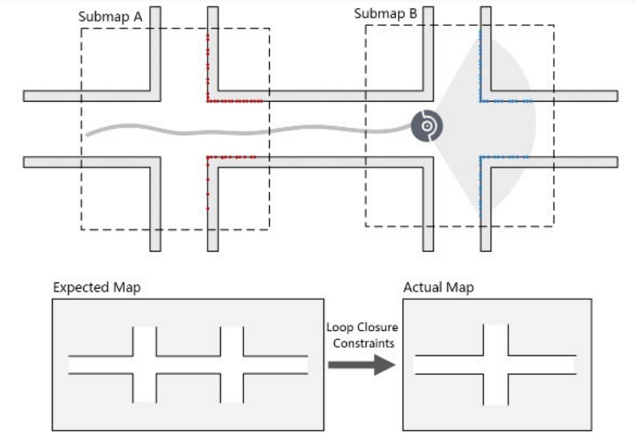
A 2-D LiDAR-SLAM Algorithm for Indoor Similar Environment With Deep Visual Loop Closure

Zongkun Zhou, Chi Guo, *Member, IEEE*, Yanyue Pan, Xiang Li, and Weiping Jiang

***Abstract*—Simultaneous localization and mapping (SLAM) is the key technology in the implementation of robot intelligence. Compared with the camera, the higher accuracy and stability can be achieved with light detection and ranging (LiDAR) in the indoor environment. However, LiDAR can only acquire the geometric structure information of the environment, and LiDAR SLAM with loop detection is prone to failure in scenes where the geometric structure information is missing or similar. Therefore, we propose a loop closure algorithm, which fuses visual and scan information, makes use of the deep features for loop detection, and then combines camera and LiDAR data for loop verification. We name it fusion SLAM (FSLAM), which uses a tight coupling to fuse the two for loop correction. We compare the differences between visual feature extraction based on deep neural network hierarchical feature network (HF-Net) and handcrafted feature extraction algorithm ORB. The proposed FSLAM method is able to successfully mapping in scenes with similar geometric structures, while its localization and mapping accuracy is significantly improved compared to other algorithms.**



***Index Terms*— Data fusion, light detection and ranging (LiDAR), loop closure, simultaneous localization and mapping (SLAM), visual feature extraction.**

I. Introduction

WITH the rapid development of artificial intelligence and the growing demand for industrial and social intelligence, autonomous robots have received extensive attention. Due to the increasingly complex tasks, the demand for intelligent robots is also increasing. Simultaneous localization and mapping (SLAM) is a key technology in the implementation of robot intelligence [1]. When a robot starts from an unknown location in an unknown environment, it can build a consistent map and determine its position in the map. With SLAM technology, robots can achieve indoor and outdoor localization and navigation in unknown environments.

Unlike outdoor positioning, global navigation satellite system (GNSS) signals are weak indoors due to building occlusion, which makes it difficult to perform positioning and navigation tasks. Generally, indoor positioning and navigation tasks usually use multiple base stations, such as wireless fidelity (Wi-Fi), Bluetooth, or ultrawideband (UWB) [2], [3], [4]. By using these passive positioning technologies, the three-dimensional (3-D) location can be obtained by measuring the distance between the robot and the base station. Nevertheless, the major drawback of these implementations is that a very dense network of base stations has to be preestablished, which is costly and unrealistic in large-scale areas. In contrast, SLAM uses only the robot's own sensors to implement navigation tasks without any nearby base station signals, which makes SLAM technology become an important part of indoor localization and navigation.

SLAM can be divided into light detection and ranging (LiDAR)-based and visual-based, and with the development of computer vision, visual SLAM has received much attention and is a hot spot in the current research field because of its large amount of information and a wide range of applications.

Manuscript received 19 January 2023; revised 16 March 2023; accepted 17 March 2023. Date of publication 24 March 2023; date of current version 29 June 2023. This work was supported in part by the Science and Technology Major Project of Hubei Province of China under Grant 2021AAA010 and in part by the Open Fund of Hubei Luojia Laboratory. The associate editor coordinating the review of this article and approving it for publication was Prof. Kazuaki Sawada. *(Corresponding author: Chi Guo.)*

Zongkun Zhou and Weiping Jiang are with the GNSS Research Center, Wuhan University, Wuhan 430072, China (e-mail: zhouzk@whu.edu.cn; wpjiang@whu.edu.cn).

Chi Guo is with the GNSS Research Center and the Artificial Intelligence Institute, Wuhan University, Wuhan 430072, China, and also with the HuBei Luojia Laboratory, Wuhan 430072, China (e-mail: guochi@whu.edu.cn).

Yanyue Pan and Xiang Li are with the Artificial Intelligence Institute, Wuhan University, Wuhan 430072, China (e-mail: panyy1998@163.com; 2017302650114@whu.edu.cn).

Digital Object Identifier 10.1109/JSEN.2023.3260104

Also, LiDAR-based SLAM is the most stable mainstream SLAM technology with higher stability and accuracy by emitting laser beams in all directions to directly acquire environmental information. Most applications use 2-D LiDARs to localization and mapping and are also widely used in the industry.

However, the LiDAR can only obtain the geometric information of the environment, which may make the LiDAR-only SLAM prone to positioning or mapping problems, especially in the case of being lack of geometric texture or repeated geometric structure. In the environment where parallel corridors exist, because of the similar geometric structure of the corridors and their proximity in the map, two corridors will be treated as the same corridor when the map is built and thus be fused. In the library environment, because the geometric structures of the shelves are very similar to each other, it is easy to mistake the shelves in different rows as detecting the loop closure, and the optimization will fuse the shelves in different rows, resulting in serious errors in mapping.

To solve these problems, we propose an accurate and robust system for localization and mapping in indoor similar environment, which fuses 2-D LiDAR and monocular information with a tightly coupled manner. Our design presents the following contributions.

1) We propose a loop closure verification method that combines a visual frame and laser frame. The loop frame is detected by visual data with deep learning network and verified by laser data to avoid false loop closure.
2) We design a loop constraint and use scan-to-submap matching to fuse scan and visual in tight coupling. In addition, the nonlinear optimization is used to reduce the cumulative error of the system. This method can be used effectively even on sparse 2-D LiDAR point cloud.
3) We compare the performance of feature extraction algorithm between the deep learning technology and the traditional handcrafted way, verify the superiority of deep learning in a similar environment, and design the loop closure detection algorithm accordingly.
4) The proposed SLAM system can operate successfully in scenes with similar geometric structures, while its localization and mapping accuracy is significantly improved.

The rest of this article is organized as follows. Related work is reviewed in Section II. The overview of the proposed system is presented in Section III. Section IV presents the detailed algorithm descriptions applied in our system, followed by experimental results in Section V. Finally, Section VI provides a summary of this article.

II. Related Work

In this section, we briefly review works on LiDAR SLAM and loop closure detection.

A. LiDAR SLAM

Two-dimensional laser-based SLAM is often performed by using a filter-based method, such as the GMapping algorithm, which uses Rao–Blackwellized particle filters (RBPFs) to predict the state transition function [5]. In order to reduce the cumulative error of filter methods, more and more systems adopt optimization methods. Karto-SLAM, which proposes sparse pose adjustment (SPA) to efficiently compute the sparse matrix from the constraint graph, greatly accelerates the convergence speed [6]. Hector-SLAM is based on the Gauss–Newton iteration formula, and the scan matching by multiresolution grid map effectively improves the real-time line, but the algorithm does not include loop detection, so it is also difficult to eliminate the cumulative error [7]. Furthermore, cartographer uses scan-to-submap matches to solve the pose, using a nonlinear optimization to make the result better, and it also adds a loop detection module, which is one of the widely used algorithms in 2-D laser-based SLAM algorithm [8]. After that, the 2-D laser SLAM algorithm focuses more on simplifying the calculation, such as VinySLAM [9], [10], [11], to reduce its dependence on high-cost hardware.

As a result, more and more work focuses on visual-LiDAR fusion-based SLAM, which can be divided into four research routes [12]. The first is the classical formulation of extended Kalman filter (EKF) SLAM [13], [14]; the second method is to use LiDAR measurement to improve visual SLAM [15], [16], [17]. Conversely, there are also ways to optimize LiDAR slam with visual data [18], [19], and other studies try to combine both LiDAR and visual-SLAM results [20], [21], [22].

B. Loop Closure Detection

Loop closure detection and correction are essential to high-precision SLAM data processing. The long-term error accumulation could be corrected by matching the historical state in the loop detection module. Compared with LiDAR information, visual data can obtain more environmental information such as structure and texture, so it is more conducive to loop detection.

In the field of computer vision, the common loop detection is realized by using traditional methods. When SIFT, SURF, or oriented FAST and rotated BRIEF (ORB) features are first extracted in the new frames, we match them to the recent frames, and then, the camera motion can be reliably estimated, such as ORB-SLAM [23], [24], [25].

