Neurocomputing 590 (2024) 127760

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A comprehensive overview of core modules in visual SLAM framework

Dupeng Cai [a,1], Ruoqing Li [b,1], Zhuhua Hu [a,*], Junlin Lu [a], Shijiang Li [a], Yaochi Zhao [b]

[a] School of Information and Communication Engineering, Hainan University, Haikou 570228, China
[b] School of Cyberspace Security (School of Cryptology), Hainan University, Haikou 570228, China

ARTICLE INFO

Keywords:
Visual SLAM
SLAM framework
Localization and mapping
Navigation
Environmental sensing

ABSTRACT

Visual Simultaneous Localization and Mapping (VSLAM) technology has become a key technology in autonomous driving and robot navigation. Relying on camera sensors, VSLAM can provide a richer and more precise perception means, and its advancement has accelerated in recent years. However, current review studies are often limited to in-depth analysis of a specific module and lack a comprehensive review of the entire VSLAM framework. The VSLAM system consists of five core components: (1) The camera sensor module is responsible for capturing visual information about the surrounding environment. (2) The front-end module uses image data to roughly estimate the camera's position and orientation. (3) The back-end module optimizes and processes the pose information estimated by the front-end. (4) The loop detection module is used to correct accumulated errors in the system. (5) The mapping module is responsible for generating environmental maps. This review provides a systematic and comprehensive analysis of the SLAM framework by taking the core components of VSLAM as the entry point. Deep learning brings new development opportunities for VSLAM, but it still needs to solve the problems of data dependence, cost and real-time in practical application. We deeply explore the challenges of combining VSLAM with deep learning and feasible solutions. This review provides a valuable reference for the development of VSLAM. This will help push VSLAM technology to become smarter and more efficient. Thus, it can better meet the needs of future intelligent autonomous systems in multiple fields.

1. Introduction

The rapid development of artificial intelligence has promoted breakthroughs in areas such as autonomous driving, robot navigation and augmented reality [1–6]. Behind these achievements, localization and mapping techniques in unfamiliar environments have played a crucial role. In the past, Autonomous driving technology used to rely on satellite devices such as GPS, BDS, Galileo, and GLONASS, or real-time kinematic (RTK) to achieve precise localization and ensure safe navigation. However, these methods are often limited by high costs and factors such as signal interference. In contrast, Simultaneous Localization and Mapping (SLAM) technology [7] has gradually become an effective means to solve the localization and mapping problems with its advantages. It can analyze the sensor data of unknown environment in real time, and complete the localization and mapping at the same time to provide the basic navigation and perception functions for intelligent systems such as autonomous driving and robot navigation [8–10]. Among them, Lidar SLAM based on Lidar sensors, with its robustness to the environment and system autonomy advantages, started earlier and has long been considered the preferred solution for mobile robots and other fields [11–13]. However, Lidar's high cost and complex development cycle limit its popularity. In contrast, with the advancement of computer vision technology, Visual Simultaneous Localization and Mapping (VSLAM) technology has emerged. Relying on camera sensors, VSLAM provides a richer and more accurate perspective for environmental perception. More importantly, in situations where sensor information is scarce, VSLAM can still show excellent performance with its precise recognition of image structure and features. [14]. As shown in Fig. 1, the technical framework of VSLAM is mainly composed of the following core modules [15]: The camera sensor module is responsible for capturing visual information about the surrounding environment. The front-end module roughly estimates the relative position and direction of the camera by analyzing the image data of adjacent frames. The back-end module optimizes the pose of the front-end to ensure the construction of globally consistent motion trajectory and environment map. The loop module is able to identify and confirm that the system has returned to the visited area, thus correcting the accumulated errors in the SLAM system. The mapping module is responsible for generating

* Corresponding author.
E-mail addresses: caidp3060@163.com (D. Cai), rqli@hainanu.edu.cn (R. Li), eagler_hu@hainanu.edu.cn (Z. Hu), lujunlin762@gmail.com (J. Lu), 15508065186@163.com (S. Li), zhyc@hainanu.edu.cn (Y. Zhao).
[1] Contributed equally to this work.

https://doi.org/10.1016/j.neucom.2024.127760
Received 7 February 2024; Received in revised form 2 April 2024; Accepted 22 April 2024
Available online 25 April 2024

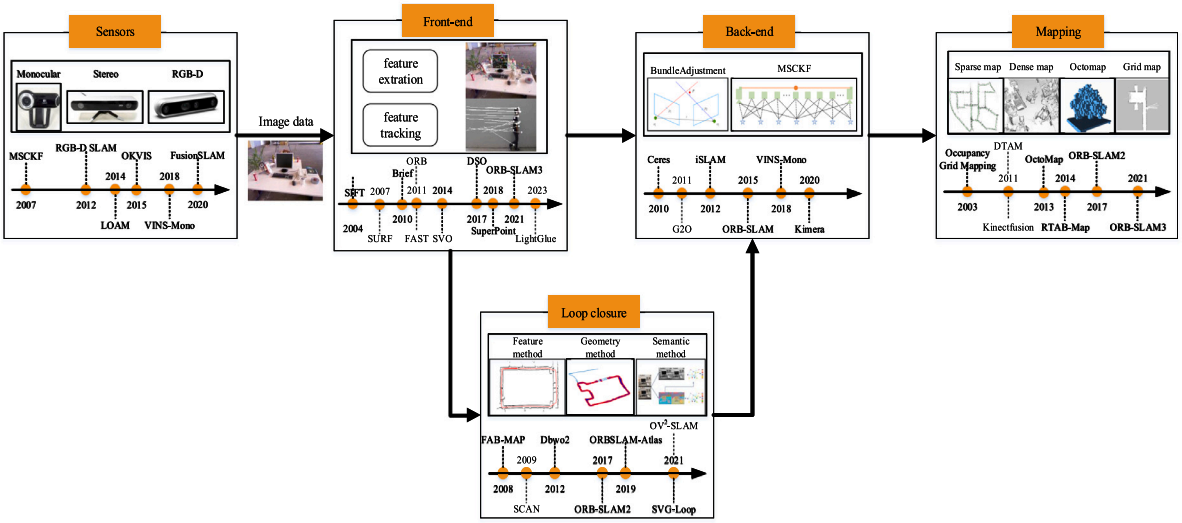


Fig. 1. The classical technical framework of VSLAM.

environmental maps based on collected data, providing key information for navigation, obstacle avoidance, and other tasks.

The organization of this paper is illustrated in Fig. 1. In the data acquisition module, we describe cameras such as monocular camera, stereo camera, RGB-D camera and event camera. We deeply analyze the efficient capture ability of VSLAM to the surrounding environment information, and emphasize the advantages of multi-source information fusion, and list the common sensor combinations and their respective advantages. In the front-end module, we not only introduce its key role in VSLAM system, but also emphasize the rapid development of this module in recent years. In addition, we also make a clear comparison of three common front-end methods (feature method, direct method and semi-direct method), so that readers can fully understand the characteristics and application scenarios of these methods. At the same time, we also point out the complex problems faced by deep learning in practical applications, such as hardware cost, importance evaluation of feature matching, and advantages and limitations in specific scenarios. In the back-end module, we systematically divide the current mainstream back-end optimization methods, including filtering, bundle adjustment (BA) and geometry optimization methods. In addition to these traditional methods, we also discussed some new back-end optimization method, makes a clear assessment and the advantages and disadvantages. We have detailed the main components of the mathematical model of these methods so that readers can fully understand the principle and implementation details. In the loop detection module, we identify three common problems faced by the current loop detection module, and divide the loop detection methods into three categories according to the three principles of feature-based, geometric information based and semantic information based. For each class of loop methods, we detail some of the current leading methods and highlight their characteristics and advantages. In the mapping module, we have identified several main map forms based on the actual application scenarios of robots, including point cloud maps, occupied grid maps, octree maps, and semantic maps. We have provided a detailed introduction to each map form, including its mapping methods and specific roles in robot applications. It is worth noting that in complex tasks such as intelligent obstacle avoidance, target recognition, interaction, etc., semantic maps are more widely used compared to traditional visual maps, providing strong support for intelligent decision-making of robots.

Finally, we summarize the rapid development of VSLAM technology in recent years and the key role of deep learning in it. Meanwhile, we point out the complex problems faced by deep learning in practical applications such as data dependence, cost requirements and real-time requirements. To address these challenges, we propose a series of solutions, including using synthetic data to make up for data deficiency, choosing lightweight model structure to reduce computing resource consumption, optimizing hardware structure to improve computing efficiency, pruning model and optimizing inference engine to improve real-time performance and other measures to promote the development of SLAM technology in practical applications.

2. Data acquisition module

For SLAM systems, obtaining data from sensors is the core and key link of the entire system [16]. It not only lays the foundation for localization and mapping processes, but also helps the system analyze and understand unknown environments in real-time and accurately. In VSLAM, camera sensor plays an important role. Monocular cameras, stereo cameras, RGB-D cameras and event cameras each have their own characteristics and play different roles in VSLAM [17–19]. Monocular cameras are widely used due to their simple structure and low cost, but they are limited in depth acquisition. Stereo cameras through two lenses to simulate human eyes, able to obtain more accurate depth information. RGB-D cameras can capture more three-dimensional information by integrating depth sensors. Event cameras are particularly suitable for dynamic and high contrast environments due to their high sensitivity to brightness changes. As shown in Table 1, the detailed environmental information captured by these various camera sensors provides a rich and precise perspective for navigation, obstacle avoidance, and other intelligent control tasks. The comprehensive utilization of this information promotes the continuous development and application of VSLAM technology in the fields of automatic driving and robot navigation.

However, a single vision sensor has obvious limitations in the application of VSLAM. Firstly, visual sensors are susceptible to the influence of lighting conditions, for example, in environments that are too bright or too dim, visual sensors may not be able to accurately extract effective feature points. Second, the visual sensor is highly dependent on the texture and color of the object in the scene, and the surface with no texture or a single color may lead to difficulty in extracting feature points. In addition, rapid motion and blur may also interfere with the performance of visual sensors, resulting in insufficient interpretation of dynamic environments. Moreover, phenomena such as occlusion and reflection may also render visual sensors ineffective in specific scenarios. These factors collectively lead to a decrease in accuracy and robustness of a single visual sensor in complex and ever-changing environments, limiting its feasibility in a wider range of application scenarios.

Table 1
Common camera sensors.

Types	Descriptions	Advantages	Disadvantages
	Cameras with only one visual sensor to acquire scene information by capturing a continuous sequence of images	Lightweight, inexpensive and easy to install	Cannot measure depth directly, 3D reconstruction is more challenging
	Consisting of two cameras arranged in parallel, depth information is calculated by comparing parallaxes	Accurate 3D modeling and object distance measurement	High calibration and alignment requirements and higher costs
	Capable of capturing conventional color information and depth information for each pixel, usually achieved through infrared light, structured light, and other technologies	Provide rich 3D reconstruction and object recognition information to improve accuracy and stability	Higher price, may be affected by ambient light and material
	Records pixel-level luminance change events when there is a significant change in luminance	Highly robust to light variations and motion blur, very high time resolution	Complex image interpretation and relatively new algorithm development

2.1. IMU and camera

Given the limitations of a single visual sensor in lighting changes, texture loss, and dynamic environments, multi-source fusion strategies have gradually become an important solution. Among them, Visual Inertial Odometry (VIO) has shown great potential in many application scenarios by integrating visual sensors and Inertial Measurement Units (IMUs). IMUs can provide high-frequency localization and orientation data, thereby compensating for the shortcomings of visual sensors in situations involving fast motion or poor lighting conditions. In turn, visual sensors can help mitigate the cumulative drift of IMUs. In the history of VIO research, the significance of precise state estimation was underscored by Titterton et al. [20] as early as 2005. Subsequent research has led to the development of two primary mainstream approaches: tightly coupled and loosely coupled [21]. Tightly-coupled VIO systems, such as the MSCKF method proposed by Mourikis et al. [22], integrate camera image data and IMU measurements within a unified optimization framework. This approach ensures high accuracy and robustness but comes at the cost of relatively high computational complexity. Similar methods include OKVIS [23], VINS-MONO [24], and ICE-BA [25]. Conversely, loosely coupled VIO systems adopt a more flexible approach by separately processing visual and IMU data and then selectively fusing or optimizing the estimation results of both, as demonstrated in the work of Konolige et al. [26], Tardif et al. [27], and Weiss et al. [28]. While this approach reduces computational complexity and enhances system flexibility, it may compromise on accuracy. In addition, factors such as increased complexity of the environment (such as the presence of dense obstacles) or structural degradation (such as lack of obvious visual features or textures) may interfere with the functionality of visual sensors, and even with the assistance of IMUs, the performance of the system may be affected.

In the process of combining the camera with the IMU, there are also some shortcomings. Firstly, the accelerometer data measured by the IMU may be affected by drift in the direction of gravity, which can lead to cumulative errors in pose estimation, especially over long periods of use. Secondly, there may be differences in the data acquisition frequency and clock between the camera and IMU, resulting in asynchronous data, which can affect the accuracy of the fusion algorithm. Finally, the accelerometer and gyroscope data measured by IMU usually contain high-frequency noise, which requires filtering and integration processing to obtain accurate motion estimation, which increases the complexity of the system.

2.2. Lidar, IMU and camera

The integration of Lidar, IMU and vision sensors provides a robust and precise solution for 3D environmental measurements and fast robot movement. Integrating the precise distance measurement of Lidar with its robustness to motion, as well as the high resolution of visual sensors and the excellent motion tracking ability of IMUs, helps the system maintain high adaptability when dealing with complex or structurally degraded environments, while ensuring high accuracy and smoothness of trajectories. For instance, approaches like VLOAM [29] and Pronto [30], which loosely integrate visual inertial odometry (VIO) with Lidar sensors, have achieved satisfactory results in 3D reconstruction. Nevertheless, these techniques still have room for improvement in real-time environmental mapping. In contrast, several tightly coupled approaches that support both joint optimization of VIO and Lidar inertial odometry (LIO), as well as their individual optimization [31–35], have further elevated accuracy and robustness. Among these, VIL-SLAM [31] incorporates visual feature loop detection and Lidar alignment methods to optimize global localization. This approach effectively leverages the high-resolution capabilities of visual sensors and the robust distance measurement capabilities of Lidar, providing strong support for navigation and recognition in complex environments. More recently, Liu et al. [36] have advanced this field by introducing the first Lidar-inertial vision fusion simultaneous localization and mapping (SLAM) system with formidable position recognition capabilities, robustness, and efficiency. By combining two subsystems, the Visual inertial odometer (VIO) and the Lidar inertial odometer (LIO), they optimize the 2D–3D photometric error and achieve robust 2D–3D alignment using a global optimization and closed-loop detection network. This innovative approach not only enhances the accuracy and efficiency of localization and mapping but also bolsters the overall robustness of the SLAM system, holding significant implications for future research in robot navigation and environmental perception.

The integration of Lidar, IMU and vision sensors can improve the performance and robustness of environmental awareness systems, but there are some shortcomings. Firstly, complex algorithms are needed to handle the fusion of data from different sensors, which will increase the complexity of the system. Secondly, the data acquisition frequency and clock of different sensors may be different, requiring accurate time synchronization and data alignment. Finally, integrating multiple sensors increases system cost and power consumption.

2.3. Lidar and camera

The combination of Lidar, IMU and vision sensors provides a powerful 3D environmental measurement and localization solution that accurately captures the details of complex environments and supports accurate motion estimation. However, the challenges in hardware configuration, computational complexity, system calibration, and cost of this integration solution limit its universality in certain application scenarios. In contrast, the fusion of Lidar and vision sensors, which focuses on the robust range measurement of Lidar and the high-resolution characteristics of vision sensors, provides a more economical and flexible solution for specific scenarios [37–42]. Recently, Zhang et al. [43] further promoted the development of this field and proposed the first LiDAR-Camera Panoptic Segmentation network (LCPS), which mainly realizes feature fusion through three stages. Firstly, a pixel alignment

module is used to solve the coordinate misalignment caused by different sensor frequencies; Then, the mapping relationship between points and pixels is extended through the region alignment module, while maintaining semantic consistency; Finally, integrate the entire scene information through the feature propagation module. This method embodies an innovative multimodal fusion strategy, providing a new perspective for the efficient integration of radar and visual information.

