Robotica (2024), **42**, pp. 3480–3504
doi:10.1017/S0263574724001474

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

A semantic knowledge database-based localization method for UAV inspection in perceptual-degraded underground mine

Qinghua Liang[1], Minghui Zhao[2,3], Shigang Wang[1] and Min Chen[1]

[1]Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China
[2]School of Electronics and Information Engineering, Tongji University, Shanghai, China
[3]China Coal Technology & Engineering Group Shanghai Co. Ltd. Shanghai, China
Corresponding author: Min Chen; Email: 13541210030@163.com

Received: 6 October 2023; **Revised:** 4 August 2024; **Accepted:** 11 August 2024; **First published online:** 30 October 2024

Keywords: robot localization; aerial robotics; pose estimation and registration; topological modeling of robots; automation

Abstract

In recent years, unmanned aerial vehicles (UAVs) have been applied in underground mine inspection and other similar works depending on their versatility and mobility. However, accurate localization of UAVs in perceptually degraded mines is full of challenges due to the harsh light conditions and similar roadway structures. Due to the unique characteristics of the underground mines, this paper proposes a semantic knowledge database-based localization method for UAVs. By minimizing the spatial point-to-edge distance and point-to-plane distance, the relative pose constraint factor between keyframes is designed for UAV continuous pose estimation. To reduce the accumulated localization errors during the long-distance flight in a perceptual-degraded mine, a semantic knowledge database is established by segmenting the intersection point cloud from the prior map of the mine. The topological feature of the current keyframe is detected in real time during the UAV flight. The intersection position constraint factor is constructed by comparing the similarity between the topological feature of the current keyframe and the intersections in the semantic knowledge database. Combining the relative pose constraint factor of LiDAR keyframes and the intersection position constraint factor, the optimization model of the UAV pose factor graph is established to estimate UAV flight pose and eliminate the cumulative error. Two UAV localization experiments conducted on the simulated large-scale Edgar Mine and a mine-like indoor corridor indicate that the proposed UAV localization method can realize accurate localization during long-distance flight in degraded mines.

1. Introduction

The unmanned aerial vehicle (UAV) has the advantages of lightness, flexibility, and highly maneuverable. By equipping the UAV with some environmental sensing equipment, such as LiDAR and vision sensors, the UAV can replace workers to complete complex patrol inspection and other similar tasks in the underground mine. In this way, the incidence of coal mine safety accidents can be greatly reduced, and production efficiency can be improved. Furthermore, mechanization, automation, informatization, and intelligence of the coal mining industry could be rapidly promoted. However, the underground mine environment is extremely complex. The roadways of the mine are narrow and featureless. There are no GPS signals and the light condition is poor. These limitations make it difficult for the existing localization methods to achieve accurate positioning in the underground mine. The two commonly used types of sensors for localization in general environments are camera and LiDAR. Therefore, we review vision-based and LiDAR-based localization methods applied in the underground mine environment below.

© The Author(s), 2024. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

Vision-based localization

ORB-SLAM [1–3] is currently the most widely used robot localization method based on visual feature point matching. Rogers et al. [4] tested the localization accuracy of ORB-SLAM2 in the open-source underground mine. However, the lack of image texture features in underground roadways leads to the maximum localization error of ORB-SLAM up to 30 m. Compared to visual localization algorithms based on feature matching, direct sparse visual odometry exploits pixel gradients rather than image features to estimate the relative pose change, which is usually applied in featureless environments. Özaslan et al. [5–7] equipped a UAV with an active light and a camera and estimated the axial displacement of the UAV based on Lucas–Kanade optical flow [8] during flight in a dark tunnel. The experimental results show that the maximum localization error is up to 40%. The large localization error results from the fact that the active lighting carried by UAV during the flight in the dark tunnel makes it difficult to satisfy the theory of the gray-scale invariance assumption of optical flow. To improve the localization accuracy of the vision-based method in underground mines, Jacobson et al. [9] proposed a semi-supervised SLAM algorithm for pose tracking of mining vehicles. The semi-supervised SLAM established multiple positioning nodes and stored the keyframe images to form an image database and then used the optical flow to estimate the vehicle pose. Whereas, the semi-supervised SLAM requires a large amount of image data and manual intervention. With the development of robot intelligence, semantic features [10] are conducive to navigation and localization. Furthermore, to cope with the sensitivity of lighting changes and motion blur of single-camera images, vision inertial odometry (VIO) was applied to the localization of robots in the underground mine. Kramer et al. [11] tested the localization accuracy of existing OKVIS [12] algorithms. Chen et al. [13] evaluated VIO-Mono [14] state estimator's accuracy in featureless indoor environments like underground mine environment. Papachristos et al. [15–17] combined the images from an infrared camera and data from an IMU to localize the UAV indoors and proposed RITIO and KTIO algorithm [18]. Compared with visible light images, although infrared images can be adapted to the harsh lighting of underground mines to some extent, the lack of enough features leads to a large localization error. In summary, since the mine is located hundreds of meters underground, where most areas have only weak light and some areas are dark, the vision-based localization method cannot capture enough features or satisfy the gray-scale invariance assumption, making it difficult to achieve accurate localization.

LiDAR-based localization

In the visually degraded underground mine environment, LiDAR sensors can directly measure the distance and orientation information of the target object, which has the advantages of wide-ranging scope and high sensing accuracy compared to vision sensors. Thus, the LiDAR-based method is more suitable for robot localization and mapping [19]. Meanwhile, the LiDAR data is almost unaffected by the changes in ambient light, making it suitable for dark coal mine environments. LiDAR sensors can be divided into single-line LiDAR (2D LiDAR) and multi-line LiDAR (3D LiDAR) depending on the number of laser beams. 2D LiDAR was commonly used in the localization of indoor mobile robots in the early stage, which can only scan in a plane and cannot measure the height of the object. The ATRV mobile robot developed by Bakambu et al. [20] was equipped with two 2D LiDARs to explore the mine. They used line segments scanned by 2D LiDAR as the basic elements for localization and mapping. In addition, the Groundhog robot developed by Sebastian et al. [21] also used the point cloud data from 2D LiDAR aligned with a map to estimate the pose of the robot based on the Iterative Closest Point (ICP) matching. Since the 2D LiDAR cannot measure the spatial information, the above research can only acquire the x-y translation and heading angle of the mobile robot. Compared to 2D LiDAR, 3D LiDAR can measure the spatial geometric structure of the object. LOAM [22] and Lego_LOAM [23] are two typical 3D point cloud-based localization algorithms. Li et al. [24] combined Lego_LOAM with the Normal Distributions Transform (NDT)-based feature matching algorithm to localize the pose of a mobile robot in a mine. The experimental results showed that the maximum error was up to 29%.

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

Papachristos et al. [25] fused LOAM and IMU based on the Extended Kalman Filter (EKF) to estimate the pose of a UAV during flight in a mine. However, they did not quantitatively analyze the localization accuracy of the UAV during flight since it is difficult to obtain the ground truth of the UAV pose in the real mine environment. Chow et al. [26] conducted experimental analysis in an indoor environment for three SLAM methods: Hector_slam [27], Gmapping [28], and Cartographer [29]. The results showed that the trajectory generated by Cartographer fluctuated greatly and Gmapping failed in some tests. Koval et al. [30] used a dataset collected in an underground area with multiple tunnels to conduct experimental analysis of various LiDAR-based positioning algorithms. However, all of the tested algorithms showed large drift in the Z-axis since the underground mine environment had sparse features and only one LiDAR was used. In addition to traditional pose estimation methods, Wang et al. [31] proposed a supervised 3D point cloud learning model, PWCLO-Net, for LiDAR localization. However, the supervised learning method requires a large amount of point cloud data labeled with ground truth for training. However, the ground truth trajectory in the underground mine is hard to obtain. In conclusion, LiDAR sensors have the advantage of accurate ranging, which are not affected by illumination. The localization accuracy of the LiDAR-based method can reach the decimeter level in general outdoor scenes. However, in degraded mine environments, the sparse point clouds measured by LiDAR cannot directly distinguish the similar geometry structure of the mine roadway, resulting in a non-negligible cumulative error in the LiDAR-based localization method.

Multi-sensor fusion-based localization

As mentioned above, each type of sensor has inherent defects, such as the visible light vision sensor being sensitive to illumination, the resolution of infrared vision image being low, and the LiDAR point cloud being sparse. Multi-sensor data fusion can effectively incorporate the advantages of each sensor to better adapt to complex mines. Alexis et al. [32] proposed a concurrent fusion framework consisting of LOAM, ROVIO, and KTIO to estimate the pose of a UAV in the mine. However, they did not perform a quantitative evaluation of the localization accuracy. Jacobson et al. [33] fused LiDAR point cloud with IMU data for pose estimation and used visual images for scene recognition. However, their method needs extensive human intervention to build maps for localization. In addition, some studies combined active sensors such as RFID, UWB, and WIFI with environmental sensing sensors such as LiDAR and camera to localize robot in a mine. Lavigne et al. [34] fused 2D LiDAR, RFID, and absolute optical encoders to construct an underground localization network of passive RFID tags. Nevertheless, this method restricts the localization to a two-dimensional plane, which results in a large deviation in the robot's pose when the height of the mine changes. Li et al. [35] proposed an underground mine pseudo-GPS system, composed of UWB to localize the robot in the mine. Unfortunately, the deployment cost of UWB is high, which limits its practical application in underground positioning. Wang et al. [36] integrated multi-source data containing WIFI, LiDAR, and a camera for estimating the pose of an underground search and rescue robot. However, the intricate tunnel structure of the mine has a large impact on the signal transmission, making it difficult to obtain the robot's pose in areas where the WIFI signal intensity is low. In summary, the existing multi-sensor fusion-based localization methods are also limited in the perceptually degraded mine, and it is challenging to accurately estimate the UAV pose in the complex mine.

Based on the above review of localization methods for the robot/UAV in mines, the vision-based and LiDAR-based methods suffer from localization error accumulation due to the lack of enough image features and high similarity geometric structure. Most of the current robot/UAV localization methods for underground mines are based on the alignment of image features or geometric features. While with the development of the robot intelligence, semantic features [37, 38] are conducive to improving navigation and localization. Inspired by manual inspection, intersections are the critical semantic features to achieve accurate UAV localization in underground mines. Therefore, based on our previous work proposed in [39] and [40], this paper proposes a semantic knowledge database-based localization method for UAVs in underground mines. Based on the intersection recognition method proposed in [40], a semantic

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

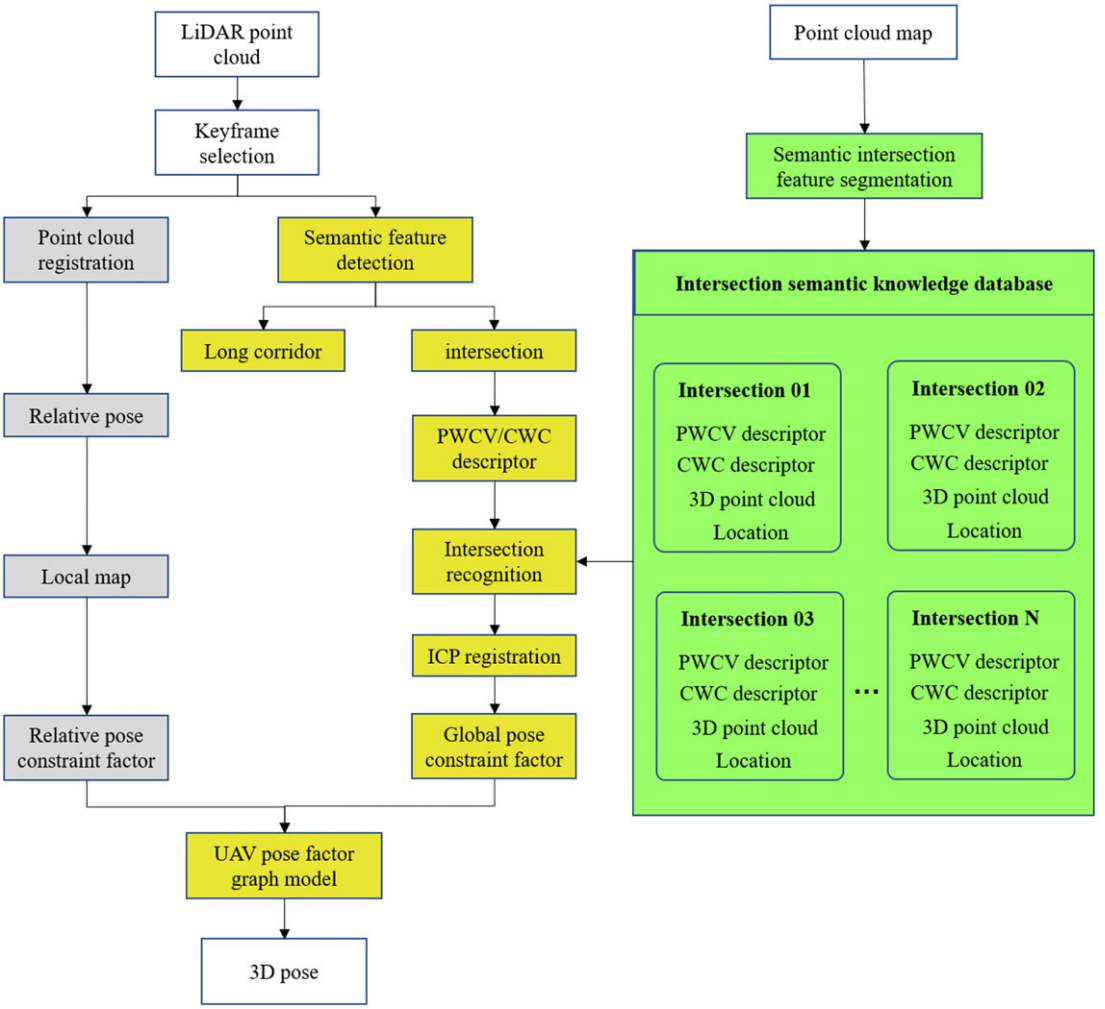


Figure 1. *The framework for UAV pose estimation based on semantic knowledge database.*

knowledge database is established by segmenting the intersection point cloud from the pre-built map of the mine. Furthermore, the global pose constraint of the current frame with the semantic knowledge database is constructed by detecting semantic intersection features of the mine in real time during UAV flight. Combining the relative pose constraints between the keyframes of the LiDAR point cloud, the pose graph model is established for UAV pose optimization.