However, traditional feature extraction is easily affected by factors such as light intensity change and motion blur, so deep learning algorithms have also been widely used in loop detection of slam in recent years [26]. Deep learning and convolutional neural networks can generate image descriptors; afterward, the descriptors of two frames of images are matched for loop detection [27], [28]. Some studies, such as SuperPoint and GCN, indicate that deep learning networks can obtain image feature points instead of ORB features and then combine with bag-of-words library for loop closure detection [29], [30], [31]. There are also some algorithms, such as DXSLAM [32], which obtain image global descriptors and image feature points simultaneously through deep learning network and use image global descriptors for loop detection.

III. System Overview

The overall system structure is presented in Fig. 1. The SLAM framework in this article uses a 2-D LiDAR as the core

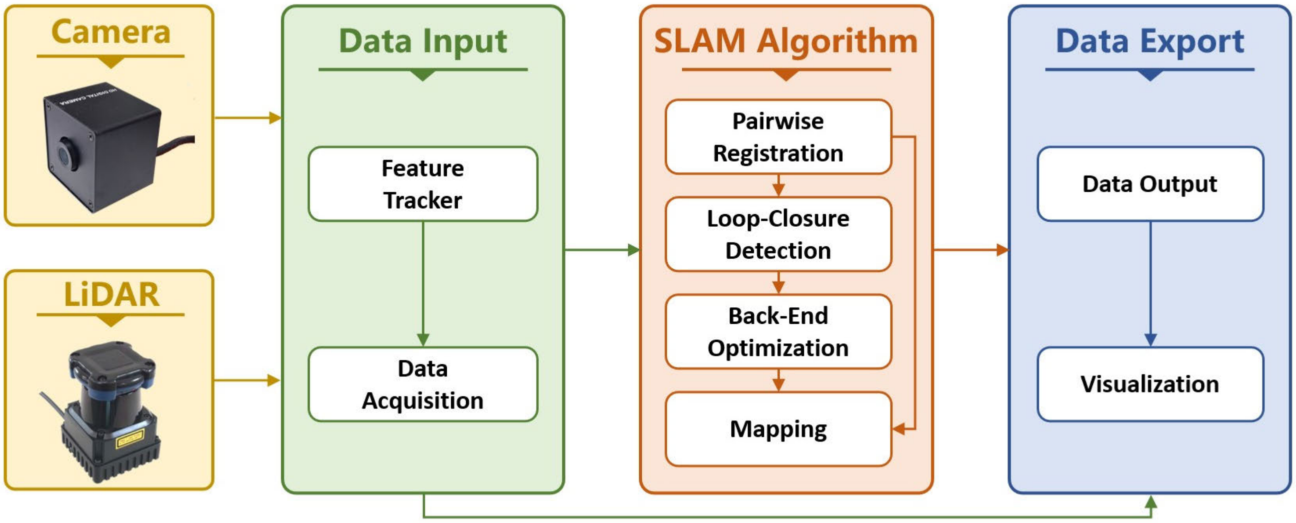


Fig. 1. Structure of FSLAM system.

sensor for localization and mapping. Because the monocular has low accuracy and robustness in motion estimation, it is only used in loop closure detection. The system is built based on cartographer [6], [8], [33], which is the classical 2-D LiDAR SLAM algorithm. The proposed loop closure algorithm, which used HF-Net, replaces the original one and maintains compatibility with the previous system. Also, the complete SLAM system includes the data input module, localization and mapping module, and data output and visualization module.

IV. Methodology

A. Keyframe Construction

A front-end registration module uses sensor data to estimate the motion of the robot. Because LiDAR and visual sensors are both used in this article, the accuracy of keyframe construction is critical to the accuracy of subsequent mapping. The difficulties of keyframe construction mainly include the processing of the visual data, and the fusion of the visual data and laser scans. There are four basic steps: the calibration of camera and LiDAR, the temporal calibration and interpolation, solving the world coordinates of visual feature points, and the keyframe construction.

Zhang et al. [34] proposed an algorithm for extrinsic calibration of monocular cameras and 2-D LiDAR. This article is based on the algorithm theory for the external calibration between monocular camera and 2-D LiDAR, and the transformation matrix $T_{cl} = (R_{cl}, t_{cl})$ between the LiDAR frame and the camera frame can be obtained through the extrinsic calibration, where R_{cl} and t_{cl} represent rotation and translation, respectively. The data can be transformed between two coordinate systems to ensure the spatial consistency according to the transformation matrix.

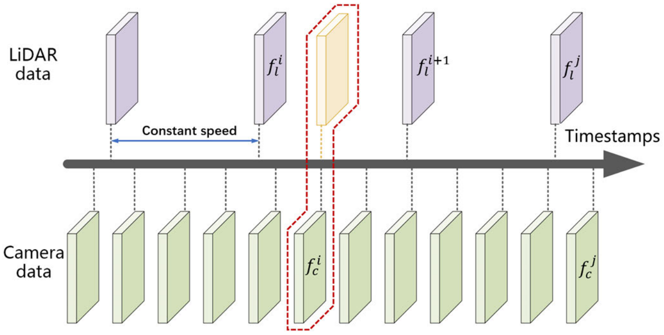


Fig. 2. Schematic of the temporal calibration and interpolation.

Temporal calibration and interpolation are used to solve the problem of sensors with different acquisition frequencies. As shown in Fig. 2, due to the different sampling frequencies, the laser frame may not exist under the corresponding time stamp when the visual frame is acquired. Thus, the coordinates of the two data cannot be directly converted. In this article, we interpolate the number of laser frames to fit the visual frame, and only the laser frame poses are transformed to synchronize with the visual frame to reduce the computational effort.

We define the scan frame as f_l^i, and the transformation from the laser coordinate system to the world coordinate system at this moment is represented by the rotation matrix $R_{wl}^i \in \mathbb{R}^{3\times3}$ and the translation vector $t_{wl}^i \in \mathbb{R}^3$, which are obtained by scan matching. The image frame that arrives at between f_l^i and f_l^{i+1} is f_c^i. The transformation from the camera coordinate system to the world coordinate system is represented as the rotation matrix $R_{wc}^j \in \mathbb{R}^{3\times3}$ and the translation vector $t_{wc}^j \in \mathbb{R}^3$, which are the quantities to be solved in this section. The rotation matrix $R_{cl}^i \in \mathbb{R}^{3\times3}$ and translation vector $t_{cl}^i \in \mathbb{R}^3$ from the LiDAR coordinate system to the camera coordinate system are obtained by extrinsic calibration.

The purpose of temporal interpolation is to solve for the laser frame pose (R_{wl}^j, t_{wl}^j) at the moment corresponding to the image f_c^j. The LiDAR motion is modeled with constant angular and linear velocities during a sweep, and then, the amount of change in the system state can also be considered as constant speed in a short time interval. Thus, we can interpolate the laser frames uniformly between f_l^i and f_l^{i+1}, which could be calculated through the following equations:

$$\theta_j = \theta_i + \frac{\theta_{i+1} - \theta_i}{\Delta t_{i,i+1}} \Delta t_{i,j} \tag{1}$$

$$R_{wl}^{j} = \text{Rodrigues}\left(\theta_j\right) \tag{2}$$

$$t_{wl}^{j} = t_i + \frac{t_{wl}^{i+1} - t_{wl}^{i}}{\Delta t_{i,i+1}} \Delta t_{i,j} \tag{3}$$

where θ_i and θ_j are the rotation in the LiDAR frame f_l^i and camera frame f_c^j, respectively, Δt is the time between two timestamps, and R_{wl}^j is a rotation matrix with the Z-axis defined by the Rodrigues formula.

Then, we can transform the visual data from the camera coordinate system to the world coordinate system, which is computed by

$$R_{wc}^{j} = R_{wl}^{j} R_{lc} = R_{wl}^{j} R_{cl}^{T} \tag{4}$$

$$t_{wc}^{j} = R_{wl}^{j} t_{lc} + t_{wl}^{j} = R_{wl}\left(-R_{cl}^{-1} t_{cl}\right) + t_{wl}^{j}. \tag{5}$$

The key to solving for 3-D position of feature point in the global frame is to compute the relative pose between two frames to triangulate a set of feature points, and monocular SLAM requires motion information to solve this problem because depth cannot be recovered from a single image. We use the above solved camera pose for the triangulation directly, which reduces the computational effort and avoids the scale ambiguity for monocular camera.

