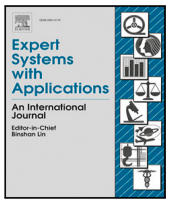
Expert Systems With Applications 289 (2025) 127574

Contents lists available at ScienceDirect

Expert Systems With Applications

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



Review

Point-line feature-based vSLAM systems: A survey

Hangzhou Qu [a,b], Zhuhua Hu [a,b,*], Yaochi Zhao [c], Junlin Lu [a,b], Kunkun Ding [a,b], Guangfeng Liu [a,b], Yongqing Chen [a,b], Chunyan Shao [a,b]

[a] School of Information and Communication Engineering, Hainan University, Haikou, 570228, China
[b] State Key Laboratory of Marine Resource Utilization in South China Sea, Hainan University, Haikou, 570228, China
[c] School of Cyberspace Security (School of Cryptology), Hainan University, Haikou 570228, China

ARTICLE INFO

Keywords:
vSLAM
Line feature matching
Backend optimization
Loop closure detection

ABSTRACT

The point-line feature-based vSLAM technology significantly enhances the accuracy and robustness of localization and mapping in complex environments by comprehensively utilizing both point and line geometric information. This paper provides a comprehensive survey of methods and applications for point-line feature-based Simultaneous Localization and Mapping (SLAM) systems. Firstly, it focuses on the core components of the visual frontend in SLAM systems, with a detailed analysis of line feature detection methods and their descriptors, covering both traditional algorithms and learning-based approaches, as well as further improvements to these methods. The paper also discusses several common line feature parameterization methods and different line feature matching strategies. In addition, the paper delves into the backend optimization and loop closure detection mechanisms of SLAM systems, which are critical factors in enhancing the system's accuracy and robustness. By reviewing these methods and applications, this paper aims to provide a comprehensive understanding of integrated point-line SLAM systems, analyzing the strengths and weaknesses of different technologies, and exploring potential directions for future research. This work offers theoretical foundations and practical guidance from a global perspective for the subsequent design and optimization of SLAM systems.

Contents

* Corresponding author at: School of Information and Communication Engineering, Hainan University, Haikou, 570228, China.
E-mail addresses: 22120854000008@hainanu.edu.cn (H. Qu), eagler_hu@hainanu.edu.cn (Z. Hu), zhyc@hainanu.edu.cn (Y. Zhao), lujunlin762@gmail.com (J. Lu), dky254926@163.com (K. Ding), a912034094@163.com (G. Liu), 20081000110019@hainanu.edu.cn (Y. Chen), shaochunyan@hainanu.edu.cn (C. Shao).

https://doi.org/10.1016/j.eswa.2025.127574
Received 14 October 2024; Received in revised form 29 March 2025; Accepted 2 April 2025
Available online 30 May 2025

1. Introduction

Simultaneous Localization and Mapping (SLAM) technology enables robots to achieve self-localization and construct maps of their surrounding environment using sensor data in unknown environments. The concept of SLAM, first introduced by Smith in 1986 (Smith & Cheeseman, 1986), was aimed at discussing the representation and estimation of spatial uncertainty. With the rapid advancements in mobile robotics (Clemens, Kluth, & Reineking, 2019; Guan, Ristic, Wang, & Palmer, 2019; Yuan, Li, & Su, 2019), autonomous driving (Fern et al., 2021; Henein, Zhang, Mahony, & Ila, 2020), and augmented reality (AR) (Moezzi, Krcmarik, Hlava, & Cýrus, 2020; Rohacz, Weißenfels, & Strassburger, 2020) technologies, SLAM has be-come a core technology in these fields. Continuous research and improvements have led to significant progress in the accuracy, robustness, and real-time performance of SLAM, propelling its widespread application in various scenarios (Cadena et al., 2016).

From a technical development perspective, SLAM has undergone three major stages (Kazerouni, Fitzgerald, Dooly, & Toal, 2022). Initially, SLAM was based on traditional filtering methods, with representatives such as the early EKF-SLAM (Leonard & Durrant-Whyte, 1991) and FastSLAM (Montemerlo, 2002). As sensor technology, graphics processing, and computational capabilities significantly advanced, SLAM entered a new era centered on algorithmic analysis. Vision-based SLAM methods developed and gained widespread application, primarily using cameras as sensors to estimate robot trajectories and construct environmental maps by analyzing consecutive images. The low cost and the ability to capture rich environmental information have greatly accelerated the development of SLAM technology. The latest stage of SLAM's development focuses on feature fusion. Current systems not only leverage point features but also integrate line, plane, and other features along with Inertial Measurement Unit (IMU) data. This comprehensive feature fusion enables SLAM systems to provide more accurate and stable localization and mapping capabilities in complex indoor and outdoor environments.

Visual SLAM Early SLAM systems primarily processed visual sensor data using feature-based methods, such as the Extended Kalman Filter (EKF) and Particle Filter (PF) techniques. Among them, the most representative is the monocular vision-based MonoSLAM system (Davison, Reid, Molton, & Stasse, 2007), which was based on the EKF and performed well in small-scale environments. However, as the environment's scale increased, its performance and efficiency gradually decreased, making real-time processing challenging. To address this issue, Klein and Murray innovatively proposed the Parallel Tracking and Mapping (PTAM) system (Klein & Murray, 2007, 2008, 2009), which is based on keyframes to optimize SLAM performance. This system relies on FAST corner points (Rosten & Drummond, 2006) to separate tracking and mapping into two parallel threads and uses keyframes to estimate the map (Mouragnon, Lhuillier, Dhome, Dekeyser, & Sayd, 2009), significantly improving processing speed and accuracy. Additionally, the system employs Bundle Adjustment (BA) optimization to achieve more accurate results than filtering techniques. As technology advanced, many real-time visual SLAM methods were subsequently proposed, further improving feature-based SLAM approaches. The ORB-SLAM (Mur-Artal, Montiel, & Tardos, 2015) system was introduced, utilizing Oriented FAST and Rotated BRIEF (ORB) (Rublee, Rabaud, Konolige, & Bradski, 2011) features and Binary Robust Independent Elementary Features (BRIEF) descriptors (Calonder, Lepetit, Strecha, & Fua, 2010) for data association. It follows the framework of Klein and Murray (2007) and implements real-time pose estimation using three parallel threads for tracking, local mapping, and loop closure, achieving more robust camera tracking and map construction. Based on these advancements, direct visual SLAM systems have also emerged, omitting the feature extraction and computation process, thereby enhancing the real-time performance of visual SLAM algorithms. Direct methods rely on the assumption of photometric consistency and recover poses by minimizing photometric errors. Representative studies include the LSD-SLAM system and its optimizations (Caruso, Engel, & Cremers, 2015; Engel, Schöps, & Cremers, 2014; Engel, Stückler, & Cremers, 2015), the SVO system (Forster, Pizzoli, & Scaramuzza, 2014), and the DSO system (Engel, Koltun, & Cremers, 2017). To address the issue of cumulative error in direct visual SLAM systems, improvements were made in studies (Gao, Wang, Demmel, & Cremers, 2018; Lee & Civera, 2018) to the DSO system by incorporating loop closure functionality. Furthermore, the DSM system (Zubizarreta, Aguinaga, & Montiel, 2020) introduced the concept of map reuse, further enhancing the mapping capabilities of direct SLAM.

In recent years, the rapid development of deep learning has significantly impacted visual SLAM research. Wang, Clark, Wen, and Trigoni (2017) proposed an end-to-end visual odometry system named DeepVO, which integrates Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), substantially improving pose estimation accuracy. Furthermore, (Almalioglu, Saputra, De Gusmao, Markham, & Trigoni, 2019; Zhou, Zhang, Shen, & Jia, 2017) optimized the application of deep learning methods in visual SLAM, enhancing its reliability in dynamic scenes. Another approach is the CNN-SLAM system (Tateno, Tombari, Laina, & Navab, 2017), which utilizes CNNs for depth prediction, providing absolute scale information for monocular SLAM systems. The method presented in Bruno and Colombini

(2021), using the LIFT network (Yi, Trulls, Lepetit, & Fua, 2016) for feature extraction in conjunction with the traditional geometric approach of ORB-SLAM (Mur-Artal et al., 2015), achieves robust and accurate camera pose estimation. This study demonstrates that deep learning features can significantly enhance the performance of traditional visual SLAM systems without the need for IMU sensor assistance. Li, Gao, Chen and Zhang (2023) introduced a visual semantic SLAM system called USP-SLAM, which combines SuperPoint (DeTone, Malisiewicz, & Rabinovich, 2018) and an improved U-Net network (Ronneberger, Fischer, & Brox, 2015), aiming to address localization and mapping challenges in dynamic environments. Through efficient feature extraction and precise semantic segmentation, this system effectively removes dynamic feature points, thereby significantly improving the robustness and accuracy of the system.

Visual-Inertial SLAM (VI-SLAM) Although vision-based SLAM systems (Engel et al., 2017; Forster et al., 2014; Forster, Zhang, Gassner, Werlberger, & Scaramuzza, 2016; Mur-Artal & Tardós, 2017a) are popular due to their low hardware cost and robust localization performance, they encounter difficulties in scenarios with rapid motion, dynamic lighting, viewpoint changes, and occlusions. Inertial navigation is resilient to these visual challenges, but cumulative drift makes achieving long-term SLAM difficult. Visual-Inertial SLAM (VI-SLAM) (Campos, Elvira, Rodríguez, Montiel, & Tardós, 2021; Cioffi & Scaramuzza, 2020; Leutenegger, Lynen, Bosse, Siegwart, & Furgale, 2015; Mur-Artal & Tardós, 2017b; Qin, Li, & Shen, 2018; Von Stumberg, Usenko, & Cremers, 2018) overcomes the limitations of single-sensor systems by integrating high-frequency inertial measurements with diverse visual geometric information. This fusion strategy utilizes the spatial details of visual data to enhance map accuracy and depends on the temporal precision of IMU data to refine the continuity and stability of motion trajectories. In situations where visual information is limited, for instance, when the camera view is obstructed or image quality degrades, the IMU supplies essential motion data to maintain the system's stability. Moreover, the camera can effectively calibrate the drift in IMU data by analyzing environmental features, thereby jointly enhancing the accuracy of long-term trajectory estimation.

The Visual-Inertial SLAM (VI-SLAM) systems can be classified into loosely-coupled and tightly-coupled types depending on the method of integrating visual and IMU data. In loosely coupled methods, the visual and IMU modules perform motion estimation independently, whereas in tightly coupled methods, raw data from the camera and IMU are jointly optimized, leading to a more globally consistent estimation. In the development of VI-SLAM, Mourikis and Roumeliotis (2007) first introduced the tightly-coupled approach in 2007 through the Multistate Constraint Kalman Filter (MSCKF). This method was further refined by the same team in 2013, resulting in MSCKF2.0 (Li & Mourikis, 2013), which featured improvements in the original filtering strategy. Additionally, the tightly-coupled SLAM technique has been adopted in other systems such as ROVIO (Bloesch, Omari, Hutter, & Siegwart, 2015), OKVIS (Leutenegger et al., 2015), VIORB (Mur-Artal & Tardós, 2017b), and VINS-Mono (Qin et al., 2018). OKVIS (Leutenegger et al., 2015) and S-MSCKF (Sun et al., 2018) are renowned stereo visual-inertial odometry systems. Based on ORB-SLAM (Mur-Artal et al., 2015), VIORB (Mur-Artal & Tardós, 2017b) introduced IMU initialization processes. VINS-Mono (Qin et al., 2018), a widely recognized monocular visual-inertial SLAM system that supports loop closure detection and map reuse, achieving accuracy on par with ORB-SLAM (Mur-Artal et al., 2015), making it one of the leading visual SLAM frameworks. In 2020, building on the 2017 release of ORB-SLAM2 (Mur-Artal & Tardós, 2017a), ORB-SLAM3 (Campos et al., 2021) was open-sourced, integrating IMU data and employing multi-map fusion technology, further enhancing system robustness and demonstrating outstanding performance across various public datasets. Recent significant advancements have been made in visual-inertial odometry (VIO) and SLAM systems for wide-field-of-view cameras. In 2022, LF-VIO (Wang et al., 2022) addressed the feature extraction problem of negative half-plane field-of-view cameras by introducing unit-length feature point vectors, surpassing existing methods. In 2023, LF-VISLAM (Wang et al., 2023) proposed an improved loop closure detection method, enhancing the system's robustness and accuracy in complex environments. In 2024, LF-PGVIO (Wang et al., 2024) introduced a wide-field VIO framework based on geodesic segments, further improving both positioning accuracy and robustness.

Fusion of Point-Line Features in vSLAM Feature point-based SLAM methods often encounter tracking drift and failures in visual pose estimation in low-texture environments. While the IMU can partially compensate for pose estimation errors caused by the absence of point features, prolonged feature loss still leads to cumulative IMU errors. Consequently, some studies (Davison et al., 2007; Klein & Murray, 2007, 2008, 2009; Montemerlo, 2002; Mouragnon et al., 2009; Mur-Artal et al., 2015; Rosten & Drummond, 2006) have begun incorporating line features into visual SLAM systems. These features are more prevalent in man-made environments and exhibit greater robustness against changes in lighting and viewpoint. Line features are not only widely present in artificial settings but also provide rich geometric information, such as building edges, door frames, and windows, aiding in overcoming the limitations of SLAM systems that rely solely on point features. As a result, SLAM methods that integrate point and line features have increasingly drawn the attention of researchers.

In 2017, Pumarola et al. integrated line features into ORB-SLAM (Mur-Artal et al., 2015) to propose the classic PL-SLAM system (Pumarola, Vakhitov, Agudo, Sanfeliu, & Moreno-Noguer, 2017). By fusing point and line features, this system demonstrated significant advantages in handling challenging texture conditions. In 2018, He et al. developed PL-SVO (Gomez-Ojeda, Briales, & Gonzalez-Jimenez, 2016), a dual-thread semi-direct visual odometry system, based on SVO (Forster et al., 2014). This system utilizes sparse pixels and line features for motion tracking and local mapping, enhancing system accuracy. Following this, PL-VIO (He, Zhao, Guo, He, & Yuan, 2018) further integrated point and line features, based on (Qin et al., 2018), to achieve effective fusion of these features. In 2020, Fu et al. (2020) optimized PL-VIO by adjusting the implicit parameters of the LSD algorithm, thereby significantly improving the system's operational efficiency. Thereafter, DPLVO (Zhou, Wang, & Kaess, 2021) and ED-PLVO (Zhou, Huang, Mao, Wang, & Kaess, 2022) utilized line feature constraints to improve the accuracy of camera pose estimation. DPLVO predicts camera poses using depth sampling on detected lines and the DSO algorithm but may introduce depth errors due to its dependence on line point depth and keyframe poses. To address this, EDPLVO employs a point tracking method along epipolar lines, reducing the need for precise depth accuracy. In 2022, Alamanos et al. proposed the ORB-LINE-SLAM system (Alamanos & Tzafestas, 2023) based on ORB-SLAM3 (Campos et al., 2021), which was the first open-source SLAM system to operate solely on line features, successfully addressing challenges in complex environments. In 2024, DynPL-SLAM (Zhang, Dong, Zhao, & Qi, 2024) significantly improved positioning accuracy and real-time performance in dynamic environments by employing dynamic feature detection and dynamic object removal. In 2025, PLE-SLAM (He, Li, Wang, & Wang, 2025) accelerated IMU bias estimation optimization through deep learning techniques, while improving dynamic feature removal and loop closure detection modules, effectively solving IMU drift issues in low-texture environments and under fast rotations. AirSLAM (Xu, Hao, Yuan, Wang, & Xie, 2025), on the other hand, uses a unified convolutional neural network (PLNet) to simultaneously detect keypoints and structural lines, combining both for tracking, mapping, and relocalization, effectively addressing both short-term and long-term challenges posed by lighting changes.

The integration of point and line features into vSLAM not only enhances system robustness and accuracy but also extends its applicability in complex environments (Yuan, Xu, & Zhou, 2023; Zhou, Zhang, Deng, & Fan, 2021). With the advancement of deep learning techniques

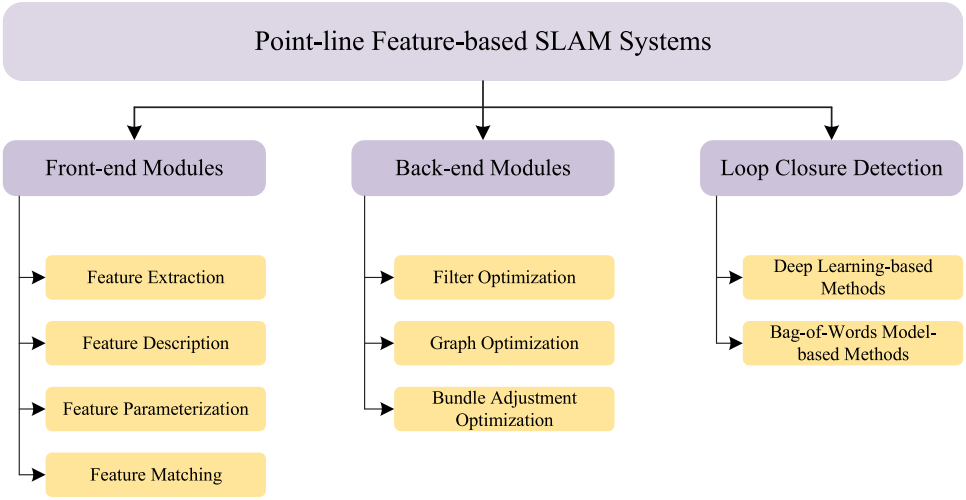


Fig. 1. Main research framework of the paper.

and increased computational capabilities, SLAM systems are advancing towards real-time processing and more sophisticated applications, such as dynamic environment understanding, semantic SLAM, and the integration of surface and structural line features (He, Li, Wang, & Wang, 2024; Yan et al., 2024; Yu et al., 2023). Table 1 provides a comparison of typical point-line feature-based SLAM systems, covering their sensor configurations, algorithm frameworks, feature types, and adaptability to different environments. Fig. 1 illustrates the main research framework of this study, highlighting the three core modules of point-line feature-based SLAM systems: the frontend module, backend module, and loop closure detection module. This framework offers an in-depth analysis of the key technological pathways and future development trends of SLAM systems that integrate point and line features.