2. System overview

To realize stable UAV flight over a long distance in underground mines, a UAV pose estimation framework is proposed in this paper based on LiDAR odometry constraint and semantic intersection pose constraint. The framework is shown in Figure 1. The gray part represents the process of relative pose constraint factor construction, and the yellow part represents the process of global pose constraint factor construction.

To construct the relative pose constraint during UAV flight, the keyframe selection procedure is performed first to improve the computational efficiency of pose estimation, using equal distance intervals and equal angle intervals selection strategies [41, 42]. That is, if the relative translation distance between

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

the new frame and the last keyframe is larger than a set distance threshold, or if the rotation angle between the new frame and the last keyframe is larger than a set angle threshold, the new frame will be selected as the keyframe.

Based on the selected keyframes, the geometric features of the point cloud are extracted for estimating the pose change between the neighboring keyframes. Inspired by the feature-extracting method proposed by LOAM, the curvature features of each point in the measured LiDAR point cloud are calculated for extracting edge features and plane features. By establishing the point-to-line and point-to-plane spatial distance functions based on the extracted edge and plane features between keyframes, the relative pose and local map can be estimated. Furthermore, the relative pose constraint factor is constructed taking into account the noise during UAV flight.

As stated in [43, 44], map-based localization algorithms are currently considered as the most accurate ones. In this paper, the intersection semantic knowledge database is established based on the pre-built point cloud map. The pre-built point cloud maps are often referred to as High Definition Maps (HD Maps). The resolution of HD Maps can reach centimeter-level accuracy. The intersection pose constraint factor building process is shown in the yellow part of Figure 1. The intersection semantic knowledge database is established based on the pre-built point cloud map. The green part of Figure 1 shows the building procedures of mine intersection semantic knowledge database. The pre-built point cloud map of the underground mine is the basis of the semantic knowledge database, which can be reconstructed by our previous work proposed in [39]. Based on the pre-built point cloud map, the dense point cloud information of each intersection is segmented. Accordingly, the location of each intersection, as well as the Polar Wall Contour Vector (PWCV) and Cylinder Wall Contour (CWC) descriptors, are generated for building the intersection semantic knowledge database. Then, the semantic features of the sampled LiDAR point cloud during UAV flight are detected in real time. The candidate intersection is selected by comparing the intersection similarity between the current keyframe and the intersections in the semantic knowledge database. Next, the ICP algorithm is applied to register the detected intersection with the corresponding intersection in the mine intersection semantic knowledge database. If the ICP distance after registration is below the minimum ICP distance threshold, the intersection pose factor is constructed. Combining the relative pose constraint factor and the intersection pose constraint factor, the pose factor graph model is established for UAV pose estimation.

3. Intersection-based factor graph model

The factor graph model is a Bayesian network-based graph optimization model, which is proposed by Kschischang et al. [45] and suitable for modeling complex state estimation problems in SLAM and Structure From Motion (SFM). The factor graph is an undirected graph that contains two types of nodes, that is, variable nodes and factor nodes. The variable nodes represent the unknown variables to be estimated, such as the UAV pose during flight. The factor nodes represent probabilistic constraint relationships between variable nodes, which can be obtained from the sensor measurements or prior knowledge. The factor graph can handle back-end optimization problems incrementally. When the new edges and nodes are added to the graph model, only the nodes connected to the newly added nodes need to be optimized. Therefore, this paper proposes an intersection-based factor graph model for UAV pose estimation.

The proposed UAV pose factor graph model is shown in Figure 2. The UAV pose $\boldsymbol{x}_i$ is the variable node to be estimated. I_k is the position of the intersection. The gray box represents the relative pose constraint factor between LiDAR keyframes. The yellow box represents the intersection pose constraint factor between the current keyframe and the observed intersection. The pose variable of the graph factor model to be optimized is denoted as:

$$\boldsymbol{\chi} = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_i, \ldots, \boldsymbol{x}_\tau\}, \boldsymbol{x}_i = \{\boldsymbol{t}_i^{\boldsymbol{W}}, \boldsymbol{\theta}_i^{\boldsymbol{W}}\} \tag{1}$$

where $\boldsymbol{\chi}$ is the set of pose variables to be estimated. $\boldsymbol{t}_i^{\boldsymbol{W}} = (t_x, t_y, t_z)$ is the x-y-z translation of the UAV in the world coordinate system. $\boldsymbol{\theta}_i^{\boldsymbol{W}} = (\theta_x, \theta_y, \theta_z)$ represents the roll-pitch-yaw rotation of the UAV in the

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

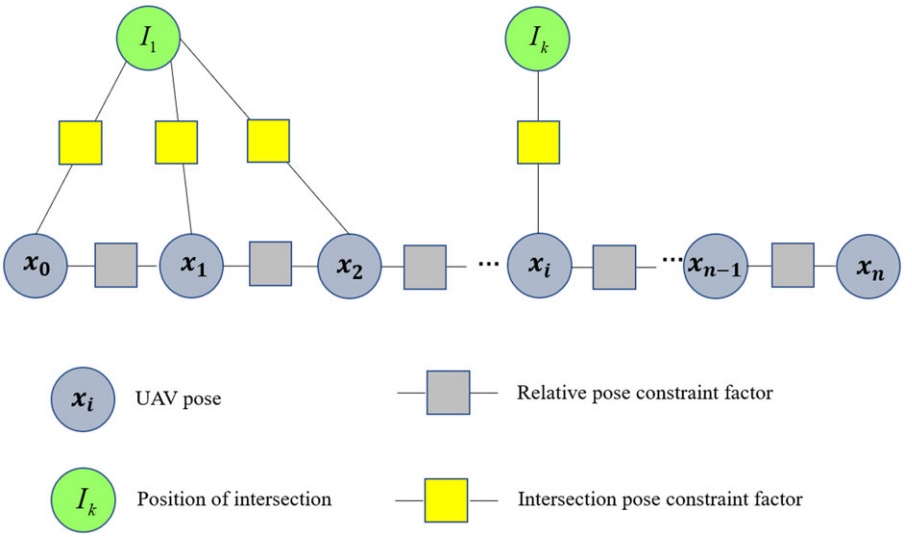


Figure 2. *The pose factor graph model of UAV.*

world coordinate system. According to the Bayesian rule, the joint probability expression of $\boldsymbol{\chi}$ is

$$P(\boldsymbol{\chi}|\boldsymbol{Z}) = \prod_{k,i} P(z_k|\boldsymbol{x}_i) \prod_i P(\boldsymbol{x}_i|\boldsymbol{x}_{i-1}) \tag{2}$$

where $\boldsymbol{Z}$ represents the set of observations. The first term $P(I_k|\boldsymbol{x}_i)$ is the likelihood probability, which represents the probability of obtaining observation z_k at UAV pose $\boldsymbol{x}_i$. The second term $P(\boldsymbol{x}_i|\boldsymbol{x}_{i-1})$ is the prior probability, which represents the relationship between pose $\boldsymbol{x}_{i-1}$ and $\boldsymbol{x}_i$. More generally, Eq. (2) can be represented by the product of relative pose constraint factor $\boldsymbol{\psi}_L(\boldsymbol{\chi}_i)$ and intersection pose constraint factor $\boldsymbol{\psi}_I(\boldsymbol{\chi}_i)$ as:

$$P(\boldsymbol{\chi}|\boldsymbol{Z}) = \prod \boldsymbol{\psi}_L(\boldsymbol{\chi}_i) \prod \boldsymbol{\psi}_I(\boldsymbol{\chi}_i) \tag{3}$$

where $\boldsymbol{\chi}_i$ denotes the set of variables $\boldsymbol{x}_i$ associated with the factor $\boldsymbol{\psi}_i$. In the following, we use unified $\boldsymbol{\psi}_i$ to represent the factors $\boldsymbol{\psi}_L$ and $\boldsymbol{\psi}_I$, respectively. Each factor $\boldsymbol{\psi}_i$ is a function of the pose variables in $\boldsymbol{\chi}_i$.

Based on the above derivation, the solution of the UAV pose factor graph model is to find the optimal estimation of the variables $\boldsymbol{\chi}$ for a prior relative pose and a known observation. It can then be converted into a maximum likelihood problem as:

$$\boldsymbol{\chi}^*_{MLE} = \arg\max \prod \boldsymbol{\psi}_L(\boldsymbol{\chi}_i) \prod \boldsymbol{\psi}_I(\boldsymbol{\chi}_i) \tag{4}$$

With the assumption that the measurement noise $\boldsymbol{\varpi}_i$ follows the Gaussian distribution [46], the maximum likelihood problem can be transformed into a simpler form. The general form of the measurement function is

$$z_i = h_i(\boldsymbol{\chi}_i) + \boldsymbol{\varpi}_i \tag{5}$$

where the noise term $\boldsymbol{\varpi}_i$ follows the distribution $\boldsymbol{\varpi}_i \sim \mathrm{N}(0, \boldsymbol{\Omega}_i)$. Then, the probability density function expansion of the corresponding constraint factor $\boldsymbol{\psi}_i$ takes the form:

$$\boldsymbol{\psi}_i(\boldsymbol{\chi}_i) \propto \exp\{-\tfrac{1}{2} \left\| h_i(\boldsymbol{\chi}_i) - z_i \right\|^2_{\boldsymbol{\Omega}_i}\} = \exp\{-\tfrac{1}{2} \left\| \boldsymbol{e}_i(\boldsymbol{\chi}_i) \right\|^2_{\boldsymbol{\Omega}_i}\} \tag{6}$$

where $\boldsymbol{e}_i(\boldsymbol{\chi}_i)$ denotes the residual function of the ith constraint factor $\boldsymbol{\psi}_i(\boldsymbol{\chi}_i)$. $\left\| \boldsymbol{e}_i(\boldsymbol{\chi}_i) \right\|^2_{\boldsymbol{\Omega}_i} = \boldsymbol{e}_i(\boldsymbol{\chi}_i)^T \boldsymbol{\Omega}_i^{-1} \boldsymbol{e}_i(\boldsymbol{\chi}_i)$ is defined as the square of the Mahalanobis distance with covariance matrix $\boldsymbol{\Omega}_i$.

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

Substituting Eq. (6) into Eq. (4), the maximum likelihood estimation of the pose variable $\boldsymbol{\chi}$ is equivalent to minimizing the negative logarithm of the constraint factor. That is, the objective function for the optimization of the UAV flight pose $\boldsymbol{\chi}^*_{MLE}$ can be established as:

$$\boldsymbol{\chi}^*_{MLE} = \arg\min_{\chi} \left\{ \left\| \boldsymbol{e}_L(\boldsymbol{\chi}_i) \right\|^2_{\boldsymbol{\Omega}_L} + \left\| \boldsymbol{e}_I(\boldsymbol{\chi}_i) \right\|^2_{\boldsymbol{\Omega}_I} \right\} \tag{7}$$

where $\boldsymbol{e}_L(\boldsymbol{\chi}_i)$ is the residual function of the LIDAR relative pose constraint factor $\boldsymbol{\psi}_L(\boldsymbol{\chi}_i)$. $\boldsymbol{\Omega}_L$ is the covariance matrix of the LIDAR relative pose estimation, which determines the weighting coefficients of the residual function $\boldsymbol{e}_L(\boldsymbol{\chi}_i)$ in the UAV pose objective function. $\boldsymbol{e}_I(\boldsymbol{\chi}_i)$ is the residual function of the intersection pose constraint factor $\boldsymbol{\psi}_I(\boldsymbol{\chi}_i)$. $\boldsymbol{\Omega}_I$ is the covariance matrix of the intersection observation, which determines the weighting coefficients of the residual function $\boldsymbol{e}_I(\boldsymbol{\chi}_i)$ in the UAV pose objective function. The following subsection will separately describe the modeling of the LiDAR relative pose constraint factor $\boldsymbol{\psi}_L(\boldsymbol{\chi}_i)$ and the intersection pose constraint factor $\boldsymbol{\psi}_I(\boldsymbol{\chi}_i)$ in detail.

3.1. LiDAR relative pose constraint factor

The scanning frequency of LiDAR is 10 Hz. The continuously scanned LiDAR point clouds contain some redundant information, so LiDAR keyframes are selected to build the UAV factor graph model. Therefore, the equal interval relative pose change sampling criterion is applied to ensure the uniform distribution of the LiDAR keyframes. That is, the first scanned LiDAR frame is selected as the first keyframe point cloud. Then, the newly scanned frame is aligned with the previous keyframe for calculating the relative pose change $\boldsymbol{\Delta T}$:

$$\boldsymbol{\Delta T} = \begin{bmatrix} \boldsymbol{\Delta R} & \boldsymbol{\Delta t} \\ 0 & 1 \end{bmatrix} \tag{8}$$

where $\boldsymbol{\Delta R}$ is the relative rotation matrix and $\boldsymbol{\Delta t}$ is the relative translation vector. Based on $\boldsymbol{\Delta R}$ and $\boldsymbol{\Delta t}$, the translation distance $||\boldsymbol{\Delta t}||_2$ and the relative rotation angle $\Delta\theta$ between the current frame and the previous keyframe can be calculated by:

$$\begin{cases} \|\boldsymbol{\Delta t}\|_2 = \sqrt{\Delta t_x^2 + \Delta t_y^2 + \Delta t_z^2} \\ \Delta\theta = \arccos \dfrac{trace(\boldsymbol{\Delta R}) - 1}{2} \end{cases} \tag{9}$$

If the relative distance $||\boldsymbol{\Delta t}||_2$ is larger than the set minimum translation distance threshold ε_t or the relative rotation angle $\Delta\theta$ is larger than the set minimum rotation angle threshold ε_θ, the current frame can be selected as a new keyframe. Then, the new keyframe can be added to the factor graph model as a new node.

