Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Novel robust simultaneous localization and mapping for long-term autonomous robots*

Wei WEI[1], Xiaorui ZHU‡[1,2], Yi WANG[1]

[1]*School of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen 518055, China*

[2]*Zhuhai Big Data Research Institute, Zhuhai 519000, China*

E-mail: weirui9003@gmail.com; xiaoruizhu@hit.edu.cn; wangyi601@aliyun.com

Received July 18, 2020; Revision accepted Nov. 18, 2020; Crosschecked Sept. 18, 2021; Published online Feb. 5, 2022

Abstract: A fundamental task for mobile robots is simultaneous localization and mapping (SLAM). Moreover, long-term robustness is an important property for SLAM. When vehicles or robots steer fast or steer in certain scenarios, such as low-texture environments, long corridors, tunnels, or other duplicated structural environments, most SLAM systems might fail. In this paper, we propose a novel robust visual inertial light detection and ranging (LiDaR) navigation (VILN) SLAM system, including stereo visual-inertial LiDaR odometry and visual-LiDaR loop closure. The proposed VILN SLAM system can perform well with low drift after long-term experiments, even when the LiDaR or visual measurements are degraded occasionally in complex scenes. Extensive experimental results show that the robustness has been greatly improved in various scenarios compared to state-of-the-art SLAM systems.

Key words: Simultaneous localization and mapping (SLAM); Long-term; Robustness; Light detection and ranging (LiDaR); Visual inertial LiDaR navigation (VILN)

https://doi.org/10.1631/FITEE.2000358 **CLC number:** TP399

1 Introduction

Simultaneous localization and mapping (SLAM) is still a challenging problem for long-term autonomous mobile robots because the real world is full of highly dynamic, unstructured, and complex scenarios. In recent decades, numerous outstanding SLAM frameworks have been developed. For sparse visual SLAM, MonoSLAM (Davison et al., 2007) is the first real-time mono-SLAM system that is based on extended Kalman filter (EKF). Parallel tracking and mapping (PTAM) (Klein and Murray, 2007) is the first SLAM system that features parallel tracking and mapping. It adopts, for the first time, bundle adjustment to optimize and implement the concept of keyframes. For semi-dense visual SLAM, Engel et al. (2014) proposed a novel direct tracking method, namely large-scale direct monocular SLAM (LSD-SLAM), which operates on Lie algebra and the direct method. SVO (Forster et al., 2014) is a technique of semi-direct visual odometry. It uses sparse model-based image alignment to obtain a high speed. Engel et al. (2018) proposed direct sparse odometry (DSO), a direct and sparse method that does not detect any feature points. For dense visual SLAM, dense tracking and mapping (DTAM) (Newcombe et al., 2011) is a technique involving a novel non-convex optimization framework that reconstructs a three-dimensional (3D) model in real time. Kerl et al. (2013) proposed a dense visual SLAM method for RGB-D cameras, namely DVO, which uses an entropy-based similarity measure for keyframe selection and loop closure detection based on the g2o framework. For the

‡ Corresponding author

* Project supported by the National Key R&D Program of China (No. 2018YFB1305500) and the National Natural Science Foundation of China (No. U1813219)

ORCID: Wei WEI, https://orcid.org/0000-0002-8998-045X; Xiaorui ZHU, https://orcid.org/0000-0003-1400-059X

light detection and ranging (LiDaR) SLAM framework, Gmapping (Grisetti et al., 2007) is a SLAM system based on Rao-Blackwellization particle filter (RBPF), which is the most used SLAM package in robots. KartoSLAM (Konolige et al., 2010) is a graph-based SLAM system, which uses large pose-graph optimization called sparse pose adjustment (SPA). Deschaud (2018) presented a scan-to-model matching framework, implicit moving least squares (IMLS)-SLAM, which yields low-drift results and uses only 3D LiDaR data. For LiDaR and visual fusion SLAM systems, Xu et al. (2018) proposed a robust indoor SLAM by switching mode and data fusion, which uses RGB-D cameras and two-dimensional (2D) low-cost LiDaR measurements. However, none of the above approaches have been able to solve the problem of long-term robust running in certain scenarios, such as low-texture environments, long corridors, tunnels, or other duplicated structural environments.

Specifically for long-term robustness, Zhao HJ et al. (2008) proposed a method of SLAM in a dynamic large outdoor environment using a laser scanner. They found that the method is very time-consuming for tracking many static or moving objects. Sünderhauf et al. (2013) described a novel concept of learning to predict systematic changes in the appearance of environments and then, by using this learned knowledge, to predict its appearance under different environmental conditions. However, it requires the learning of different vocabularies for discrete sets of environmental conditions. Zhang and Singh (2015) proposed a general framework, termed vision-LiDaR odometry and mapping (V-LOAM), which combines visual odometry and LiDaR odometry. The method starts with visual odometry and fine-tunes motion estimation and point cloud registration at the same time by scan matching based LiDaR odometry, which does not perform well for continuous darkness. Further, it uses consecutive frames to estimate poses, which can hardly build maps for reuse or long-term running. Hemann et al. (2016) have presented a method that tightly couples the measurements of the inertial measurement unit (IMU) and accumulates the LiDaR heightmap in the form of an error-state Kalman filter. However, duplicated structural environments, such as long corridors or tunnels, might cause long-term robustness problems. ORB-SLAM2 (Mur-Artal and Tardós, 2017), which is a complete SLAM system for monocular, stereo, and RGB-D cameras, might not run for a long period in low-texture environments. A cloud-based real-time outsourcing localization architecture has been proposed by Zhu et al. (2017) to allow a ground mobile robot to identify its location relative to a road network map and reference images in the cloud, which heavily depends on the quality of the network. Banerjee et al. (2019) proposed a method for pruning views in a visual SLAM system to maintain its speed and accuracy for long-term use. However, the rapid steering of vehicles or robots with visual sensors might cause the images to be blurred. Moreover, the visual sensors might be over-exposed by sudden light changes, e.g., when the vehicles come out of a tunnel. These complex situations might hinder the system from long-term running. Shao et al. (2019) presented a stereo visual inertial LiDaR SLAM using the iterative closest point (ICP) to refine loop closure, which can easily fail when the initial parameters are not appropriate. Kim et al. (2019) presented a robust year-round localization performance even when the learning occurs in just a single day. However, in new scenarios, the algorithm needs to learn the whole environment again.