The key contributions of this paper are as follows:

- First Comprehensive Survey: This paper presents the first comprehensive survey on the integration of point and line features in visual Simultaneous Localization and Mapping (vSLAM) systems. It systematically summarizes the current research status and development trends in this field. Through an in-depth analysis of the application of point-line features in vSLAM, this paper fills the gap in existing literature regarding line feature research and provides a clear theoretical framework and research directions, laying a solid foundation for future exploration.
- Core Technologies and Latest Advancements: This paper thoroughly explores the core technologies and latest research advancements in vSLAM systems that integrate point and line features. By analyzing various vSLAM algorithms, this paper focuses on key technical modules such as feature detection, matching, backend optimization, and loop closure. Particularly, the paper introduces several line feature detectors based on both traditional algorithms and deep learning, accompanied by performance evaluations, high-lighting the latest progress and practical applications in this domain.
- Integration of Expert Systems and SLAM: This paper is the first to explore the integration of SLAM systems with expert systems, illustrating the novelty of incorporating point-line vSLAM into expert systems.
- Future Research Directions: This paper discusses the future research directions for point-line feature-based vSLAM systems. Future research will focus on lightweight models, end-to-end frameworks, and multimodal data fusion, driving the further application of this technology in fields such as autonomous driving and robotic navigation.

2. Front-end modules

This section delves into the front-end modules of vSLAM systems that integrate point and line features. As the core component of vSLAM, the front-end modules are responsible for extracting stable and reliable visual features from complex environments, and ensure system accuracy and robustness through effective feature description, parameterization, and matching strategies. First, we introduce line feature extraction methods, covering both traditional techniques and recent advances based on deep learning. Subsequently, we will discuss the description methods of line features, focusing on commonly used local binary descriptors, and examine their application in feature matching and strategies to improve matching accuracy. Next, we will elaborate on the parameterization methods for line features, as these techniques are essential for enhancing the overall performance of vSLAM systems. Finally, we will explore line feature matching strategies and examine their pivotal role in improving feature matching accuracy and the overall performance of SLAM systems.

2.1. Line feature extraction

Line feature extraction is a fundamental step in point-line feature-based vSLAM systems, significantly influencing the subsequent feature matching and overall robustness of the system. This subsection provides a comprehensive review of the evolution of line feature extraction techniques, including traditional algorithms such as the Hough Transform (Hough, 1962), as well as more advanced methods such as the Line Segment Detector (LSD) (Von Gioi, Jakubowicz, Morel, & Randall, 2008), the Edge Drawing Line Detector (EDLines) (Akinlar & Topal, 2011), and recent deep learning-based extraction schemes. We focus on analyzing the underlying principles, recent advancements, strengths and limitations of these methods, and summarize their typical applications within vSLAM frameworks.

2.1.1. Hough transform

The Hough Transform (HT), proposed by Paul Hough in 1962, is a parameter space method used for detecting straight lines in images (Canny, 1986). It works by mapping points from the image space into a parameter space and identifying collinear point sets to detect lines. While the Hough Transform is robust against noise and suitable for initial line detection, its high computational complexity limits its real-time application in SLAM systems (Furukawa & Shinagawa, 2003; Vakhitov, Funke, & Moreno-Noguer, 2016; Xu, Shin, & Klette, 2015).

The core of the Hough Transform lies in the mapping relationship between the image space and the parameter space, where collinear

Table 1
Point-line feature-integrated vSLAM systems.

Methods	Year	Sensors			Framework		Feature type		Env		IMU	Op
		Mo	St	RG	Fil	Opt	PL	PLP	In	Out		
PLVO (Lu & Song, 2015)	2015			✓		✓	✓		✓			✓
Pop-up slam (Yang, Song, Kaess, & Scherer, 2016)	2016	✓			✓			✓	✓	✓		✓
PL-StVO (Gomez-Ojeda & Gonzalez-Jimenez, 2016)	2016		✓			✓	✓		✓	✓		✓
PL-SVO (Gomez-Ojeda et al., 2016)	2016	✓				✓	✓		✓	✓		✓
PL-SLAM (Pumarola et al., 2017)	2017	✓				✓	✓		✓	✓		✓
Trifo-VIO (Zheng et al., 2018)	2018		✓		✓	✓	✓		✓	✓	✓	✓
PL-VIO (He et al., 2018)	2018	✓				✓	✓		✓	✓	✓	✓
PL-SLAM (Gomez-Ojeda, Moreno, Zuniga-Noël, Scaramuzza, & Gonzalez-Jimenez, 2019)	2019		✓			✓	✓		✓	✓		✓
StructVIO (Zou, Wu, Pei, Ling, & Yu, 2019)	2019			✓		✓			✓	✓	✓	✓
PLP-VIO (Li, He, Lin, & Liu, 2020)	2020	✓				✓		✓	✓	✓	✓	✓
PL-VINS (Fu et al., 2020)	2021	✓				✓	✓		✓		✓	✓
PELI-SLAM (Rong et al., 2021)	2021		✓			✓	✓		✓	✓		
DPLVO (Zhou, Wang et al., 2021)	2021	✓				✓	✓		✓	✓		✓
IF-SLAM (Wang et al., 2021)	2021	✓				✓	✓		✓	✓		✓
PLF-VINS (Lee & Park, 2021)	2021			✓		✓	✓		✓	✓	✓	✓
PLD-VINS (Zhu, Jin, Lou, & Zhao, 2021)	2021	✓				✓	✓		✓	✓	✓	✓
EDPLVO (Zhou et al., 2022)	2022	✓				✓	✓		✓	✓		✓
UV-SLAM (Lim, Jeon, & Myung, 2022)	2022	✓				✓	✓		✓	✓	✓	✓
PLS-VIO (Xu et al., 2022)	2022	✓				✓	✓	✓	✓	✓	✓	✓
Struct-MDC (Jeon, Lim, Seo, & Myung, 2022)	2022			✓		✓	✓		✓	✓	✓	✓
PLI-SLAM (Wang, Guan, Yu, & Zhang, 2022)	2022	✓				✓	✓		✓		✓	✓
DPL-SLAM (Yu et al., 2023)	2023	✓				✓	✓		✓			✓
PLV-IEKF (Hua et al., 2023)	2023	✓			✓			✓	✓	✓	✓	
EPLF-VINS (Xu, Yin, Shi, Jiang, & Huang, 2022)	2023	✓				✓	✓		✓	✓	✓	✓
AirVO (Xu, Hao, Yuan, Wang, & Xie, 2023)	2023	✓				✓	✓		✓	✓		✓
PLP-SLAM (Shu, Wang, Pagani, & Stricker, 2023)	2023	✓	✓	✓		✓	✓		✓	✓		✓
PLPF-vSLAM (Yan et al., 2024)	2024			✓		✓		✓	✓			✓
PLE-SLAM (He et al., 2024)	2024		✓			✓	✓		✓		✓	✓
PLVS (Freda, 2023)	2024		✓	✓		✓	✓		✓		✓	✓
EPL-VINS (Zeng, Liu, Huang, & Liu, 2024)	2024	✓	✓			✓	✓		✓		✓	
FLM PL-VIO (Lin et al., 2024)	2024	✓	✓			✓	✓		✓	✓	✓	
DynPL-SLAM (Zhang et al., 2024)	2024	✓	✓	✓		✓	✓		✓	✓	✓	
Air-SLAM (Xu et al., 2025)	2025	✓	✓			✓			✓	✓	✓	✓

Note: Mo means Monocular; St means Stereo; RG means RGBD; Fi means Filter-based; Opt means Optimization-based; PL means Point and Line; PLP means Point, Line, and Plane; Env means Environment; In means Indoor; Out means Outdoor; OP means Open Source.

point sets are identified by counting the intersection points in the parameter space. Duda and Hart optimized the original method (Duda & Hart, 1972), while Ballard extended it to detect arbitrary shapes such as circles and ellipses (Ballard, 1981). To improve the accuracy and robustness of line segment extraction, researchers have continuously refined the Hough Transform, including analyzing the peak distribution in the parameter space and applying closed-form Hough Transforms to accelerate pose estimation (Maire, Arbelaez, Fowlkes, & Malik, 2008).

Although the Hough Transform is simple and robust against noise, it still faces challenges in terms of accuracy and efficiency in practical applications. With advancements in computer vision, ongoing improvements and optimizations to the Hough Transform aim to enhance its performance for real-time systems while maintaining its robustness.

2.1.2. LSD

LSD and Its Development The LSD algorithm, proposed by Von Gioi et al. (2008), is a gradient-based line segment detection method widely employed in SLAM frameworks. The algorithm begins by computing the gradient magnitude and direction for each pixel, using the pixel with the highest gradient as a starting point for region growing to form line-support regions. Subsequently, line segment parameters are extracted through rectangular approximation, and detection accuracy is further improved by calculating false positives and applying refinement procedures. The advantages of LSD lie in its parameter-free nature, sub-pixel precision, and linear time complexity, allowing it to effectively control false detection rates.

Building on the LSD algorithm, several improvements have been proposed. The Multiscale Line Segment Detector (MLSD) (Salaün, Marlet, & Monasse, 2016) employs a multi-scale technique to reduce over-segmentation and enhance robustness in low-contrast regions, making it particularly suitable for feature-sparse scenarios. It significantly improves the robustness and accuracy of line-based Structure from Motion (SfM) reconstruction. The Point-Line Segment Detector (PLSD) (Yu, Xu, Cheng, & Zhu, 2020) addresses over-segmentation through intelligent grouping and a multi-scale framework, improving the detection of long line segments while incorporating geometric and gradient constraints to prevent incorrect merging. The MPG-LSD (Wang, Zhong, Chen, & Zheng, 2024) further enhances the continuity, direction, and position accuracy of line segments by performing perceptual grouping in multi-scale space, achieving higher quality line segment detection in high-resolution and noisy images.

Although traditional LSD methods have made progress in detection performance, computational time and generalization capability remain bottlenecks in complex scenes. To address these issues, deep learning-based LSD improvement methods have gradually emerged (Huang et al., 2020). DeepLSD (Pautrat and Barath and Larsson and Oswald, & Pollefeys, 2023) combines deep learning with traditional detection techniques, utilizing line attraction fields generated by deep networks to improve detection accuracy under varying lighting conditions and noise interference. However, deep learning methods typically require GPU support, which limits their real-time applicability. To overcome this challenge, M-LSD (Gu et al., 2022) optimizes the network structure, significantly increasing inference speed and compressing model size, thus achieving a balance between performance and efficiency. Nevertheless, these methods still face significant computational resource demands. Future research could focus on improving computational efficiency to better suit low-resource environments. A summary of the progress of other related improvements is presented in Fig. 2.

The application of LSD in vSALM The application of deep learning-based line feature detection in point-line integrated vSLAM remains relatively limited, where-as traditional line feature extraction methods have reached a more mature stage. Among these, the LSD algorithm is widely used due to its strong performance. Nonetheless, it still faces

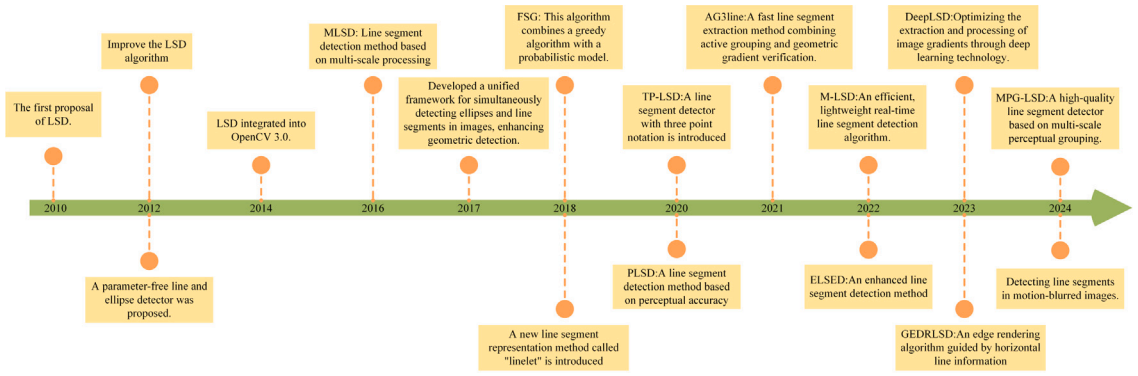


Fig. 2. Timeline of LSD and Its Major Improvements (Cho, Yuille, & Lee, 2017; Gu et al., 2022; Huang et al., 2020; Lin, Zhou, Liu, & Zhu, 2023b; Pătrăucean, Gurdjos, & von Gioi, 2016; Pătrăucean, Gurdjos, & Von Gioi, 2012; Pautrat and Barath et al., 2023; Salaün et al., 2016; Suárez, Buenaposada, & Baumela, 2022; Suárez, Muñoz, Buenaposada, & Baumela, 2018; Von Gioi et al., 2008; Von Gioi, Jakubowicz, Morel, & Randall, 2012; Wang, Zhong et al., 2024, 2024; Xia, Meng, Zhang, Zhang, & Hu, 2022; Yu, Li, Yang, Yu, & Xia, 2023; Yu et al., 2020).

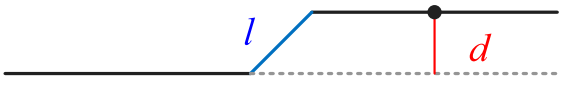


Fig. 3. Distances between two line segments (Zuo et al., 2017).

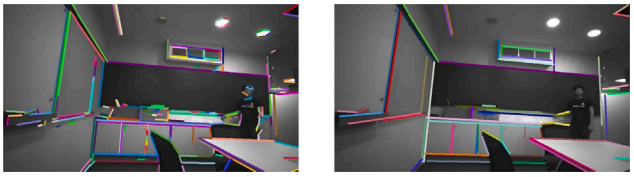


Fig. 4. Merging short line segments to form stable longer lines, showing a comparison of the original (left) and improved LSD detection results (right) (Xu et al., 2023).

challenges in practical applications, such as instability in line feature detection, difficulties in feature matching, and constraints on real-time performance.

In most point-line integrated vSLAM systems, line features are often processed using strategies such as merging or filtering short line segments (Lim, Kim, Jung, Hu, & Myung, 2021). Compared to short line segments, long segments exhibit higher repeatability and lower sensitivity to noise, thus providing more stable and accurate constraints. For instance, Zuo, Xie, Liu, and Huang (2017) proposed enhancing the robustness of the LSD algorithm by merging line segments with directional and distance differences below a predefined threshold, as shown in Fig. 3. Similarly, Xu et al. (2023) combined short line segments detected by LSD and filtered out those still below the threshold after merging, retaining only the long segments to ensure stability and accuracy, as illustrated in Fig. 4. Another common approach is to map image points into Hough space and merge line segments with the same start and end points to achieve more robust and accurate data association (Zhi, 2022).

The LSD algorithm was initially designed for contour description. To adapt it for spatial line segment localization, Zhang, Liu, Li, Pang, and Wang (2022) modified the LSD algorithm by introducing adaptive line segment constraints, significantly improving the speed and efficiency of line feature extraction and matching. With the integration of multi-scale processing, the LSD algorithm can detect line segments across different image resolutions, enhancing both accuracy and robustness (Li et al., 2023). This method effectively addresses the challenges of matching line features across varying perspectives and scales and improves the adaptability of SLAM systems to environmental changes (Von Gioi et al., 2008). However, in highly dynamic environments, the LSD algorithm still faces issues, such as the fragmentation of long straight lines into multiple segments due to occlusion and local blurring, which negatively impacts the stability and real-time performance of inter-frame line feature matching (Yang, Geneva, Eckenhoff, & Huang, 2019; Zhang, Li, Zong, Zhu, & Wang, 2017). To address this, researchers have optimized the LSD algorithm by adjusting parameters and setting a minimum length threshold to speed up detection and improve the real-time performance and accuracy of line feature matching (Fu et al., 2020; Liu, Wen, & Zhang, 2023; Shu et al., 2023). Additionally, to address the issue of short, overlapping, or coincident line segments commonly encountered in structured scenes, an adaptive threshold line segment extraction algorithm was proposed, which significantly enhanced the robustness and accuracy of the algorithm, further improving line segment detection capabilities in complex environments for visual SLAM systems (Zhao, Song, Xing, Lei, & Wang, 2022).

2.1.3. EDLines

EDLines and Its Development EDLines EDLines (Akinlar & Topal, 2011) is an efficient line segment detection algorithm introduced by Cihan Topal and Cuneyt Akinlar in 2012, designed for real-time computer vision and image processing applications. This algorithm generates clear chains of edge pixels based on the Edge Drawing (ED) technique and validates line segments using Helmholtz's principle (Desolneux, Moisan, & Morel, 2007), effectively reducing false detections without the need for manual parameter adjustment. Compared to the LSD (Von Gioi et al., 2008) algorithm, EDLines demonstrates significant advantages in speed, accuracy, and robustness, capable of handling both geometric and non-geometric structures. Improved versions of EDLines, such as GEDRLSD (Lin et al., 2023b), incorporate iso-line information, enhancing the robustness and precision of line segment detection. The E2LSD (Lin, Zhou, Liu, & Zhu, 2023a) algorithm further improves detection efficiency by introducing dual consistency constraints and decoupled fitting methods.