Once a new keyframe is added to the factor graph model, the relative pose constraint between the new and previous keyframe needs to be constructed. For two keyframe point clouds i and j, the relative pose transformation matrix T_i^j can be calculated by registering the spatial geometry features. The relative pose transformation matrix T_i^j satisfies the following equation:

$$\boldsymbol{T}_i^j = (\boldsymbol{T}_j^W)^{-1} \boldsymbol{T}_i^W \tag{10}$$

where $\boldsymbol{T}_i^W$ is the transformation matrix from the ith keyframe to the world coordinate system and $\boldsymbol{T}_j^W$ is the transformation matrix from the jth keyframe to the world coordinate system.

Due to the accumulated error caused by the environmental perception deviation during UAV flight, Eq. (10) will not be strictly valid. Then, the residual function $\boldsymbol{e}_L$ between the ith keyframe and jth keyframe can be denoted as:

$$\boldsymbol{e}_L(\boldsymbol{x}_i, \boldsymbol{x}_j) = ln((\boldsymbol{T}_i^j)^{-1} (\boldsymbol{T}_i^W)^{-1} \boldsymbol{T}_j^W) \tag{11}$$

In the above residual function, there are two pose variables to be optimized: the pose $\boldsymbol{x}_i$ of ith keyframe and the pose $\boldsymbol{x}_j$ of jth keyframe. Therefore, it is required to find the derivative of $\boldsymbol{e}_L(\boldsymbol{x}_i, \boldsymbol{x}_j)$ with respect to the variables $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. It is equivalent to finding the derivative of $\boldsymbol{e}_L(\boldsymbol{x}_i, \boldsymbol{x}_j)$ with respect to $\boldsymbol{T}_i^W$ and $\boldsymbol{T}_j^W$.

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

However, the transformation matrices $\boldsymbol{T}_i^W$ and $\boldsymbol{T}_j^W$ are not closed under addition and subtraction. Thus, Lie group and Lie algebra [47] are introduced to solve the deviation. The transformation matrix $\boldsymbol{T}$ is labeled as the special Euclidean group $SE(3)$ and its corresponding Lie algebra is $se(3)$. In Lie algebra, each pose is denoted by $\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix}$, where $\boldsymbol{\xi}$ is a six-dimensional vector. The first three dimensions $\boldsymbol{\rho}$ denote the translations and the last three dimensions $\boldsymbol{\phi}$ denote the rotations.

According to the transformation relation between Lie groups and Lie algebras, the Lie algebra form of the pose $\boldsymbol{x}_i$ of the ith keyframe is denoted as $\boldsymbol{\xi}_i$, and the Lie algebra form between ith and jth keyframe is denoted as $\boldsymbol{\xi}_{ij}$. By adding the left perturbation terms $\boldsymbol{\delta\xi}_i$ and $\boldsymbol{\delta\xi}_j$ to the poses $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$, the residual function $\boldsymbol{e}_L(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is changed to the following form:

$$\hat{\boldsymbol{e}}_L(\boldsymbol{x}_i, \boldsymbol{x}_j) = \ln\left(\left(\boldsymbol{T}_i^j\right)^{-1}\left(\boldsymbol{T}_i^W\right)^{-1}\exp\left(\left(-\boldsymbol{\delta\xi}_i\right)^{\wedge}\right)\exp\left(\left(\boldsymbol{\delta\xi}_j\right)^{\wedge}\right)\boldsymbol{T}_j^W\right)^{\vee} \tag{12}$$

Based on the derivation rule of Lie algebra, the derivatives of the residual function $\boldsymbol{e}_L(\boldsymbol{x}_i, \boldsymbol{x}_j)$ with respect to $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$ are equivalent to the derivatives of the residual function $\hat{\boldsymbol{e}}_L(\boldsymbol{x}_i, \boldsymbol{x}_j)$ with respect to the left perturbation terms $\boldsymbol{\delta\xi}_i$ and $\boldsymbol{\delta\xi}_j$:

$$\frac{\partial \boldsymbol{e}_L(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{\xi}_i} = \frac{\partial \hat{\boldsymbol{e}}_L(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{\delta\xi}_i} = -\boldsymbol{J}_r^{-1}\left(\boldsymbol{e}_L(\boldsymbol{x}_i, \boldsymbol{x}_j)\right) Ad\left(\left(\boldsymbol{T}_j^W\right)^{-1}\right) \tag{13}$$

$$\frac{\partial \boldsymbol{e}_L(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{\xi}_j} = \frac{\partial \hat{\boldsymbol{e}}_L(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{\delta\xi}_j} = \boldsymbol{J}_r^{-1}\left(\boldsymbol{e}_L(\boldsymbol{x}_i, \boldsymbol{x}_j)\right) Ad\left(\left(\boldsymbol{T}_j^W\right)^{-1}\right) \tag{14}$$

where $\boldsymbol{J}_r$ is the right multiplication of the Jacobi matrix and $Ad(\cdot)$ represents the adjacency matrix.

Furthermore, it is also necessary to estimate the covariance matrix $\boldsymbol{\Omega}_L$ of the relative pose factors. The inverse of the covariance matrix $(\boldsymbol{\Omega}_L)^{-1}$ is alternatively called the information matrix, which reflects the weight of residuals of each factor in the factor graph model. The covariance matrix $\boldsymbol{\Omega}_L$ between ith and jth keyframes can be constructed based on the uncertainty of the matched geometric feature points. Supposing that ${}^i\boldsymbol{p}_f$ and ${}^j\boldsymbol{p}_f$ are a pair of feature points between the ith and jth keyframes, the two feature points satisfy the following projection relationship:

$${}^j\boldsymbol{p}_f = \boldsymbol{R}_i^j\left({}^i\boldsymbol{p}_f - \boldsymbol{t}_j^i\right) \tag{15}$$

Based on the above feature points, the covariance matrix can be estimated by calculating the sum of all matched feature point pairs in the two keyframes:

$$\boldsymbol{\Omega}_L(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{m=1}^{N_f} \boldsymbol{H}^T \boldsymbol{\Lambda}_L^{-1} \boldsymbol{H} \tag{16}$$

where N_f is the total number of pairs. $\boldsymbol{\Lambda}_L$ is the zero-mean Gaussian matrix (the noise matrix of LiDAR). $\boldsymbol{H}$ is the Jacobi matrix, which is defined as:

$$\boldsymbol{H} = \left[\left({}^j\boldsymbol{p}_f\right)^{\wedge}\ \boldsymbol{R}_i^j\right] \tag{17}$$

The noise matrix $\boldsymbol{\Lambda}_L$ is defined as:

$$\boldsymbol{\Lambda}_L = \begin{bmatrix} 2\sigma_x^2 & 0 & 0 \\ 0 & 2\sigma_y^2 & 0 \\ 0 & 0 & 2\sigma_z^2 \end{bmatrix} \tag{18}$$

where σ_x, σ_y, and σ_z are the measurement noise along the X, Y, and Z axes of the LiDAR sensor with units of m. By setting the covariance matrix at the initial pose as the zero matrix $\boldsymbol{0}_{6\times6}$, the covariance matrix corresponding to the kth keyframe can be iterated according to the following equation as the LiDAR odometry is accumulated:

$$\boldsymbol{\Omega}_L^k \leftarrow \boldsymbol{\Omega}_L^{k-1} + \boldsymbol{H}_{k,k\text{-}1}{}^T \boldsymbol{\Lambda}_L^{-1} \boldsymbol{H}_{k,k\text{-}1} \tag{19}$$

where $\boldsymbol{H}_{k,k\text{-}1}$ corresponds to the Jacobi matrix between the kth keyframe and the $(k-1)$th keyframe.

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

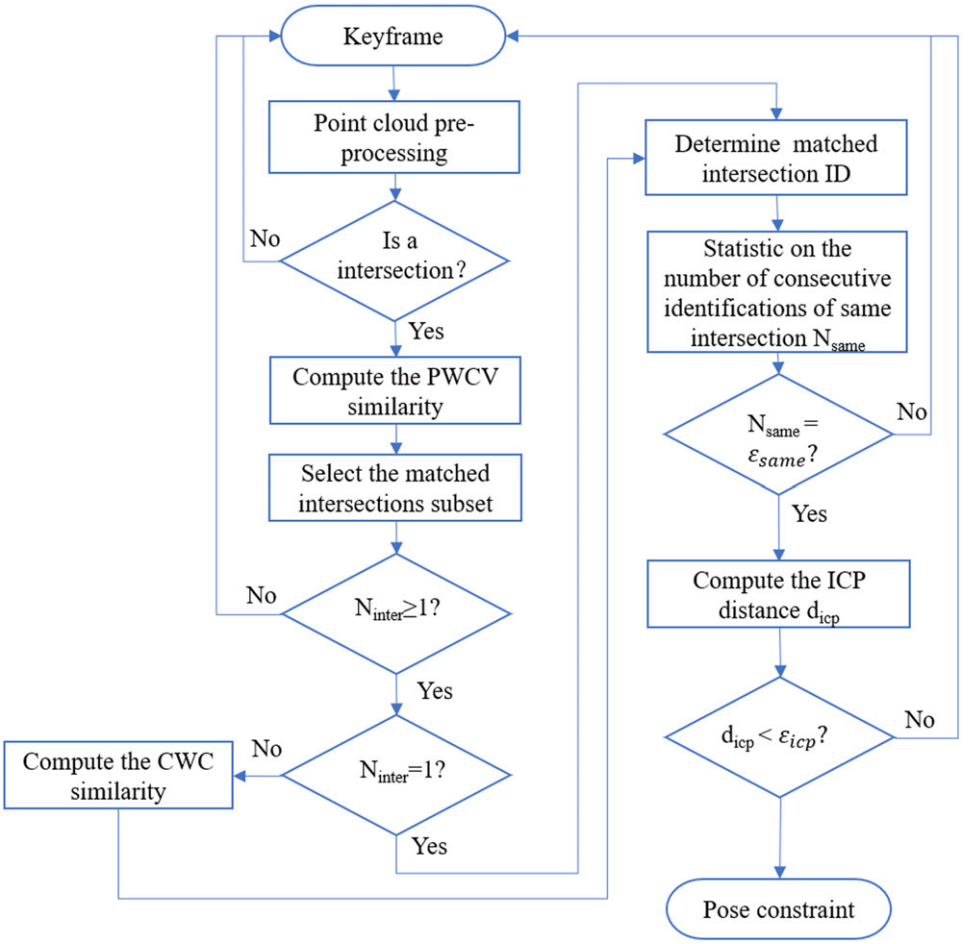


Figure 3. *The procedure for intersection location constraints constructing.*

3.2. Intersection pose constraint factor

Since most areas of underground roadways are long and narrow corridors with high similarity of geometric structures, the point cloud scanned by LiDAR in such degraded scenarios lacks adequate discriminative features, resulting in increasing accumulated positioning errors. Therefore, the intersection pose constraint factor is added to the factor graph model to provide reliable pose constraint for UAV pose optimization.

Based on our previous work proposed in [40], we first construct an intersection semantic knowledge database from the pre-built point cloud map of the underground mine. The constructed semantic knowledge database contains the geometric invariant point location $\boldsymbol{C}^{I_i}$, the dense point cloud $\boldsymbol{P}^{I_i}$, the PWCV descriptor $\boldsymbol{V}_d^{I_i}$, and CWC descriptor $\boldsymbol{M}_d^{I_i}$ of each intersection. While the UAV is inspecting the underground mine, the pose constraint constructing process between the current keyframe and the intersection in the semantic knowledge database is shown in Figure 3. The detailed steps are as follows.

1. The keyframe point cloud is preprocessed first. The key steps of preprocessing include point cloud filtering, mine ground plane detection, and wall point cloud segmentation.
2. Based on the segmented wall point cloud, the topology feature detection is performed to identify whether the current keyframe is an intersection or not. If the current keyframe is not an

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

intersection, it returns to the first step. If the current keyframe is an intersection, then the geometrical invariant point location of the intersection is computed.