$\left(R_{cw}^{i}, t_{cw}^{i}\right)$ and (R_{cw}^{j}, t_{cw}^{j}) are the camera poses at f_c^i and f_c^j, and the relative poses of the two frames are solved as follows:

$$R_{ij} = R_{cw}^{i} R_{wc}^{j} = R_{wc}^{iT} R_{wc}^{j} \tag{6}$$

$$t_{ij} = R_{cw}^{i} t_{wc}^{j} + t_{cw}^{i} = R_{wc}^{iT} t_{wc}^{j} + \left(-R_{wc}^{iT} t_{wc}^{i}\right). \tag{7}$$

The feature points of the two frames are matched based on the Manhattan distance, and we can use an epipolar geometry to estimate the 3-D coordinates of two corresponding 2-D image points in a stereo image pair

$$P_i^T F_{ij} P_j = 0 \tag{8}$$

where F_{ij} is the fundamental matrix of two frames and $P_i = (u_i, v_i)$ and $P_j = (u_j, v_j)$ are the matching feature of f_c^i and f_c^j, respectively. However, the point is not in the theoretical position because of the errors, so we check whether the distance of epipolar line is below a threshold th_{epipolar} to definitively accept the image pair.

When multiple feature matches are obtained, the 3-D coordinates can be solved by a direct linear transform (DLT) method to construct 3-D map points

$$\begin{bmatrix} p_i \times T_{iw} P_w \\ p_j \times T_{jw} P_w \end{bmatrix} = \begin{bmatrix} x_i T_{iw}^{3T} - T_{iw}^{1T} \\ y_i T_{iw}^{3T} - T_{iw}^{2T} \\ x_j T_{jw}^{3T} - T_{jw}^{1T} \\ y_j T_{jw}^{3T} - T_{jw}^{2T} \end{bmatrix} P_w = A P_w = 0 \tag{9}$$

where $P_w = (X, Y, Z, 1)$ is the 3-D coordinates of P in the world coordinate system and $p_i = (x_i/z_i, y_i/z_i, 1)$ and $p_j = (x_j/z_j, y_j/z_j, 1)$ are the camera normalized coordinates of the corresponding image frames f_c^i and f_c^j, respectively. The transformation of f_c^i and f_c^j from the camera coordinate system to the world coordinate system is, respectively, represented as $T_{iw} = (R_{iw}, t_{iw})$ and $T_{jw} = \left(R_{jw}, t_{jw}\right)$. T_{iw}^{kT} represents the k column of T_{iw}. Solving the problem by singular value decomposition (SVD), we can get P_w. We will check the results and remove the outliers by the chi-square test. A 3-D map point is accepted when the reprojection error of both map point projection to image f_c^i and f_c^j is less than threshold; otherwise, it will be discarded.

After the above operations, we put the matched frames as one keyframe and denote it as $F = \{f_l, f_c, m_{\text{sub}}\}$. f_l is the scan frame containing the point cloud data and the pose of LiDAR; f_c is the visual frame containing the features, the local descriptor, the global descriptor, and the 3-D map point set; and m_{sub} is the submap.

B. Loop Closure Detection

Two-dimensional LiDAR has limited access to environmental information because it can only acquire data in the $x-y$ plane. When using a large-scale scene data for loop closure detection, the limitations of point cloud can lead to mismatching or even wrong state estimation. Therefore, we use visual data for loop closure detection and LiDAR to provide more accurate environmental structure information, making full utilization of the advantages of both data and compensating for their respective shortcomings.

The keyframes F, which were constructed previously, contain numerous pieces of information. Searching for similar images in the history frames will bring a large amount of computation as the robot motion time gradually grows, so the system has embedded a bag-of-words (BoW) [35] place recognition module to alleviate the computational effort of image matching. By using the k-dimensional (k-D) trees in the BoW model, the most similar image, which is the loop frame, is gradually searched by comparing the distance between the word vectors and the clustering center.

We construct covisibility information between keyframes inspired by ORB-SLAM [23], and keyframes with covisible relationship will be excluded when searching for loop frames. Considering the different scenes and the real-time performance of the algorithm, we use the Euclidean distance of global descriptors between the current keyframe and the loop closure frame as the matching score instead of verifying all loop closure frames, which is calculated as follows:

$$s_{\text{global}} = d\,(x, y) = \sqrt{\sum\nolimits_i (x_i - y_i)^2} \tag{10}$$

where x is the global descriptor of the current frame F_{current} and y is the global descriptor of the loop closure frame. The loop candidates are sorted according to this matching score, and then, we will check the consistency in sequence starting from the higher scores. Once a loop candidate frame passes verification, we will discard subsequent candidates.

Since loop closure is prone to errors if only image information in the 2-D pixel plane of the keyframe is used, the environmental information of the scan is used for the verification

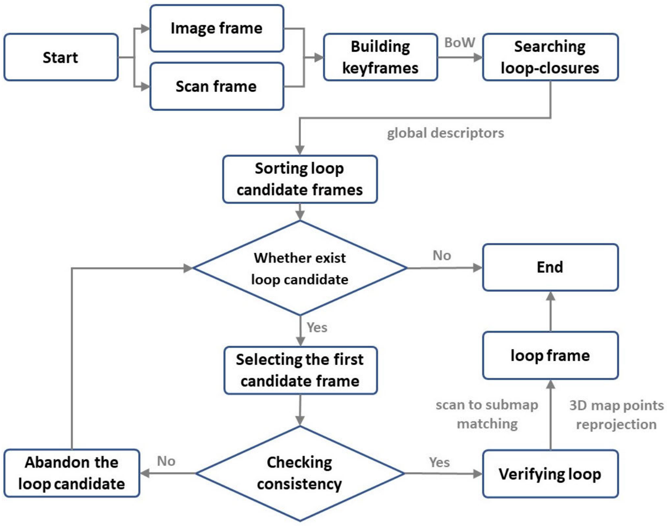


Fig. 3. Structure of the loop closure detection.

of loop frames. We perform a branch-and-bound approach [33] for computing scan-to-submap matches as constraints, and it means searching for a robot pose that best matches the current scan data to the submap

$$T = \underset{T \in W}{\operatorname{argmax}} \sum_{k=1}^{K} M_{\text{nearest}}\left(T h_k\right) \tag{11}$$

where T transforms scan points $\{h_k\}_{k=1,\ldots,K}$, $h_k \in \mathbb{R}^2$ from the scan frame to the submap frame according to the scan pose, K is the scan points, W is the search window, and $M_{\text{nearest}}(Th_k)$ is submap M extended to all of point by rounding its arguments to the nearest grid point first, which is extending the value of a grid points to the corresponding pixel [8].

Moreover, for the loop submap that passed the scan validation, we reproject the 3-D map points in this submap to the current image frame to find the matching feature points and accept the loop frame if the number of matched feature point pairs exceeds the threshold. This step verifies the loops while ensuring enough matching points for subsequent loop closure optimization.

The structure of the algorithm is shown in Fig. 3.

C. Loop Closure Optimization

The current frame and the loops frame in loop closure allow us to correct for pose and scale drift by detecting when the robot has reached a previously reached position in the environment. In this stage, this article also fuses the visual and laser data to avoid the mismatching problem of LiDAR in large scenes. To fuse the two kinds of data in order to optimize the pose and improve the accuracy, we design a loop closure correction algorithm, which is based on nonlinear optimization of tight coupling.

A reasonable loss function is the key to optimization. For laser point clouds and visual map points, we construct loss functions based on scan-to-submap matching and reprojection errors. After normalizing the data to unify the data magnitudes, we also add weights to the loss terms for considering the reliability of the data sources. Considering that the optimized poses should not differ much from the initial poses, the translational and rotational variations are also constructed as loss terms to avoid mismatching at a distance. Finally, we construct a joint loss function by comprehensive consideration.

For LiDAR, laser scans and submap are at the best estimated position when the scan matcher is responsible for finding a scan pose that maximizes the probabilities at the scan points in the submap. We cast this as a nonlinear least-squares problem. The laser loss term is normalized as follows:

$$F\left(\xi_{ij}\right) = \left(\frac{1}{K}\sum_{k=1}^{K}\left\|1 - M_{\text{smooth}}\left(T_{\xi_{ij}} h_k\right)\right\|^2\right) \tag{12}$$

where $\xi_{ij} = (t_{ij}, \theta_{ij})$ is the pose of the scan frame in the coordinate system of the loop submap j and $T_{\xi_{ij}}$ is the conversion matrix of the scan point h_k to the submap j.

For vision, there are multiple sets of matching 3-D map points between the submap and the current frame, and there is a reprojection error between these map points and their 2-D projection points into the current frame. Therefore, we construct the visual loss term by minimizing the total projection error of all matching points, which is computed by

$$F\left(R_{cw}, t_{cw}\right)_c = \left(\frac{1}{M}\sum_{m=1}^{M}\frac{\left\|P_P^m - C\left(R_{cw}P_w^m + t_{cw}\right)\right\|^2}{2r^2}\right) \tag{13}$$

where $P_p = \left\{P_P^m\right\}_{m=1,\ldots,M}$ is the set of features in the current frame that gets matched with the map points in the loop submap, M is the number of matching point pairs, the camera intrinsic matrix is C, and r is the search radius when searching for feature matching. (R_{cw}, t_{cw}) is the transformation from the world coordinate system to the current frame camera coordinate system.