The application of EDLines in vSALM EDLines, due to its higher processing speed compared to LSD, is widely used in point-line vSLAM. However, it often produces line segments that include irrelevant parts, limiting detection accuracy (Ma, Wang, He, Mei, & Zhao, 2019). By incorporating outlier elimination methods, EDLines can achieve performance similar to LSD without sacrificing accuracy (Ma & Ning, 2020). As a result, EDLines is more suitable for real-time SLAM applications. The improved EDLines algorithm (Xu, Yin et al., 2022) enhances the quality of line features through short-line merging and adaptive threshold strategies, ensuring a well-distributed spatial layout of the features, which provides stable constraints for SLAM systems. Additionally, in RGB-D visual-inertial SLAM, the modified EDLines algorithm introduces length suppression and short-line merging strategies (Zhu et al., 2021), significantly improving detection accuracy and stability while maintaining the original algorithm's speed advantage.

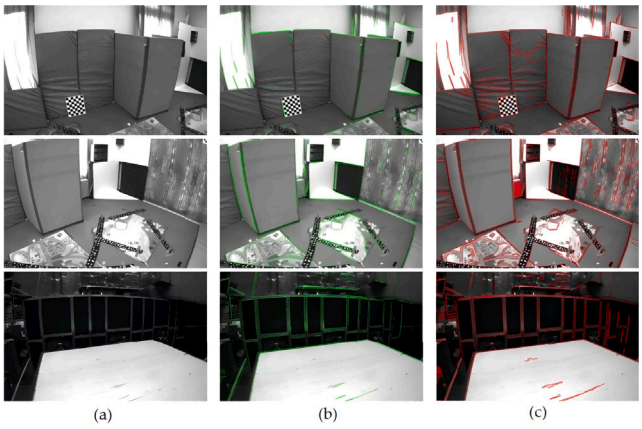


Fig. 5. The extracted lines of LSD and EDlines in the EuRoC dataset (Tateno et al., 2017). (a) The original images. (b) The green lines in the left column are the line features detected by LSD. (c) The red lines in the right column are the line features detected by EDlines.

Fig. 6. The comparison of line feature detection methods: Performance of traditional methods and deep learning methods on the YorkUrban dataset.(Akinlar & Topal, 2011; Huang et al., 2020; Pautrat and Barath et al., 2023; Pautrat, Lin, Larsson, Oswald, & Pollefeys, 2021; Von Gioi et al., 2008; Xue et al., 2023)

As shown in Fig. 5, the improved EDLines performs better than LSD in complex environments. Furthermore, EDLines has been further optimized by incorporating curvature detection and an improved minimum line segment length criterion to reduce false positives (Teng et al., 2023), further enhancing line segment matching accuracy. The ELSED (Enhanced Line Segment Detector) algorithm (Suárez et al., 2022) improves robustness against complex backgrounds and high-noise images through edge enhancement, noise suppression, and multi-scale analysis. It maintains high accuracy and low computational complexity in dynamic environments. The advantages of this algorithm have led to its application in line-based SLAM, significantly boosting the robustness of the system (Seo, Lim, Lee, Lim, & Myung, 2023).

2.1.4. Learning-based methods

Although traditional line feature detection methods are widely used in point-line fusion SLAM, they exhibit significant limitations under variations in lighting, scale, and camera viewpoints. With the development of deep learning, learning-based methods have gradually become an effective solution to these issues. By automatically learning

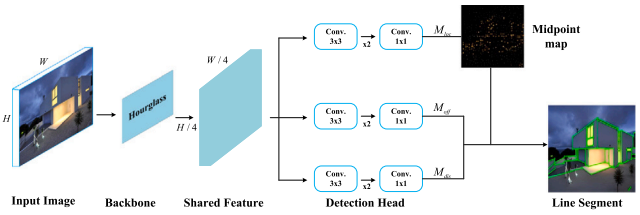


Fig. 7. Architecture of the VLSE Network (Gao et al., 2022).

feature representations, deep learning-based line feature detection has demonstrated significant advantages in complex environments. Recent research (Huang et al., 2018) introduced large public datasets and heuristic line-frame fusion algorithms, making deep learning-based line segment detection feasible. For example, L-CNN (Zhou, Qi, & Ma, 2019) uses an end-to-end neural network to directly generate vectorized line frames from images, avoiding the intermediate steps of traditional methods and significantly improving detection accuracy. Meanwhile, the Holistically-Attracted Wireframe Parser (HAWP) (Xue et al., 2023) achieves unified detection of line segments and junctions through an Attraction Field Map (AFM), and optimizes the inference process based on (Xue et al., 2019), improving both real-time performance and detection accuracy.

Compared to traditional methods, deep learning-based models offer more robust line detection, particularly in handling environmental changes. Kannapiran et al. (2023) employed the SOLD2 (Self-supervised Occlusion-aware Line Detection and Description) algorithm for line feature detection and description. The CARLA driving simulator (Dosovitskiy, Ros, Codevilla, Lopez, & Koltun, 2017) enhanced the performance of SOLD2 under low-light and adverse weather conditions. Additionally, VLSE (Gao et al., 2022) uses a new line segment representation and hybrid convolution modules to directly predict the endpoints of line segments(see Fig. 7), simplifying the post-processing steps while significantly improving detection efficiency and accuracy. Liu, Cao et al. (2024), based on the method in Pautrat, Suárez, Yu, Pollefeys, and Larsson (2023), extracts line segment features from images using a deep learning model and unifies key points, 2D line segments, and their descriptors into a bounding box structure, thereby improving the accuracy and robustness of visual localization. Fig. 6 compares the performance of several traditional and deep learning-based methods for line feature extraction. Table 2 contrasts these two categories of methods, highlighting their examples, advantages, disadvantages, suitable applications, and challenges. Table 3 further evaluates the performance of various line feature detection methods on the Wireframe and YorkUrban datasets.

Although these deep learning methods significantly improve the accuracy and robustness of visual localization in SLAM tasks, they typically require high computational resources, which may become a bottleneck in practical applications. Future research may focus on reducing computational complexity through lightweight network architectures and multi-feature strategies, thus enhancing their potential for real-time SLAM applications.

2.2. Line feature description

Line feature description is a key step in inter-frame feature matching for point-line SLAM, aimed at generating stable and distinctive descriptors for line segments in images to achieve accurate matching. Traditional descriptors perform well in simple scenes but exhibit lower accuracy in environments with weak texture and fragmented line segments. In contrast, deep learning-based methods enhance robustness and adaptability in complex and dynamic environments by automatically learning feature representations.

Table 2
Comparison of traditional and deep learning-based line feature extraction methods.

Method	Traditional methods	Deep learning-based methods
Examples	Hough Transform (Hough, 1962), LSD (Von Gioi et al., 2008), EDLines (Akinlar & Topal, 2011)	DeepLSD (Pautrat and Barath et al., 2023), SOLD2(Kannapiran et al., 2023), VLSE (Gao et al., 2022), HAWP (Xue et al., 2023)
Advantages	Simple to implement Real-time performance Low computational cost Robust against noise	Automatic feature learning Robust in dynamic and complex environments Handle varying lighting conditions End-to-end optimization
Disadvantages	Sensitive to noise in complex scenes Requires manual tuning Struggles with dynamic environments and occlusion	High computational cost Requires GPU support Data-intensive May struggle with real-time constraints in low-power systems
Best Suited For	Simple, static environments Resource-limited systems Low-complexity applications	Dynamic environments Varying illumination Complex and large-scale scenes Systems with high computational resources
Challenges	Limited accuracy in complex scenarios Inability to handle large-scale variation	High computational overhead Requires large training datasets Real-time limitations in certain systems

Table 3
Performance comparison of line feature detection methods.

Methods	Wireframe dataset			YorkUrban dataset			FPS
	sAP^5	sAP^{10}	sAP^{15}	sAP^5	sAP^{10}	sAP^{15}	FPS
AFM (Xue et al., 2019)	18.5	24.4	27.5	7.3	9.4	11.1	10.4[a]
L-CNN (Zhou et al., 2019)	27.7	32.4	34.8	9.5	11.6	13.2	8.0[a]
LETR (Denis, Elder, & Estrada, 2008)	59.2	65.2	67.7	23.9	27.6	29.7	2.0
TP-LSD (Huang et al., 2020)	56.4	59.7	59.7	24.8	26.8	31.2	78.2
F-Clip (Xue et al., 2019)	64.3	68.3	69.1	28.6	31.0	32.4	82.3
ELSD (Xue et al., 2019)	64.3	68.9	70.9	27.6	30.2	31.8	42.6[a]
HAWPv2(Xue et al., 2023)	65.7	69.7	71.3	28.9	31.2	32.6	85.6
PLNet (Xu et al., 2025)	65.2	69.2	70.9	29.3	32.0	33.5	79.4[a]

[a] These numbers are cited from the original paper.

2.2.1. Traditional descriptors

Line Band Descriptor (LBD) (Zhang & Koch, 2013) is a commonly used line feature descriptor in point-line SLAM, which constructs the descriptor by extracting gradient information around line segments and incorporating geometric properties. LBD demonstrates good robustness to image transformations such as rotation and lighting changes. MSLD (Multiscale Line Segment Descriptor) (Wang, Wu, & Hu, 2009) is an improvement of LBD, which enhances the adaptability to images at different scales through multiscale techniques, while maintaining strong robustness under various image transformations.

2.2.2. Learning-based descriptor methods

In recent years, the advancement of deep learning technology has significantly propelled the progress of line segment detection methods. Traditional line feature detection techniques primarily rely on local information, such as pixel gradients, which limits their ability to fully exploit global structural information. In contrast, convolutional neural network (CNN)-based approaches (Xue et al., 2019; Zhou et al., 2019) are more capable of understanding the line structures within an environment and generating semantically meaningful line segments, such as object boundary lines or junction lines in specific scenes, which is crucial for achieving robust SLAM in complex environments. For instance, the DLD method (Lange, Schweinfurth, & Schilling, 2019) generates high-performance line segment descriptors by inputting the region surrounding line segments into a ResNet network. Moreover, the learnable line segment detector and descriptor (L2D2) method (Abdellali, Frohlich, Vilagos, & Kato, 2021) enables efficient extraction and matching of 2D line segments based on the angular distance of 128-dimensional unit descriptor vectors. The graph convolutional network-based line segment matching technique (Ma, Jiang, & Lai, 2020) introduces a new perspective for line feature matching by learning local line segment descriptors through end-to-end training. Additionally, SOLD2, introduced in Pautrat et al. (2021), is a self-supervised line detector similar to SuperPoint, which does not rely on any annotated information, thereby enhancing the system's adaptability to diverse and variable scenes.

In the field of point-line vSLAM, deep learning-driven line feature description methods have also been introduced and developed. For instance, LLD-Net (Vakhitov & Lempitsky, 2019) utilizes a deep convolutional architecture to generate learnable line segment descriptors,as shown in Fig. 8. This method processes the input images and detected line segments through a fully convolutional neural network, significantly improving processing efficiency, and enhances the processing speed of multiple line segments by sharing computational results. Superline (Qiao, Bai, Xiang, Qian, & Bi, 2021) proposes an end-to-end model for detecting semantic line features in images and generating robust descriptors. The model aggregates multiscale information using a line selection mechanism and spatial pyramid pooling module, improving matching stability. Experimental results show that Superline performs excellently on the Wireframe (Huang et al., 2018) and York Urban (Denis et al., 2008) datasets, validating its efficiency and reliability in SLAM tasks. Furthermore, SOLD2 (Kannapiran et al., 2023; Matsumoto, Nakano, & Ogura, 2024) further enhances line feature description performance through self-supervised learning, providing improved robustness, especially in occlusion and complex environments. This method enhances the stability of line feature matching in dynamic scenes by considering the occlusion relationships between

Table 4
Comparison of traditional and deep learning-based line feature description methods.

Category	Method	Advantages	Disadvantages	Best suited for
Traditional-based	LBD (Zhang & Koch, 2013) MSLD (Wang et al., 2009)	5cmRobust to image variance Mathematically explainable Relatively insensitive to low texture	5cmSensitive to low texture Fewer global features	5cmVisual SLAM Structure-from-Motion Robotics
Learning-based	DLD (Lange et al., 2019) L2D2(Abdellali et al., 2021) SOLD2(Pautrat et al., 2021) LLD-Net (Vakhitov & Lempitsky, 2019)	Learns local and global features Robust in complex environments Efficient for real-time applications	Deeply reliant on training datasets High computational cost	Dynamic SLAM Autonomous driving Scene recognition

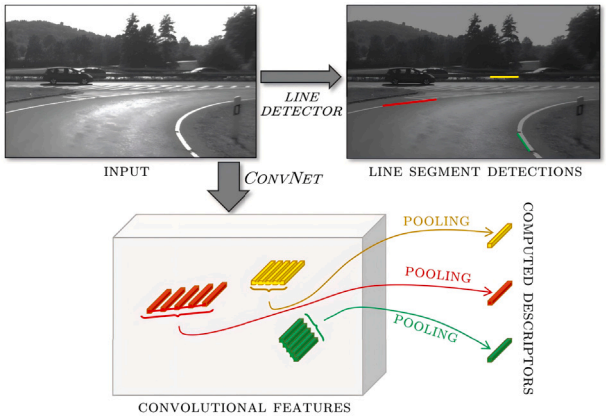
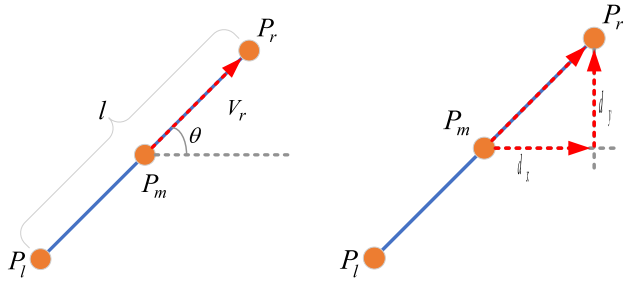


Fig. 8. The architecture of the line feature descriptor (Vakhitov & Lempitsky, 2019).



(a) Traditional line representation with length l and angle θ

(b) Horizontal and vertical distances d_x and d_y from the midpoint to the endpoints

Fig. 9. Line segment representation (Gao et al., 2022).

line segments. Table 4 summarizes a comparison of major line feature descriptor methods.

In summary, deep learning-based line feature descriptor methods have achieved significant improvements in terms of accuracy and robustness. Through the application of convolutional neural networks, self-supervised learning, and graph convolutional networks, it is now possible to efficiently extract line segment descriptors and achieve more reliable line feature matching in dynamic and complex environments.

2.3. Line feature parameterization

To effectively utilize line features, To effectively utilize line features, it is necessary to convert them from image space into a three-dimensional geometric representation. In vSLAM systems, the parameterization of line features directly affects feature extraction, matching efficiency, and ultimately, localization accuracy. Common parameterization methods include the endpoint method (Yang et al., 2019), Plücker coordinates (Joswig & Theobald, 2013), and orthogonal representation (Montiel, Tardós, & Montano, 2000). The endpoint method represents line features using the coordinates of two endpoints, which is intuitive and easy to implement. Plücker coordinates represent a line in 3D space with six parameters and are widely used in transformations and projection calculations. The orthogonal representation, on the other hand, uses four parameters and provides a more precise optimization effect with better convergence. These methods can effectively enhance the performance of visual SLAM systems, ensuring accurate perception and navigation capabilities of robots in complex environments. In Table 5, we summarize several typical point-line vSLAM systems that employ different line feature parameterization methods.

2.3.1. Endpoint representation method

The endpoint method represents a three-dimensional line using two endpoint coordinates, simplifying the derivation of the line equation. Initially, this method was applied in small-scale scenarios within the EKF framework (Montemerlo & Thrun, 2003; Zhao, Sun, Zhang, & Xiong, 2023), and later evolved to accommodate environments with slow motion and small viewpoint changes (Castellanos & Tardos, 2012; Gee & Mayol-Cuevas, 2006; Wan & Van Der Merwe, 2000). Today, the endpoint method, combined with nonlinear optimization and keyframe strategies, is widely used in Visual Odometry (VO) (Gomez-Ojeda & Gonzalez-Jimenez, 2016; Lu & Song, 2015) and vSLAM (Quan & Lan, 1999; Sturm, Engelhard, Endres, Burgard, & Cremers, 2012; Vakhitov et al., 2016). It has notably improved system accuracy and efficiency, particularly in complex and low-texture environments. Shu et al. (2023) was the first to combine the endpoint method, Plücker coordinates, and orthonormal representation. The endpoint method is used to simplify the projection and matching of line segments, while Plücker coordinates are employed for optimization tasks, using a six-dimensional vector to represent 3D line segments for easier transformation. During the BA optimization process, Plücker coordinates are converted into a four-degree-of-freedom orthonormal representation, enhancing computational efficiency. Inspired by Dai, Gong, Wu, Yuan, and Ma (2022), Zhang, Luo, Qin, He, and Liu (2021), Gao et al. (2022) introduced an innovative endpoint method, as shown in Fig. 9. This method parameterizes the line segment by predicting the horizontal and vertical distances of the two endpoints relative to the midpoint, effectively reducing memory and time consumption. Compared to traditional methods, this endpoint-based representation significantly improves the detection accuracy and stability of long line segments, demonstrating superior performance, especially in large-scale datasets.

However, the endpoint-based method has certain limitations, including issues with precision and stability, as well as an inability to effectively describe infinitely extending lines. These drawbacks limit its

Table 5
Line feature parameterization methods in typical point-line vSLAM systems.