3. Centering at the geometrical invariant point within radius R_{pwcv}, the intersection point cloud is decoded as a PWCV descriptor $\boldsymbol{V}_{\boldsymbol{curr}}$. Then, the similarity between the current PWCV descriptor $\boldsymbol{V}_{\boldsymbol{curr}}$ and the PWCV descriptor in the semantic knowledge database is calculated one by one.
4. According to the calculated similarity, the candidate intersections subset with high similarity is selected. By setting the minimum PWCV similarity threshold ε_{pwcv}, the intersections with similarity greater than ε_{pwcv} are selected to form a candidate intersection subset.
5. The matched intersection is determined by the number of the candidate intersection subset N_{inter}. If N_{inter} is equal to 0, the current keyframe is not a real intersection. The subsequent processes are stopped, and then return to the first step to wait for the processing of the next keyframe. If N_{inter} is equal to 1, it means the unique intersection in the candidate intersections subset is the matched intersection, and the process returns to step 8. If the number of candidate intersections N_{inter} is larger than 1, it means that the current keyframe is similar to multiple intersections of the semantic knowledge database. It is difficult to identify the matched intersection using only the PWCV descriptor similarity results. It is necessary to further compare the similarity of CWC descriptors between the current keyframe and the intersections in the semantic knowledge database.
6. The CWC descriptor of the current keyframe is generated to select the final matched intersections. The CWC is a cylinder wall contour descriptor, which is generated by adding the height feature encoding on the basis of the PWCV descriptor. The single-frame LiDAR point cloud is sparse and contains less spatial information. Therefore, before generating the CWC descriptor of the current keyframe, the nearest neighboring N_{his} keyframes are transformed to the current keyframe for staking dense point cloud to obtain richer spatial features. The dense point cloud is stacked by:

$$\boldsymbol{P}_{\boldsymbol{dense}} = \boldsymbol{P}_{\boldsymbol{curr}} + \sum_{i=1}^{N_{his}} \boldsymbol{T}_{\boldsymbol{his_i}}^{\boldsymbol{curr}} \boldsymbol{P}_{\boldsymbol{his_i}} \tag{20}$$

where $\boldsymbol{P}_{\boldsymbol{dense}}$ is the dense point cloud after stacking. $\boldsymbol{P}_{\boldsymbol{curr}}$ is the point cloud of current keyframe. $\boldsymbol{P}_{\boldsymbol{his_i}}$ is the ith nearest-neighbor keyframe point cloud. $\boldsymbol{T}_{\boldsymbol{his_i}}^{\boldsymbol{curr}}$ is the transformation matrix from the current keyframe to the ith nearest-neighbor keyframe. Based on the stacked dense point cloud $\boldsymbol{P}_{\boldsymbol{dense}}$, the CWC descriptor $\boldsymbol{M}_{\boldsymbol{curr}}$ is generated. Meanwhile, its similarity with the CWC descriptors of intersections in the candidate subset is calculated one by one. The intersection with the highest similarity is selected as the final matched intersection.
7. To avoid false intersection recognition, the number of consecutive recognitions of the same intersection is counted. When the UAV flies through an intersection, this intersection should be recognized by multiple keyframes. At the beginning, the number of keyframes N_{same} continuously recognized for the same intersection is set to 0. If the matched intersection ID of the current keyframe is the same as the matched intersection ID of the latest keyframe, N_{same} is increased by 1. If the matched intersection ID of the current keyframe is different from the matched intersection ID of the latest keyframe, N_{same} is initialized to 0 and the subsequent steps are terminated. The process then returns to the first step and waits for the processing of the next keyframe.
8. Judge whether the same intersection is stably detected based on the number of consecutive recognition N_{same}. When N_{same} is smaller than the set minimum number of consecutive recognition ε_{same}, the subsequent step is terminated until N_{same} is equal to ε_{same}. Then, it proceeds to the next step, and N_{same} is initialized to 0 simultaneously.
9. The ICP distance d_{icp} between the current keyframe and the matched intersection is used for distance verification. If the ICP distance d_{icp} is larger than the minimum distance threshold ε_{icp}, it is considered a wrong match and the intersection pose constraint will not be added to the fact graph model. If the ICP distance d_{icp} is smaller than ε_{icp}, it is considered as a correct match. Then,

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

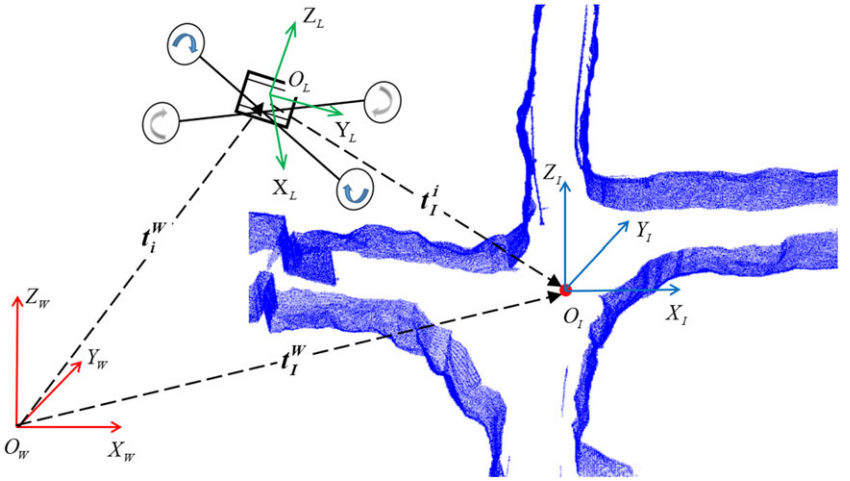


Figure 4. *The 3D location observation of intersection.*

the intersection pose constraint between the current keyframe and the matched intersection in the semantic knowledge database is added to the factor graph model for UAV pose optimization.

According to the above procedures, the intersection pose constraint can be established. When the UAV detects a stable intersection at pose $\boldsymbol{x}_i$, the spatial pose relationship among the UAV pose (the LiDAR coordinate system $O_L X_L Y_L Z_L$), the world coordinate system $O_W X_W Y_W Z_W$, and the intersection coordinate system $O_I X_I Y_I Z_I$ is shown in Figure 4.

As Figure 4 shown, $\boldsymbol{t}_I^W$ displayed as the orange dashed line is the 3D vector of the observed intersection in the world coordinate system at pose $\boldsymbol{x}_i$. $\boldsymbol{T}_i^W$ and $\boldsymbol{t}_I^i$ are displayed as the purple dashed line. $\boldsymbol{T}_i^W$ represents the 3D vector of the LiDAR sensor, and $\boldsymbol{t}_I^i$ represents the 3D vector of the observed intersection in the current LiDAR coordinate system. Therefore, the observation $\boldsymbol{z}_I^W$ of the intersection in the world coordinate system can be denoted as:

$$^W\boldsymbol{z}_I = \boldsymbol{t}_I^W = \boldsymbol{T}_i^W + \boldsymbol{R}_i^W \boldsymbol{t}_I^i \tag{21}$$

where $\boldsymbol{R}_i^W$ is the rotation matrix of the LiDAR coordinate system to the world coordinate system.

The error between the intersection observation $^W\boldsymbol{z}_I$ at pose $\boldsymbol{x}_i$ and the prior intersection location provided by the intersection semantic knowledge database $\hat{t}_I^W$ constitutes the residual function $\boldsymbol{e}_I\left(\boldsymbol{z}_I^W, \boldsymbol{\chi}_i\right)$ of the current intersection pose constraint factor:

$$\boldsymbol{e}_I\left(\boldsymbol{z}_I^W, \boldsymbol{\chi}_i\right) = \hat{t}_I^W - \boldsymbol{t}_i^W - \boldsymbol{R}_i^W \boldsymbol{t}_I^i \tag{22}$$

where the variables to be optimized are as follows:

$$\boldsymbol{\chi}_i = \{\boldsymbol{R}_i^W, \boldsymbol{T}_i^W, \boldsymbol{t}_I^i\} \tag{23}$$

Similarly, since the rotation matrix $\boldsymbol{R}_i^W$ is not closed under the addition and subtraction, the derivative of the residue function $\boldsymbol{e}_I\left(\boldsymbol{z}_I^W, \boldsymbol{\chi}_i\right)$ with respect to the $\boldsymbol{R}_i^W$ cannot be computed directly. It is required to convert the rotation matrix $\boldsymbol{R}_i^W$ into Lie algebra form for derivation. The rotation matrix $\boldsymbol{R}$ is known as the special orthogonal group SO(3), and the Lie algebra form of SO(3) is $\boldsymbol{\phi}$.

Thus, the Lie algebra form of rotation matrix $\boldsymbol{R}_{Li}^W$ in the residual function $\boldsymbol{e}_I\left(\boldsymbol{z}_I^W, \boldsymbol{\chi}_i\right)$ (Eq. (22)) is defined as $\boldsymbol{\phi}_i$. Then, the derivative of the residue function $\boldsymbol{e}_I\left(\boldsymbol{z}_I^W, \boldsymbol{\chi}_i\right)$ with respect to the $\boldsymbol{R}_i^W$ can be denoted as:

$$\frac{\partial \boldsymbol{e}_I(\boldsymbol{z}_I^W, \boldsymbol{\chi}_i)}{\partial \boldsymbol{\phi}_i} = \frac{\partial(-\boldsymbol{R}_i^W \boldsymbol{t}_I^i)}{\partial \boldsymbol{\phi}_i} = \frac{\partial(-\exp(\boldsymbol{\phi}_i^\wedge)\boldsymbol{t}_I^i)}{\partial \boldsymbol{\phi}_i} \tag{24}$$

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

By introducing the left perturbation terms $\boldsymbol{\delta\phi}_i$, the derivative of the residual function $\boldsymbol{e}_I\left(z_I^W, \boldsymbol{x}_i\right)$ with respect to the rotation $\boldsymbol{\phi}_i$ is equal to:

$$\frac{\partial \boldsymbol{e}_I(z_I^W, \boldsymbol{\chi}_i)}{\partial \boldsymbol{\phi}_i} = \lim_{\boldsymbol{\delta\phi}_i \to 0} \frac{-\exp\left(\left(\boldsymbol{\delta\phi}_i\right)^\wedge\right) \exp\left(\boldsymbol{\phi}_i^\wedge\right) \boldsymbol{t}_I^i - \exp\left(\boldsymbol{\phi}_i^\wedge\right) \boldsymbol{t}_I^i}{\boldsymbol{\delta\phi}_i} \tag{25}$$

Expanding $\exp\left(\left(\boldsymbol{\delta\phi}_i\right)^\wedge\right)$ in Eq. (25) with Taylor function, the derivation $\frac{\partial \boldsymbol{e}_I(z_I^W, \boldsymbol{\chi}_i)}{\partial \boldsymbol{\phi}_i}$ can be simplified as:

$$\begin{aligned}\frac{\partial \boldsymbol{e}_I(z_I^W, \boldsymbol{\chi}_i)}{\partial \boldsymbol{\delta\phi}_i} &= \lim_{\boldsymbol{\delta\phi}_i \to 0} \frac{-\left(\boldsymbol{E}_3 + (\boldsymbol{\delta\phi}_i)^\wedge\right) \exp\left(\boldsymbol{\phi}_i^\wedge\right) \boldsymbol{t}_I^i - \exp\left(\boldsymbol{\phi}_i^\wedge\right) \boldsymbol{t}_I^i}{\boldsymbol{\delta\phi}_i} \\ &= \lim_{\boldsymbol{\delta\phi}_i \to 0} \frac{-\left(\boldsymbol{\delta\phi}_i\right)^\wedge \exp\left(\boldsymbol{\phi}_i^\wedge\right) \boldsymbol{t}_I^i}{\boldsymbol{\delta\phi}_i} \\ &= \lim_{\boldsymbol{\delta\phi}_i \to 0} \frac{\boldsymbol{\delta\phi}_i \left(\exp\left(\boldsymbol{\phi}_i^\wedge\right) \boldsymbol{t}_I^i\right)^\wedge}{\boldsymbol{\delta\phi}_i} \\ &= \left(\boldsymbol{R}_i^W \boldsymbol{t}_I^i\right)^\wedge\end{aligned} \tag{26}$$

In addition, according to Eq. (22), the derivatives of the residual function $\partial \boldsymbol{e}_I(z_I^W, \boldsymbol{\chi}_i)$ with respect to the optimization variables $\boldsymbol{T}_i^W$ and $\boldsymbol{t}_I^i$ are as follows:

$$\frac{\partial \boldsymbol{e}_I(z_I^W, \boldsymbol{\chi}_i)}{\partial \boldsymbol{t}_i^W} = -\boldsymbol{E}_3 \tag{27}$$

$$\frac{\partial \boldsymbol{e}_I(z_I^W, \boldsymbol{\chi}_i)}{\partial \boldsymbol{t}_I^i} = -\boldsymbol{R}_i^W \tag{28}$$

To construct the intersection pose constraint factor, it is also important to analyze the uncertainty of the intersection observation by estimating the covariance matrix $\boldsymbol{\Omega}_I$. Based on the registered points correspondence between the current keyframe and the matched intersection in the semantic knowledge database, the covariance matrix $\boldsymbol{\Omega}_I$ can be calculated. Assuming ${}^W\boldsymbol{p}_{I_f}$ and ${}^i\boldsymbol{p}_{I_f}$ are a pair of registered feature points in the semantic knowledge database and the ith keyframe, they will obey the following projection function:

$${}^i\boldsymbol{p}_{I_f} = \boldsymbol{R}_W^i \left({}^W\boldsymbol{p}_{I_f} - \boldsymbol{t}_i^W\right) \tag{29}$$

Then, the covariance matrix $\boldsymbol{\Omega}_I$ can be estimated by calculating the sum of the distance among all matched feature points between the ith keyframe and intersection in the semantic knowledge database:

$$\boldsymbol{\Omega}_I\left({}^W\boldsymbol{z}_I, \boldsymbol{\chi}_i\right) = \sum_{m=1}^{N_{I_f}} \boldsymbol{Q}^T \boldsymbol{\Lambda}_I^{-1} \boldsymbol{Q} \tag{30}$$

where N_{I_f} is the total number of matched point pairs. $\boldsymbol{\Lambda}_I$ is the zero-mean Gaussian noise matrix of matched intersections. Based on Eq. 29, the Jocabi matrix $\boldsymbol{Q}$ is defined as:

$$\boldsymbol{Q} = \left[\left({}^W\boldsymbol{p}_{I_f}\right)^\wedge \; -\boldsymbol{R}_W^i\right] \tag{31}$$