Finally, the joint loss function is shown in the following:

$$\begin{aligned} F\left(\xi_{ij}\right) &= \alpha F\left(\xi_{ij}\right)_l + \beta\left(\xi_{ij}\right)_c \\ &\quad + \gamma\left(F\left(\xi_{ij}\right)_{\text{trans}} + F\left(\xi_{ij}\right)_{\text{rot}}\right) \\ &= \left(\alpha\left(\frac{1}{K}\sum_{k=1}^{K}\left\|1 - G\left(\xi_{ij}h_k\right)\right\|^2\right)\right. \\ &\qquad + \left(\beta\frac{1}{M}\sum_{m=1}^{M}\frac{\left\|P_P^m - CT_{\xi_{ij}p}P_w^m\right\|^2}{2r^2}\right) \\ &\qquad \left. + \gamma\left(\frac{\Delta x}{th_x} + \frac{\Delta y}{th_y} + \frac{\Delta\theta}{\pi}\right)\right) \end{aligned} \tag{14}$$

where α, β, and γ are the weights assigned to the different loss terms, which can be set according to the sensor and application scenario. If the environment has rich geometric structure or less visual texture, or if the reliability of the camera is weak, boost α, and vice versa for β. Considering that the optimized poses should not differ much from the initial poses, the translational and rotational changes quantities are also constructed as loss terms to avoid mismatching at long distances, so Δx, Δy, and $\Delta\theta$ are variables of translational and angular and th_x and th_y are the predefined translation thresholds.

D. Evaluation Methods

In order to test the feature extraction algorithm's performance, we use precision to measure the accuracy of image retrieval. Loop closure accuracy is also known as check accuracy, and the higher the accuracy, the higher the probability that the retrieval is accurate, and the higher the accuracy of loop closure detection

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{15}$$

TP means the number of true positive loops, FP indicates the number of false positive loops, and the sum of TP and FP is the total number of loops N.

To implement the robustness analysis, we measure the difference of feature extraction algorithms in terms of tracking loss rate (LR) and average retracking frames (ARF) of the vision module. Considering that different algorithms have different computation rates, the ratio of trace loss time to total run time is used as the LR

$$\text{LR} = \frac{\sum_i^N \left(t_i^{\text{end}} - t_i^{\text{start}}\right)}{t^{\text{end}} - t^{\text{start}}} \tag{16}$$

where N is the number of losses, t_i^{start} and t_i^{end} are the start and end moments of the ith loss, respectively, and t^{start} and t^{end} are the start and end moments of the system operation, respectively.

Retracking frames refer to the number of frames it takes after a tracking loss to successfully track again. This reflects the ability to quickly restart tracking after a tracking loss. ARF is calculated as in (16)

$$\text{ARF} = \frac{1}{N}\sum_i^N N_i^{\text{frame}} \tag{17}$$

where N is the number of losses and N_i^{frame} is the number of frames required for the ith successful retracking.

V. Experiments

In this section, we present some results of our SLAM algorithm computed from recorded sensor data. We find that indoor datasets in the SLAM field are usually based on single-sensor data. Even if some datasets contain multisensor data, except for the main sensor, the other sensor data are mostly used as reference values, so calibration information between sensors is not provided. The situation makes it difficult for the existing datasets to evaluate the indoor SLAM algorithm of camera and LiDAR fusion. Therefore, considering that it is easier to obtain the true values in the simulation environment, this article has made a test dataset for verifying the indoor SLAM algorithm of the fusion sensor in the simulation environment. At the same time, the dataset of the real world is also collected to prove the feasibility and accuracy of the algorithm in the actual environment.

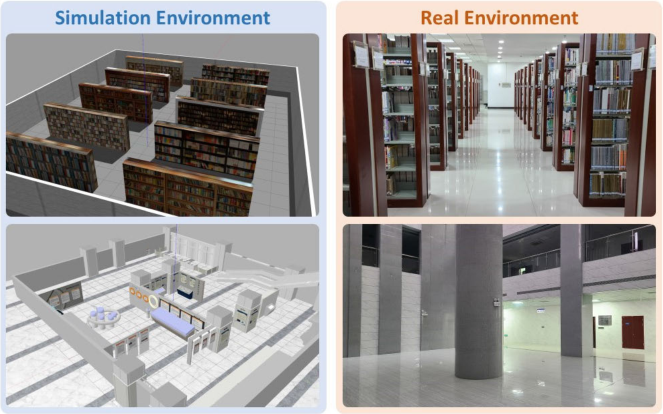


Fig. 4. Simulation and real environment of library shelf and hall.

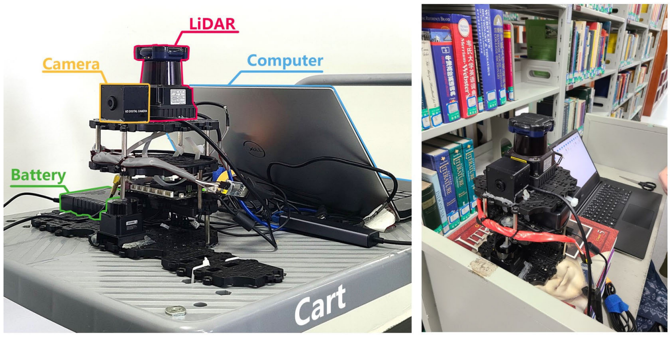


Fig. 5. Data collection equipment on a cart.

We experiment in two areas, the exhibition hall and the library shelves. Considering the evaluation of the trajectory absolute error and the convenience of the experiment, we build a simulation environment based on Gazebo nine under the ROS Melodic operating system before the field experiment. We record the real trajectory of the movement to compare the accuracy of the algorithm in the simulation environment and compare the reliability of the actual trajectory and the built map in the real scene. The simulation and real scenes are shown in Fig. 4.

The library bookshelf environment is a typical geometrically similar scene, and our cart goes around many bookshelves that have the same spacing and height. The appearance of each bookshelf has a large difference because books on the shelf have different colored covers, which can solve the mapping error problem in texture. The exhibition hall contains no similar geometric structures but has similar appearance textures. Less appearance texture information can easily lead to tracking failure problems for algorithms that rely on visual localization. These two different experimental environments can help to evaluate the performance and effectiveness of the algorithm we proposed.

The experimental equipment is shown in Fig. 5. The Inter Core i7-1075H CPU, 16-GB RAM, and Nvidia GeForce RTX2060 GPUs are used for the simulations; the turtlebot3 robot, which made by Korean company ROBOTIS, is used as the frame, with a Hokuyo 2-D LiDAR on top and a monocular camera placed in the vicinity, and the calibration of LiDAR and camera is performed via Autoware software. The motion system of the robot is removed and the whole robot is tied on the library cart. The Raspberry Pi control board of the upper computer is replaced by PC. The frame rate of the LiDAR is 10 Hz, and the photo resolution is 680 × 480 with 20-Hz frequency. The resolution can be effective in feature extraction at the same time, but also to ensure that the amount of calculation is not too large. The camera is wired to the computer via USB, and the LiDAR is connected with a network cable to ensure the low latency of data transmission. The acquired data will be postprocessed, and the experimental configuration used for postprocessing is consistent with the simulation experiment.

To further verify the performance of the FSLAM system, we choose three kinds of algorithm, cartographer without loops, cartographer, and ORB-FSLAM as the control group; ORB-FSLAM is FSLAM using the ORB feature extraction algorithm. The evaluation of the localization accuracy is performed in both qualitative and quantitative ways. For a qualitative analysis, it is possible to visually compare the differences between the trajectories estimated by each SLAM algorithm and the true values, and the robustness of different algorithms in different scenarios and different routes can be compared through a qualitative analysis. In quantitative assessment of localization, absolute trajectory error (ATE) is often used to measure global consistency. Since the algorithm in this article focuses on reducing the global cumulative error, the ATE and its corresponding statistical metrics, which contain mean, median, standard deviation (Std), the sum of squares due to error (SSE), and root-mean-square error (RMSE), will be used for algorithm evaluation. The analysis of the map accuracy will be performed in a quantitative way. Because the true map is difficult to obtain the distance between key points, we can measure the distance of door to door, shelf size, corridor length, and so on as the true value. We calculate the error between the true value and the measurement value in the map to assess the mapping quality.

A. Feature Extraction

Among the handcrafted visual features, the ORB algorithm is widely known [36]. ORB feature builds on the FAST keypoint detector and the BRIEF descriptor. Also, ORB-SLAM has become one of the most representative approaches in visual SLAM. On the other hand, feature extraction based on deep learning networks has replaced the handcrafted features to dominant related research and applications. SuperPoint in SuperPoint-SLAM is a representative algorithm of deep learning feature extraction algorithm used in the SLAM field [4], which can extract features in real time. SuperPoint is a self-supervised framework for keypoint detection and feature description, which has a better performance than ORB on some datasets. Another representative algorithm is HF-Net, which is applied in DXSLAM [32] for feature extraction. HF-Net [37] can give both local features (key points and their descriptors) and global features (image descriptors) with a single CNN model and also achieve good results in real-time conditions. We experiment with a variety of feature extraction algorithms in the preexperimental stage, and after multiple comparisons, we choose the above two as representatives of traditional and deep learning methods for validation.

