Accurate LiDAR-Camera Fused Odometry and RGB-Colored Mapping

Zhipeng Lin, Zhi Gao, Ben M. Chen, *Fellow, IEEE*, Jingwei Chen, and Chenyang Li

***Abstract*—Due to the complementary properties between sensors, multi-sensor fusion can effectively promote accuracy and tackle the challenging scenes in simultaneous localization and mapping (SLAM) tasks. To this end, we propose a novel LiDAR-camera fused method for odometry and mapping using dense colored point clouds. With the camera well calibrated to the LiDAR, we can acquire colored point clouds, providing constraints of color and geometric features for SLAM tasks. Our LiDAR-camera fused odometry and mapping system leverages the geometric features from the point cloud and color information from the camera. The main innovation is projecting the colored point to the local point cloud plane and formulating an RGB color objective function for SLAM tasks. We optimize the geometric and color objective functions jointly to estimate the precise pose of the robot. In particular, we maintain a color feature map and a planar feature map separately in the optimization process, which reduces the algorithm's computation significantly. The evaluation experiments are performed on a UGV platform and a handheld platform. We demonstrate the effectiveness of our LiDAR-camera fusion method using the solid-state LiDAR and camera on an Intel RealSense L515 sensor. The results show that our method effectively promotes localization accuracy, works well in challenging environments, and outperforms existing methods. We will share the code publicly to benefit the community (after the review stage).**

***Index Terms*—Localization, mapping, sensor fusion, simultaneous localization and mapping (SLAM).**

I. Introduction

FOR mobile robots, simultaneous localization and mapping (SLAM) is an essential and fundamental capability for environment perception, especially in unknown environments. Researchers have designed numerous SLAM frameworks for single perceptual sensors, such as camera and LiDAR [1], [2]. However, every single sensor has its limitations, leading to degenerate situations. Multi-sensor fusion is one solution to these problems, as different types of sensors have complementary properties. Moreover, since the data patterns of sensors are different, multi-sensor fusion can offer additional independent constraints for robot pose optimization and enhance the final result. Recent years have witnessed great progress in the multi-sensor fused SLAM works. Therefore, sensor-fusion based SLAM is necessary for the development of robotics and has increasingly attracted researchers to explore the sensor fusion frameworks to develop practical SLAM systems. Recently, the introduction of solid-state LiDAR provides a cost-effective and lightweight solution for LiDAR SLAM systems. As shown in Fig. 1, solid-state LiDAR can achieve a very dense point cloud and is often integrated with other sensors such as a camera. Designing SLAM methods to make full use of the specifications of the solid-state LiDAR is of great significance.

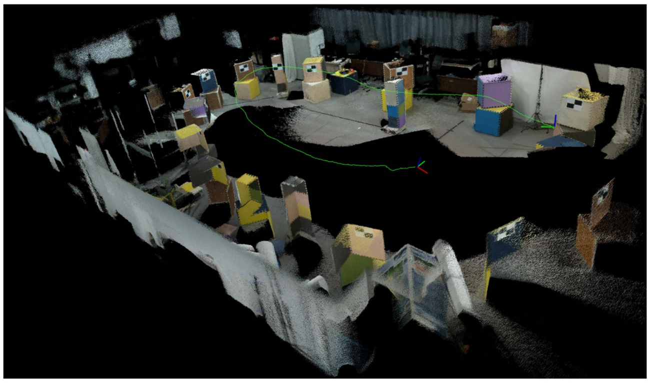

Fig. 1. Localization and mapping result of our method using Intel RealSense L515 mounted on a UGV in our experiment site. The mapping is accurate and the average localization error is within 3 cm.

Existing LiDAR SLAM approaches [3], [4], [5] are mainly developed for traditional LiDAR sensors, which generate the sparse point cloud in a low frequency. Although they have achieved impressive experimental results on large-scale mapping, their usage is limited due to the high cost and low angular resolution. As the data pattern is different, existing LiDAR-camera fused SLAM work cannot be directly applied to the solid-state LiDAR. Moreover, some LiDAR-camera fused SLAM work consists of a visual SLAM aided with LiDAR [6]. The LiDAR usually provides depth information for visual SLAM [7], [8]. These methods can not take full advantage of the geometric information of the LiDAR points. Some other LiDAR-camera

Manuscript received 6 November 2023; accepted 5 January 2024. Date of publication 22 January 2024; date of current version 1 February 2024. This letter was recommended for publication by Associate Editor R. Sagawa and Editor P. Vasseur upon evaluation of the reviewers' comments. This work was supported in part by Hubei Province Natural Science Foundation under Grant 2021CFA088, and in part by Science and Technology Major Project under Grants 2021AAA010 and 2021AAA010-3. *(Corresponding author: Zhi Gao.)*

Zhipeng Lin is with the Department of Mechanical and Automation Engineering, Chinese University of Hong Kong, Hong Kong 999077, China, and also with the Department of Mathematics and Theories, Peng Cheng Laboratory, Shenzhen 518000, China (e-mail: zplin@link.cuhk.edu.hk).

Zhi Gao is with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430072, China, and also with the Hubei Luojia Laboratory, Wuhan University, Wuhan 430079, China (e-mail: gaozhinus@gmail.com).

Ben M. Chen is with the Department of Mechanical and Automation Engineering, Chinese University of Hong Kong, Hong Kong 999077, China (e-mail: bmchen@cuhk.edu.hk).

Jingwei Chen and Chenyang Li are with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430072, China (e-mail: jingweichen@whu.edu.cn; chenyangli@whu.edu.cn).

Digital Object Identifier 10.1109/LRA.2024.3356982

fused SLAM works are two-stage loosely-coupled fusion methods [9], [10]. These methods typically estimate robot pose using camera and LiDAR separately thus cannot achieve global optimal result and is easily affected by sensor failure. Moreover, there are tightly-coupled fusion methods based on the Kalman filter or graph optimization. However, these complex methods usually rely on an IMU odometry [11], [12] and take up a significant computation load. Our method can be considered as a tightly coupled LiDAR-camera fused odometry that saves computation resources and achieves high accuracy.