Paper	Year	Parameterization				LSEM			Env		Ac	Ro
		EP	Plücker	Orth	Oth	LSD	EDL	DL	In	Out		
PL-SVO (Gomez-Ojeda et al., 2016)	2016	✓				✓			✓	✓	Med	Med
PL-SLAM (Pumarola et al., 2017)	2017	✓				✓			✓		High	High
graph-based SLAM (Zuo et al., 2017)	2017		✓	✓		✓			✓	✓	High	High
PL-VIO (He et al., 2018)	2018		✓	✓		✓			✓	✓	Med	High
PL-SLAM (Gomez-Ojeda et al., 2019)	2019	✓				✓			✓	✓	High	High
LLD (Vakhitov & Lempitsky, 2019)	2019				✓			✓	✓	✓	High	High
PLP-VIO (Li, He et al., 2020)	2020		✓	✓		✓			✓		High	High
S-SLAM (Li, Brasch, Wang, Navab, & Tombari, 2020)	2020	✓				✓			✓		Med	High
PLR-SLAM (Ma & Liang, 2021)	2021				✓	✓			✓		High	High
DPLVO (Zhou, Wang et al., 2021)	2021				✓	✓			✓		High	High
EDLVO (Zhou et al., 2022)	2022	✓				✓	✓		✓	✓	Med	Med
ORB-LINE-SLAM (Alamanos & Tzafestas, 2023)	2022	✓				✓	✓		✓	✓	Med	Med
Airvo (Xu et al., 2023)	2023		✓	✓		✓			✓		High	High
IDLL (Li et al., 2023)	2023				✓	✓			✓		High	High
Structure plp-slam (Shu et al., 2023)	2023		✓	✓		✓			✓		High	High
EPL-VINS (Zeng et al., 2024)	2024	✓		✓			✓		✓	✓	High	High
LOC-PL (Matsumoto et al., 2024)	2024			✓				✓	✓	✓	Med	Med

Note: EP means EndPoints Method; Orth means Orthonormal Representation; Oth means Others; LSEM means Line Segment Extraction Method; LSD means Line Segment Detector; EDL means EDLines; DL means Deep Learning-based methods; Env means Environment; Ac means Accuracy; Ro means Robustness.

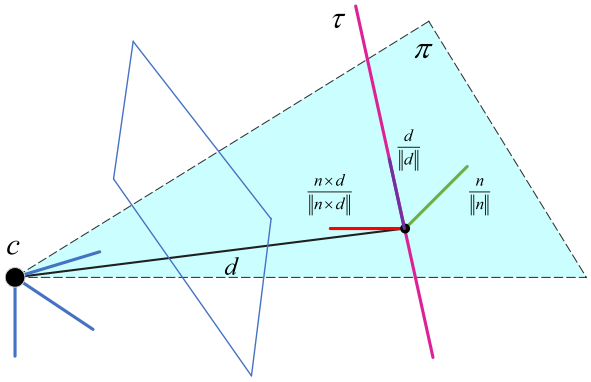


Fig. 10. Plücker line coordinates (He et al., 2018).

applicability in long-term data association. Overall, the development and application of the endpoint-based method in vSLAM reflect a transition from early SLAM approaches towards addressing the demands for higher precision and large-scale environment mapping.

2.3.2. The Plücker coordinate method

The Plücker coordinates, introduced by the German mathematician and physicist Julius Plücker in the 19th century, constitute a coordinate system for representing lines in three-dimensional space (Joswig & Theobald, 2013). By mapping a spatial line into a six-dimensional vector space, Plücker coordinates simplify geometric operations such as intersection, transformation, and projection, proving particularly advantageous in fields like computer vision and computer graphics.

The Plücker coordinates consist of two three-dimensional vectors: the direction vector of the line and the normal vector of the plane, as shown in Fig. 10. Here, C represents the origin, L denotes the line, and π indicates the plane in which the line lies. The distance from the line to the origin can be expressed as:

$$d = \frac{\|\mathbf{v}\|}{\|\mathbf{d}\|} \tag{1}$$

Assume that the line L passes through points p_1 and p_2, then the direction vector $\mathbf{d}$ is defined as:

$$\mathbf{d} = p_1 - p_2 \tag{2}$$

The normal vector $\mathbf{n}$ represents the normal direction of the plane determined by the line L and the origin. It is defined as: $\mathbf{n} = p_1 \times p_2$, where "×" denotes the cross product of vectors. Therefore, the Plücker coordinates $\mathcal{L}$ can be represented as:

$$\mathcal{L} = \left(\mathbf{n}^T, \mathbf{d}^T\right)^T \in \mathbb{R}^6 \tag{3}$$

where $\mathbf{n} \in \mathbb{R}^3$, $\mathbf{d} \in \mathbb{R}^3$.

The Plücker coordinates have a constraint condition, meaning that there is an implicit constraint between the direction vector $\mathbf{d}$ and the normal vector $\mathbf{n}$, expressed as:

$$\mathbf{n}^T\mathbf{d} = 0 \tag{4}$$

This is because $\mathbf{n}$ is obtained from the cross product of P_1 and P_2, and the result of the cross product is perpendicular to the plane containing both vectors. Due to the constraint mentioned above, the Plücker coordinates are an over-parameterized representation. In fact, representing a line only requires four degrees of freedom, so in practical optimization processes, a more concise parameterization method should be employed.

In geometric modeling and triangulation, representing a 3D line using a combination of normal and directional vectors is widely favored due to its intuitiveness and operational convenience. Subsequently, to transform a line from the world coordinate system into the camera coordinate system, the following transformation matrix will be applied:

$$\mathbf{T}_{cw} = \begin{bmatrix} \mathbf{R}_{cw} & \mathbf{P}_{cw} \\ \mathbf{0} & 1 \end{bmatrix} \tag{5}$$

Thus, the Plücker coordinates in the camera coordinate system are:

$$\mathcal{L}^c = \begin{bmatrix} \mathbf{n}^c \\ \mathbf{d}^c \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{cw} & [\mathbf{P}_{cw}]_\times \mathbf{R}_{cw} \\ \mathbf{0} & \mathbf{R}_{cw} \end{bmatrix} \mathcal{L}^w \tag{6}$$

Where $\mathbf{R}_{cw}$ is the rotation matrix, $[\mathbf{P}_{cw}]_\times$ is the skew-symmetric matrix of the translation vector, $\mathcal{L}^w$ represents the Plücker coordinates of the line in the world coordinate system, and $\mathcal{L}^c$ represents the Plücker coordinates of the line in the camera coordinate system.

In point-line SLAM, research solely focusing on Plücker coordinates is relatively sparse, mainly due to the issue of over-parameterization. Most studies address this problem by combining them with orthogonal representations, as seen in works (Rong et al., 2021; Xu et al., 2023; Xu, Yin et al., 2022; Yuan et al., 2023; Zhang, Fang, Luo, & Liu, 2023), thus improving computational efficiency and numerical stability. For example, the PLV-IEKF framework proposed in Hua et al. (2023) combines point, line, and vanishing point features, using a joint parameterization of 3D line segments through Plücker coordinates and orthogonal representations, enhancing the accuracy and robustness of the SLAM system.

By employing Singular Value Decomposition (SVD) for line feature triangulation, a modified Plücker coordinate method was proposed in Seo et al. (2023), which enables the stable estimation of the initial 3D line feature pose and ensures the stability of line feature representation by using SVD decomposition. Despite the redundancy of Plücker coordinates, they are typically transformed into orthogonal representation during optimization to reduce computational complexity (Lim et al., 2022).

2.3.3. Orthonormal representation

In three-dimensional space, the representation of lines significantly influences the performance of optimization algorithms. The Plücker coordinates, due to their redundancy, introduce additional degrees of freedom during the optimization process, thereby increasing computational costs and affecting system stability. In contrast, the orthonormal representation proposed by Montiel et al. (2000) uses a minimal set of four parameters to represent a line in space, providing a more compact parameterization and demonstrating superior accuracy and convergence performance (Zhang, Lee, Lim, & Suh, 2015). The orthonormal representation utilizes QR decomposition (Bartoli & Sturm, 2005) to calculate the representation of a 3D line. Through QR decomposition, a 3D line in Plücker coordinates is transformed into a product of a rotation matrix and a diagonal matrix, yielding a minimal representation of the line:

where $\mathbf{U}$ is the rotation matrix, and $\mathbf{\Sigma}$ is a diagonal matrix containing information about the direction and distance of the line. After QR decomposition, the line representation is compressed into four degrees of freedom: three from the rotation matrix, which converts the line coordinate system to the camera coordinate system, and one degree of freedom representing the distance from the origin to the line. As shown in Fig. 11, this provides a geometric interpretation of using four parameters θ in the process of updating the orthogonal representation (Suárez et al., 2018).

In point-line SLAM systems, the orthonormal representation is typically used in back-end optimization because it minimizes errors and achieves decoupling. In contrast, Plücker coordinates are more commonly used for tasks such as camera projection, line feature endpoint trimming, and initialization. The combined parameterization strategy of Plücker coordinates and orthonormal representation has been widely adopted in various studies, improving computational efficiency and robustness of the system. Although the orthonormal representation has a smaller parameterization, its projection equation is not directly linear. Therefore, further transformations (e.g., using Eq. (5)) are required for effective computation. This transformation ensures the validity of the minimal representation, which can be optimally represented using $[\varphi, \theta]^T$ in the optimization process. The advantage of using this representation in SLAM is that it not only reduces computational complexity but also enhances the stability of the optimization process, particularly in large-scale scene processing.

2.3.4. Other line feature parameterization methods

In point-line SLAM, in addition to the three commonly used methods mentioned above, recent research has proposed several innovative line feature parameterization techniques (Zhang, Zeng, & Zha, 2019; Zhao & Vela, 2018) to improve the accuracy and robustness of the system. Below are a few typical methods:

The Uniform Sampling Representation (Ma & Liang, 2021) creates map features by uniformly sampling multiple points along a line segment and calculating the reprojection error based on these points.This method introduces additional geometric constraints, effectively reducing the uncertainty in keyframe pose estimation during BA, especially in environments with low texture, rapid motion, and significant illumination changes. It enhances the localization and mapping capabilities of SLAM systems.

The Inverse Depth Line Feature Parameterization The two innovative methods optimize 3D line feature parameterization through

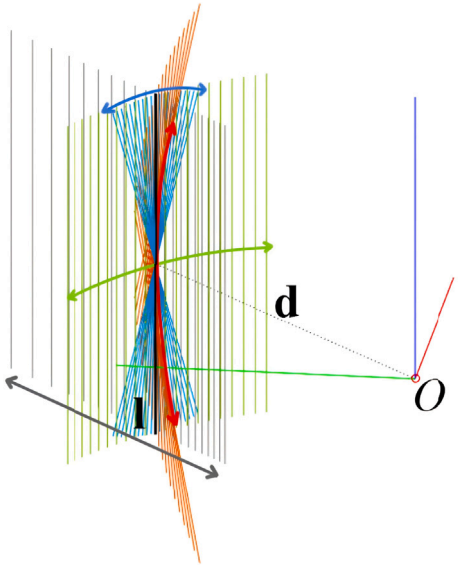


Fig. 11. Provides a geometric interpretation of using four parameters ∂_θ in the process of updating the orthogonal representation (Zuo et al., 2017).

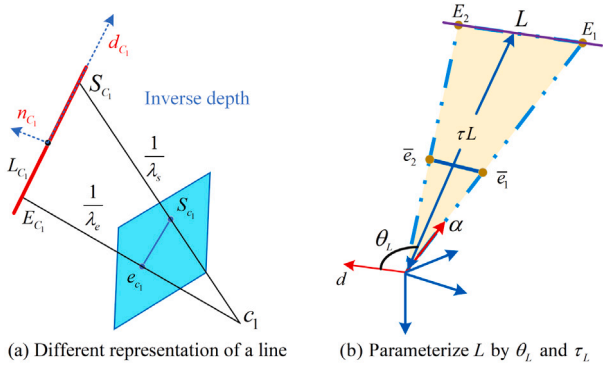


Fig. 12. Two innovative line feature representation methods (Li et al., 2023; Zhou, Wang et al., 2021).

different dimensions, as shown in Fig. 12. The inverse depth line feature parameterization (Li et al., 2023) reduces the degrees of freedom of the 3D line from six to two by using endpoint inverse depth parameters, which decreases computational complexity. Meanwhile, the simplified method based on Plücker coordinates (Zhou, Wang et al., 2021) compresses the degrees of freedom from four to two, avoiding singularities and enhancing depth estimation accuracy through the collinearity constraint. Both methods significantly improve the robustness and efficiency of vSLAM in low-texture, dynamic lighting, and fast-moving scenes.

These innovative line feature parameterization methods each have their advantages in different scenarios. The uniform sampling method is particularly suitable for environments with significant illumination changes, while the inverse depth parameterization method excels in low-texture and low-light scenes. As SLAM technology continues to advance, these methods may be further combined to enhance the system's adaptability and robustness.

2.4. Line feature matching

In point-line vSLAM systems, line feature matching is a key step, as its stability and accuracy directly affect the overall performance and robustness of the system. Line feature matching methods can

be broadly categorized into three types: descriptor-based methods, optical flow-based methods, and learning-based methods. Descriptor-based methods (Hartley, 1995; Zhang & Koch, 2013) involve generating descriptors for line features and using these descriptors for matching, such as LBD. Learning-based methods, on the other hand, employ deep learning models (Ma et al., 2020; Pautrat et al., 2021), such as Graph Convolutional Networks (GCN), to improve matching accuracy and robustness.

2.4.1. Descriptor-based methods

Descriptor-based line feature matching methods, recognized for their high matching accuracy and noise resistance, have been extensively utilized across various scenarios. In particular, the integration of the LBD with geometric information has demonstrated notable effectiveness. This approach not only provides detailed descriptions of line segments but also incorporates geometric constraints, thereby substantially enhancing the robustness and precision of feature matching.

The combination of LBD descriptors with geometric information A line feature matching method that integrates descriptors and geometric relationships was proposed in Zuo et al. (2017). This method first extracts line features using an improved LSD algorithm, reducing segmentation errors by merging line segments with similar directions and distances, thereby enhancing detection robustness and accuracy. Then, LBD descriptors are used to match the line segments, and geometric conditions such as angle difference, length ratio, and overlap length are employed to further filter the matching results. Similar approaches have been applied in Gomez-Ojeda et al. (2019), He et al. (2018), Lee and Hwang (2019), Xu, Chen, Zhang, and Wang (2020), demonstrating that combining geometric constraints can effectively improve the accuracy and robustness of matching in low-texture or complex environments.

Combination of LBD Descriptors and KNN Algorithm The KNN algorithm is widely used in descriptor-based line feature matching to improve matching accuracy and robustness. For example, Alamanos and Tzafestas (2023) effectively enhanced matching efficiency and accuracy in low-texture environments by combining LBD descriptors with the KNN method within a fixed window for stereo matching. In Yang et al. (2019), an initial matching of monocular images was optimized by combining undistortion processing with significant length constraints, while Fu et al. (2020) further reduced mismatches using LBD descriptors, an angle-thresholded KNN algorithm, and an inlier optimization strategy. Additionally, Zhao et al. (2022) optimized the traditional LSD algorithm by combining LBD descriptors and KNN, which outperformed the LSD+KLT optical flow algorithm.

Methods for Improving Robustness and Accuracy in Line Feature Matching To address the challenges related to the stability and robustness of line feature matching, researchers have proposed various innovative methods to cope with issues such as image blur, occlusion, and low-texture environments. For instance, Zhou, Wang et al. (2021) improved 3D line segment matching accuracy by combining photometric error minimization with sample point tracking. The hierarchical method proposed by Li, Yao, Lu, Li, and Zhang (2016) enhances matching accuracy in scenes with image transformations and textureless areas through local homography estimation and multi-scale pyramid processing. Seo et al. (2023) introduced a semi-dense epipolar search method that effectively reduces mismatches by limiting the search range for line features. The GLSM method (Wei, Zhang, Liu, Li, & Li, 2021) combines geometric constraints and graph sorting algorithms to further enhance the accuracy and reliability of multi-view line feature matching. To address challenges caused by lighting variations, traditional LBD descriptors and LSD algorithms exhibit instability under dynamic lighting conditions, and some studies have tackled this issue through point tracking and 2D-3D matching strategies (Di Stefano, Mattoccia, & Tombari, 2005; Xu et al., 2022). Furthermore, Xu et al. (2023) addressed the matching problem in complex lighting environments by associating point features and line segments through the distance between points and line segments, and performing line matching based on point matching results,As shown in Fig. 13.

Fig. 13. Line Matching in Challenging Scenes by Xu et al. (2023). The matched lines are depicted in the same color. Circles on the lines represent points associated with each line.

Overall, descriptor-based line feature matching methods effectively address challenges such as image blur, occlusion, and low-texture environments by combining geometric constraints with efficient descriptors. Although some instability remains under dynamic lighting conditions, researchers have significantly improved the performance of line feature matching through innovative algorithms and strategies.

2.4.2. Optical flow-based methods

Optical flow plays a crucial role in feature point and line matching by estimating the motion vector field of pixels in an image sequence. Classical optical flow algorithms, such as the Lucas–Kanade optical flow method (Lucas & Kanade, 1981) and the Pyramidal Lucas–Kanade optical flow method (Bouguet et al., 2001), are widely used in feature tracking. By leveraging the continuity of optical flow vectors and the geometric information of line segments, optical flow methods can accurately track line features in images, overcoming challenges such as translation, rotation, lighting variations, and occlusion. As a result, optical flow methods have significant applications in vSLAM.

Combination of Optical Flow and Line Segment Descriptors Several optical flow-based line feature matching methods have been proposed to address challenges in low-texture, complex scenes, and rapid motion. For instance, Zhu et al. (2021) combined an improved EDLines algorithm with the KLT optical flow method (Tomasi & Kanade, 1991), enhancing the accuracy and robustness of SLAM systems in low-texture and complex environments. An innovative method combining optical flow and LBD was proposed in Lim et al. (2021), where initial matching is performed using LBD, and optical flow algorithms are then employed to predict and correct mismatches. The final best matching results are selected by comparing the matching scores. This method improves the robustness and continuity of the matching process and addresses the degradation caused by translational camera motion through structural constraints (Hartley & Zisserman, 2003; Ok et al., 2012; Sugiura, Torii, & Okutomi, 2015; Yang et al., 2019).