The combination of Lidar and camera can make up for the limitations of each other alone, but there are also some shortcomings. Firstly, the Lidar provides distance information, while the camera provides color and texture information, so there may be mismatches and inconsistencies in the data fusion of the two. Second, the camera may be affected under insufficient lighting or complex background conditions. Therefore, in some circumstances, they may not be able to make full use of their strengths.

2.4. Wheel and camera

Furthermore, the wheel speedometer sensor has a unique advantage in providing high-frequency measurement of the body's state, especially in special motion scenarios where it does not cause degradation issues. By fusing the information of the wheel speedometer, as proved by Wu et al. [44], the problem of unmeasurable scale can be avoided, thus improving the localization accuracy. SLAM technology, which combines vision, IMU and wheel speedometer, has also achieved desirable results in some applications [45–47]. However, in a small range of indoor and outdoor flat motion scenes, only the combination of vision and wheel speed meter may be a better choice. On the one hand, in these specific cases, the introduction of IMU may bring more noise and error, and have a limited effect on accuracy improvement. On the other hand, the fusion of IMU will also increase the state dimension and initialization difficulty of the system, thus increasing the burden of calculation and calibration. The combination of vision and wheel speed meter itself can effectively capture key information in the scene, with good accuracy and robustness. Several studies [48,49] have shown that localization and mapping techniques that fuse vision sensors with the wheel speedometer can enhance operational efficiency while improving localization accuracy. Therefore, in certain application scenarios, such an approach not only meets accuracy requirements but may also offer a more attractive solution in terms of resource utilization and cost-effectiveness.

The combination of the wheel speed meter and the camera enhances the performance and robustness of the environment awareness system. However, the wheel speed meter is easily affected by factors such as tire slip and ground friction during driving, which leads to measurement error and drift. These factors may affect the accuracy and stability of the data fusion algorithm.

2.5. Other sensor combinations

With the continuous advancement of VSLAM technology, various sensor types are integrated into the overall solution to bolster system robustness and adaptability. Satellite navigation systems, such as the Global Navigation Satellite System GNSS [50–52], offer global coverage and precise absolute localization in outdoor scenarios. When combined with visual localization, these systems help overcome the limitations of visual approaches in extensive and complex environments. However, satellite navigation systems can be affected by weakened signals and weather. WiFi localization [53–55] estimates location by measuring the signal strength of surrounding WiFi hotspots, especially in indoor environments, providing a reliable and low-cost localization supplement. However, it is affected by factors such as base station setting and signal propagation. Optical flow sensors estimate object motion by analyzing pixel changes between continuous images, and can provide accurate relative motion information in high-speed and complex dynamic environments [56–59]. But it is affected by environmental factors such as changes in light and loss of texture. In different application scenarios and environmental conditions, the combination of these sensors ensures that the VSLAM system can achieve more accurate and robust localization and mapping to meet a wide range of practical needs.

As shown in Table 2, by integrating data from different types of sensors (such as satellites, IMUs, Lidar, etc.), VSLAM can have a more comprehensive understanding of the environment. Thus, the advantages and limitations of each sensor can be complemented, and the accuracy and reliability of localization and mapping can be improved.

3. Front-end modules

The front-end module is a key component in the VSLAM system and is responsible for processing successive frames to roughly estimate the relative position and orientation of the camera. Front-end methods mainly include feature method, direct method and semi-direct method which combines the advantages of both, as shown in Fig. 2 and Table 4.

3.1. Feature methods

The feature methods have always been one of the main techniques in early VSLAM systems, estimating camera motion by extracting key points or features from images and matching them between consecutive frames. As early as 1988, Harris corner detection [60] detected corner points through the gray change of the window region, which laid the foundation for subsequent algorithms. It has rotation invariance and certain robustness, but lacks scale invariance. Subsequently, SIFT [61] fills this gap by providing scale and rotation invariance through extreme point detection in scale space, but the computation is relatively slow. SURF [62] has accelerated the calculation speed based on SIFT, using integrated images and box filters, but sacrificing some accuracy. FAST [63] rapidly detects corner points through brightness differences, which is more suitable for real-time applications, but has shortcomings in rotation and scale invariance. FAST combined with the BRIEF descriptor [64] can be both real-time and robust [65]. Its feature matching effect is relatively ideal and it has stable rotation invariance. Specifically, the algorithm first builds an image pyramid, FAST quickly finds corners by detecting brightness differences in the image, and BRIEF uses grayscale comparisons of a set of random position pairs to quickly create binary descriptors. However, the feature method is not ideal in scenes where the features are not obvious or sparse. Common feature extraction algorithms are shown in Table 3.

3.2. Direct methods

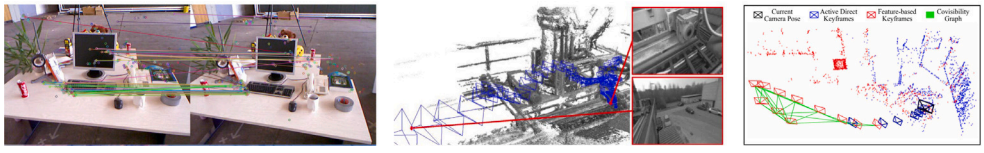
Unlike the feature method, the direct method does not rely on feature extraction and matching, but directly uses the pixel intensity of the image to estimate camera motion. The direct method was first proposed by Strasdat et al. [72], who proposed a direct method SLAM system based on grayscale image brightness residual. In recent years, the direct method has received widespread attention and application in the field of VSLAM. For example, DSO (Direct Sparse Odometry) [73] can directly optimize camera pose by minimizing photometric error, while LSD-SLAM (Large-Scale Direct Monocular SLAM) [74] performs pose estimation by direct image alignment. And reconstruct a semi-dense depth map of the 3D environment in real time. These direct methods can clearly detect scale drift and perform well on challenging sequences of scene scale changes, but may also lose tracking in complex texture environments. Another classic algorithm, DTAM (Dense Tracking and Mapping) [75], estimates 6DOF motion through dense pixel tracking and generates surface patches for deep mapping, achieving excellent tracking under fast motion. However, due to its reliance on dense pixel analysis, it has a high demand for computing resources and may be sensitive to scenes with complex lighting and textures.

Table 2
Common sensor combinations.

Sensor combinations	Characteristics	Applications
Camera (single vision)	Low cost, rich visual information that captures the structure and appearance of a scene	[17–19]
IMU and camera (VIO)	Enhanced localization accuracy to counteract problems caused by rapid motion or light changes and reduce cumulative drift	[20,22–28]
Lidar and camera	Provides greater robustness and captures accurate distance information for harsh weather and complex environments	[37–42]
Lidar, IMU and camera	Combines the benefits of multiple sensors for improved accuracy and robustness across a wide range of environments and motion conditions	[29–36]
Wheel and camera	Provides good performance in low-speed and structured environments, aiding map creation and route planning	[45–49]
Other sensor combinations	Can be flexibly combined according to specific applications and needs to achieve the best balance of performance and cost-effectiveness	[50–59]

Table 3
Comparison of commonly used feature extraction algorithms.

Method	Time	Type	Speed	Rotation invariance	Scale invariance	Illumination invariance	Anti-invariance
Canny [66]	1986	Edge	High	Yes	Yes	No	Strong
Harris [60]	1988	Corner	Low	Yes	No	Yes	Week
Shi-Tomasi [67]	1994	Corner	Middle	Yes	No	Yes	Week
SIFT [61]	2004	Point	Low	Yes	Yes	Yes	Strong
HOG [68]	2005	Region	Middle	Yes	Yes	No	Stronger
SURF [62]	2008	Point	Middle	Yes	Yes	No	Week
LSD [69]	2008	Line	Middle	Yes	Yes	Yes	Stronger
FAST [63]	2011	Point	High	No	Yes	No	Week
BRISK [70]	2011	Point	High	Yes	Yes	Yes	Stronger
AKAZE [71]	2012	Point	High	Yes	Yes	Yes	Stronger
ORB [65]	2012	Point	High	Yes	Yes	Yes	Stronger



(a) Feature based method (b) Direct method (c) Semi-direct method

Fig. 2. Comparison of VSLAM front-end technologies.

3.3. Semi-direct methods

Although direct methods can make better use of image information, especially in areas without significant features, they often involve significant computational complexity, which becomes their main challenge. On the contrary, the feature method performs better in scenes with obvious features, but its effect is limited in areas with less obvious features. As a novel visual odometry technology, semi direct method combines the advantages of direct method and feature method, providing a creative solution for VSLAM front-end. Semi-Direct Visual Odometry (SVO) [76] is a classic example of a semi-direct method. It uses direct method to accurately estimate camera rotation at image pixel level, and then uses feature point matching to accurately estimate translation. This design takes into account the speed advantage of the direct method and the robustness of the feature matching method, which is especially suitable for fast and dynamic scenarios. Furthermore, OKVIS [77] realizes the tightly coupled fusion of vision and inertial sensor through nonlinear optimization combined with bounded window processing of key frames. This method performs well in stereo and monocular vision systems, especially when compared with real conditions on the ground, showing good consistency and accuracy. However, the semi-direct method has some challenges. For example, SVO may have difficulty locating areas with sparse textures, while OKVIS's algorithm complexity is higher and may not be suitable for resource-constrained hardware platforms, especially in long running or feature-rich environments, which may suffer from reduced computational efficiency.

3.4. Recent front-end developments

With the continuous evolution of technology, a series of new methods have emerged in the front-end field of VSLAM in recent years, as shown in Table 5. CNN-SVO [84] significantly reduces the depth uncertainty of SVO initializing map points, resulting in more reliable feature correspondence and fast depth estimation, improving robustness and tracking accuracy. But this method performs poorly in transitional exposure scenarios. Qi et al. [93]. improved the quality of keyframe selection in dynamic scenes by employing an adaptive thresholding method to dynamically adjust the threshold based on the number of feature points in the observation model. This approach significantly enhances accuracy in highly dynamic scenes while maintaining real-time performance. But this method will add additional computing resources. Guo S et al. [96] solved the challenge of performance degradation of Lidar SLAM in complex and degraded scenarios. By combining feature extraction based on precise principal component analysis (PCA) and efficient two-stage matching strategy, they achieved more robust, accurate and globally consistent estimation performance. Obviously improve the ranging accuracy and the consistency of mapping. But its computational complexity is relatively high. This is only a small part of the rapid development of VSLAM technology. With the continuous development of artificial intelligence technology, the application of deep learning in VSLAM has gradually become a hot topic with its unique advantages. Tateno K et al. [97] earlier explored the use of convolutional neural networks (CNNs) to predict depth maps in order to achieve accurate and dense results in monocular reconstruction. They combined the depth maps predicted by CNN with the depth

Table 4
Comparison of VSLAM front-end technologies.

Methods	Depends on feature points	Posture estimation	Scene reconstruction	Tracking robustness	Applicable scenes
Feature methods [61,62,65]	Yes	Feature matching	Sparse/semi-dense maps	Dependent on feature quality	Feature-rich scenes
Direct methods [73–75]	No	Direct pixel intensity matching	Dense maps	May fail with complex textures	Full image information
Semi-direct methods [76,77]	Partially dependent	Combination of feature + pixel intensity	Sparse/semi-dense maps	Balance feature and pixel utilization	Combine advantages of feature and direct methods

Table 5
Comparison of the application of visual SLAM front-end technology.

Methods	Time	Input[a]	Indoor	Outdoor	Large scenes	Dynamic scenes	Real-time[b]
DTAM [75]	2011	1	√	√	–	–	1
LSD-SLAM [69]	2014	1	√	√	√	–	2
SVO [76]	2014	1	√	√	√	–	1
OKVIS [23]	2015	1	√	√	√	–	1
ORB-SLAM2 [78]	2017	1,2,3	√	√	√	–	1
SuperPoint [79]	2017	1	√	√	–	–	2
DSO [73]	2017	1	√	√	√	–	1
PL-SLAM [80]	2017	2	√	√	√	–	2
DS-SLAM [81]	2018	3	√	√	–	√	2
DynaSLAM [82]	2018	1,2,3	√	√	√	√	2
VINS-Mono [24]	2018	1	√	√	√	–	1
GCNv2 [83]	2019	3	√	–	–	–	2
CNN-SVO [84]	2019	1	√	√	√	–	2
SuperGlue [85]	2020	1	√	√	√	–	2
LOFTR [86]	2021	1	√	√	√	–	2
ORB-SLAM3 [87]	2021	1,2,3	√	√	√	–	1
Orb-slam2s [88]	2021	1	√	√	√	–	1
Fast ORB-SLAM [89]	2021	3	√	√	√	–	1
CBAM-SLAM [90]	2022	1,2,3	√	√	√	√	2
MISD-SLAM [91]	2022	1,2,3	√	√	√	√	2
AGAM-SLAM [92]	2023	1,2,3	√	√	√	√	2
ATY-SLAM [93]	2023	3	√	√	√	√	1
LightGlue [94]	2023	1	√	√	√	–	2
TOHF [95]	2024	1,2,3	√	√	√	–	1

[a] Input unit represents different sensors. 1 = monocular camera, 2 = stereo camera, 3 = RGB-D camera.
[b] Real-time unit represents different support, 1-CPU real-time, 2 = GPU real-time.

measurements of direct monocular SLAM, specifically making depth predictions at image locations where monocular SLAM is prone to fail, such as low-texture regions. Through this combination, one of the main limitations of monocular SLAM is overcome successfully, and semantic coherent scene reconstruction from a single perspective is realized.

In addition, many excellent results based on deep learning have followed. For example, DeepTIO [98] uses convolutional neural networks to jointly process images and inertial sensor data to provide more accurate and robust odometer estimates. The advantage of SuperGlue [85] in computer vision tasks that require feature matching and relative pose estimation has greatly promoted the development of VSLAM. It can automatically find the correspondence between two images given them, which can be used to create a 3D scene model or for camera localization, and perform feature matching in an end-to-end manner, while providing relative pose estimation. Furthermore, Yin J et al. [99] proposed a three-dimensional Lidar assisted monocular Visual SLAM (LAMV-SLAM) framework, which combines online photometric calibration and deep fusion algorithms, and is suitable for mobile robots in outdoor environments and can build a real scale dense map. At the same time, deep learning also has advantages in dealing with complex and dynamic scenarios. D2-Net [100] can also achieve satisfactory results in complex situations such as Angle changes and occlusion. DynaSLAM [82] combines dynamic object detection with SLAM technology to achieve modeling and tracking of dynamic objects, which to some extent improves the accuracy of localization and mapping in dynamic environments. You Y et al. [91] proposed a multimodal semantic SLAM system (MISD-SLAM), which can effectively solve the challenges of dynamic environments and complex semantic scenes. Firstly, an instance segmentation network is used for semantic recognition, and the ORB characteristics on dynamic objects are directly removed to reduce their interference with the system. Then, combining multi view geometric constraints and K-means clustering algorithm, undefined but moving pixels are removed, and a three-dimensional dense point cloud map with semantic information is reconstructed to restore the static background. MISD-SLAM significantly improves localization accuracy and system robustness. We combine the CBAM (Convolutional Block Attention Module) attention mechanism [101] with the Mask R-CNN [102] feature extraction network to better perform semantic segmentation on dynamic objects, thereby improving the segmentation accuracy of the VSLAM system [90]. The Global Attention Mechanism (GAM) [103] can enhance the attention of channels and spaces, effectively capturing the global information of input features, thereby making the Mask R-CNN model more focused on key features. Therefore, we propose a semantic VSLAM system based on Global Attention Mechanism (GAM) [92], which improves accuracy by 38.78% compared to DynaSLAM [82].