In this work, we propose a LiDAR-camera fused odometry and mapping system for devices that generate dense colored point clouds such as solid-state LiDAR. To the best of our knowledge, this is the first work of LiDAR-camera fused odometry using the dense colored point cloud. The main novelty is that we project the colored point to a local plane and approximate the color using a linear function by calculating the color gradient in RGB channels and formulating the color objective in joint optimization with the geometric objective to promote accuracy and robustness in localization and mapping. Our method fuses LiDAR and the camera at the data level by generating the colored point cloud and fuses their features directly in the optimization process. The experimental results prove the advantages of our LiDAR-camera fused framework that exceeds the existing works. The main contributions of our work are as follows:

- We propose a LiDAR-camera fused odometry framework for dense colored point clouds. We perform the first-order approximation of the point color in the local point plane and optimize the color and geometric objectives jointly in scan-to-map matching for pose estimation and mapping.
- We introduce a novel RGB-channel color and geometric feature extraction method based on plane fitting and color gradient computation. We maintain color and geometric feature maps separately in scan-to-map matching to reduce computational costs.
- We perform extensive experiments to evaluate our proposed method. More specifically, we integrate an Intel RealSense L515 sensor to UGV and a handheld platform to test the performance of our proposed method, which outperforms the state-of-the-art techniques.

II. Related Work

Fusing these multi-sensor measurements, particularly from camera and LiDAR, is often addressed within a SLAM framework. Here, we survey the works related to LiDAR-camera fused SLAM and odometry. Specifically, we adopt SSL-SLAM [13] as the baseline, the state-of-the-art SLAM work for solid-state LiDAR sensors. SSL-SLAM will fail in LiDAR degenerate scenes as it only uses edge and planar features from laser points.

A. Visual SLAM Aided With LiDAR Depth

There are many LiDAR-camera fused SLAM works that adopt visual SLAM as their main framework and use LiDAR points to generate the depth map to enhance the visual SLAM. Some work [8], [14] extract depth from LiDAR measurements for camera feature tracking and estimates pose among keyframes based bundle adjustment, which formulates an RGBD odometry system. In addition, some works [7] use the sparse LiDAR depth to enhance a direct visual SLAM system and iteratively minimize photometric errors. In CamVox [15], the LiDAR points provide depth information associated with camera images to enable the RGB-D mode of ORB-SLAM [16]. These methods only use the depth information extracted from LiDAR points. They cannot take full advantage of the environmental geometric information from laser points in their LiDAR-camera fusion framework.

B. Two-Stage Loosely-Coupled Fusion Method

Recently, some works have proposed loosely-coupled fusion methods as they optimize robot pose using the data from the camera and LiDAR in different stages. One typical work [9] starts with LiDAR-enhanced RGBD odometry to estimate the ego-motion and then uses lidar odometry to refine motion estimates and remove distortion. DV-LOAM [10] is a direct vision LiDAR fusion SLAM framework that first adopts a two-staged direct visual odometry to estimate the accurate pose efficiently. Then, the LiDAR mapping module is utilized to refine the pose of the keyframe. The limitation of these methods is that different sensor measurements are not jointly used in the optimization process. And the more recent approaches rarely adopt this kind of fusion framework.

C. Tightly-Coupled Fusion Method Based on Filter or Graph Optimization

Many tightly-coupled filter based fusion method includes stereo camera, laser, and IMU methods. They usually use EKF (Extended Kalman Filter), or MSCKF (Multi-State Constraint Kalman Filter) for pose estimation, which is fast and need low computational cost, but are sensitive to time synchronization [17]. LIC-fusion [18], [19] fuses the LiDAR odometry and visual-inertial odometry tightly within a Multi-State Constrained Kalman Filter (MSCKF) framework. To further enhance the robustness of the LiDAR scan matching, R3LIVE++ [11] is a tightly-coupled LiDAR-visual-inertial fused SLAM work, which achieves an accurate state estimation by extracting LiDAR and image features and then includes re-projection error within the Iterated Error State Kalman Filter framework.

The graph optimization based methods adopt local maps or sliding windows to reduce the effect of time synchronization. In addition, they optimize history poses and achieve real-time performance with bundle adjustment. LVI-SAM [12] fuses a visual-inertial system and a LiDAR-inertial system in a factor graph [20]. In these methods, point cloud features and vision features are not used jointly in the optimization of pose estimation. Thus, they cannot take full advantage of the complementary properties between LiDAR and the camera.

Different from the above methods, our method is a tightly coupled LiDAR-camera fused odometry and mapping for dense colored point clouds. We fuse the color feature and geometric feature in the optimization formulation directly. In this way, we reduce the computation burden and achieve state-of-the-art performance.

III. Methods

In this section, the proposed method is introduced in detail. The overview of our method is shown in Fig. 2, which mainly consists of a feature extraction module, an odometry estimation module, and a mapping module. We first introduce the extraction of geometric features and RGB color features. Then, we show the objective formulation and optimization process of the

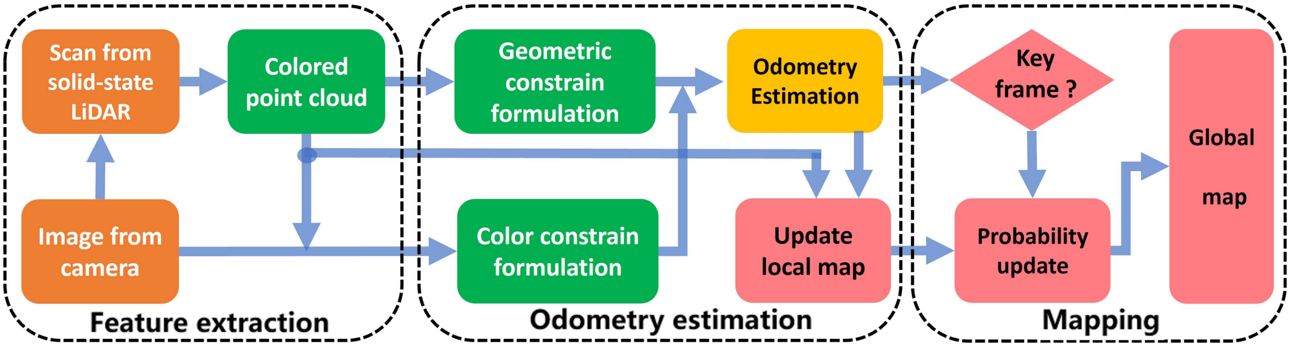

Fig. 2. Overview of the architecture of our proposed method.

fused odometry estimation. Finally, we present the method of probability map construction.

A. Geometric Feature Extraction

A solid-state LiDAR usually has higher resolution and higher update frequency compared to mechanical LiDAR [13]. Registering the raw point clouds can be a heavy computational burden. Traditional methods such as LOAM [3] usually leverage edge and planar matching. As the edge is the intersection of planes, the edge feature cannot provide additional constraints for optimization. Many other works [21], [22], [23] adopt this idea in their framework and achieve high performance. We conduct voxel filtering on the point cloud in advance for more robust feature extraction. Also, the point cloud is downsampled to reduce the computation.