IMU-assisted Optical Flow Methods To address the challenges posed by rapid motion and illumination changes, the "prediction-matching" strategy proposed in Wei, Tang, Xu, Zhang, and Wu (2021) significantly enhances the tracking length and matching accuracy of line segments in weak texture and motion-blurred scenarios, utilizing an IMU-assisted KLT algorithm and geometric constraints. In response to the limitations of traditional optical flow tracking (Baker & Matthews, 2004) in complex scenarios involving illumination changes and rapid camera motion, the method presented in Liu et al. (2023)

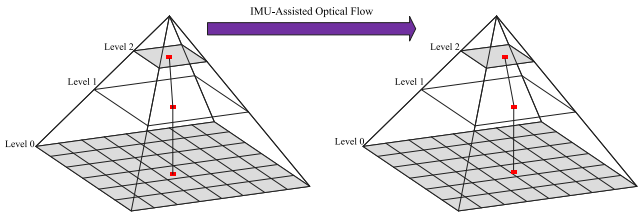


Fig. 14. IMU-Assisted Hierarchical Grid Optical Flow Tracker (Liu et al., 2023).

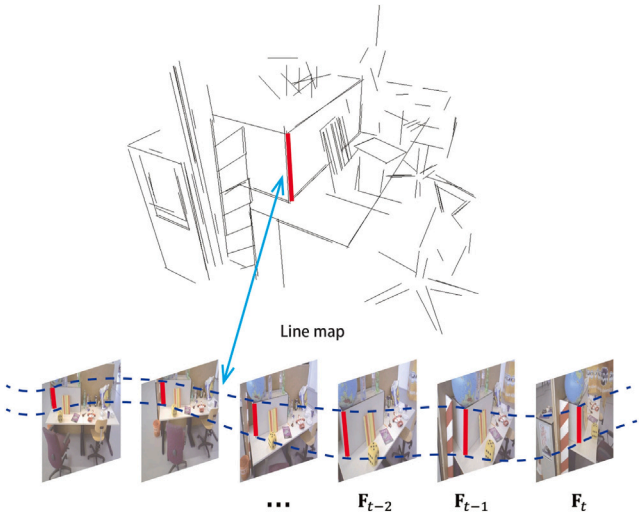


Fig. 15. Line flow in consecutive images. All line segments in the flow correspond to the same 3-D line (Wang et al., 2021).

introduces an IMU-assisted hierarchical optical flow tracking approach (as shown in Fig. 14). This method integrates IMU data and the RANSAC algorithm (Fischler & Bolles, 1981), predicting the starting and ending positions of line features in the current frame and calculating the Hamming distance to find the best matching points, thereby effectively improving matching accuracy and reducing computational load.

Introduction of Line Flow Techniques To address the real-time challenges in SLAM systems, researchers have proposed line flow-based matching methods that enhance performance in scenarios with occlusion, blur, and repetitive textures by predicting and updating the continuous 2D projections of line segments (Wang et al., 2021), as shown in Fig. 15. Although this method simplifies the process of line feature description and matching, reducing computational complexity, it still requires line extraction and matching for each frame, resulting in performance that lags behind traditional methods. In contrast, the LOFT algorithm achieves efficient single-frame detection and multi-frame tracking by utilizing gray-invariance and collinearity constraints, showing superior performance, particularly in environments with repetitive textures (Xu, Yin et al., 2022).

This section discusses the innovative applications of optical flow methods in point-line vSLAM systems. These methods effectively address matching issues in low-texture, complex scenes, and lighting variations by integrating geometric information from optical flow, IMU-assisted tracking, and line flow matching techniques. Despite certain limitations, the research progress provides significant technical support for the further enhancement of vSLAM systems.

2.4.3. Learning-based methods

In recent years, deep learning-based line feature matching methods have made significant progress in various fields. The GLSP method (Chen & Pan, 2023) utilizes Graph Neural Networks (GNN) for line segment matching, while the AirLine method (Lin & Wang, 2023) proposes a learnable edge-based line detection algorithm suitable for real-time applications. Additionally, the semantic line combination detector improves line segment detection accuracy in complex images through semantic analysis (Ko, Jin, & Kim, 2024). For specific scenarios, Line-Transformers introduces the Transformer model to optimize line segment matching in scenes with long lines and uneven feature point distributions (Yoon & Kim, 2021). Moreover, the GlueStick method (Ma et al., 2021), which combines CNNs and GCNs, has made significant progress in handling unmatched line segments and image transformations. This method integrates keypoint and line segment matching and leverages graph neural networks to enhance the robustness and accuracy of image matching tasks.

It is worth noting that deep learning techniques have significantly improved line feature matching performance in point-line SLAM. For example, the line segment matching method based on Graph Convolutional Networks (GCN) (Ma et al., 2020) learns line segment descriptors through end-to-end training and utilizes top-k pooling techniques, demonstrating excellent performance. Under conditions of rotation, blur, and scale variation, the line segment descriptors generated by the lightweight fully convolutional neural network method (Vakhitov & Lempitsky, 2019) show efficient computation and matching accuracy, outperforming traditional descriptors. To address the issues of scene occlusion and inconsistent line segment lengths, Liu, Cao et al. (2024) proposed a matching strategy based on line segment collinearity, improving matching efficiency and accuracy through 2D-2D and 2D-3D line segment matching. Moreover, the matching mechanism based on Attention GNN (Kannapiran et al., 2023) effectively enhances matching precision and optimizes the matching results through the Sinkhorn algorithm (Sinkhorn, 1964), ensuring efficient and accurate line segment matching, as shown in Fig. 16. Fig. 17 illustrates the performance of three line feature detection matching methods: LSD + LBD provides reliable detection but may suffer from line segment fragmentation; LSD + KLT improves cross-frame tracking, but the continuity and accuracy of line segments in dynamic environments need further enhancement; DeepLSD performs exceptionally well in complex and low-texture scenarios, demonstrating stronger line segment detection and robustness, making it particularly suitable for dynamic environments.

Although these methods demonstrate superior performance under specific conditions, their application on resource-constrained platforms remains limited due to high computational resource requirements. Therefore, further innovations are needed in this field to improve efficiency and address the challenges in practical applications.

3. Back-end modules

The backend optimization of SLAM aims to eliminate the accumulated errors from the frontend by enforcing global consistency constraints, serving as a core component in enhancing the system's robustness and accuracy. Depending on the optimization strategy, backend optimization can be classified into three categories: filter-based optimization, graph optimization, and BA optimization. Although BA can be considered a subset of graph optimization, its specificity and widespread application in vSLAM make it an independent research direction. Table 6 provides a systematic comparison of the core differences among these three methods, covering aspects such as mathematical models and applicable scenarios.

3.1. Filter optimization

The filter-based optimization is a core technology for achieving real-time state estimation in point-line SLAM systems (Kannapiran, Van Baar, & Berman, 2021), as it balances computational efficiency and accuracy through recursive data fusion. This section systematically reviews filter-based optimization methods, analyzing them from three aspects: the basic framework, consistency improvements, and

Table 6
Comparison of optimization methods in vSLAM systems.

Method	Filter optimization	Graph optimization	Bundle adjustment
Mathematical Models	State Equation and Observation Equation Integration, Kalman Gain Update: $x_{k\|k} = x_{k\|k-1} + K_k(z_k - h(x_{k\|k-1}))$	Nonlinear Least Squares: $X^* = \arg\min \sum \|e_i(X)\|^2$	Nonlinear Least Squares: $X^* = \arg\min \sum \|\pi(T_i, P_j) - z_{ij}\|^2$
Optimization Variables	IMU prediction error, 3D point/line features, local geometric parameters.	Camera pose, 3D point/line features, IMU parameters, sensor biases, dynamic object states, etc.	Camera pose, 3D point/line features.
Constraint Types	IMU prediction error, Point/line reprojection error, Geometric priors.	Reprojection error, IMU pre-integration, Loop closure detection, Geometric priors.	Point/line reprojection error.
Typical Application Scenarios	Systems with high real-time requirements, VIO.	Global optimization, complex dynamic scenes.	Local/global optimization in vSLAM, static scene reconstruction.
Advantages	High computational efficiency, Low memory usage, Adaptability to dynamic environments.	Global consistency optimization, Strong multi-sensor fusion capability, Support for complex constraints.	High visual optimization accuracy, Relatively high computational efficiency (static scenes).
Disadvantages	Local optimization leads to poor global consistency, Linearization errors (EKF), Long-term accumulation of errors.	High computational complexity, Limited real-time performance.	Inability to integrate data from other sensors, Sensitivity to dynamic environments.

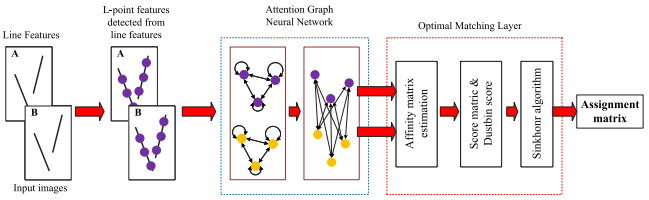
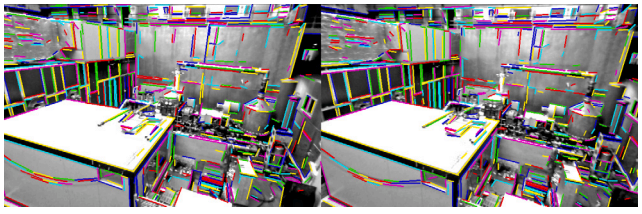
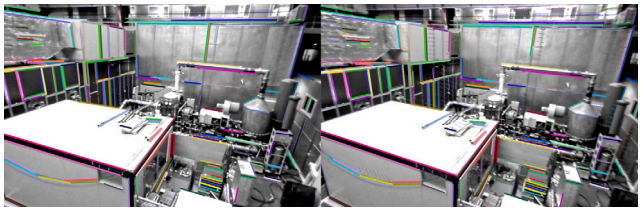


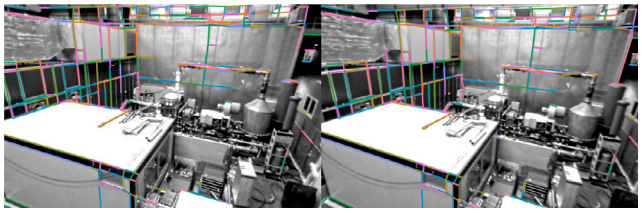
Fig. 16. Architecture of point and line feature matching (Kannapiran et al., 2023).



(a) LSD+LBD



(b) LSD+KLT



(c) DeepLSD

Fig. 17. Performance Evaluation of Three Line Feature Detection and Matching Methods.

feature-specific strategies for line features. Additionally, performance boundaries are quantified through experimental data.

Fundamental Filter Framework and Numerical Models The EKF performs state propagation and update by linearizing the nonlinear system model, with its core steps including:

- **state prediction:** Based on the IMU dynamic model, the system state is predicted as $x_{k|k-1} = f(x_{k-1}, u_k) + w_k$, where $x = [T_{cw}, v, b_g, b_a, L_w]^T$ represents the camera position, velocity, IMU biases, and linear feature parameters, and $w_k \sim \mathcal{N}(0, Q_k)$ is the process noise.

- **observation update:** The line feature observation model is given by $z_L = h(T_{cw}, L_w) + n_k$, where $n_k \sim \mathcal{N}(0, R_k)$. The observation process is linearized using the Jacobian matrix $H_k = \frac{\partial h}{\partial x}$, the Kalman gain K_k is computed, and the state covariance is updated:

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \tag{7}$$

$$x_{k|k} = x_{k|k-1} + K_k(z_k - h(x_{k|k-1})) \tag{8}$$

The Multi-State Constraint Kalman Filter (MSCKF) (Mourikis & Roumeliotis, 2007) maintains historical states within a sliding window and optimizes line features across multiple frames of observation, avoiding the inclusion of features in the state vector, which significantly reduces computational complexity. Its objective function is:

$$\min_{L_w} \sum_{i=1}^{N} \left\| z_L^{(i)} - h(T_{cw}^{(i)}, L_w) \right\|_{R^{-1}}^2 \tag{9}$$

where N is the number of frames in the window, and $T_{cw}^{(i)}$ is the historical relative pose.

Consistency improvement and invariance filtering Traditional EKF suffer from inconsistency due to linearization errors, which manifest as false information gains in unobservable directions (Heo, Jung, & Park, 2018). The Invariant Extended Kalman Filter (IEKF) (Bonnabel, 2007) addresses this issue by defining invariant errors on Lie groups, and it has been widely applied and improved in a series of studies (Bailey, Nieto, Guivant, Stevens, & Nebot, 2006; Barrau & Bonnabel, 2015; Hua, Li, & Pei, 2023; Huang & Dissanayake, 2007; Huang, Mourikis, & Roumeliotis, 2008, 2010; Li & Mourikis, 2013).

- Invariant Error Model Define the error state $\eta = [\delta\theta^T, \delta p^T, \delta v^T, \delta b_g^T, \delta b_a^T]^T$, whose dynamics satisfy:

$$\dot{\eta} = F\eta + Gw, \tag{10}$$

where F is the error propagation matrix, and G is the noise propagation matrix.

- **Observation Invariance** The line feature residual is designed to be invariant under rigid body transformations, such as the projection error based on Plücker coordinates:

$$e_L = \frac{n_c^T(x_s \times x_e)}{\|n_c\|}, \tag{11}$$

where n_c is the line feature normal vector, and x_s, x_e are the endpoints of the line segment.

IEKF has been integrated into frameworks such as the Multi-State Constraint Kalman Filter (MSCKF) (Heo et al., 2018) and the Unscented Kalman Filter (UKF) (Brossard, Bonnabel, & Barrau, 2018a). Early studies (Brossard, Bonnabel, & Barrau, 2018b) also explored the combination of IEKF with MSCKF to handle landmark uncertainty. Although these methods have made progress in improving filter consistency, the relationships among them require further investigation. In paper (Hua et al., 2023), an IEKF-based VIO system was proposed, incorporating point, line, and vanishing point features to enhance localization accuracy and system consistency in man-made environments. By defining alternative nonlinear errors, the IEKF effectively addresses the inconsistency issues present in the traditional EKF framework and ensures the correct dimensionality of the unobservable subspace. Furthermore, the study demonstrated that this approach maintains system consistency in a decoupled state using conventional error representation.

Line feature-specific optimization strategy: In Zou et al. (2019), the line feature optimization employed the EKF, using line features as external measurement data, and performing local Manhattan world assumption to minimize the measurement error with newly updated parameters. The approach improved the precision and robustness of the system by using system information aggregation, and applied measurements to the line's prior distribution. StructVIO assumed local environment constraints and the harmonic structure, observing the line orientation relative to the coordinate axes. The residual error is added as an orthogonal correction term:

$$e_{\text{ortho}} = v_L^T a_M, \tag{12}$$

where v_L is the line orientation, and a_M is the harmonic axis.

Traditional filtering optimization methods typically use point features for position estimation, neglecting the influence of line features. To address this issue, Chen, Miao, Liu, Wu, and Wang (2024) proposed a hybrid method that combines point-line feature fusion with the Extended Kalman Filter (hybrid MSCKF). This approach integrates the MSCKF framework with line features, enabling SLAM state estimation:

State Vector Expansion:

$$x = [x_{\text{leg}}, x_{\text{pos}}, x_{\text{p}}, x_l]^T, \tag{13}$$

where $x_l = [O_1, O_2, \ldots, O_K]^T$ represents the line feature state.

Observation Update: The line feature residual model is:

$$r_k = H_l \delta x_l + H_{xe} \delta x_{\text{pos}} + n_k, \tag{14}$$

where H_l is the Jacobian matrix, which can be solved through linearization to estimate the line feature error, position, and external parameters.

This method fully utilizes the structural information of line features, especially in weakly dynamic or noisy environments, improving both the precision and robustness of the estimation.

Filter-based optimization performs exceptionally well in real-time applications; however, due to its local optimization nature, it is prone to error accumulation over long-term operation. Future research should focus on nonlinear observation models, such as the Unscented Kalman Filter (UKF) or second-order Taylor expansions, to reduce linearization errors. Additionally, dynamic feature processing should be integrated to remove dynamic line features, and multi-sensor fusion should be explored to enhance adaptability in complex environments. Although graph optimization and bundle adjustment are increasingly becoming mainstream, filter-based optimization still holds advantages in resource-constrained scenarios.

3.2. Graph optimization

Graph optimization is a core technique for ensuring global consistency in point-line feature fusion SLAM systems. By modeling camera poses, environmental features, and multi-source constraints as a graph structure, it enables high-precision joint optimization in large-scale environments. Compared to filter-based optimization, graph optimization can globally adjust all variables, significantly suppress accumulated errors, and improve map consistency after loop closure detection. In graph optimization, nodes represent camera poses or feature points in the map, while edges denote the geometric constraints between camera poses or between camera poses and feature points. By minimizing the error of these geometric constraints, graph optimization jointly optimizes the positions of all nodes, significantly improving the accuracy of pose estimation and map reconstruction. Commonly used graph optimization algorithms include g2o (Kümmerle, Grisetti, Strasdat, Konolige, & Burgard, 2011) and Ceres Solver (Agarwal, Mierle, et al., 2012), which are capable of efficiently handling large-scale maps and multi-sensor data, making them particularly suitable for global optimization after loop closure detection. This section provides an analysis from three aspects: mathematical modeling, system implementation, and experimental data comparison.

