TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 09/24 pp415–429
DOI: 10.26599/TST.2023.9010010
Volume 29, Number 2, April 2024

Camera, LiDAR, and IMU Based Multi-Sensor Fusion SLAM: A Survey

Jun Zhu, Hongyi Li, and Tao Zhang*

Abstract: In recent years, Simultaneous Localization And Mapping (SLAM) technology has prevailed in a wide range of applications, such as autonomous driving, intelligent robots, Augmented Reality (AR), and Virtual Reality (VR). Multi-sensor fusion using the most popular three types of sensors (e.g., visual sensor, LiDAR sensor, and IMU) is becoming ubiquitous in SLAM, in part because of the complementary sensing capabilities and the inevitable shortages (e.g., low precision and long-term drift) of the stand-alone sensor in challenging environments. In this article, we survey thoroughly the research efforts taken in this field and strive to provide a concise but complete review of the related work. Firstly, a brief introduction of the state estimator formation in SLAM is presented. Secondly, the state-of-the-art algorithms of different multi-sensor fusion algorithms are given. Then we analyze the deficiencies associated with the reviewed approaches and formulate some future research considerations. This paper can be considered as a brief guide to newcomers and a comprehensive reference for experienced researchers and engineers to explore new interesting orientations.

Key words: multi-sensor fusion; Simultaneous Localization And Mapping (SLAM); navigation; localization

1 Introduction

Simultaneous Localization And Mapping (SLAM)[1] is a technology that estimates the state (e.g., position, orientation, velocity, sensor bias, and calibration parameters) of a robot, and at the same time constructs a model of the environment where the robot is moving using data perceived by sensors on the robot. Over the past 36 years, significant progress has been made by the SLAM community, enabling wide applications in related industries. Early SLAM research introduced the main probabilistic formulations of SLAM[2], then fundamental properties (observability, convergence, and consistency) of SLAM were analyzed[3], and nowadays the essential requirements to consider are robust performance, high-level understanding of the environment, resource awareness and task-driven perception[4]. Nevertheless, a single sensor is hardly capable of these demands. Although Global Navigation Satellite System (GNSS) can provide absolute position, it is not always available or accurate in the environments like tunnels, caves, city canyons, etc. Low-cost and light-weight IMU has been widely used, but its measurements are corrupted by noise and bias, such that it cannot provide reliable pose estimates for long-term navigation. The monocular camera suffers from scale drift, and LiDAR fails in structure-less environments. Therefore, with multi-sensor fusion, the deficiencies of stand-alone sensors can be compensated, and more reliable estimates will be provided.

Recently, several surveys about multi-sensor fusion SALM have been proposed. Some reviews[4–7] focus on multi-sensor fusion in autonomous driving, while most reviews[8–13] pay attention to visual-inertial SLAM. There are few surveys about LiDAR-inertial or visual-LiDAR SLAM[14].

• Tao Zhang, Jun Zhu, and Hongyi Li are with Department of Automation, Tsinghua University, Beijing 100084, China. E-mail: taozhang@tsinghua.edu.cn; j-zhu20@mails.tsinghua.edu.cn; lihy20@mails.tsinghua.edu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2022-09-17; revised: 2022-12-18; accepted: 2023-02-21

Reference [5] is mainly focusing on the multi-target tracking in automated driving but not SLAM in particular. On the other hand, Ref. [6] is like Ref. [14], focusing on the visual-LiDAR fusion in SLAM context.

This paper mainly focuses on three types of sensors (visual sensor, LiDAR, and IMU), which are the most popular sensors in multi-sensor fusion algorithms. To make this paper accessible to new researchers on multi-sensor fusion SLAM, we first present a brief introduction of the state estimator formation in Section 2. Then, Section 3 divides the sensor fusion methods into four categories, i.e., visual-inertial, LiDAR-inertial, visual-LiDAR, and LiDAR-visual-inertial fusion algorithms, and presents a comprehensive and systematic review for each category separately over the last ten years, and especially attaches attention to deficiency compensation. We discuss the challenges and future research directions in Section 4. Finally, we draw our conclusions in Section 5.

2 Brief Introduction of the State Estimator Formation

Kalman Filter (KF) and sliding window optimization are the most commonly used state estimator formations in multi-sensor fusion. In this section, we will give a brief introduction to them.

2.1 KF

In SLAM, prior values are usually recursively derived from sensors, such as IMU and encoder. Measurement values are usually obtained from sensors, such as GPS, camera, and LiDAR. The posterior value is the fusion result, which also is positioning output. In actual robot state estimation, the posterior probability density with estimation can be expressed as

$$p(\boldsymbol{x}_k|\check{\boldsymbol{x}}_0, \boldsymbol{v}_{1:k}, \boldsymbol{y}_{0:k}) \tag{1}$$

where k is the index of IMU measurement, $\boldsymbol{x}_k$ is robot position at k-th state vector, $\check{\boldsymbol{x}}_0$ is the initial state vector, $\boldsymbol{v}_{1:k}$ means input vector from 1st to k-th, and $\boldsymbol{y}_{0:k}$ means observational vector from the initial state to k-th.

The kinematic equation and observational equation are as follows:

$$\boldsymbol{x}_k = \boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{v}_k) + \boldsymbol{\omega}_k \tag{2}$$

$$\boldsymbol{y}_k = \boldsymbol{g}(\boldsymbol{x}_k) + \boldsymbol{n}_k \tag{3}$$

where $\boldsymbol{\omega}_k$ is the process noise vector that is assumed to be zero-mean Gaussian noise with the covariance $\boldsymbol{R}_k$, $\boldsymbol{n}_k$ is the measurement noise vector that is assumed to be zero-mean Gaussian noise with the covariance $\boldsymbol{Q}_k$, function $\boldsymbol{f}(\,)$ can be used to compute the predicted state from the previous estimate, and function $\boldsymbol{h}(\,)$ can be used to compute the predicted measurement from the predicted state.

Using KF to solve robot state estimation is a common method. It is one of the best Bayesian filters research technologies, but it can only solve the linear Gaussian system. The overview of KF is given in Algorithm 1, where F_{k-1} is the state transition model, B_k is the control-input model, G_k is the observation model, $\check{x}_k$ is the predicted state estimate, $\hat{x}_{k-1}$ and $\hat{x}_k$ are the updated state estimates, $\check{P}_k$ is the predicted covariance estimate, $\hat{P}_{k-1}$ and $\hat{P}_k$ are the updated covariance estimates, and K_k is the Kalman gain.

2.1.1 Extended Kalman Filters (EKF)

EKF is obtained by extending KF to nonlinear problems. The overview of EKF is given in Algorithm 2, where F_{k-1} is the Jacobian matrix of $f(x_{k-1}, v_k)$, G_k is the Jacobian matrix of $g(\check{x}_k)$.

Algorithm 1 KF

1: Kinematic equation:

$$\boldsymbol{x}_k = \boldsymbol{F}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{B}_k\boldsymbol{v}_k + \boldsymbol{\omega}_k, \boldsymbol{\omega}_k \sim \mathcal{N}(0, \boldsymbol{R}_k).$$

2: Observational equation:

$$\boldsymbol{y}_k = \boldsymbol{G}_k\boldsymbol{x}_k + \boldsymbol{n}_k, \boldsymbol{n}_k \sim \mathcal{N}(0, \boldsymbol{Q}_k).$$

3: State propagation:

$$\check{\boldsymbol{x}}_k = \boldsymbol{F}_{k-1}\hat{\boldsymbol{x}}_{k-1} + \boldsymbol{B}_k\boldsymbol{v}_k,$$
$$\check{\boldsymbol{P}}_k = \boldsymbol{F}_{k-1}\hat{\boldsymbol{P}}_{k-1}\boldsymbol{F}_{k-1}^{\mathrm{T}} + \boldsymbol{R}_k.$$

4: Kalman gain:

$$\boldsymbol{K}_k = \check{\boldsymbol{P}}_k\boldsymbol{G}_k^{\mathrm{T}}\left(\boldsymbol{G}_k\check{\boldsymbol{P}}_k\boldsymbol{G}_k^{\mathrm{T}} + \boldsymbol{Q}_k\right)^{-1}.$$

5: Update:

$$\hat{\boldsymbol{x}}_k = \check{\boldsymbol{x}}_k + \boldsymbol{K}_k\left(\boldsymbol{y}_k - \boldsymbol{G}_k\check{\boldsymbol{x}}_k\right),$$
$$\hat{\boldsymbol{P}}_k = \left(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{G}_k\right)\check{\boldsymbol{P}}_k.$$

Algorithm 2 EKF

1: Kinematic equation:

$$\boldsymbol{x}_k \approx \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{v}_k) + \boldsymbol{F}_{k-1}(\boldsymbol{x}_{k-1} - \hat{\boldsymbol{x}}_{k-1}) + w_k.$$

2: Observational equation:

$$\boldsymbol{y}_k \approx \boldsymbol{g}(\check{\boldsymbol{x}}_k) + \boldsymbol{G}_k(\boldsymbol{x}_k - \check{\boldsymbol{x}}_k) + \boldsymbol{n}_k.$$