In our odometry system, we use point-to-plane objectives to construct geometric constraints. The plane formulation in 3D space is $ax + by + cz + d = 0$, and when $d \neq 0$ can be formulated as $ax + by + cz + 1 = 0$, the normal vector is $\mathbf{n} = (a, b, c)$. For a point from the source cloud, find n nearest points in the target point cloud and extract a plane. For n points, we have:

$$\begin{aligned} &ax_1 + by_1 + cz_1 + 1 = 0 \\ &ax_2 + by_2 + cz_2 + 1 = 0 \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &ax_n + by_n + cz_n + 1 = 0 \end{aligned} \tag{1}$$

We solve a, b, c using QR decomposition. And we investigate the distance of the plane and sampled point to validate if it is a planar feature point, which will finally be used for the geometric objective optimization.

B. RGB Color Feature Extraction

For well-calibrated LiDAR and camera, we can get the correct correspondence of the laser point and the pixel in the image. Otherwise, our method cannot utilize color information and will degenerate into a pure LiDAR-based method. For RGB color information, we process 3 channels independently. For other color spaces, such as HSV, the gradient computing is more complex than that in RGB color space. Also, the RGB color space is used in most visual SLAM works. So we adopt RGB color space only. Using the Gaussian-Newton method, we need to perform differential operations in color space to compute the Jacobian matrix. However, suppose a color function $\mathbf{C}(\mathbf{p})$ that can output the color of any existing point $\mathbf{p}$ in a point cloud. Apparently, $\mathbf{C}(\mathbf{p})$ is not a continuous function and does not even have an analytical formula. The first work to enable the optimization is performing approximation [24].

For a point $\mathbf{p}$ from a point cloud, now we can get its plane formula from Section III-A, and the normal vector is $\mathbf{n_p}$. We have a nearby point $\mathbf{q}$. The projection of $\mathbf{q}$ on the $\mathbf{p}$-plane is

$$\mathbf{f_C}(\mathbf{q}) = \mathbf{q} - \mathbf{n_p}(\mathbf{q} - \mathbf{p})^\top \mathbf{n_p} \tag{2}$$

We suppose $\mathbf{C_p}(\mathbf{q})$ is a continuous function that describe the color of a point $\mathbf{q}$ near $\mathbf{p}$, and it can be approximated as:

$$\mathbf{C_p}(\mathbf{q}) \approx \mathbf{C}(\mathbf{p}) + \mathbf{g_c}^\top(\mathbf{f_C}(\mathbf{q}) - \mathbf{p}) \tag{3}$$

The $\mathbf{g_c}^\top$ is defined as the gradient of color, and we have to calculate it in the following part. We project the point to the plane because the color is a kind of 2-D information. And (12) shows the assumption that the color changes linearly in a small area.

Then compute the color gradient $\mathbf{g_c}^\top$. We suppose the $\mathbf{C_p}(\mathbf{q})$ perform well in $\mathbf{p}$ nearby set $\mathcal{N}_\mathbf{p}$. Then we can formulate an optimization objective:

$$\begin{aligned} \mathbf{g_c} &\approx arg \min_{\mathbf{g_c}} \sum_{\mathbf{q} \in \mathcal{N}_\mathbf{p}} \left(C(\mathbf{p}) + \mathbf{g_c}^\top \left(\mathbf{f_C}\left(\mathbf{q}\right) - \mathbf{p}\right) - C\left(\mathbf{q}\right)\right)^2 \\ &\approx arg \min_{\mathbf{g_c}} \sum_{\mathbf{q} \in \mathcal{N}_\mathbf{p}} \left(\left(\mathbf{f_C}\left(\mathbf{q}\right) - \mathbf{p}\right)^\top \mathbf{g_c} - \left(C\left(\mathbf{q}\right) - C(\mathbf{p})\right)\right)^2 \end{aligned} \tag{4}$$

which is the form of quadratic programming without constraint [24]. Or it can be considered as a least squares problem, whose form is $L = \sum_i (\mathbf{A}_i\mathbf{x} - \mathbf{b}_i)^2$, where

$$\mathbf{A}_i = (\mathbf{f_C}(\mathbf{q}) - \mathbf{p})^\top \qquad \mathbf{x} = \mathbf{g_c} \qquad \mathbf{b}_i = C(\mathbf{q}) - C(\mathbf{p}) \tag{5}$$

Now we can calculate the color gradient $\mathbf{g_c}^\top$ and thus get the continuous color function $\mathbf{C_p}(\mathbf{q})$. We investigate the L2 norm of the color gradient $\mathbf{g_c}^\top$ to find the color variation area, which is called RGB color feature extraction. The example result is shown in the experiment section. And we maintain a submap that contains color feature points specifically for the RGB color objective optimization.

C. The Point-to-Plane Objective in Odometry Estimation

Now we have the plane equation and normal vector $\mathbf{n} = (a, b, c)$. Our objective is to determine the transformation T, which transform the input point $p = (x, y, z)$ to $p' = (x', y', z')$.

And p' should be on the target plane. Then we formulate the point-to-plane objective $\mathbf{E_G} = \sum \parallel \mathbf{r_G} \parallel_\mathbf{2}^\mathbf{2}$:

$$\begin{aligned}\mathbf{r_G} &= \mathbf{f_G}\left(p; \boldsymbol{T}\right) = \left(\boldsymbol{T}p\right)^\top \mathbf{n} + 1 \\ &= (p')^\top \mathbf{n} + 1 = ax' + by' + cz' + 1\end{aligned} \tag{6}$$

Calculating the derivative of the transformation $\boldsymbol{T}$ directly is difficult. Here we use the left perturbation scheme and apply increment on the Lie Group [13]. Note that using other methods, such as quaternion, is equally effective.

Thus, let $\xi = [\rho, \phi] = (\alpha\,\beta\,\gamma\,x\,y\,z)^T \in \mathfrak{se}(3)$, and the transformation matrix $\boldsymbol{T} = \exp(\xi^\wedge)$.

