NALO-VOM: Navigation-Oriented LiDAR-Guided Monocular Visual Odometry and Mapping for Unmanned Ground Vehicles

Ziqi Hu, Jing Yuan, *Member, IEEE*, Yuanxi Gao, Boran Wang, and Xuebo Zhang, *Senior Member, IEEE*

***Abstract*—Monocular visual odometry (VO) is a fundamental technique for unmanned ground vehicle (UGV) navigation. However, traditional monocular VO methods always suffer from sparse environment maps which cannot be directly used for navigation because of the lack of structural information. In this article, we propose a navigation-oriented LiDAR-guided monocular visual odometry and mapping (NALO-VOM) to obtain scale-consistent camera poses and a semi-dense environment map which is more suitable for navigation of UGVs. The structure representation ability of the 3D LiDAR point cloud is learned by a major-plane prediction network and then transferred into the monocular VO system in NALO-VOM. As a result, NALO-VOM can construct a more dense and high-quality map using only a monocular camera. To be specific, the major-plane prediction network is trained offline using 3D LiDAR geometric information, which predicts major-plane mask (MP-Mask) for each frame of the visual image during the localization. Then, MP-Mask is used for scale optimization and semi-dense map building. Experiments are performed on the public dataset and self-collected sequences. The results show the competitive performance on the localization accuracy and mapping quality compared with other visual odometry methods.**

***Index Terms*—Navigation-oriented visual odometry, semi-dense map building, unmanned ground vehicles.**

I. Introduction

MONOCULAR cameras are indispensable for some environment perception tasks as cameras can capture texture information from the environment, and are flexible to be set up on autonomous vehicles. Therefore, monocular visual odometry (VO) has attracted increasing attention in the fields of automation driving [1], [2]. To be specific, the monocular VO plays an important role in the visual simultaneous localization and mapping (vSLAM) by providing an accurate state estimate. However, the environment map built by the traditional monocular VO is merely used to assist estimating camera poses [3], [4], instead of representing the structural information of the environment. As a result, the environment map built by the traditional monocular VO is usually too sparse to be used in UGV navigation. In order to suit a monocular VO method to UGV navigation, a navigation-oriented monocular VO should have three basic characteristics. First, it can provide accurate state estimate of the UGV. This is a general requirement for an odometry which can be achieved with a single inertial measurement unit (IMU) and a global positioning system (GPS) module [5], [6], [7], [8], when visual information is not available. Second, a environment map can be simultaneously built during the localization process. Third, a navigation-oriented monocular VO should build a dense enough map that can be transformed into a 2D grid map or 2.5D elevation map, which is used for motion planning and navigation.

In the current monocular VO frameworks, both indirect methods [9], [10] and direct methods [3], [11] require sufficient changes on the pixel gradient of map points, resulting in few map points can be obtained in the texture-less areas (e.g., white walls and grounds). Moreover, the environment map (or the feature map) constructed by indirect methods is too sparse to contain sufficient structural information. In such a situation, the UGV cannot obtain an available navigation map for motion planning and decision making. To this end, some monocular dense SLAM works [12], [13], [14] proposed to integrate a depth prediction network into the VO or SLAM system. In this way, pixel-wise depths are estimated and a dense environment map is constructed. However, the depth prediction network suffers from high computational complexity and the map accuracy strongly relies on the network performance in specific environments. Therefore, these methods are difficult to apply to UGV systems in other environments whose appearances are largely different from training datasets.

Inspired by the 3D LiDAR point cloud which can precisely describe the accessible regions around the UGV, we propose a navigation-oriented LiDAR-guided monocular visual odometry and mapping, named NALO-VOM. The geometric structure representation ability of LiDARs is learned and transferred to the monocular VO, such that a navigation-oriented map can

Manuscript received 10 June 2023; revised 20 July 2023; accepted 29 July 2023. Date of publication 14 August 2023; date of current version 23 February 2024. This work was supported in part by the Natural Science Foundation of China under Grants U21A20486 and 62073178, in part by the Tianjin Science Fund for Distinguished Young Scholars under Grant 20JCJQJC00140, in part by the major basic Research Projects of the Natural Science Foundation of Shandong Province under Grant ZR2019ZD07, and in part by the Tianjin Natural Science Foundation under Grant 20JCYBJC01470. *(Corresponding author: Jing Yuan.)*

The authors are with the College of Artificial Intelligence, Nankai University, Tianjin 300350, China, also with the Tianjin Key Laboratory of Intelligent Robotics, Nankai University, Tianjin 300350, China, and also with the Engineering Research Center of Trusted Behavior Intelligence, Ministry of Education, Nankai University, Tianjin 300350, China (e-mail: huziqi@mail.nankai.edu.cn; nkyuanjing@gmail.com; gaoyuanxi@mail.nankai.edu.cn; wangbr@mail.nankai.edu.cn; zhangxuebo@nankai.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2023.3303355.

Digital Object Identifier 10.1109/TIV.2023.3303355

be built with only a monocular camera. Specifically, a major-plane prediction network is trained offline with non-artificial labels produced by camera-LiDAR pairs. Then, the resultant network is integrated into the monocular VO system to predict a major-plane mask (MP-Mask) in each image frame. Based on the visual stereo matching method, different sizes of major-planes are estimated using MP-Mask to construct a dense and accurate environment map. Moreover, the ground is extracted from the major-planes to constrain the trajectory scale. The contributions of this article can be summarized as follows:

- We propose a navigation-oriented LiDAR-guided monocular visual odometry system. To our best knowledge, it is the first time to transfer the structure representation ability of the 3D LiDAR to monocular visual odometry with the major-plane prediction network. It is worth noting that the proposed monocular VO system only uses a monocular camera to construct a high-quality map that can be used for motion planning and decision making.
- MP-Mask is estimated from the major-plane prediction network to help achieve dense front-end tracking and extract the ground plane for constraining the scale drift. To our best knowledge, it is the first time to incorporate the major-plane into the monocular VO system to optimize the scale and build a denser map. As a result, accuracy of the VO and density of the map can be largely improved.
- We perform experiments to evaluate the proposed method on the public dataset and real-world outdoor sequences. The results show superior localization performance compared to state-of-the-art visual VO methods and a denser map can be obtained which is more convenient for UGV navigation.

II. Related Work

A. Monocular Visual SLAM

The monocular visual SLAM can be categorized into indirect and direct methods. The indirect method (feature-based method) extracts significant features from images to perform data association and represent an environment map. In order to ensure the accuracy of data association, features (e.g. FAST [15], SIFT [16], SURF [17], and ORB [18]) should have strong distinctiveness in the feature space. Therefore, the indirect methods [9], [10] only build a sparse map, which can't describe the geometrical structure of the environment very well. Such a sparse feature-point map cannot be used for motion planning and navigation of UGVs. Compared to the indirect method, the direct method relaxes the feature selection criteria, which directly uses the pixels with large gradients to estimate depths, and then builds a denser map. Despite the enhancement of map representation ability compared to the indirect methods, the existing direct methods [3], [11] still fail to tackle texture-less areas (e.g., walls and grounds), which remains challenging. On the other hand, both indirect and direct methods suffer from scale drift in large-scale environments. To solve this problem, Wang et al. [19] proposed a novel VO system which takes the ground plane model into consideration as feedback when performing ground detection to obtain high-accuracy ground geometric information. After that, [19] performed scale recovery based on the ground plane and obtained low-drift results during long-term localization.

B. Multi-Sensor Fusion SLAM

The LiDAR plays an important role in the mapping process of multi-sensor fusion SLAM, since the visual map points are less accurate than those of the LiDAR. Radmanesh et al. [20] proposed a LiDAR-visual SLAM system to improve localization accuracy and mapping performance by fusing LiDAR and visual information based on an unsupervised object discovery method. Shao et al. [21] proposed a system that fused stereo visual and LiDAR information, which improved the localization performance of the LiDAR under degenerate scenarios with the assistance of stereo vision. However, the visual information was only used for localization, which means map building still relies on LiDARs. Lin et al. [22] proposed a real-time LiDAR-inertial-visual tightly-coupled SLAM system, using an error-state iterated Kalman filter and modified factor graph optimization to achieve real-time localization and dense map building. However, the denseness of the map is mainly contributed by the LiDAR information. If the LiDAR happens to break down, the camera cannot construct the environment map solely [20], [21], [23].