Zuo et al. (2017) first systematically derived the analytical Jacobian matrix of the line feature reprojection error with respect to pose and line parameters, addressing the efficiency and accuracy bottlenecks of traditional numerical differentiation methods. In graph optimization frameworks, the integration of line features relies on precise error term modeling and efficient Jacobian matrix computation. The graph model is constructed with camera pose nodes $\boldsymbol{T}_{k,w}$, 3D point nodes $\boldsymbol{X}_{w,t}$, and line feature nodes $\boldsymbol{\mathcal{L}}_{w,j}$, with key error constraints defined as follows:

- Reprojection Errors for Points and Lines
The following two equations represent the Point Feature Error (Geometric Projection) and the Line Feature Error (Plücker Coordinates Projection), respectively.

$$\mathbf{e}_{p_{k,i}} = \boldsymbol{x}_{k,i} - \mathbf{K}\boldsymbol{T}_{k,w}\boldsymbol{X}_{w,t}, \tag{15}$$

where $\boldsymbol{x}_{k,i}$ is the image point coordinate, and $\mathbf{K}$ is the camera intrinsic matrix.

$$\mathbf{e}_{l_{k,j}} = \mathrm{d}\left(\boldsymbol{x}_{k,j}, \mathcal{K}_{\text{th}}[\mathcal{H}_{\text{cw}}\boldsymbol{\mathcal{L}}_{w,j}]\right), \tag{16}$$

where $\mathrm{d}(\cdot)$ is the endpoint distance between the detected line segment and the projected line, and $\mathcal{H}_{\text{cw}}$ is the transformation matrix from the world to camera frame.

- Unified Cost Function
Robust Huber kernels ρ_π and ρ_Γ are introduced to suppress outliers:

$$C = \sum_{k,i} \rho_\pi\left(\mathbf{e}_{p_{k,i}}^\top \boldsymbol{\Sigma}_{p_{k,i}}^{-1} \mathbf{e}_{p_{k,i}}\right) + \sum_{k,j} \rho_\Gamma\left(\mathbf{e}_{l_{k,j}}^\top \boldsymbol{\Sigma}_{l_{k,j}}^{-1} \mathbf{e}_{l_{k,j}}\right). \tag{17}$$

- Analytical Jacobian Derivation for Line Features
To efficiently solve the nonlinear optimization, analytical Jacobians of line feature errors with respect to pose increments $\delta_\xi = [\delta_\rho^\top, \delta_\phi^\top]^\top$ and line parameters δ_θ are systematically derived:

Error Derivative with Respect to Projected Line.

$$\frac{\partial \mathbf{e}_l}{\partial \boldsymbol{l}'} = \frac{1}{l_n}\begin{bmatrix} u_1 - \frac{l_1 e_1}{l_1^2} & v_1 - \frac{l_2 e_1}{l_n^2} & 1 \\ u_2 - \frac{l_1 e_2}{l_n^2} & v_2 - \frac{l_2 e_2}{l_n^2} & 1 \end{bmatrix}_{2\times 3}, \tag{18}$$

where $\boldsymbol{l}' = [l_1, l_2, l_3]^\top$ is the projected line parameter, $l_n = \sqrt{l_1^2 + l_2^2}$, and e_1, e_2 are endpoint projection errors.

Chain Rule Decomposition and Lie Algebra Transformation. By decomposing the pose increment δ_ξ into translation δ_ρ and rotation δ_ϕ via Lie group-algebra mapping, the Jacobians of Plücker coordinates are derived:

$$\frac{\partial \mathcal{L}_c^*}{\partial \delta_\rho} = \begin{bmatrix} -[\mathbf{R}_{cw}\boldsymbol{v}_w]_\times \\ \mathbf{0} \end{bmatrix}, \tag{19}$$

$$\frac{\partial \mathcal{L}_c^*}{\partial \delta_\phi} = \begin{bmatrix} -[\mathbf{R}\boldsymbol{u}]_\times - [\boldsymbol{t}]_\times[\mathbf{R}\boldsymbol{v}]_\times \\ -[\mathbf{R}\boldsymbol{v}]_\times \end{bmatrix}. \tag{20}$$

$$\frac{\partial \mathcal{L}_c^*}{\partial \delta_\xi} = \begin{bmatrix} -[\mathbf{R}\boldsymbol{u}]_\times - [\boldsymbol{t}]_\times[\mathbf{R}\boldsymbol{v}]_\times & -[\mathbf{R}\boldsymbol{v}]_\times \\ -[\mathbf{R}\boldsymbol{v}]_\times & \mathbf{0} \end{bmatrix}_{6\times 6}. \tag{21}$$

Chain Rule Propagation: The Jacobians propagate to line parameter updates through the chain rule.

$$\mathbf{Jl}_\xi = \frac{\partial \mathbf{e}_l}{\partial \delta_\xi} = \frac{\partial \mathbf{e}_l}{\partial \boldsymbol{l}'}\frac{\partial \boldsymbol{l}'}{\partial \mathcal{L}_c}\frac{\partial \mathcal{L}_c}{\partial \delta_\xi}, \tag{22}$$

$$\mathbf{Jl}_\theta = \frac{\partial \mathbf{e}_l}{\partial \delta_\theta} = \frac{\partial \mathbf{e}_l}{\partial \boldsymbol{l}'}\frac{\partial \boldsymbol{l}'}{\partial \mathcal{L}_c}\frac{\partial \mathcal{L}_c}{\partial \mathcal{L}_w}\frac{\partial \mathcal{L}_w}{\partial \delta_\theta}. \tag{23}$$

Eqs. (22)–(23) ensure efficient updates of line parameters during iterative optimization.

PLS-VIO (Wen, Tian, & Li, 2020) is the first stereo visual-inertial odometry system to leverage point-line feature fusion for optimization. This system employs a sliding window model that integrates IMU residuals, as well as point and line feature reprojection errors, optimizing the objective function using the Dog-Leg algorithm, which significantly enhances the system's accuracy and robustness. Gomez-Ojeda and Gonzalez-Jimenez (2016) optimizes the line features by minimizing the projection errors between consecutive stereo frames and applies Maximum Likelihood Estimation (MLE), combined with a pseudo-Huber loss function to improve robustness. In addition, Lim et al. (2021) utilizes the Ceres Solver to construct a nonlinear cost function, incorporating various measurement and structural constraints to accurately estimate and optimize 3D line features. Furthermore, Xu et al. (2023) proposes a line feature optimization method by measuring the angle between plane normals as residuals, which avoids the limitations of traditional line segment endpoints and significantly improves accuracy. By incorporating these residuals into graph optimization and employing the Ceres Solver for sliding window optimization, joint optimization of camera poses and the map is achieved. Xu et al. (2023) further improves the overall system performance by constructing reprojection errors for both points and lines to optimize the positions of keyframes, 3D points, and lines.UPLP-SLAM (Yang, Yuan, Gao, Sun, & Zhang, 2023) was the first to introduce the SP model into the optimization framework, proposing a unified multi-feature tightly coupled joint optimization framework. The aim is to achieve joint optimization of feature parameters and camera poses by uniformly representing point features, line features, plane features, and camera poses, while constructing a consistent error function. Compared to existing multi-feature SLAM methods (Gomez-Ojeda et al., 2019; Li, Yunus, Brasch, Navab, & Tombari, 2021; Yunus, Li, & Tombari, 2021), this system can handle a broader range of features and provides more accurate and consistent estimation results. The backend optimization framework of SG-VIO (Yao et al., 2024) adopts a factor graph-based multi-sensor joint graph optimization method, significantly extending the scope of traditional BA optimization, as shown in Fig. 18. Its core lies in unifying the reprojection errors of visual features, IMU pre-integration constraints, and geometric priors of structured line features (e.g.,lines parallel or perpendicular to the gravity direction) within the factor graph structure. State variables are managed through a sliding window, and a Schur complement edge marginalization strategy is employed to control computational complexity. Structured lines are integrated with gravity direction constraints and IMU fusion to achieve collaborative optimization of multi-source heterogeneous data. Compared to traditional BA, the optimization framework of SG-VIO not only enhances robustness in low-texture environments but also effectively balances global consistency and real-time demands through a two-stage marginalization strategy.

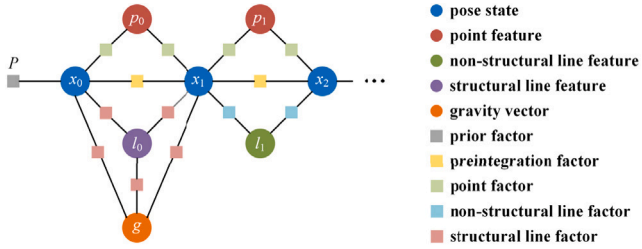


Fig. 18. Typical Factor Graph Example (Yao et al., 2024).

Graph optimization, through global constraint fusion and multi-feature collaboration, has become a core pillar for high-precision mapping in point-line SLAM. Despite significant progress in structured environments, the parameterization ambiguity of line features and dynamic adaptability remain key challenges. Future research should integrate incremental computation, learning techniques, and hardware acceleration to drive practical breakthroughs in graph optimization for open dynamic environments.

3.3. Bundle adjustment optimization

LBD as the standard method for multi-view geometry optimization, plays a dual role in point-line fusion SLAM systems: it serves both as the core subproblem in graph optimization and as an independent optimization paradigm for achieving high-precision state estimation. Traditional BA constructs a graph model of camera poses and 3D features, performing global optimization with reprojection errors as constraints. In the point-line fusion framework, however, BA optimization faces a new challenge of dimensional expansion—it must coordinate the geometric constraints of heterogeneous features (discrete points and continuous line segments) in a unified mathematical space. This introduces opportunities for precision improvement but also poses challenges in computational complexity.

Recent research has demonstrated that the integration of line features with point features in BA optimization has yielded significant results. For example, Zhou et al. (2022) extended the traditional photometric error model to include line features, proving that the 3D coordinates of any point on a 2D line can be determined by the inverse depth of the 2D line endpoints. This approach reduces the number of optimization variables and improves computational efficiency. The study employed a sliding window optimization strategy, combining photometric errors of points and lines with the collinearity constraints of line features to construct a comprehensive cost function, which was optimized using a two-step minimization process. First, 3D line segments were fitted, and then the fitting parameters were used to update the camera pose and inverse depth until convergence.BA optimizes the camera pose and map point depth by minimizing the reprojection error of point features. However, the stability of depth estimation depends on the number of tracked frames. When the number of tracked frames is small, depth estimation can be easily influenced by noise, leading to instability. To address this issue, Lee and Park (2021) proposed the Point-Line Coupling Residual (PLC Residual), as shown in Fig. 19. This method enhances the robustness of BA optimization by

introducing geometric constraints from line features, and constructs the optimization objective using the geometric relationship between points and lines. In another study (Yu et al., 2023), a unified cost function incorporating both point and line feature reprojection errors was constructed, achieving global optimization of the camera pose. This work demonstrated that DPL-SLAM significantly outperformed traditional methods in terms of localization accuracy in low-texture environments, validating the effectiveness of point-line feature fusion optimization. Additionally, the system proposed in Gomez-Ojeda et al. (2016) first estimates the camera motion between consecutive frames through direct image alignment and optimizes the line segment endpoints to minimize photometric error. This is followed by feature alignment to further optimize the positions of 2D line segment endpoints, ultimately achieving optimization of the camera pose and 3D structure by minimizing the reprojection error between 3D and 2D features. This method not only improves the accuracy of line feature matching but also reduces computational complexity, making the optimization of line features more efficient and robust. Vakhitov and Lempitsky (2019) introduced a method that incorporates line features into ORB-SLAM2, achieving line feature backend optimization by combining point and line features through Local BA. The ALVIO system (Jung, Kim, Lim, & Myung, 2021) integrates the reprojection errors of point and line features with IMU pre-integration errors into a unified cost function, achieving precise pose estimation through iterative optimization. The ORB-LINE-SLAM system (Alamanos & Tzafestas, 2023) combines multiple error functions to optimize camera poses and feature positions in the map. Building on the strength of traditional BA methods for point features, this study designed error functions specifically for line features and introduced a novel adjustment factor to balance the participation of point and line features in the BA process, resulting in enhanced stability and robustness. Similarly, the PL-VIO system (He et al., 2018) uses a sliding window optimization framework to simultaneously optimize IMU states, along with point and line feature parameters, utilizing IMU pre-integration as a constraint to avoid redundant integration, thereby improving computational efficiency. The system minimizes IMU pre-integration error and the reprojection errors of point and line features, and employs the Levenberg–Marquardt algorithm to achieve efficient fusion of visual and inertial sensor information. PL-VINS (Fu et al., 2020) is the first real-time monocular VINS method that integrates point and line features based on real-time optimization. It adopts a sliding window strategy and achieves high accuracy and real-time performance by jointly minimizing the reprojection residuals of point and line features, along with IMU measurement residuals, within a fixed-size window. In PL-VIO (He et al., 2018) and PL-VINS (Fu et al., 2020), the line reprojection error is defined by calculating the distance between the observed line segment endpoints and the reprojection of those endpoints, with this error being related to the accuracy of the line endpoints. However, due to issues such as inconsistent length of line features in space, variations in length on the image plane, and occlusions, traditional methods may result in large errors. Therefore, a new method was proposed in Xu, Yin et al. (2022), as shown in Fig. 20, which overcomes these issues by constructing residuals that are independent of the line segment endpoints. This method uses the line equation on the normalized plane to express the line reprojection error and computes the sensitivity of the error to state variables through the Jacobian matrix. In PLD-VINS (Zhu et al., 2021), BA is used for backend optimization of line features within the sliding window, and a keyframe extraction strategy is applied to effectively control the scale of BA optimization, enhancing the system's accuracy and robustness. Several studies have also proposed new optimization improvements. For instance, Zhou, Wang et al. (2021) introduced collinearity constraints to address depth estimation issues in line features during photometric error minimization, reducing the 4 degrees of freedom of 3D lines to 2 degrees of freedom, thereby decreasing computational complexity. This research significantly improved the performance and real-time capabilities of line features in direct visual odometry by utilizing collinearity priors and optimized 3D line representation.

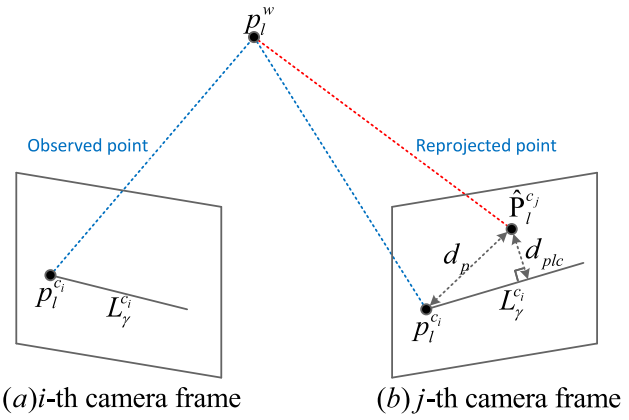


Fig. 19. Minimization distance error model between point and line (Lee & Park, 2021).

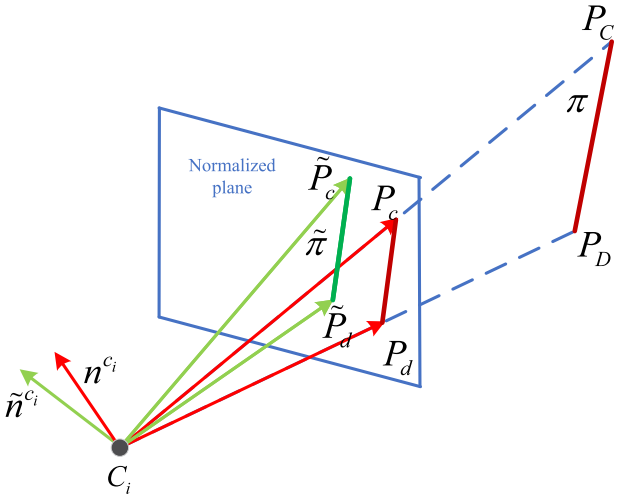


Fig. 20. Computation of the line reprojection error (Xu, Yin et al., 2022).

Table 7 further compares recent representative systems in terms of their methods, advantages, and disadvantages in BA (Bundle Adjustment) optimization. In summary, recent studies have demonstrated that by integrating point and line features, employing sliding window strategies, and efficiently representing line features in BA optimization, the performance, accuracy, and robustness of vSLAM systems have been significantly enhanced. These advancements have further promoted the practicality and application potential of SLAM technology in complex environments.

4. Loop closure detection

In SLAM problems, pose estimation is typically a recursive process, where the pose of the current frame is derived from the previous frame. Consequently, errors accumulate frame by frame over time, affecting the overall accuracy of the system. Loop closure detection is an effective method to eliminate such accumulated errors by identifying whether the robot has returned to a previously visited location. This process provides important constraints for back-end optimization, thereby creating a topologically consistent trajectory map and improving the accuracy and robustness of the SLAM system. Although traditional point feature-based SLAM loop closure detection methods perform well in various scenarios, their robustness and accuracy are often limited in environments with drastic lighting changes (Cai et al., 2024; Hu, Qi, Ding, Liu, & Zhao, 2024; Liu, Hu et al., 2024), significant viewpoint variations (Lin, Zhou, Liu, & Zhu, 2024), or sparse textures (Yan et al.,

Table 7
Comparative analysis of BA optimization methods in vSLAM systems integrating point and line features.

