NDT is taken to construct the lidar odometry constraints and planar surfaces in the coal mine laneway are extracted by iterative RANSAC to serve as the landmarks, improving the mapping consistency and localization robustness. Lightweight loop closure is realized by detecting appearance similarity and distance between keyframes. Multi-thread acceleration is used to further improve the computation speed to meet real-time applications. Evaluation of KITTI datasets is compared with the ICP-graph-SLAM and NDT-Odometry approach, showing the effectiveness of our improved approach on loop detection and pose graph building. The results of the field test in an underground coal mine demonstrated consistency with the actual environment, and the time cost can satisfy the application of localization and mapping for CMRRs in coal mine rescue missions.

In future work, a multi-sensor fusion including IMU and infrared vision will be carried out to improve accuracy performance in degenerate directions. After the coal mine model is constructed, a multi-sensor integrated positioning approach based on NDT will also be further studied, which could be used for precise positioning of other mine robots and trackless vehicles to solve the problem of a limited location method in the mine environment.

REFERENCES

[1] A. Morris *et al.*, "Recent developments in subterranean robotics," *J. Field Robot.*, vol. 23, no. 1, pp. 35–57, Jan. 2006.
[2] R. R. Murphy, J. Kravitz, S. L. Stover, and R. Shoureshi, "Mobile robots in mine rescue and recovery," *IEEE Robot. Autom. Mag.*, vol. 16, no. 2, pp. 91–103, Jun. 2009.
[3] W. Wang, W. Dong, Y. Su, D. Wu, and Z. Du, "Development of search-and-rescue robots for underground coal mine applications," *J. Field Robot.*, vol. 31, no. 3, pp. 386–407, May/Jun. 2014.
[4] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.
[5] S. Thrun *et al.*, "Autonomous exploration and mapping of abandoned mines," *IEEE Robot. Autom. Mag.*, vol. 11, no. 4, pp. 79–91, Dec. 2004.
[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] D. F. Huber and N. Vandapel, "Automatic three-dimensional underground mine mapping," *Int. J. Robot. Res.*, vol. 25, no. 1, pp. 1–7, Jan. 2006.
[8] F. Moosmann and C. Stiller, "Velodyne SLAM," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 393–398.
[9] E. López *et al.*, "A multi-sensorial simultaneous localization and mapping (SLAM) system for low-cost micro aerial vehicles in GPS-denied environments," *Sensors*, vol. 17, no. 4, p. 802, Apr. 2017.
[10] M. Bosse, R. Zlot, and P. Flick, "Zebedee: Design of a spring-mounted 3-D range sensor with application to mobile mapping," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1104–1119, Oct. 2012.
[11] J. Zhang and S. Singh, "Enabling aggressive motion estimation at low-drift and accurate mapping in real-time," in *Proc. IEEE Int. Conf. Robot. Autom.*, May/Jun. 2017, pp. 5051–5058.
[12] M. Leingartner, J. Maurer, A. Ferrein, and G. Steinbauer, "Evaluation of sensors and mapping approaches for disasters in tunnels," *J. Field Robot.*, vol. 33, no. 8, pp. 1037–1057, Dec. 2016.
[13] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," *J. Field Robot.*, vol. 24, no. 10, pp. 803–827, Oct. 2007.
[14] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Auto. Robots*, vol. 34, no. 3, pp. 133–148, Apr. 2013.
[15] L. Armesto, J. Minguez, and L. Montesano, "A generalization of the metric-based iterative closest point technique for 3D scan matching," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 1367–1372.
[16] A. V. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Proc. Robot., Sci. Syst. Conf.*, Jun./Jul. 2009, pp. 1–8.
[17] F. Pomerleau, F. Colas, and R. Siegwart, "A review of point cloud registration algorithms for mobile robotics," *Found. Trends Robot.*, vol. 4, no. 1, pp. 1–104, May 2015.
[18] O. Wulf, A. Nüchter, J. Hertzberg, and B. Wagner, "Benchmarking urban six-degree-of-freedom simultaneous localization and mapping," *J. Field Robot.*, vol. 25, no. 3, pp. 148–163, 2008.
[19] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann, "6D SLAM—3D mapping outdoor environments," *J. Field Robot.*, vol. 24, nos. 8–9, pp. 699–722, Aug./Sep. 2007.
[20] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *Proc. Int. Conf. Intell. Robots Syst.*, Oct. 2018, pp. 4758–4765.
[21] K. Pathak, A. Birk, N. Vaskevicius, M. Pfingsthorn, S. Schwertfeger, and J. Poppinga, "Online three-dimensional SLAM by registration of large planar surface segments and closed-form pose-graph relaxation," *J. Field Robot.*, vol. 27, no. 1, pp. 52–84, Jan./Feb. 2010.
[22] K. Pathak, N. Vaskevicius, J. Poppinga, M. Pfingsthorn, S. Schwertfeger, and A. Birk, "Fast 3D mapping by matching planes extracted from range sensor point-clouds," in *Proc. Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 1150–1155.
[23] J. Weingarten and R. Siegwart, "3D SLAM using planar segments," in *Proc. Int. Conf. Intell. Robot. Syst.*, Oct. 2006, pp. 3062–3067.
[24] A. J. B. Trevor, J. G. Rogers, and H. I. Christensen, "Planar surface SLAM with 3D and 2D sensors," in *Proc. IEEE Int. Conf. Robots Autom.*, May 2012, pp. 3041–3048.
[25] P. Biber and W. Strasser, "The normal distributions transform: A new approach to laser scan matching," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Oct. 2003, pp. 2743–2748.
[26] M. Magnusson, A. Nüchter, C. Lörken, A. J. Lilienthal, and J. Hertzberg, "Evaluation of 3D registration reliability and speed—A comparison of ICP and NDT," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3907–3912.
[27] M. Magnusson, N. Vaskevicius, T. Stoyanov, K. Pathak, and A. Birk, "Beyond points: Evaluating recent 3D scan-matching algorithms," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2015, pp. 3631–3637.
[28] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg, "The efficient extension of globally consistent scan matching to 6 DoF," in *Proc. Int. Symp. 3D Data Process., Vis. Transmiss.*, Jun. 2008, pp. 29–36.
[29] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intell. Transp. Syst. Mag.*, vol. 2, no. 4, pp. 31–43, 2010.
[30] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—A modern synthesis," in *Proc. Int. Workshop Vis. Algorithms, Theory Pract.*, Sep. 1999, pp. 298–372.
[31] Y. Latif, C. Cadena, and J. Neira, "Robust loop closing over time for pose graph SLAM," *Int. J. Robot. Res.*, vol. 32, no. 14, pp. 1611–1626, Oct. 2013.
[32] J. Sivic and A. Zisserman, "Efficient visual search of videos cast as text retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 591–606, Apr. 2009.
[33] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2016, pp. 1271–1278.
[34] M. Magnusson, H. Andreasson, A. Nuchter, and A. J. Lilienthal, "Appearance-based loop detection from 3D laser data using the normal distributions transform," in *Proc. Int. Conf. Robot. Autom.*, May 2009, pp. 23–28.
[35] E. Mendes, P. Koch, and S. Lacroix, "ICP-based pose-graph SLAM," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot.*, Oct. 2016, pp. 195–200.
[36] E. Einhorn and H.-M. Gross, "Generic 2D/3D SLAM with NDT maps for lifelong application," in *Proc. IEEE Eur. Conf. Mobile Robots*, Sep. 2013, pp. 240–247.
[37] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1–4.
[38] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, Nov. 2007, pp. 225–234.