C. Monocular Dense vSLAM

In recent years, depth prediction networks are adopted in some monocular dense SLAM systems [24], [25]. They predict a depth map for the input image. The resultant depth map is used to construct a dense environment map. Tateno et al. [12] fused the convolutional neural networks (CNN) for depth estimation [26] and semantic segmentation network [27] into LSD-SLAM [11] to build a global dense map and recover monocular scale. Czarnowski et al. [28] proposed a deep-learning-based SLAM system in a probabilistic framework. By integrating learned priors over geometry with classical SLAM formulations, the system was capable of building a global dense environment map. Koestler et al. [29] fused the depth map predicted by a multi-view stereo network (MVSNet) into the global map using the truncated signed distance function (TSDF) method. Yokozuka et al. [30] utilized extremely dense feature points to estimate camera poses and mesh the sparse map points into the global map using the non-local total generalized variation (NLTGV) method [31]. Generally, dense mapping requires as accurate as possible depth prediction on each pixel. However, for UGV navigation, the scenarios are normally variable and complex, which is challenging for generalization of the depth prediction network. And to some extent, existing depth prediction networks cannot meet the accuracy requirement for building a navigation map. Meanwhile, a large amount of depth ground truth is required for training, however, in outdoor scenarios, obtaining such ground truth still remains an issue [32].

III. System Overview

Fig. 1 depicts the overview of the proposed system. The system is divided into two parts which are offline training

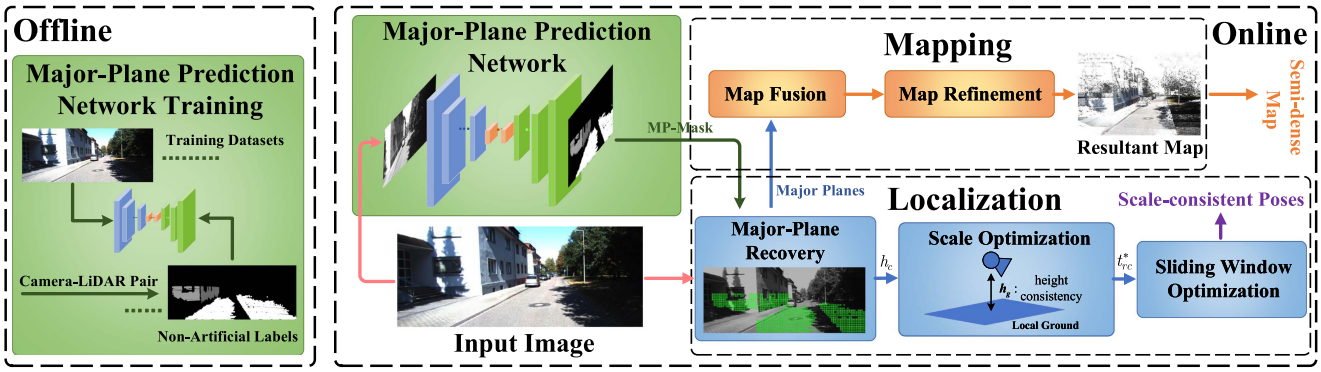

Fig. 1. Overview of the proposed system.

and online odometry process. During the offline training, a monocular camera and a 3D LiDAR are running synchronously to produce non-artificial labels which are used to train the major-plane prediction network. During the online odometry process, only the monocular camera is used. The RGB image and MP-Mask from the major-plane prediction network are fed to the localization module to perform major-plane recovery. After major-plane recovery, the ground height is extracted to constrain the relative scale in scale optimization which then feeds the optimized pose of the current keyframe to sliding window optimization. In the end, the scale-consistent camera poses are estimated as the system outputs. The major-planes estimated by localization module are fed to the mapping module to perform map fusion and refinement, which then construct a navigation-oriented map.

IV. Major-Plane Prediction

The major-plane refers to an area that satisfies three conditions: 1) possessing highly uniform planar curvature everywhere; 2) occupying a big enough area; 3) presenting good connectivity when projected onto the image plane. Major-planes contain abundant geometry information of the environment, especially for most texture-less areas (e.g., ground and white walls). With this additional information, texture-less areas can also be used in the tracking and mapping process without the need for sufficient features. Different from the VO/SLAM systems that extract planes using depth information provided by RGB-D sensors [33], [34], [35], extracting major-planes from a solo RGB image remains unresolved [36], [37], [38]. Since lack of 3D depth information makes it hard to distinguish between 3D planes at different distances. In contrast, it is easier to extract major-planes from LiDAR point clouds, since point clouds contain more structural information. Therefore, we use LiDAR point clouds to guide major-plane extraction from an image by embedding point cloud geometry information into a major-plane prediction network. The network structure proposed in [39] is adopted in the major-plane prediction network. The training loss $E(g)$ is defined as

$$E(g) = \frac{1}{c}\sum_{i=0}^{h}\sum_{j=0}^{w} g_{(i,j)}^2 - \frac{\lambda}{c^2}\left(\sum_{i=0}^{h}\sum_{j=0}^{w} g_{(i,j)}\right)^2, \quad (1)$$

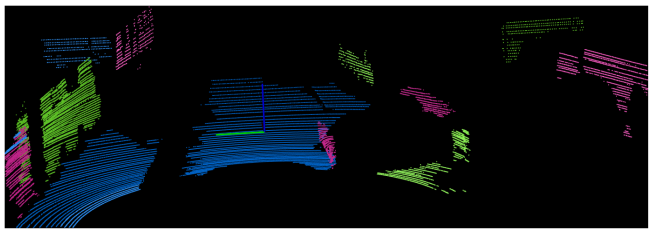

Fig. 2. Major-planes extracted from the point cloud map.

where $g_{(i,j)} = \log \tilde{s}_{(i,j)} - \log s_{(i,j)}$ with the gray scale $\tilde{s}_{(i,j)}$ and $s_{(i,j)}$ at the location (i, j) on the predicted image and the label image, respectively. h and w denote the height and width of the image, respectively. c denotes the number of pixels on the image and $\lambda = 0.7$. In fact, $E(g)$ is a sum of the variance and a weighted squared mean of the error in log space [39].

Noting that, the artificial depth labels used in the depth prediction network [39] are not suitable for the LiDAR structural prior learning in the major-plane prediction network. Therefore, we propose a camera-LiDAR pairing self-labeled method to generate projection labels for the major-plane prediction network training. The major-plane prediction network is trained offline using a set of images and point clouds collected by the well-calibrated camera and 3D LiDAR, respectively. Major-planes are first extracted from each scan of the point cloud obtained by the LiDAR. In order to ensure the precision and recall of major-planes, region-growing segmentation is firstly applied to the point cloud to find as many as possible plane regions in the environment. Secondly, each plane region is optimized using the random sample consensus (RANSAC) method to reject outliers and improve the accuracy of the plane parameters. Despite that not all the planes are extracted, the major-planes represent most texture-less areas (e.g. roads and buildings) of the environment, as shown in Fig. 2. Moreover, merely estimating major-planes reduces the difficulty in network training and improves its generalization performance.

The extracted major-plane point clouds are projected onto the image plane. The projections are labeled as the positive pixels $\mathbf{p}'_i = [x, y]^{\mathrm{T}}$ which are then assigned with the gray scale of $I(\mathbf{p}'_i) = 255 \cdot n_{\mathbf{p}'_i}/n_{\max}$, where $n_{\mathbf{p}'_i}$ and $n_{\max}$ are the point numbers of the point cloud containing $\mathbf{p}'_i$ and the largest point

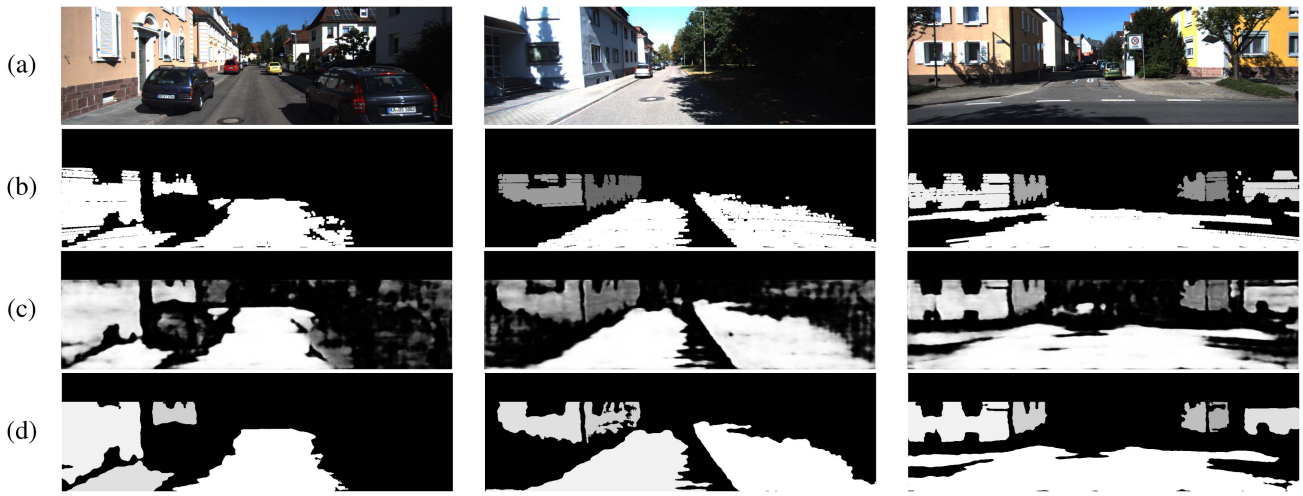

Fig. 3. LiDAR point cloud is projected onto the image plane, which is then used as the training label after upsampling. (a) The input images, (b) the upsampled point cloud projected images which are used as the training labels, (c) the prediction results of the network, (d) the refined prediction results which are used as MP-Mask in NALO-VOM.