Recently, Lidenberger et al. [94] re-examined the design decision of SuperGlue and proposed LightGlue, which further improved the accuracy, efficiency and training ease of the system. LightGlue enables faster inference on easy-to-match image pairs, while maintaining accuracy on challenging image pairs, by predicting correspondence and allowing model introspection. Li et al. [95] proposed a texture-oriented and homogenizing FAST feature extractor (TOHF), which integrates HVS(Human vision system) to enhance texture analysis. By adopting a two-stage threshold strategy, the feature point distribution is dynamically adjusted to balance computing efficiency and storage

Table 6
Comparison of VSLAM back-end techniques.

Back-end methods	Filter methods [110–112]	Bundle adjustment methods [113–115]	Graph optimization methods [116–118]
Data processing methods	Online, incremental updates	Batch processing, processing large amounts of data at once.	Global optimization, considering all observed data.
Advantages	1. High computational efficiency, suitable for real-time applications. 2. Suitable for processing online data streams.	1. High precision, global optimal solution can be obtained. 2. Suitable for high-precision scenarios.	1. Able to handle large-scale problems. 2. Suitable for sparse and nonlinear problems. 3. Able to integrate a variety of sensors.
Disadvantages	1. Difficult to deal with highly nonlinear systems. 2. Difficulty dealing with large-scale problems.	1. Large amount of calculation, not suitable for real-time applications. 2. Sensitive to initial values.	1. High algorithm complexity. 2. High requirements for graph construction and management.
Suitable conditions	1. The scenario of high real-time demand. 2. Environments where data streams are continuous and dynamically changing.	1. The application of high accuracy. 2. Sufficient computational resources for offline processing.	1. Large-scale, long-term SLAM applications. 2. Multiple sensor information needs to be integrated.
Rate of convergence	Middle	High	High
Robustness	Middle	High	High
Memory and compute efficiency	Low	High	Middle
Global consistency	Middle	High	High
Sensitivity to initial values	High	Low	Middle

requirements. This method is helpful to improve the robustness of the system in complex scenarios. These optimization has brought technological progress to the field of SLAM, and also has a wider prospect in practical engineering applications. In addition, many scholars have reviewed and summarized related technologies [104–109]. Obviously, in the development process of VSLAM, the introduction of deep learning has also become a key technology. It breaks through the limitations of traditional methods and utilizes advanced semantic analysis to enhance the perception and intelligence level of the system.

However, in practical applications and landing, traditional methods still dominate, rather than deep learning or neural network techniques, which can be attributed to the following: (1) Hardware cost and computing power. Deep learning typically relies on GPUs for computation, which are not only expensive but may also compete for resources with other semantic aware project teams. In contrast, traditional methods can run on the CPU, with fast speed and low computational cost. (2) Evaluation of the importance of feature matching. In visual SLAM, feature matching is important but not a decisive factor. Many SLAM algorithms are similar in the front-end, with keyframe processing and back-end optimization being more critical, and deep learning features may not necessarily bring significant improvements in this regard. (3) Advantages and limitations in specific scenarios. Deep learning features may perform well in complex scenes with drastic lighting changes, camera movements, or long-term localization problems. But this does not mean it is applicable in all scenarios, as traditional features can already meet the requirements in many situations.

There is no doubt that deep learning has great potential and value in VSLAM, but its wide application still needs to consider complex factors such as technology maturity, specific needs, hardware cost and applicable scenarios. Future research should focus on how to optimize the extraction efficiency of deep learning features, improve applicability, and flexible application in different scenarios to achieve broader and more effective practical deployment.

4. Back-end modules

In the VSLAM system, the front-end module extracts information from continuous image or video frames and performs preliminary data association and motion estimation. However, the estimated results may be affected by various factors such as sensor noise, data loss, dynamic environmental changes, or unstable lighting conditions, resulting in a certain degree of localization and mapping errors. The backend module optimizes and corrects the initial estimates provided by the front-end globally or locally with the idea of minimizing errors, thereby obtaining more accurate and robust trajectories and three-dimensional environmental maps. In addition, when the system detects a loop, the backend module will also cooperate with the loop detection module to introduce new constraints to correct accumulated errors, thereby improving the accuracy and robustness of the entire VSLAM system. In back-end optimization, we divide the main optimization methods into three types: filter optimization method, Bundle Adjustment (BA) optimization method, and geometry optimization method, as shown in Table 6.

4.1. Filter optimization methods

In VSLAM system, filter optimization is a linear optimization method. Common filter optimization methods include Extended Kalman Filter (EKF) [110] and Batch Filter Method [111]. EKF is a recursive filtering method that is computationally efficient. It estimates state variables through a process of state prediction and observation update. EKF is typically suitable for the back-end optimization module of real-time systems. However, it may face challenges when dealing with large-scale systems and complex nonlinear problems. On the other hand, batch filtering methods are least-squares-based optimization techniques [112]. These methods perform global optimization by defining a cost function to adjust the state variables of the system. However, their computational complexity increases rapidly with the size of the system.

Recent research has seen improvements in nonlinear filtering optimization methods, particularly for the Kalman filtering method used in earlier VSLAM back-ends. For instance, Wang et al. [119] introduced a back-end optimization method based on the Kalman filter and the factor graph. This approach employed Multi-State Constrained Kalman Filtering (MSCKF) to estimate the state of each frame, transformed the Kalman gain form, and converted the measurement inversion problem into a state inversion problem. Factor graph optimization was applied to key frames, enabling simultaneous optimization within a sliding window. This optimization system ensures a high degree of coupling within the system while maintaining a certain level of real-time performance and accuracy. However, this method may face high computational complexity and memory consumption when processing large-scale data. Abdollahi et al. [120] introduced an enhanced multi-state constrained Kalman filtering method suitable for visual inertial odometers. To address the computational cost associated with the existing MSCKF optimization method, they devised a faster alternative known as Fast MSCKF. This modification involves extracting features solely from keyframe images and implementing an optimized strategy with three cases designed to trigger filter updates. The Fast MSCKF method offers improved speed and accuracy compared to the traditional MSCKF approach. But it is limited by the number of tracking

feature points. Goor et al. [121], building upon the concept of Lie group symmetry, proposed an isovariant filter called EQF. They implemented this filter in a more robust VIO (Visual-Inertial Odometry) system, with the majority of its back-end optimization performed using EQF. EQF consists of the state matrix, input matrix, and isovariant output matrix, prioritizing feature symmetry and computational order over the traditional Extended Kalman Filter (EKF). As a result, more accurate results can be obtained. However, the computational complexity of EQF method is high, which may lead to low operating efficiency.

The filter based optimization method is especially suitable for the application scenarios requiring real-time response and low computational load. However, filter methods typically only allow for local optimization, which only considers the optimization of the current state without considering global consistency. During long-term operation, this local optimization may lead to accumulated errors and drift in the system, thereby reducing overall accuracy. Meanwhile, filter methods require storing a large number of states, which makes them less suitable for handling large-scale scenarios. With the improvement of computer processing power and algorithm, graph optimization and Bundle Adjustment (BA) optimization have become the mainstream.

4.2. Bundle adjustment(BA) optimization methods

In VSLAM systems, Bundle Adjustment (BA) [122] optimization and geometric graph optimization are classic nonlinear optimization methods. It is mainly used to simultaneously optimize the state variables of the system to minimize the cost function and reduce the reprojection error of all observations. BA optimization, as a core component of visual SLAM, aims to optimize both the pose of the camera and the position of 3D points in the scene. This process aims to improve the accuracy of camera localization and map points by reducing the difference between the position of feature points detected in the image and the position of feature points estimated based on 3D points and camera pose. BA optimization can ensure the global consistency, effectively correct the drift and accumulation errors in the system, and improve the accuracy and stability of the system. However, its computational complexity is high, especially when processing large-scale data, it will consume a lot of computing resources, which makes it relatively difficult to apply BA optimization in real-time scenarios.

BA method was first proposed by Robert M. Haralick et al. [123] and applied to photogrammetry and cartography. With the development of computer vision and digital photography, BA method has been widely used, and has made important progress in the fields of 3D reconstruction, SLAM and structured light. Recently, some scholars have made improvements on the basis of BA optimization. Han et al. [114] significantly reduced the complexity of the problem while maintaining accuracy by decomposing the nonlinear problem into linear components (feature position) and nonlinear components (camera attitude). This method provides an effective global attitude estimation strategy and is especially suitable for devices with limited computing power. However, this method may have precision loss when dealing with some complex scenes. Ming et al. [124] proposed a BA optimization method based on adaptive robust kernels during pose estimation. The adaptive robust kernel replaces the original error double precision paradigm metric, making the backend optimization of the system smoother and more stable. But this method has a strong dependence on the selection of adaptive robust kernel parameters. Fan et al. [113] proposed a new decentralized approach to solve large-scale BA problems. By reexpressing the reprojection error and deriving a new proxy function, the authors can decoupage the optimization variables from different devices. This makes it possible to decompose large BA problems into independent subproblems for parallel solving. This method helps to improve the performance and accuracy of SLAM systems in large-scale and complex scenarios. However, this method can be affected by local minima. Wang et al. [125] proposed a BA optimization method for saliency, which captures the semantic and geometric information of the scene through the saliency model [115], simulates the ability of human eyes, and takes the value of saliency graph as the weight of feature points. Different from the general BA optimization, all feature points in the image are extracted. Instead, the necessary feature points are selected through the significance model more efficiently for BA optimization, which improves the performance to a certain extent and ensures the real-time performance of the system. But this method may be affected by the accuracy and robustness of the significance model.

4.3. Geometric graph optimization methods

In VSLAM systems, geometric optimization methods are widely used in back-end optimization to estimate the relationship between camera attitude, map point position and sensor observation. Graph optimization mainly consists of vertices and edges. Vertices typically represent state variables, such as camera pose, map point positions, etc. The edges represents the constraint relationship between vertices, which can be observation values, relative pose relationships, etc. By minimizing the cost function, these nodes (state variables) are adjusted to minimize the observation error, which helps to improve the accuracy and robustness of VSLAM systems in localization and mapping. The method based on geometric diagram optimization shows its unique advantages in VSLAM applications dealing with large-scale and complex environments. However, this method has a relatively high computational load and may not be suitable for scenarios with high real-time requirements.

Geometric graph optimization methods originated from the basic theories of graph theory and geometry, and were first proposed by Leonhard Euler et al. [126]. Early geometric optimization methods are mainly used in topological structure analysis, path planning and layout optimization. With the development of computer vision and robotics, geometric graph optimization methods have been gradually applied and expanded, and a variety of variants and improved algorithms have been derived. In recent years, methods based on geometric graph optimization have been further developed and improved. Akimoto et al. [116] proposed an information geometry optimization method that uses proxy functions to guide the progress of evolutionary algorithms. This method can help reduce the number of evaluation of the objective function, thus improving the efficiency of the optimization process. But this method may be influenced by the selection and training of proxy functions. Lu et al. [117] introduced pose graph and BA optimization techniques into the training process of deep learning networks. This innovative approach allows the network to be iteratively updated when estimating camera motion and depth information to better meet unsupervised luminosity and geometric constraints. Compared with other unsupervised monocular visual mileage calculation methods, this method has achieved significant improvement in motion estimation. But this method has a high computational complexity. Wei et al. [118] adopted a method based on motion consistency constraints to select a set of matching samples from existing feature matching algorithms to obtain higher-quality matching samples, thereby improving the computational accuracy of the geometric transformation model and the estimation of the current pose. But this method has poor robustness in complex scenarios.

For the whole VSLAM system, the back-end module can not only optimize the pose state, but also improve the overall accuracy of the system, but there are still some problems and challenges. The complexity of back-end optimization may increase with the expansion of state and map scale, so it is necessary to develop more efficient optimization algorithms and data structures to cope with it. The nonlinear optimization in the back-end module is easy to fall into the local minimum, which leads to the deviation of the optimization result from the real solution, so it needs to set the appropriate initial value, optimization strategy and constraint conditions to solve it. In real-time applications, the back-end module needs to complete the optimization process in a limited time, and how to achieve better optimization results in the shortest time is also a challenge.

Table 7
Relevant mathematical models.

Model names	Mathematical models	Impacts and effects
Extended Kalman filter [110]	State prediction: $\hat{x}_{k\|k-1} = f\left(\hat{x}_{k-1\|k-1}, u_{k-1}\right)$ $P_{k\|k-1} = F_{k-1}P_{k-1\|k-1}F_{k-1}^T + Q_{k-1}$ Observation update: $y_k = z_k - h(\hat{x}_{k\|k-1})$ $S_k = H_k P_{k\|k-1} H_k^T + R_k$ $K_k = P_{k\|k-1} H_k^T S_k^{-1}$ $\hat{x}_{k\|k} = \hat{x}_{k\|k-1} + K_k y_k$ $P_{k\|k} = (I - K_k H_k) P_{k\|k-1}$	Extended Kalman filter can effectively deal with noise and uncertainties in the system, thus improving the accuracy and robustness of VSLAM.
Fast MSCKF [127]	Kalman gain: $S = H_n P_{k+1} H_n^T$ $K = P_{k+1\|k} H_n^T S^{-1}$ Updated status: $P_{k+1\|k+1} = P_{k+1\|k} - KSK^{-1}$ $\tilde{X} = Kr_n$ $\hat{X}_{k+1\|k+1} = \hat{X}_{k+1\|k} \oplus \tilde{X}$	The proposed Fast MSCKF has a processing time of 250 (ms) per frame and the original MSCKF has a processing time of 1400 (ms) per frame.
Significance BundleAdjustment [128]	Significant weights: $w_i = aS^2(x_i, y_i) + b$ BundleAdjustment optimization: $E_{k,j} = w_i \|x_{(\cdot)}^j - \pi_{(\cdot)}(R_k X^j + t_k)\|_{\Sigma}^2$	The ATE values on the KITTI dataset and the EuRoc dataset were tested by evo and were basically better than the current mainstream vSLAM frameworks.
Adaptive robust kernel function [129]	Robust kernel function: $\{R, t\} = argmin_{R,t} \sum_{\lambda \in x} \rho_h(\|p^\lambda - F(RP^\lambda + t)\|_{\Sigma}^2)$ Adaptive robust kernel functions: $\{R, t, \delta\} = argmin_{R,t,\delta} \sum_{\lambda \in x} Q(\|P^\lambda - F(RP^\lambda + t)\|)$	Incorporating this function improves performance in real-world environments compared to ORB-SLAM3.
Equal variable filter EQF [121]	EQF: $\dot{\hat{X}} = \hat{X}\Lambda(\phi(\hat{X}, \hat{\xi})(\Omega, a))\Delta\hat{X}$, $\Delta = D_{E\|id}\phi_{\hat{\xi}}(E)^{\dagger} \cdot D_{\xi\|\hat{\xi}}\vartheta(\xi)^{-1}\sum C_t^{\star T} N_t^{-1}(y - h(\hat{\xi}))$, $\dot{\sum} = \dot{A}_t \sum + \sum \dot{A}_t^T + M_\varepsilon + B_t M_t^m B_t^T - \sum C_t^{\star T} N_t^{-1} C_t^{\star} \sum$	Introduces EQF, which is 2.14x faster than the second fastest algorithm in the EuRoc dataset and 5.3x faster than the second fastest algorithm in the UZH FPV dataset

In Table 7, we list several cutting-edge back-end optimization methods and show some related mathematical model formulations, as well as relevant explanations of their impacts and roles. So that the reader can fully understand the principles and implementation details.

5. Loop closure modules

In VSLAM systems, the loop detection module is a key component, especially in navigation and mapping tasks over long distances or large-scale spatial ranges [130]. Due to various factors such as sensor noise, dynamic environmental changes, and fluctuations in lighting conditions, the front-end and back-end modules may generate accumulated localization and mapping errors during data processing. The main responsibility of the loop detection module is to identify whether the robot or camera has returned to a previously visited location through highly accurate and reliable algorithms, thus determining whether a "loop" has been formed. Once a loop event is successfully detected, the module works in conjunction with the back-end optimization module to correct and distribute these accumulated errors by introducing additional geometric or algebraic constraints. This not only improves the localization accuracy of the system on a local scale, but also helps achieve map consistency and accuracy on a global scale, thus significantly enhancing the overall performance and reliability of the VSLAM system.

