Progress in Medical Devices
ISSN: 2957-5478

Zentime

Review Article https://doi.org/10.61189/521889ygxkmc

Overview of the current development in Visual-Inertial Systems

Mingxia Wei, Qingyun Meng

School of Health Science and Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China.

Corresponding author: Qingyun Meng.

Declaration of conflict of interest: None.

Received May 15, 2024; Accepted June 25, 2024; Published December 31, 2024

Highlights

- A comprehensive overview of Visual-Inertial Navigation Systems.
- Exploration of key technologies, including image processing methods for visual odometry.

Abstract

The Visual-inertial navigation system (VINS) integrates visual and inertial sensors, providing advanced navigation capabilities. Recent improvements in these sensors have made VINS widely applicable in areas such as robotic navigation and autonomous driving, due to its complementary functionality and decreasing sensor costs. This paper reviews the latest developments in VINS, introducing visual simultaneous localization and mapping and its role in VINS. Key technologies, including direct and indirect methods for image processing in visual odometry and Inertial Measurement Unit preintegration for robot motion estimation, are discussed. Both filter-based and optimization-based state estimation methods in VINS are examined. Filter-based methods, like the Extended Kalman Filter, offer real-time state updates for attitude tracking and map construction, while optimization-based methods focus on minimizing reprojection or other error metrics to improve localization accuracy and robustness. The application of dynamic simultaneous localization and mapping, which addresses dynamic objects in complex environments, is also explored. This paper summarizes current research challenges and proposes future directions in the field of dynamic simultaneous localization and mapping.

Keywords: Visual-Inertial Navigation System, visual simultaneous localization and mapping, extended Kalman Filter, dynamic simultaneous localization and mapping, semantic methods

Introduction

Simultaneous localization and mapping (SLAM) is a computational process of simultaneously constructing or updating maps of unknown environments while tracking an agent's position within them. It is a key concept in robotics and autonomous driving, with a development history spanning several decades. Early research focused on fundamental SLAM, such as determining a robot's position and mapping the environment without prior knowledge, primarily relying on probabilistic methods and estimation theory [1].

As SLAM research evolved towards more practical solutions—particularly in handling large-scale environments and managing the increasing computational complexity of accumulating data—sensor fusion approaches gained prominence, with Visual-inertial navigation system (VINS) attracting significant attention [2]. VINS employs cameras and inertial measurement units (IMUs) as sensors to measure displacement and attitude changes. Visual SLAM (VSLAM), an integrated technology combining localization and mapping, relies on extracting feature information or pixel grayscale from image sequences captured by visual sensors to infer the camera's pose. VSLAM provides rich environmental information and avoids drift when static, delivering strong localization and mapping performance in static environments. In contrast, IMUs, independent of external sig-

Address correspondence to: Qingyun Meng, School of Health Science and Engineering, University of Shanghai for Science and Technology, Yangpu District, Shanghai 200093, China. Tel: +86-13761813609. E-mail: mengqy@sumhs.edu.cn.

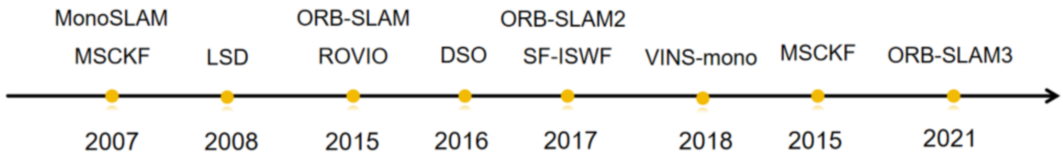

Figure 1. Classic filtering based and optimization based VINS algorithms. VINS, Visual-inertial navigation system.