cloud, respectively. The remaining pixels except the positive ones are assigned with the gray scale of 0. However, only a few pixels can be associated with the projections due to the sparseness of point clouds, which could make the network training difficult to converge. To this end, the projected image is upsampled to make positive pixels denser. In the upsampling process, the pixels belonging to the same major-plane area are assigned with the same gray scale value. The upsampled images are taken as training labels in the major-plane prediction network training. Note that, during the VO process, the predicted MP-Mask is refined by removing small regions and normalizing the pixel values of each major-plane area. To be specific, firstly, dilation and erosion are applied to the MP-Mask to eliminate small outlier regions caused by the noises of the major-plane prediction network. Then, the regions of which the sizes are smaller than a threshold $s_{\text{area}} = s_{\max}/\eta$ are further removed as the outlier regions, where $s_{\max}$ is the size of the largest major-plane area, η is a constant ($\eta = 8$ in the experiment). Finally, the pixels within the same region are normalized to the same gray scale. Fig. 3 shows some samples of input images, upsampled projected images, prediction results of the network, and MP-Mask. Note that, the shadows caused by cars and trees can affect the continuity of the ground plane by affecting the texture consistency, as shown in Fig. 3(a). On the other hand, the ground plane can also be affected by the labels of training data. To be specific, the labels obtained by the 3D LiDAR may have some flaws, since the 3D LiDAR cannot receive the strong reflection signal on some spots of the ground caused by some external factors (e.g., surface material, occlusions, etc.). As a result, small vacant regions may appear on the corresponding areas of labels. However, such discontinuity has little effect on localization performance of NALO-VOM, since the edge and shape information of the ground plane is not used in relative scale optimization and dense tracking process.

Note that, although the plane-assisted VO/SLAM systems have been intensively studied, they only utilize the ground plane obtained by the depth prediction network as the extra geometric information to improve the accuracy of localization. Different from [19], [40], [41], [42], which only used the ground plane, the major planes in NALO-VOM contain not only the ground plane, but also some other planes, such as walls and texture-less areas. Accordingly, the geometrical information used in relative scale optimization and dense tracking are provided by all sorts of major planes. The major-plane plays an important role not only in high-accuracy localization, but also in the semi-dense mapping process.

V. Visual Localization and Navigation-Oriented Map Building

The major-plane prediction network obtained by offline LiDAR-guided training is integrated into the monocular VO system to provide the system with the structure representation ability of LiDAR. NALO-VOM only takes RGB images as the inputs (i.e., the system performs as a monocular visual odometry system). At the frontend, NALO-VOM uses DSO [3] for dense tracking. Since the monocular VO cannot keep the scale stable, the ground is extracted using MP-Mask to constrain the relative scale. Then, the navigation-oriented map is constructed with the help of MP-Mask.

A. Dense Front-End Tracking

In DSO, the depth map D_p is propagated to the reference frame from older keyframes in a sliding window. However, D_p is sparse, and thus cannot provide a robust estimation when applying direct image alignment between the reference and the current frame. TANDEM [29] proposed to incorporate the TSDF-rendered depth map to utilize more interframe data association. Compared to TANDEM, we directly add more pixels with depths d_p^* to the front-end tracking using MP-Mask. Specifically, pixels of D_p on the same plane are back-projected to generate 3D points, and then they are used to fit the plane parameter $\boldsymbol{\pi} = [\vec{\mathbf{n}}^{\mathrm{T}}, \sigma]^{\mathrm{T}}$, where $\vec{\mathbf{n}} \in \mathbb{R}^3$ is the unit normal of the plane, $\sigma \in \mathbb{R}$ is the vertical distance from the origin to the

plane. The newly added depth d_p^* is given by

$$d_p^* = \vec{\mathbf{n}}^{\mathrm{T}} \cdot \mathbf{K}^{-1} \cdot \mathbf{p} \cdot \sigma^{-1}, \tag{2}$$

where $\mathbf{K}$ is the camera intrinsic parameter matrix, and $\mathbf{p}$ is the homogeneous pixel coordinate. Compared with the depth map directly obtained by DSO, the denser depth map in NALO-VOM yields more robust and accurate image alignments.

B. Scale Optimization

Scale drift is a common problem for long-term monocular VO. Scale priors can't be propagated to the current frame when a large change of the visual angle occurs, since a large number of old 3D points cannot be projected onto the current frame. In order to keep the scale consistent, an assumption is made that the vertical distance between the camera and the ground is constant. In the city scenarios, the assumption is valid since most grounds in the city are locally flat. And for most navigation tasks, the rigid transformation from the camera coordinate system to the UGV coordinate system is invariant. By constraining the distance between the origin of the camera coordinate system and the local ground to be a constant h_g, scale consistency can be achieved.

Scale refinement is performed after the sliding window optimization. The scale constraint ρ_c is imposed on the current keyframe f_c by $\rho_c = h_g/h_c$, where h_c is the ground height estimated in f_c. ρ_c represents the scale ratio between the fixed scale and the relative scale of f_c. The translation t_{cr}^* between the current frame f_c and the reference frame f_r is updated by $t_{cr}^* = \rho_c \cdot t_{cr}$. Because of the estimation error of h_c, the sliding window optimization is applied to update the poses and inverse depths of keyframes in the window. The loss function is defined by

$$\mathcal{L} = \underset{\xi_r, \xi_t, \Theta_{id}}{\arg\min} \sum_{r=1}^{W} \sum_{t=1}^{W} \sum_{\mathbf{p} \in \mathcal{N}_r} \omega_{\mathbf{p}} ||I_t(\mathbf{p}') - I_r(\mathbf{p})||_\gamma, \tag{3}$$

$$\omega_{\mathbf{p}} = \frac{m^2}{m^2 + ||\nabla I_r(\mathbf{p})||_2^2}, \tag{4}$$

where $\xi_r \in \mathfrak{se}(3)$ and $\xi_t \in \mathfrak{se}(3)$ represent the poses of the reference frame and target frame ($\mathbf{p}$ is observed in a target frame), respectively. Θ_{id} is the set of inverse depths of active pixels (i.e., the pixels with convergent inverse depths) in the window. W is the length of the sliding window. $\mathcal{N}_r$ is the set of active pixels in the frame f_r. I_r and I_t are the pixel values of $\mathbf{p}$ in the frame f_r and f_t, respectively, and $||\cdot||$ is the Huber norm. m is a constant. $\mathbf{p}'$ stands for the projected point position of $\mathbf{p}$ with the inverse depth d_p, given by $\mathbf{p}' = z^{-1} \cdot \mathbf{K} \cdot \mathbf{P}_t$, with

$$\mathbf{P}_t = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R}(d_p^{-1}\mathbf{K}^{-1}\mathbf{p}) + \mathbf{t}, \tag{5}$$

$$\mathbf{T}_{tr} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}, \tag{6}$$

where $\mathbf{T}_{tr} \in \mathrm{SE}(3)$ is the transformation matrix of a point from f_r to f_t.

The ground is selected from major-planes according to the direction, distance to the origin, and amount of pixels covered by the plane. Because the ground plane always occupies a large portion of the image, the number of pixels representing the ground in MP-Mask is utilized to rapidly and roughly distinguish the ground from other surfaces. In cases where the ground plane contains insufficiant pixels due to occlusions, the current frame is excluded during the scale optimization process. Without loss of generality, we assume the *x*-*z* plane of the camera frame is parallel to the ground. The normal vector direction of the ground needs to be as same as the direction of the *y* axis of the camera frame, and the distance from the origin of the camera frame to the ground needs to be as far as possible. We use the method in Section V. A to obtain the ground parameters. Then, we fix h_g once the ground parameters converge.