The errors in loop detection can be divided into two categories: one is to recognize different scenes as the same scene, and the other is to detect the same scene as different scenes. This article briefly describes four common loop back module issues. (1) The scale change problem refers to the judgment error caused by the change in the distance ratio between the camera position and the object at different time points. (2) The problem of perspective change refers to the judgment error caused by the camera's perspective change when observing the same scene at different time points. (3) Dynamic object interference means that a dynamic object may be incorrectly identified as a loop feature. Dynamic objects can also cause changes in the location and appearance of the access scene. The front-end module of the VSLAM system may also produce false guidance when tracking dynamic objects. Loop detection module may face the problem of loop obstacle in dynamic environment. (4) The illumination change problem is that factors such as weather, time and season may affect the illumination in the scene, and the illumination change may lead to the change of features in the same scene, causing the scene to be incorrectly identified by the loop detection module. These problems make the feature matching of loop detection module difficult.

With the development of the times and the progress of technology, many loop detection methods based on different principles have appeared recently. As is shown in Table 8, we classify them into three categories based on features, based on geometric information and based on semantic information.

5.1. Feature information based loop detection methods

Feature-based methods for loop detection extract and match visual features in images to identify similar features or keyframes. These methods facilitate the recognition of previously encountered scenes but are susceptible to variations in scale, viewpoint, and dynamic environments.

In the early stage, loop detection was mainly based on feature point matching or descriptor matching. For example, Gutmann et al. [139] proposed a loop detection method based on SIFT descriptor to achieve loop closure by matching SIFT feature points. Subsequently, Mur-Artal et al. [140] proposed a loop detection method based on ORB feature points, which used ORB descriptors to match feature points and realize loop identification. In recent years, the feature information loop detection method has also been greatly developed. ORB-SLAM3 [87] uses DBoW2 [131] as a feature bag for location recognition algorithms in loop detection. When ORB-SLAM3 creates a new keyframe in the mapping thread, the system will start the position recognition process connected to the word bag. ORB-SLAM3 will obtain a set of candidate frames from multiple map systems [132] that have similar features to the current new keyframe. After obtaining the set of candidate frames, ORB-SLAM3 performs geometric validation on each candidate frame to ensure that they are consistent with the current new keyframe in space. For candidate frames that have passed geometric validation, ORB-SLAM3 calculates a Sim3 transform to align the current new keyframe with the candidate frame. If the similarity transformation is successfully calculated, ORB-SLAM3 will perform global bundle adjustment to optimize the consistency of the entire map. Finally, ORB-SLAM3

Table 8
Comparison of VSLAM loop techniques.

Loop methods	Feature information [87,131,132]	Geometry information [133–135]	Semantic information [136–138]
Principle	By extracting the key points in the image and their feature descriptors,and then matching the key points stored in the database, the repeated scenes in the environment can be detected.	Use 3D point clouds or depth information to estimate the geometry of the scene and compare it to known structures.	By identifying and understanding the semantic information in the scene and using it for environment recognition.
Advantage	Changes of light and perspective has certain robustness.	It can provide more accurate spatial information.	Performs better in dynamic environments and is able to understand high-level information about the scene.
Disadvantage	It does not work well in environments where features are lacking or repeated.	It is difficult to deal with dynamic objects and complex scenes.	It is highly dependent on the accuracy of semantic segmentation and recognition algorithm.
Application scenario	1. Structured and obvious environment. 2. Scenes with rich textures. 3. Scenes with stable lighting conditions.	1. Large scale outdoor environment. 2. Scenes with distinct 3D structures.	1. Dynamic and complex environment. 2. Navigation application field.
Computational complexity	Middle	Middle	High
Robustness	High	High	Middle
Real-time	Middle	Middle	Low

updates map points and keyframes, fuses similar map points, and updates the state of keyframes to reflect the latest system state and observed scenes. The algorithm framework is shown in Algorithm 1. DBoW3 [141] further improves the detection accuracy and efficiency by introducing new feature descriptors, optimized matching algorithms and more effective data structures on the basis of DBoW2. FBoW [142] speeds up the feature matching process by optimizing data structures and search algorithms to reduce computation time while maintaining high accuracy, which is about 80 times faster than DBoW2.

Algorithm 1 ORB-SLAM3 Loop Closure

Initialization
Load new frame F_t at time t
Compute BoW representation for F_t
Detect Loop Candidates
if BoW similarity suggests potential loop closure **then**
    Candidates[ ] = retrieve frames that are likely to be loop closures
end if
Geometric Verification
for each frame F_i in Candidates[ ] **do**
    if geometrically consistent with F_t **then**
        Add F_i to Verified_Candidates[ ]
    end if
end for
Compute Similarity Transformation (Sim3)
for each frame F_j in Verified_Candidates[ ] **do**
    Compute Sim3 transform that aligns F_t with F_j
end for
Global Bundle Adjustment
if Sim3 transform is successfully computed **then**
    Perform Global Bundle Adjustment
end if
Map Points and Keyframe Update
Fuse similar map points
Update keyframes

Offline trained word bag models may be affected by complex scenes, and online trainable word bag models have been introduced. In the loop detection of OV^2-SLAM [143], its loop detection is based on an online trainable word bag. Its strategy is to use the newly acquired image frames for word bag training, and choose to only perform loose BA optimization on the modified map, greatly reducing the time required for loop back detection. Tsintotas et al. [144] proposed a scalable bag-of-words model using an incremental bag of tracked words, known as BoTW. This method encodes the trajectory by selectively tracking unique visual features and incrementally constructing the bag of words. The bag-of-words scores are calculated through a nearest-neighbor voting approach [145], with optimizations in both score acquisition and computational complexity. This maintains high precision and recall in long-distance trajectories. Furthermore, we introduce a loop detection algorithm built upon an enhanced, real-time updated bag-of-words model that utilizes a monocular camera [146]. This fused bag-of-words is designed to be relevant for mobile robot applications. It is generated by extracting feature descriptors from online images and fusing them with preloaded offline descriptors. The fused bag of words is dynamically adaptable to changes in robot application scenarios, thereby significantly boosting the system's loop detection capabilities.

The loop detection method of feature information is usually based on matching unique feature points or feature descriptors, which has a certain degree of robustness for local changes in the environment, such as lighting changes or object movement. This method can provide richer environmental information by matching feature points and comparing descriptors, which helps to recognize and understand the environment. However, feature extraction and matching algorithms often require significant computational resources, especially when dealing with large-scale data, which may result in heavy computational burden and limit the real-time and scalability of the methods. Meanwhile, this method is also susceptible to changes in scale and perspective, as well as the influence of dynamic scenes.

5.2. Geometric information based loop detection methods

The loop detection method based on geometric information aims to obtain geometric information such as point cloud and depth map by depth sensor to assist the system in loop detection. It improves the accuracy of loop detection by comparing point cloud data and depth map data of different time steps, but this method is susceptible to the change of illumination.

In the early days, Durrant Whyte and Bailey [147] proposed a loop detection method based on geometric constraints, which matches feature points and utilizes geometric relationships for loop closure. But this method performs unstable in complex scenes. Subsequently, many scholars further improved the loop detection method for geometric information. An et al. [135] introduced a loop detection method based on depth features and proximity maps. They extracted both global and local depth information features using convolutional neural networks and constructed a visual database for querying similar positions on the time axis. This approach was evaluated using local depth information features and demonstrated very high loop detection performance. But this method consumes more computing resources. Zhou et al. [148]

employed local 3D depth descriptors for loop detection in VSLAM. They also proposed a novel overlap metric for loop detection, which involved estimating the relative pose registration of loop candidate point clouds and calculating the metric error of the points on mutual nearest neighbor descriptors. This method outperformed traditional Lidar-based loop detection approaches. However, there may be certain challenges in loop matching and pose estimation in complex scenarios using this method. Yang et al. [134] used the point cloud segmentation algorithm to divide the point cloud space into different units to ensure the continuity and convergence of the cost function. In order to solve the recognition problem that the visual closed-loop detection method relies heavily on the appearance of the environment, they introduce a deep completion algorithm to fuse sensor data to ensure the robustness of the algorithm. However, the real-time and accuracy of the point cloud segmentation algorithm still need to be further improved. Wang et al. [133] proposed a closed-loop detection method based on multi-scale point cloud feature transformer. In this method, the original point cloud features with different resolutions are obtained by voxel sparse convolution technology, and the context relationship between these features is established by transformer network, so as to realize multi-scale feature fusion and global descriptor generation. By integrating multi-scale point cloud feature information, this method can not only enhance global modeling capability but also reduce information loss. Experimental results show that the proposed method can improve the accuracy and robustness of VSLAM localization and mapping. But this method may face challenges when dealing with large-scale data.

The work flow of common geometric information loop detection methods is as follows: Firstly, the system acquires depth data, such as point cloud data or depth map data. Secondly, the system preprocesses the acquired data, such as filtering and downsampling. Thirdly, the system extracts 3D geometric features from the pre-processed data. If it is a new scenario, the extracted features are stored in the database. Then, the features of the current frame are matched with those in the database. If the feature is matched successfully, the geometry verification is entered. The geometric verification phase can be verified using ICP algorithm. If the geometric verification meets the criteria, a closed loop is performed. Finally, update the SLAM map and track. As shown in Algorithm 2.

Algorithm 2 Geometric Information-Based Loop Closure Detection

Input: Depth data
Output: Loop closure detection results and pose adjustment
Begin
Data Acquisition: Acquire depth data (point clouds or depth maps)
Data Preprocessing (e.g., filtering, downsampling)
Feature Extraction
Extract 3D geometric features
Feature Database
if new scene **then**
    Store features in database
end if
Feature Matching
Match current frame features with database features
Matching Evaluation
if match reaches threshold **then**
    goto Geometric Verification
else
    goto Data Acquisition
end if
Geometric Verification: Perform geometric validation using algorithms like ICP
if geometric validation meets standards **then**
    Confirm loop closure
else
    goto Data Acquisition
end if
Update SLAM map and trajectory based on confirmed loop closure
End

Loop detection methods with geometric information usually use geometric information for matching, which is relatively simple to implement compared to complex semantic or deep learning methods. Since the processing of geometric information usually does not involve complex neural network computation or semantic analysis, the computational efficiency is high and it is suitable for real-time applications. However, the loop detection method of geometric information usually depends on the geometric structure of the scene and is more sensitive to environmental changes.

5.3. Semantic information based loop detection methods

Semantic information-based loop-back detection methods aim to distinguish the similarity of scenes by identifying the semantic information of objects in the scene, so as to enhance the robustness and accuracy of loop-back detection. However, semantic information-based methods usually have poor real-time performance and consume a lot of resources.

In the early days, Gupta et al. [149] introduced a semantic segmentation method based on conditional random fields to extract semantic information from scenes. In recent years, with the development of deep learning and neural network technology, semantic information loop detection methods have also been greatly improved. Osman et al. [136] proposed a multiscale semantically-aware visual loop detection model. This model enhances scene understanding through a multiscale deep self-coding network called PlaceNet. It improves the semantic fusion layer for scene understanding by training the network to learn grayscale semantic maps of static and dynamic objects in an image, enabling differentiation between dynamic targets. The model combines various elements, including the semantic fusion network, multiscale coder–decoder architecture, weighted scale loss, and temporal similarity-based similarity detection, to create an instantly usable loop detection model. Yuan et al. [137] introduced a visual loop detection model that integrates semantic, visual, and geometric information. This method combined semantic information and visual features to construct a semantic bag-of-words model, which encodes the geometric relationships of semantic graphs and designs a semantic landmark vector model, effectively fusing semantic and visual features to achieve robust loop detection. Wang et al. [150] introduced a two-stage loop detection strategy, combined local continuity constraints at the front end and semantic model based on deep learning at the back end, and provided a new and more effective solution for the loop detection based on semantic information. Li et al. [138] quickly and accurately detected the loop closure by integrating Gist features, semantic features and appearance features. Experimental results show that the algorithm can detect the loop closure quickly and accurately under the condition of illumination, view point and object change. Wang et al. [151] used YOLOv5 for object detection and recognition, marked dynamic and static objects, and then constructed semantic topology diagram based on the location information of static objects. By introducing topological similarity value to determine relocation and loop detection, the accuracy of relocation and loop detection can be effectively improved in dynamic environment.

The workflow of common semantic information loop detection methods is as follows: (1) The system will obtain image data and perform preprocessing. (2) The system extracts feature points from the image. (3) The system utilizes deep learning models for semantic segmentation and annotation of images, identifying and classifying different objects and regions in the image. (4) Based on feature points and semantic information, the system establishes a three-dimensional

map of the environment. (5) The system detects potential loop candidates by comparing the feature points of the current image with the feature points of the previous image stored in the map. (6) For each potential loopback candidate, the system performs a semantic consistency check. (7) The system confirms the existence of loops by combining feature matching and semantic consistency analysis. (8) Once the loop is confirmed, the system will update the map based on this information. As shown in Algorithm 3.

Algorithm 3 Semantic Loop Closure Detection in VSLAM

```
Input: Image or video data from cameras or sensors
Output: Updated Map with confirmed loop closures
Data_Acquisition()
Preprocessing()
Feature_Extraction()
Semantic_Segmentation_and_Labeling()
Mapping_and_Localization()
while Exploring or Localizing do
    Candidates ← Loop_Closure_Candidate_Detection()
    for all candidate in Candidates do
        if Semantic_Consistency_Check(candidate) then
            if Loop_Closure_Confirmation(candidate) then
                Map_Update(candidate)
            end if
        end if
    end for
end while
return Updated Map
```

The loop detection method of semantic information can better understand the types, structures, and semantic relationships of objects in the environment by utilizing semantic information, thereby improving the cognitive and comprehension ability of the environment. However, the acquisition and processing cost of semantic information is relatively high. Meanwhile, the accuracy and reliability of semantic information are limited by the quality of the semantic model and the types and changes of objects in the environment, so its applicability may be limited in different environments.

6. Mapping modules

Unlike the front-end module that makes preliminary motion estimates from image data, and the back-end module that fine-adjusts these preliminary estimates through optimization algorithms, the main task of the mapping module is to use these optimized data to build a highly accurate and usable 3D map of the environment. This process is constrained by various factors such as sensor accuracy, data processing capabilities, and environmental complexity. Through advanced data fusion and spatial analysis techniques, the mapping module aims to generate a complex 3D model that can meet real-time navigation needs and is also suitable for subsequent data analysis and applications. It should be emphasized that the mapping module not only relies on accurate positioning information provided by the front-end and back-end modules. And after successful loop detection, it will work closely with the loop module to further optimize and correct the entire map structure. This can improve the consistency and reliability of the map on a global scale. The close collaboration of this series of modules ensures that the VSLAM system can demonstrate excellent stability and high accuracy in various complex and challenging application scenarios. The form of the map depends on the application scenario of SLAM, which can be divided into two types: metric map and topological map [152]. Metric maps are commonly used to represent information about the current environment, which can accurately depict the positional relationships between various objects in the map, further subdivided into sparse maps and dense maps. Feature based algorithms are commonly used to generate sparse maps. The direct method can obtain semi dense or dense maps. Topological maps place greater emphasis on connectivity within the environment, abstracting it as a network structure of nodes and edges. Nodes represent key locations in the environment, such as rooms, corridors, or intersections, while edges represent passable paths between these nodes.