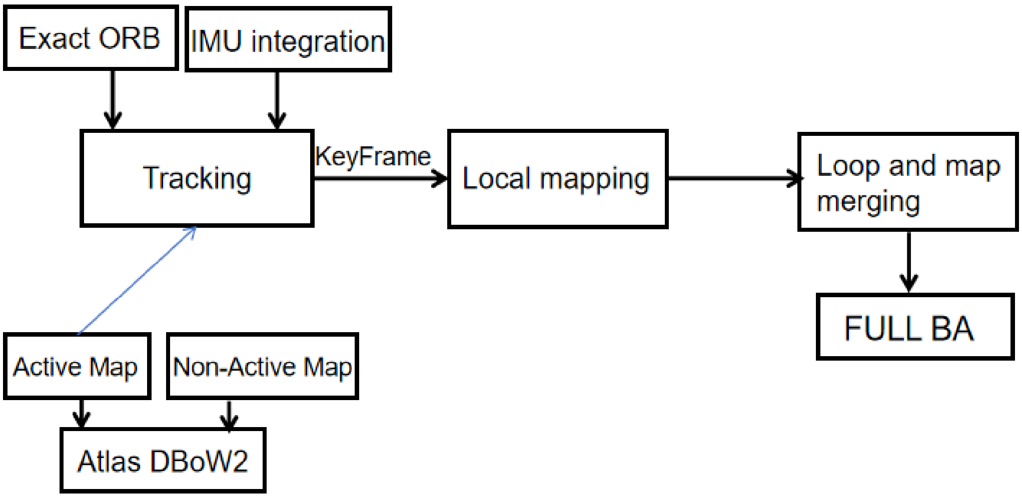

Figure 2. ORB-SLAM3 Algorithm Framework. ORB-SLAM3, Oriented FAST and Rotated BRIEF-Simultaneous Localization and Mapping 3.

nals and with a high sampling rate, capture rapid motion and determine the carrier's position and orientation through double integration.

VINS leverages the complementary strengths of visual sensors, which provide detailed environmental information, and IMUs, which measure motion. VSLAM helps correct IMU drift and performs loop closure detection, while IMUs assist VSLAM in addressing localization challenges in scenes with sparse textures or rapid motion, thereby enhancing the accuracy of visual-inertial systems in diverse settings [3]. VINS has broad applications in fields such as autonomous driving, drone navigation, augmented reality, and robotics. With advances in hardware and algorithm optimization, VINS continues to evolve, with ongoing research focusing on improving localization accuracy, enhancing map quality, and achieving more effective data fusion, particularly in dynamic environments. The following sections will elaborate on the current state of VINS development.

Key Technologies

Visual Sensors

With the advancement of computer technology, visual sensors have seen significant improvements in resolution, pixel density, and focus. Based on their operational mode, they can be categorized into monocular cameras, stereo cameras, and RGB-D cameras [4]. Monocular cameras are known for their simple design, low cost, and high computational efficiency, but they lack depth perception. Stereo cameras, on the other hand, derive depth information from the baseline distance between two monocular cameras, enabling depth estimation in both indoor and outdoor settings. However, stereo cameras require more complex configuration and calibration, as well as significant computational resources. RGB-D cameras, which became popular around 2010, capture both color and depth information directly, making them suitable for various routine visual tasks [5]. **Figure 1** shows the timeline of classic VINS algorithms based on filtering and optimization.

Visual odometry

Nister introduced the concept of visual odometry (VO), which involves analyzing image sequences to estimate the pose (position and orientation) of a moving robot in real-time using computer vision techniques [6]. This method overcomes the limitations of traditional odometry methods, providing more accurate localization. There are three main types of algorithms for processing images captured by visual sensors: feature-based methods, optical flow methods, and direct methods. According to the residual model definition, feature-based and optical flow methods are collectively referred to as indirect methods. **Figure 2** is ORB-SLAM3 Algorithm's Framework.

Feature-based methods extract sparse features from images through detection or general-purpose operations. These features are tracked between consecutive frames by matching feature points and adding descriptors to enhance robustness. Prominent feature-based methods include MonoSLAM, proposed by Davison et al. in 2007, which was the first real-time monocular visual SLAM system; ORB-SLAM, ORB-SLAM2, and ORB-SLAM3 [7-10]. Features in images may have specific structures, such as corners, edges, or block objects. It is generally easier to match corners or edges between images than block objects. To improve robustness, line feature extraction has gained attention in recent research. In 2016 , Meier et al. proposed an algorithm using Bezier curves as static landmark primitives instead of point features, combining curve landmarks for mapping to generate continuous curve-based maps [11]. Experiments demonstrated that line features improved computational efficiency compared to point features.