C. Map Reconstruction

The map reconstruction is based on the inverse depth estimation of each tracking point. To be specific, the tracking point $\mathbf{p}$ on the reference frame f_r is projected onto the current frame f_c to find the matching point $\mathbf{p}_{\mathrm{match}} = [u_m, v_m]^{\mathrm{T}}$ on the epipolar line. Then the inverse depth $d_{\mathbf{p}}$ of $\mathbf{p}$ can be obtained by

$$d_{\mathbf{p}} = \frac{x_{\mathbf{p}_c} - u_m z_{\mathbf{p}_c}}{u_m z_{\mathbf{t}} - x_{\mathbf{t}}}, \tag{7}$$

where $[x_{\mathbf{p}_c}, y_{\mathbf{p}_c}, z_{\mathbf{p}_c}]^{\mathrm{T}} = \mathbf{KRK^{-1}p}$, $[x_{\mathbf{t}}, y_{\mathbf{t}}, z_{\mathbf{t}}]^{\mathrm{T}} = \mathbf{Kt}$, $\mathbf{K}$ is the camera intrinsic parameter matrix, $\mathbf{R}$ and $\mathbf{t}$ are the rotation matrix and translation vector from f_r to f_c, respectively. Since the noise exists in the estimation of $d_{\mathbf{p}}$, the uncertainty range $(d_{\mathbf{p}}^{\min}, d_{\mathbf{p}}^{\max})$ of $d_{\mathbf{p}}$ is given by

$$d_{\mathbf{p}}^{\min} = \frac{x_{\mathbf{p}_c} - (u_m + \sigma_\lambda) z_{\mathbf{p}_c}}{(u_m + \sigma_\lambda) z_{\mathbf{t}} - x_{\mathbf{t}}}, \tag{8}$$

$$d_{\mathbf{p}}^{\max} = \frac{x_{\mathbf{p}_c} - (u_m - \sigma_\lambda) z_{\mathbf{p}_c}}{(u_m - \sigma_\lambda) z_{\mathbf{t}} - x_{\mathbf{t}}}, \tag{9}$$

where the uncertainty σ_λ is calculated using the method proposed in [43].

In existing dense SLAM methods, reconstruction of textureless areas always requires the pixel-wise depth map predicted by a deep network [12], [29]. They rely on the precision of the network which needs a large amount of training data. Unlike the methods mentioned above, we introduce major-planes into the map building to fill the voids in the sparse feature map. Specifically, the pixels in f_n indicated by MP-Mask are selected evenly and assigned with depths calculated by (2). These pixels are then back-projected to Cartesian space to generate 3D point cloud $\Psi_n = \{{}^n\boldsymbol{\pi}_k | k = 1, 2, \ldots, N\}$, where ${}^n\boldsymbol{\pi}_k$ represents for the kth major-plane. The sparse point cloud map generated by DSO is given by M_s. M_s is built by back-projecting active pixels to Cartesian space to generate 3D points on each keyframe using D_p. Due to the sparse characteristic of M_s, it cannot fully represent the environment structure. To this end, NALO-VOM constructs a navigation-oriented map by combining the sparse map M_s and major-planes. ${}^n\boldsymbol{\pi}_k$ is optimized by aligning to the

segmented plane ${}^n\boldsymbol{\pi}_k^{DSO} \in M_s$, where ${}^n\boldsymbol{\pi}_k^{DSO}$ is the nearest plane to ${}^n\boldsymbol{\pi}_k$ in M_s. The loss function of ${}^n\mathcal{L}_k$ is given by

$$\mathcal{L}_k = \sum_{\mathbf{p}\in {}^n\boldsymbol{\pi}_k} \frac{||{}^n\boldsymbol{\pi}_k^{DSO^{\mathrm{T}}} \cdot \mathbf{p}||_2}{{}^nN_k}, \tag{10}$$

where ${}^nN_k$ is the point number of ${}^n\boldsymbol{\pi}_k$. Then, the resultant major-planes are integrated with M_s to create an environment map, which is much denser than M_s. Also, the constructed map fully represents the environment structure both in the texture-less areas and the fine-grained areas. It can be transformed into a 2D grid map or 2.5D elevation map used for UGV navigation.

VI. Experiments and Results

We evaluate the performance of the proposed NALO-VOM on the public dataset KITTI [44] and self-collected sequences. The KITTI dataset has 11 sequences that are provided with ground truth using GPS/IMU localization unit [45]. In this article, we compare our system with seven state-of-the-art monocular VO and vSLAM systems, including DSO [3], ORB-SLAM2 [9], VISO [46], and some other VO methods which contain ground plane estimation and scale optimization [19], [40], [41], and [42]. Loop closure and relocalization are disabled in ORB-SLAM2 for fairness. The real-world experiments are conducted on self-collected sequences. NALO-VOM runs on a laptop with an Intel i7-9700 K CPU, and an NVIDIA RTX 3070 with 8 GB VRAM.

A. Evaluation Metrics

The metrics used to evaluate all seven methods are:

Localization Accuracy: The localization accuracy is evaluated with the metrics δ_{rot} and δ_{trans} given by KITTI [45] representing the error of the rotation and translation of camera poses, respectively. The estimated trajectory is scaled aligning to the ground truth because the absolute scale is not available in the monocular VO.

Mapping Quality: Two metrics are used for evaluating the mapping quality, including the map similarity and the point cloud density. Furthermore, the estimated point cloud maps are transformed into 2D navigation grid maps. And the Hybrid A* path planning algorithm [47] is performed on the 2D navigation grid map to verify whether the estimated point cloud map is suitable for UGV navigation.

- $\mathcal{S}_{map}^{\lambda}$ measures the map similarity between the estimated map M_{est} and the map M_{lidar} built by LiDAR-based SLAM [48]. $\mathcal{S}_{map}^{\lambda}$ is given by calculating the proportion of the point $\mathbf{p} \in M_{lidar}$ which satisfies $d(\mathbf{p}, M_{est}) < \lambda d(\mathbf{p}, M_{lidar})$, where $d(\cdot)$ gives the minimal distance between the point and the map (i.e., the nearest point in the map) and λ is a ratio factor. The bigger $\mathcal{S}_{map}^{\lambda}$ is, the more similar the distribution of points in M_{est} is to that in M_{lidar}. Note that, we compare M_{est} with M_{lidar} after alignment between them, since M_{est} does not include absolute scale information.
- The point cloud densities $\varphi_1, \varphi_2 \ldots \varphi_7$ measure the number of points in different objects (i.e., roads, terrains, cars, buildings, sidewalks, fences, and vegetation in KITTI). φ is given by calculating the point ratio between M_{est} and the ground truth map M_{gt} obtained by the point cloud segmentation provided by KITTI. M_{gt} is built using the ground truth poses and point cloud provided by KITTI.
- In order to further verify the quality of the obtained point cloud maps, they are transformed into 2D navigation grid maps. To be specific, the map points are projected down to the ground plane along with the normal direction of the ground plane. The grids cells occupied by the projections are defined as the obstacle cells, and the rest cells are defined as the accessible space. It is worth noting that, not all the map points are selected to be projected onto the ground plane. In the KITTI dataset, the map points with the vertical distance $d_{\text{height}} \in [0.1, 3]$ from the ground plane are selected. On the one hand, the map points belonging to the ground are removed, since the ground should not be considered as the obstacles in UGV navigation. On the other hand, considering the actual height of the UGV, the points distributed in the space which is much higher than the UGV are not considered as the obstacles as well.

TABLE I
$\mathcal{R}_f$ of MP-Mask on the KITTI Dataset

	00	01	02	03	04	05
$\mathcal{R}_f$	89.97%	93.74%	90.20%	90.39%	93.36%	97.90%
	06	07	08	09	10	
$\mathcal{R}_f$	98.64%	93.82%	91.18%	86.05%	93.59%	

B. Major-Plane Prediction Network

For the implementation of the major-plane prediction network, we use ResNet-50 [49] as the encoder part of the major-plane prediction network, whose parameters are pre-trained on the ILSVRC dataset [50]. We use Adam [51] as the optimizer with $\beta_1 = 0.92$, $\beta_2 = 0.99$, and $\epsilon = 10^{-6}$. In order to achieve a convergent optimization, the initial learning rate is set to 10^{-5} and then follows the polynomial decay with power $p = 0.9$. The major-plane prediction network is trained on the raw dataset of KITTI with the input resolution of 352×704 on a NVIDIA RTX A6000 with 48 G memeory for 150 epochs with a batch size of 16. In order to further improve the generalization performance of the network, data augmentation is applied by randomly rotating the image and adjusting the contrast and the brightness of the image.