Let

$$\mathbf{f_G}\left(p; \boldsymbol{T}\right) = g(h) = (h)^\top \mathbf{n} + 1 \tag{7}$$

$$h(p) = \boldsymbol{T}p \qquad \mathbf{f_G} = g \circ h \tag{8}$$

According to the derivative rule for a composite function,

$$\mathbf{J_G} = \frac{\partial \mathbf{r_G}}{\partial \xi} = \frac{\partial \mathbf{f_G}}{\partial \xi} = \frac{\partial g}{\partial h}\frac{\partial h}{\partial \xi} \qquad \frac{\partial g}{\partial h} = \mathbf{n}^\top \tag{9}$$

$$\begin{aligned}\frac{\partial h}{\partial \xi} &= \mathbf{J}_p = \frac{\partial \boldsymbol{T}\mathbf{p}}{\partial \xi} = \lim_{\xi \to \mathbf{0}} \frac{\left(\exp\left(\xi^\wedge\right) \cdot \boldsymbol{T}\mathbf{p} - \boldsymbol{T}\mathbf{p}\right)}{\xi} \\ &= \begin{bmatrix} -\left[\boldsymbol{T}\mathbf{p}\right]_\times & \mathbf{I}_{3\times3} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3}\end{bmatrix}\end{aligned} \tag{10}$$

Thus, we can get the final Jacobian matrix:

$$\mathbf{J_G} = \mathbf{n}^\top \mathbf{J}_p \tag{11}$$

Now it is sufficient to use Gaussian-Newton method to optimize the $\boldsymbol{T}$ transformation using geometric features.

D. RGB Color Objective Construction and Jacobian Computation in Odometry Estimation

Then we formulate the color objective function and compute the Jacobian matrix. Recall the (12) that we have the continuous color function. Suppose $\mathbf{s}$ is a point from the source point cloud. $\mathbf{s}$ becomes $\mathbf{q}$ after $\boldsymbol{T}$ transformation. And $\mathbf{p}$ is in the target point cloud and is the closest point of $\mathbf{q}$. We define the transformation function as follows:

$$\mathbf{q} = \boldsymbol{t}(\boldsymbol{s}) = \boldsymbol{T}\boldsymbol{s} \tag{12}$$

Then the color function for $\mathbf{s}$ is defined as:

$$\mathbf{H}(\mathbf{s}) = \mathbf{C_p}(\mathbf{t}(\mathbf{s})) \approx \mathbf{C}(\mathbf{p}) + \mathbf{g_c}^\top(\mathbf{f_C}(\mathbf{t}(\mathbf{s})) - \mathbf{p}) \tag{13}$$

Then we can define our color objective $\mathbf{E_C} = \sum \parallel \mathbf{r_C} \parallel_2^2$,

$$\begin{aligned}\mathbf{r_C} &= \mathbf{H}(\mathbf{s}) - \mathbf{C}(\mathbf{s}) \\ &\approx \mathbf{C}(\mathbf{p}) + \mathbf{g_c}^\top(\mathbf{f_C}(\mathbf{t}(\mathbf{s})) - \mathbf{p}) - \mathbf{C}(\mathbf{s})\end{aligned} \tag{14}$$

As $\mathbf{H}(\mathbf{s}) = \mathbf{C_p} \circ \mathbf{f_C} \circ \mathbf{t}$, compute the Jacobian matrix:

$$\mathbf{J_C} = \frac{\partial \mathbf{r_C}}{\partial \xi} = \frac{\partial(\mathbf{H}(\mathbf{s}) - \mathbf{C}(\mathbf{s}))}{\partial \xi} = \frac{\partial \mathbf{H}(\mathbf{s})}{\partial \xi} = \frac{\partial \mathbf{C_p}}{\partial \mathbf{f_C}}\frac{\partial \mathbf{f_C}}{\partial t}\frac{\partial t}{\partial \xi} \tag{15}$$

$$\frac{\partial \mathbf{C_p}}{\partial \mathbf{f_C}} = \mathbf{g_c}^\top \qquad \frac{\partial \mathbf{f_C}}{\partial t} = \mathbf{I} - \mathbf{n_p}\mathbf{n_p}^\top \tag{16}$$

$$\begin{aligned}\mathbf{J}_p = \frac{\partial \boldsymbol{t}}{\partial \xi} &= \frac{\partial \boldsymbol{T}\mathbf{p}}{\partial \xi} = \lim_{\xi \to \mathbf{0}} \frac{\left(\exp\left(\xi^\wedge\right) \cdot \boldsymbol{T}\mathbf{p} - \boldsymbol{T}\mathbf{p}\right)}{\xi} \\ &= \begin{bmatrix} -\left[\boldsymbol{T}\mathbf{p}\right]_\times & \mathbf{I}_{3\times3} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3}\end{bmatrix}\end{aligned} \tag{17}$$

Then we have the final Jacobian matrix formula:

$$\mathbf{J_C} = \mathbf{g_c}^\top(\mathbf{I} - \mathbf{n_p}\mathbf{n_p}^\top)\mathbf{J}_p \tag{18}$$

E. Final Optimization Formulation in Odometry Estimation

As we optimize the geometric objective and color objective jointly, we have the overall objective formulation. The Jacobian matrix of the joint objective also can be computed. Then, we can use the Gaussian-Newton method to implement the optimization. Note that the RGB channels are processed separately. We have three color objectives and Jacobians of each channel, and we use E_C and J_C to represent them all.

$$\mathbf{E} = \lambda_G \mathbf{E_G} + \lambda_C \mathbf{E_C} \qquad \mathbf{J} = \lambda_G \mathbf{J_G} + \lambda_C \mathbf{J_C} \tag{19}$$

The odometry estimation uses the historical laser scan p_1, p_2,...,p_{k-1} to calculate the pose of sensor $T \in SE(3)$ in global coordinate. The scan-to-scan matching suffers from drift in the long run as a single laser frame contains less environmental information. Thus, we retain a color map and planar map separately for scan-to-map matching to improve the estimation accuracy. And the sliding window technique is used to ease the computational burden.

F. Mapping

We adopt the mapping module from SSL-SLAM [13]. To avoid a too-large global map, we select keyframes based on the displacement of rotation or translation. And we only add the keyframes to the global map. Note that in the optimization process, we use extracted feature point to reduce the computational cost. However, in the mapping process, we use the origin dense RGB-colored point clouds to build the dense global map.

In addition, we construct an octree to increase the search efficiency. For each cell in octree, we use $P(n \mid z_{1:t})$ to present the probability of the existence of an object [13]:

$$P(n\,|z_{1:t}) = \left[1 + \frac{1 - P(n\,|\,z_t)}{P(n\mid z_t)} \cdot \frac{1 - P(n\,|z_{1:t-1})}{P(n\,|z_{1:t-1})} \cdot \frac{P(n)}{1 - P(n)}\right]^{-1} \tag{20}$$

where z_t is the new measurement, $z_{1:t-1}$ is the old measurements from keyframes, and $P(n)$ is the prior probability, which is set to 0.5 if it is not known.