Gomez-Ojeda et al. proposed the PL-SLAM, which combines point and line features for robust performance, especially in scenarios where point features are sparse or unevenly distributed [12]. PL-SLAM does not assume line endpoint positions, enabling its tracking frontend to handle partially occluded lines. He et al. introduced the PL-VIO monocular visual-inertial odometry system, which tightly couples point and line features [13]. It optimizes system state by minimizing IMU preintegration constraints and point and line reprojection errors in a sliding window. Fu et al. improved the LSD algorithm from OpenCV by using Plucker coordinates to represent spatial lines as infinite lines and modeling line reprojection residuals based on point-to-line distances [14-16]. Integrated with VINS-Mono, the improved LSD algorithm significantly increased speed and introduced the first complete VINS framework for point-line feature fusion [17]. Xu et al. developed the EPLF-VINS algorithm, which enhances traditional line feature detection by fusing short lines, ensuring uniform line feature distribution, and using adaptive threshold extraction methods [18]. Teng et al. presented a tightly coupled stereo visual-inertial SLAM system with point-line features, improving line feature detection by introducing new criteria for curvature detection and minimum line segment length [19]. Zheng et al. proposed the LRPL-VIO visual-inertial odometry algorithm, which assumes invariant luminance values between endpoints and midpoints of consecutive frames for fast line matching and to address localization challenges in complex environments [20].

Direct methods minimize photometric errors, while indirect methods minimize geometric displacements. Direct methods, with smaller attraction regions, require good initial guesses, whereas indirect methods consume additional computational resources for feature extraction and matching. Indirect methods are more widely used in practical applications due to their robustness and maturity. However, direct methods, based on photometric consistency, require good initial guesses and high frame rates for accurate image alignment and are more suitable for dense mapping, showing potential in textureless scenes. SVO, introduced by C. Forster et al. in 2014, is a semi-direct monocular VO algorithm [21]. It uses semi-sparse direct methods for pixel-level matching and features for stable tracking and employs image pyramids for scale-space feature extraction and tracking. Engel et al. introduced direct sparse odometry, which minimizes photometric errors while jointly optimizing all model parameters, making it one of the few systems to compute VO purely through direct methods [22]. Zhao et al. enhanced monocular direct VO by developing deep direct sparse odometry, a new architecture that improves scale consistency in the trajectory network TrajNet using geometric constraints [23].

Direct methods minimize photometric errors, while indirect methods minimize geometric displacements. Direct methods require good initial guesses due to their small attraction regions, while indirect methods require additional computational resources for feature extraction and matching. Indirect methods are more robust and widely used, but direct methods are more adaptable for dense mapping and show promise in textureless environments [17].

IMU preintegration

IMU preintegration is a crucial step in VINS for estimating a device's displacement, velocity, and attitude over time by integrating linear acceleration and angular velocity measurements from the IMU. This process addresses challenges such as repeated integration of IMU measurements, which occur at a higher frequency compared to visual sensor data, and error accumulation during prolonged IMU motion [24-27]. IMU preintegration typically involves preprocessing raw IMU data to remove noise, performing coordinate transformations, estimating attitude, velocity, and position using filters like the Kalman or Extended Kalman filters, and predicting motion using physical models. The predicted values are then integrated using a mathematical model to reduce error accumulation.

IMU preintegration is essential for improving pose estimation accuracy in visual-inertial SLAM systems. By accounting for the correlation between IMU measurements and errors, it provides more accurate pose information, thus enhancing the overall performance of SLAM systems.

State Estimation Optimization

State estimation in VINS is an optimization problem where camera pose variations are computed through iterative image alignment [3]. Solutions are primarily divided into filter-based methods and optimization-based methods. Below is an introduction to these two types of approaches.

Filter-based methods

Filter-based methods are a class of state estimation techniques that utilize sensor data to predict and update a system's state. Based on Bayesian filtering theory, these methods combine sensor data with the system's dynamic model to perform recursive state estimation. Common filter-based approaches include the Kalman filter, extended kalman filter (EKF), unscented kalman filter, and particle filter. These methods typically estimate the state by recursively predicting and updating the probability distribution of the system's state.

Previously, filter-based methods were the primary approach for the back-end of V-SLAM, with the core objective of iteratively updating the state using Bayesian filtering. In VINS, EKF and unscented kalman filter, extensions of the Kalman filter, are commonly used to handle nonlinear systems. EKF achieves state estimation by linearizing the state transition and observation models, followed by applying standard Kalman filter methods. Although simple to implement, EKF may introduce errors when linearizing nonlinear functions and requires computing Jacobian matrices, reducing convergence and accuracy in highly nonlinear systems.