The major-plane prediction network is tested on the visual odometry datasets of KITTI which are different from the training dataset. Since the label images are obtained by the heuristic self-labeled method (introduced in Section IV), the labeled values actually don't have practical meanings, which are only used to distinguish different major-planes. Considering that one of the most important functions of the major-plane prediction network is to detect the ground plane areas, we evaluate the performance of the network by calculating the frame percentage $\mathcal{R}_f$ of which the ground plane area is detected on the MP-Mask on 11 sequences of the KITTI dataset. Specifically, $\mathcal{R}_f$ is the ratio of the number of the frames in which the ground plane can

TABLE II
LOCALIZATION RESULTS OF δ_{trans} AND δ_{rot} ON THE KITTI DATASET

Seq	Len (m)	ORB-SLAM2-noLC [9]		DSO [3]		VISO-M [46]		Song *et.al* [40]		Wagstaff *et.al* [41]		DNet [42]		Wang *et.al* [19]		Ours	
		Trans (%)	Rot (deg/m)	Trans (%)	Rot (deg/m)	Trans (%)	Rot (deg/m)	Trans (%)	Rot (deg/m)	Trans (%)	Rot (deg/m)	Trans (%)	Rot (deg/m)	Trans (%)	Rot (deg/m)	Trans (%)	Rot (deg/m)
00	3724	28.84	0.1982	29.78	0.2023	11.91	0.0209	2.04	0.0048	1.86	–	1.94	–	**1.01**	**0.0014**	1.19	0.0028
01	2453	*	*	1.79	0.0014	–	–	–	–	–	–	–	–	–	–	**1.11**	**0.0009**
02	5067	2.63	**0.0016**	5.43	0.0038	3.33	0.0114	1.50	0.0035	2.27	–	3.07	–	**0.93**	0.0018	1.91	0.0029
03	560	1.12	0.0020	0.79	0.0021	10.66	0.0197	3.37	0.0021	–	–	–	–	**0.52**	**0.0010**	0.82	0.0021
04	393	2.25	**0.0018**	0.89	0.0021	7.40	0.0093	2.19	0.0028	–	–	–	–	1.16	0.0023	**0.85**	0.0020
05	2205	8.53	0.0055	7.80	0.0015	12.67	0.0328	1.43	0.0038	1.50	–	3.32	–	1.45	**0.0014**	**1.01**	0.0015
06	1232	18.21	0.0074	18.08	0.0019	4.74	0.0157	2.09	0.0081	2.05	–	2.74	–	2.92	0.0027	**1.33**	**0.0019**
07	694	9.60	0.0121	7.52	0.0038	–	–	–	–	1.78	–	2.74	–	1.73	**0.0023**	**1.59**	0.0036
08	3222	12.39	0.0032	8.36	0.0028	13.94	0.0203	2.37	0.0044	2.05	–	2.72	–	1.18	**0.0017**	**0.90**	0.0021
09	1705	16.64	0.0053	9.54	**0.0019**	4.04	0.0143	1.76	0.0047	1.50	–	3.70	–	1.17	0.0020	**1.02**	0.0022
10	919	5.07	0.0063	5.49	0.0034	25.20	0.0388	2.12	0.0085	3.70	–	5.09	–	0.93	0.0029	**0.85**	**0.0025**
Avg	2015.82	10.53	0.0243	8.68	0.0206	10.43	0.0204	2.10	0.0047	2.09	–	3.17	–	1.25	**0.0020**	**1.14**	0.0022
Std	1424.57	8.30	0.0580	8.13	0.0575	6.42	0.0092	0.54	0.0021	0.66	–	0.87	–	0.65	**0.0006**	**0.33**	0.0007

* represents the method fails on this sequence.
The bold values represent the lowest translation error and rotation error, respectively.

be detected using the method in Section V to the total number of the frames on each sequence. The results are shown in Table I. It can be seen that the major-plane prediction network achieves a high $\mathcal{R}_f$ on most sequences, which shows the ground plane can be successfully detected in most frames.

C. Localization Results

Table II gives the quantitative results on all 11 sequences in the KITTI dataset. NALO-VOM achieves the best performance on 8 sequences in terms of the translation error and 3 sequences in terms of the rotation error. The VO system in [19] implemented a much stricter criterion on filtering the ground plane according to its norm direction, which leads to a slightly better performance on the rotation estimation on 5 sequences. On the other hand, [19] extracted local ground surfaces using the Delaunay Triangulation method, making it easier to deal with the ground planes which are not strictly flat. Therefore, [19] obtains a slightly better performance in terms of the translation error on sequences 00, 02, and 03, where the scenes contain a lot of slopes. However, [19] didn't takes the distance between the camera and the ground plane into account during estimation of the ground geometric model. In addition, it also strongly relies on the quantity and location of feature points during the ground plane detection. When the feature points cannot satisfy the requirements, the localization accuracy will be easily affected. In general, NALO-VOM presents much better performance than [19] in terms of the average translation error, while it reaches the approximately same level as [19] in terms of the average rotation error.

It is worth noting that ORB-SLAM2 performs better than the proposed method on sequence 02 in terms of rotation estimation, where the scenes are mainly composed of slope roads on hills, rather than flat surfaces. In such a situation, the proposed method cannot fully make use of the advantages of the scale optimization which is based on the global ground consistency assumption. On sequences 03 and 04, the vehicle just moves forward along a straight road, and thus no large change of the view angle of the camera occurs. As a result, the performance of the rotation estimation cannot be significantly distinguished between ORB-SLAM2 and NALO-VOM.

Sequence 01 is collected in a high-way scenario, on which ORB-SLAM2 fails to estimate a valid trajectory due to a large scale drift. This result is caused by the lack of enough features to constrain the scale. Since DSO optimizes the photometric error directly, it has better performance than ORB-SLAM2. NALO-VOM further improves localization accuracy by performing scale optimization based on MP-Mask. The works in [40], [41] and [42] also optimize the scale, which yields higher localization accuracy than ORB-SLAM2 [9], DSO [3] and VISO [46] without scale recovery. The method in [40] extracted the ground plane within a fixed area of the image plane. Therefore, on sequence 02 which has a clean driving road (i.e., the fixed area on the image plane always be the true ground plane), this method obtains slightly better performance than NALO-VOM. However, it cannot achieve robust and accurate performance in scenes with dynamic objects like sequences 03, 07, and 08. The methods in [41] and [42] need to predict depth maps for estimating the ground plane, which relies on the precision of the depth prediction network. In addition, both in translation errors and rotation errors, NALO-VOM achieves the lowest standard variance, which demonstrates the strong robustness of NALO-VOM.

Fig. 4 shows localization results on six sequences, where NALO-VOM is compared with ORB-SLAM2 [9] (without loop closure), DSO [3], and ground truth provided by KITTI. NALO-VOM outperforms ORB-SLAM2 and DSO on all six sequences, since it maintains a relatively consistent scale. It is worth noting that NALO-VOM obtains good localization results on sequences 02, 08, and 10 (contains a large height variation along the normal direction of the ground), which proves that the scale optimization in Section IV is available in the scenarios which don't contain a globally consistent ground plane. We would like to point out that although both roads and sidewalks in the KITTI dataset belong to the ground area, only roads are involved in the scale optimization process. In future work, sidewalks could be taken into account to introduce more ground constraints and further improve the accuracy of scale optimization.

We test NALO-VOM, DSO and ORB-SLAM2 on the KITTI dataset to evaluate the running time and memory consumption. The frame rate of NALO-VOM is 14 FPS when running on a desktop with an NVIDIA RTX 3070 with 8 GB VRAM, which can meet the real-time requirement from a perspective of semi-dense map building and navigation. The frame rates of DSO and ORB-SLAM2 are 19 FPS and 28 FPS, respectively, which have higher real-time performance because they do not perform dense tracking and relative scale optimization. We