Recently, to solve the long-term SLAM problem in dynamic complex scenarios, some researchers have studied deep learning based technologies that could introduce semantic information for the SLAM system. Zhao ZR et al. (2019) labeled the point clouds with semantic segmentation information, but there was no improvement in accuracy and long-term robustness. Patel et al. (2019) used semantically enhanced feature matching and visual-inertial bundle adjustment to improve the long-term robustness of odometry, especially in feature-sparse environments, but this is not suitable for outdoor environments. Nair et al. (2020) designed a monocular multi-body SLAM system to perform dynamic multi-object and ego localization in a unified framework in metric scale. However, the end-to-end network structure is not conducive to the integration of heterogeneous data (Wang et al., 2020) to improve the long-term robustness, and most deep learning SLAM systems (e.g., those in Lee et al. (2020) and Wagstaff et al. (2020)) can hardly balance accuracy, efficiency, and long-term running compared with traditional SLAM systems. Therefore, in this study, a novel SLAM

system is proposed to achieve long-term running and high robustness, as well as good accuracy in various real-time scenarios.

In this paper, we propose a robust visual inertial LiDaR navigation (VILN) SLAM framework, which includes tightly coupled stereo visual-IMU, loosely coupled LiDaR odometry, a LiDaR enhanced visual loop closure system, and a LiDaR and vision fused map. By combining the LiDaR point cloud and image feature (oriented FAST (features from accelerated segment test), rotated BRIEF (binary robust independent elementary features), or ORB (oriented FAST and rotated BRIEF)) points as constraints, we use the graph optimization method to optimize the robot pose. Meanwhile, the bag of words (BoW) based on ORB features and the grid scan-match based on the LiDaR point cloud are involved in loop closure detection, and then the LiDaR grid map is further optimized. In the end, a fused map including both the LiDaR grid and visual features is established which could perform well with low drift after long-term experiments based on relocalization.

The main contributions of our work are as follows:

1. A novel robust VILN SLAM framework is proposed. It provides fused loop closure correction and real-time accurate state estimation.

2. A robust visual-IMU-LiDaR odometry is proposed. It provides long-term robustness in low-texture environments and duplicated structural environments. Moreover, it is validated when the vehicle or robot steers fast or the visual sensors are overexposed by light changes.

3. A LiDaR enhanced visual loop closure system is presented. It uses visual BoW and LiDaR scan for mapping, which is more robust than ICP and can detect and correct the loop even in low-texture environments or duplicated structural environments. It will reduce the cumulative error, improve the positioning accuracy, and guarantee robust and long-term running.

4. A LiDaR and vision fused map is established which contains only the ORB feature points and LiDaR submap and can easily run for a long term with low drift by relocalization.

2 VILN SLAM framework

We develop our framework under tightly coupled visual-IMU and loosely coupled LiDaR fusion. We use the raw IMU measurements and preintegration method with visual and LiDaR measurements to optimize the states of the whole system, which could work well in vision or LiDaR degraded cases or when the motions are fast. To the best of our knowledge, our framework is one of the few visual-IMU-LiDaR fusion algorithms that can run in complex environments and exhibit long-term robustness.

VILN SLAM consists of three major systems, i.e., stereo visual inertial odometry (VIO), a visual-LiDaR mapping system, and a LiDaR enhanced visual loop closure system. Fig. 1 shows the complete framework of VILN SLAM. The stereo VIO consists of a visual frontend and a backend optimizer. The visual frontend performs frame-to-frame tracking and stereo matching and then outputs stereo matches

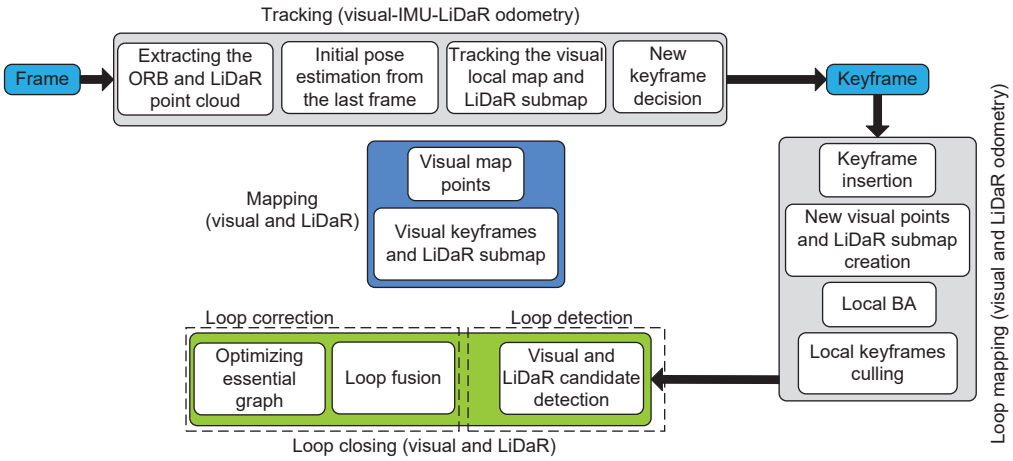


Fig. 1 Overview of the VILN SLAM framework

as visual measurements. The backend optimizer takes the stereo matches and IMU measurements and performs both IMU pre-integration and tightly coupled smoothing over a pose graph. When VILN SLAM is configured to perform LiDaR feedback, the pose graph has one additional constraint added, which is the pose between the factors formulated from the LiDaR mapping poses. The VIO backend optimizer outputs the pose estimate at both the IMU rate and camera rate in real time. In addition, we use a novel scan-matching method in the LiDaR mapping system. It uses the motion estimate from the VIO and performs LiDaR scan to map registration. The LiDaR enhanced visual loop closure system conducts visual loop detection and initial loop constraint estimation, which is further validated by random sample consensus (RANSAC) geometric verification and refined by point cloud alignment. A global pose graph constraining all LiDaR poses is optimized incrementally to obtain a globally corrected trajectory and LiDaR pose correction in real time once there is a loop. These optimized poses are sent back to the LiDaR mapping module for map update and relocalization. In the post-processing stage, using the LiDaR scans relative to the best LiDaR pose estimate yields the mapping results.

3 Stereo visual inertial odometry

3.1 Hybrid visual frontend

Visual frontend performs frame-to-frame feature tracking and stereo matching for generating a set of stereo-matched sparse feature points, namely, stereo matches. The frame-to-frame tracking performance directly affects the quality of temporal constraints. Stereo matching is important for establishing the system and generating high-quality matches to constrain the scale. These two tasks are important for the stereo visual frontend. Traditionally, the feature-based visual frontend performs both of the tasks in the descriptor space. However, we observe that this method is sensitive to parameter tuning and is time-consuming. More importantly, this method does not use the prior information (the previous frame) in the tracking task. We use the Kanade–Lucas–Tomasi (KLT) feature tracker (Forster et al., 2014) to track all the feature points of the previous stereo matches, in either the left or the right image. We have a tracked stereo match, and it is pushed into the output when they are both tracked. We still use feature-based methods for the stereo-matching task, as they are better while handling large baselines than KLT. Hence, the system that combines the direct and feature-based methods becomes a hybrid of the two.