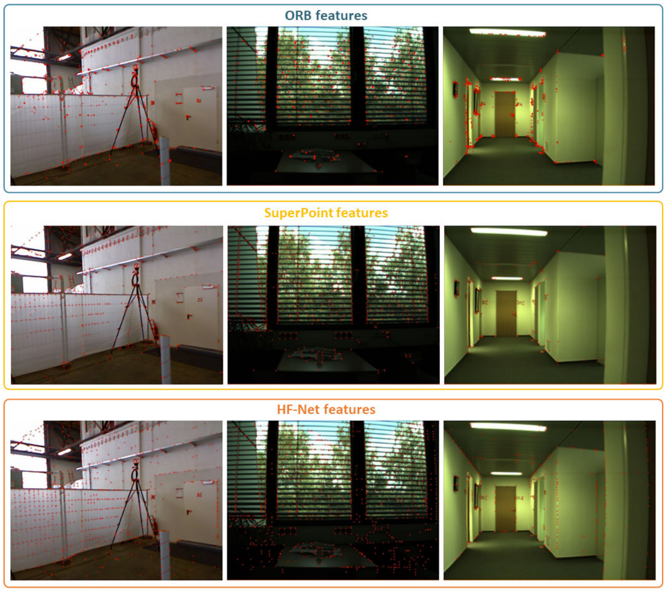


Fig. 6. Result of feature extraction.

To clarify the advantages of deep features, this article compares the differences between deep and handcrafted feature extraction algorithms. To deal with this task, we have built an image test dataset. The data come from multiple public datasets [38], [39], [40], [41], and only the indoor scene images in the public dataset are retained. We conduct a qualitative experimental analysis on the extraction effects of ORB, SuperPoint, and HF-Net. The maximum number of features extracted by both algorithms is 1000, and the experimental results are shown in Fig. 6. Three representative images in the test set are selected as examples to demonstrate the feature point extraction effect of various algorithms. The first column is a low-textured region, where the ORB feature extractor obtains fewer feature points on the white wall, while the deep feature extractor almost restores the texture on the wall; the second column is a dim environment, as seen in the lower part of the image, where the ORB extracts fewer feature points, while the deep feature extractor can still extract more feature points in this environment; and the third column of the image is a long corridor environment with low-textured, where the HF-Net can extract more structured feature points than others. It can be verified that the deep features can be applied in more environments than ORB features, are less affected by illumination, and can have good feature point detection ability even in environments lacking visual textures.

In the quantitative experiments, we apply the three feature extraction algorithms to FSLAM and conduct the experiments in two simulation environments, and the results are shown in Table I. In both scenarios, HF-FSLAM detects 251 and 230 loop frame pairs, which is significantly higher than the handcrafted algorithm. Although the precision of ORB-FSLAM is also good, its low number of loop frame pairs indicates that it detects fewer effective loops. Meanwhile, the number of loop frame pairs detected by SP-FSLAM in the two

TABLE I
ACCURACY OF DIFFERENT FEATURE EXTRACTION METHODS

Algorithm	Library bookshelf		Exhibition hall	
	Loop pairs	Precision	Loop pairs	Precision
ORB-FSLAM	25	0.960	8	0.989
SP-FSLAM	13	0.846	142	0.620
HF-FSLAM	251	0.992	230	0.996

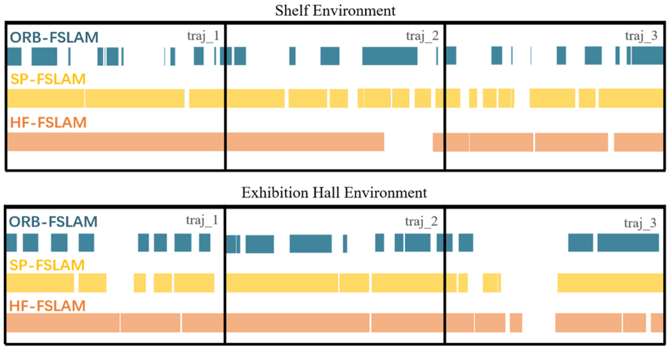


Fig. 7. Tracking diagram in virtual environment.

TABLE II
RESULTS OF TRACKING ROBUSTNESS

Algorithm	Library bookshelf		Exhibition hall	
	Loss rate	ARF	Loss rate	ARF
ORB-FSLAM	0.591	25	0.432	22
SP-FSLAM	0.135	7	0.202	12
HF-FSLAM	0.050	7	0.095	8

environments is quite different, which indicates that the loop detection rate fluctuates greatly, and the algorithm is unstable in two scenes.

In the robustness experiments of loopback detection, we recorded three sets of motion data in two simulation environments. As shown in Fig. 7, the solid colored line indicates the actual tracking situation, and the broken part is the location where tracking is lost. The robustness results of the two algorithms in the library simulation environment are shown in Table II. Regardless of the environment, the LR of HF-FSLAM can be maintained below 0.1, and the tracking continuity is much better than that of ORB-FSLAM and SP-FSLAM. In terms of retracking frames, HF-FSLAM and SP-FSLAM perform much better than ORB-FSLAM, and the value of HF-FSLAM is lower than that of SP-FSLAM, which indicates that it is more capable of restarting tracking after losing tracking.

From the above analysis, the algorithm with deep features has a lower tracking LR than the traditional handcrafted method while ensuring the real-time performance of the algorithm, which indicates that the former has a higher rate of effective tracking and has better robustness. Meanwhile, the method with deep features has a lower average retracking frame rate, which means that it can restart tracking more quickly after tracking loss. Therefore, the deep feature extraction method is more suitable for this article in terms of the number of features and robustness.

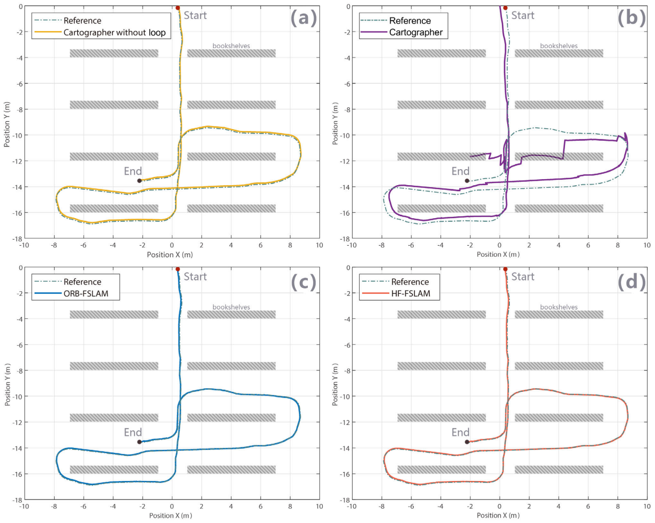


Fig. 8. Comparison of trajectory errors for library shelf scenes.

B. Results of the Trajectory

The rosbag is recorded, respectively, in the simulation environment of the library environment and the exhibition hall, which record the sensor data and the true value of the pose of the robot. We run each group of algorithms separately and align the trajectories of our system and the ground truth with a similarity transformation to get the evaluation result of absolute positional error. The trajectory maps of two scenarios in four states are obtained as follows.

Fig. 8 shows the ATE schematic of the library environment obtained in the simulation scenario, and Fig. 9 shows the ATE schematic of the exhibition hall environment. a, b, c, and d Represent the trajectory of cartographer without loop, cartographer, ORB-FSLAM, and HF-SLAM respectively. The dashed line in the figure is the true value trajectory as a reference, and the colored trajectory is the trajectory obtained by the corresponding algorithm.

From Fig. 8(b), it can be visualized that the cartographer trajectory is more confusing and the difference with the real trajectory is larger, even than that without loop, which also proves that the wrong loop closure detection will make the SLAM system completely crash. Also, as shown in Fig. 8(c) and (d), FSLAM produces clearly more accurate trajectories for all those sequences in which they seem to suffer less drift, regardless of the visual feature extraction algorithm used.

Fig. 9 shows that the robot's motion creates a loop. Comparing the subplots, both the cartographer using scan loop closure and the FSLAM system using fused loop closure proposed in this article have less error than the cartographer system without loop, which also reflects that the role of the loop closure optimization is useful.