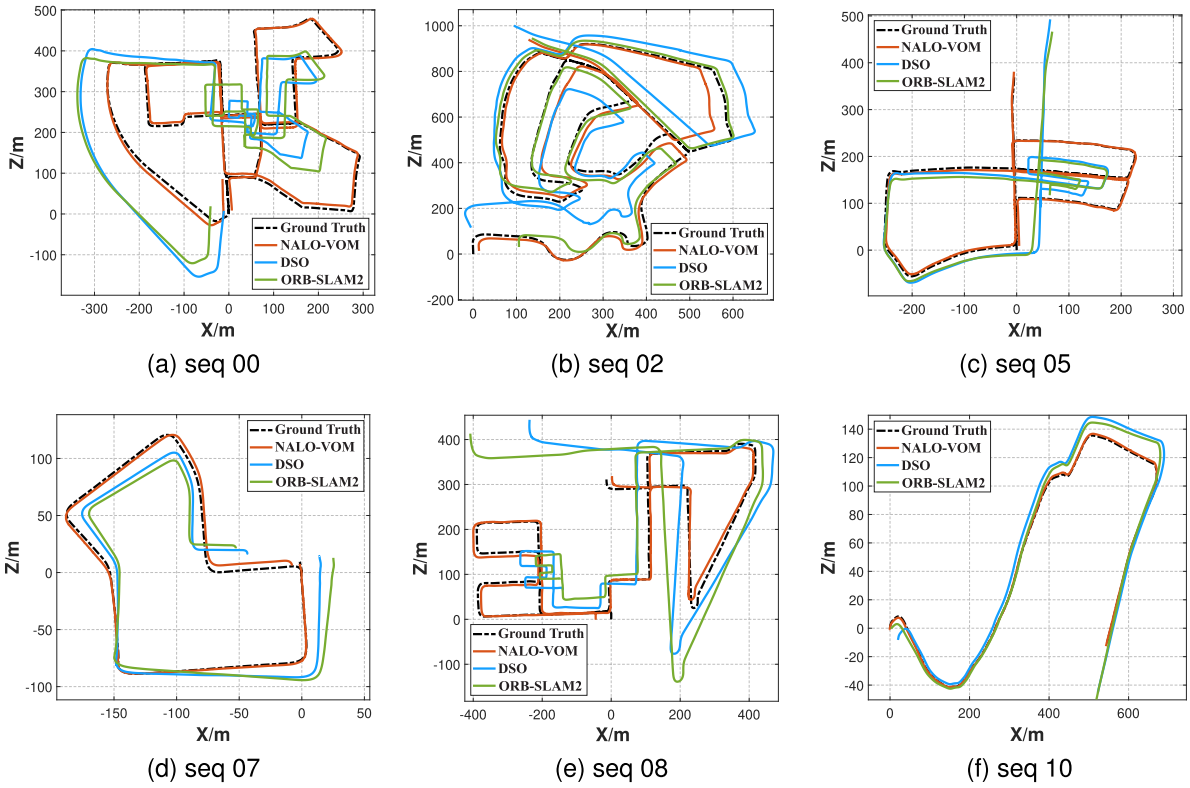

Fig. 4. Localization trajectories of NALO-VOM, ORB-SLAM2 and DSO on KITTI sequences.

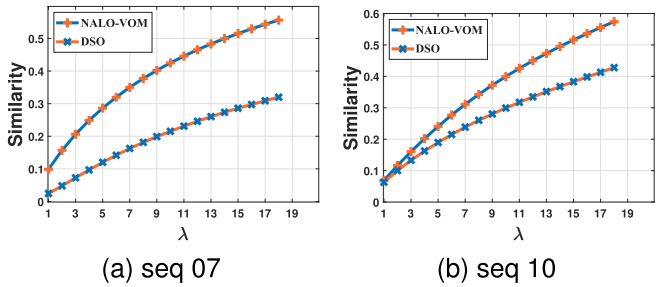

Fig. 5. Map similarity $\mathcal{S}^{\lambda}_{map}$ of NALO-VOM compared with that of DSO under different ratio factor λ on seq 07 and 10.

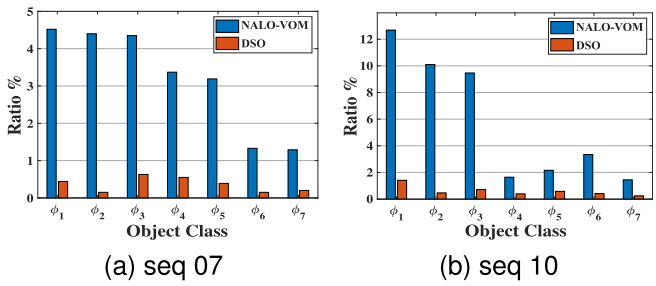

Fig. 6. Point cloud densities of different objects in NALO-VOM compared with that in DSO on seq 07 and 10.

also test the memory consumption of NALO-VOM, DSO and ORB-SLAM2 on sequence 04 of the KITTI dataset and the results are 0.804 GB, 0.544 GB, and 0.597 GB, respectively. The larger memory consumption of NALO-VOM comes from the semi-dense mapping strategy, which stores a much denser point cloud map.

D. Mapping Results

We evaluate the mapping quality on sequences 07 and 10 by comparing NALO-VOM with DSO [3]. Fig. 5 shows the similarity of $\{M_{\text{est\_NALO}}, M_{\text{lidar}}\}$ and $\{M_{\text{est\_DSO}}, M_{\text{lidar}}\}$ with different λ. DSO cannot recover map points in texture-less areas like walls. In the contrast, NALO-VOM utilizes the LiDAR-guided geometry priors to fill the map voids, which makes $M_{\text{est\_NALO}}$ more similar to M_{lidar}. Fig. 6 shows the point cloud densities of objects in the scenes, which are non-ignorable when building a navigation-oriented map for UGVs. The point densities of all the listed objects built by NALO-VOM greatly exceed those built by DSO, as shown in Fig. 6. It is worth noting that, the point densities of driving-aware objects are greatly increased, which can facilitate more accurate obstacle avoidance and motion planning. On the other hand, the denser description of high-level objects helps achieve a better semantic understanding of the environment, which further improves the applicability of the resultant map.

Fig. 7 shows the 2D navigation grid maps generated by LOAM [48] (a LiDAR-based odometry and mapping method), NALO-VOM and DSO on sequences 07 and 10, respectively. It can be seen that the 2D navigation grid map built by NALO-VOM is much more similar to that by LOAM than that by DSO. To be specific, the wall is not continuous on the grid map of DSO,

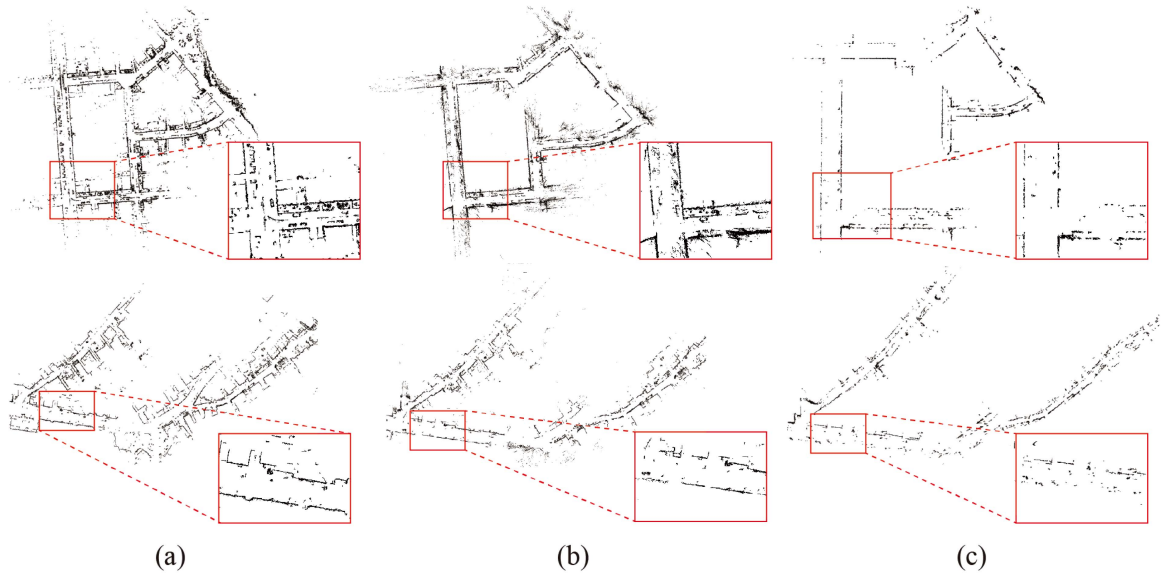

Fig. 7. 2D navigation grid maps generated by (a) LOAM, (b) NALO-VOM, and (c) DSO. The upper row shows the maps built on seq 07, and the lower row shows the maps built on seq 10.

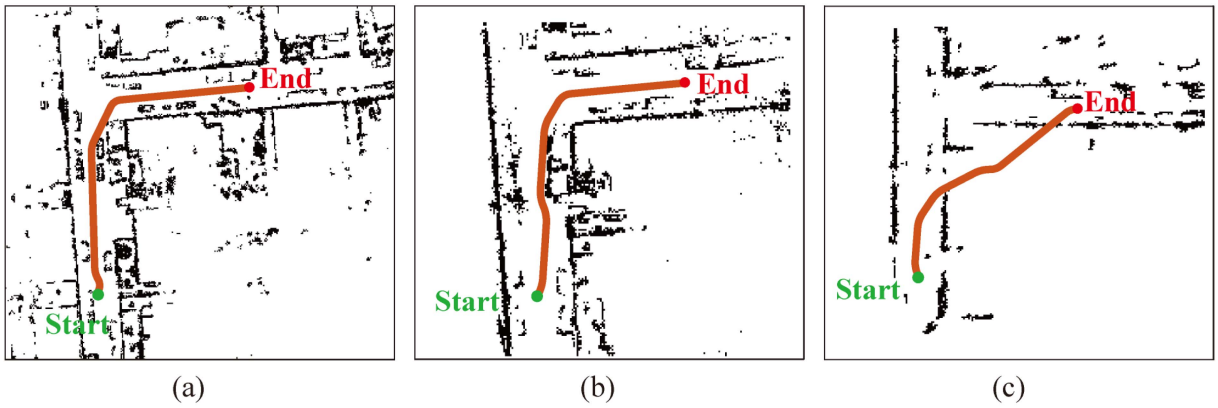

Fig. 8. Path planning results on the 2D grid maps of (a) LOAM, (b) NALO-VOM, and (c) DSO, respectively, using the hybrid A* algorithm on seq 07.

because of the sparseness of the point cloud in the texture-less areas. In order to further verify whether these grid maps can be used for navigation, the hybrid A* path planning algorithm is performed on the 2D grid maps generated by LOAM, NALO-VOM, and DSO, respectively. Fig. 8 shows the path planning results. It can be seen that the hybrid A* algorithm fails to plan a valid path on the 2D grid map of DSO. Since the walls are not continuous, the planned path is across the walls which will cause collisions in the real world. In the contrast, the grid map built by NALO-VOM can generate a similar path with that by LOAM. Generally, the point cloud map built by NALO-VOM can have a good trade-off between accuracy and density, which makes it more suitable to be used in UGV navigation.