3: State propagation:

$$\check{\boldsymbol{x}}_k = \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{v}_k),$$
$$\check{\boldsymbol{P}}_k = \boldsymbol{F}_{k-1}\hat{\boldsymbol{P}}_{k-1}\boldsymbol{F}_{k-1}^{\mathrm{T}} + \boldsymbol{R}_k.$$

4: Kalman gain:

$$\boldsymbol{K}_k = \check{\boldsymbol{P}}_k\boldsymbol{G}_k^{\mathrm{T}}\left(\boldsymbol{G}_k\check{\boldsymbol{P}}_k\boldsymbol{G}_k^{\mathrm{T}} + \boldsymbol{Q}_k\right)^{-1}.$$

5: Update:

$$\hat{\boldsymbol{x}}_k = \check{\boldsymbol{x}}_k + \boldsymbol{K}_k\left(\boldsymbol{y}_k - \boldsymbol{g}(\check{\boldsymbol{x}}_k)\right),$$
$$\hat{\boldsymbol{P}}_k = \left(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{G}_k\right)\check{\boldsymbol{P}}_k.$$

2.1.2 Iterated Extended Kalman Filters (IEKF)

The closer between the linearization operating point and the truth value, the smaller error will be brought. So to gradually find the exact linearization point through iteration, thus improving the accuracy. The overview of IEKF is given in Algorithm 3, where $\check{x}_{option,k}$ is the linearization operating point.

Unlike EKF, IEKF requires repeatedly calculating Kalman gain $\boldsymbol{K}_k$ and posterior mean $\hat{\boldsymbol{x}}_k$ until the results change a little, and at last, updates posterior covariance $\hat{\boldsymbol{P}}_k$ once.

2.1.3 Error-State Kalman Filter (ESKF)

In error-state filter formulations, denoting as follows:

$$\boldsymbol{x}_t = \boldsymbol{x} + \delta\boldsymbol{x} \tag{4}$$

where $\boldsymbol{x}_t$ is true state values, $\boldsymbol{x}$ is nominal state values, and $\delta\boldsymbol{x}$ is error state values. High-frequency IMU data are integrated into a nominal state $\boldsymbol{x}$. But It does not consider noise terms and other possible model imperfections, leading to accumulated errors. These errors are collected in the error-state $\delta\boldsymbol{x}$ and estimated with the ESKF, this time incorporating all the noise and perturbations. The overview of ESKF is given in Algorithm 4, where $\delta\check{x}_k$ is the predicted error state estimate, $\delta\hat{x}_{k-1}$ and $\delta\hat{x}_k$ are the updated error state estimates.

2.2 Sliding window optimization

Sliding window optimization, which optimizes all states in a sliding window, has been widely used in multi-sensor fusion algorithms because of its advantage of bounded computation costs and relatively sufficient accuracy. For a sliding window of n states, the optimal states $\mathcal{X} = \left[\boldsymbol{X}_1^{\mathrm{T}}, \boldsymbol{X}_2^{\mathrm{T}}, \ldots, \boldsymbol{X}_n^{\mathrm{T}}\right]^{\mathrm{T}}$ are obtained by minimizing the residuals,

$$\min_{\mathcal{X}} \left\{ \left\|\boldsymbol{r}_p\left(\mathcal{X}\right)\right\|^2 + \sum_{k\in\mathcal{I}} \left\|\boldsymbol{r}_{\mathcal{I}}\left(k,\mathcal{X}\right)\right\|^2_{\boldsymbol{P}^k_{\mathcal{I}}} + \sum_{k\in\mathcal{A}} \left\|\boldsymbol{r}_{\mathcal{A}}\left(k,\mathcal{X}\right)\right\|^2_{\boldsymbol{P}^k_{\mathcal{A}}} \right\} \tag{5}$$

where $\boldsymbol{r}_{\mathcal{I}}(k,\mathcal{X})$ is the IMU residual term which incorporates the relative motion constraints among frames and is usually computed by preintegration to avoid repropagating IMU states. $\boldsymbol{r}_{\mathcal{A}}(k,\mathcal{X})$ is visual or LiDAR residual term incorporating geometric constraints from visual or LiDAR measurements. $\boldsymbol{P}^k_{\mathcal{I}}$ and $\boldsymbol{P}^k_{\mathcal{A}}$ are corresponding covariance matrices. $\mathcal{I}$ is the set of all IMU measurements and $\mathcal{A}$ is the set of all visual or LiDAR features in current window. $\boldsymbol{r}_p(\mathcal{X})$ denotes the prior residual term from marginalization due to window-sliding. Thanks to marginalization, the sliding window optimization limits the computational complexity without substantial information loss[15].

Algorithm 3 IEKF

1: Kinematic equation:

$$\boldsymbol{x}_k \approx \boldsymbol{f}\left(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{v}_k\right) + \boldsymbol{F}_{k-1}(\boldsymbol{x}_{k-1} - \hat{\boldsymbol{x}}_{k-1}) + w_k.$$

2: Observational equation:

$$\boldsymbol{y}_k \approx \boldsymbol{g}\left(\check{\boldsymbol{x}}_{option,k}\right) + \boldsymbol{G}_k(\boldsymbol{x}_k - \check{\boldsymbol{x}}_{option,k}) + \boldsymbol{n}_k.$$

3: State propagation:

$$\check{\boldsymbol{x}}_k = \boldsymbol{f}\left(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{v}_k\right),$$
$$\check{\boldsymbol{P}}_k = \boldsymbol{F}_{k-1}\hat{\boldsymbol{P}}_{k-1}\boldsymbol{F}_{k-1}^{\mathrm{T}} + \boldsymbol{R}_k.$$

4: Kalman gain:

$$\boldsymbol{K}_k = \check{\boldsymbol{P}}_k\boldsymbol{G}_k^{\mathrm{T}}\left(\boldsymbol{G}_k\check{\boldsymbol{P}}_k\boldsymbol{G}_k^{\mathrm{T}} + \boldsymbol{Q}_k\right)^{-1}.$$

5: Update:

$$\hat{\boldsymbol{x}}_k = \check{\boldsymbol{x}}_k + \boldsymbol{K}_k\left[\boldsymbol{y}_k - \boldsymbol{g}\left(\check{\boldsymbol{x}}_{option,k}, \boldsymbol{n}_k\right)\right] - \boldsymbol{K}_k\left[\boldsymbol{G}_k\left(\check{\boldsymbol{x}}_k - \check{\boldsymbol{x}}_{option,k}\right)\right],$$
$$\hat{\boldsymbol{P}}_k = \left(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{G}_k\right)\check{\boldsymbol{P}}_k.$$

Algorithm 4 ESKF

1: Kinematic equation:

$$\delta\boldsymbol{x}_k = \boldsymbol{f}\left(\boldsymbol{x}_{k-1}, \delta\boldsymbol{x}_{k-1}, \boldsymbol{v}_k, w_k\right) \approx \boldsymbol{F}_{k-1}\delta\boldsymbol{x}_{k-1} + w_k.$$

2: Observational equation:

$$\boldsymbol{y}_k = \boldsymbol{g}\left(\boldsymbol{x}_{t,k}\right) + \boldsymbol{n}_k.$$

3: State propagation:

$$\delta\check{\boldsymbol{x}}_k = \boldsymbol{F}_{k-1}\delta\hat{\boldsymbol{x}}_{k-1},$$
$$\check{\boldsymbol{P}}_k = \boldsymbol{F}_{k-1}\hat{\boldsymbol{P}}_{k-1}\boldsymbol{F}_{k-1}^{\mathrm{T}} + \boldsymbol{R}_k.$$

4: Kalman gain:

$$\boldsymbol{K}_k = \check{\boldsymbol{P}}_k\boldsymbol{G}_k^{\mathrm{T}}\left(\boldsymbol{G}_k\check{\boldsymbol{P}}_k\boldsymbol{G}_k^{\mathrm{T}} + \boldsymbol{Q}_k\right)^{-1}.$$

5: Update:

$$\hat{\boldsymbol{P}}_k = \left(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{G}_k\right)\check{\boldsymbol{P}}_k,$$
$$\delta\hat{\boldsymbol{x}}_k = \boldsymbol{K}_k\left[\boldsymbol{y}_k - \boldsymbol{g}\left(\boldsymbol{x}_{t,k}\right)\right].$$

6: Note:

$$\boldsymbol{G}_k = \frac{\partial\boldsymbol{g}}{\partial(\delta\boldsymbol{x}_k)} = \frac{\partial\boldsymbol{g}}{\partial\boldsymbol{x}_{t,k}}\frac{\partial\boldsymbol{x}_{t,k}}{\partial(\delta\boldsymbol{x})}.$$

3 Multi-Sensor Fusion Algorithms

In this paper, we mainly consider three kinds of sensors: monocular camera, LiDAR, and IMU. Firstly, we divide the multi-sensor fusion algorithms into four categories, i.e., visual-inertial, LiDAR-inertial, visual-LiDAR, and LiDAR-visual-inertial fusion algorithms. Then, we give detailed descriptions of the State-Of-The-Art (SOTA) methods for each category. Representative methods for each category are shown in Table 1.