3.2 Backend optimizer

Providing real-time locally consistent state estimate at a relatively high frequency is the goal of the backend optimizer, which will be served as the motion model for the LiDaR mapping algorithm. It will be a good trade-off between accuracy and efficiency when we use a tightly coupled fixed-lag smoother to operate on a pose graph. Since a fixed-lag pose graph optimizer bounds the maximum number of variables, the computation cost is bounded. Another advantage of formulating the problem as a pose graph optimization problem is that it unifies different kinds of observations into the factor representation. This simplifies the procedure of adding new sensor inputs or constraints into the optimization problem.

The proposed VIO has the IMU pre-integration factor and structureless vision factor as constraints. The pose-graph formulation is shown in Fig. 2. Variables to be optimized are the states inside the window. Denote $\boldsymbol{S}_t$ as the state variable at the stereo frame time t:

$$\boldsymbol{S}_t \doteq [\xi_t, \boldsymbol{v}_t, \boldsymbol{b}_t^{\mathrm{a}}, \boldsymbol{b}_t^{\mathrm{g}}], \tag{1}$$

where ξ_t is the six-degree-of-freedom (6-DoF) system pose (IMU-centered robot pose at time t), $\boldsymbol{v}_t$ is the associated linear velocity, and $\boldsymbol{b}_t^{\mathrm{a}}$ and $\boldsymbol{b}_t^{\mathrm{g}}$ are the accelerometer bias and gyroscope bias, respectively. The window of state variables being estimated is of the most recent N stereo frames. Past state variables are marginalized, producing prior factors on related variables.

3.2.1 IMU pre-integration factor

We generate relative IMU measurements between $\boldsymbol{S}_i$ and $\boldsymbol{S}_j$ through the IMU pre-integration method (Forster et al., 2017). Optimization of relinearization can be performed efficiently by IMU pre-integration. We denote $\boldsymbol{r}_{ij}^{\mathrm{I}}$ as the residual of the IMU pre-integration factor, which contains three terms: the residual of pose $\boldsymbol{r}_{\triangle\xi_{ij}}$, velocity $\boldsymbol{r}_{\triangle v_{ij}}$, and bias $\boldsymbol{r}_{\triangle b_{ij}}$.

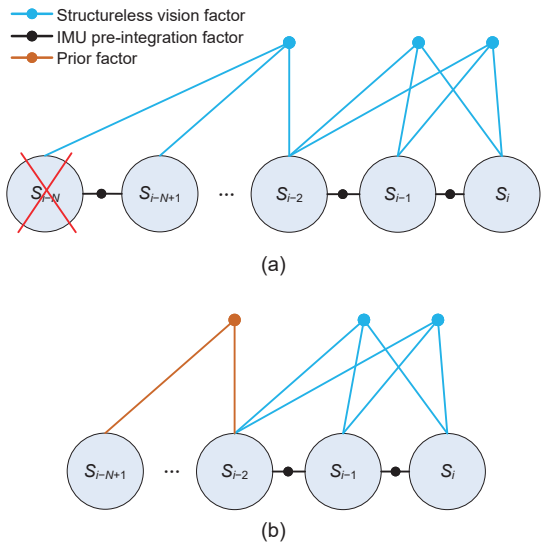


Fig. 2 Pose-graph formulation in the visual inertial odometry: (a) the state to be marginalized is crossed; (b) after marginalization, prior factors are added back on the related variables (Shao et al., 2019). References to color refer to the online version of this figure

3.2.2 Structureless vision factor

We model visual measurements in a structureless fashion, as in the work of Shao et al. (2019). The benefits are two-fold. First, the computational cost is bounded, since the variable size is bounded to be the sliding window size at any point in time. Second, it is easier to manage the landmark variables. Consider a landmark p, whose position in the global frame is $\boldsymbol{x}_p \in \mathbb{R}^3$, which is observed by multiple states. Denote the set of states observing p as $\{\boldsymbol{S}\}_p$. For any state $\boldsymbol{S}_k$ in $\{\boldsymbol{S}\}_p$, denote the residual formed by measuring p in the left camera image as $\boldsymbol{r}^{\mathrm{V}}_{\xi_{k,\mathrm{lc}},p}$:

$$\boldsymbol{r}^{\mathrm{V}}_{\xi_{k,\mathrm{lc}},p} = \boldsymbol{z}_{\xi_{k,\mathrm{lc}},p} - h(\xi_{k,\mathrm{lc}}, \boldsymbol{x}_p), \tag{2}$$

where $\xi_{k,\mathrm{lc}}$ is the left camera pose, obtained by applying an IMU-camera transformation to ξ_k, $\boldsymbol{z}_{\xi_{k,\mathrm{lc}},p}$ is the pixel measurement of p in the image, and $h(\xi_{k,\mathrm{lc}}, \boldsymbol{x}_p)$ encodes a perspective projection. We derive the residual for the right camera image in the same way. Since an iterative method is required for optimizing the pose graph, it is necessary to linearize the above residual. The following equation shows the linearized residuals for landmark p:

$$\sum_{\boldsymbol{S}_p} \|\boldsymbol{F}_{kp}\delta\xi_k + \boldsymbol{E}_{kp}\delta\boldsymbol{x}_p + \boldsymbol{b}_{kp}\|^2, \tag{3}$$

where the Jacobians $\boldsymbol{F}_{kp}$, $\boldsymbol{E}_{kp}$, and the residual error $\boldsymbol{b}_{kp}$ are the results from linearization and are normalized by $\Sigma_C^{1/2}$, the visual measurement covariance, and δ represents the tangent space. Then, we have the following:

$$\|\boldsymbol{r}^{\mathrm{V}}_p\|^2_{\Sigma_C} = \|\boldsymbol{F}_p\delta\xi_k + \boldsymbol{E}_p\delta\boldsymbol{x}_p + \boldsymbol{b}_p\|^2. \tag{4}$$

To avoid optimizing over $\boldsymbol{x}_p$, the residual is projected into the null space of $\boldsymbol{E}_p$: Premultiply each term by $\boldsymbol{Q}_p \doteq \boldsymbol{I} - \boldsymbol{E}_p(\boldsymbol{E}_p^{\mathrm{T}}\boldsymbol{E}_p)^{-1}\boldsymbol{E}_p^{\mathrm{T}}$, an orthogonal projector of $\boldsymbol{E}_p$. Then we have the structureless vision factor for landmark p as follows:

$$\|\boldsymbol{r}^{\mathrm{V}}_p\|^2_{\Sigma_C} = \|\boldsymbol{Q}_p\boldsymbol{F}_p\delta\xi_k + \boldsymbol{Q}_p\boldsymbol{b}_p\|^2. \tag{5}$$