[39] G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese, and C. Hertzberg, "Hierarchical optimization on manifolds for online 2D and 3D mapping," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 273–278.
[40] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. New York, NY, USA: Cambridge Univ. Press, 2004.
[41] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G^2o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 3607–3613.
[42] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. Robots*, vol. 34, no. 3, pp. 189–206, Apr. 2013.
[43] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Aug. 2013.
[44] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580.
[45] Y.-T. Li, H. Zhu, M.-G. Li, and P. Li, "A novel explosion-proof walking system: Twin dual-motor drive tracked units for coal mine rescue robots," *J. Central South Univ.*, vol. 23, no. 10, pp. 2570–2577, Oct. 2016.

MENGGANG LI received the master's degree in mechanical engineering from the China University of Mining and Technology, Xuzhou, China, in 2017, where he is currently pursuing the Ph.D. degree in vehicle engineering. His research interests include positioning and navigation for special robots.

HUA ZHU received the Ph.D. degree from the School of Mechatronic Engineering, China University of Mining and Technology, Xuzhou, China, where he is currently a Professor, the Supervisor of Ph.D. students, and the Deputy Director of the Mine Robot Research Institute. His research interests include the tribology theory and application, fractal and chaos theory, noise and vibration control, and mine rescue robot research.

SHAOZE YOU received the bachelor's degree in mechanical engineering from the Shandong University of Science and Technology, Qingdao, China, in 2016. He is currently pursuing the Ph.D. degree with the School of Mechatronic Engineering, China University of Mining and Technology, Xuzhou, China. His research interests include computer vision and intelligent robot.

LEI WANG received the bachelor's degree in mechanical engineering from Jiangsu Normal University, Xuzhou, China, in 2017. He is currently pursuing the master's degree with the School of Mechatronic Engineering, China University of Mining and Technology, Xuzhou. His research interests include computer vision and intelligent robot.

CHAOQUAN TANG received the Ph.D. degree from the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China. He is currently a Lecturer with the School of Mechatronic Engineering, China University of Mining and Technology. His research interests include robot modeling and control, bionic robot, and mine rescue robot research.