IV. Experiments

A. Experiment Setup

In this section, we perform extensive experiments to evaluate the proposed method. Our method is evaluated in a room equipped with a motion capture system called NOKOV. It is implemented on a professional and powerful Unmanned Ground Vehicle (UGV) designed by AGILEX company with a size of 930×699×349 (mm), which is widely used in industrial and civil applications. As shown in Fig. 3(b), an Intel RealSense L515 sensor is mounted on a UGV via an aluminum frame. The Intel Realsense L515 is a small FoV solid-state LiDAR with a 70 × 55 viewing angle and 30 Hz update frequency. To further illustrate the robustness, we evaluate the proposed method in handheld mode as shown in Fig. 3(a). The algorithm is coded in C++ and implemented on Ubuntu 20.04 and ROS Noetic [25].

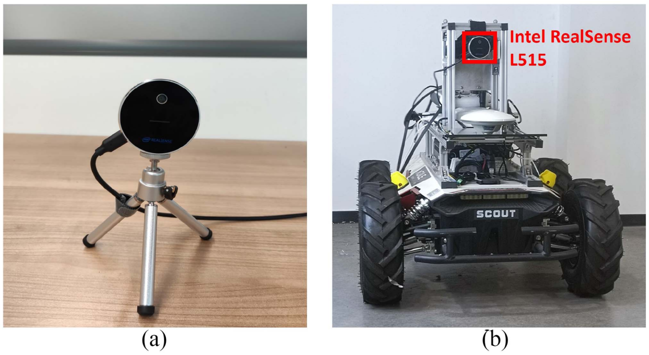

Fig. 3. Our handheld device and UGV platform. (a) The Intel RealSense L515 sensor in handheld settings. (b) The UGV platform mounted with an Intel RealSense L515 sensor.

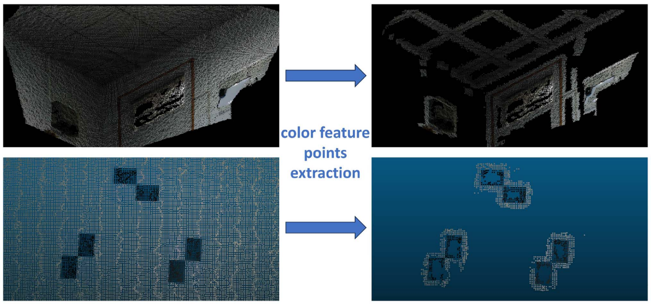

Fig. 4. Extraction of RGB color feature points whose color information changes dramatically.

For the experiment on the UGV and handheld device, a desktop PC computation platform is used, which has an AMD Ryzen 9 7900X 12-Core Processor CPU.

B. Result of RGB Color Feature Extraction

Computing all the color constraints is computationally inefficient as the point cloud is dense. Furthermore, the effective color constraint only takes place where the color variation is obvious. As is described in III. C, the way to measure the color variation is to compute the color gradient $\mathbf{g}_\mathbf{c}^\top$. We investigate the L2 norm of the color gradient to extract the color feature. The feature extraction result can be described in Fig. 4. It is evident that points with slight color variation are removed, and points with significant color variation are retained. In this way, a lot of computational resources can be reserved in the optimization process.

C. Results on Public Benchmark Dataset

In this subsection, we perform experiments on the data from SSL-SLAM [13], where a robot is manually controlled to move around a room of size 4 m × 4 m and finally return to its origin as shown in SSL-SLAM [13]. The length of the path is 12.18 m. And the sensor used to generate the data is Intel RealSense L515. Since our method is odometry with no loop closure, we remove the loop closure module from SSL-SLAM and ORB-SLAM to make a fair comparison. Note that ORB-SLAM supports the RGBD input. So, we generate depth maps from the point cloud to enable ORB-SLAM in our experiments.

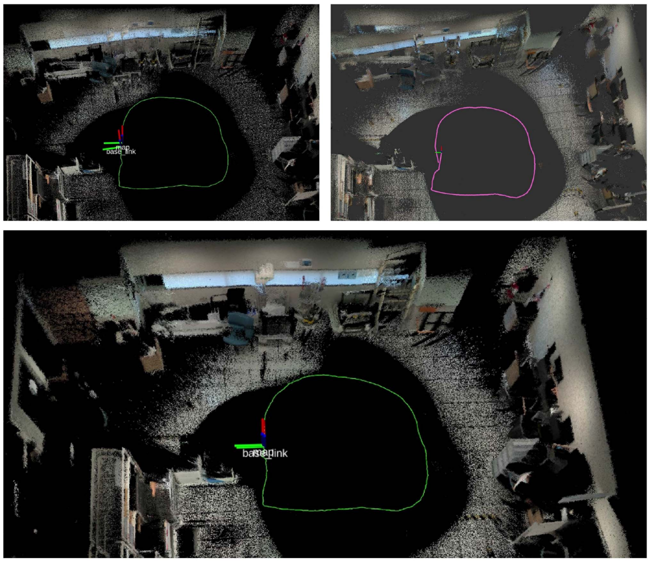

Fig. 5. Comparison on public benchmark dataset. The upleft is the result of SSL-SLAM. The upright is the result of ORB-SLAM3. The bottom is the result of our method.

TABLE I
END-TO-END POSE ERROR ON THE BENCHMARK DATASET

Method	t_x	t_y	t_z	r_x	r_y	r_z
SSL-SLAM	-0.073	0.053	-0.18	-0.027	-0.020	0.11
ORB-SLAM3	0.143	0.273	0.108	1.589	-0.151	0.160
ours	0.016	-0.014	-0.051	0.008	-0.007	0.021

The localization and mapping results are shown in Fig. 5. Note that in this dataset, no localization ground truth is provided. We present the end-to-end translation error as the performance criteria shown in Table I. It is obvious that our method achieves the best localization and mapping results. Compared with SSL-SLAM, we have color information aided. For ORB-SLAM3, providing depth for visual features means the other useful geometry information is wasted. So, the result of ORB-SLAM3 is even worse.