On the other hand, unscented kalman filter offers higher accuracy and stability, as it does not require Jacobian matrices and avoids linearization errors, making it more efficient in highly nonlinear systems. In 2007, Mourikis et al. proposed the Multi-State Constraint Kalman Filter, an extension of EKF, which improves robustness and accuracy by introducing multiple state constraints [1]. Multi-State Constraint Kalman Filter handles multiple state variables and observation constraints, fusing them through nonlinear optimization, making it suitable for complex VINS applications.

In 2015, Bloesch et al. introduced the ROVIO algorithm, a VIO based on the direct EKF method [28]. It enhances the state estimation of visual and inertial measurements by incorporating outlier suppression and preintegration techniques, achieving strong performance in dynamic environments. SR-ISWF is an extension of Multi-State Constraint Kalman Filter that uses a single-precision representation in square root form, avoiding numerical instability [29]. It employs an inverse filter for iterative linearization, similar to optimization-based algorithms, batch graph optimization, or bundle adjustment techniques, to optimize all measurement data for the best state estimation.

Optimization-based methods

Optimization-based methods adjust camera poses or 3D point clouds by minimizing reprojection error, photometric error, or other cost functions. The primary goal is to improve pose estimation accuracy, with reprojection error being a common metric. Reprojection error measures the Euclidean distance between predicted and observed image feature point positions. Optimization techniques like the Newton-Gauss method or the Levenberg-Marquardt algorithm are often employed. Levenberg-Marquardt, an algorithm for nonlinear least squares problems, combines the strengths of gradient descent and Newton's method, offering good convergence and robustness.

Another widely used optimization approach is pose graph optimization, which minimizes the error between consecutive poses and constraints from loop closures or sensor measurements. This method is frequently applied in real-time visual SLAM systems, where new sensor data is continuously processed, and incremental optimization is required.

In 2016, Engel et al. introduced direct sparse odometry, a direct VO method that estimates camera motion by minimizing brightness error between consecutive frames, without relying on feature extraction and matching [22]. It employs sliding window optimization, marginalizing out old camera poses and points that leave the field of view. In 2018, Qin et al. proposed VINS-mono, which features a robust initialization process [17]. By optimizing only a fixed number of recent IMU poses and velocities, VINS-mono reduces computational complexity. It also introduces a tightly coupled relocalization module that minimizes drift by establishing feature-level connections between loop closure candidates and the current frame, integrating them into the monocular VIO module.

In 2021, Campos et al. presented ORB-SLAM3, a tightly integrated visual-inertial SLAM system that builds upon its predecessors, ORB-SLAM and ORB-SLAM2 [10]. ORB-SLAM3 fully utilizes maximum a posteriori estimation during IMU initialization, resulting in significantly higher accuracy in both indoor and outdoor environments, with improvements up to tenfold over previous versions. Additionally, ORB-SLAM3 introduced the first multi-map system, leveraging short-term, medium-term, and long-term data associations to reuse information from high-parallax co-visible keyframes across all stages of the algorithm. This system performs loop closure and relocalization using the DBoW2 vocabulary library.

Chen et al. analyzed the application of the first estimate jacobian in VINS [30]. When implemented correctly in a nonlinear optimization framework, first estimate jacobian addresses information error caused by marginalization during linearization of fixed points, ensuring estimation consistency and improving system accuracy.

Overall, optimization-based methods provide a flexible and powerful framework for state estimation, allowing for the incorporation of complex sensor models and system dynamics. These methods are particularly well-suited for scenarios requiring accurate estimates and where computational resources are available for solving large-scale optimization problems.

Dynamic SLAM

Most visual SLAM algorithms assume a static scene, leading to poor performance in dynamic environments. Dynamic objects can introduce outliers that significantly affect the accuracy and stability of these algorithms. Dynamic SLAM methods are generally categorized into geometric-based and semantic-based approaches. Below is an overview of the principles and recent research achievements in these two categories.