Paper	BA optimization method	Advantages	Disadvantages
DPL-SLAM (Yu et al., 2023)	Jointly optimizes point and line features in BA, combining depth features with line reprojection errors; adopts a two-step optimization method to improve efficiency.	Combines deep learning feature extraction with traditional BA, enhancing robustness in dynamic environments.	High computational cost, relying on effective feature matching.
UV-SLAM (Lim et al., 2022)	Integrates vanishing point measurements with line features to improve mapping accuracy; uses a BA-based method to minimize reprojection errors.	Effectively processes point and line features, accurately capturing structural information with vanishing points.	High computational complexity, may require extensive parameter tuning.
PLI-VINS (Zhao et al., 2022)	Incorporates line feature reprojection errors into a sliding window BA model; employs a 4-parameter orthogonal representation for efficient optimization.	Achieves real-time integration of line features, reducing unnecessary calculations.	Precision may degrade in low-texture environments and is sensitive to initialization errors.
PLF-VINS (Lee & Park, 2021)	Utilizes point-line coupling residuals and parallel line constraints in the BA framework, optimized using the Levenberg–Marquardt (LM) algorithm.	Effectively fuses point and line features, handling parallel line constraints well, achieving high accuracy.	Ineffective in sparse environments, with parallel line constraints increasing complexity.
PLD-VINS (Zhu et al., 2021)	Employs a sliding window method combined with line reprojection error models and uses IMU pre-integration to enhance optimization.	Efficiently handles IMU data and line features, reducing computational complexity.	Accuracy may be affected in dynamic scenes, requiring precise IMU calibration.
IF-SLAM (Wang et al., 2021)	Uses an incremental optimization method based on Bayesian networks, combining short-term and long-term BA to achieve precise line feature tracking and mapping.	Provides comprehensive line feature tracking and mapping, adapting well to complex environments.	Maintaining consistency in large-scale environments poses challenges, with high computational demand.

2024). To address these limitations, some studies have proposed SLAM loop closure detection strategies that integrate both point and line features. Compared to single point features, line features are more effective in describing complex geometric structures and offer more stable detection results under varying lighting conditions and viewpoints. By combining point and line features, it is possible to leverage the strengths of both, thereby significantly enhancing the accuracy and robustness of loop closure detection.

Currently, loop closure detection in point-line integrated vSLAM primarily employs two methods: BoW-based and Deep Learning (DL)-based approaches. Each of these methods has its distinct characteristics and application scenarios. The BoW-based method focuses on rapid matching using existing visual vocabularies, while the DL-based method enhances recognition accuracy through powerful feature extraction capabilities. The following discussion will delve into the specific implementations of these methods and evaluate their performance and effectiveness in practical applications.

4.1. Bag-of-words (BoW)-based methods

The point-line integrated vSLAM systems have extended the traditional BoW framework (Csurka, Dance, Fan, Willamowski, & Bray, 2004) by incorporating line segment features. These systems utilize both point and line features during loop closure detection, achieving more precise feature matching and improved robustness. This section explores the application of BoW models that integrate point and line features in SLAM loop closure detection, analyzing how the combination of these features enhances system performance under various environmental conditions. The BoW model converts images into vectors, focusing solely on the presence of features in the image. A typical BoW-based loop closure detection algorithm is illustrated in Fig. 21, where the BoW model trained on a dataset is shown in the green region on the left, while the front-end descriptors are in the yellow region on the right, indicating that the descriptors for the query image can be obtained through front-end computation.

Yang et al. (2016), Zuo et al. (2017) investigated the integration of line features and BoW techniques in vSLAM systems to enhance loop closure detection. In Zuo et al. (2017), a visual BoW approach was used to cluster ORB point features and LBD line features into a vocabulary. During loop closure detection, the system refined the matching between new keyframes and candidate keyframes through temporal consistency checks (Mur-Artal & Tardós, 2014) and calculated the SE(3) transformation matrix using the EPnP algorithm (Lepetit, Moreno-Noguer, & Fua, 2009) and RANSAC (Fischler & Bolles, 1981). Once a loop was detected, pose graph optimization was performed, followed by global BA to further reduce drift errors and ensure global consistency and accuracy. Gomez-Ojeda et al. (2019) extended the traditional BoW model (Gálvez-López & Tardos, 2012) by incorporating line segment descriptors to improve loop closure detection performance. Separate visual vocabularies were constructed for point and line features (Cadena et al., 2016). During loop closure detection, the system conducted dual searches within both point and line segment vocabularies and combined the matching results using a weighted strategy. The overall similarity score accounted for the number and distribution of features, thereby enhancing the robustness and accuracy of loop closure detection. For loop correction, the system employed Pose Graph Optimization (PGO) to adjust trajectory errors, utilizing the g2o library to solve the optimization problem, and ultimately updated the poses of keyframes and landmarks, merging local maps on both sides of the loop. This method, which combines point and line features, significantly improves the accuracy and robustness of loop closure detection and correction across various scenarios, particularly in low-texture and complex environments. Fig. 22 illustrates the loop closure

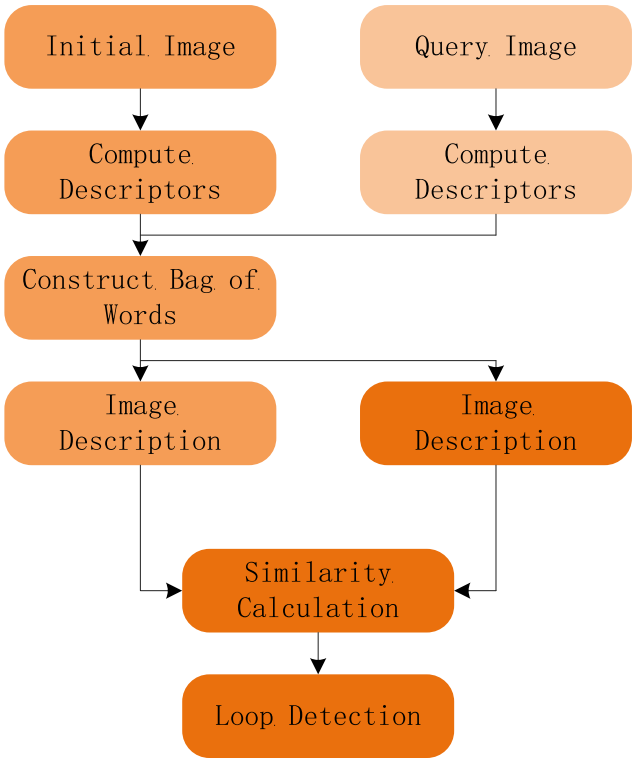


Fig. 21. Logic diagram of loop detection method based on bag of words.

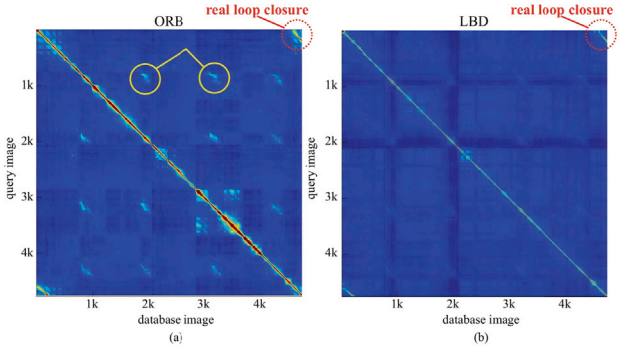


Fig. 22. Similarity matrices for a certain dataset where the (a) ORB keypoint-only bag-of-words approach yields false positives that are not present in the (b) LBD line-only approach (Gomez-Ojeda et al., 2019).

detection principle, which includes two similarity matrices (see Fig. 22).

Unlike the approach in Gomez-Ojeda et al. (2019), which reconstructs the BoW dictionary using LBD descriptors to address place recognition issues, Shu et al. (2023) employs ORB features (Mur-Artal & Tardós, 2014) to construct a DBoW dictionary for initial loop closure detection, primarily relying on point features. When a loop candidate keyframe is detected, the system computes the similarity transformation of 3D line features based on Plücker coordinates (Bartoli & Sturm, 2001), determining the transformation parameters through a scale factor, rotation matrix, and translation matrix. During the loop correction process, the system adjusts the line feature map represented by Plücker coordinates, optimizes the similarity transformation in the fundamental graph, distributes loop closure errors, and corrects scale drift (Strasdat, Montiel, & Davison, 2010), while simultaneously transforming the point and line features in the map. This process ultimately enhances the accuracy and robustness of loop closure detection through global optimization and endpoint refinement. Furthermore, (Qian, Zhao, Li, Ma, & Yu, 2019) introduced an innovative Bag-of-Point-Line-Words (BoPLW) pair method for detecting potential loop closures, with the algorithm framework illustrated in Algorithm 1. Unlike the method in Pumarola et al. (2017), which independently evaluates the similarity between existing keypoints and line BoW, (Qian et al., 2019) incorporates a constraint model that combines keypoints and line features, taking into account their co-occurrence and spatial proximity in loop closure detection. This approach effectively reduces false positives and addresses perceptual aliasing issues commonly encountered in traditional DBoW methods (Yu et al., 2018). Point features are represented using 32-bit binary ORB descriptors, while line features are represented by their endpoints using LSD descriptors. When the point feature circle intersects with the line segment, this method forms word pairs and employs a K-D tree (Bentley, 1975) for efficient searching. This combined feature approach enhances the robustness and accuracy of loop closure detection, particularly in environments with similar structures, such as industrial substations, by reducing false detections and improving scene recognition.

Algorithm 1: Loop-Closure Detection Using BoPLW Pairs

Input: Current keyframe f_i and the keyframe set $F_c = \{f_j\}$, where $f = f_i$, each f_j is associated with a kd-tree T_j;
Output: A detected revisit frame f_{matched};

1. tf-idf based retrieval of images where visual word of point or line occurs as candidate keyframes F_c;
$m_1 = \max\left(\text{NumOfCommonViewWords}(f_i, f_j)\right), \forall f_j \in F_c$;
$m_2 = \max\left(\text{NumOfCommonViewWordsOfPLPairs}(f_i, f_j)\right), \forall f_j \in F_c$;
for *each* $f_j \in F_c$ **do**
  if $(\textit{NumOfCommonViewWords}(f_i) < 0.8 \times m_1)$ && $(\textit{NumOfCommonViewWordsOfPLPairs}(f_i) < 0.8 \times m_2)$ **then**
    Update $F_c \leftarrow f_j \cup F_c$;
    Compute similarity, $score_i = \frac{N_{pS_p} + N_{lS_l}}{N_p + N_l}$;
    $AccScore_i = \sum_{i \in \text{neighbor}(f_i)} score_i$;
$AccScore_{\max} = \max_{f_j \in F_c}(AccScore_j)$;
for *each* $f_j \in F_c$ **do**
  if $score_j < 0.8 \times AccScore_{\max}$ **then**
    Delete those f_j with $score_j < 0.8 \times AccScore_{\max}$;
Perform space consistency verification on the selected candidate keyframes to obtain the final f_{matched};

Rong et al. (2021) proposed a point-line integrated stereo visual SLAM system (PEL-SLAM) to address the issue of loop closure detection in low-textured scenes. The system employs the ORB and EDlines algorithms to detect point and line features, respectively, and filters high-confidence line features by calculating their entropy values. An improved PL-BoW model is constructed, combining point and line features, and utilizing the co-occurrence information and spatial proximity within the visual vocabulary for loop closure detection. This approach enhances the accuracy of keyframe matching and system robustness, significantly reducing mismatches, especially in low-texture and varying illumination environments. Additionally, this method can generate a complete point-line map in real time, thereby enhancing the overall performance of the SLAM system. To address the shortcomings of traditional point-feature BoW models that are prone to false positives in low-texture or varying lighting conditions, Teng et al. (2023) introduced a loop closure detection method combining point and line features. This method employs a time- and space-based weighting mechanism to optimize similarity scoring, effectively reducing the false positive rate and improving system robustness. The system uses DBoW2 to detect key points and line features, and applies global BA

to optimize the map, significantly improving loop closure detection accuracy, making it suitable for environments with low textures and frequent changes in lighting.

Due to the high computational cost associated with matching line features across the entire map, most systems primarily rely on point features for loop closure detection, with line features often serving only as a supplementary role. However, line features contribute to improving the overall map accuracy in local mapping, thereby indirectly supporting loop closure detection. To reduce computational costs, point-line SLAM systems typically utilize only point features during loop closure detection. For instance, in the RGB-D SLAM system proposed by Fu et al. (2019), as well as in ORB-LINE-SLAM (Alamanos & Tzafestas, 2023), point features are predominantly used. Nevertheless, the latter incorporates line features for executing similarity transformations and pose graph optimization, which significantly enhances the system's robustness and accuracy in environments with low texture and drastic lighting changes.

4.2. Deep learning-based methods

Traditional point-line SLAM loop closure detection primarily relies on the matching of local and global features, with typical methods including BoW-based detection. However, these methods exhibit certain limitations in the feature extraction and matching processes, such as poor performance in low-texture environments, sensitivity to lighting variations, and low computational efficiency. To address these challenges, researchers have begun to incorporate deep learning techniques into loop closure detection. Deep learning not only enables the automatic extraction of point and line features from images but also significantly enhances scene understanding and matching accuracy. Compared to traditional BoW methods, deep learning demonstrates notable advantages in feature representation, matching speed, and robustness. Nevertheless, the integration of line features into deep learning-based loop closure detection remains relatively underexplored, indicating that this area still holds substantial potential for future research and development.

In loop closure detection, existing research mainly integrates point and line features with deep learning techniques in two ways:

Indirect Fusion Although line features are not directly involved in deep learning-based loop closure detection, they can indirectly improve environmental representation accuracy through local map building or pose optimization. For example, PLD-SLAM (Zhang, 2022) introduces the Global Grayscale Similarity (GGS) algorithm, which replaces the traditional Bag-of-Words (BoW) model and optimizes loop closure detection in dynamic scenes using a dynamic grid strategy. In this process, line features reduce dynamic interference through local Bundle Adjustment (BA) and enhance the physical plausibility of loop closure candidates through geometric verification. Experimental results show that the GGS method outperforms BoW in real-time performance on the KITTI and NEU datasets. PLE-SLAM [46] extracts features using SuperPoint (DeTone et al., 2018) and LSD (Von Gioi et al., 2008), but only employs the BoW vectors (DBoW2) generated by SuperPoint and the SuperGlue (Sarlin, DeTone, Malisiewicz, & Rabinovich, 2020) graph neural network for loop closure detection. With the support of line features, local BA and pose graph optimization provide geometric constraints, thereby indirectly improving the accuracy of loop closure correction. Pose optimization that integrates line features not only enhances loop closure detection but also significantly improves trajectory accuracy.DPL-SLAM (Lin, Zhang, Tian, Yu, & Lan, 2024) performs loop closure detection based on the fusion of deep learning-based point features and global descriptors: it extracts SuperPoint point features and NetVLAD global descriptors via HFnet and constructs a visual vocabulary using the FBow framework for fast keyframe matching and loop closure detection. While line features are not directly involved in loop closure detection, they enhance localization robustness in low-texture environments by jointly constraining the camera pose

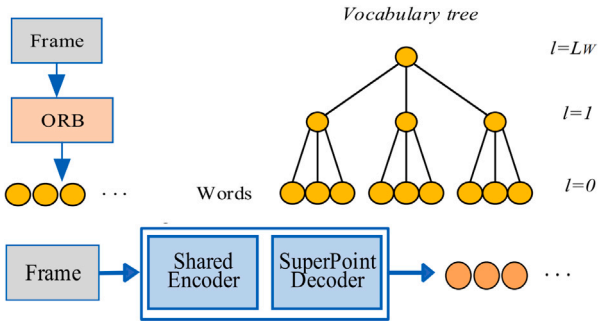


Fig. 23. Feature Clustering and BoW Tree Generation Method Based on ORB and SuperPoint (Lin, Zhang et al., 2024).

with point features. A comparison of the visual word bag construction process between traditional and deep learning methods is shown in Fig. 23, which involves generating the BoW tree using the K-means algorithm. The process begins with feature point extraction via ORB, followed by clustering. Then, the SuperPoint network is used to extract local features and generate matching BoW trees.AirSLAM (Xu et al., 2025) achieves illumination-robust loop closure detection through a three-stage process: first, a coarse retrieval is performed using the keypoint word bag (DBoW2); then, candidate frames are filtered through a structure graph constructed from line feature endpoints (junctions) and a scene-dependent junction vocabulary; finally, pose verification is carried out using LightGlue matching and PnP solving. Line features do not directly participate in the initial matching, but instead enhance the filtering process through the geometric information of the endpoints' structure.

Direct Fusion A few studies have attempted to integrate line features into deep learning frameworks, but their practical application is limited due to high computational complexity. DynPL-SLAM (Zhang et al., 2024) introduces the Dynamic Local Line Group (LLG) method to dynamically detect line features. Line features directly contribute to the system's loop closure detection by improving scene recognition accuracy and reducing drift, thus enhancing robustness. The system uses a Histogram of Region Similarity (HRS) model to quickly compute scene similarity, enabling loop closure detection through keyframe comparison. The use of line features improves robustness against dynamic objects by eliminating mismatched features, ensuring that only static features are involved in loop closure detection.DNN-PL-SLAM (Li et al., 2020) builds upon the ORB-SLAM framework and uses a Deep Neural Network (DNN) to extract hidden layer representations of point and line features, constructing a scene difference matrix. Line features directly participate in the generation of deep learning descriptors. Feature training is conducted using a stacked autoencoder to extract image patch features, which are optimized for robustness through denoising, sparsity, and continuity constraints. Line features serve as a supplement to point features when initialization fails, but the core of loop closure detection relies on the scene difference matrix constructed from DNN-extracted hidden layer features. Through normalization and singular value decomposition, scene ambiguity is reduced, and loop closure is ultimately determined based on a matrix similarity threshold.

Loop closure detection methods based on deep learning significantly overcome the limitations of traditional bag-of-words models in low-texture, dynamic lighting, and complex geometric environments through automated feature extraction and global scene understanding. Existing studies have further enhanced system robustness by indirectly or directly fusing point-line features. Experiments show that deep learning methods outperform traditional approaches in terms of accuracy and recall rate, especially in dynamic or low-texture scenes (as shown in Fig. 24). Table 8 compares the translational error (RMSE) of different SLAM systems on the EUROC dataset with and without loop closure, based on different features (points, lines) across various sequences. The

Table 8
Comparison of translational errors (RMSE) among SLAM systems on the EUROC dataset.