Common map forms in visual SLAM include point cloud maps, occupied grid maps, octree maps, and semantic maps, as shown in Table 9. All maps are designed to assist robots in completing tasks such as localization, navigation, obstacle avoidance, reconstruction, and interaction [161]. Localization, as a basic function, can usually be implemented through sparse maps. The sparse map mainly describes the key feature points in the environment and their position relations, and provides enough information for the robot to determine its own position. Dense maps contain more details, such as the shape and size of objects, so they can provide robots with more abundant environmental information to facilitate complex tasks such as path planning, avoiding obstacles or rebuilding the environment [162]. In addition, when it comes to interactions between robots and humans or maps, semantic maps need to be used. The semantic map contains not only the spatial information of the environment, but also the semantic information of the environment elements, such as object categories, attributes, etc., which helps the robot to better understand and adapt to the environment. In addition, dense maps can be further converted to occupy raster maps or octree maps, which are more effective in dealing with large-scale environments or complex shaped obstacles and can be used for accurate navigation and obstacle avoidance.

6.1. Sparse maps

In VSLAM systems, sparse maps are an efficient way to express the environment. Sparse maps only focus on key points or feature points in the environment, such as corner points, edge points, etc. Because these key points have rich geometric information and are relatively easy to detect and track, they are selected as representative parts of the map. By utilizing these key points, the VSLAM system can estimate the camera's motion and establish a simplified but effective representation of the environment. Fig. 3 depicts the sparse map generated by ORB-SLAM2 in an outdoor environment [78].

In the early days, Smith and Cheeseman [163] used particle filtering to handle the uncertainty of sensor data and construct maps, but this method may encounter computational efficiency issues when facing complex environments. Subsequently, Montemerlo et al. [164] proposed a sparse point cloud map construction method based on probability graph by introducing probability graph model, which improved the accuracy and stability of the map. However, there may be some challenges in the real-time performance of the method. Recently, many scholars have further improved sparse maps. Bokovoy et al. [165] adopted outlier removal and upsampling methods to remove outlier points and noise, which further improved the density and quality of sparse maps. But this method may lead to information loss. Lyu et al. [166] utilized a SpConv accelerator to realize high-speed and energy-saving sparse point cloud processing, and utilized the inherent sparsity of each voxel to reduce heavy computing workload, thus eliminating computational redundancy and optimizing on-chip memory management. However, this method performs poorly in handling complex scenes. Xie et al. [167] proposed a monocular SLAM system that fused point features and line features. Based on sparse image alignment, the system realizes fast tracking of non-key frames and effectively extracts and matches the point features and line features in key frames. This important improvement not only significantly improves the performance of SLAM in motion estimation, but also enhances the detailed representation of environmental maps. But its computing resource consumption is relatively high.

Sparse point cloud maps contain fewer data points and take up less memory, allowing the system to process and analyze environmental information faster. However, due to the limited data points, sparse point cloud maps may lose some details of the environment.

Table 9
Common map forms.

Map types	Descriptions	Resource consumption	Localization	Navigation	Obstacle avoidance	Interaction	Typical algorithms
Sparse map	Consists of a small number of feature points, requiring less computational effort and storage space	Low	Middle	Middle	Low	Middle	ORB-SLAM [153] PTAM [154] SVO [76]
Dense map	The environment is described using a large number of 3D spatial points, discretizing all objects in the environment into a dense point cloud.	High	High	High	High	High	KinectFusion [155] ElasticFusion [156]
Occupy grid map	The environment is divided into a number of grids, each with three states: occupied, idle and unknown.	Middle	Middle	Middle	Middle	Middle	Gmapping [157] Rtabmap [158]
Octree map	Rational compression of the 3D environment space using an octree storage method.	High	High	High	High	Middle	DS-SLAM [81]
Semantic map	It contains not only geometric information about geospatial location, but also semantic information about various objects in the environment.	High	High	High	High	High	CubeSLAM [159] QuadricSLAM [160]

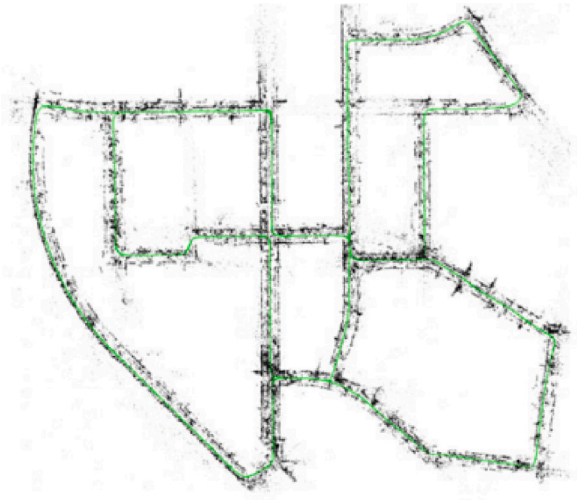


Fig. 3. Sparse maps generated by ORB-SLAM2 [78].

6.2. Dense maps

Dense maps usually contain a large number of feature points or pixel level depth information observed by cameras in the scene, so that each position in the map has corresponding depth values or feature points. Dense maps provide a more detailed and accurate representation of the environment, which can more accurately reflect the geometric structure and surface shape of the scene.

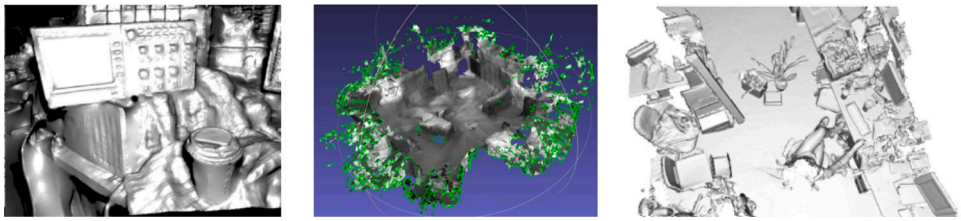
Dense maps can be generated using monocular cameras, stereo cameras, and RGB-D cameras. The use of monocular cameras for dense mapping typically involves using polar search and block matching techniques to determine the position of pixels in one frame of the image in other frames. Then, by using deep filtering to determine the depth value of each pixel, a dense 3D map is constructed [152]. The DTAM [75] system is a dense map constructed using a monocular camera, as shown in Fig. 4(a). The construction of dense map by stereo camera mainly uses the parallax information of stereo camera. There is a certain displacement in the left and right images taken by stereo cameras, and the parallax of each pixel can be calculated by matching the corresponding pixels in the left and right images. Based on the relationship between parallax and baseline, we can use the principle of triangulation to calculate the depth information of each pixel in the scene. Then, by interpolating and filtering the depth information of each pixel, the precise depth information of each point in the scene can be obtained. By mapping this depth information into the map according to the pixel position, a dense three-dimensional map can be constructed. Wen et al. [168] proposed a lightweight and fast stereo vision inertial dense mapping method. The experimental results show that this method has faster tracking speed, better depth map and point cloud reconstruction, as shown in Fig. 4(b). RGB-D cameras, on the other hand, directly provide depth information for each pixel. This feature has garnered significant interest among researchers. Newcombe et al. introduced KinectFusion, a real-time 3D reconstruction system based on RGB-D cameras. KinectFusion utilizes the point cloud generated from depth images to estimate camera positions via ICP (Iterative Closest Point), subsequently aligns multi-frame point clouds based on camera positions, and expresses the reconstruction results using a TSDF (Truncated Signed Distance Function) model. This process results in the construction of a dense map, as illustrated in Fig. 4(c).

In an early stage, Newcombe et al. [75] proposed a map construction method based on dense optical flow, which enabled real-time dense map construction. However, this method is sensitive to the change of illumination. Subsequently, Engel et al. [169] introduced the dense map construction method based on the direct method, which further improved the accuracy and stability of the map. However, this method still faces challenges in complex scenarios. In recent years, with the development of deep learning and neural network technology, the construction method of dense maps has been greatly improved. Jia et al. [170] integrated a lightweight convolutional neural network at the front end of the VSLAM system, designed to efficiently remove dynamic feature points and map points on dynamic objects. In addition, they also conducted in-depth processing of the processed static map points to build a denser point cloud map. The key contribution of this paper is that it not only significantly reduces the interference of dynamic objects to the map-building function of VSLAM systems, but also enables more precise capture and presentation of environmental information. However, this method depends on the accuracy of the network model.

Dense point cloud map can provide rich environmental details, which is suitable for high-precision perception and navigation, but VSLAM system needs more computing resources and time to process dense point cloud data.

6.3. Occupancy grid maps

Occupancy grid map, initially introduced by Elfes [171], serve as a crucial map representation format utilized for robot perception and navigation. These maps are primarily employed to depict the spatial arrangement of the environment and the distribution of obstacles



(a) DTAM [75] (b) Stereo VINS [171] (c) KinectFusion [155]

Fig. 4. Dense maps.

within it. Occupancy grid maps divide the environment into discrete grid cells, with each cell having one of three possible states: occupied, unoccupied (free or idle), or unknown. The structure of occupancy grid maps is illustrated in Fig. 5. These maps play a pivotal role in enabling machines and robots to gain a comprehensive understanding of their surroundings, facilitating more precise autonomous navigation and path planning.

Then, many scholars have improved the occupied grid map. Jang et al. [172] proposed a dynamic occupancy grid map (DOGM) that can represent information such as position and velocity of nearby objects. They used DOGM data measured by 3D Lidar sensors to detect and classify objects based on free probability, velocity, and height information. This paper applies the information data such as the position and speed of objects to the raster map, which makes the raster map more abundant and accurate to represent the environment information. However, this method may be limited by the accuracy and efficiency of real-time detection and classification of dynamic objects in the environment. Fisher et al. [173] built a vertical framework to efficiently store occupied raster maps, which achieved an order of magnitude reduction in memory footprint while also achieving a slight acceleration in map insertion and access times. But this method faces challenges in storage and access efficiency when dealing with large-scale maps. Yatim et al. [174] combined infrared sensors with neural networks and applied them to occupy grid map algorithms. They use neural networks to interpret the measurement values of adjacent sensors as occupancy values in the grid map. The experimental results show that in the process of robot navigation, the neural network ensemble algorithm can better interpret and estimate the environmental map. But this method is limited by the noise and accuracy of the sensor.

The advantage of occupancy grid map is that they are easy to build and understand, and are suitable for basic obstacle avoidance and navigation tasks. However, due to their discrete nature, occupancy grid map may not be accurate enough for tasks that require high-precision perception.

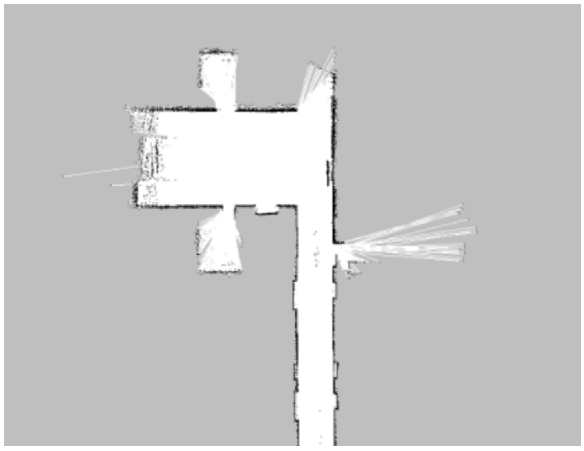


Fig. 5. Occupancy grid map [175].

6.4. Octree maps

An octree map is a versatile, space-efficient, and dynamically updated map representation. In this map structure, each node within the octree corresponds to a cubic volume, referred to as a voxel. The cubes are subdivided recursively into eight smaller sub-blocks until a specified minimum voxel size is achieved, as illustrated in Fig. 6. Notably, in an octree, if all the children of a node are either occupied or unoccupied, there is no need to further subdivide the node. This feature significantly reduces the storage requirements, making octree maps an efficient choice for representing spatial data. Fig. 7 shows an example of querying an octree map of occupied voxels at different resolutions, where multiple solutions of the same map can be obtained at any time by limiting the depth of the query.

In the early days, Finkel and Bentley [176] proposed an octree construction method based on spatial partitioning, which can effectively manage data in large-scale scenes. But this method is limited to dynamic or frequently changing scenarios. Subsequently, Samet [177] further improved the spatial index and query algorithm of octree to improve the efficiency and accuracy of map query. However, this method may face challenges of efficiency and accuracy in the face of complex shaped objects or high dimensional data. In recent years, with the development of machine learning and deep learning technology, the construction method of octree map has been greatly improved. Yu et al. [81] combined the semantic segmentation network with the motion consistency checking method to reduce the influence of dynamic targets on VSLAM systems. Finally, an octree map containing dense semantic information is generated for the VSLAM system to perform advanced tasks. But this method consumes more computing resources. Vespa et al. [178] proposed an efficient and intensive SLAM framework, which uses octree representation to improve mapping, planning and control, improves performance and flexibility in many aspects, and provides strong support for applications in the field of robotics. But this method may be limited when dealing with large-scale data.

Octree maps can represent the environment at different resolutions as needed, balancing precision and computational cost. However, building and maintaining octree maps requires not only more complex algorithms and data structures, but also more storage space.

6.5. Semantic maps

Semantic maps is a common form of map representation, which includes not only the geometric information of the environment, but also the semantic information of each location in the environment. Compared with traditional maps, semantic maps can provide richer and deeper understanding of the environment and can capture semantic information in the environment. Traditional visual SLAM technology has a significant problem in mapping, that is, the lack of high-level environment semantic information, which limits the performance of robots in intelligent obstacle avoidance, recognition, interaction and other complex tasks. In order to solve this problem better, it becomes more and more important to build accurate and reliable 3D semantic maps.

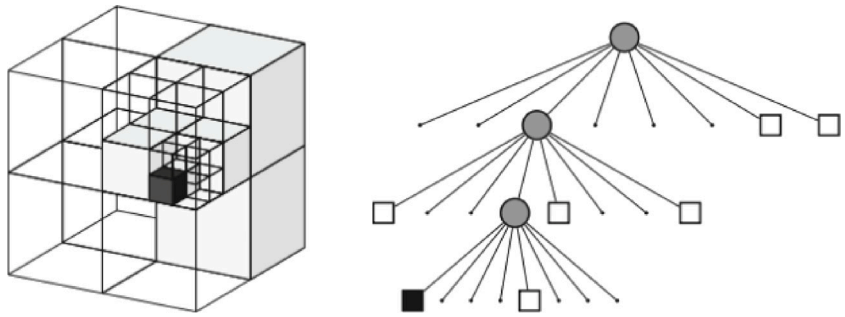


Fig. 6. Octree map [179].

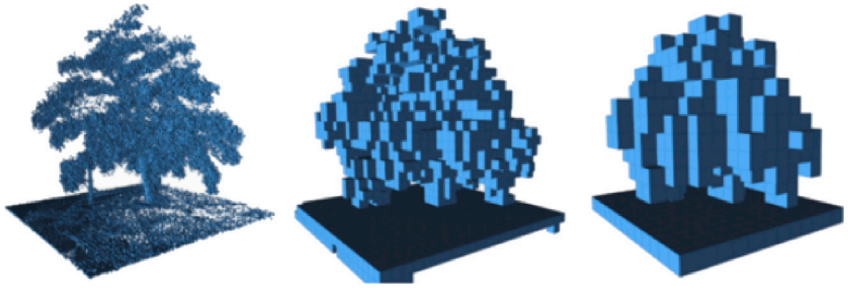


Fig. 7. The resolutions of occupied voxels were 0.08 m, 0.64 m, and 1.28 m, respectively [179].

In the early days, a prior object Computer-Aided Design(CAD) model database was often used to build 3D semantic maps [180], which could restore the real scene and save a lot of space required to store dense point cloud maps. However, CAD models are limited to objects in a predefined database. In subsequent studies, some scholars built static dense semantic maps [181–184], which combined dense visual SLAM with semantic segmentation labels, thus giving maps richer semantic information. In order to address the challenges of dynamic environments, researchers divide objects in the scene into background, moving objects, and potential moving objects through instance aware semantic segmentation [185–187]. However, these methods have limitations in real-time performance. Aiming at the real-time problem of semantic visual SLAM technology in mapping, methods for constructing sparse semantic maps have been proposed, represented by [159,160,188–194], based on the ORB-SLAM2 framework, Semantic objects are combined with sparse 3D semantic maps in real time, thus balancing the accuracy and real-time performance of the maps.