Table 1 SOTA methods of multi-sensor fusion.

Fusion type	Year	Method	Type	Loop closure	Sensor type	Fusion strategy
Visual-inertial	2013	MSCKF 2.0[16]	FB	–	MC+IMU	MSCKF
	2015	ROVIO[17]	FB	–	MC+IMU	EKF-SLAM
	2018	VINS-Mono[18]	OB	FAST + DBoW	MC+IMU	SWO+PGO
LiDAR-inertial	2021	LION[19]	LC	–	ML+IMU	SWO
	2019	LIOM[20]	TC	–	ML+IMU	SWO
	2020	LINS[21]	TC	–	ML+IMU	Iterated ESKF
	2020	LIO-SAM[22]	TC	Euclidean distance	ML+IMU	FGO
	2021	LILI-OM[15]	TC	Euclidean distance	ML or SL+IMU	SWO + PGO+ FGO
	2021	FAST-LIO[23]	TC	–	SL+IMU	Iterated ESKF
	2022	Faster-LIO[24]	TC	–	SL+IMU	Iterated ESKF
Visual-LiDAR	2017	DEMO[25]	LC	–	ML+RC	BA
	2020	Method[26]	LC	ORB + DBoW	ML+RC	SWO + PGO
	2018	LIMO[27]	LC	–	ML+MC	BA
	2020	Method[28]	TC	ORB + DBoW	ML+MC	BA
	2021	TVL-SLAM[29]	TC	ORB + DBoW	ML+MC	BA
	2022	Method [30]	TC	FAST + DBoW	ML+MC	PGO
LiDAR-visual-inertial	2020	LIC-Fusion 2.0[31]	TC	–	ML+MC+IMU	MSCKF
	2021	Super odometry[32]	TC	–	ML+MC+IMU	FGO
	2021	LVI-SAM[33]	TC	Euclidean distance FAST + DBoW	ML+MC+IMU	FGO
	2022	R3LIVE[34]	TC	–	ML+MC+IMU	Iterated ESKF

Note: (1) Type: FB denotes filtering-based method, OB denotes optimization-based method, LC denotes loo-sely-coupled method, and TC denotes tightly-coupled method. (2) Loop closure: FAST denotes features from accelerated segment test, ORB denotes oriented fast and rotated brief, and DBoW denotes distributed bag of words. (3) Sensor type: MC denotes monocular camera, ML denotes mechanical LiDAR, SL denotes solid-state LiDAR, and RC denotes RGB-D camera. (4) Fusion strategy: FGO denotes factor graph optimization, BA denotes bundle adjustment, SWO denotes sliding window optimization, and PGO denotes pose graph optimization.

3.1 Visual-inertial fusion algorithms

In a navigation system, we want to estimate the six Degree-Of-Freedom (DOF) poses (orientations and positions) of a sensing platform. IMU has been widely used in navigation systems because of its small size, lightweight, low cost, and, most importantly, the ability to measure three-axis angular velocities and linear accelerations of the sensing platform to which it is rigidly attached at high frequency. However, the navigation system with IMU-only suffers from unbounded errors caused by the integration of IMU measurements with bias and noise, and cannot provide reliable pose estimates for long-term navigation. Additional sensors are needed to overcome this problem. A small and lightweight monocular camera that provides good tracking and rich map information about the environment around the sensing platform could serve as one of the idea complementary sensors to IMU. The fusion of IMU and camera yields Visual-Inertial Navigation Systems (VINS) which have attracted significant attention over the last two decades. Generally, VINS algorithms can be divided into optimization-based and filtering-based methods based on the type of data fusion.

3.1.1 Filtering-based methods

To enable efficient estimation, filtering-based methods usually restrict the inference process to the latest state of the system, namely the current camera pose and features observed from it, resulting in the complexity growing quadratically in the number of features. A structureless approach, maintaining a window of camera poses to fully use all features and allow real-time operation, is a good alternative. And MSCKF[35] is an elegant example of the structureless approach, in which a static feature is used to define geometric constraints involving all the camera poses where it is viewed. When a feature goes out of the field of view, its position is estimated using all its measurements by Gauss-Newton minimization. Then residual equations are established, and the introduction of left nullspace makes sure that the residual vector is independent of the feature position errors. The delayed linearization approach does not need the assumption that feature positions are Gaussian distributions at each time

step and its complexity is only linear in the number of features.

However, the MSCKF suffers from inconsistency in long trajectories. Li and Mourikis[16, 36] proved that, for a standard EKF-based VINS, the observability properties of the linearized system do not match those of the underlying nonlinear system because of linearizing the measurement models with updated estimates. Thus they proposed the MSCKF 2.0 algorithm in which the appropriate observability properties are ensured by using the first available estimate for each state when calculating Jacobians. Besides, many other works focus on improving the consistency of the filtering-based methods, such as observability constrained algorithm[37–39], optimal-state-constraint EKF[40], MSCKF-LG[41], robocentric VIO algorithm[42, 43], invariant Kalman filter[44–46], and so on.

When feature positions are included in the state vector, the parameterization of feature positions that have an influence on the consistency has to be considered. The parameterization could be divided into two main approaches: delayed and undelayed initializations. The former usually refers to the Cartesian-coordinate parametrization, where the feature depth with high uncertainty cannot be well-represented by the Gaussian distribution, resulting in degrading accuracy and consistency[16]. To overcome this problem, undelayed initializations, such as inverse-depth feature parametrization, homogeneous feature parametrization, and anchored homogeneous feature parametrization, were proposed to enable features newly detected to be used in filter immediately[47]. The inverse-depth feature parametrization was firstly proposed in monocular SLAM[48, 49] before being adopted by several VINS algorithms[50, 51]. Filtering-based methods have been applied to many platforms because of their high-accuracy state estimation and low computational requirement. Kim and Sukkarieh[52] proposed the first airborne SLAM implementation with actual flight and observation data, where the bias in the accelerometers and gyros are not included in the state vector, and observations provide the relative locations of the landmarks from the UAV. The problem of feature position parameterization and initialization is avoided because of the known size of the landmarks. Lynen et al.[53] proposed a framework for large-scale pose estimation and tracking where the employment of map, descriptor compression schemes, and efficient search algorithms enable real-time performance on mobile platforms with limited resources. Fang et al.[54] proposed a visual-inertial based real-time motion-tracking approach for mobile AR/VR, where an adaptive filter was proposed to alleviate the jitter phenomenon.

3.1.2 Optimization-based methods

Optimization-based methods could be divided into fixed-lag smoothing algorithms and full smoothing algorithms based on the number of camera poses involved in the estimation. The latter estimates all poses and features in history by solving a large nonlinear optimization problem to ensure high accuracy with high computational demand[55–57], while the former only considers a window of recent states.

Methods like MSCKF, also called EKF-based fixed-lag smoothing approaches, are fragile to a gradual accumulation of linearization errors[58], while optimization-based ones process state estimation by solving the least square nonlinear problem where measurements are re-linearized iteratively to treat nonlinearity better. OKVIS[59] is an optimization-based fixed-lag smoothing algorithm, which combines IMU error and the feature reprojection error in a single cost function, and marginalizes old states to bound the complexity. The keyframe paradigm is employed in this method for drift-free estimation, especially when it is slow or there is no motion at all. The use of stereo vision in OKVIS makes the metric scale observable. However, in the monocular case, estimator initialization is a significant challenge, since acceleration excitation is needed to have metric scale observable which implies that monocular VINS estimators cannot start from a stationary condition. Besides, IMU processing and camera-IMU extrinsic calibration have to be considered[60]. And these issues are addressed by VINS-mono[18], as shown in Fig. 1, including five parts: measurement preprocessing, estimator initialization, nonlinear optimization-based VIO, loop closure, and global pose graph optimization. VINS-mono is a robust and versatile monocular visual-inertial estimator, which has been successfully applied to AR[61] and MAVs[62].