D. Results on Private Dataset of UGV Platform

We compare our method with the localization ground truth to evaluate performance. Since the Intel RealSense L515 will interfere with capture systems such as VICON or NOKOV, we cannot use them to provide ground truth. We use an AprilTag [26] localization system with TagSLAM [27] to produce the ground truth. The accuracy of the AprilTag localization system is validated in the APPENDIX section. We roughly plan the trajectory around the room and manually control the UGV to move along the trajectory, as shown in Fig. 6. The result comparison is shown in Fig. 7. The length of the path is 13.49 m. SSL-SLAM cannot work properly when the sensor is closely facing the wall; thus, the localization result is not fitted with the ground truth, and the mapping result is obviously not correct, as shown in Fig. 6. The absolute pose error (APE) of translation in Table II further proves that our method outperforms other methods, which is only 0.03 m. Specifically, we introduce the result of ORB-SLAM3 with IMU and our works with only one combined color channel,

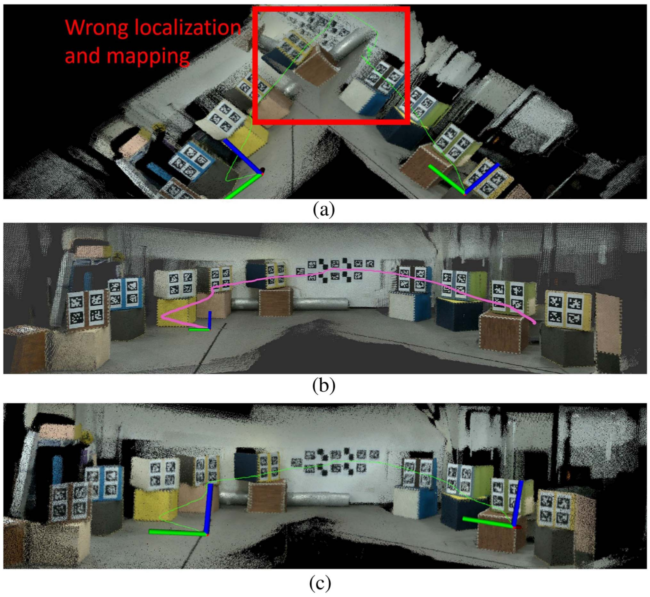

Fig. 6. Localization and mapping results on UGV. The AprilTags are only used to generate ground truth. (a) The localization and mapping result of SSL-SLAM. (b) The result of ORB-SLAM3. (c) The result of our method.

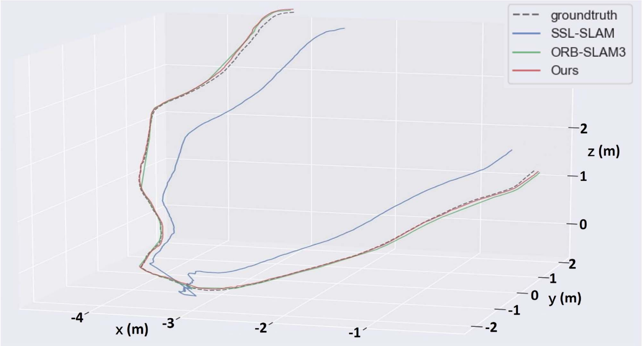

Fig. 7. Localization result comparison on our UGV dataset.

TABLE II
APE of Translation on Our Private Dataset

Method	max	mean	min	rmse	std
SSL-SLAM	0.790	0.419	0.245	0.435	0.117
ORB-SLAM3	0.103	0.038	0.009	0.043	0.020
ORB-SLAM3 with IMU	0.102	0.036	0.009	0.041	0.018
Ours 1-channel	0.103	0.032	0.002	0.037	0.020
Ours 3-channel	0.095	0.030	0.003	0.045	0.018

showing that our method can outperform the IMU-based method and our RGB color fused strategy is better than only one channel.

What is counter-intuitive is that ORB-SLAM3 achieves the best visual effect while our proposed method achieves the best localization performance. One reason is that the ORB-SLAM3 falls into the local minimum, where the visual effect is better, but the z-axis (depth-axis) pose estimation is worse. As shown in Fig. 8, the planes of ORB-SLAM3 mapping are thicker and have more noise. Also, the noise in the visual feature points and depth map association process can cause deviation in the recovered scale of motion. All of those above can lead to the loss of accuracy in global localization.

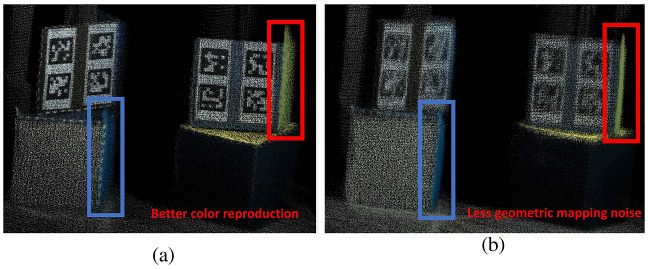

Fig. 8. Mapping details of (a) ORB-SLAM3 method. (b) our method.

TABLE III
APE of Translation on the Data From Handheld Device

Method	t_x	t_y	t_z	t
SSL-SLAM	-0.33	-0.036	0.10	0.35
ORB-SLAM3 (failed)	-0.68	-1.52	-0.75	1.82
ours	-0.19	0.082	0.051	0.21

E. Performance on Handheld Device

The proposed method is also implemented on a handheld device to demonstrate robustness further. Compared to the implementation on the UGV robot, the handheld device suffers from vibration and large viewing angle change, which can cause tracking loss and localization failure [13]. In the experiment, we roughly planned the trajectory around the room in a library. Then, we hold the Intel RealSense L515 along the path at normal walking speed. The localization and mapping result is shown in Fig. 10, with the trajectory plotted in green. As shown in Fig. 10 and Table III, the end-to-end translation error of our method is significantly smaller than other methods. Note that the length of the path is 21.289 m. ORB-SLAM3 failed because the lighting in this scene is very complex. ORB-SLAM3 relies on the visual feature tracking and can not make full use of the geometric information.

F. Performance in Large Scale Scenes

To further demonstrate the performance of our method in large-scale scenes, the proposed method is also evaluated in a library of a long movement. The data is publicly available from SSL-SLAM [13]. It is difficult to obtain the ground truth of localization in large indoor scenes. In this experiment, we can only qualitatively prove our performance in large-scale localization and mapping in Fig. 9 as the mapping results maintain the correct geometrical structures in the scenes.