3.2.3 Optimization and marginalization

The pose graph optimization is a maximum a posteriori (MAP) problem, whose optimal solution is

$$S^*_W = \arg\min_{S^*_W}(\|\boldsymbol{r}_0\|^2_{\Sigma_0} + \sum_{i\in W}\|\boldsymbol{r}^{\mathrm{I}}_{i(i+1)}\|^2_{\Sigma_{\mathrm{I}}} + \sum_p \|\boldsymbol{r}^{\mathrm{V}}_p\|^2_{\Sigma_C}), \tag{6}$$

where S^*_W is the set of state variables inside the window, $\boldsymbol{r}_0$ and Σ_0 are the prior factors and their associated covariance, respectively, and Σ_{I} is the covariance of the IMU measurements. The initial guess is given by the frontend pose estimation in Section 3.1, and the Levenberg–Marquart (LM) optimizer and Schur-complement marginalization (Sibley et al., 2010) are applied to solve the nonlinear optimization problem.

4 Visual LiDaR fused map and loop closure

In this part, we introduce a visual-IMU-LiDaR fused mapping method, as well as a fast loop closing approach via the map. We use the VIO poses output from a high-rate IMU as the motion prior to performing 3D LiDaR scan to map registration.

4.1 LiDaR scan distortion correction

First, since LiDaR scan points are time-stamped differently, we need to correct the motion distortion of 3D LiDaR point clouds. Denote any time within a scan as t_i. We correct the distortion of all points to the time of end of scan t_{k+1} based on the IMU rate

VIO poses. Denote a LiDaR point at t_i as $\boldsymbol{P}_i$ and the undistorted point itself as $\tilde{\boldsymbol{P}}_i$; we have

$$\tilde{\boldsymbol{P}}_i = (\boldsymbol{T}_{k+1}^{\mathrm{L}})^{-1}\boldsymbol{T}_i^{\mathrm{L}}\boldsymbol{P}_i, \tag{7}$$

where $\boldsymbol{T}_{k+1}^{\mathrm{L}}$ and $\boldsymbol{T}_i^{\mathrm{L}}$ are pose transformation matrices of the $(k+1)^{\mathrm{th}}$ and i^{th} frames of LiDaR from the closest VIO poses of the high-rate IMU, respectively.

4.2 Scans and submaps

In our approach, each consecutive scan is matched against a small chunk of the world, called a submap M, using a nonlinear optimization that aligns the scan with the submap. We build a different submap from the best submap building method so far (Hess et al., 2016). Submap construction is an iterative process of repeatedly aligning scan and submap coordinate frames, further referred to as frames. With the origin of the scan at $\mathbf{0} \in \mathbb{R}^3$, we now write the information about the scan points as $H = \{\boldsymbol{h}_k\}_{k=1,2,\ldots,K}, \boldsymbol{h}_k \in \mathbb{R}^3$. The pose ξ of the scan frame in the submap frame is represented as the transformation $\boldsymbol{T}_\xi$, which rigidly transforms the scan points from the scan frame into the submap frame, defined as follows:

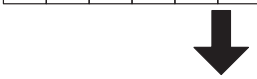
$$\boldsymbol{T}_\xi \boldsymbol{p} = \boldsymbol{R}_\xi \boldsymbol{p} + \boldsymbol{t}_\xi, \tag{8}$$

where $\boldsymbol{R}_\xi$ is a 3×3 rotation matrix and $\boldsymbol{t}_\xi$ is a 3×1 translation vector.

A few consecutive scans are used to build a submap. These submaps take the form of Euclidean distance (Maurer et al., 2003) grids $M : r\mathbb{Z} \times r\mathbb{Z} \times r\mathbb{Z} \to [0, d_{\max}]$, which maps from discrete grid points at a given resolution r, e.g., 0.05 m, to values.

Whenever a scan is to be inserted into the Euclidean distance grid, the relative Euclidean distance submap needs to be computed once. We call a discrete grid a 3D binary image, which is a function I from the elements of an $n_1 \times n_2 \times n_3$ array to {0,1}, where "1" represents that at least one point is in the $r \times r \times r$ grid (voxel). Voxels of value "0" and "1" are called background voxels and foreground feature voxels (FVs), respectively. For the Euclidean distance metric, the distance transform (DT) of an image I is an assignment to each voxel $\boldsymbol{x}$ of the distance between $\boldsymbol{x}$ and the closest feature voxel (CFV) in I. For example, in Fig. 3 for the 2D case, we transform a 10×10 binary image to values in $[0, 5]$. In addition, in Fig. 4, we transform a reference scan to the Euclidean distance grid whose values are in $[0, 10]$.

0	**1**	0	0	0	0	0	0	**1**	0
0	0	0	0	0	0	**1**	0	0	0
0	0	0	0	0	0	0	**1**	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	**1**	0	**1**
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	**1**	0	0	0	0	0	0	0	0



1	**0**	1	2	2.236	1.414	1	1	**0**	1
1.414	1	1.414	2.236	2	1	**0**	1	1	1.414
2.236	2	2.236	2.828	2.236	1.414	1	**0**	1	2
3	3	3.162	3.606	2.828	2.236	1.414	1	1.414	1
2	2.236	2.828	3.162	3	2	1	**0**	1	**0**
1	1.414	2.236	2.236	2	2.236	1.414	1	1.414	1
0	1	2	1.414	1	1.414	2.236	2	2.236	2
1	1.414	2	1	**0**	1	2	3	3.162	3
1.414	1	1.414	1.414	1	1.414	2.236	3.162	4.123	4
1	**0**	1	2	2	2.236	2.828	3.606	4.472	5

Fig. 3 Transforming a binary image to the Euclidean distance grid

4.3 Three-dimensional Euclidean distance scan matching

The scan pose ξ is optimized relative to the current local submap using a novel scan matcher, which we call the Euclidean distance scan matcher. The scan matcher is responsible for finding a scan pose that minimizes the Euclidean distance in the submap. We cast this as a nonlinear least squares problem:

$$\arg\min_{\xi} \sum_{k=1}^{K} M(\boldsymbol{T}_\xi \boldsymbol{h}_k), \tag{9}$$

where $\boldsymbol{T}_\xi$ transforms $\boldsymbol{h}_k$ from the scan frame to the submap frame according to the scan pose. The func-

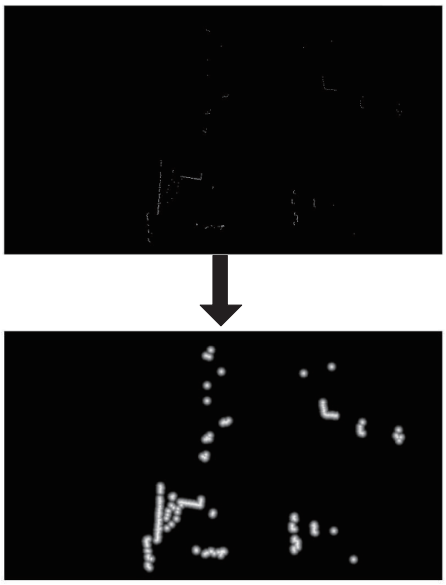


Fig. 4 The reference 2D scan (top) and the resulting Euclidean distance grid ($r = 0.05$ m, $d_{\max} = 10$) (bottom)

tion $M : \mathbb{R}^3 \to \mathbb{R}$ is the value in the local Euclidean distance submap of the scan point.