Geometric-based dynamic SLAM

Geometric-based dynamic SLAM methods rely on matching key feature points extracted from consecutive image frames to provide geometric constraints for SLAM. These methods remove outliers from dynamic objects by analyzing motion models or applying geometric constraints. In 2017, Wang et al. proposed an end-to-end monocular VO framework based on deep recursive convolutional neural networks [31]. Recursive convolutional neural networks combine convolutional neural networks to extract spatial features from images with recurrent neural networks to capture temporal dynamics, enabling simultaneous feature extraction and sequential VO modeling. The model estimates new poses by inputting consecutive image frames.

In 2018, DeTone et al. introduced SuperPoint, a method unrelated to visual-inertial navigation but relevant for automatically detecting interest points in images and describing them using deep learning techniques [32]. SuperPoint can be used for keypoint detection and description in VINS. Bian et al. published a study on unsupervised learning of depth and camera motion from monocular videos [33]. They proposed geometric consistency losses for scale-consistent prediction and introduced self-discovery masks to handle moving objects and occlusions, addressing the issue of frame scale ambiguity in geometric image reconstruction. In 2022, Liu et al. presented lightweight feature tracking based on RGB-D sensor depth images and IMU inertial data. They divided images into grids, padded grid boundaries to prevent edge features from being ignored, and performed feature detection only on grids with insufficient feature matching [34]. By integrating geometric and depth information, they achieved dynamic feature recognition comparable to semantic segmentation.

Geometric-based methods offer good stability, providing reliable localization and mapping with high precision in various environments. They are computationally efficient and can achieve real-time localization. However, these methods perform poorly in feature-scarce environments and may cause instability in localization and mapping during scene changes. Additionally, their robustness in dynamic scenes is lower than that of semantic-based methods, as they lack semantic information.

Semantic-based dynamic SLAM

Semantic segmentation is a crucial task in computer vision that uses deep learning models to classify each pixel in an image into different semantic categories. This process generates a semantically rich segmentation map, where each pixel is labeled with a specific category, such as road, car, pedestrian, or building. Semantic segmentation is typically modeled using deep convolutional neural networks, with popular architectures including fully convolutional networks, U-Net, DeepLab, and SegNet. These networks perform convolution and deconvolution operations on images, progressively upsampling feature maps to generate segmentation maps of the same size as the original image, assigning each pixel to its most probable semantic category.

Sünderhauf et al. proposed a method to integrate semantic information into vision-based SLAM, utilizing feature-associated semantic data within a factor graph framework to improve outlier/inlier computation, based on a pre-trained deep learning network [35]. Niko et al. introduced a semantic mapping method by combining ORB-SLAM2 with a deep learning-based object detector, performing three-dimensional unsupervised segmentation using planar information from the object detector [36]. Bavle et al. extended this approach to develop a more complete SLAM framework, using landmark covariance to improve Mahalanobis distance calculations, optimizing the overall process [37].

Parkhiya et al. proposed using deep networks to build class-specific models for monocular object SLAM, estimating the relative 3D pose of semantic objects and integrating these with the robot's VO poses into a graph optimization framework [38]. Mccormac et al. introduced Fusion++, an online object-level SLAM system that constructs accurate 3D models of objects, using Mask- recursive convolutional neural network instance segmentation for initializing compact volumetric signed distance fields and incrementally improving them via depth fusion for tracking, relocalization, and loop detection, while optimizing pose graphs [39].

Some research combines geometric and semantic methods to enhance system understanding and environmental processing. Yan et al. proposed DGS-SLAM, which combines geometric and semantic information through a motion segmentation detection module based on a multi-residual model and semantic segmentation techniques, designed to identify potential moving targets in dynamic scenes [40]. Cheng et al. introduced SG-SLAM, a real-time semantic visualization system built on the ORB-SLAM2 framework, which adds two parallel threads for target detection and semantic segmentation [41]. These threads provide dynamic object information for feature rejection strategies and integrate 2D keyframe information and 3D point clouds into a semantic object database, generating an intuitive semantic metric map in the ROS system. Sun et al. proposed a method for classifying dynamic objects based on their motion states, using IMU data and geometric constraints to identify and verify temporarily static features [42]. Additionally, a new adaptive weight loss function based on BA dynamic factors is used in the backend to refine camera poses and spatial point positions.