The common semantic map generation process is shown in Fig. 8. In semantic VSLAM system, object detection, semantic segmentation and instance segmentation can be used to obtain semantic information in images. At the same time, the VSLAM system obtains point cloud information through the camera sensor. For the obtained semantic information and point cloud information, we need to carry out a series of processing operations, including semantic label recognition, semantic region division and semantic map fusion. Thus a map with semantic information is constructed.

Compared with traditional visual maps, semantic maps are more widely used in intelligent scenes. However, building semantic maps requires addressing challenges such as high computational complexity, recognition of various types of objects, and map storage. In future research, addressing these challenges will drive further development of semantic mapping technology in the field of intelligent robots.

7. Analysis and discussion

In recent years, thanks to the rapid development of computer vision hardware technology and the improvement of sensor performance, VSLAM system can achieve higher real-time and robustness in complex environments. The traditional method has also made remarkable progress in feature extraction, matching algorithm and back-end optimization algorithm, which further improves the performance and stability of SLAM system. In the future, the development trend of traditional SLAM technology will mainly focus on the further improvement of real-time, multi-sensor fusion and the introduction of semantic understanding. With the continuous progress of hardware technology, we can expect the SLAM system to achieve higher real-time performance while maintaining high precision and robustness, so as to adapt to more application scenarios with high real-time requirements. At the same time, the fusion of different sensor information is one of the important ways to improve the performance of SLAM system, and the future SLAM system will be more flexible to use a variety of sensors to obtain more comprehensive and accurate environmental information. In addition, the introduction of semantic understanding will make the SLAM system's cognition of the environment more intelligent. By recognizing objects and structures in the environment, SLAM systems can provide more intelligent behavioral decisions for robots or autonomous driving systems.

Deep learning, as a popular research direction in computer vision, has also attracted much attention in the field of SLAM [97,135,136]. Deep learning enables more accurate feature extraction and matching through efficient processing of sensor data. Second, precise semantic understanding enables machines to better understand the environment, significantly improving the accuracy and robustness of localization and mapping. At the same time, the use of end-to-end learning simplifies the design of the entire VSLAM system and makes it more efficient. Deep learning has provided a powerful driving force for the development of VSLAM technology, as well as improved robustness and accuracy of the system. This technology has significant support for the localization and mapping tasks of robots in unknown environments, and is of great significance in the fields of robotics and computer vision. But combining deep learning with VSLAM methods also comes with some challenges.

Firstly, data dependency is a prominent issue. The success of deep learning in VSLAM largely depends on the available data. However, obtaining datasets with high-quality annotations may be relatively difficult, especially in specific environments or scenarios where the lack of corresponding calibration data can have a negative impact on the

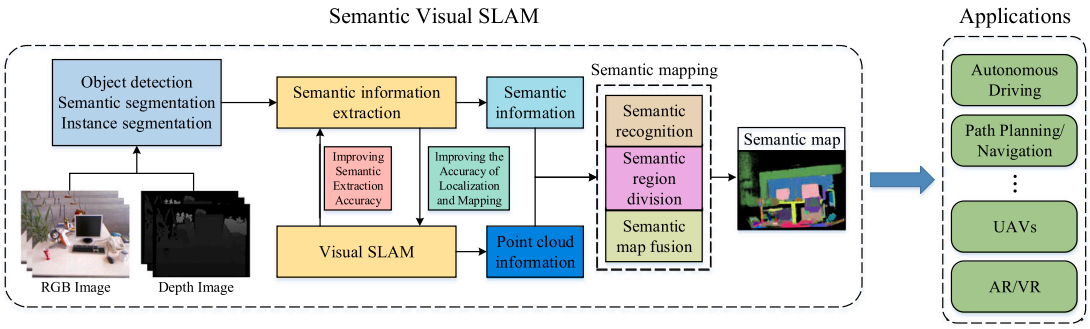


Fig. 8. Semantic map generation process.

performance of deep learning methods. Synthetic data can generate artificial datasets similar to actual data by simulating sensor data in specific environments or scenarios. In this way, the amount of data available for training can be significantly increased, thereby reducing the dependence on real calibration data. When the calibration data in actual scenarios is limited or difficult to obtain, synthetic data can compensate for the problem of insufficient data. In addition, we can first train in a relatively easy to obtain high-quality calibration data scenario, and then apply the learned knowledge and features to localization and mapping tasks in specific environments or special scenarios through transfer learning. In this way, even if the data in the target scenario is limited, the knowledge learned in the source scenario can be utilized to improve performance.

Second, deep learning tends to require higher computing resources. Training deep learning models requires a lot of computational resources, especially for complex neural network structures, which may require the use of high-performance computing devices such as GPU or TPU. This will increase the hardware cost of VSLAM systems and limit their application in resource-constrained environments. We can choose a suitable lightweight model structure to reduce the computational resources consumed during the training and inference stages. In addition, by properly allocating hardware resources, we can improve computing efficiency. For many computing processes in SLAM tasks, we can optimize the hardware structure to fully leverage the advantages of parallel computing, thereby significantly improving computational efficiency. In adapting to the needs of mobile devices and embedded systems, we need to minimize energy consumption while ensuring performance. Memory optimization is also a crucial aspect, and by adopting special memory architectures or caching strategies, we can improve data access efficiency. In order to meet the special requirements of SLAM algorithm for data format and reduce the overall system cost and complexity, we also need to support specific data formats and provide highly integrated functions. These measures will provide strong support for the performance improvement of SLAM systems, further promoting their development and application in practical applications. Finally, real-time requirement is a key index of VSLAM system. In real-time SLAM applications, mapping and localization need to be completed in a limited time, and the time consumption of deep learning models in the inference stage may affect the real-time performance of SLAM systems. Simplifying the trained deep learning model and cutting off some redundant and unnecessary connections or parameters in the model can reduce the size and complexity of the model, so as to improve the inference speed. The pruned model retains the main feature extraction and matching capabilities, while reducing the amount of computation, making it more efficient in the inference stage. In addition, by optimizing the design and implementation of the inference engine, the efficiency of the model running on specific hardware can be improved. For example, for a specific hardware platform, we can design an efficient execution strategy of the graph and use hardware acceleration to improve the reasoning speed. The combination of model pruning and inference engine optimization can significantly improve the speed of the model in the inference stage without sacrificing the performance of the model, and effectively solve the real-time challenge of deep learning in the field of SLAM.

8. Conclusions

The intelligence and efficiency of VSLAM technology is an inevitable trend in its development. This is not only an inevitable trend to meet the needs of localization and mapping in current complex environments, but also a key driving force for the development of fields such as autonomous driving and robot navigation. By introducing intelligent technology, the VSLAM system will be able to better adapt to the needs of future multi domain intelligent autonomous systems. This review comprehensively analyzes the classic technical framework of VSLAM technology, and deeply analyzes its five core components, including data acquisition, front-end module, back-end module, loop module and mapping module. For each section, we provide a detailed literature review, analysis and summary, presenting readers with the latest research developments in the field. Meanwhile, we focus on the key role of deep learning in VSLAM, and clarify the many complex problems faced by its practical application. To this end, we propose a series of solutions, such as using synthetic data to make up for the lack of data, choosing lightweight model structure to reduce the consumption of computing resources. This review provides a clear theoretical framework and practical guidance for researchers in the field of VSLAM, as well as a practical reference for engineers and decision makers in the field of intelligent systems such as autonomous driving and robot navigation. We emphasize that VSLAM technology is in a developmental stage of constant exploration and challenge. The summary analysis in this paper aims to provide solid support for further research, application promotion and interdisciplinary cooperation in this important field. From a broader perspective, our review also reflects the trend of integration in the field of artificial intelligence and machine vision, provides the possibility of rich for future research.

CRediT authorship contribution statement

Dupeng Cai: Writing – review & editing, Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization. **Ruoqing Li:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization. **Zhuhua Hu:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Junlin Lu:** Writing – original draft, Conceptualization. **Shijiang Li:** Writing – original draft, Investigation, Formal analysis, Conceptualization. **Yaochi Zhao:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant no. 62161010 and 62361024), the Key Research and Development Project of Hainan Province, China (Grant no. ZDYF2022GXJS348 and Grant no. ZDYF2022SHFZ039), and the Hainan Province Natural Science Foundation, China (623RC446). The authors would like to thank the referees for their constructive suggestions, and Shenzhen Umouse Technology Development Co., Ltd. (HD-KYH-2021307) for their support in equipment and experimental conditions.

References

[1] Y. Chang, Y. Tian, J.P. How, L. Carlone, Kimera-multi: a system for distributed multi-robot metric-semantic simultaneous localization and mapping, in: 2021 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2021, pp. 11210–11218.
[2] J. Cheng, L. Zhang, Q. Chen, X. Hu, J. Cai, A review of visual SLAM methods for autonomous driving vehicles, Eng. Appl. Artif. Intell. 114 (2022) 104992.
[3] L. Jinyu, Y. Bangbang, C. Danpeng, W. Nan, Z. Guofeng, B. Hujun, Survey and evaluation of monocular visual-inertial SLAM algorithms for augmented reality, Virtual Real. Intell. Hardw. 1 (4) (2019) 386–410.
[4] D. Dworakowski, C. Thompson, M. Pham-Hung, G. Nejat, A robot architecture using contextslam to find products in unknown crowded retail environments, Robotics 10 (4) (2021) 110.
[5] J.P.M. Covolan, A.C. Sementille, S.R.R. Sanches, A mapping of visual SLAM algorithms and their applications in augmented reality, in: 2020 22nd Symposium on Virtual and Augmented Reality, SVR, IEEE, 2020, pp. 20–29.
[6] L. Jinyu, Y. Bangbang, C. Danpeng, W. Nan, Z. Guofeng, B. Hujun, Survey and evaluation of monocular visual-inertial SLAM algorithms for augmented reality, Virtual Real. Intell. Hardw. 1 (4) (2019) 386–410.
[7] R.C. Smith, P. Cheeseman, On the representation and estimation of spatial uncertainty, Int. J. Robot. Res. 5 (4) (1986) 56–68.
[8] J.A. Placed, J. Strader, H. Carrillo, N. Atanasov, V. Indelman, L. Carlone, J.A. Castellanos, A survey on active simultaneous localization and mapping: State of the art and new frontiers, IEEE Trans. Robot. (2023).
[9] N. Sünderhauf, Switchable Constraints for Robust Simultaneous Localization and Mapping and Satellite-Based Localization, vol. 137, Springer Nature, 2023.
[10] T.H. Nguyen, S. Yuan, L. Xie, VR-SLAM: A visual-range simultaneous localization and mapping system using monocular camera and ultra-wideband sensors, 2023, arXiv preprint arXiv:2303.10903.
[11] C. Debeunne, D. Vivet, A review of visual-LiDAR fusion based simultaneous localization and mapping, Sensors 20 (7) (2020) 2068.
[12] S. Arshad, G.-W. Kim, Role of deep learning in loop closure detection for visual and lidar slam: A survey, Sensors 21 (4) (2021) 1243.
[13] D. Van Nam, K. Gon-Woo, Solid-state LiDAR based-SLAM: A concise review and application, in: 2021 IEEE International Conference on Big Data and Smart Computing (BigComp), IEEE, 2021, pp. 302–305.
[14] Y. Liu, J. Miura, RDS-SLAM: Real-time dynamic SLAM using semantic segmentation methods, Ieee Access 9 (2021) 23772–23785.
[15] J. Cheng, L. Zhang, Q. Chen, X. Hu, J. Cai, A review of visual SLAM methods for autonomous driving vehicles, Eng. Appl. Artif. Intell. 114 (2022) 104992.
[16] J. Borenstein, H.R. Everett, L. Feng, D. Wehe, Mobile robot positioning: Sensors and techniques, J. Robot. Syst. 14 (4) (1997) 231–249.
[17] A.J. Davison, I.D. Reid, N.D. Molton, O. Stasse, MonoSLAM: Real-time single camera SLAM, IEEE Trans. Pattern Anal. Mach. Intell. 29 (6) (2007) 1052–1067.
[18] R. Mur-Artal, J.D. Tardós, Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras, IEEE Trans. Robot. 33 (5) (2017) 1255–1262.
[19] K. Huang, S. Zhang, J. Zhang, D. Tao, Event-based simultaneous localization and mapping: A comprehensive survey, 2023, arXiv preprint arXiv:2304.09793.
[20] D. Titterton, J.L. Weston, Strapdown Inertial Navigation Technology, vol. 17, IET, 2004.
[21] W. Jinke, Z. Xingxing, Z. Xiangrui, L. Jiajun, L. Yong, Status and challenges of multi-source fusion SLAM, China J. Image Graph. 27 (02) (2022) 368–389.
[22] A.I. Mourikis, S.I. Roumeliotis, A multi-state constraint Kalman filter for vision-aided inertial navigation, in: Proceedings 2007 IEEE International Conference on Robotics and Automation, IEEE, 2007, pp. 3565–3572.
[23] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, P. Furgale, Keyframe-based visual–inertial odometry using nonlinear optimization, Int. J. Robot. Res. 34 (3) (2015) 314–334.
[24] T. Qin, P. Li, S. Shen, Vins-mono: A robust and versatile monocular visual-inertial state estimator, IEEE Trans. Robot. 34 (4) (2018) 1004–1020.
[25] H. Liu, M. Chen, G. Zhang, H. Bao, Y. Bao, Ice-ba: Incremental, consistent and efficient bundle adjustment for visual-inertial slam, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1974–1982.
[26] K. Konolige, M. Agrawal, J. Sola, Large-scale visual odometry for rough terrain, in: Robotics Research: The 13th International Symposium ISRR, Springer, 2011, pp. 201–212.
[27] J.-P. Tardif, M. George, M. Laverne, A. Kelly, A. Stentz, A new approach to vision-aided inertial navigation, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2010, pp. 4161–4168.
[28] S. Weiss, R. Siegwart, Real-time metric state estimation for modular vision-inertial systems, in: 2011 IEEE International Conference on Robotics and Automation, IEEE, 2011, pp. 4531–4537.
[29] J. Zhang, S. Singh, Visual-lidar odometry and mapping: Low-drift, robust, and fast, in: 2015 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2015, pp. 2174–2181.
[30] M. Camurri, M. Ramezani, S. Nobili, M. Fallon, Pronto: A multi-sensor state estimator for legged robots in real-world scenarios, Front. Robot. AI 7 (2020) 68.
[31] W. Shao, S. Vijayarangan, C. Li, G. Kantor, Stereo visual inertial lidar simultaneous localization and mapping, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2019, pp. 370–377.
[32] K. Liu, X. Zhou, B. Zhao, H. Ou, B.M. Chen, An integrated visual system for unmanned aerial vehicles following ground vehicles: Simulations and experiments, in: 2022 IEEE 17th International Conference on Control & Automation, ICCA, IEEE, 2022, pp. 593–598.
[33] K. Liu, Y. Zhao, Q. Nie, Z. Gao, B.M. Chen, Ws3d supplementary material, in: European Conference on Computer Vision (ECCV). Springer, Cham, 2022, pp. 37–55.
[34] Y. Zhao, L.-M. Po, T. Lin, X. Wang, K. Liu, Y. Zhang, W.-Y. Yu, P. Xian, J. Xiong, Legacy photo editing with learned noise prior, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 2103–2112.
[35] K. Liu, H. Ou, A light-weight lidar-inertial slam system with high efficiency and loop closure detection capacity, in: 2022 International Conference on Advanced Robotics and Mechatronics, ICARM, IEEE, 2022, pp. 284–289.
[36] K. Liu, A lidar-inertial-visual slam system with loop detection, 2023, arXiv preprint arXiv:2301.05604.
[37] Y. Seo, C.-C. Chou, A tight coupling of vision-lidar measurements for an effective odometry, in: 2019 IEEE Intelligent Vehicles Symposium, IV, IEEE, 2019, pp. 1118–1123.
[38] Y.-S. Shin, Y.S. Park, A. Kim, DVL-SLAM: Sparse depth enhanced direct visual-LiDAR SLAM, Auton. Robots 44 (2) (2020) 115–130.
[39] J. Zhang, S. Singh, Visual-lidar odometry and mapping: Low-drift, robust, and fast, in: 2015 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2015, pp. 2174–2181.
[40] J. Graeter, A. Wilczynski, M. Lauer, Limo: Lidar-monocular visual odometry, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2018, pp. 7872–7879.
[41] M. Labbé, F. Michaud, RTAB-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation, J. Field Robot. 36 (2) (2019) 416–446.
[42] W. Zhen, Y. Hu, H. Yu, S. Scherer, LiDAR-enhanced structure-from-motion, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 6773–6779.
[43] Z. Zhang, Z. Zhang, Q. Yu, R. Yi, Y. Xie, L. Ma, LiDAR-camera panoptic segmentation via geometry-consistent and semantic-aware alignment, 2023, arXiv preprint arXiv:2308.01686.
[44] K.J. Wu, C.X. Guo, G. Georgiou, S.I. Roumeliotis, Vins on wheels, in: 2017 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2017, pp. 5155–5162.
[45] M. Zhang, Y. Chen, M. Li, Vision-aided localization for ground robots, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2019, pp. 2455–2461.
[46] J. Liu, W. Gao, Z. Hu, Visual-inertial odometry tightly coupled with wheel encoder adopting robust initialization and online extrinsic calibration, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2019, pp. 5391–5397.
[47] W. Lee, K. Eckenhoff, Y. Yang, P. Geneva, G. Huang, Visual-inertial-wheel odometry with online calibration, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2020, pp. 4559–4566.