This is a local optimization; hence, good initial estimates are required. A VIO in Section 3 can be used to estimate the pose between LiDaR scan matches. Now we illustrate the scan matching method in detail. We are interested in the optimal, pixel-accurate match:

$$\xi^* = \arg\min_{\xi\in\mathcal{W}} \sum_{k=1}^{K} M_{\text{nearest}}(\boldsymbol{T}_\xi \boldsymbol{h}_k), \tag{10}$$

where $\mathcal{W}$ is the search window and M_{nearest} is M extended to all of $\mathbb{R}^3$ by rounding its arguments to the nearest grid point first, i.e., extending the value of a grid point to the corresponding pixel.

We compute an integral number of steps that could cover the given linear and angular search window sizes, e.g., $W_x = W_y = W_z = 0.2$ m, $W_{r_x} = W_{r_y} = W_{r_z} = 4°$, and the angular step size $\delta_\theta = 0.2°$.

$$\begin{aligned} w_x = \frac{W_x}{r}, w_y = \frac{W_y}{r}, w_z = \frac{W_z}{r}, \\ w_{r_x} = \frac{W_{r_x}}{\delta_\theta}, w_{r_y} = \frac{W_{r_y}}{\delta_\theta}, w_{r_z} = \frac{W_{r_z}}{\delta_\theta}, \end{aligned} \tag{11}$$

where r_x, r_y, and r_z are the roll, pitch, and yaw angles, respectively. In addition, a finite search window set $\mathcal{W}$ is created around an estimate ξ_0 placed in its center:

$$\begin{aligned} \overline{\mathcal{W}} &= \{-w_x, \dots, w_x\} \cdot \{-w_y, \dots, w_y\} \\ &\quad \cdot \{-w_z, \dots, w_z\} \cdot \{-w_{r_x}, \dots, w_{r_x}\} \\ &\quad \cdot \{-w_{r_y}, \dots, w_{r_y}\} \cdot \{-w_{r_z}, \dots, w_{r_z}\}, \\ \mathcal{W} &= \{\xi_0 + (rj_x, rj_y, rj_z, \delta_\theta j_{r_x}, \delta_\theta j_{r_y}, \delta_\theta j_{r_z}) : \\ &\quad (j_x, j_y, j_z, j_{r_x}, j_{r_y}, j_{r_z}) \in \overline{\mathcal{W}}\}. \end{aligned} \tag{12}$$

We use Algorithm 1 to find ξ^*. A multi-level resolution implementation is used to accelerate the algorithm. Since the ICP algorithm is more sensitive to noise (Rusinkiewicz and Levoy, 2001), and our method is more robust to sensor noise because of the involvement of submaps and the Euclidean distance grid, our scan matcher is more robust and stable.

Algorithm 1: Scan matching

```
best_score ← +∞
for (j_x, j_y, j_z, j_rx, j_ry, j_rz) ∈ W̄ do
    score ← Σ_{k=1}^{K} M_nearest(T_{ξ0+Δξ} h_k)
    Δξ = (rj_x, rj_y, rj_z, δθ j_rx, δθ j_ry, δθ j_rz)
    if score < best_score then
        match ← ξ0 + Δξ
        best_score ← score
return best_score and match
```

4.4 Loop closure

Since scans are matched only against a submap and the VIO matches only a few recent keyframes, the approach above will accumulate errors. The accumulated error is small for only a few consecutive scans and visual frames.

Recognizing the past places and adding loop pose constraints to the pose graph can effectively reduce the cumulative error and improve the positioning accuracy. LiDaR SLAM systems are often unable to detect the loop effectively in certain scenarios, since LiDaR scans can describe only the environment structure, which usually lacks unique features of the scene. Indeed, there may be multiple scenes that have very similar LiDaR scans to be recognized, such as long corridors and office areas with similar structures. However, the rich visual textures can make up for this defect.

We use the BoW to construct the dictionary corresponding to the keyframes using the visual feature points. To insert a new keyframe, all the following conditions must be met:

1. more than 20 frames must have passed from the last global relocalization;

2. local mapping is idle, or more than 20 frames have passed from the last keyframe insertion;

3. the current frame tracks at least 50 points;

4. the current frame tracks fewer than 90% key points of the last keyframe.

For each keyframe, the ORB features and the visual words in the BoW dictionary are extracted and saved as a bag of features. As a result, a keyframe stores the robot pose and bag of features, as well as the LiDaR scan points obtained in that pose. The keyframes are then used for loop detection and relocalization, and the robot poses of the keyframes are updated after global optimization.

Then, we use Algorithm 2 to detect and correct the loop. If the matching error is small enough, the loop will be validated and the matching result will be added as a constraint to the graph optimization, so that the accumulative errors can be eliminated.

Algorithm 2: Loop closing

```
if visual loop is detected (BoW) then
    if LiDaR loop is detected (Algorithm 1) then
        loop correction (graph optimization) by the LiDaR loop detection result
    else
        loop correction (graph optimization) by the visual loop detection result
else
    if LiDaR loop is detected (Algorithm 1) then
        loop correction (graph optimization) by the LiDaR loop detection result
    else
        loop correction fails
return success or failure
```

5 Experiments and discussion

5.1 Methods and procedure

To evaluate the long-term large-scale operation, localization accuracy, and efficiency of the pose-graph optimization, we have performed an extensive experimental validation of our VILN system using outdoor sequences from the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) dataset (Geiger et al., 2013). In our custom complex environments, such as low-texture and duplicated structural scenes, we evaluate the general performance of the system. We also validate our new system when the robot steers fast or the visual sensors are over-exposed.

Our system runs in real time and processes the images and LiDaR scans at exactly the same frame rate in which they were acquired. We have carried out all experiments with an Intel Core i7-5500U (two cores @2.40 GHz) and 8 GB RAM. The robot platform is shown in Fig. 5.