Semantic methods accurately classify each pixel into specific semantic categories, providing high-quality information for subsequent image analysis and addressing issues with object detection bounding boxes. However, semantic methods heavily depend on neural network quality, making it difficult to balance speed and accuracy simultaneously. Combining geometric and semantic methods will be essential for enhancing the robustness of VINS in dynamic environments.

Conclusion

VINS remains a prominent research topic in fields like robot navigation and autonomous driving. This paper provides a review of the current state of visual-inertial systems, highlighting the continuous advancements in filtering/optimization-based methods and geometric/semantic-based methods in dynamic SLAM. While significant progress has been made over the past decade, several challenges remain:

Data Association: Many methods based on IMU and VO rely on prior assumptions, such as normal distribution and stationary noise, or the accuracy of sensors.

Sensor Fusion: Integrating additional sensors, such as LiDAR, can provide richer information for visual SLAM algorithms.

Deep Learning: Incorporating deep learning methods offers a learnable strategy for SLAM, enabling the use of past SLAM experiences to improve performance in unknown environments.

Author Contributions: Mingxia Wei contributed to the conception of the study and performed the discussion and wrote the manuscript. Qingyun Meng contributed to check the manuscript.

References

[1] Mourikis AI, Roumeliotis SI. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. Proceedings 2007 IEEE International Conference on Robotics and Automation 2007;3565-3572.

[2] He M, Zhu C, Huang Q, et al. A review of monocular visual odometry. Visual Comput 2019;36(5):1053-1065.

[3] Gui J, Gu D, Wang S, et al. A review of visual inertial odometry from filtering and optimisation perspectives. Adv Rob 2015;29(20):1289-1301.

[4] Huang G. Visual-Inertial Navigation: A Concise Review. 2019 International Conference on Robotics and Automation (ICRA) 2019;9572-9582.

[5] Kerl C, Sturm J, Cremers D. Dense visual SLAM for RGB-D cameras. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems 2013;2100-2106.

[6] Nister D, Naroditsky O, Bergen J. Visual odometry. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. 2004;I-I.

[7] Davison AJ, Reid ID, Molton ND, et al. MonoSLAM: real-time single camera SLAM. IEEE Trans Pattern Anal Mach Intell 2007;29(6):1052-1067.

[8] Mur-Artal R, Montiel JMM, Tardos JD. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Trans Rob 2015;31(5):1147-1163.

[9] Mur-Artal R, Tardos JD. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. IEEE Trans Rob 2017;33(5):1255-1262.

[10] Campos C, Elvira R, Rodriguez JJG, et al. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. IEEE Trans Rob 2021;37(6):1874-1890.

[11] Meier K, Chung SJ, Hutchinson S. Visual-inertial curve SLAM. 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2016;1238-1245.

[12] Gomez-Ojeda R, Moreno F-A, Zuniga-Noel D, et al. PL-SLAM: A Stereo SLAM System Through the Combination of Points and Line Segments. IEEE Trans Rob 2019;35(3):734-746.

[13] He Y, Zhao J, Guo Y, et al. PL-VIO: Tightly-Coupled Monocular Visual-Inertial Odometry Using Point and Line Features. Sensors (Basel). 2018:10;18(4):1159.

[14] Fu Q, Wang J, Yu H, et al. PL-VINS: Real-Time Monocular Visual-Inertial SLAM with Point and Line Features. arXiv:2009.07462 2022.

[15] Grompone von Gioi R, Jakubowicz J, Morel JM, et al. LSD: a fast line segment detector with a false detection control. IEEE Trans Pattern Anal Mach Intell 2010;32(4):722-732.

[16] Kaehler A, Bradski G. Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library. O'Reilly Media, Inc., 2016.

[17] Qin T, Li P, Shen S. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. IEEE Trans Rob 2018;34(4):1004-1020.

[18] Xu L, Yin H, Shi T, et al. EPLF-VINS: Real-Time Monocular Visual-Inertial SLAM With Efficient Point-Line Flow Features. IEEE Rob Autom Lett 2023;8(2):752-759.

[19] Teng Z, Han B, Cao J, et al. PLI-SLAM: A Tightly-Coupled Stereo Visual-Inertial SLAM System with Point and Line Features. Remote Sens 2023;15(19):4678.