State propagation, in the filtering-based methods, is the most straightforward approach to IMU processing. While for optimization-based methods, IMU measurements are typically integrated among frames to form relative motion constraints[56, 59, 63, 64]. However, the state estimate changes at each iteration of optimization, resulting in repeated IMU integration among all frames. To avoid this, Lupton and Sukkarieh[65] first proposed a reparametrization

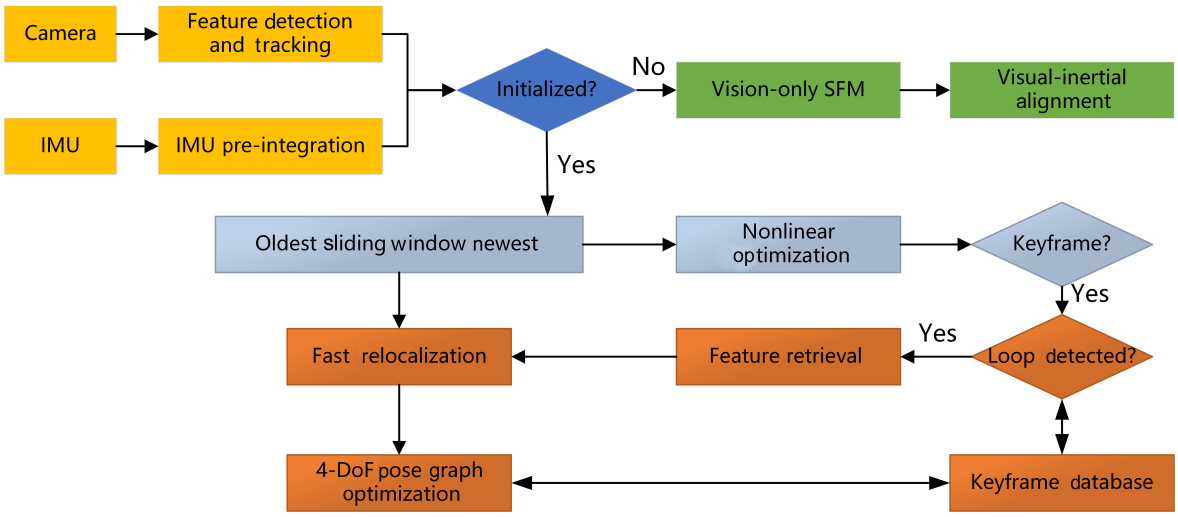

Fig. 1 Pipeline of VINS-mono reproduced from Ref. [18].

of the relative motion constraints, called IMU preintegration, which parametrizes rotation error using Euler angles. Then Shen et al.[66] developed an on-manifold rotation formulation for IMU preintegration and Forster et al.[67, 68] further brought the theory of IMU preintegration to maturity.

3.2 LiDAR-inertial fusion algorithms

In recent years, there has been a growing focus on LiDAR-inertial fusion algorithms, since IMU measures instant motion at a high frequency, which can be utilized to recover point clouds from highly dynamic motion distortion and predict the relative pose between two LiDAR frames. According to sensor fusion type, LiDAR-inertial fusion algorithms can be categorized into either loosely-coupled methods or tightly-coupled methods. Loosely-coupled methods, appealing for runtime, consider the estimation of the LiDAR and the estimation of the IMU separately, resulting in information loss and inaccurate estimates. While tightly-coupled methods, aiming at accurate estimates, fuse point clouds and IMU measurements in an optimization-based or filtering-based framework with higher computational cost. The current state-of-the-art approaches to the two fusion types will be presented in this part.

3.2.1 Loosely-coupled methods

LOAM[69] is a classical 3D LiDAR SLAM method, whose structure is composed of three main modules, namely, feature extraction, odometry, and mapping. The structure has been typically inherited by existing works. In LOAM, edge points and planar points extracted in the growing point cloud of the current sweep are used to find correspondences in the last sweep to update the pose transform from the last recursion. With the assumption of constant angular and linear velocities during a sweep, the pose transform at different times within a sweep can be computed by linear interpolation of the pose transform from the last recursion. However, when velocity changes fast, LOAM suffers from low accuracy, which can be mitigated by IMU. Integrating IMU measurements provides poses of different times during a sweep, which can effectively compensate for the motion distortion, leading to considerably increasing accuracy and robustness[70].

LION[19] is a loosely-coupled LiDAR inertial odometry algorithm that shares similar odometry with LOAM without feature extraction and mapping for low computational cost. The condition number is used in LION as an observability metric to determine whether other more reliable odometry sources need to be used. To make better use of historical frame information, tightly-coupled LiDAR-inertial methods[15, 20–23] usually adopt scan-to-local map registration, where the local map consists of a small number of recent LiDAR frames.

3.2.2 Tightly-coupled methods

LIOM[20] provides the first open-source implementation for the tightly coupled LiDAR inertial fusion method inspired by visual-inertial works[18, 59]. With the same assumption of LiDAR motion as LOAM, the position of each point during a sweep can be corrected by linear interpolation of predicted LiDAR motion by IMU propagation. LIOM maintains a sliding window consisting of current LiDAR sweep and recent sweeps, where the frame of pivot LiDAR sweep is used as the local frame, and all sweeps in the window are

transformed to the local frame to get the local map. LIOM optimizes the pivot LiDAR pose and the following ones rather than the current pose only in the sliding window via a cost function containing the prior items from marginalization, and the residual of the relative LiDAR constraints and IMU constraints. Before carrying out non-linear optimization within the local window, the IMU states are initialized by the methods in VINS-mono[18] with the IMU measurements and the poses of the LiDAR provided by LOAM in the initialization step. To improve the runtime efficiency, LINS[21] introduced an iterated ESKF and robocentric formulation. In LIO-SAM[22], LiDAR-inertial odometry is formulated atop a factor graph[71], making it especially suitable for multi-sensor fusion.

Solid-state LiDAR often has non-repetitive and irregular scan patterns with small FoVs, for which common feature extraction methods are not well suitable. To tackle this, LILI-OM[15] presents the first tightly-coupled solid-state LiDAR-inertial fusion algorithm where a new feature extraction method is developed by performing eigendecomposition for small point patch split in the time domain. Besides, a keyframe scheme is used in sliding window optimization to ensure real-time performance since exploiting all sensor readings is time-consuming. LILI-OM adopts a similar approach as LOAM to compensate for motion distortion, while in FAST-LIO2[23, 70], a back-propagation process is performed. FAST-LIO2 does not extract any features, but directly registers raw points to the map maintained by an incremental k-d tree data structure. The k-d tree proposes supports downsampling on the tree, ensuring the sparsity of the map and fast k-nearest search. Faster-LIO[24], basically develops from the FAST-LIO2, proposes a sparse and incremental voxel-based LiDAR-inertial odometry for fast-tracking.

3.3 Visual-LiDAR fusion algorithms

Visual sensors, such as monocular camera, are usually cheap, and the extraction of visual features enables loop closure detection. However, the vision-based navigation system is sensitive to illumination change and texture deficiency. LiDAR, as an active sensor, shows better accuracy and robustness to changing environments but suffers from structure-less scenarios, such as long corridors, even if rich texture information exists. Due to the complementary strengths of these two types of sensors, several works have been proposed, which can be divided into two categories: loosely-coupled methods and tightly-coupled methods. Some works focus on the frontend integration while others pay attention to the backend optimization, and a detailed discussion about them will be given in the following.

3.3.1 Loosely-coupled methods

Zhang et al.[25, 72] utilized LiDAR depth information to enhance visual odometry in DEMO. They utilized the estimated pose of the camera to register a depth map, where new points from point clouds in the front of the camera are added. The map points are converted into a spherical coordinate system and stored in a 2D k-d tree based on the two angular coordinates. Then, for each feature, the depth can be obtained by projecting onto a planar patch formed by the three nearest points of the feature in the k-d tree. The LiDAR information is not fully exploited in this method. Besides, the undistortion of the LiDAR point cloud is not mentioned. Loop closure detection, which is not considered in this method, was addressed later in Ref. [73] by applying ORB features and bags-of-word. Shin et al.[26, 74] used a similar strategy as DEMO to enhance visual SLAM by depth information from LiDAR. They did not extract features from images like DEMO. Instead, they solved the problem within DSO[75] framework by projecting LiDAR points onto the images as features. Then the same multi-frame photometric optimization as DSO was performed to estimate the poses of the keyframes. Yan et al.[76] simply combined the state-of-the-art visual odometry[77] and LiDAR odometry[69] in a loosely-coupled way that the LiDAR odometry is used only when the visual odometry failed.

To deal with challenging environments, learning methods have been exploited. LIMO[27] levers the power of deep learning to remove features on dynamic objects. LIV-LAM[78] proposes unsupervised learning for object discovery and uses detected features of the objects as landmark features.

3.3.2 Tightly-coupled methods

Zhang and Singh[79] proposed V-LOAM, based on their previous work: DEMO and LOAM[69], without the assistance of IMU measurements to compensate for rapid motion. In V-LOAM, the frequency of the camera is much higher than that of LiDAR, such that the enhanced visual odometry with observable scale could be used to undistort the LiDAR point cloud. Besides, modeling the drift of visual odometry with linear motion within a sweep improves the performance of the undistortion procedure. Then the undistorted point cloud is matched and registered to the currently built map to refine the estimated pose. However, removing distortion heavenly

relies on the result of visual odometry, making it susceptible to texture-less or dynamic environments where visual odometry may fail. Moreover, it is difficult to achieve a recovery mechanism if the last estimate goes wrong because of its frame-to-frame motion estimation[32].