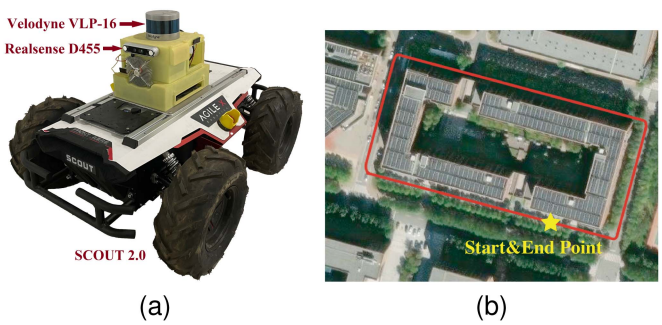

Fig. 9. (a) The UGV platform (Scout 2.0) used in the real-world experiment, (b) the corresponding self-collected trajectory overlaid with the satellite map for visualization.

E. Real-World Experiments

In order to verify the applicability of NALO-VOM on different platforms, the real-world experiments are conducted on two different platforms. One is a Scout 2.0 UGV equipped with a 3D LiDAR (Velodyne VLP-16) and a camera (Realsense D455) as shown in Fig. 9(a). The other is a Pioneer P3-DX mobile robot equipped with a 3D LiDAR (Livox MID-70) and a camera (Kinect V2) as shown in Fig. 10(a). The 3D LiDARs on the two

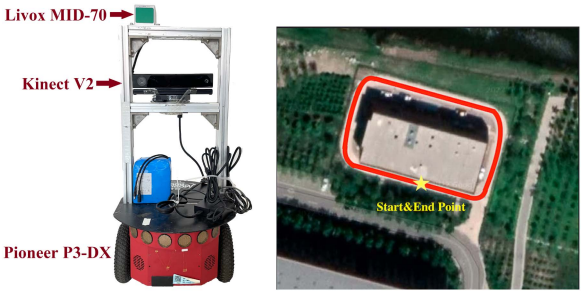

Fig. 10. (a) The mobile robot platform (Pioneer P3-DX) used in the real-world experiment, (b) the corresponding self-collected trajectory overlaid with the satellite map for visualization.

TABLE III
RELATIVE DRIFT ON SELF-COLLECTED SEQUENCES

	DSO	ORB-SLAM2	NALO-VOM
Scout 2.0	14.60%	8.01%	**0.67%**
Pioneer P3-DX	9.70%	5.66%	**1.24%**

The bold values represent the lowest relative drift error.

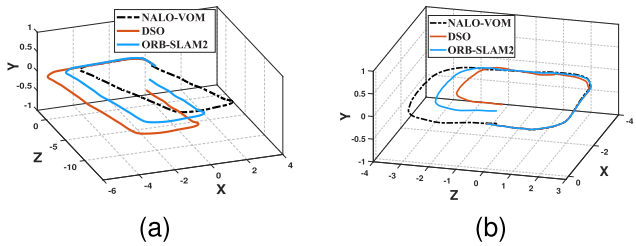

Fig. 11. Estimated trajectories of NALO-VOM, ORB-SLAM2, and DSO on the self-collected sequence on the (a) Scout 2.0, (b) Pioneer P3-DX.

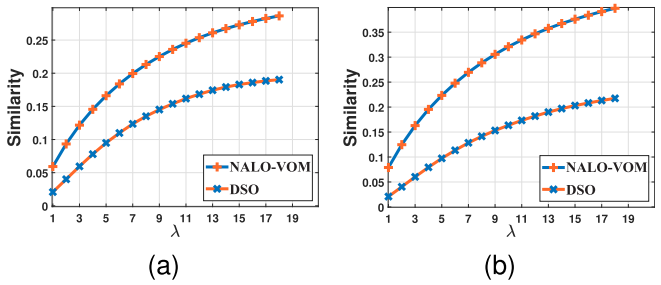

Fig. 12. Map similarity $\mathcal{S}^{\lambda}_{map}$ of NALO-VOM compared with that of DSO under different ratio factor λ on the self-collected sequence on the (a) Scout 2.0, (b) Pioneer P3-DX.

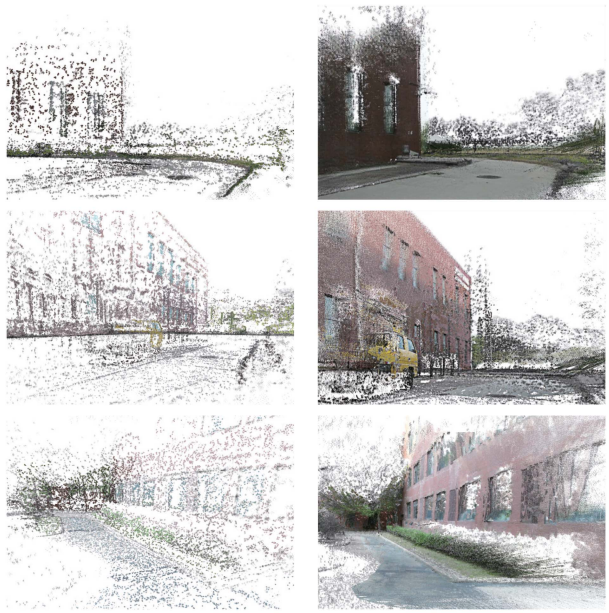

Fig. 13. (a) The sparse point cloud maps of DSO, (b) corresponding semi-dense point cloud maps of NALO-VOM.

platforms are used for obtaining the ground truth map. The RGB image sequences are acquired by running around two different buildings where the start point is the same as the end point, as shown in Figs. 9(b) and 10(b). The full length of the two trajectories are about 325.2 m and 213.4 m, respectively. Since the absolute scale is unavailable, the localization accuracy is measured by the relative drift of the estimated trajectory, i.e., the ratio of the drift to the total length of the trajectory. NALO-VOM is compared with DSO and ORB-SLAM2 (without loop closure), and the relative drift is shown in Table III. The estimated trajectories of the two platforms by three monocular visual odometry algorithms are shown in Figs. 11(a) and 11(b), respectively. Neither DSO nor ORB-SLAM2 conducts scale recovery, therefore their trajectories can't be closed up because of the scale drift. Compared with DSO and ORB-SLAM2, NALO-VOM obtains much more accurate trajectories on different platforms, which shows great applicability of NALO-VOM on different experimental platforms.

Figs. 12(a) and 12(b) show the mapping results of DSO and NALO-VOM. It can be seen that the map built by NALO-VOM is more similar to the LiDAR map compared to that built by DSO, which mainly benefits from the more accurate localization and denser mapping strategy of NALO-VOM. The visualization results of the obtained point cloud maps are shown in Fig. 13.

It is worth noting that MP-Mask used in this experiment is obtained by the prediction network trained on the KITTI dataset in advance. Therefore, the localization result in this experiment shows a good generalization ability of NALO-VOM.

VII. CONCLUSION

In this article, we have proposed NALO-VOM, a system using LiDAR priors to assist the monocular VO to build a navigation-oriented environment map and generate a scale-consistent localization performance. The geometric structure representation ability of the LiDAR is extracted by an off-line trained major-plane prediction network using non-artificial projection labels and then transferred into the monocular VO with the resultant MP-Mask. The ground plane is extracted using MP-Mask to constrain the pose scale at the frontend. Finally, the environment map is refined using geometry priors provided by MP-Mask,

where the map voids are filled using major-planes. NALO-VOM combines traditional stereo-based depth estimation and the deep-network based method to balance the accuracy and density of the environment map. The resultant dense point cloud map is more suitable to generate an accurate navigation map (e.g., 2D grid map) for motion planning and decision making. Moreover, the scale consistency of NALO-VOM makes it promising to apply to UGVs in long-term localization. In future work, the orientation of the ground normal vector will be more thoroughly considered and incorporated into the scale optimization process to handle scenarios with inclined ground surfaces.