[48] F. Zheng, Y.-H. Liu, Visual-odometric localization and mapping for ground vehicles using SE (2)-XYZ constraints, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 3556–3562.

[49] X.Z. Zhu Daojun, M. Tingting, C. Pingguo, S. Quanjun, Research on the localization algorithm of wheeled robot based on tight coupling of vision and wheel speedometer, J. Sens. Technol. (2021).

[50] K.-W. Chiang, D.T. Le, T.T. Duong, R. Sun, The performance analysis of INS/GNSS/V-SLAM integration scheme using smartphone sensors for land vehicle navigation applications in GNSS-challenging environments, Remote Sens. 12 (11) (2020) 1732.

[51] Z. Niu, X. Zhao, J. Sun, L. Tao, B. Zhu, A continuous positioning algorithm based on RTK and VI-SLAM with smartphones, IEEE Access 8 (2020) 185638–185650.

[52] J. Cremona, J. Civera, E. Kofman, T. Pire, GNSS-stereo-inertial SLAM for arable farming, J. Field Robotics (2023).

[53] A. Arun, R. Ayyalasomayajula, W. Hunter, D. Bharadia, P2slam: Bearing based wifi slam for indoor robots, IEEE Robot. Autom. Lett. 7 (2) (2022) 3326–3333.

[54] K. Ismail, R. Liu, Z. Qin, A. Athukorala, B.P.L. Lau, M. Shalihan, C. Yuen, U.-X. Tan, Efficient WiFi LiDAR SLAM for autonomous robots in large environments, in: 2022 IEEE 18th International Conference on Automation Science and Engineering, CASE, IEEE, 2022, pp. 1132–1137.

[55] J. Xu, H. Cao, D. Li, K. Huang, C. Qian, L. Shangguan, Z. Yang, Edge assisted mobile semantic visual slam, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 1828–1837.

[56] C. Theodorou, V. Velisavljevic, V. Dyo, Visual SLAM for dynamic environments based on object detection and optical flow for dynamic object removal, Sensors 22 (19) (2022) 7553.

[57] P. Su, S. Luo, X. Huang, Real-time dynamic SLAM algorithm based on deep learning, IEEE Access 10 (2022) 87754–87766.

[58] Z. Zhu, S. Peng, V. Larsson, Z. Cui, M.R. Oswald, A. Geiger, M. Pollefeys, Nicer-slam: Neural implicit scene encoding for rgb slam, 2023, arXiv preprint arXiv:2302.03594.

[59] Y. Liu, J. Miura, RDMO-SLAM: Real-time visual SLAM for dynamic environments using semantic label prediction with optical flow, IEEE Access 9 (2021) 106981–106997.

[60] C. Harris, M. Stephens, et al., A combined corner and edge detector, in: Alvey Vision Conference, Vol. 15, Citeseer, 1988, pp. 10–5244.

[61] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2004) 91–110.

[62] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (SURF), Comput. Vis. Image Underst. 110 (3) (2008) 346–359.

[63] D.G. Viswanathan, Features from accelerated segment test (fast), in: Proceedings of the 10th Workshop on Image Analysis for Multimedia Interactive Services, London, UK, 2009, pp. 6–8.

[64] M. Calonder, V. Lepetit, C. Strecha, P. Fua, Brief: Binary robust independent elementary features, in: Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11, Springer, 2010, pp. 778–792.

[65] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: An efficient alternative to SIFT or SURF, in: 2011 International Conference on Computer Vision, Ieee, 2011, pp. 2564–2571.

[66] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. (6) (1986) 679–698.

[67] J. Shi, et al., Good features to track, in: 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 1994, pp. 593–600.

[68] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'05, Vol. 1, Ieee, 2005, pp. 886–893.

[69] R.G. Von Gioi, J. Jakubowicz, J.-M. Morel, G. Randall, LSD: A fast line segment detector with a false detection control, IEEE Trans. Pattern Anal. Mach. Intell. 32 (4) (2008) 722–732.

[70] S. Leutenegger, M. Chli, R.Y. Siegwart, BRISK: Binary robust invariant scalable keypoints, in: 2011 International Conference on Computer Vision, Ieee, 2011, pp. 2548–2555.

[71] P.F. Alcantarilla, A. Bartoli, A.J. Davison, KAZE features, in: Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12, Springer, 2012, pp. 214–227.

[72] H. Strasdat, J. Montiel, A.J. Davison, Scale drift-aware large scale monocular SLAM, in: Robotics: Science and Systems, Vol. 2, 2010, p. 5.

[73] J. Engel, V. Koltun, D. Cremers, Direct sparse odometry, IEEE Trans. Pattern Anal. Mach. Intell. 40 (3) (2017) 611–625.

[74] J. Engel, T. Schöps, D. Cremers, LSD-SLAM: Large-scale direct monocular SLAM, in: European Conference on Computer Vision, Springer, 2014, pp. 834–849.

[75] R.A. Newcombe, S.J. Lovegrove, A.J. Davison, DTAM: Dense tracking and mapping in real-time, in: 2011 International Conference on Computer Vision, IEEE, 2011, pp. 2320–2327.

[76] C. Forster, M. Pizzoli, D. Scaramuzza, SVO: Fast semi-direct monocular visual odometry, in: 2014 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2014, pp. 15–22.

[77] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, P. Furgale, Keyframe-based visual–inertial odometry using nonlinear optimization, Int. J. Robot. Res. 34 (3) (2015) 314–334.

[78] R. Mur-Artal, J.D. Tardós, Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras, IEEE Trans. Robot. 33 (5) (2017) 1255–1262.

[79] D. DeTone, T. Malisiewicz, A. Rabinovich, Superpoint: Self-supervised interest point detection and description, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 224–236.

[80] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, F. Moreno-Noguer, PL-SLAM: Real-time monocular visual SLAM with points and lines, in: 2017 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2017, pp. 4503–4508.

[81] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, Q. Fei, DS-SLAM: A semantic visual SLAM towards dynamic environments, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2018, pp. 1168–1174.

[82] B. Bescós, J.M. Fácil, J. Civera, J. Neira, Dynaslam: Tracking, mapping, and inpainting in dynamic scenes, Comput. Sci. 3 (4) (2018).

[83] J. Tang, L. Ericson, J. Folkesson, P. Jensfelt, GCNv2: Efficient correspondence prediction for real-time SLAM, IEEE Robot. Autom. Lett. 4 (4) (2019) 3505–3512.

[84] S.Y. Loo, A.J. Amiri, S. Mashohor, S.H. Tang, H. Zhang, CNN-SVO: Improving the mapping in semi-direct visual odometry using single-image depth prediction, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 5218–5223.

[85] S. Paul-Edouard, D. Daniel, M. Tomasz, R. Andrew, Superglue: Learning feature matching with graph neural networks, in: Computer Vision and Pattern Recognition, 2020, pp. 4937–4946.

[86] J. Sun, Z. Shen, Y. Wang, H. Bao, X. Zhou, LoFTR: Detector-free local feature matching with transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8922–8931.

[87] C. Campos, R. Elvira, J.J.G. Rodríguez, J.M. Montiel, J.D. Tardós, Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam, IEEE Trans. Robot. 37 (6) (2021) 1874–1890.

[88] Y. Diao, R. Cen, F. Xue, X. Su, Orb-slam2s: A fast orb-slam2 system with sparse optical flow tracking, in: 2021 13th International Conference on Advanced Computational Intelligence, ICACI, IEEE, 2021, pp. 160–165.

[89] Q. Fu, H. Yu, X. Wang, Z. Yang, Y. He, H. Zhang, A. Mian, Fast ORB-SLAM without keypoint descriptors, IEEE Trans. Image Process. 31 (2021) 1433–1446.

[90] Y. Fu, B. Han, Z. Hu, X. Shen, Y. Zhao, CBAM-SLAM: A semantic SLAM based on attention module in dynamic environment, in: 2022 6th Asian Conference on Artificial Intelligence Technology, ACAIT, IEEE, 2022, pp. 1–6.

[91] Y. You, P. Wei, J. Cai, W. Huang, R. Kang, H. Liu, et al., MISD-SLAM: multimodal semantic SLAM for dynamic environments, Wirel. Commun. Mob. Comput. 2022 (2022).

[92] D. Cai, Z. Hu, R. Li, h. Qi, y. Xiang, Y. Zhao, AGAM-SLAM: An adaptive dynamic scene semantic SLAM method based on GAM, in: 2023 International Conference on Intelligent Computing, IEEE, 2022, pp. 1–6.

[93] H. Qi, Z. Hu, Y. Xiang, D. Cai, Y. Zhao, ATY-SLAM: A visual semantic SLAM for dynamic indoor environments, in: 2023 International Conference on Intelligent Computing, 2023.

[94] P. Lindenberger, P.-E. Sarlin, M. Pollefeys, LightGlue: Local feature matching at light speed, 2023, arXiv preprint arXiv:2306.13643.

[95] R. Li, Y. Zhao, Z. Hu, W. Qi, G. Liu, TOHF: A feature extractor for resource-constrained indoor VSLAM, J. Syst. Simul. (2024).

[96] S. Guo, Z. Rong, S. Wang, Y. Wu, A LiDAR SLAM with PCA-based feature extraction and two-stage matching, IEEE Trans. Instrum. Meas. 71 (2022) 1–11.

[97] K. Tateno, F. Tombari, I. Laina, N. Navab, Cnn-slam: Real-time dense monocular slam with learned depth prediction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6243–6252.

[98] S.M.R. U., P.B. de Gusmao Pedro, L.C. Xiaoxuan, A. Yasin, R. Stefano, C. Changhao, W. Johan, W. Wei, M. Andrew, T. Niki, DeepTIO: A deep thermal-inertial odometry with visual hallucination, in: IEEE International Conference on Robotics and Automation, 2019, pp. 1672–1679.

[99] J. Yin, D. Luo, F. Yan, Y. Zhuang, A novel lidar-assisted monocular visual SLAM framework for mobile robots in outdoor environments, IEEE Trans. Instrum. Meas. 71 (2022) 1–11.

[100] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, T. Sattler, D2-Net: A trainable CNN for joint detection and description of local features., in: Computer Vision and Pattern Recognition, 2019, pp. 8092–8101.

[101] S. Woo, J. Park, J.-Y. Lee, I.S. Kweon, Cbam: Convolutional block attention module, in: European Conference on Computer Vision, 2018, pp. 3–19.

[102] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969.

[103] Y. Liu, Z. Shao, N.a. Hoffmann, Global attention mechanism: Retain information to enhance channel-spatial interactions, 2021.

[104] S. Milz, G. Arbeiter, C. Witt, B. Abdallah, S. Yogamani, Visual slam for automated driving: Exploring the applications of deep learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 247–257.

[105] M.R.U. Saputra, A. Markham, N. Trigoni, Visual SLAM and structure from motion in dynamic environments: A survey, ACM Comput. Surv. 51 (2) (2018) 1–36.

[106] L. Jing, Y. Tian, Self-supervised visual feature learning with deep neural networks: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 43 (11) (2020) 4037–4058.

[107] H. Taheri, Z.C. Xia, SLAM; definition and evolution, Eng. Appl. Artif. Intell. 97 (2021) 104032.

[108] S. Li, D. Zhang, Y. Xian, B. Li, T. Zhang, C. Zhong, Overview of deep learning application on visual SLAM, Displays (2022) 102298.

[109] H. Pu, J. Luo, G. Wang, T. Huang, H. Liu, Visual SLAM integration with semantic segmentation and deep learning: A review, IEEE Sens. J. (2023).

[110] G.A. Einicke, L.B. White, Robust extended Kalman filtering, IEEE Trans. Signal Process. 47 (9) (1999) 2596–2599.

[111] M. Skoglund, Z. Sjanic, F. Gustafsson, Initialisation and estimation methods for batch optimisation of inertial/visual SLAM, 2013.

[112] Å. Björck, Least squares methods, Handb. Numer. Anal. 1 (1990) 465–652.

[113] T. Fan, J. Ortiz, M. Hsiao, M. Monge, J. Dong, T. Murphey, M. Mukadam, Decentralization and acceleration enables large-scale bundle adjustment, 2023, arXiv preprint arXiv:2305.07026.

[114] L. Han, L. Xu, D. Bobkov, E. Steinbach, L. Fang, Real-time global registration for globally consistent rgb-d slam, IEEE Trans. Robot. 35 (2) (2019) 498–508.

[115] A. Stark, S. Sunyaev, R.B. Russell, A model for statistical significance of local similarities in structure, J. Mol. Biol. 326 (5) (2003) 1307–1316.

[116] Y. Akimoto, Monotone improvement of information-geometric optimization algorithms with a surrogate function, in: Proceedings of the Genetic and Evolutionary Computation Conference, 2022, pp. 1354–1362.

[117] G. Lu, Deep unsupervised visual odometry via bundle adjusted pose graph optimization, in: 2023 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2023, pp. 6131–6137.

[118] S. Wei, S. Wang, Matching filter-based vslam optimization in indoor environments, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. 48 (2023) 679–685.

[119] Z. Wang, B. Pang, Y. Song, X. Yuan, Q. Xu, Y. Li, Robust visual-inertial odometry based on a Kalman filter and factor graph, IEEE Trans. Intell. Transp. Syst. (2023).

[120] M. Abdollahi, S.H. Pourtakdoust, M. Nooshabadi, H. Pishkenari, An improved multi-state constraint Kalman filter for visual-inertial odometry, 2022, arXiv preprint arXiv:2210.08117.

[121] P. van Goor, R. Mahony, EqVIO: An equivariant filter for visual-inertial odometry, IEEE Trans. Robot. (2023).

[122] B. Triggs, P.F. McLauchlan, R.I. Hartley, A.W. Fitzgibbon, Bundle adjustment—a modern synthesis, in: Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings, Springer, 2000, pp. 298–372.

[123] B.M. Haralick, C.-N. Lee, K. Ottenberg, M. Nölle, Review and analysis of solutions of the three point perspective pose estimation problem, Int. J. Comput. Vis. 13 (1994) 331–356.

[124] D. Ming, X. Wu, Y. Wang, Z. Zhu, H. Ge, R. Liu, A real-time monocular visual SLAM based on the bundle adjustment with adaptive robust kernel, J. Intell. Robot. Syst. 107 (3) (2023) 35.

[125] K. Wang, S. Ma, F. Ren, J. Lu, SBAS: Salient bundle adjustment for visual SLAM, IEEE Trans. Instrum. Meas. 70 (2021) 1–9.