G. Results Under Varying Lighting Conditions

We can use lighting variation conditions as the visual degradation scene to test our method as it has a fatal effect on visual odometry. The lighting conditions are simulated by the image processing method [28] as shown at the top of Fig. 11. The localization and mapping result of our approach is shown at the bottom of Fig. 11. Note that the data is a part of Section IV-D. Our method achieves accurate localization results with the APE of 0.029 m. ORB-SLAM3 completely fails in this scenario as the visual feature points cannot be extracted and tracked normally in the lighting variation conditions. Our LiDAR-camera fused odometry can utilize geometric structures from laser points to

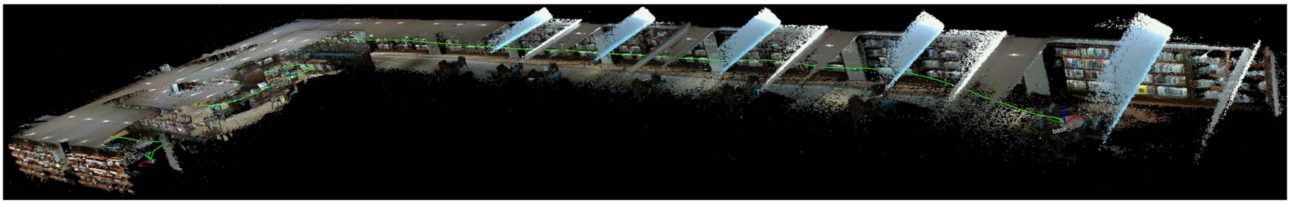

Fig. 9. Localization and mapping results of our method in large-scale library scenes.

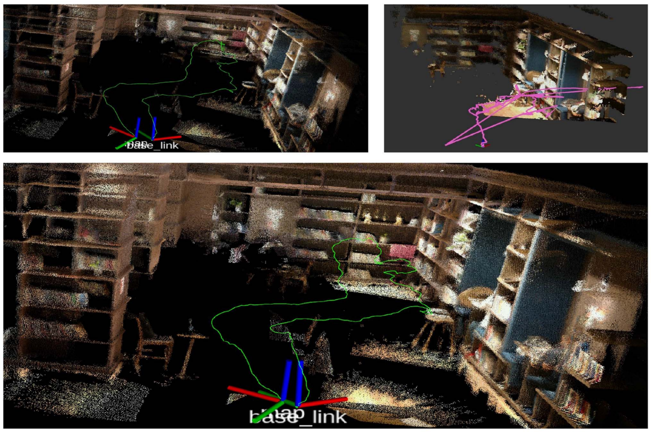

Fig. 10. Comparison of methods on handheld device. The upleft is the result of SSL-SLAM. The upright is the result of ORB-SLAM3. The bottom is the result of our method.

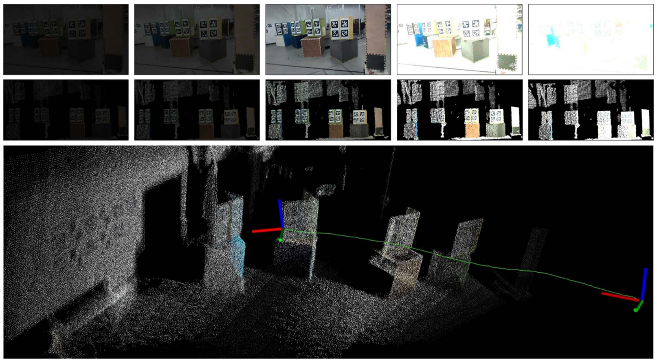

Fig. 11. Localization and mapping result in varying lighting conditions.

produce accurate results when the vision information is not reliable.

H. Results in Pure LiDAR Degenerated Scene

The most innovative of our work is that we fuse the color information from the camera to promote the result of LiDAR SLAM. Here, we perform an ablation study to show that the color information helps to promote the performance in our work, especially in a challenging LiDAR degenerated scene. The experiment setup and results are shown in Fig. 12. Our experimental environment is a white wall plane with black rectangle markers. With only this wall plane, the pure LiDAR SLAM will fail because the geometric constraint is insufficient. The results show that our method works well in this degenerate situation. Correct localization and mapping results can be obtained even under complex motions. This demonstrates that in LiDAR degraded scenes our LiDAR-camera fusion method can utilize vision information to produce accurate result.

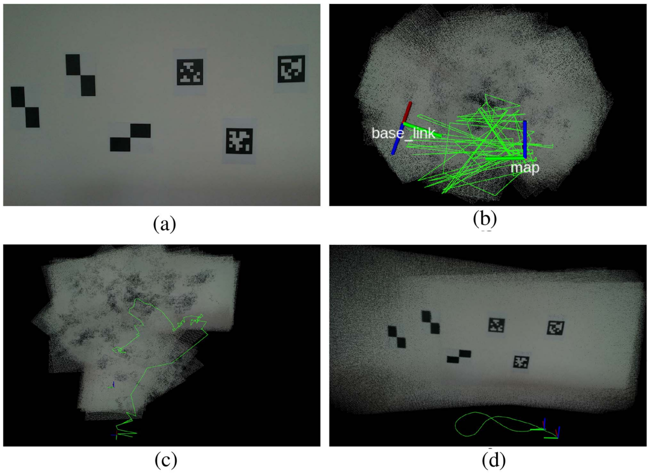

Fig. 12. Ablation study in pure LiDAR Degenerated scenes. (a) The experiment setup with a white wall and some markers. (b) The localization and mapping result of SSL-SLAM. (c) The result of our method without color information. (d) The result of our method with color information.

V. Conclusion

In this letter, we present a LiDAR-camera fused SLAM framework for the dense colored point cloud. Our framework mainly consists of a feature extraction module, an odometry estimation module, and a probability map construction module. Extensive experiments have been conducted to validate our proposed method, including experiments on the UGV platform, the handheld mobile device, and the public datasets. The experiments demonstrate the robustness and accuracy of our method in localization and mapping results. The fusion of LiDAR and camera is effective as our method successfully maintains high performance in challenging environments.