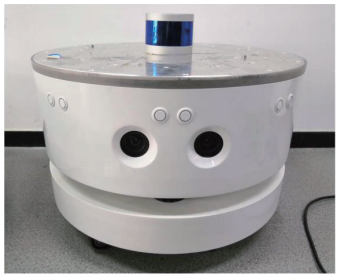


Fig. 5 Experimental platform. All experiments are carried out with this platform, which contains a stereo camera, an IMU, a 3D laser scanner, an Intel Core i7-5500U (two cores @2.40 GHz), and 8 GB RAM

The odometry benchmark from the KITTI dataset contains 11 sequences from a car driven around a residential area with accurate ground truth from GPS and a Velodyne laser scanner. This is a very challenging dataset for the SLAM system due to the high steering velocity and the relatively high car speed, the sequences being recorded at 10 frames/s (Mur-Artal et al., 2015). The dataset contains a highway (such as sequence 01), residential (such as sequences 00, 05, and 08), and some other crucial environments. We play the sequences at the real frame rate at which they were recorded, and VILN is able to process all the sequences. Sequences that contain loops are correctly detected and closed by our system. Next, we compare our results with those from ORB-SLAM2 (Mur-Artal and Tardós, 2017).

Our custom-collected datasets contain several sequences from a robot traversing a campus and the adjacent pedestrian streets. The sequences are recorded by a stereo camera at 10 frames/s and a resolution of 1280×600, an IMU, and a 3D laser scanner (10 Hz, 16 beams, 0.09° angular resolution, and 2 cm distance accuracy), collecting 320 000 points/s, with a horizontal field of view 360° and vertical 30°,

and a range of 70 m. The sensors rotate very fast at some places, which will happen often when the robot steers fast; therefore, it is necessary to evaluate the performance of VILN in this situation. Different experiments are undertaken to evaluate the performance in terms of mapping, accuracy, and long-term robustness. In the experiments, the parameter r in Eq. (11) is set to 0.05 m by considering the accuracy and real-time requirement of the system in different environments.

5.2 Large-scale scenarios in the KITTI dataset

We present the relative median translation root mean square error (RMSE) of our keyframe trajectories in Table 1. The relative median translation RMSE is defined as

$$\text{error} = \frac{1}{T_{\text{total}}} \sum_{k=1}^{K} \|\boldsymbol{p}_k - \hat{\boldsymbol{p}}_k\|^2 / K, \qquad (13)$$

where $\boldsymbol{p}_k = (x_k, y_k, z_k)^{\mathrm{T}}$ is the position of each frame estimated using our framework, $\hat{\boldsymbol{p}}_k = (\hat{x}_k, \hat{y}_k, \hat{z}_k)^{\mathrm{T}}$ is the ground truth of each frame, and T_{total} is the total mileage of the dataset.

The results demonstrate that VILN is more accurate and long-term robust, the trajectory relative error being typically below 0.33% of its mileage, sometimes even less as in sequence 00 with an error of 0.064% or higher as in sequence 09 with an error of 1.67%. In sequence 00, there are several loops, and VILN detects these correctly. Fig. 6 demonstrates that it is necessary to use loop correction to achieve long-term robustness and more accurate results. As shown in Table 1, our system outperforms ORB-SLAM2 in most sequences. In sequence 09, for the frontend odometry of our system, we use tightly coupled visual-IMU odometry first and then loosely coupled LiDaR odometry to estimate the poses, which relies heavily on the accuracy of LiDaR correction. Moreover, there may be some duplicated structural

Table 1 Comparison of accuracy in the KITTI dataset

Sequence	Dimension (m×m)	Relative median translation RMSE (%)		Mileage (m)
		VILN	ORB-SLAM2	
KITTI 00	496×564	**0.064**	0.70	3744.90
KITTI 01	1157×1827	**0.19**	1.39	2461.75
KITTI 02	946×599	**0.091**	0.76	5251.44
KITTI 03	199×471	**0.23**	0.71	567.40
KITTI 04	394×0.5	**0.12**	0.48	393.43
KITTI 05	426×479	**0.11**	0.40	2224.17
KITTI 06	457×23	**0.17**	0.51	1236.30
KITTI 07	209×191	**0.088**	0.50	701.12
KITTI 08	391×808	**0.33**	1.05	3236.36
KITTI 09	568×465	1.67	**0.87**	1698.60
KITTI 10	177×671	**0.14**	0.60	924.93

Bold values represent better results

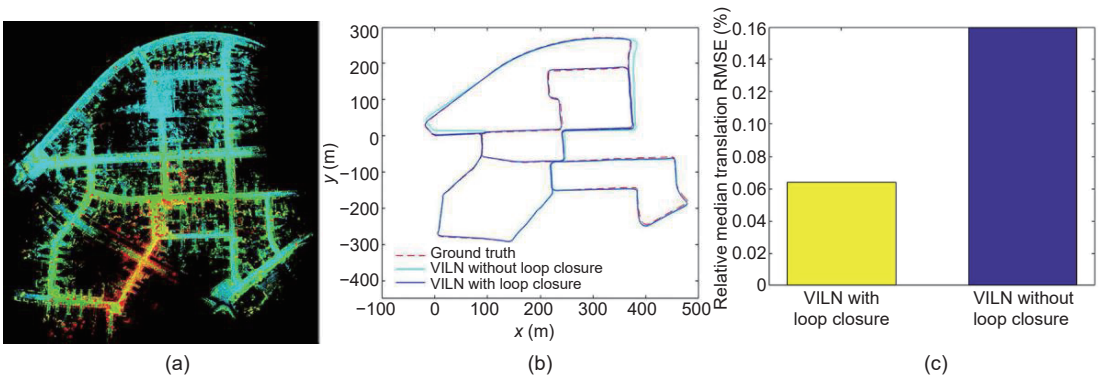


Fig. 6 Sequence 00 from the odometry benchmark of the KITTI dataset: (a) fusion maps; (b) ground truth and trajectories with and without loop closure; (c) relative RMSE comparison

LiDaR keyframes when visual correction is not so accurate. All the above situations may be the reason that the result of our system is worse. In future work, we will use tightly coupled visual-IMU-LiDaR odometry to improve the performance of VILN.

Some of our trajectories and the ground truth are shown in Fig. 7. We have aligned the keyframe trajectories of the VILN system and the ground truth with a similarity transformation.

5.3 System performance in custom-collected datasets

For visual sensors, as shown in Fig. 8, the rapid steering of the robot causes the image to be blurred, and some scenes cause over-exposure in our custom-collected datasets. For the above reasons, the pure visual SLAM system ORB-SLAM2 does not finish the sequences. Therefore, we evaluate only the performance of our VILN system using the custom-collected datasets.