To improve the accuracy and robustness of pose estimation, other environmental structure features, such as line features and planar features, have been leveraged in recent works[28, 80]. Huang et al.[28] introduced a novel visual-LiDAR odometry method using point and line features detected by the line segment detector[81] and described by the line band descriptor[82]. Huang et al.[80] proposed a grid-based method to explicitly detect scene planes from the point cloud to include as much as possible pixel information in the photometric term. To reduce the deterioration of occluded points, they exploited a novel method to predict which LiDAR points would be occluded during the viewpoint change. Seo and Chou[83] attempted to make full use of visual and LiDAR measurements in a novel way to avoid the potential issue of assigning the depths of LiDAR to non-corresponding visual features. They maintained visual and LiDAR measurements separately and built two different maps, an LiDAR voxel map and a visual map, which were used together when to solve the residuals for pose estimation. Wang et al.[84] proposed a direct vision LiDAR fusion SLAM framework, similar to DVL-SLAM[26, 74]. To get better robustness in various complex environments, a frame-to-frame tracking strategy, an LiDAR-based scan-to-map matching method, and a Parallel Global and Local Search Loop Closure Detection (PGLS-LCD) module are used in their framework.

Camera-LiDAR extrinsic calibration, which is usually ignored in existing works, has been considered in the most recent work. TVL-SLAM[29] is a tightly-coupled visual-LiDAR fusion algorithm, where the visual and LiDAR measurements are used independently in the frontend instead of enhancing one via another, while all measurements are incorporated in the backend optimization in a tightly coupled way. It is assumed that LiDAR point cloud and stereo image pair are acquired at the same timestamp and the camera-LiDAR extrinsic is known and fixed before global bundle adjustments, making it possible to refine the pose using all visual and LiDAR residuals by solving a bunch optimization. The camera-LiDAR extrinsic is estimated in global bundle adjustment when a visual or LiDAR loop is detected, and each visual map point is matched to the nearest LiDAR voxel to create a constraint, ensuring good convergence. Moving object removal in a stop-and-run scenario was also discussed, but only visual features were considered. Meng et al.[30] also jointly optimized visual and LiDAR measurements in a unified framework like TVL-SLAM except the visual features were enhanced by LiDAR depth information.

3.4 LiDAR-visual-inertial fusion algorithms

LiDAR-only approaches are vulnerable to environments with degenerate geometries, such as long tunnels or wide-open spaces. IMU measurements could be an excellent supplement to LiDAR-only methods, however, they only provide reliable pose estimates within a few seconds. Therefore, LiDAR-inertial methods also suffer from the degenerate case, especially for solid-state LiDAR, whose FOV is small. To cope with these issues, fusing with other sensors, particularly cameras which provide rich visual information is necessary and has been attached growing attention. For consistency, we also divide LiDAR-visual-inertial methods into two categories as above.

3.4.1 Loosely-coupled methods

Shao et al.[85] proposed a VIL-SLAM, which uses stereo cameras as visual sensors to achieve better performance in certain degenerate cases like traveling through a tunnel, where the pure LiDAR system usually fails. By fusing stereo matches and IMU measurements in a tightly-coupled fixed-lag smoothing, the stereo VIO outputs IMU-rate and camera-rate VIO pose, which is used to remove motion distortion and perform scan-to-map registration in LiDAR mapping. They used pure visual information to detect loop closure and construct initial loop constraint estimation, which was further refined by LiDAR measurements. Similar work was proposed by Wang et al.[86] with additional consideration about module failure. Camurri et al.[87] presented a loosely-coupled framework for legged robots operating in real-world scenarios, and Khattak et al.[88] presented a complementary multi-modal sensor fusion approach for aerial robot pose estimation in subterranean environments.

3.4.2 Tightly-coupled methods

Zhang and Singh[89] presented a sequential, multilayer processing pipeline, where the motion is firstly predicted by IMU measurements, then estimated by visual-inertial odometry, and finally refined by scan-to-map registration. To compensate for possible calibration variations

among the un-synchronized sensors, Zuo et al.[90] proposed a lightweight processing pipeline, called LIC-Fusion, within the MSCKF framework. To efficiently and robustly process the LiDAR measurements, they additionally introduced a novel planar feature tracking algorithm to LIC-Fusion and proposed LIC-Fusion 2.0[31], where planar points are extracted from LiDAR points after removing distortion by IMU measurements and tracked across the sliding window with an outlier rejection criteria proposed for higher quality data association by taking into account the uncertainty of the LiDAR scan transformations.

Loosely-coupled methods are known for their simplicity, extendibility, and low computational demand, while tightly-coupled methods show better performance in terms of accuracy and robustness. To combine the advantages of loosely-coupled methods with tightly-coupled methods, Super odometry[32] employs an IMU-centric data processing pipeline, which consisted of three parts: IMU odometry, visual-inertial odometry, and LiDAR-inertial odometry. The IMU bias is constrained by the pose prior provided by the visual-inertial odometry and LiDAR-inertial odometry, which receives the motion prediction from IMU odometry. Besides, a dynamic octree is applied to ensure high performance in real-time. The key insight behind their design is that the estimate of IMU odometry can be quite accurate if the bias drift is well-constrained by other sensors since the IMU produces smooth measurements with noise but little outliers.

By integrating VINS-mono and LIO-SAM, Shan et al.[33] proposed a publicly-available system, LVI-SAM, which is built atop a factor graph and composed of two sub-systems, an Visual-Inertial System (VIS) and an LiDAR-Inertial System (LIS). Different from Super Odometry, in LVI-SAM, feature depth could be optionally extracted from LiDAR scans using a depth association method, and candidate matches for loop closure are first identified by the VIS and further optimized by the LIS. A factor graph is used to optimize all constraints jointly from VIS, LIS, IMU preintegration, and loop closure.

To achieve real-time performance, Lin et al.[91] proposed a framework of error-state iterated Kalman filter, where the LiDAR point-to-plane residuals, the image re-projection errors, and the IMU propagation are fused tightly. For each image of camera input, fast corners are detected and tracked with a map of visual landmarks to compute the re-projection error. Besides, a factor graph optimization is exploited to further improve the accuracy of visual measurements within a local sliding window. Instead of extracting features from LiDAR point clouds and images, Zheng et al. proposed FAST-LIVO[92], which is composed of two direct odometry subsystems: an LIO subsystem directly adapted from FAST-LIO2[70] and a VIO subsystem similar to Ref. [93]. The points of the map built by the LIO are additionally attached with image patches and then used to align a new image in VIO by minimizing the direct photometric errors, leading to a time-saving backend. A similar framework is adopted by R3LIVE[34] with additional the Perspective-n-Point (PnP) projection error. Loop closure is not enabled in the above three methods and the LiDAR sensor is solid-state LiDAR.

4 Challenge and Future Research Direction

Although a lot of multi-sensor fusion algorithms with different frameworks have been proposed in recent years, there are still several challenges, such as sensor-to-sensor calibration, efficient data association, good initialization, and dynamic environments.

In terms of camera-to-IMU calibration, early methods[16, 94–97] depended on artificial markers or accurate initialization. To tackle these issues, Yang and Shen[98] proposed a methodology that is able to get accurate camera-IMU extrinsic calibration on the fly. However, their method assumed that sufficient features could be tracked. For camera-LiDAR calibration, Geiger et al.[99] proposed an automatic approach using a single shot by detecting and matching special marker boards in both camera and LiDAR FOVs. Markerless calibration by maximizing mutual information among the sensor-measured surface intensities was introduced in Ref. [100]. To ensure good convergence, Chou and Chou[29] made the extrinsic an adjustable variable in global bundle adjustment by adding pure geometry constraints via registration between visual map points and LiDAR voxel maps, while this method needs good initial values and vision-only methods might not provide visual map points with accurate scale.

More data can bring higher accuracy, but it usually demands more computational resources. As shown in Fig. 2, super odometry[32] consists of three parts: IMU odometry, visual-inertial odometry, and LiDAR-inertial odometry, which means that real-time performance on a limited-resource platform is not guaranteed. Fusing the results of odometry may take much more time than directly associating data from sensors. LVI-

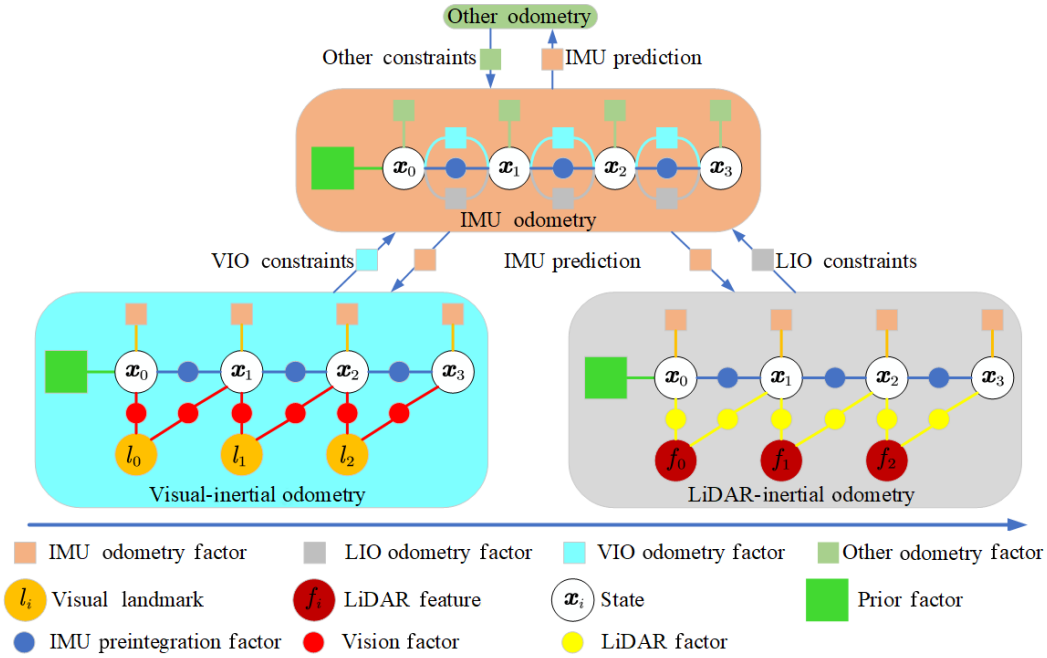

Fig. 2 Overview of super odometry algorithm reproduced from Ref. [32].