REFERENCES

[1] M. Venator, E. Bruns, and A. Maier, "Robust camera pose estimation for unordered road scene images in varying viewing conditions," *IEEE Trans. Intell. Veh.*, vol. 5, no. 1, pp. 165–174, Mar. 2020.

[2] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 2, no. 3, pp. 194–220, Sep. 2017.

[3] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.

[4] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 15–22.

[5] Y. Shu, P. Xu, X. Niu, Q. Chen, L. Qiao, and J. Liu, "High-rate attitude determination of moving vehicles with GNSS: GPS, BDS, GLONASS, and Galileo," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 5501813, doi: 10.1109/TIM.2022.3168896.

[6] L.-T. Hsu, Y. Gu, and S. Kamijo, "Intelligent viaduct recognition and driving altitude determination using GPS data," *IEEE Trans. Intell. Veh.*, vol. 2, no. 3, pp. 175–184, Sep. 2017.

[7] A. Brunker, T. Wohlgemuth, M. Frey, and F. Gauterin, "Odometry 2.0: A slip-adaptive EIF-based four-wheel-odometry model for parking," *IEEE Trans. Intell. Veh.*, vol. 4, no. 1, pp. 114–126, Mar. 2019.

[8] M. Brossard, A. Barrau, and S. Bonnabel, "AI-IMU dead-reckoning," *IEEE Trans. Intell. Veh.*, vol. 5, no. 4, pp. 585–595, Dec. 2020.

[9] R. Mur-Artal and J. D. 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.

[10] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "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.

[11] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 834–849.

[12] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-time dense monocular slam with learned depth prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6565–6574.

[13] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2320–2327.

[14] C. Fan, J. Hou, and L. Yu, "Large-scale dense mapping system based on visual-inertial odometry and densely connected U-Net," *IEEE Trans. Instrum. Meas.*, vol. 72, 2023, Art. no. 5008916, doi: 10.1109/TIM.2023.3250301.

[15] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. 9th Eur. Conf. Comput. Vis.*, 2006, pp. 430–443.

[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[17] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 404–417.

[18] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to sift or SURF," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.

[19] X. Wang, H. Zhang, X. Yin, M. Du, and Q. Chen, "Monocular visual odometry scale recovery using geometrical constraint," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 988–995.

[20] R. Radmanesh, Z. Wang, V. S. Chipade, G. Tsechpenakis, and D. Panagou, "LIV-LAM: LiDAR and visual localization and mapping," in *Proc. IEEE Amer. Control Conf.*, 2020, pp. 659–664.

[21] W. Shao, S. Vijayarangan, C. Li, and G. Kantor, "Stereo visual inertial LiDAR simultaneous localization and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 370–377.

[22] J. Lin, C. Zheng, W. Xu, and F. Zhang, "R ^{2} LIVE: A. robust, real-time, LiDAR-inertial-visual Tightly-Coupled state estimator and mapping," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 7469–7476, Oct. 2021.

[23] C. Shu and Y. Luo, "Multi-modal feature constraint based tightly coupled monocular visual-LiDAR odometry and mapping," *IEEE Trans. Intell. Veh.*, vol. 8, no. 5, pp. 3384–3393, May 2023.

[24] F. Wimbauer, N. Yang, L. von Stumberg, N. Zeller, and D. Cremers, "MonoRec: Semi-supervised dense reconstruction in dynamic environments from a single moving camera," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6108–6118.

[25] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison, "CodeSLAM - Learning a compact, optimisable representation for dense visual slam," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2560–2568.

[26] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Proc. IEEE 4th Int. Conf. 3D Vis.*, 2016, pp. 239–248.

[27] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. Yuille, "Towards unified depth and semantic prediction from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2800–2809.

[28] J. Czarnowski, T. Laidlow, R. Clark, and A. J. Davison, "DeepFactors: Real-time probabilistic dense monocular SLAM," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 721–728, Apr. 2020.

[29] L. Koestler, N. Yang, N. Zeller, and D. Cremers, "TANDEM: Tracking and dense mapping in real-time using deep multi-view stereo," in *Proc. Conf. Robot Learn.*, 2022, pp. 34–45.

[30] M. Yokozuka, S. Oishi, S. Thompson, and A. Banno, "VITAMIN-E: Visual tracking and mapping with extremely dense feature points," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9633–9642.

[31] W. N. Greene and N. Roy, "FLaME: Fast lightweight mesh estimation using variational smoothing on delaunay graphs," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4696–4704.

[32] M. Zhu et al., "Monocular depth prediction through continuous 3D loss," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 10742–10749.

[33] D. Chen et al., "VIP-SLAM: An efficient tightly-coupled RGB-D visual inertial planar SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 5615–5621.

[34] X. Zhang, W. Wang, X. Qi, Z. Liao, and R. Wei, "Point-plane SLAM using supposed planes for indoor environments," *Sensors*, vol. 19, no. 17, 2019, Art. no. 3795.

[35] R. Wang et al., "Semantic ground plane constraint in visual SLAM for indoor scenes," in *Proc. Chin. Conf. Pattern Recognit. Comput. Vis.*, 2020, pp. 268–279.

[36] X. Wang, M. Christie, and E. Marchand, "Relative pose estimation and planar reconstruction via superpixel-driven multiple homographies," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 10625–10632.

[37] C. Chen, P. Geneva, Y. Peng, W. Lee, and G. Huang, "Monocular visual-inertial odometry with planar regularities," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 6224–6231.

[38] F. Wu and G. Beltrame, "Direct sparse odometry with planes," *IEEE Robot. Automat. Lett.*, vol. 7, no. 1, pp. 557–564, Jan. 2022.

[39] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," 2019, *arXiv:1907.10326*.

[40] S. Song, M. Chandraker, and C. C. Guest, "High accuracy monocular SFM and scale correction for autonomous driving," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 730–743, Apr. 2016.

[41] B. Wagstaff and J. Kelly, "Self-Supervised scale recovery for monocular depth and egomotion estimation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 2620–2627.

[42] F. Xue, G. Zhuo, Z. Huang, W. Fu, Z. Wu, and M. H. Ang, "Toward hierarchical self-supervised monocular absolute depth estimation for autonomous driving applications," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 2330–2337.

[43] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1449–1456.

[44] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[45] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[46] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3D reconstruction in real-time," in *Proc. IEEE Intell. Veh. Symp.*, 2011, pp. 963–968.
[47] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," *Ann Arbor*, vol. 1001, no. 48105, pp. 18–80, 2008.
[48] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," in *Proc. Robot.: Sci. Syst.*, 2014, vol. 2, pp. 1–9.
[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[50] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, pp. 211–252, 2015.
[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

Ziqi Hu received the B.Eng. degree in intelligence science and technology and the M.S. degree in control science and engineering from the Institute of Robotics and Automatic Information System from Nankai University, Tianjin, China, in 2020 and 2023, respectively. His research interests include visual SLAM, unmanned ground vehicle navigation, and machine learning.

Jing Yuan (Member, IEEE) received the B.S. degree in automatic control and the Ph.D. degree in control theory and control engineering from Nankai University, Tianjin, China, in 2002 and 2007, respectively. From 2007 to 2018, he was with the College of Computer and Control Engineering, Nankai University. He is currently a Professor with the College of Artificial Intelligence, Nankai University. His research interests include robotic control and SLAM. Dr. Yuan is an Associate Editor for the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT.

Yuanxi Gao received the B.Eng. degree in intelligence science and technology in 2019 from Nankai University, Tianjin, China, where he is currently working toward the Ph.D. degree in control science and engineering with the Institute of Robotics and Automatic Information System. His research interests include multisensor fusion, mobile robot navigation, and SLAM.

Boran Wang received the B.Eng. degree in intelligence science and technology from the Hebei University of Technology, Tianjin, China, in 2021. He is currently working toward the M.S. degree in control engineering, Nankai University, Tianjin, China. His research interests include deep learning and computer vision.

Xuebo Zhang (Senior Member, IEEE) received the B.Eng. degree in automation from Tianjin University, Tianjin, China, in 2002, and the Ph.D. degree in control theory and control engineering from Nankai University, Tianjin, in 2011. He is currently a Professor with the College of Artificial Intelligence, Nankai University. His research interests include motion planning, visual servoing, and localization and mapping. Dr. Zhang is the Technical Editor of the IEEE/ASME TRANSACTIONS ON MECHATRONICS and an Associate Editor for the *ASME Journal of Dynamic Systems, Measurement, and Contro*l.