Fig. 9 shows a long corridor environment, which may cause a pure LiDaR system to fail. In Fig. 10, we show the mapping results of VILN with and without loop correction. By observing the middle part of Fig. 10, it is clear that the right image (with loop correction) is more accurate than the left one (without loop correction).

Fig. 11 shows several examples of the custom sequences using the VILN system. While Fig. 11a includes low-texture environments that may be unfriendly for pure visual methods, Fig. 11b contains complex environments, such as duplicated structures or light changes, which may cause over-exposure of the visual sensors. Such environments could render the pure visual or LiDaR methods unsuccessful. However, our system could run stably for a long term in such environments.

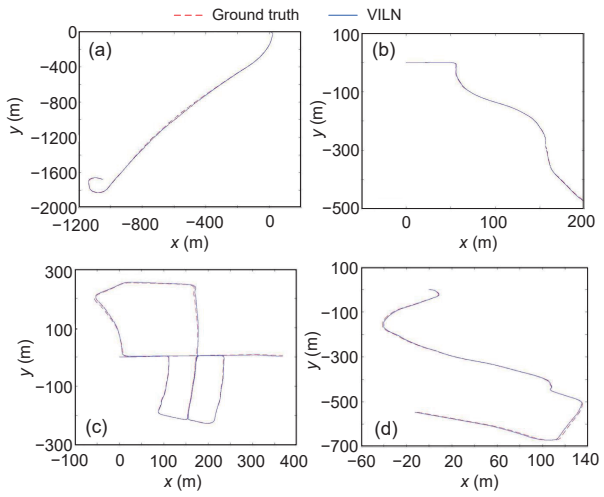


Fig. 7 Estimated trajectory (blue) and ground truth (red) in KITTI sequences 01 (a), 03 (b), 05 (c), and 10 (d). References to color refer to the online version of this figure

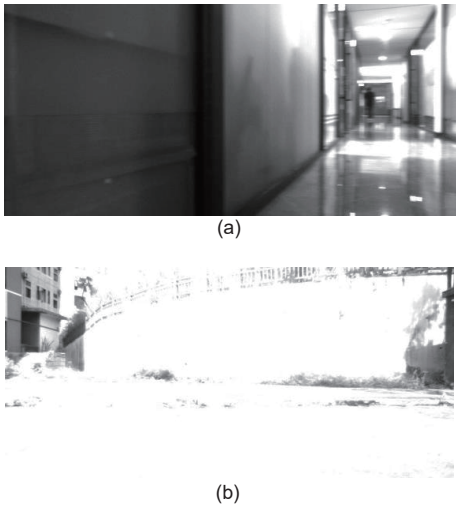


Fig. 8 Some crucial situations for pure visual SLAM systems: (a) the robot steers very fast; (b) the cameras are over-exposed by changes in light conditions

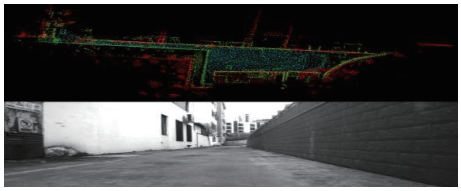


Fig. 9 Experimental mapping result of the proposed method in long corridor environments

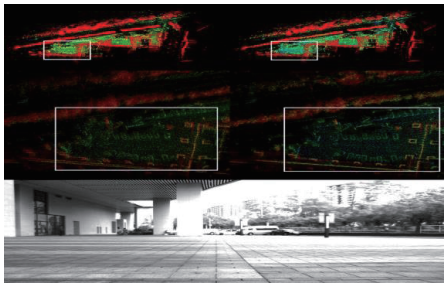


Fig. 10 Experimental mapping results: comparison of the proposed method with and without loop correction. Top: left is the result without loop correction, and the right panel is with loop correction; middle: details of the top results (clearly, the result on the right is more accurate than that on the left); bottom: the experimental environment

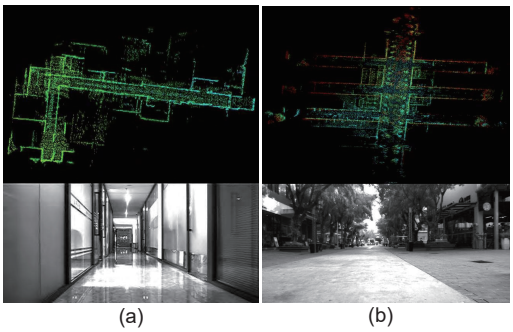


Fig. 11 Experimental mapping results of the proposed method. (a) includes low-texture environments that may be unfriendly for pure visual methods, and (b) contains complex environments, such as changes in light conditions and duplicated structures, which will cause the pure visual or LiDaA methods to fail

6 Conclusions and future work

In this paper, a new SLAM framework based on the fusion of vision-IMU-3D LiDaR, namely, VILN SLAM, has been presented, and an extensive evaluation using custom-collected datasets and public datasets has been provided and discussed. This newly proposed VILN SLAM system uses a robust visual-IMU-LiDaR odometry to handle critical environments, a LiDaR enhanced visual loop closure system to eliminate the drift caused by long-term running, and a LiDaR and vision fused map to achieve relocalization after long-term running with low drift. We have demonstrated that our VILN SLAM system is consistently valid in different indoor and outdoor scenarios, including more complex environments such as low-texture and duplicated structural environments. The error of the system is typically $\leq 0.33\%$ regardless of the scale (small or large) of the scenarios. In the future, we will focus on developing a new tightly coupled visual-IMU-LiDaR fusion navigation system to further improve the accuracy and long-term robustness under more complex and dynamic environments.

Contributors

Wei WEI, Xiaorui ZHU, and Yi WANG designed the research. Wei WEI processed the data. Wei WEI and Xiaorui ZHU drafted the paper. Yi WANG helped organize the paper. Wei WEI and Xiaorui ZHU revised and finalized the paper.

Compliance with ethics guidelines

Wei WEI, Xiaorui ZHU, and Yi WANG declare that they have no conflict of interest.