[20] Zheng F, Zhou L, Lin W, et al. LRPL-VIO: A Lightweight and Robust Visual-Inertial Odometry with Point and Line Features. Sensors 2024;24(4):1322.

[21] Forster C, Pizzoli M, Scaramuzza D. SVO: Fast semi-direct monocular visual odometry. 2014 IEEE International Conference on Robotics and Automation (ICRA) 2014;15-22.

[22] Engel J, Koltun V, Cremers D. Direct Sparse Odometry. IEEE Trans Pattern Anal Mach Intell 2018;40(3):611-625.

[23] Zhao C, Tang Y, Sun Q, et al. Deep Direct Visual Odometry. IEEE trans Intell Transp Syst 2022;23(7):7733-7742.

[24] Liu Z, Shi DX, Yang SW, et al. Review of visual-inertial navigation system initialization method. J Natl Univ Def Technol, China 2023;45(02):15-26.

[25] Lupton T, Sukkarieh S. Visual-Inertial-Aided Navigation for High-Dynamic Motion in Built Environments Without Initial Conditions. IEEE Trans Rob 2012;28(1):61-76.

[26] Christian F, Luca C, Frank D, et al. IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation. Robotics: Sci Syst 2015.

[27] Forster C, Carlone L, Dellaert F, et al. On-Manifold Preintegration for Real-Time Visual--Inertial Odometry. IEEE Trans Rob 2017;33(1):1-21.

[28] Bloesch M, Omari S, Hutter M, et al. Robust visual inertial odometry using a direct EKF-based approach. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2015;298-304.

[29] Paul MK, Wu K, Hesch JA, et al. A comparative

analysis of tightly-coupled monocular, binocular, and stereo VINS. 2017 IEEE International Conference on Robotics and Automation (ICRA) 2017;165-172.
[30] Chen C, Geneva P, Peng Y, et al. Optimization-Based VINS: Consistency, Marginalization, and FEJ. 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2023;1517-1524.
[31] Wang S, Clark R, Wen H, et al. DeepVO: Towards end-to-end visual odometry with deep Recurrent Convolutional Neural Networks. 2017 IEEE International Conference on Robotics and Automation (ICRA) 2017;2043-2050.
[32] DeTone D, Tomasz Malisiewicz, Andrew Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. arXiv:1712.07629 2018.
[33] Bian JW, Li Z, Wang N, et al. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In: editors. Proceedings of the 33rd International Conference on Neural Information Processing Systems. Curran Associates Inc., 2019. p. Article 4.
[34] Liu J, Li X, Liu Y, et al. RGB-D Inertial Odometry for a Resource-Restricted Robot in Dynamic Environments. IEEE Rob Autom Lett 2022;7(4):9573-9580.
[35] Murali V, Chiu HP, Samarasekera S, et al. Utilizing semantic visual landmarks for precise vehicle navigation. 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC) 2017;1-8.
[36] Sünderhauf N, Pham TT, Latif Y, et al. Meaningful maps with object-oriented semantic mapping. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2017;5079-5085.
[37] Bavle H, Puente PDL, How JP, et al. VPS-SLAM: Visual Planar Semantic SLAM for Aerial Robotic Systems. IEEE Access 2020;8:60704-60718.
[38] Parkhiya P, Khawad R, Murthy JK, et al. Constructing Category-Specific Models for Monocular Object-SLAM. 2018 IEEE International Conference on Robotics and Automation (ICRA) 2018;4517-4524.
[39] Mccormac J, Clark R, Bloesch M, et al. Fusion++: Volumetric Object-Level SLAM. 2018 International Conference on 3D Vision (3DV) 2018;32-41.
[40] Yan L, Hu X, Zhao L, et al. DGS-SLAM: A Fast and Robust RGBD SLAM in Dynamic Environments Combined by Geometric and Semantic Information. Remote Sens 2022;14(3):795.
[41] Cheng S, Sun C, Zhang S, et al. SG-SLAM: A Real-Time RGB-D Visual SLAM Toward Dynamic Scenes With Semantic and Geometric Information. IEEE Trans Instrum Meas 2023;72:1-12.
[42] Sun Y, Wang Q, Yan C, et al. D-VINS: Dynamic Adaptive Visual–Inertial SLAM with IMU Prior and Semantic Constraints in Dynamic Scenes. Remote Sens 2023;15:3881.