Based on the PWCV descriptor generation process proposed in [40], the nonzero portion of the wall contour component of the final formed PWCV descriptor is tightly correlated with the relative pose between the current UAV and the detected intersection geometric invariant point. Therefore, the completeness of the wall contour component of the PWCV descriptor is proposed to estimate the noise matrix $\boldsymbol{\Lambda}_I$. Figure 5(a) shows the PWCV descriptor of an example intersection in the semantic database, and its nonzero eigenvalue dimensions of the wall contour component is k_b. Figure 5(b) is the PWCV

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

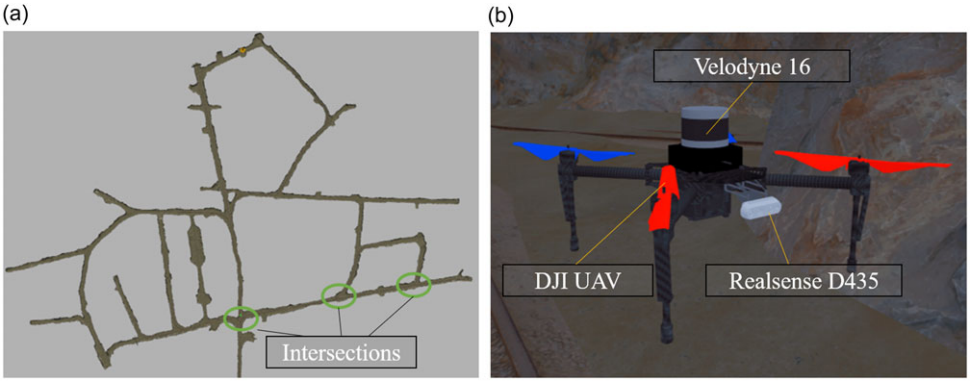


Figure 5. *UAV flight localization experiment in the Edgar Mine environment. (a) The Edgar Mine deployed in the ROS Gazebo. (b) The UAV simulation platform in the ROS Gazebo.*

descriptor of the detected intersection during UAV flight, and its nonzero eigenvalue dimensions of the wall contour component is k_c. The noise matrix $\mathbf{\Lambda_I}$ is defined as:

$$\mathbf{\Lambda_I} = \begin{bmatrix} \frac{k_c}{k_b} & 0 & 0 \\ 0 & \frac{k_c}{k_b} & 0 \\ 0 & 0 & \frac{k_c}{k_b} \end{bmatrix} \tag{32}$$

Thus, the relative pose constraint factor residual function (Eq. (11)) and its covariance matrix (Eq. (16)), and the intersection pose constraint factor residual function (Eq. (22)) and its covariance matrix (Eq. (30)) have been established. By substituting the residual functions and their covariance matrices to the UAV pose factor graph model (Eq. (7)), the UAV pose can be optimized in nonlinear iterations based on the results of deviations (Eq. (13), Eq. (14), Eq. (26), and Eq. (27)).

4. Experiments and results

In this section, two UAV localization tests are conducted in the large-scale Edgar Mine and a mine-like indoor corridor environment. In the two experiments, we compared the localization accuracy with two typical LiDAR-based localization methods, that is, LOAM and Scan Context. LOAM is an open-loop laser odometry localization method. LOAM + Scan Context adds the Scan Context global point cloud descriptor to LOAM for scene recognition. The loop constraints based on Scan Context are added to correct the cumulative localization error during UAV flight after recognizing the same scene.

4.1. UAV localization in Edgar Mine

The first UAV localization experiment is conducted in the simulated Edgar Mine. The simulated Edgar Mine is developed by importing the Edgar Mine environment model file from DARPA [48] on the ROS Gazebo platform. As shown in Figure 5 (a), the total length of Edgar Mine is 2.8 km and it consists of a number of narrow roadways and several intersections. The DJI UAV platform equipped with Velodyne 16 LiDAR and Realsense D435 camera is added to the ROS Gazebo for mine exploration, as shown in Figure 5 (b).

First, the UAV is controlled manually to fly slowly along the Edgar Mine tunnel for one cycle to record the LiDAR data and RGB-D image data. The Edgar Mine point cloud map is reconstructed based on our previous work [39] and the obtained sensor data. Furthermore, the semantic topology information

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

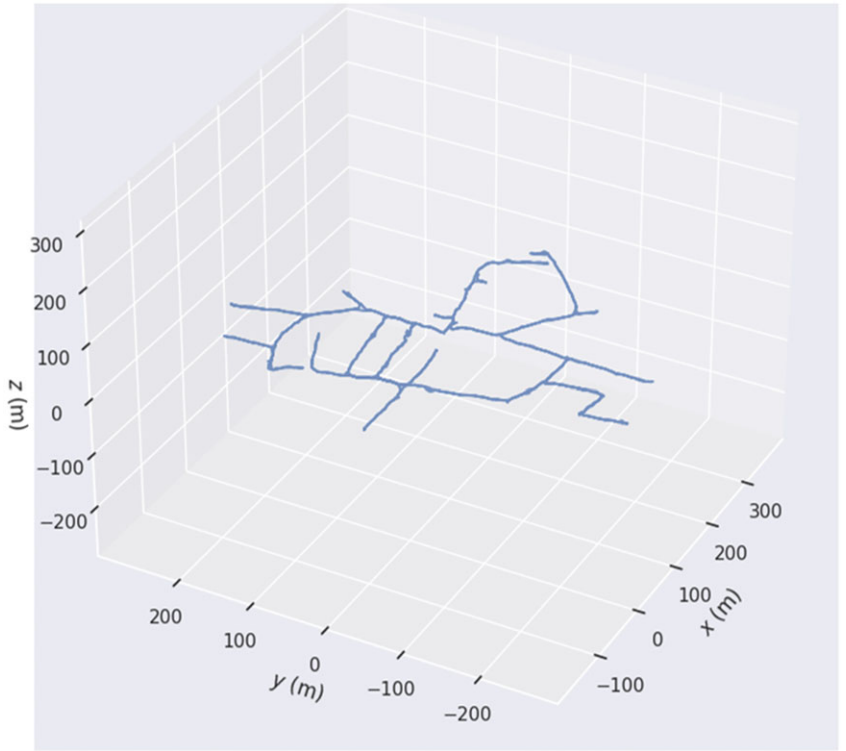


Figure 6. *The UAV flight trajectory.*

is segmented from the pre-built map to construct a semantic knowledge database for the Edgar Mine. Finally, a long-distance flight trajectory is recorded for UAV flight localization accuracy comparison. The trajectory is shown in Figure 6. The total length of this trajectory is 5.68 km. The LiDAR point cloud data and the ground truth pose are recorded during the UAV flight. In particular, the ground truth pose is obtained by a high-precision IMU sensor without zero drift.

4.1.1. Semantic knowledge database construction

With reference to the point cloud fusion framework [39], the LiDAR data and depth image are fused based on the Bayesian Kriging model to reconstruct a single-frame high-precision dense point cloud of the mine roadway. Furthermore, the ISS3D key points and FPFH descriptors are extracted from the reconstructed single-frame dense point cloud for spatial feature coarse matching and ICP registration. The reconstructed point cloud map of Edgar Mine is shown in Figure 7.

The roadway of Edgar Mine is complicated and contains several loops and intersections (the ID of each intersection is shown in the yellow box in Figure 7). To construct the intersection semantic knowledge database, the intersection dense point cloud is first segmented from the reconstructed map. Then, the intersection type, geometric invariant points location, the PWCV descriptors, and CWC descriptors are generated for each intersection based on our previous work [40]. The constructed intersection semantic knowledge database is shown in Figure 8.

4.1.2. UAV localization accuracy analysis

Figure 9 shows the estimated pose trajectory of UAV based on competing methods and the proposed localization method. Comparing the trajectories plotted in Figure 9, it can be found that the trajectory estimated by LOAM gradually deviates from the ground truth trajectory. This is because the LOAM

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

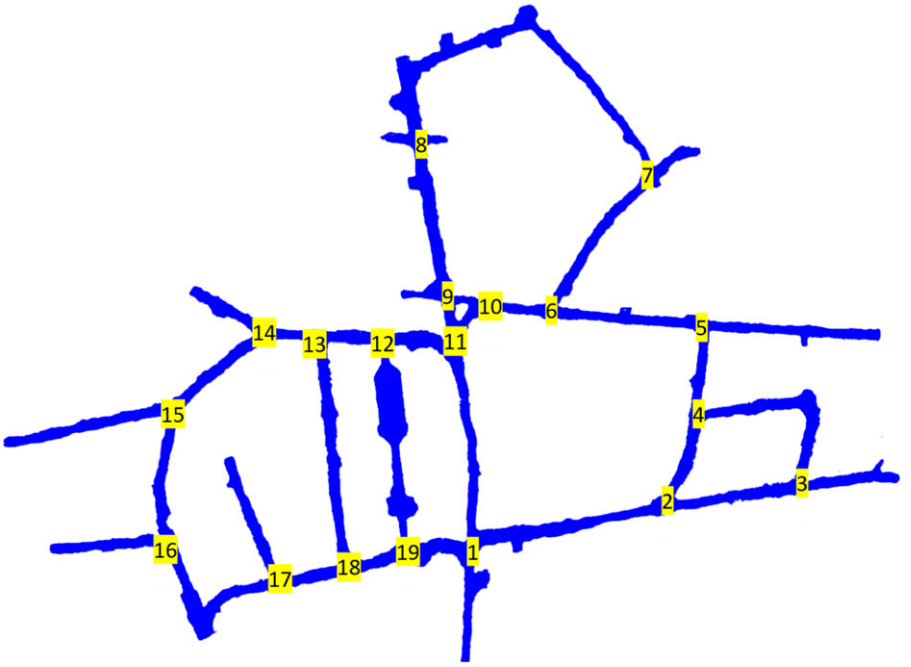


Figure 7. *The reconstructed environmental map of Edgar Mine.*

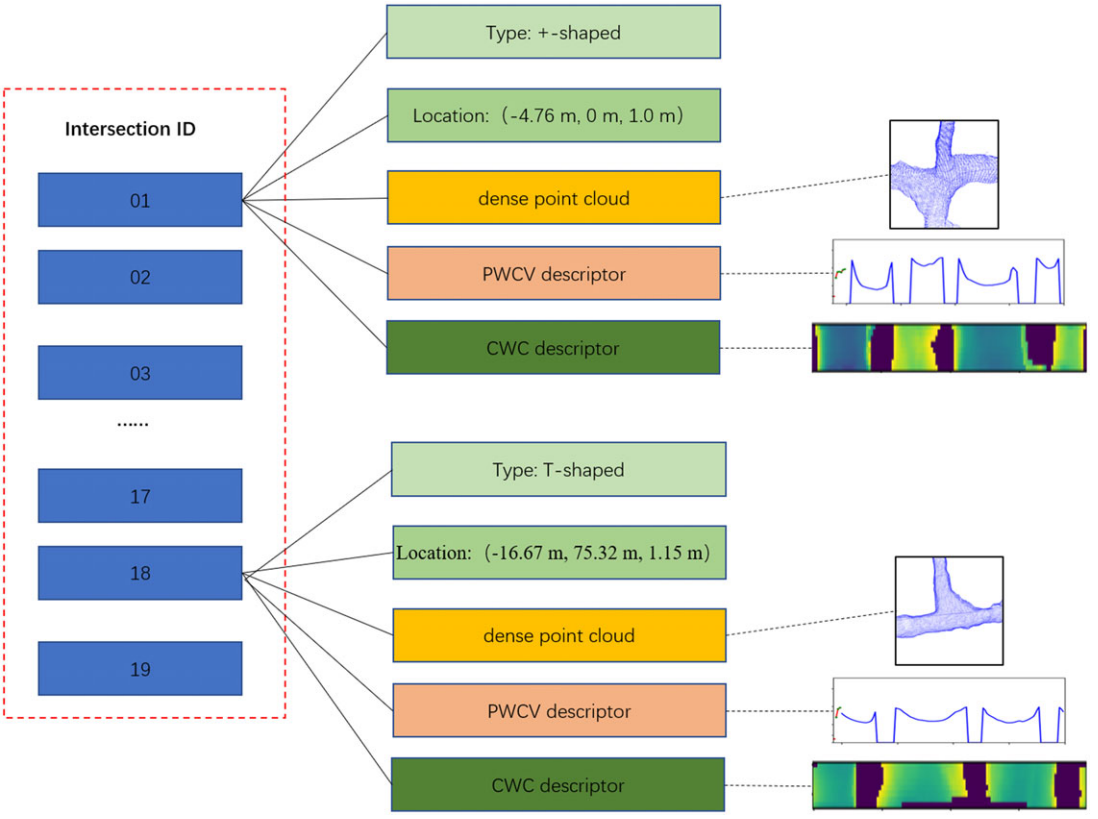


Figure 8. *The intersection semantic knowledge database of the Edgar Mine.*

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

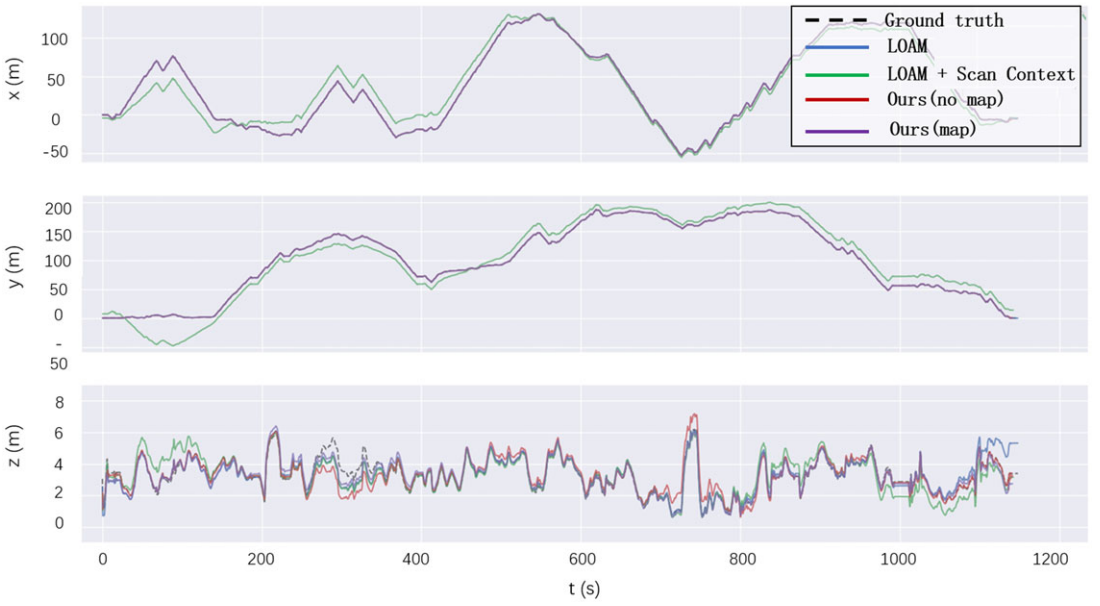


Figure 9. *The estimated trajectories during UAV flight based on different methods in Edgar Mine.*

localization algorithm only relies on the neighboring point cloud registration for pose estimation, which results in error accumulation during long-distance flights in underground mines with similar geometry. LOAM + Scan Context shows a large localization deviation. This is due to the fact that the similar mine roadways cause Scan Context to recognize the scene incorrectly. In contrast, the localization method proposed in this paper can accurately recognize different intersections without false scene recognition. Therefore, in the case where the prior map is not pre-built, the accumulated error at the same intersection can be eliminated, resulting in a reduction of the whole trajectory error. Furthermore, by establishing the semantic knowledge database, the intersection pose constraint between the current frame and the intersection in the semantic knowledge database can be added. Once a stable intersection is detected during flight, the accumulated localization error between the last detected intersection and the current intersection can be eliminated immediately.