SAM[33] takes advantage of accurate LiDAR depth information to greatly promote the visual-inertial odometry initialization and R3LIVE[34] directly exploits the LiDAR point cloud for feature tracking on the image without feature extraction and triangulation, which remarkably accelerates the visual-inertial odometry. However, this approach may fail when mechanical LiDAR is used. More general and efficient data association is still challenging in the LiDAR-Visual-Inertial fusion algorithm.

Due to the nonlinearity of the visual-inertial methods, a poor initialization can have a dramatic impact on their performance. By leveraging relative rotations from short-term IMU pre-integration, Refs. [98, 101] proposed a linear estimator initialization method without gyroscope bias, resulting in unreliable initialization when visual features are far away from the sensor suite[18]. A closed-form solution to the visual-inertial Structure from Motion (SfM) problem was derived in Ref. [102] and improved in Ref. [103] by modeling the gyroscope bias. Built on top of ORB-SLAM[104], Ref. [105] introduced an IMU initialization method, which requires a few seconds for scale convergence. To achieve a fast and robust initialization, Qin and Shen[60] aligned metric IMU pre-integration with the visual-only SfM results to get initial values. Instead of SfM, Cheng et al.[106] used the ORB-SLAM for faster convergence. Besides, new methods[107–110] are emerging recently for faster and more accurate initialization.

Most existing fusion algorithms assume that the environment is static; however, this is not always the case in the real world. For example, walking people and moving vehicles are common dynamic objects existing in the real world. Point clouds from dynamic objects will deteriorate the accuracy of scan-to-map registration or scan-to-scan registration, leading to wrong relative pose estimation. Compared with mechanical LiDAR, the field of view of the camera is much smaller, making it more vulnerable to moving objects. Suppose the tracked features on the moving objects are not properly rejected. In that case, the motion estimator will compute a false motion, which further deteriorates the local or global optimization and causes the system to fail[29].

According to the literature reviewed and the above challenges, we propose some future research directions:

- **Versatile and efficient fusion framework:** The current state of the art of algorithms are generally designed for particular platforms, making them hard to deploy on other platforms with similar sensors. Automatic sensor-to-sensor calibration is vital and accurate initialization should be guaranteed, especially for platforms equipped with visual sensors. Besides, efficient data association should be exploited to ensure real-time performance.
- **Deep learning aided methods:** It is a growing field in multi-sensor fusion framework to exploit deep learning, which can be used for feature extraction, moving objects detection, environment presentation, and

so on.

• **Distributed cooperative methods:** Different robots equipped with different sensors for the same SLAM task will significantly reduce the burden of the single robot, while this is a quite challenging problem and there is little literature about it.

5 Conclusion

The multi-sensor fusion technology has gained growing attention recently in the field of robotics. This study provided a brief introduction to famous state estimate formation and summarized multi-sensor fusion methods over the last ten years. We firstly divided the multi-sensor fusion algorithms into four categories according to the combination of sensors and then classified them based on data fusion. The most exemplary techniques of each method are presented. In addition, challenges and future research directions are discussed to make the technology versatile, robust, and substantial.

Acknowledgment

This work was supported by the Scientific and Technological Innovation 2030 (No. 2021ZD0110900).

References

[1] R. C. Smith and P. Cheeseman, On the representation and estimation of spatial uncertainty, *Int. Journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.

[2] T. Bailey and H. Durrant-Whyte, Simultaneous localization and mapping (SLAM): Part II, *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[3] A. R. Khairuddin, M. S. Talib, and H. Haron, Review on simultaneous localization and mapping (SLAM), in *Proc. 5th IEEE Int. Conf. on Control System, Computing and Engineering*, Batu Ferringhi, Malaysia, 2015, pp. 85–90.

[4] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age, *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[5] Z. J. Wang, Y. Wu, and Q. Q. Niu, Multi-sensor fusion in automated driving: A survey, *IEEE Access*, vol. 8, pp. 2847–2868, 2020.

[6] M. Chghaf, S. Rodriguez, and A. El Ouardi, Camera, LiDAR and multi-modal SLAM systems for autonomous ground vehicles: A survey, *Journal of Intelligent & Robotic Systems*, vol. 105, no. 1, pp. 1–35, 2022.

[7] A. Singandhupe and L. H. Manh, A review of SLAM techniques and security in autonomous driving, in *Proc. 3rd IEEE Int. Conf. on Robotic Computing*, Naples, Italy, 2019, pp. 602–607.

[8] D. N. Van and G. W. Kim, Multi-sensor fusion towards VINS: a concise tutorial, survey, framework and challenges, in *Proc. IEEE Int. Conf. on Big Data and Smart Computing*, Busan, Republic of Korea, 2020, pp. 459–462.

[9] Y. Alkendi, L. Seneviratne, and Y. Zweiri, State of the art in vision-based localization techniques for autonomous navigation systems, *IEEE Access*, vol. 9, pp. 76847–76874, 2021.

[10] G. Huang, Visual-inertial navigation: a concise review, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Montreal, Canada, 2019, pp. 9572–9582.

[11] C. Chen, H. Zhu, M. G. Li, and S. Z. You, A review of visual-inertial simultaneous localization and mapping from filtering-based and optimization-based perspectives, *Robotics*, vol. 7, no. 3, p. 45, 2018.

[12] M. Servieres, V. Renaudin, A. Dupuis, and N. Antigny, Visual and visual-inertial SLAM: state of the art, classification, and experimental benchmarking, *Journal of Sensors*, vol. 2021, pp. 1–26, 2021.

[13] A. Macario Barros, M. Michel, Y. Moline, G. Corre, and F. Carrel, A comprehensive survey of visual SLAM algorithms, *Robotics*, vol. 11, no. 1, p. 24, 2022.

[14] C. Debeunne and D. Vivet, A review of visual-LiDAR fusion based simultaneous localization and mapping, *Sensors*, vol. 20, no. 7, p. 2068, 2020.

[15] K. L. Li, M. Li, and U. D. Hanebeck, Towards high-performance solid-state-LiDAR-inertial odometry and mapping, *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5167–5174, 2021.

[16] M. Y. Li and A. I. Mourikis, High-precision, consistent EKF-based visual-inertial odometry, *Int. Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

[17] M. Bloesch, S. Omani, M. Hutter, and R. Siegwart, Robust visual inertial odometry using a direct EKF-based approach, in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Hamburg, Germany, 2015, pp. 298–304.

[18] T. Qin, P. L. Li, and S. J. Shen, VINS-Mono: A robust and versatile monocular visual-inertial state estimator, *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[19] A. Tagliabue, J. Tordesillas, X. Y. Cai, A. Santamaria-Navarro, J. P. How, L. Carlone, and A. A. Aghamohammadi, LION: LiDAR-inertial observability-aware navigator for vision-denied environments, *Springer Proceedings in Advanced Robotics*, vol. 19, pp. 380–390, 2021.

[20] H. Y. Ye, Y. Y. Chen, and M. Liu, Tightly coupled 3D LiDAR inertial odometry and mapping, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Montreal, Canada, 2019, pp. 3144–3150.

[21] C. Qin, H. Y. Ye, C. E. Pranata, J. Han, S. Y. Zhang, and M. Liu, LINS: A LiDAR-inertial state estimator for robust

and efficient navigation, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Virtual Event, 2020, pp. 8899–8905.

[22] T. X. 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. on Intelligent Robots and Systems*, Virtual Event, 2020, pp. 5135–5142.

[23] W. Xu and F. Zhang, FAST-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter, *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, 2021.

[24] C. Bai, T. Xiao, Y. J. Chen, H. Q. Wang, F. Zhang, and X. Gao, Faster-LIO: lightweight tightly coupled LiDAR-inertial odometry using parallel sparse incremental voxels, *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4861–4868, 2022.

[25] J. Zhang, M. Kaess, and S. Singh, A real-time method for depth enhanced visual odometry, *Autonomous Robots*, vol. 41, no. 1, pp. 31–43, 2017.