[126] L. Euler, Solutio problematis ad geometriam situs pertinentis, Comment. Acad. Sci. Petropolitanae (1741) 128–140.

[127] M. Abdollahi, S.H. Pourtakdoust, M. Nooshabadi, H. Pishkenari, An improved multi-state constraint Kalman filter for visual-inertial odometry, 2022, arXiv preprint arXiv:2210.08117.

[128] K. Wang, S. Ma, F. Ren, J. Lu, SBAS: Salient bundle adjustment for visual SLAM, IEEE Trans. Instrum. Meas. 70 (2021) 1–9.

[129] D. Ming, X. Wu, Y. Wang, Z. Zhu, H. Ge, R. Liu, A real-time monocular visual SLAM based on the bundle adjustment with adaptive robust kernel, J. Intell. Robot. Syst. 107 (3) (2023) 35.

[130] A. Angeli, S. Doncieux, J.-A. Meyer, D. Filliat, Real-time visual loop-closure detection, in: 2008 IEEE International Conference on Robotics and Automation, IEEE, 2008, pp. 1842–1847.

[131] D. Gálvez-López, J. Tardós, Dbow2: Enhanced hierarchical bag-of-word library for C++, 2012.

[132] R. Elvira, J.D. Tardós, J.M. Montiel, ORBSLAM-Atlas: a robust and accurate multi-map system, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2019, pp. 6253–6259.

[133] S. Wang, D. Zheng, Y. Li, LiDAR-SLAM loop closure detection based on multi-scale point cloud feature transformer, Meas. Sci. Technol. 35 (3) (2023) 036305.

[134] L. Yang, Z. Yu, C. Deng, et al., Three-dimensional lidar localization and mapping with loop-closure detection based on dense depth information, Mathematics 11 (9) (2023) 2211.

[135] S. An, H. Zhu, D. Wei, K.A. Tsintotas, A. Gasteratos, Fast and incremental loop closure detection with deep features and proximity graphs, J. Field Robotics 39 (4) (2022) 473–493.

[136] H. Osman, N. Darwish, A. Bayoumi, PlaceNet: A multi-scale semantic-aware model for visual loop closure detection, Eng. Appl. Artif. Intell. 119 (2023) 105797.

[137] Z. Yuan, K. Xu, X. Zhou, B. Deng, Y. Ma, SVG-Loop: Semantic–visual–geometric information-based loop closure detection, Remote Sens. 13 (17) (2021) 3520.

[138] J. Li, P. Wang, C. Ni, D. Zhang, W. Hao, Loop closure detection for mobile robot based on multidimensional image feature fusion, Machines 11 (1) (2022) 16.

[139] J.-S. Gutmann, K. Konolige, Incremental mapping of large cyclic environments, in: Proceedings 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation. CIRA'99 (Cat. No. 99EX375), IEEE, 1999, pp. 318–325.

[140] R. Mur-Artal, J.D. Tardós, Fast relocalisation and loop closing in keyframe-based SLAM, in: 2014 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2014, pp. 846–853.

[141] J. Wan, A. Yilmaz, L. Yan, Dcf-bow: Build match graph using bag of deep convolutional features for structure from motion, IEEE Geosci. Remote Sens. Lett. 15 (12) (2018) 1847–1851.

[142] Z. Li, Y. Zhou, R. Bao, An image classification method based on optimized fuzzy bag-of-words model., Trait. Signal 36 (3) (2019).

[143] M. Ferrera, A. Eudes, J. Moras, M. Sanfourche, G. Le Besnerais, OV^2– SLAM: A fully online and versatile visual SLAM for real-time applications, IEEE Robot. Autom. Lett. 6 (2) (2021) 1399–1406.

[144] K.A. Tsintotas, L. Bampis, A. Gasteratos, Modest-vocabulary loop-closure detection with incremental bag of tracked words, Robot. Auton. Syst. 141 (2021) 103782.

[145] M. Gehrig, E. Stumm, T. Hinzmann, R. Siegwart, Visual place recognition with probabilistic voting, in: 2017 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2017, pp. 3192–3199.

[146] X. Shen, L. Chen, Z. Hu, Y. Fu, H. Qi, Y. Xiang, J. Wu, A closed-loop detection algorithm for online updating of bag-of-words model, in: Proceedings of the 2023 9th International Conference on Computing and Data Engineering, 2023, pp. 34–40.

[147] H. Durrant-Whyte, T. Bailey, Simultaneous localization and mapping: Part I, IEEE Robot. Autom. Mag. 13 (2) (2006) 99–110.

[148] Y. Zhou, Y. Wang, F. Poiesi, Q. Qin, Y. Wan, Loop closure detection using local 3D deep descriptors, IEEE Robot. Autom. Lett. 7 (3) (2022) 6335–6342.

[149] S. Gupta, R. Girshick, P. Arbeláez, J. Malik, Learning rich features from RGB-D images for object detection and segmentation, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13, Springer, 2014, pp. 345–360.

[150] Z. Wang, Z. Peng, Y. Guan, L. Wu, Two-stage vSLAM loop closure detection based on sequence node matching and semi-semantic autoencoder, J. Intell. Robot. Syst. 101 (2021) 1–21.

[151] Y. Wang, Y. Zhang, L. Hu, W. Wang, G. Ge, S. Tan, A semantic topology graph to detect re-localization and loop closure of the visual simultaneous localization and mapping system in a dynamic environment, Sensors 23 (20) (2023) 8445.

[152] W.W. Fang German, L. Natural, H. Xiyan, P. Lord, VIO-SLAM overview, Electr. Opt. Control 27 (12) (2020) 58–62.

[153] R. Mur-Artal, J.M.M. Montiel, J.D. Tardos, ORB-SLAM: a versatile and accurate monocular SLAM system, IEEE Trans. Robot. 31 (5) (2015) 1147–1163.

[154] G. Klein, D. Murray, Parallel tracking and mapping for small AR workspaces, in: 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, IEEE, 2007, pp. 225–234.

[155] R.A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohi, J. Shotton, S. Hodges, A. Fitzgibbon, Kinectfusion: Real-time dense surface mapping and tracking, in: 2011 10th IEEE International Symposium on Mixed and Augmented Reality, Ieee, 2011, pp. 127–136.

[156] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, A. Davison, ElasticFusion: Dense SLAM without a pose graph, Robot.: Sci. Syst. (2015).

[157] B. Balasuriya, B. Chathuranga, B. Jayasundara, N. Napagoda, S. Kumarawadu, D. Chandima, A. Jayasekara, Outdoor robot navigation using Gmapping based SLAM algorithm, in: 2016 Moratuwa Engineering Research Conference (MERCon), IEEE, 2016, pp. 403–408.

[158] M. Labbé, F. Michaud, RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation, J. Field Robot. 36 (2) (2019) 416–446.

[159] S. Yang, S. Scherer, Cubeslam: Monocular 3-d object slam, IEEE Trans. Robot. 35 (4) (2019) 925–938.

[160] L. Nicholson, M. Milford, N. Sünderhauf, Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam, IEEE Robot. Autom. Lett. 4 (1) (2018) 1–8.

[161] X. Gao, T. Zhang, Y. Liu, Q. Yan, 14 Lectures on Visual SLAM: From Theory to Practice, Publishing House of Electronics Industry, 2017.

[162] Z. Yang, D. Shi, Mapping technology in visual slam: A review, in: Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence, 2018, pp. 291–295.
[163] R.C. Smith, P. Cheeseman, On the representation and estimation of spatial uncertainty, Int. J. Robot. Res. 5 (4) (1986) 56–68.
[164] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, et al., Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges, in: IJCAI, Vol. 3, 2003, pp. 1151–1156.
[165] A. Bokovoy, K. Yakovlev, Sparse 3D point-cloud map upsampling and noise removal as a vslam post-processing step: Experimental evaluation, in: Interactive Collaborative Robotics: Third International Conference, ICR 2018, Leipzig, Germany, September 18–22, 2018, Proceedings 3, Springer, 2018, pp. 23–33.
[166] D. Lyu, Z. Li, Y. Chen, J. Zhang, N. Xu, G. He, SpOctA: A 3D sparse convolution accelerator with octree-encoding-based map search and inherent sparsity-aware processing, 2023, arXiv preprint arXiv:2308.09249.
[167] H. Xie, D. Zhang, J. Wang, M. Zhou, Z. Cao, X. Hu, A. Abusorrah, Semi-direct multimap SLAM system for real-time sparse 3-D map reconstruction, IEEE Trans. Instrum. Meas. 72 (2023) 1–13.
[168] S. Wen, X. Liu, H. Zhang, F. Sun, M. Sheng, S. Fan, Dense point cloud map construction based on stereo VINS for mobile vehicles, ISPRS J. Photogramm. Remote Sens. 178 (2021) 328–344.
[169] J. Engel, J. Sturm, D. Cremers, Semi-dense visual odometry for a monocular camera, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1449–1456.
[170] S. Jia, H. Yu, IDMC-VSLAM: Improved dense map construction and visual SLAM in dynamic environments, in: 2021 2nd International Conference on Computer Engineering and Intelligent Control, ICCEIC, IEEE, 2021, pp. 81–85.
[171] A. Elfes, Using occupancy grids for mobile robot perception and navigation, Computer 22 (6) (1989) 46–57.
[172] H. Jang, J. Cho, S. Heo, et al., Performance improvement of deep learning object detection method using dynamic occupancy grid map, Trans. Korean Soc. Automot. Eng. 30 (10) (2022) 839–847.
[173] A. Fisher, R. Cannizzaro, M. Cochrane, et al., ColMap: A memory-efficient occupancy grid mapping framework, Robot. Auton. Syst. 142 (2021) 103755.
[174] N.A. Yatim, N. Buniyamin, Z.M. Noh, et al., Occupancy grid map algorithm with neural network using array of infrared sensors, in: Journal of Physics: Conference Series, Vol. 1502, IOP Publishing, 2020, 012053.
[175] S. Thrun, Probabilistic robotics, Commun. ACM 45 (3) (2002) 52–57.
[176] R.A. Finkel, J.L. Bentley, Quad trees a data structure for retrieval on composite keys, Acta Inform. 4 (1974) 1–9.
[177] H. Samet, The quadtree and related hierarchical data structures, ACM Comput. Surv. 16 (2) (1984) 187–260.
[178] E. Vespa, N. Nikolov, M. Grimm, L. Nardi, P.H. Kelly, S. Leutenegger, Efficient octree-based volumetric SLAM supporting signed-distance and occupancy mapping, IEEE Robot. Autom. Lett. 3 (2) (2018) 1144–1151.
[179] A. Hornung, K.M. Wurm, M. Bennewitz, C. Stachniss, W. Burgard, OctoMap: An efficient probabilistic 3D mapping framework based on octrees, Auton. Robots 34 (2013) 189–206.
[180] R.F. Salas-Moreno, R.A. Newcombe, H. Strasdat, P.H. Kelly, A.J. Davison, Slam++: Simultaneous localisation and mapping at the level of objects, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 1352–1359.
[181] J. McCormac, A. Handa, A. Davison, S. Leutenegger, Semanticfusion: Dense 3d semantic mapping with convolutional neural networks, in: 2017 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2017, pp. 4628–4635.
[182] A. Sharma, W. Dong, M. Kaess, Compositional and scalable object slam, in: 2021 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2021, pp. 11626–11632.
[183] K. Tateno, F. Tombari, I. Laina, N. Navab, Cnn-slam: Real-time dense monocular slam with learned depth prediction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6243–6252.
[184] J. McCormac, R. Clark, M. Bloesch, A. Davison, S. Leutenegger, Fusion++: Volumetric object-level slam, in: 2018 International Conference on 3D Vision (3DV), IEEE, 2018, pp. 32–41.
[185] I.A. Bârsan, P. Liu, M. Pollefeys, A. Geiger, Robust dense mapping for large-scale dynamic environments, in: 2018 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2018, pp. 7510–7517.
[186] F. Zhong, S. Wang, Z. Zhang, Y. Wang, Detect-SLAM: Making object detection and SLAM mutually beneficial, in: 2018 IEEE Winter Conference on Applications of Computer Vision, WACV, IEEE, 2018, pp. 1001–1010.
[187] M. Runz, M. Buffier, L. Agapito, Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects, in: 2018 IEEE International Symposium on Mixed and Augmented Reality, ISMAR, IEEE, 2018, pp. 10–20.
[188] Z. Qian, K. Patath, J. Fu, J. Xiao, Semantic slam with autonomous object-level data association, in: 2021 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2021, pp. 11203–11209.
[189] J. Zhang, M. Gui, Q. Wang, R. Liu, J. Xu, S. Chen, Hierarchical topic model based object association for semantic SLAM, IEEE Trans. Vis. Comput. Graph. 25 (11) (2019) 3052–3062.
[190] M. Hosseinzadeh, K. Li, Y. Latif, I. Reid, Real-time monocular object-model aware sparse SLAM, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 7123–7129.
[191] Y. Wu, Y. Zhang, D. Zhu, Y. Feng, S. Coleman, D. Kerr, Eao-slam: Monocular semi-dense object slam based on ensemble data association, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2020, pp. 4966–4973.
[192] B. Li, D. Zou, D. Sartori, L. Pei, W. Yu, Textslam: Visual slam with planar text features, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 2102–2108.
[193] K. Chen, J. Liu, Q. Chen, Z. Wang, J. Zhang, Accurate object association and pose updating for semantic slam, IEEE Trans. Intell. Transp. Syst. 23 (12) (2022) 25169–25179.
[194] K. Chen, J. Zhang, J. Liu, Q. Tong, R. Liu, S. Chen, Semantic visual simultaneous localization and mapping: A survey, 2022, arXiv preprint arXiv:2209.06428.

Dupeng Cai obtained his B.S. degree in Communication Engineering from Henan University of Engineering in 2022. He is currently pursuing his Master's degree in the School of Information and Communication Engineering at Hainan University, China, with a primary research focus on Visual SLAM.

Ruoqing Li obtained her B.S. degree in Network Engineering from the School of Computer and Cybersecurity at Hainan University in 2020. She is currently pursuing her Master's degree in the School of Cybersecurity at Hainan University, China, with a primary research focus on Visual SLAM.

Zhuhua Hu (IEEE Senior Member) received the BEng degree from the Jilin University in 2002, the MEng degree from the Jilin University in 2005, and the Ph.D. degree from the Hainan University in 2019. He was a software engineer at Ningbo BIRD Research Institute of China from 2005 to 2006. He was a software engineer at Nanjing Research Institute of ZTE from 2006 to 2007. He was a minister of the software department at Shanghai Aoxun Information Technology Co., Ltd. from 2007 to 2009. He is a professor at the School of Information and Communication Engineering, Hainan University since 2020. He is IEEE Senior Member and CCF Senior Member. He is currently a high-level talent in Hainan Province. He acted as the Reviewer for IEEE IoT Journal, EAAI, TETC, KBS, Remote Sensing, IEEE-ACM TRANSACTIONS ON NETWORKING, Applied Artificial Intelligence, ICASSP2023. His current research interests include artificial intelligence, signal and information processing.

Jinlin Lu obtained his B.S. degree in Communication Engineering from Hainan University in 2021. He is currently pursuing his Master's degree in the School of Information and Communication Engineering at Hainan University, China, with a primary research focus on Visual SLAM.

Shijiang Li obtained his B.S. degree in Electronic Information from Southwest University of Science and Technology in 2022. He is currently pursuing his Master's degree in the School of Information and Communication Engineering at Hainan University, China, with a primary research focus on Visual SLAM.

Yaochi Zhao received her M.S. degree in pattern recognition and intelligent system from Central South University, Changsha, China, in 2005, and she is pursuing a Ph.D. degree at Tianjin University. She worked in Ningbo BIRD Research Institute and Shanghai Wingtech Communication Co., Ltd. for three years. Later, she was engaged in teaching and research work at the College of Information Science & Technology, Hainan University, Haikou, China. Now, she is an associate professor at the school of computer and cyberspace security at Hainan University. Her current research interests include computer vision and artificial intelligence.