Tables III and IV show the results of the quantitative analysis of the absolute positional errors, and cartographer_wl means cartographer without loop. Regardless of the algorithm,

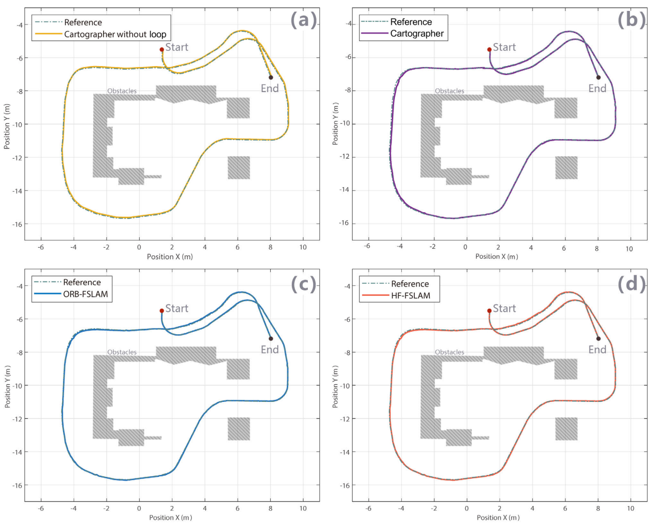


Fig. 9. Comparison of trajectory errors for exhibition hall scene.

TABLE III
ABSOLUTE ERROR IN SHELF ENVIRONMENT (m)

Algorithm	RMSE	Mean	Median	Std	SSE
Cartographer_wl	0.0313	0.0276	0.0247	0.0148	8.5236
Cartographer	0.8434	0.6299	0.5021	0.5608	6165.3799
ORB-FSLAM	0.0241	0.0202	0.0162	0.0132	5.0639
HF-FSLAM	**0.0194**	**0.0170**	**0.0160**	**0.0093**	**3.2626**

TABLE IV
ABSOLUTE ERROR IN EXHIBITION HALL ENVIRONMENT (m)

Algorithm	RMSE	Mean	Median	Std	SSE
Cartographer_wl	0.0424	0.0389	0.0365	0.0170	13.3658
Cartographer	0.0401	0.0352	0.0341	0.0193	11.9643
ORB-FSLAM	0.0227	0.0199	0.0178	0.0108	3.8138
HF-FSLAM	**0.0184**	**0.0154**	**0.0142**	**0.0101**	**2.5059**

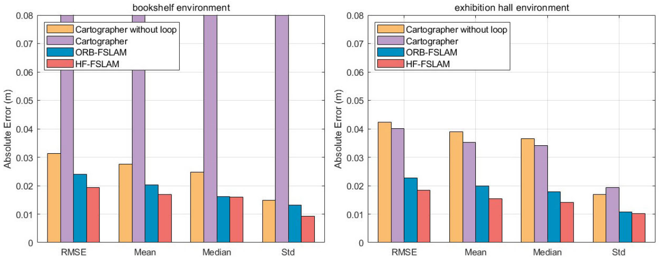


Fig. 10. Absolute error in two environments.

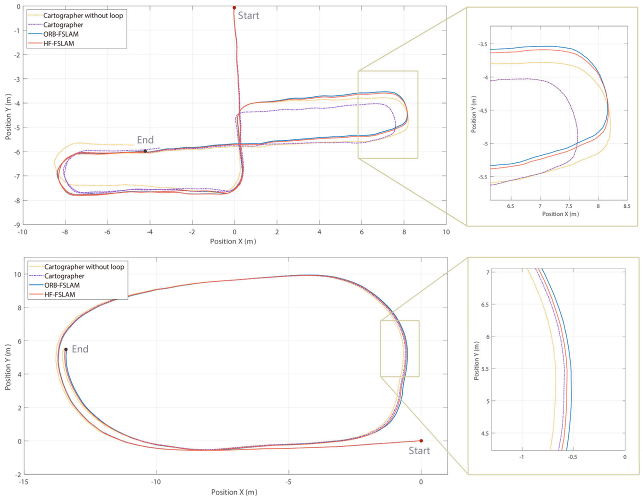


Fig. 11. Trajectory comparison in real environment.

the error is generally higher in the library environment with similar geometry than in the exhibition hall environment, which also implies that scene with similar geometry is a complex structure for SLAM. The less impact from the absence of environmental appearance texture is due to the advantage of the laser sensor, which makes the system have better robustness compared to the pure visual SLAM. Also, Fig. 10 shows the histogram of the quantitative analysis results, from which the accuracy difference of each group of algorithms can be compared more intuitively. In the bookshelf environment, our method improves significantly. In the hall sense, the RMSE of HF-FSLAM improves by 54.1% and 18.9% than cartographer and ORB-FSLAM, respectively. Thus, the FSLAM system designed in this article can significantly reduce the trajectory error and has higher positioning accuracy compared with others.

To demonstrate the feasibility of the algorithm in practice, we also put the algorithm designed in this article in the real environment with the above equipment, and the four trajectories are shown in Fig. 11. In the exhibition hall environment, the trajectories are basically similar because of the obvious structure and more feature points. In the library environment, the experimental results are consistent with what we expect. The pure LiDAR algorithm with loop produces the worst trajectory because of the wrong loop caused by the similar structure. Cartographer is significantly shorter than the other trajectories in the x coordinate, proving that the LiDAR odometry is disturbed by the environment. Because of the accumulated error, the closer to the endpoint, the greater the drift of the trajectory of cartographer without loop. In contrast, the trajectories of ORB-FSLAM and HF-SLAM are maintained at a better level.

C. Results of the Mapping

The mapping experiment is also carried out under four conditions in different environments. The shelf size and the spacing between each shelf in this environment are selected as the standard.

We still select the above four algorithms for mapping comparison, and the same rosbag from the previous set of experiments is applied; the key location data in the final constructed maps are measured in the RVIZ interface and the errors are calculated with the corresponding true values. The mean error is used as an evaluation of the accuracy for mapping.

The mapping results of simulation environments are shown in Fig. 12. The first line is the bookshelf environment, and the second line is the hall environment. For the bookshelf scene,

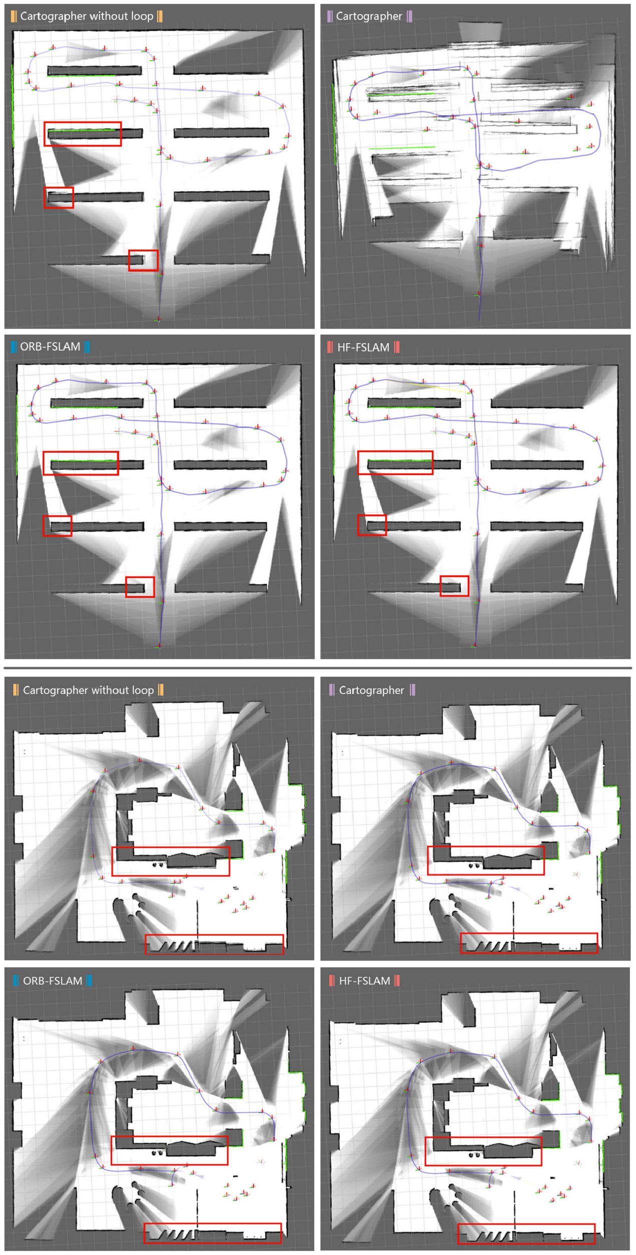


Fig. 12. Mapping results in simulation environments.