[26] Y. S. Shin, Y. S. Park, and A. Kim, DVL-SLAM: sparse depth enhanced direct visual-LiDAR SLAM, *Autonomous Robots*, vol. 44, no. 2, pp. 115–130, 2020.

[27] J. Graeter, A. Wilczynski, and M. Lauer, LIMO: LiDAR-monocular visual odometry, in *Proc. 25th IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Madrid, Spain, 2018, pp. 7872–7879.

[28] S. S. Huang, Z. Y. Ma, T. J. Mu, H. B. Fu, and S. M. Hu, LiDAR-monocular visual odometry using point and line features, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Virtual Event, 2020, pp. 1091–1097.

[29] C. C. Chou and C. F. Chou, Efficient and accurate tightly-coupled visual-LiDAR SLAM, *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14509–14523, 2021.

[30] L. B. Meng, C. Ye, and W. Y. Lin, A tightly coupled monocular visual LiDAR odometry with loop closure, *Intelligent Service Robotics*, vol. 15, no. 1, pp. 129–141, 2022.

[31] X. X. Zuo, Y. L. Yang, P. Geneva, J. J. Lv, Y. Liu, G. Q. Huang, and M. Pollefeys, LIC-Fusion 2.0: LiDAR-inertial-camera odometry with sliding-window plane-feature tracking, in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Virtual Event, 2020, pp. 5112–5119.

[32] S. B. Zhao, H. R. Zhang, P. Wang, L. Nogueira, and S. Scherer, Super odometry: IMU-centric LiDAR-visual-inertial estimator for challenging environments, in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Virtual Event, 2021, pp. 8729–8736.

[33] T. X. 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. on Robotics and Automation*, Xi'an, China, 2021, pp. 5692–5698.

[34] J. R. Lin and F. Zhang, R3LIVE: A robust, real-time, RGB-colored, LiDAR-inertial-visual tightly-coupled state estimation and mapping package, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Philadelphia, AZ, USA, 2022. pp. 10672–10678.

[35] A. I. Mourikis and S. I. Roumeliotis, A multi-state constraint Kalman filter for vision-aided inertial navigation, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Rome, Italy, 2007, pp. 3565–3572.

[36] M. Y. Li and A. I. Mourikis, Improving the accuracy of EKF-based visual-inertial odometry, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Saint Paul, MN, USA, 2012, pp. 828–835.

[37] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, Consistency analysis and improvement of vision-aided inertial navigation, *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 158–176, 2014.

[38] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, Camera-IMU-based localization: observability analysis and consistency improvement, *Int. Journal of Robotics Research*, vol. 33, no. 1, pp. 182–201, 2014.

[39] G. Q. Huang, M. Kaess, and J. J. Leonard, Towards consistent visual-inertial navigation, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Hong Kong, China, 2014, pp. 4926–4933.

[40] G. Q. Huang, K. Eckenhoff, and J. Leonard, Optimal-state-constraint EKF for visual-inertial navigation, in *Proc. 12th Int. Symposium on Robotics Research*, Sestri Levante, Italy, 2015, pp. 125–139.

[41] S. Heo and C. G. Park, Consistent EKF-based visual-inertial odometry on matrix Lie group, *IEEE Sensors Journal*, vol. 18, no. 9, pp. 3780–3788, 2018.

[42] Z. Huai and G. Q. Huang, Robocentric visual-inertial odometry, in *Proc. 25th IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Madrid, Spain, 2018, pp. 6319–6326.

[43] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback, *Int. Journal of Robotics Research*, vol. 36, no. 10, pp. 1053–1072, 2017.

[44] M. Brossard, S. Bonnabel, and A. Barrau, Invariant Kalman filtering for visual inertial SLAM, in *Proc. 21st Int. Conf. on Information Fusion*, Cambridge, UK, 2018, pp. 2021–2028.

[45] K. Z. Wu, T. Zhang, D. Su, S. D. Huang, and G. Dissanayake, An invariant-EKF VINS algorithm for improving consistency, in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vancouver, Canada, 2017, pp. 1578–1585.

[46] T. Zhang, K. Z. Wu, J. W. Song, S. D. Huang, and G. Dissanayake, Convergence and consistency analysis for a 3–D invariant-EKF SLAM, *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 733–740, 2017.

[47] J. Sola, Consistency of the monocular EKF-SLAM algorithm for three different landmark parametrizations, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Anchorage, AK, USA, 2010, pp. 3513–3518.

[48] J. Civera, A. J. Davison, and J. M. M. Montiel, Inverse depth parametrization for monocular SLAM, *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, 2008.

[49] J. Civera, A. J. Davison, and J. M. M. Montiel, Inverse depth to depth conversion for monocular SLAM, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Rome, Italy, 2007, pp. 2778–2783.

[50] P. Pinies, T. Lupton, S. Sukkarieh, and J. D. Tardos, Inertial aiding of inverse depth SLAM using a monocular camera, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Rome, Italy, 2007, pp. 2797–2802.

[51] M. Kleinert and S. Schleith, Inertial aided monocular SLAM for GPS-denied navigation, in *Proc. IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, Salt Lake City, UT, USA, 2010, pp. 20–25.

[52] J. H. Kim and S. Sukkarieh, Airborne simultaneous localisation and map building, in *Proc. 20th IEEE Int. Conf. on Robotics and Automation*, Taipei, China, 2003, pp. 406–411.

[53] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart, Get out of my lab: large-scale, real-time visual-inertial localization, in *Proc. 11th Conf. on Robotics - Science and Systems*, Rome, Italy, 2015, p. 1.

[54] W. Fang, L. Y. Zheng, H. J. Deng, and H. B. Zhang, Real-time motion tracking for mobile augmented/virtual reality using adaptive visual-inertial fusion, *Sensors*, vol. 17, no. 5, p. 1037, 2017.

[55] M. Bryson, M. Johnson-Roberson, and S. Sukkarieh, Airborne smoothing and mapping using vision and inertial sensors, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Kobe, Japan, 2009, pp. 3143–3148.

[56] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, Information fusion in navigation systems via factor graph based incremental smoothing, *Robotics and Autonomous Systems*, vol. 61, no. 8, pp. 721–738, 2013.

[57] A. Patron-Perez, S. Lovegrove, and G. Sibley, A spline-based trajectory representation for sensor fusion and rolling shutter cameras, *Int. Journal of Computer Vision*, vol. 113, no. 3, pp. 208–219, 2015.

[58] T. C. Dong-Si and A. I. Mourikis, Motion tracking with fixed-lag smoothing: algorithm and consistency analysis, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Shanghai, China, 2011, pp. 5655–5662.

[59] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, Keyframe-based visual-inertial odometry using nonlinear optimization, *Int. Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[60] T. Qin and S. J. Shen, Robust initialization of monocular visual-inertial estimation on aerial robots, in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vancouver, Canada, 2017, pp. 4225–4232.

[61] P. L. Li, T. Qin, B. T. Hu, F. Y. Zhu, and S. J. Shen, Monocular visual-inertial state estimation for mobile augmented reality, in *Proc. 16th IEEE Int. Conf. on Symposium on Mixed and Augmented Reality*, Nantes, France, 2017, pp. 11–21.

[62] Y. Lin, F. Gao, T. Qin, W. L. Gao, T. B. Liu, W. Wu, Z. F. Yang, and S. J. Shen, Autonomous aerial navigation using monocular visual-inertial fusion, *Journal of Field Robotics*, vol. 35, no. 1, pp. 23–51, 2018.

[63] V. Indelman, A. Melim, and F. Dellaert. Incremental light bundle adjustment for robotics navigation, in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Tokyo, Japan, 2013, pp. 1952–1959.

[64] N. Keivan and G. Sibley, Asynchronous adaptive conditioning for visual-inertial SLAM, *Int. Journal of Robotics Research*, vol. 34, no. 13, pp. 1573–1589, 2015.

[65] T. Lupton and S. Sukkarieh, Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions, *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 61–76, 2012.

[66] S. J. Shen, N. Michael, and V. Kumar, Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, USA, 2015, pp. 5303–5310.

[67] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation, in *Proc. 11th Conf. on Robotics - Science and Systems*, Rome, Italy, 2015, pp. 1–10.

[68] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, On-manifold preintegration for real-time visual-inertial odometry, *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2017.

[69] J. Zhang and S. Singh. LOAM: LiDAR odometry and mapping in real-time, *Robotics- Science and Systems*, vol. 2, no. 9, pp. 1–9, 2014.

[70] W. Xu, Y. X. Cai, D. J. He, J. R. Lin, and F. Zhang, FAST-LIO2: fast direct LiDAR-inertial odometry, *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[71] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, iSAM2: incremental smoothing and mapping using the Bayes tree, *Int. Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[72] J. Zhang, M. Kaess, and S. Singh, Real-time depth enhanced monocular odometry, in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* , Chicago, IL, USA, 2014, pp. 4973–4980.

[73] X. Liang, H. Y. Chen, Y. J. Li, and Y. H. Liu, Visual laser-SLAM in large-scale indoor environments, in *Proc. IEEE Int. Conf. on Robotics and Biomimetics*, Qingdao, China, 2016, pp. 19–24.