To quantitatively analyze the localization error of each method [49], the maximum error (MAE), root mean square error (RMSE), and the relative error percentage (REP) are used to evaluate the localization accuracy. REP-1, REP-2, REP-3, REP-4, and REP-5 stand for 20%, 40%, 60%, 80%, and 100% of the trajectory. The localization errors are listed in Table I.

As listed in Table I, the MAE, RMSE, and REP-5 of LOAM are 39.15 m, 10.15 m, and 0.51 %, respectively. The MAE, RMSE, and REP-5 of LOAM + Scan Context are 6.62 m, 2.63 m, and 0.22 %, which cannot completely eliminate the accumulated error of LOAM. In comparison, the MAE, RMSE, and REP-5 of the proposed method are 2.22 m, 1.22 m, and 0.17 %. By adding the intersection pose constraint based on the semantic knowledge database with the pre-built map, the MAE, RMSE, and REP-5 of the proposed method are 2.06 m, 0.60 m, and 0.13 %, which shows over three times performance improvement compared to LOAM. It can be concluded that the proposed method can achieve accurate localization by optimizing the accumulative error of open-loop LOAM.

To further analyze the relationship between the distribution of localization error and the flight trajectory, the localization error maps of the comparison methods and the proposed method are plotted in Figure 10.

It can be seen from Figure 10(a) that the localization error of LOAM continuously increases with the increase in the UAV flight distance. The error distribution map in Figure 10(b) shows that the LOAM + Scan Context can detect a loop in the same scene with a small pose change. However, the

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

Table I. *UAV flight trajectory errors analysis in Edgar Mine environment.*

Method	MAE (m)	RMSE (m)	REP −1(%)	REP −2(%)	REP −3(%)	REP −4(%)	REP −5(%)
LOAM	39.15	10.15	2.18	1.32	0.84	1.04	0.51
LOAM + Scan Context	6.62	2.63	1.09	0.52	0.32	0.32	0.22
Ours(no map)	**2.22**	**1.22**	1.26	0.67	0.19	0.28	**0.17**
Ours(map)	**2.06**	**0.60**	1.38	0.66	0.22	0.31	**0.13**

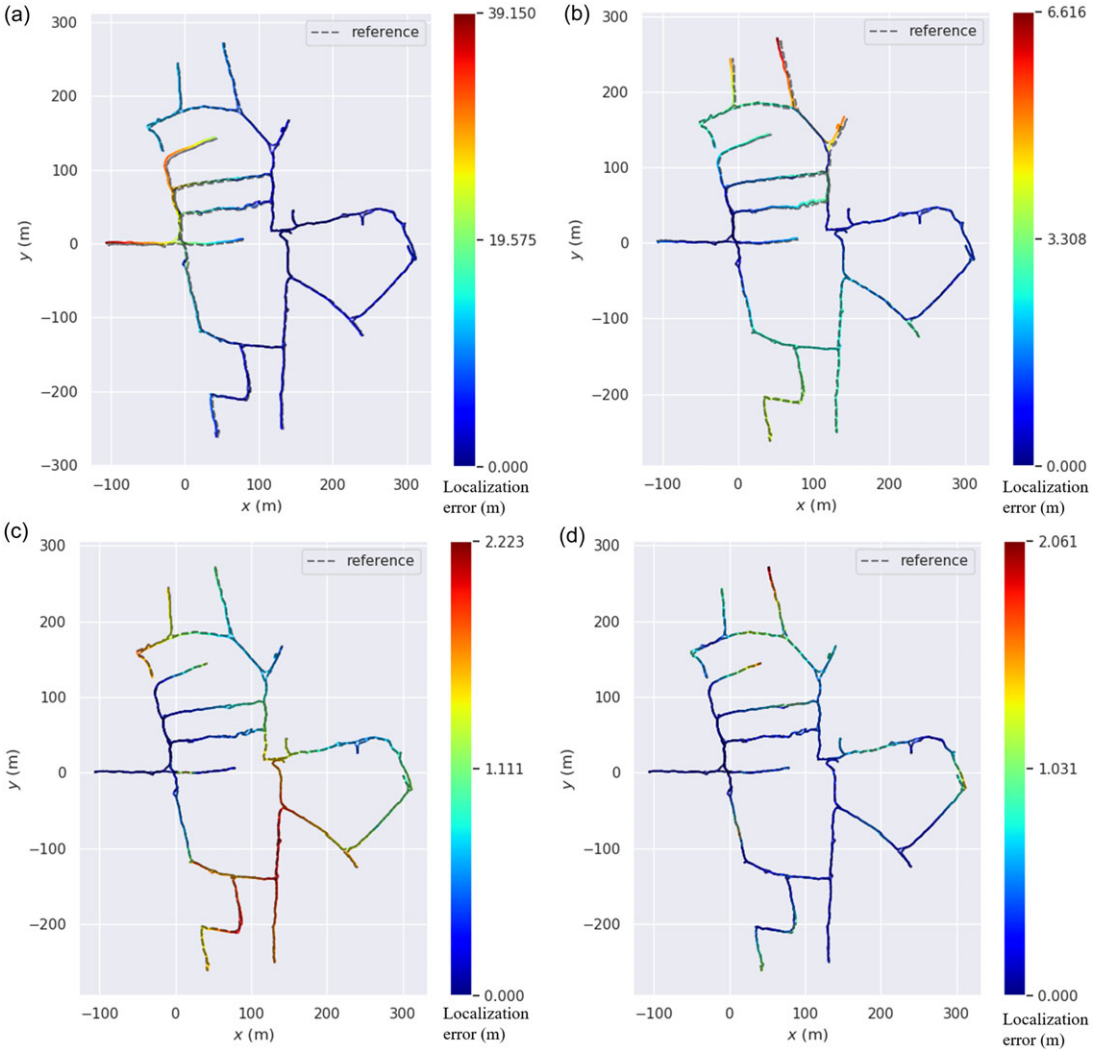


Figure 10. *The localization error maps with different methods. (a) The error distribution of LOAM. (b) The error distribution of LOAM + Scan context. (c) The error distribution of ours(no map). (d) The error distribution of ours (map).*

localization error still accumulates with the fight distance, resulting in a maximum error of up to 6.6 m. As shown in Figure 10(c), based on detecting the same intersection from different directions without a pre-built map, the proposed method can establish stable loop constraints. Then, the maximum error is reduced to 2.2 m. Figure 10(d) shows the error distribution map of the proposed method based on

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press



Figure 11. *The UAV hardware platform.*

the semantic knowledge database. When an intersection in the semantic knowledge database is detected during UAV flight, the localization error is eliminated immediately. The localization error only remains between two intersections.

4.2. UAV localization in mine-like indoor corridor

To evaluate the localization accuracy and adaptability of the proposed method in real environments, a UAV hardware platform same as the simulated UAV platform is designed and applied to conduct localization experiments in the mine-like indoor corridor. The components of the UAV hardware platform are shown in Figure 11.

Since the ground truth pose of UAV cannot be directly measured in a narrow indoor corridor, a localization accuracy evaluation method based on multiple Apriltag [50] is applied to compare the localization error of different methods. As shown in Figure 12(a), the first experiment was conducted in an indoor corridor. The total length of the indoor corridor is 210 m and its minimum width is 1.3 m. As shown in Figure 12(b), 20 Apriltags were deployed for positioning accuracy evaluation. The position of each Apriltag is listed in Table II. Figure 12(c) shows an example of Apriltag pasted on the wall. To ensure the safety of the localization experiment, the UAV was mounted on a mobile tripod, making it easy to adjust the pose change during flight.

4.2.1. Semantic knowledge database construction

Firstly, the indoor corridor prior map can be reconstructed with the point cloud data scanned by the Velodyne 16 and depth images measured by the Realsense D435, based on our previous work [39]. The reconstructed point cloud map of the indoor corridor is shown in Figure 13.

It can be seen from Figure 13 that the indoor corridor contains five intersections, which are numbered 1 to 5. The dense point cloud of each intersection is segmented from the map for constructing the semantic knowledge database, where the segmented point cloud is shown in Figure 14. It can be found from Figure 14 that the geometric structure of intersections 1, 2, and 5 are similar, resulting in the difficulty of distinguishing them only relying on the point cloud data.

Based on our previous work [40], the geometric invariant point of each intersection is detected for generating PWCV and CWC descriptors, which are shown in Figure 15. Correspondingly, the PWCV and CWC similarity matrices are computed for recognizing different intersections. The results are shown

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

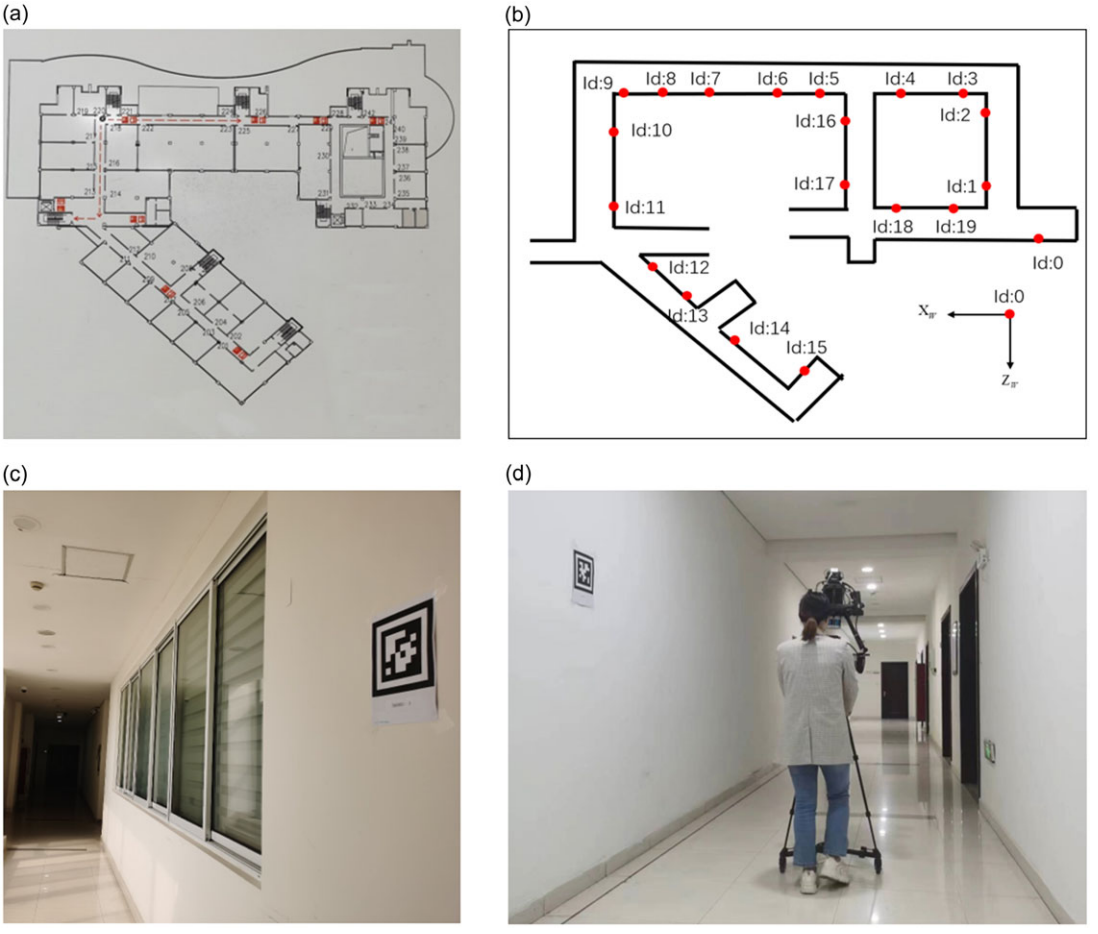


Figure 12. *The experiment of UAV localization in the indoor corridor environment. (a) The 2D map of the indoor corridor. (b) The distribution of apriltag in indoor corridor. (c) An example of Apriltag pasted on the wall of the indoor corridor. (d) The process of data recording.*

in Figure 16. It can be seen that the PWCV and CWC similarities of different intersections are less than 0.8. In the UAV localization experiment, the minimum similarity threshold is set as 0.8 so that the different intersections can be recognized correctly. Based on the segmented intersection point cloud, detected invariant point, and the descriptor, the semantic knowledge database of the indoor corridor is constructed, where the structure is the same as Figure 8.