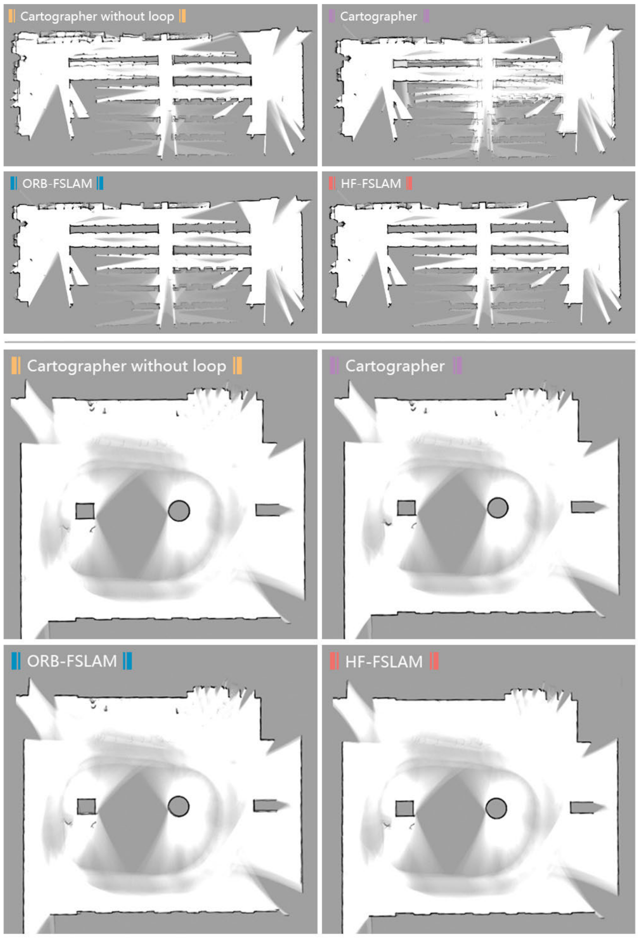


Fig. 13. Mapping results in real environments.

TABLE V
Quantitative Analysis of Mapping Error in Simulation Scene (m)

Algorithm	Shelf size error	Spacing error	Mean error
Cartographer_wl	0.0425	0.0317	0.0371
Cartographer	\	\	\
ORB-FSLAM	0.0373	0.0341	0.0357
HF-FSLAM	**0.0318**	**0.0255**	**0.0287**

the map of cartographer shows serious errors and the map is almost unusable, which demonstrates that the LiDAR-only algorithm is not reliable in the environment with similar geometric structure. In the mapping result of cartographer without loop, the mapping error can be clearly seen in the red box, the sides of the bookcase appear overlapping because the global cumulative error exists. The third picture in Fig. 13 is the ORB-FSLAM result, and the fourth is the HF-FSLAM proposed system in this article. The mapping precisions of two methods are significantly better than the SLAM without loop in the same area, which proves that the system has successfully optimized the map. After observing the mapping result carefully, we find that HF-FSLAM has less blurred edge of the bookshelves than ORB-FSLAM, which certificates the benefits of the deep feature extraction algorithms. For the bookshelf scene, in the area marked in the red box, it can also be seen that the accuracy of the algorithms with loop closure is better than that without loop. The result of FSLAM has clearer edges and better mapping effect than the algorithm without loop, which also verifies the impact of loop closure module on mapping.

The mapping results of real environments are shown in Fig. 13, which is consistent with the performance in the virtual environment. In the real bookshelf environment, HF-FSLAM is the best performing system. The map constructed by cartographer without loop has obvious cumulative error, and cartographer cannot work normally in this environment. In the real hall environment, due to the small area of the site and the obvious structures, the map accuracy of the four SLAM methods has little difference.

Tables V and VI show the results of the mapping accuracy assessment in bookshelf environments. From the data in the

TABLE VI
QUANTITATIVE ANALYSIS OF MAPPING ERROR IN REAL SCENE (m)

Algorithm	Shelf size error	Spacing error	Mean error
Cartographer_wl	0.0465	0.0849	0.0680
Cartographer	\	\	\
ORB-FSLAM	0.0372	0.0353	0.0362
HF-FSLAM	**0.0374**	**0.0260**	**0.0317**

table, we know that the LiDAR loop closure optimization algorithm almost fails in this environment and the map cannot be analyzed quantitatively. The other three categories have higher build accuracy for each algorithm regardless of whether loop closure optimization is being used or not, with errors within 0.05 m. Due to the small range of motion or the lack of measurement discrimination in the reference key locations used for the evaluation, the error ranges of the three algorithms do not differ significantly. Compared with the figures in Tables V and VI, the accuracy of the results in the simulation environment is slightly higher than that in the real environment because the former has fewer interference items. The systems that use loop had less error in map construction than those that do not. HF-FSLAM has the smallest error, which confirms the superiority of our method.

VI. CONCLUSION

LiDAR-based SLAM can obtain accurate structure information, but it lacks texture, which may affect the accuracy of robot localization. LiDAR SLAM algorithms using loop closure detection are highly prone to false detection and cause system crashes, especially in scenarios with similar geometric structures.

In this article, we discuss the causes of the loop closure error and add visual texture information to solve the problem. Besides, a reliable loop closure algorithm that fuses visual and laser data with a tight coupling is constructed. We also explore the differences between traditional extraction algorithms and deep learning visual feature extraction algorithms and verify the superiority of HF-Net via feature extraction and robustness. Finally, a complete SLAM system named FSLAM is designed. We use HF-Net to extract feature points to assist 2-D LiDAR mapping in the library shelf environment. Compared with the ORB method, the proposed method in this study demonstrates that localization accuracy and mapping accuracy can be significantly improved.

For future works, considering that there are a huge number of line features in the library, we only use point features in the loop closure. We will apply line features to promote calibration efficiency and improve the robustness further.

REFERENCES

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–108, Jun. 2006, doi: 10.1109/MRA.2006.1638022.

[2] J. Niu, B. Wang, L. Shu, T. Q. Duong, and Y. Chen, "ZIL: An energy-efficient indoor localization system using ZigBee radio to detect WiFi fingerprints," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 7, pp. 1431–1442, Jul. 2015, doi: 10.1109/JSAC.2015.2430171.

[3] N. Mair and Q. Mahmoud, "A collaborative bluetooth-based approach to localization of mobile devices," in *Proc. 8th IEEE Int. Conf. Collaborative Comput., Netw., Appl. Worksharing*, 2012, pp. 363–371, doi: 10.4108/icst.collaboratecom.2012.250437.

[4] C. Deng, K. Qiu, R. Xiong, and C. Zhou, "Comparative study of deep learning based features in SLAM," in *Proc. 4th Asia–Pacific Conf. Intell. Robot Syst. (ACIRS)*, New York, NY, USA, 2019, pp. 250–254, Accessed: Jan. 17, 2023. [Online]. Available: https://www.webofscience.com/wos/alldb/summary/5695749a-3320-4892-b459-11d37cac5355-6aff5d64/relevance/1

[5] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 34–46, Feb. 2007, doi: 10.1109/TRO.2006.889486.

[6] K. Konolige, G. Grisetti, R. Kuemmerle, B. Limketkai, and R. Vincent, "Efficient sparse pose adjustment for 2D mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, New York, NY, USA, 2010, pp. 22–29, Accessed: Jan. 17, 2023. [Online]. Available: https://www.webofscience.com/wos/alldb/summary/778e41a3-1320-4859-959a-dd82097e36de-6aff654e/relevance/1

[7] S. Kohlbrecher, O. von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot.*, Nov. 2011, pp. 1–6, doi: 10.1109/SSRR.2011.6106777.

[8] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D lidar SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, New York, 2016, pp. 1271–1278, Accessed: Jan. 17, 2023. [Online]. Available: https://www.webofscience.com/wos/alldb/full-record/WOS:000389516201021

[9] A. Huletski, D. Kartashov, and K. Krinkin, "VinySLAM: An indoor SLAM method for low-cost platforms based on the transferable belief model," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, New York, NY, USA, 2017, pp. 6770–6776, Accessed: Jan. 17, 2023. [Online]. Available: https://www.webofscience.com/wos/alldb/summary/c39f0a29-225a-452d-8120-320a6f8a91ee-6aff784d/relevance/1

[10] S. Bouraine, A. Bougouffa, and O. Azouaoui, "Particle swarm optimization for solving a scan-matching problem based on the normal distributions transform," *Evol. Intell.*, vol. 15, no. 1, pp. 683–694, Mar. 2022, doi: 10.1007/s12065-020-00545-y.

[11] K. Krinkin and A. Filatov, "Correlation filter of 2D laser scans for indoor environment," *Robot. Auto. Syst.*, vol. 142, Aug. 2021, Art. no. 103809, doi: 10.1016/j.robot.2021.103809.

[12] C. Debeunne and D. Vivet, "A review of visual-lidar fusion based simultaneous localization and mapping," *Sensors*, vol. 20, no. 7, p. 2068, 2020, doi: 10.3390/s20072068.

[13] F. Sun, Y. Zhou, C. Li, and Y. Huang, "Research on active SLAM with fusion of monocular vision and laser range data," in *Proc. 8th World Congr. Intell. Control Automat. (WCICA)*, New York, NY, USA, 2010, pp. 6550–6554, doi: 10.1109/WCICA.2010.5554412.