References

Banerjee N, Connolly RC, Lisin D, et al., 2019. View management for lifelong visual maps. IEEE/RSJ Int Conf on Intelligent Robots and Systems, p.7871-7878. https://doi.org/10.1109/IROS40897.2019.8968245

Davison AJ, Reid ID, Molton ND, et al., 2007. MonoSLAM: real-time single camera SLAM. *IEEE Trans Patt Anal Mach Intell*, 29(6):1052-1067. https://doi.org/10.1109/TPAMI.2007.1049

Deschaud JE, 2018. IMLS-SLAM: scan-to-model matching based on 3D data. IEEE Int Conf on Robotics and Automation, p.2480-2485. https://doi.org/10.1109/ICRA.2018.8460653

Engel J, Schöps T, Cremers D, 2014. LSD-SLAM: large-scale direct monocular SLAM. Proc 13th European Conf on Computer Vision, p.834-849. https://doi.org/10.1007/978-3-319-10605-2_54

Engel J, Koltun V, Cremers D, 2018. Direct sparse odometry. *IEEE Trans Patt Anal Mach Intell*, 40(3):611-625. https://doi.org/10.1109/TPAMI.2017.2658577

Forster C, Pizzoli M, Scaramuzza D, 2014. SVO: fast semi-direct monocular visual odometry. IEEE Int Conf on Robotics and Automation, p.15-22. https://doi.org/10.1109/ICRA.2014.6906584

Forster C, Carlone L, Dellaert F, et al., 2017. On-manifold preintegration for real-time visual-inertial odometry. *IEEE Trans Robot*, 33(1):1-21. https://doi.org/10.1109/TRO.2016.2597321

Geiger A, Lenz P, Stiller C, et al., 2013. Vision meets robotics: the KITTI dataset. *Int J Robot Res*, 32(11): 1231-1237. https://doi.org/10.1177/0278364913491297

Grisetti G, Stachniss C, Burgard W, 2007. Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Trans Robot*, 23(1):34-46. https://doi.org/10.1109/TRO.2006.889486

Hemann G, Singh S, Kaess M, 2016. Long-range GPS-denied aerial inertial navigation with lidar localization. IEEE/RSJ Int Conf on Intelligent Robots and Systems, p.1659-1666. https://doi.org/10.1109/IROS.2016.7759267

Hess W, Kohler D, Rapp H, et al., 2016. Real-time loop closure in 2D LIDAR SLAM. IEEE Int Conf on Robotics and Automation, p.1271-1278. https://doi.org/10.1109/ICRA.2016.7487258

Kerl C, Sturm J, Cremers D, 2013. Dense visual SLAM for RGB-D cameras. IEEE/RSJ Int Conf on Intelligent Robots and Systems, p.2100-2106. https://doi.org/10.1109/IROS.2013.6696650

Kim G, Park B, Kim A, 2019. 1-day learning, 1-year localization: long-term LiDAR localization using scan context image. *IEEE Robot Autom Lett*, 4(2):1948-1955. https://doi.org/10.1109/LRA.2019.2897340

Klein G, Murray D, 2007. Parallel tracking and mapping for small AR workspaces. Proc 6th IEEE and ACM Int Symp on Mixed and Augmented Reality, p.1-10. https://doi.org/10.1109/ISMAR.2007.4538852

Konolige K, Grisetti G, Kümmerle R, et al., 2010. Efficient sparse pose adjustment for 2D mapping. IEEE/RSJ Int Conf on Intelligent Robots and Systems, p.22-29. https://doi.org/10.1109/IROS.2010.5649043

Lee J, Hwang S, Lee K, et al., 2020. AD-VO: scale-resilient visual odometry using attentive disparity map. https://arxiv.org/abs/2001.02090

Maurer CR, Qi RS, Raghavan V, 2003. A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Trans Patt Anal Mach Intell*, 25(2):265-270. https://doi.org/10.1109/TPAMI.2003.1177156

Mur-Artal R, Tardós JD, 2017. ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Trans Robot*, 33(5):1255-1262. https://doi.org/10.1109/TRO.2017.2705103

Mur-Artal R, Montiel JMM, Tardós JD, 2015. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Trans Robot*, 31(5):1147-1163. https://doi.org/10.1109/TRO.2015.2463671

Nair GB, Daga S, Sajnani R, et al., 2020. Multi-object monocular SLAM for dynamic environments. https://arxiv.org/abs/2002.03528

Newcombe RA, Lovegrove SJ, Davison AJ, 2011. DTAM: dense tracking and mapping in real-time. Int Conf on Computer Vision, p.2320-2327. https://doi.org/10.1109/ICCV.2011.6126513

Patel N, Khorrami F, Krishnamurthy P, et al., 2019. Tightly coupled semantic RGB-D inertial odometry for accurate long-term localization and mapping. Proc 19th Int Conf on Advanced Robotics, p.523-528. https://doi.org/10.1109/ICAR46387.2019.8981658

Rusinkiewicz S, Levoy M, 2001. Efficient variants of the ICP algorithm. Proc 3rd Int Conf on 3-D Digital Imaging and Modeling, p.145-152. https://doi.org/10.1109/IM.2001.924423

Shao WZ, Vijayarangan S, Li C, et al., 2019. Stereo visual inertial LiDAR simultaneous localization and mapping. IEEE/RSJ Int Conf on Intelligent Robots and Systems, p.370-377. https://doi.org/10.1109/IROS40897.2019.8968012

Sibley G, Matthies L, Sukhatme G, 2010. Sliding window filter with application to planetary landing. *J Field Robot*, 27(5):587-608. https://doi.org/10.1002/rob.20360

Sünderhauf N, Neubert P, Protzel P, 2013. Predicting the change—a step towards life-long operation in everyday environments. Proc Challenges and Vision Workshop at RSS, p.1-4.

Wagstaff B, Peretroukhin V, Kelly J, 2020. Self-supervised deep pose corrections for robust visual odometry. IEEE Int Conf on Robotics and Automation, p.2331-2337. https://doi.org/10.1109/ICRA40945.2020.9197562

Wang ZJ, Wu Y, Niu QQ, 2020. Multi-sensor fusion in automated driving: a survey. *IEEE Access*, 8:2847-2868. https://doi.org/10.1109/ACCESS.2019.2962554

Xu YL, Ou YS, Xu TT, 2018. SLAM of robot based on the fusion of vision and LIDAR. IEEE Int Conf on Cyborg and Bionic Systems, p.121-126. https://doi.org/10.1109/CBS.2018.8612212

Zhang J, Singh S, 2015. Visual-lidar odometry and mapping: low-drift, robust, and fast. IEEE Int Conf on Robotics and Automation, p.2174-2181. https://doi.org/10.1109/ICRA.2015.7139486

Zhao HJ, Chiba M, Shibasaki R, et al., 2008. SLAM in a dynamic large outdoor environment using a laser scanner. IEEE Int Conf on Robotics and Automation, p.1455-1462. https://doi.org/10.1109/ROBOT.2008.4543407

Zhao ZR, Mao YJ, Ding Y, et al., 2019. Visual-based semantic SLAM with landmarks for large-scale outdoor environment. Proc 2nd China Symp on Cognitive Computing and Hybrid Intelligence, p.149-154. https://doi.org/10.1109/CCHI.2019.8901910

Zhu XR, Qiu CX, Deng FC, et al., 2017. Cloud-based real-time outsourcing localization for a ground mobile robot in large-scale outdoor environments. *J Field Robot*, 34(7):1313-1331. https://doi.org/10.1002/rob.21712