[74] Y. S. Shin, Y. S. Park, and A. Kim, Direct visual SLAM using sparse depth for camera-LiDAR system, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Brisbane, Australia, 2018, pp. 5144–5151.

[75] P. Tanskanen, T. Naegeli, M. Pollefeys, and O. Hilliges, Semi-direct EKF-based monocular visual-inertial odometry, in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* , Hamburg, Germany, 2015, pp. 6073–6078.

[76] M. Yan, J. Z. Wang, J. Li , and C. Zhang, Loose coupling visual-LiDAR odometry by combining VISO2 and LOAM, in *Proc. 36th Conf. on Chinese Control Conf.* , Dalian, China, 2017, pp. 6841–6846.

[77] A. Geiger, J. Ziegler, and C. Stiller, StereoScan: dense 3d reconstruction in real-time, in *Proc. IEEE Intelligent Vehicles Symposium* , Baden-Baden, Germany, 2011, pp. 963–968.

[78] R. Radmanesh, Z. Y. Wang, V. S. Chipade, G. Tsechpenakis, and D. Panagou, LIV-LAM: LiDAR and visual localization and mapping, in *Proc. American Control Conf.* , Denver, CO, USA, 2020, pp. 659–664.

[79] J. Zhang and S. Singh, Visual-LiDAR odometry and mapping: low-drift, robust, and fast, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, USA, 2015, pp. 2174–2181.

[80] K. H. Huang, J. H. Xiao, and C. Stachniss, Accurate direct visual-laser odometry with explicit occlusion handling and plane detection, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Montreal, Canada, 2019, pp. 1295–1301.

[81] R. G. von Gioi, J. Jakubowicz, J. M. Morel, and G. Randall, LSD: a fast line segment detector with a false detection control, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 722–732, 2010.

[82] L. L. Zhang and R. Koch, An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency, *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 794–805, 2013.

[83] Y. Seo and C. C. Chou, A tight coupling of vision-LiDAR measurements for an effective odometry, in *Proc. 30th IEEE Intelligent Vehicles Symposium*, Paris, France, 2019, pp. 1118–1123.

[84] W. Wang, J. Liu, C. J. Wang, B. Luo, and C. Zhang, DV-LOAM: direct visual LiDAR odometry and mapping, *Remote Sensing*, vol. 13, no. 16, p. 3340, 2021.

[85] W. Z. Shao, S. Vijayarangan, C. Li, and G. Kantor, Stereo visual inertial LiDAR simultaneous localization and mapping, in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* , Macau, China, 2019, pp. 370–377.

[86] Z. Y. Wang, J. H. Zhang, S. Y. Chen, C. E. Yuan, J. Q. Zhang, and J. W. Zhang, Robust high accuracy visual-inertial-laser SLAM system, in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* , Macau, China, 2019, pp. 6636–6641.

[87] M. Camurri, M. Ramezani, S. Nobili, and M. Fallon, Pronto: a multi-sensor state estimator for legged robots in real-world scenarios, *Frontiers in Robotics and Ai*, vol. 7, p. 68, 2020.

[88] S. Khattak, H. Nguyen, F. Mascarich, D. Tung, and K. Alexis, Complementary multi-modal sensor fusion for resilient robot pose estimation in subterranean environments, in *Proc. Int. Conf. on Unmanned Aircraft Systems*, Athens, Greece, 2020, pp. 1024–1029.

[89] J. Zhang and S. Singh, Laser-visual-inertial odometry and mapping with high robustness and low drift, *Journal of Field Robotics*, vol. 35, no. 8, pp. 1242–1264, 2018.

[90] X. X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Q. Huang, LIC-Fusion: LiDAR-inertial-camera odometry, in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* , Macau, China, 2019, pp. 5848–5854.

[91] J. R. Lin, C. R. Zheng, W. Xu, and F. Zhang, R2LIVE: A robust, real-time, LiDAR-inertial-visual tightly-coupled state estimator and mapping, *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7469–7476, 2021.

[92] C. R. Zheng, Q. Y. Zhu, W. Xu, X. Y. Liu, Q. Z. Guo, and F Zhang, FAST-LIVO: fast and tightly-coupled sparse-direct LiDAR-inertial-visual odometry, in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Kyoto, Japan, 2020, pp. 4003-4009.

[93] C. Forster, M. Pizzoli, and D. Scaramuzza, SVO: fast semi-direct monocular visual odometry, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Hong Kong, China, 2014, pp. 15–22.

[94] J. Kelly and G. S. Sukhatme, Visual-inertial sensor fusion: localization, mapping and sensor-to-sensor self-calibration, *Int. Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, 2011.

[95] E. S. Jones and S. Soatto, Visual-inertial navigation, mapping and localization: a scalable real-time causal approach, *Int. Journal of Robotics Research*, vol. 30, no. 4, pp. 407–430, 2011.

[96] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Saint Paul, MN, USA, 2012, pp. 957–964.

[97] L. Heng, G. H. Lee, and M. Pollefeys, Self-calibration and visual SLAM with a multi-camera system on a micro aerial vehicle, *Autonomous Robots*, vol. 39, no. 3, pp. 259–277, 2015.

[98] Z. F. Yang and S. J. Shen, Monocular visual-inertial state estimation with online initialization and camera-IMU extrinsic calibration, *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 39–51, 2017.

[99] A. Geiger, F. Moosmann, O. Car, and B. Schuster, Automatic camera and range sensor calibration using a single shot, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Saint Paul, MN, USA, 2012, pp. 3936–3943.

[100] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, Automatic extrinsic calibration of vision and LiDAR by maximizing mutual information, *Journal of Field Robotics*, vol. 32, no. 5, pp. 696–722, 2015.

[101] S. J. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar. Initialization-free monocular visual-inertial state estimation with application to autonomous MAVs, in *14th Int. Symposium on Experimental Robotics*, Marrakech, Morocco, 2014, pp. 211–227.

[102] A. Martinelli, Closed-form solution of visual-inertial

structure from motion, *Int. Journal of Computer Vision*, vol. 106, no. 2, pp. 138–152, 2014.

[103] J. Kaiser, A. Martinelli, F. Fontana, and D. Scaramuzza, Simultaneous state initialization and gyroscope bias calibration in visual inertial aided navigation, *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 18–25, 2017.

[104] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, ORB-SLAM: a versatile and accurate monocular SLAM system, *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[105] R. Mur-Artal and J. D. Tardos, Visual-inertial monocular SLAM with map reuse, *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.

[106] J. Cheng, L. Y. Zhang, and Q. H. Chen, An improved initialization method for monocular visual-inertial SLAM, *Electronics*, vol. 10, no. 24, p. 3063, 2021.

[107] B. Yathirajam, V. S. Meenakshisundaram, and A. C. Muniyappa, An efficient approach to initialization of visual-inertial navigation system using closed-form solution for autonomous robots, *Journal of Intelligent & Robotic Systems*, vol. 101, no. 3, pp. 1–21, 2021.

[108] W. B. Huang, W. W. Wan, and H. Liu, Optimization-based online initialization and calibration of monocular visual-inertial odometry considering spatial-temporal constraints, *Sensors*, vol. 21, no. 8, p. 2673, 2021.

[109] W. B. Huang, H. Liu, and W. W. Wan, An online initialization and self-calibration method for stereo visual-inertial odometry, *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1153–1170, 2020.

[110] L. X. Feng, Initialization improvement and map reuse based on ORBSLAM3, in *Proc. 2nd Int. Conf. on Artificial Intelligence and Information Systems*, Chongqing, China, 2021, pp. 1–7.

Tao Zhang received the BEng, MEng, and PhD degrees in control science and engineering from Tsinghua University, Beijing, China in 1993, 1995, and 1999, respectively, and the second PhD degree from Saga University, Saga, Japan in 2002. He was a visiting associate professor at Saga University, Japan from 1999 to 2003. From 2003 to 2006, he worked as a researcher at the National Institute of Informatics, Tokyo, Japan.

He is currently a professor and serves as the dean of Department of Automation, Tsinghua University, Beijing. He has authored or coauthored more than 200 papers and eight books. He is a fellow of IET, and a member of the American Institute of Aeronautics and Astronautics and Institute of Electronics, Information and Communication Engineers. He currently serves as the editorial board member and technical editor for *IEEE/ASME Transactions on Mechatronics*. His current research interests include robotics, image processing, control theory, artificial intelligent, navigation, and control of spacecraft.

Jun Zhu received the BEng degree in automation from Beihang University, China in 2020. He is currently a PhD candidate at Department of Automation, Tsinghua University, China. His research interests include multi-sensor fusion localization and mapping, and their applications in indoor environment reconstruction.

Hongyi Li received the BEng degree in energy and power engineering from Huazhong University of Science and Technology, China in 2020. He is currently a master student at Department of Automation, Tsinghua University, China. His research interests include multi-sensor fusion localization and mapping, and their applications in autonomous driving.