[14] 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, doi: 10.3390/s17040802.

[15] J. Graeter, A. Wilczynski, and M. Lauer, "LIMO: lidar-monocular visual odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, New York, NY, USA, 2018, pp. 7872–7879, Accessed: Jan. 17, 2023. [Online]. Available: https://www.webofscience.com/wos/alldb/summary/ebaf4a0d-4980-40b1-a5f3-bce45368fcb5-6b097373/relevance/1

[16] Z. Zhang, R. Zhao, E. Liu, K. Yan, and Y. Ma, "Scale estimation and correction of the monocular simultaneous localization and mapping (SLAM) based on fusion of 1D laser range finder and vision data," *Sensors*, vol. 18, no. 6, p. 1948, Jun. 2018, doi: 10.3390/s18061948.

[17] Y. Zhu, C. Zheng, C. Yuan, X. Huang, and X. Hong, "CamVox: A low-cost and accurate lidar-assisted visual SLAM system," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, New York, NY, USA, May 2021, pp. 5049–5055, doi: 10.1109/ICRA48506.2021.9561149.

[18] Z. Zhu, S. Yang, H. Dai, and F. Li, "Loop detection and correction of 3D laser-based SLAM with visual information," in *Proc. 31st Int. Conf. Comput. Animation Social Agents*, 2018, pp. 1–6, Accessed: Jan. 17, 2023. [Online]. Available: http://dl.acm.org/doi/pdf/10.1145/3205326.3205357

[19] H. Chen, H. Huang, Y. Qin, Y. Li, and Y. Liu, "Vision and laser fused SLAM in indoor environments with multi-robot system," *Assem. Autom.*, vol. 39, no. 2, pp. 297–307, Apr. 2019, doi: 10.1108/AA-04-2018-065.

[20] J. Zhang and S. Singh, “Visual-lidar odometry and mapping: Low-drift, robust, and fast,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Los Alamitos, CA, USA, May 2015, pp. 2174–2181, Accessed: Jan. 17, 2023. [Online]. Available: https://www.webofscience.com/wos/alldb/summary/e8bf1359-643c-4dd2-9222-caa85b3366b2-6aff9742/relevance/1

[21] Y. Seo and C.-C. Chou, “A tight coupling of vision-lidar measurements for an effective odometry,” in *Proc. 30th IEEE Intell. Vehicles Symp. (IV)*, New York, NY, USA, 2019, pp. 1118–1123, Accessed: Jan. 17, 2023. [Online]. Available: https://www.webofscience.com/wos/alldb/summary/3b1ad4f7-fc1a-42fa-9883-445c0b46c1b2-6aff9ccd/relevance/1

[22] J. Lin and F. Zhang, “R3LIVE: A robust, real-time, RGB-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package,” in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 10672–10678, doi: 10.1109/ICRA46639.2022.9811935.

[23] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM: A versatile and accurate monocular SLAM system,” *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015, doi: 10.1109/TRO.2015.2463671.

[24] R. Mur-Artal and J. Tardós, “ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras,” *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017, doi: 10.1109%2FTRO.2017.2705103.

[25] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel, and J. D. Tardos, “ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM,” *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021, doi: 10.1109/TRO.2021.3075644.

[26] S. Arshad and G.-W. Kim, “Role of deep learning in loop closure detection for visual and lidar SLAM: A survey,” *Sensors*, vol. 21, no. 4, p. 1243, Feb. 2021, doi: 10.3390/s21041243.

[27] Y. Hou, H. Zhang, and S. Zhou, “Convolutional neural network-based image representation for visual loop closure detection,” in *Proc. IEEE Int. Conf. Inf. Autom.*, New York, NY, USA, 2015, pp. 2238–2245, Accessed: Jan. 17, 2023. [Online]. Available: https://www.webofscience.com/wos/alldb/summary/362c9cb6-4b2e-4fe1-ab1b-6662536a7d0b-6b009fbc/relevance/1

[28] Z. Zhu, X. Xu, X. Liu, and Y. Jiang, “LFM: A lightweight LCD algorithm based on feature matching between similar key frames,” *Sensors*, vol. 21, no. 13, p. 4499, Jun. 2021, doi: 10.3390/s21134499.

[29] D. DeTone, T. Malisiewicz, and A. Rabinovich, “SuperPoint: Self-supervised interest point detection and description,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, New York, NY, USA, Jun. 2018, pp. 337–349, doi: 10.1109/CVPRW.2018.00060.

[30] J. Tang, J. Folkesson, and P. Jensfelt, “Geometric correspondence network for camera motion estimation,” *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 1010–1017, Apr. 2018, doi: 10.1109/LRA.2018.2794624.

[31] J. Tang, L. Ericson, J. Folkesson, and P. Jensfelt, “GCNV2: Efficient correspondence prediction for real-time SLAM,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3505–3512, Jul. 2019, doi: 10.1109/LRA.2019.2927954.

[32] D. Li et al., “DXSLAM: A robust and efficient visual SLAM system with deep features,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, New York, Oct. 2020, pp. 4958–4965, doi: 10.1109/IROS45743.2020.9340907.

[33] E. B. Olson, “Real-time correlative scan matching,” in *Proc. IEEE Int. Conf. Robot. Autom.*, New York, NY, USA, May 2009, pp. 1233–1239. [Online]. Available: https://www.webofscience.com/wos/alldb/summary/43aa4d72-c454-4489-a4cb-c9a58cc49e29-6b00b3ff/relevance/1

[34] Q. Zhang and R. Pless, “Extrinsic calibration of a camera and laser range finder (improves camera calibration),” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2005, doi: 10.1109/IROS.2004.1389752.

[35] D. Galvez-López and J. D. Tardos, “Bags of binary words for fast place recognition in image sequences,” *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012, doi: 10.1109/TRO.2012.2197158.

[36] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, New York, NY, USA, Nov. 2011, pp. 2564–2571, doi: 10.1109/iccv.2011.6126544.

[37] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, “From coarse to fine: Robust hierarchical localization at large scale,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New York, Jun. 2019, pp. 12708–12717, doi: 10.1109/CVPR.2019.01300.

[38] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of RGB-D SLAM systems,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580, doi: 10.1109/IROS.2012.6385773.

[39] A. Pronobis and B. Caputo, “COLD: The CoSy localization database,” *Int. J. Robot. Res.*, vol. 28, no. 5, pp. 588–594, May 2009, doi: 10.1177/0278364909103912.

[40] B. Glocker, S. Izadi, J. Shotton, and A. Criminisi, “Real-time RGB-D camera relocalization,” in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, New York, NY, USA, 2013, pp. 173–179, Accessed: Jan. 17, 2023. [Online]. Available: https://www.webofscience.com/wos/alldb/summary/41acfd20-f556-4a31-8b1a-995bb100a995-6b00f38e/relevance/1

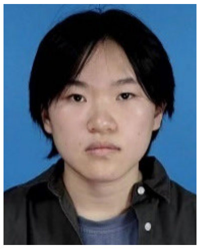
[41] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, “A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, New York, NY, USA, May 2014, pp. 1524–1531, doi: 10.1109/icra.2014.6907054.

Zongkun Zhou received the B.Eng. degree in surveying and mapping engineering from the China University of Mining and Technology, Xuzhou, China, in 2017, and the M.S. degree in guidance navigation and control from Wuhan University, Wuhan, China, in 2021, where he is currently pursuing the Ph.D. degree with the GNSS Research Center.

He is currently a student under the supervision of Prof. Weiping Jiang and Prof. Chi Guo. His areas of interest include lidar SLAM, data fusion, and GNSS/INS/lidar integration for vehicle navigation.

Chi Guo (Member, IEEE) received the Ph.D. degree in computer science from Wuhan University, Wuhan, Hubei, China, in 2010.

He is currently a Professor with the National Satellite Positioning System Engineering Technology Research Center, Wuhan University. His current research interests include BeiDou application, unmanned system navigation, and location-based services (LBS).



Yanyue Pan received the bachelor’s degree in computer science from Wuhan University, Hubei, China, in 2020, and the M.Eng. degree from the Artificial Intelligence Institute, Wuhan University, in 2022.

Her main research interests include vision navigation and deep learning.

Xiang Li received the bachelor’s degree in computer science from Wuhan University, Wuhan, Hubei, China, in 2021, where he is currently pursuing the M.Eng. degree with the Artificial Intelligence Institute.

His main research interests include lidar SLAM.

Weiping Jiang received the Ph.D. degree (Hons.) in geodesy and engineering surveying from the School of Geodesy and Geomatics, Wuhan University, Wuhan, China, in 1997.

He is currently a Professor with Wuhan University. His main research interests include GNSS data processing of large-scale networks, GNSS coordinate time series analysis, satellite altimetry, and geodynamics research.