4.2.2. UAV localization accuracy analysis

The UAV platform was driven in the indoor corridor for data recording, and the UAV pose was changed by adjusting the tripod to simulate the flight process. The driving trajectory is shown as the red line of Figure 17. The trajectory passed sequentially through the Apriltags numbered 0-1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-14-13-12-11-10-9-8-7-6-5-16-17-18-19-0, and the total length of the trajectory is 420 m.

Based on the recorded LiDAR data, the UAV pose is estimated by LOAM, LOAM + Scan Context, and the proposed method. The localization errors are computed by calculating the difference between the

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

Table II. *The 3D location of Apriltag QR codes on the wall surfaces.*

Tag ID	X_W (m)	Y_W (m)	Z_W (m)	Tag ID	X_W (m)	Y_W (m)	Z_W (m)
0	0.00	0.00	0.00	10	74.203	0.075	−18.443
1	4.177	0.107	−5.920	11	74.00	0.070	−5.193
2	4.177	0.035	−22.196	12	73.910	0.240	7.312
3	5.118	0.079	−24.418	13	58.085	0.105	23.137
4	14.365	0.105	−24.418	14	53.043	0.220	28.179
5	25.270	0.070	−24.498	15	41.701	0.262	39.521
6	34.235	−0.011	−24.103	16	17.435	0.135	−23.155
7	44.420	−0.060	−24.103	17	17.435	0.059	−5.63
8	66.020	0.145	24.103	18	13.695	0.150	−3.180
9	73.990	0.177	−24.103	19	5.255	0.340	−3.180

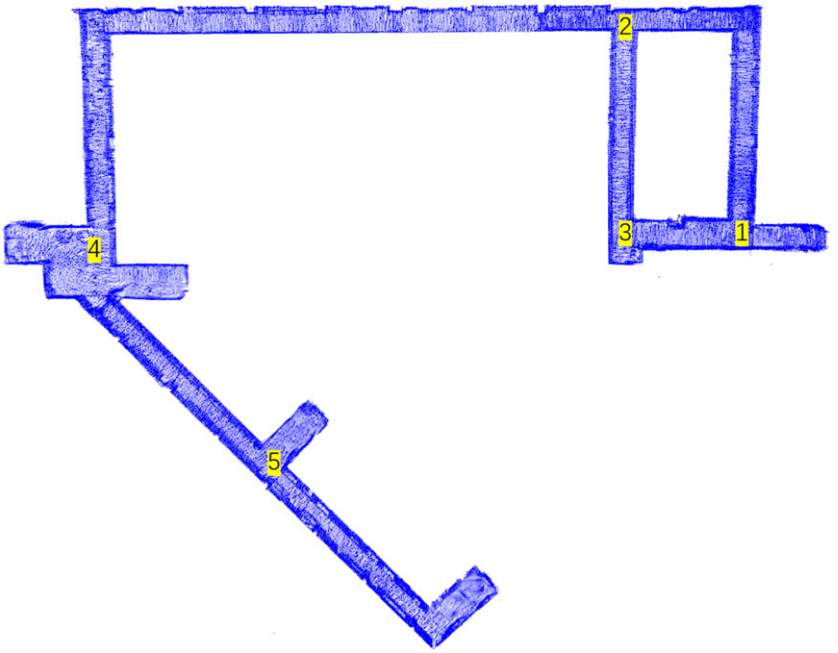


Figure 13. *The reconstructed point cloud map of the indoor corridor environment.*

estimated locations of Apriltags and the ground truth locations of Apriltags, which are listed in Table II. The localization error curves of each method are plotted in Figure 18. The horizontal coordinate represents the IDs of the 31 Apriltags passed by the UAV in sequence, and the vertical coordinate represents the calculated localization error.

As shown in Figure 18, the localization error of LOAM is increasing continuously with the increase of driving distance, where the average error is 14.18 m and the maximum error is 51.8 m. This is due to the similar geometric structure of the indoor corridor. At the same time, the similar geometric structure also results in mismatches of LOAM + Scan Context, with an average error of 20.9 m, which is higher than that of the open-loop LOAM positioning method. As the red curve shows in Figure 18, the proposed localization method can decrease the average localization error to 5.0 m without a prior map. The proposed localization method (no map) can accurately recognize the same intersection from different

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

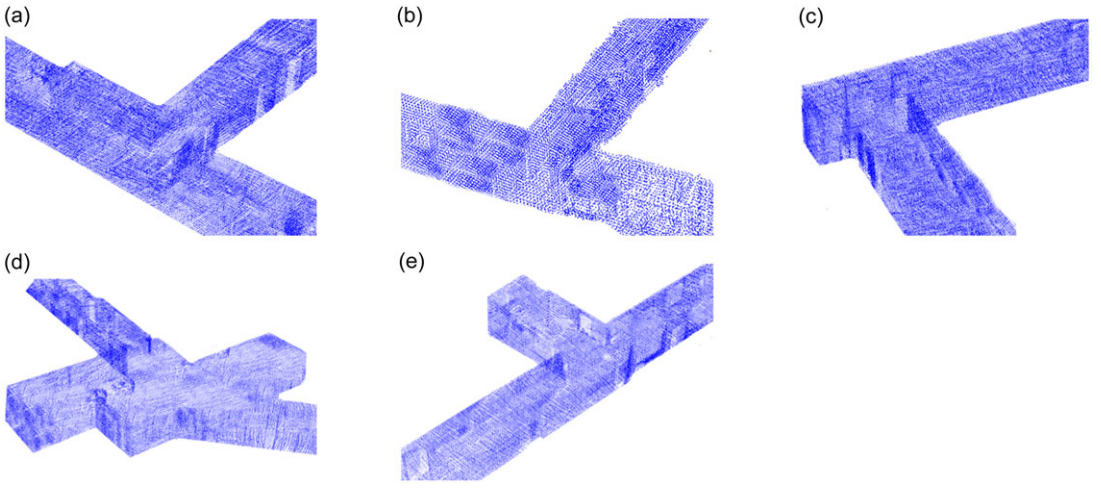


Figure 14. *The dense point clouds of each intersection in the indoor corridor environment. (a)– (e) The point cloud of intersection 1 to 5, respectively.*

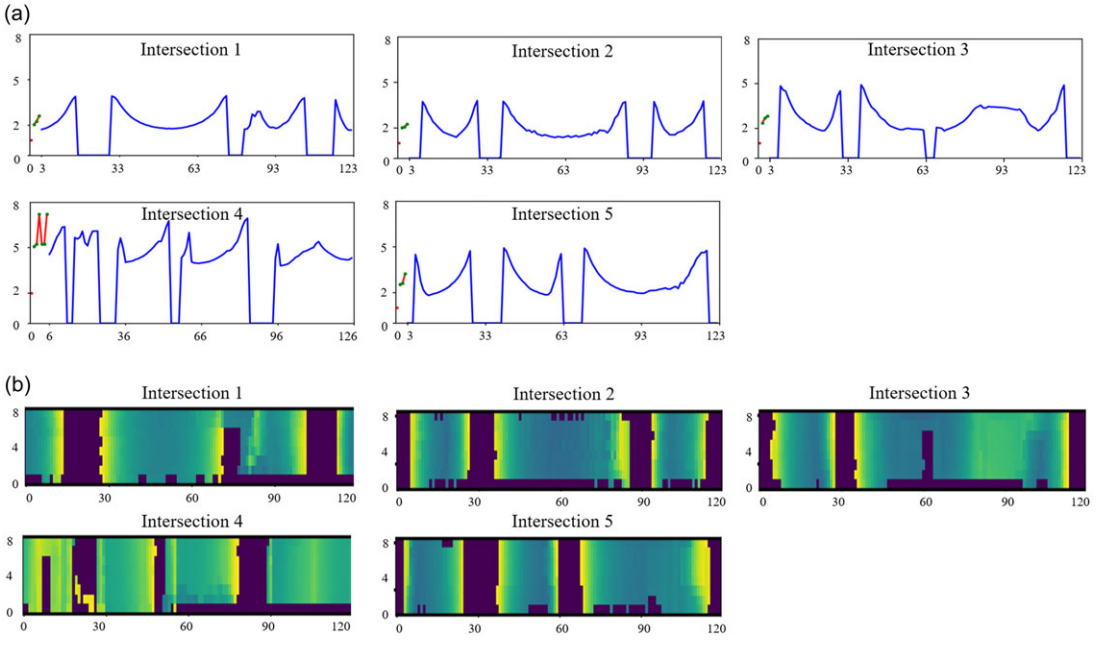


Figure 15. *The generated descriptors of each intersection. (a) The PWCV descriptors. (b) The CWC descriptors.*

directions, thereby constructing stable loop constraints. Furthermore, after adding the intersection constraint factor provided by the prior map, the proposed method can eliminate the accumulated localization error from the previous intersection to the current intersection when passing through the intersection in the semantic knowledge database. Thus, the localization error of the proposed method (with map) is optimized in segments, which greatly improves the accuracy of pose estimation and reduces the average positioning error to 1.7 m.

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

(a)

	1	2	3	4	5
1	1.00	0.75	0.07	0.16	0.51
2	0.75	1.00	0.15	0.09	0.65
3	0.07	0.15	1.00	0.05	0.75
4	0.16	0.09	0.05	1.00	0.13
5	0.51	0.65	0.75	0.13	1.00

(b)

	1	2	3	4	5
1	1.00	0.69	0.14	0.14	0.43
2	0.69	1.00	0.05	0.07	0.63
3	0.14	0.05	1.00	0.03	0.74
4	0.14	0.07	0.03	1.00	0.12
5	0.43	0.63	0.74	0.12	1.00

Figure 16. *The similarities of PWCV descriptors and CWC descriptors in the indoor corridor.*

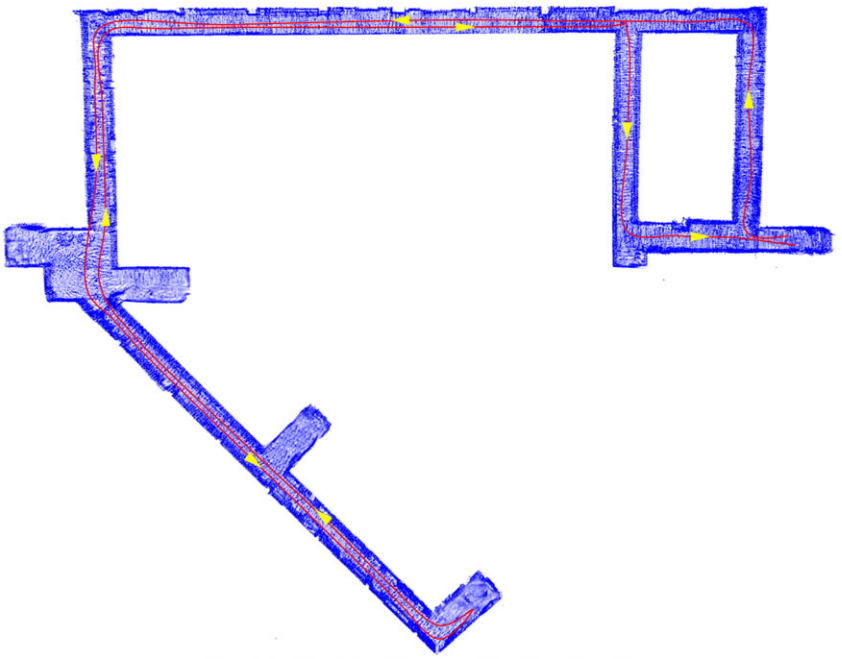


Figure 17. *The trajectory of UAV in indoor corridor.*

5. Conclusion

In this paper, a semantic knowledge database-based localization method is proposed for UAV inspection in the perceptually degraded underground mine. First, the relative pose constraint factor is constructed based on the spatial geometry features between neighboring LiDAR keyframes to realize the UAV local pose estimation. Furthermore, the dense point cloud of each intersection is extracted from the prior map. The geometrical invariant point, PWCV, and CWC descriptors are generated for constructing the semantic knowledge database. Moreover, the intersection pose constraint factor is constructed by comparing the semantic topology of the current LiDAR keyframe with the intersections in the semantic knowledge

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

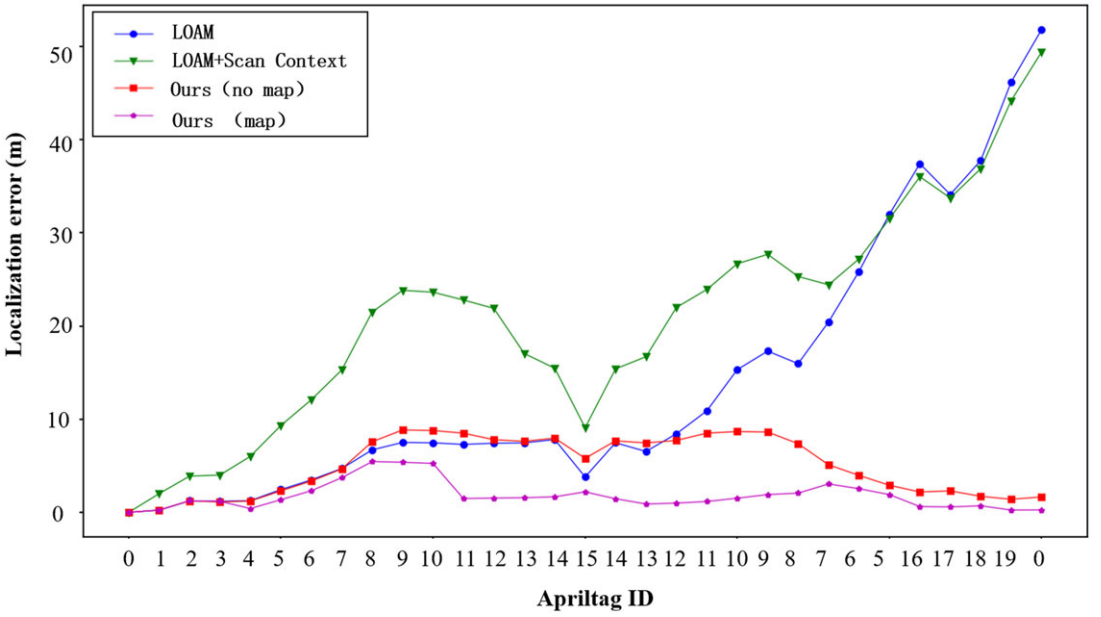


Figure 18. *The estimated trajectory error curves with different methods in the indoor corridor.*

database. Based on the pose factor graph model, the relative pose constraint factor and the intersection pose constraint factor are combined to optimize the UAV flight pose. Finally, the experimental results in the Edgar Mine and the mine-like indoor corridor demonstrate that the proposed UAV localization method proposed in this paper can realize the segmentation elimination of accumulative error, achieve high localization accuracy, and meet the needs of underground inspection and positioning.