Appendix

As we mentioned before, Intel RealSense L515 LiDAR and the motion capture system NOKOV have the same laser wavelength. They cannot work together as they will interfere with each other. So, we use AprilTag and TagSLAM to generate the localization ground truth in our experiment. Here, we should validate the accuracy of the AprilTag localization system. In this way, we can prove that using the AprilTag localization system as the ground truth is reasonable [29]. Shown in Fig. 13 is the experimental site where the AprilTag markers are installed. The

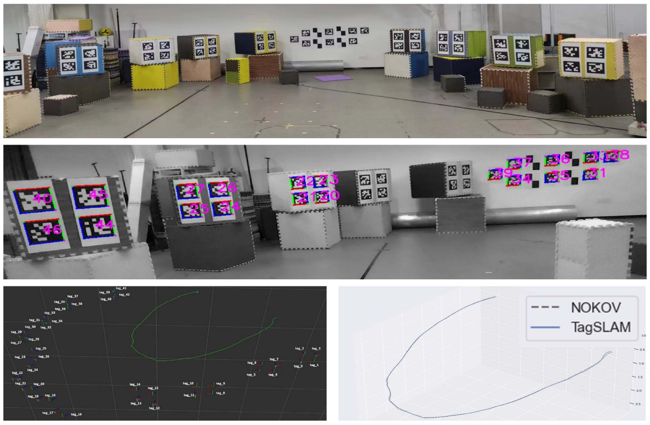

Fig. 13. Upper is the our experiment site with AprilTags. The middle is the AprilTag detection result in TagSLAM. The bottom left is the pose of AprilTags and the localization result from TagSLAM. The bottom right is the trajectory comparison of TagSLAM and NOKOV.

TABLE IV
APE of Translation Part on AprilTag Localization System

Method	max	mean	median	min	rmse	std
TagSLAM	0.046	0.018	0.017	0.0017	0.02	0.01

TABLE V
Mean APE of Different Measurement

Measurement	1	2	3	4	5	average
Mean APE	0.021	0.018	0.024	0.017	0.026	0.021

Apriltags will be detected using the camera on Intel RealSense L515 and produce the camera pose.

One typical result is shown in Fig. 13. It can be seen that the trajectory of the TagSLAM and NOKOV fit well. The total length of the path is 10.05 m. Note that the accuracy of the NOKOV reaches the sub-millimeter level. With the NOKOV as the reference, the APE of the TagSLAM is shown in Table IV. The mean error is 0.018 m. We repeat this experiment and get the average APE in Table V, which is 0.021 m. Thus, we validate that the AprilTag localization system can provide reasonable ground truth.

References

[1] X. Yan, Y. Bi, Z. Gao, M. Lao, R. Teo, and F. Lin, "Multi-plane visual odometry for unmanned aerial vehicle using a thermal camera," in *Proc. IEEE 14th Int. Conf. Control Automat.*, 2018, pp. 588–593.

[2] B. Fahima and N. Abdelkrim, "Multispectral visual odometry using SVSF for mobile robot localization," *Unmanned Syst.*, vol. 10, no. 03, pp. 273–288, 2022.

[3] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," in *Proc. Conf. Robot.: Sci. Syst.*, 2014, vol. 2, pp. 1–9.

[4] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4758–4765.

[5] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5135–5142.

[6] X. Yan, Y. Bi, Z. Gao, and F. Lin, "Laser-aided infrared visual inertial odometry for unmanned aerial vehicle with multi-plane constraint," in *Proc. IEEE 3rd Int. Conf. Adv. Robot. Mechatron.*, 2018, pp. 323–327.

[7] Y.-S. Shin, Y. S. Park, and A. Kim, "Direct visual SLAM using sparse depth for camera-LiDAR system," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 5144–5151.

[8] J. Graeter, A. Wilczynski, and M. Lauer, "LIMO: Lidar-monocular visual odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 7872–7879.

[9] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: Low-drift, robust, and fast," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 2174–2181.

[10] W. Wang, J. Liu, C. Wang, B. Luo, and C. Zhang, "DV-LOAM: Direct visual lidar odometry and mapping," *Remote Sens.*, vol. 13, no. 16, 2021, Art. no. 3340.

[11] J. Lin and F. Zhang, "R3LIVE++: A robust, real-time, radiance reconstruction package with a tightly-coupled LiDAR-inertial-visual state estimator," 2022, *arXiv:2209.03666*.

[12] T. Shan, B. Englot, C. Ratti, and D. Rus, "LVI-SAM: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 5692–5698.

[13] H. Wang, C. Wang, and L. Xie, "Lightweight 3-D localization and mapping for solid-state LiDAR," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 1801–1807, Apr. 2021.

[14] J. Zhang, M. Kaess, and S. Singh, "Real-time depth enhanced monocular odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 4973–4980.

[15] 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. Automat.*, 2021, pp. 5049–5055.

[16] C. Campos, R. Elvira, J. J. G. Rodríguez, J. 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.

[17] Y. Jia et al., "Lvio-fusion: A self-adaptive multi-sensor fusion SLAM framework using actor-critic method," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 286–293.

[18] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "LIC-fusion: LiDAR-inertial-camera odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 5848–5854.

[19] X. Zuo et al., "LIC-fusion 2.0: LiDAR-inertial-camera odometry with sliding-window plane-feature tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5112–5119.

[20] F. Dellaert et al., "Factor graphs for robot perception," *Found. Trends Robot.*, vol. 6, no. 1/2, pp. 1–139, 2017.

[21] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3D lidar inertial odometry and mapping," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 3144–3150.

[22] J. Lin and F. Zhang, "R^{3}LIVE: A robust, real-time, RGB-colored, LiDAR-inertial-visual tightly-coupled state estimation and mapping package," in *Proc. Int. Conf. Robot. Automat.*, 2022, pp. 10672–10678.

[23] C. Zheng, Q. Zhu, W. Xu, X. Liu, Q. Guo, and F. Zhang, "FAST-LIVO: Fast and tightly-coupled sparse-direct LiDAR-inertial-visual odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 4003–4009.

[24] J. Park, Q.-Y. Zhou, and V. Koltun, "Colored point cloud registration revisited," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 143–152.

[25] M. Quigley et al., "ROS: An open-source robot operating system," in *Proc. ICRA Eorkshop Open Source Softw.*, 2009, vol. 3, Art. no. 5.

[26] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 3400–3407.

[27] B. Pfrommer and K. Daniilidis, "TagSLAM: Robust SLAM with fiducial markers," 2019, *arXiv:1910.00679*.

[28] Z. Gao, X. Zhao, M. Cao, Z. Li, K. Liu, and B. M. Chen, "Synergizing low rank representation and deep learning for automatic pavement crack detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 10, pp. 10676–10690, Oct. 2023.

[29] Y. Wan, Z. Liao, J. Liu, W. Song, H. Ji, and Z. Gao, "Small object detection leveraging density-aware scale adaptation," *Photogrammetric Rec.*, vol. 38, no. 182, pp. 160–175, 2023.