Author contributions. Qinghua Liang and Min Chen wrote the main manuscript. Min Chen and Minghui Zhao conducted the experiment and analyze the data. Shigang Wang revised the manuscript. All authors reviewed the manuscript.

Financial support. This work is supported by "Research on key technologies of UAV in a coal mine," whose project number is 2019-TD-2-CXY007.

Competing interests. The authors declare no conflicts of interest exist.

Ethical approval. Not applicable.

References

[1] R. Mur-Artal, J. M. M. Montiel and J. D. Tardos, "Orb-slam: A versatile and accurate monocular slam system," *IEEE T. Robot.* **31**(5), 1147–1163 (2015).
[2] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE T. Robot.* **33**(5), 1255–1262 (2017).
[3] C. Campos, R. Elvira, J. J. Gómez Rodríguez, J.é M. M. Montiel and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE T. Robot.* **37**(6), 1874–1890 (2021).
[4] J. G. Rogers, J. M. Gregory, J. Fink and E. Stump. Test your Slam! The Subt-Tunnel Dataset and Metric for Mapping. *In 2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE (2020) pp. 955–961.
[5] T. Özaslan, S. Shen, Y. Mulgaonkar, N. Michael and V. Kumar, "Inspection of Penstocks and Featureless Tunnel-Like Environments Using Micro Uavs," **In:** *Field and Service Robotics, Tracts in Advanced Robotics* (Springer, 2015) vol. 5, pp. 123–136.
[6] T. Özaslan, K. Mohta, J. Keller, Y. Mulgaonkar, C. J. Taylor, V. Kumar, J. M. Wozencraft and T. Hood. "Towards Fully Autonomous Visual Inspection of Dark Featureless Dam Penstocks using Mavs." *In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE (2016) pp. 4998–5005.

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

[7] T. Özaslan, G. Loianno, J. Keller, C. J. Taylor, V. Kumar, J. M. Wozencraft and T. Hood, "Autonomous navigation and mapping for inspection of penstocks and tunnels with mavs," *IEEE Robot. Auto. Lett.* **2**(3), 1740–1747 (2017).

[8] J. Shin, S. Kim, S. Kang, S.-W. Lee, J. Paik, B. Abidi and M. Abidi, "Optical flow-based real-time object tracking using non-prior training active feature model," *Real-Time Imaging* **11**(3), 204–218 (2005).

[9] A. Jacobson, F. Zeng, D. Smith, N. Boswell, T. Peynot and M. Milford. "Semi-Supervised Slam: Leveraging Low-Cost Sensors on Underground Autonomous Vehicles for Position Tracking." *In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE (2018) pp. 3970–3977.

[10] S. S. Zhu, G. Wang, H. Blum, J. Liu, L. Song, M. Pollefeys and H. Wang, "SNI-SLAM: Semantic ,Neural Implicit SLAM", 2024 *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, Seattle, WA, USA, 2024, pp. 21167–21177. doi: 10.1109/CVPR52733.2024.02000.

[11] A. Kramer, M. Kasper and C. Heckman, "Vi-Slam for Subterranean Environments," **In:** *Field and Service Robotics, Proceedings in Advanced Robotics* (Springer, 2021) vol.16, pp. 159–172.

[12] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Int. J. Robot. Res.* **34**(3), 314–334 (2015).

[13] L. J. Chen, J. Henawy, B. B. Kocer and G. G. L. Seet. "Aerial Robots on the Way to Underground: An Experimental Evaluation of Vins-Mono on Visual-Inertial Odometry Camera." *In 2019 International Conference on Data Mining Workshops (ICDMW)*, IEEE (2019) pp. 91–96.

[14] T. Qin, P. Li and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE T. Robot.* **34**(4), 1004–1020 (2018).

[15] C. Papachristos, F. Mascarich and K. Alexis. "Thermal-inertial localization for autonomous navigation of aerial robots through obscurants." *In 2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE (2018) pp. 394–399.

[16] S. Khattak, F. Mascarich, T. Dang, C. Papachristos and K. Alexis. "Robust thermal-inertial localization for aerial robots: A case for direct methods." *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE (2019) pp. 1061–1068.

[17] S. Khattak, C. Papachristos and K. Alexis. "Visual-Thermal Landmarks and Inertial Fusion for Navigation in Degraded Visual Environments." *In 2019 IEEE Aerospace Conference*, IEEE (2019) pp. 1–9.

[18] S. Khattak, C. Papachristos and K. Alexis. "Keyframe-Based Direct Thermal–Inertial Odometry." *In 2019 International Conference on Robotics and Automation (ICRA)*, IEEE (2019) pp. 3563–3569.

[19] D. T. Fasiolo, L. Scalera and E. Maset, "Comparing lidar and imu-based slam approaches for 3D robotic mapping," *Robotica* **41**(9) 1–17 (2023).

[20] J. N. Bakambu and V. Polotski, "Autonomous system for navigation and surveying in underground mines," *J. Field Robot.* **24**(10), 829–847 (2007).

[21] S. Thrun, S. Thayer, W. Whittaker, C. R. Baker, W. Burgard, D. Ferguson, D. Hähnel, M. D. Montemerlo, A. Morris, Z. Omohundro and C. F. Reverte, "Autonomous exploration and mapping of abandoned mines." *IEEE Robotics & Automation Magazine* **11**, 79–91 (2004).

[22] J. Zhang and S. Singh, "Loam: Lidar Odometry and Mapping in Real-Time," **In:** *Robotics: Science and Systems*. vol. **2**, Berkeley, CA, (2014) pp. 1–9.

[23] T. Shan and B. Englot. "Lego-Loam: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain." *In. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE (2018) pp. 4758–4765.

[24] M. Li, H. Zhu, S. You, L. Wang and C. Tang, "Efficient laser-based 3d slam for coal mine rescue robots," *IEEE Access* **7**, 14124–14138 (2018).

[25] C. Papachristos, S. Khattak, F. Mascarich and K. Alexis. "Autonomous navigation and mapping in underground mines using aerial robots." *In 2019 IEEE Aerospace Conference*, IEEE (2019) pp. 1–8.

[26] J. F. Chow, B. B. Kocer, J. Henawy, G. Seet, Z. Li, W. Y. Yau and M. Pratama, "Toward underground localization: Lidar inertial odometry enabled aerial robot navigation." (2019) [J]. CoRR abs/1910.13085.

[27] S. Kohlbrecher, O. Von Stryk, J. Meyer and U. Klingauf. "A flexible and scalable slam system with full 3d motion estimation." *In 2011 IEEE international symposium on safety, security, and rescue robotics*, IEEE (2011) pp. 155–160.

[28] G. Grisetti, C. Stachniss and W. Burgard. Improving Grid-Based Slam with Rao-Blackwellized particle Filters by Adaptive Proposals and Selective Resampling. **In:** *Proceedings of the 2005 IEEE international conference on robotics and automation*, IEEE (2005) pp. 2432–2437.

[29] W. Hess, D. Kohler, H. Rapp and D. Andor. " Real-Time Loop Closure in 2D Lidar Slam." *2016 IEEE international conference on robotics and automation (ICRA)*, IEEE (2016) pp. 1271–1278.

[30] A. Koval, C. Kanellakis and G. Nikolakopoulos, "Evaluation of lidar-based 3d slam algorithms in subt environment," *IFAC-PapersOnLine* **55**(38), 126–131 (2022).

[31] G. Wang, X. Wu, Z. Liu and H. Wang. "Pwclo-net: Deep Lidar Odometry in 3D Point Clouds using Hierarchical Embedding Mask Optimization." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, (2021) pp. 15910–15919.

[32] K. Alexis. "Resilient Autonomous Exploration and Mapping of Underground Mines using Aerial Robots." *In 2019 19th International Conference on Advanced Robotics (ICAR)*, IEEE (2019) pp. 1–8.

[33] A. Jacobson, F. Zeng, D. Smith, N. Boswell, T. Peynot and M. Milford, "What localizes beneath: A metric multisensor localization and mapping system for autonomous underground mining vehicles," *J. Field Robot.* **38**(1), 5–27 (2021).

[34] N. J. Lavigne and J. A. Marshall, "A landmark-bounded method for large-scale underground mine mapping," *J. Field Robot.* **29**(6), 861–879 (2012).

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press

[35] M.-G. Li, H. Zhu, S.-Z. You and C.-Q. Tang, "Uwb-based localization system aided with inertial sensor for underground coal mine applications," *IEEE Sens. J.* **20**(12), 6652–6669 (2020).
[36] G. Wang, W. Wang, P. Ding, Y. Liu, H. Wang, Z. Fan, H. Bai, Z. Hongbiao and Z. Du, "Development of a search and rescue robot system for the underground building environment," *J. Field Robot.* **40**(3), 655–683 (2023).
[37] S. Li, J. Gu, Z. Li, S. Li, B. Guo, S. Gao, F. Zhao, Y. Yang, G. Li and L. Dong, "A visual slam-based lightweight multi-modal semantic framework for an intelligent substation robot," *Robotica*, **42**(7), 2169–2183 (2024). doi: 10.1017/S0263574724000511.
[38] T. Ma, G. Jiang, Y. Ou and S. Xu, "Semantic geometric fusion multi-object tracking and lidar odometry in dynamic environment," *Robotica* **42**(3), 891–910 (2024). doi: 10.1017/S0263574723001868.
[39] M. Chen, Y. Feng, M. Zhao, S. Wang and Q. Liang, "Fusion of sparse lidar and uncertainty-based 3d vision for accurate depth estimation with bayesian kriging," *Opt. Eng.* **61**(1), 013106–013106 (2022).
[40] M. Chen, Y. Feng, S. Wang and Q. Liang, "A mine intersection recognition method based on geometric invariant point detection using 3D point cloud," *IEEE Robot. Auto. Lett.* **7**(4), 11934–11941 (2022).
[41] L. Zhou, D. Koppel and M. Kaess, "Lidar slam with plane adjustment for indoor environment," *IEEE Robot. Auto. Lett.* **6**(4), 7073–7080 (2021).
[42] Y. Zhang, "Lilo: A novel lidar–imu slam system with loop optimization," *IEEE T. Aero. Elec. Sys.* **58**(4), 2649–2659 (2021).
[43] A. Chalvatzaras, I. Pratikakis and A. A. Amanatiadis, "A survey on map-based localization techniques for autonomous vehicles," *IEEE T. Intell. Veh.* **8**(2), 1574–1596 (2022).
[44] Q. Liu, X. Di and B. Xu, "Autonomous vehicle self-localization in urban environments based on 3D curvature feature points–monte carlo localization," *Robotica* **40**(3), 817–833 (2022).
[45] F. R. Kschischang, B. J. Frey and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE T. Inform. Theory* **47**(2), 498–519 (2001).
[46] S. Thrun, "Probabilistic robotics," *Commun. ACM* **45**(3), 52–57 (2002).
[47] X. Gao, T. Zhang, X. Gao and T. Zhang, "Lie Group and Lie Algebra," **In:** *Introduction to Visual SLAM: From Theory to Practice*, (2021) pp. 63–86.
[48] J. G. Rogers, J. M. Gregory, J. Fink and E. Stump. "Test your Slam! The Subt-Tunnel Dataset and Metric for Mapping." *In 2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE (2020) pp. 955–961.
[49] Z. Zhang and D. Scaramuzza. "A Tutorial on Quantitative Trajectory Evaluation for Visual (-Inertial) Odometry." *In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE (2018) pp. 7244–7251.
[50] J. Wang and E. Olson. "Apriltag 2: Efficient and Robust Fiducial Detection." *In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE (2016) pp. 4193–4198.

Cite this article: Q. Liang, M. Zhao, S. Wang and M. Chen (2024). "A semantic knowledge database-based localization method for UAV inspection in perceptual-degraded underground mine", Robotica **42**, 3480–3504. https://doi.org/10.1017/S0263574724001474

https://doi.org/10.1017/S0263574724001474 Published online by Cambridge University Press