System	Feature		Loop		Architecture		Sequence											
	P	L	With	Without	Trad.	DL	MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V023	Avg
Kimera-VIO (Rosinol, Abate, Chang, & Carlone, 2020)	✓			✓	✓		0.110	0.100	0.160	0.240	0.350	0.050	0.080	0.070	0.079	0.100	0.190	0.139
Struct-VIO (Zou et al., 2019)	✓	✓		✓	✓		0.119	0.100	0.283	0.275	0.256	0.075	0.189	0.161	0.081	0.152	0.177	0.170
POL-VIO (Si, Yu, Chen, & Yang, 2024)	✓	✓		✓	✓		0.056	0.061	0.145	0.061	0.089	0.130	0.141	0.100	0.117	0.025	0.179	0.104
ORB-SLAM3(Campos et al., 2021)	✓		✓		✓		0.036	0.033	0.035	0.051	0.082	0.038	0.014	0.024	0.032	0.014	0.024	0.035
OpenVINS (Geneva, Eckenhoff, Lee, Yang, & Huang, 2020)	✓		✓		✓		0.072	0.143	0.086	0.173	0.247	0.055	0.060	0.059	0.054	0.047	0.141	0.103
PL-SLAM (Gomez-Ojeda et al., 2019)	✓	✓	✓		✓		0.042	0.052	0.040	0.064	0.074	0.042	0.046	0.069	0.061	0.057	0.126	0.061
EPL-VINS (Zeng et al., 2024)	✓	✓	✓		✓		0.033	0.020	0.052	0.159	0.109	0.040	0.067	0.051	0.034	0.094	0.016	0.061
Dynam-SLAM (Yin, Li, Tao, Guo, & Huang, 2022)	✓		✓			✓	0.078	0.067	0.061	0.077	0.096	0.052	0.040	0.051	0.048	0.056	0.085	0.065
DVI-SLAM (Peng et al., 2024)	✓		✓			✓	0.042	0.046	0.081	0.072	0.069	0.059	0.034	0.028	0.040	0.039	0.055	0.051
Vins-Fusion (Qin et al., 2018)	✓			✓	✓		0.163	0.178	0.316	0.331	0.176	0.101	0.099	0.112	0.109	0.123	0.250	0.175
Vins-Fusion (Qin et al., 2018)	✓		✓		✓		0.052	0.040	0.052	0.124	0.088	0.046	0.053	0.108	0.040	0.081	0.098	0.071
PLF-VINS (Lee & Park, 2021)	✓	✓		✓	✓		0.143	0.178	0.221	0.240	0.259	0.069	0.099	0.166	0.083	0.125	0.183	0.161
PLF-VINS (Lee & Park, 2021)	✓	✓	✓		✓		0.056	0.052	0.071	0.090	0.101	0.045	0.047	0.119	0.060	0.078	0.178	0.082
EPLF-VINS (Xu, Yin et al., 2022)	✓	✓		✓	✓		0.145	0.092	0.113	0.179	0.162	0.064	0.056	0.140	0.076	0.102	0.142	0.116
EPLF-VINS (Xu, Yin et al., 2022)	✓	✓	✓		✓		0.043	0.072	0.047	0.125	0.085	0.046	0.034	0.123	0.096	0.058	0.111	0.092
AirVO (Xu et al., 2023)	✓	✓		✓		✓	0.074	0.060	0.114	0.167	0.125	0.033	0.132	0.238	0.036	0.083	0.168	0.112
AirSLAM (Xu et al., 2025)	✓	✓	✓			✓	0.019	0.013	0.025	0.056	0.051	0.032	0.014	0.025	0.014	0.018	0.068	0.030
PLE-SLAM (He et al., 2025)	✓	✓	✓			✓	0.029	0.016	0.027	0.043	0.047	0.039	0.014	0.022	0.015	0.013	0.019	0.026

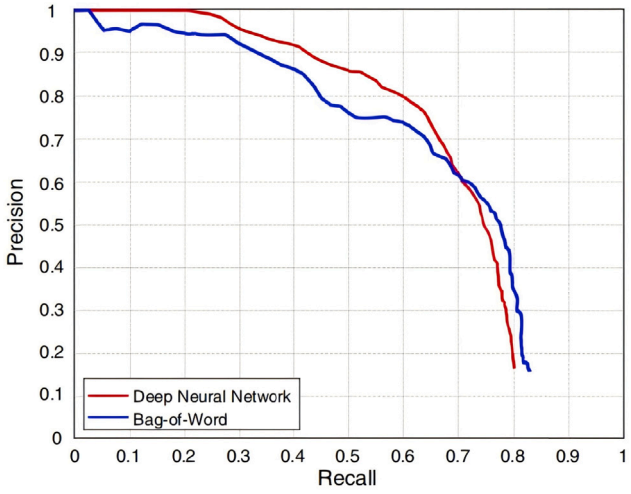


Fig. 24. Precision–recall curves of DNN and BoW (Li et al., 2020).

results indicate that incorporating line features into SLAM systems is particularly beneficial for loop closure detection in a deep learning environment. Future research needs to further explore lightweight line feature embedding strategies to balance computational efficiency and representation capability, thereby promoting the widespread application of point-line fusion deep learning loop closure detection in complex real-world environments.

5. SLAM for expert and intelligent systems

5.1. Synergy between SLAM and expert systems

There have been several innovative applications of integrating SLAM with expert systems. For example, the Expert Sample Consensus (ESAC) method combines the Mixture of Experts (MoE) and Differentiable Sample Consensus (DSAC) techniques (Brachmann & Rother, 2019), effectively merging multiple expert networks with SLAM to address ambiguities caused by repetitive structures in large-scale environments, thereby improving the robustness and accuracy of camera localization. ESAC dynamically assigns tasks through a gating network, ensuring that each expert network focuses on a specific subtask, similar to the task allocation and resource management mechanisms found in expert systems. This greatly enhances the adaptability and precision of SLAM. Additionally, the Wearable-Assisted System (WAS) (Qian et al., 2019) applies point-line fusion SLAM technology alongside Augmented Reality (AR) systems in substation inspections, providing operators with workflow guidance and task navigation through location awareness. Although it does not explicitly mention expert systems, its task-specific decision support and navigation functions resemble the intelligent decision-making mechanisms of expert systems, showcasing the potential applications of SLAM in complex environments. These examples demonstrate how SLAM can be integrated with intelligent decision systems and expert systems to enhance performance in complex dynamic environments, particularly in innovative applications involving task allocation, decision support, and dynamic adaptive control.

5.2. Innovation of integrating point-line vSLAM into expert systems

Traditional vSLAM methods primarily focus on geometric mapping and pose estimation. Integrating point-line vSLAM into expert systems brings several unique advantages and contributions:

Rich Structured Representation: By fusing point and line features, vSLAM is able to capture crucial geometric information in structured or low-texture environments. This enhanced robustness provides expert systems with reliable spatial awareness, reducing uncertainty in rule-based or model-based reasoning processes.

Real-Time Knowledge-Driven Decision Making: Expert systems can continuously update their decision-making processes by integrating the latest spatial information provided by vSLAM. This real-time integration allows expert systems to respond adaptively and intelligently to dynamic events, overcoming the limitations of traditional SLAM and expert systems individually.

Enhanced Collaborative Intelligence: In multi-agent systems, local vSLAM maps from each robot can be integrated into a unified expert system knowledge base, optimizing global planning, task allocation, and resource sharing. This high-level coordination framework significantly extends the capabilities of traditional single-agent SLAM systems and expert systems based on symbolic reasoning.

Connecting Perception with Domain Knowledge: Traditional SLAM lacks domain-specific knowledge, while expert systems often assume static or incomplete environmental information. The combination of point-line vSLAM with expert systems bridges this gap, merging robust scene perception with specialized rule-based reasoning. This makes the system more effective in tasks that require both geometric perception and semantic understanding, especially in domains such as industrial automation, urban infrastructure management, and medical robotics, relevant to ESWA.

Point-line vSLAM provides critical support for decision-making in expert systems, enhancing their adaptability, robustness, and intelligent reasoning capabilities. By offering high-precision localization and environmental mapping, particularly in structured, low-texture, or dynamic environments, point-line vSLAM improves expert systems' understanding and responsiveness to complex scenarios.

6. Discussions

This paper presents a comprehensive review of point-line feature integrated vSLAM systems, highlighting recent research progress and challenges in this field. By comparing the performance of traditional point-feature-based SLAM with point-line integrated SLAM, it is evident that the introduction of line features significantly enhances system performance in low-texture environments and improves the perception of environmental geometric structures. However, despite the clear advantages of point-line integrated SLAM in terms of accuracy and robustness, it still faces the following challenges:

1. **Complexity of Feature Extraction and Description:** Although line features provide rich geometric information, their extraction and description are more complex compared to point features, especially in dynamic environments or under varying lighting conditions. Line features are more susceptible to interference, leading to instability or mismatches. Therefore, designing efficient and robust line feature extraction and description algorithms remains a key focus for future research.
2. **Computational Efficiency and Real-Time Performance:** The introduction of line features increases the computational load of SLAM systems, particularly in the back-end optimization stage, where the integration of point-line features leads to a more complex and time-consuming optimization process. Although some studies have improved real-time performance through techniques such as sliding windows and sparse optimization, achieving efficient operation in resource-constrained systems remains challenging. Balancing computational complexity while ensuring performance will be a crucial direction for optimizing point-line SLAM systems.
3. **Integration of Deep Learning with Traditional Methods:** Deep learning has demonstrated strong capabilities in point and line feature extraction, matching, and parameterization, particularly with self-supervised learning and end-to-end frameworks providing new avenues for handling complex environments. However, existing deep learning approaches often rely heavily on large-scale training datasets and demand substantial computational resources. Achieving efficient and lightweight deep learning models on resource-constrained devices is a promising direction for future exploration. Furthermore, the integration of deep learning with traditional geometric methods holds the potential to further enhance the performance of point-line SLAM systems.

In conclusion, the integration of point and line features presents new opportunities for the application of SLAM systems in complex environments; however, challenges remain in areas such as feature extraction, matching, and optimization. Future research can focus on improving computational efficiency, integrating deep learning with geometric methods, and further advancing the practical application and development of point-line SLAM technology.

7. Conclusions and future research directions

7.1. Conclusions

This paper presents a comprehensive review of vSLAM systems integrating point and line features, encompassing the latest developments and technological advancements in the field. Firstly, the study provides an in-depth introduction to the core components of the visual front-end in SLAM systems, with a particular emphasis on delivering a comprehensive analysis of line feature detection, description, and matching methods. It thoroughly examines the application and performance of both traditional algorithms and deep learning-based improvement methods in feature parameterization and matching strategies. Subsequently, the paper classifies and compares various vSLAM systems that integrate point and line features, exploring the strengths and weaknesses of different algorithms across diverse environments, as well as the potential for integrating deep learning with traditional methods. Regarding back-end optimization, the study provides a detailed investigation into how optimization strategies, including filtering optimization and graph optimization techniques, affect system performance. Furthermore, it reviews the critical role of loop closure detection mechanisms in enhancing the accuracy and robustness of vSLAM systems. Finally, the paper provides a synthesis of the challenges encountered by point-line integrated vSLAM systems and articulates future research directions, offering theoretical guidance and reference points for subsequent studies.

7.2. Future research directions

With the rapid development of vSLAM technology, the integration of point and line features has made significant progress in applications within complex environments. However, despite these advancements, several challenges remain, including issues related to computational efficiency, system robustness, adaptability, and the effective fusion of multi-modal data. Therefore, further improving the performance of these systems remains crucial for future research. Based on the limitations of current technologies and recent advancements, we believe that the following research directions will be key areas for future exploration:

7.2.1. Lightweight models

In point-line feature-based vSLAM systems, the introduction of line features enhances the system's robustness in low-texture environments and improves its geometric perception capabilities. However, the extraction, description, and matching of line features are more complex and introduce additional computational overhead. Therefore, designing lightweight deep learning models is a key research direction for optimizing point-line feature fusion systems.

Challenges of Computational Complexity and Model Lightweighting: The extraction and description of line features involve handling continuous geometric information (e.g., direction, length, and endpoint positions), which generally leads to higher computational complexity than point features. Moreover, incorporating line features into deep learning models further increases both computational and storage demands. Therefore, achieving model lightweighting through methods such as pruning, quantization, and distillation, while retaining the rich geometric information provided by line features, will be an important direction for future research.

Lightweight Strategies Combined with Hardware Optimization: On the hardware level, the fusion of point and line features can be deeply optimized by leveraging specialized hardware such as FPGAs and TPUs. For instance, designing accelerators specifically optimized for line feature detection modules can significantly improve efficiency. Additionally, the use of edge computing and low-power chips will further enhance the real-time performance of the system in resource-constrained scenarios.

Generalization Ability of Lightweight Models: Point-line feature fusion models need to exhibit strong generalization ability to operate stably in diverse environments, such as low-texture, dynamic lighting, or occlusion scenarios. Future research may focus on optimizing models through techniques such as multi-task learning, sparse representation, and low-rank decomposition, enabling the model to perform well across different environments. Additionally, integrating geometric constraints can further enhance the robustness of the model.

Research on lightweight models not only aims to reduce the computational overhead of point feature processing but also addresses the complexity and resource demands introduced by line features. In the future, by optimizing the joint learning framework for point-line features, improving lightweight strategies, and incorporating efficient hardware platforms, vSLAM systems will likely find broader applications in low-power, high-efficiency environments.

7.2.2. End-to-end framework

In future vSLAM research, developing an end-to-end framework that simultaneously performs point and line feature detection is a key direction. By tightly integrating feature detection, matching, and optimization, an end-to-end system can significantly improve matching accuracy and robustness, enhancing system performance in dynamic environments.

Integration and Optimization of Point-Line Features: In vSLAM, the joint learning framework of point and line features effectively captures the diversity and complexity of the scene, improving the system's robustness under extreme conditions. In the future, the end-to-end framework needs to further optimize the coordination among various stages, such as feature detection, descriptor computation, matching, and optimization, to eliminate inconsistencies between modules and enhance the overall system's precision.

Enhancing Computational Efficiency and System Robustness: End-to-end learning optimizes performance metrics directly, reducing the need for manual tuning and improving system efficiency and adaptability, especially in dynamic and unknown environments. However, it also faces challenges such as high computational resource consumption and insufficient diversity in training data. Therefore, improving computational efficiency while maintaining robustness will be crucial in the design of future end-to-end frameworks.

Practical Application Prospects: The end-to-end framework not only improves the performance of SLAM systems in static environments but also opens up opportunities for applications in dynamic environments, such as autonomous driving, robotic navigation, and augmented reality. By optimizing the collaboration of different modules, the end-to-end system enhances real-time performance and stability, enabling it to better handle complex environments.

Despite the significant potential of the end-to-end framework in vSLAM, challenges remain in terms of training data diversity, computational resource requirements, and model complexity. Efficiently addressing these issues will be an important focus of future research.

7.2.3. Multimodal data fusion

Traditional vSLAM systems rely on single-source visual data, which are susceptible to challenges such as lighting changes, dynamic occlusions, and textureless environments, making it difficult to address long-term error accumulation. To overcome these limitations, the integration of multi-source data has emerged as an effective solution. Research on multi-modal data fusion can be explored in the following directions:

Fusion Algorithms and Optimization Techniques: Efficient fusion algorithms are central to the integration of multi-modal data. Commonly used techniques include filtering methods and graph optimization approaches. In graph optimization, factor graphs are employed to unify data constraints from different sensors into an optimization problem. Additionally, deep learning-based fusion methods can learn shared features and fusion representations from sensor data, further enhancing system performance.

Integration of Geometric and Semantic Information: Combining geometric and semantic information can enhance the system's understanding of the environment. For example, semantic segmentation results can improve the classification capabilities of point-line features, and by integrating line, surface features, and semantic information, semantic-labeled 3D maps can be generated, improving navigation accuracy.

Application-Specific Optimizations: Different application scenarios have distinct requirements for multi-modal fusion. For instance, in autonomous driving, the combination of GNSS and LiDAR provides macro-level localization support and enhances navigation capabilities, while in indoor localization, the fusion of depth cameras and inertial sensors improves positioning accuracy.

The synergy between lightweight models, end-to-end frameworks, and multi-modal data fusion forms a collaborative loop in point-line fusion vSLAM technology: Lightweight design reduces the deployment cost of end-to-end frameworks, which can achieve lightweight design through multi-task learning, while multi-modal data provides self-supervised signals for model training. Future research should focus on the collaborative innovation of algorithms, hardware, and applications to tackle core technological bottlenecks such as dynamic environment perception and resource-constrained deployment.

CRediT authorship contribution statement

Hangzhou Qu: Drafting, Revising the manuscript. **Zhuhua Hu:** Research framework, Developed the theoretical structure. **Yaochi Zhao:** Conducted experiments, Experimental verification. **Junlin Lu:** Developed tables & figures. **Kunkun Ding:** Collected & compiled the data. **Guangfeng Liu:** Collected & compiled the data. **Yongqing Chen:** Managed the layout, Formatting review. **Chunyan Shao:** Oversaw, Guided the research team.

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.

Acknowledgments

This research was supported by the Key Research and Development Project of Hainan Province, China (Grant no. ZDYF2022GXJS348 and Grant no. ZDYF2024GXJS021), the National Natural Science Foundation of China (Grant No. 62361024 and Grant no. 62161010), the Hainan Province Natural Science Foundation, China (623RC446), the 2023 Hainan Province Graduate Innovation Research Project, China (Grant No. SA2400003283),and the Hainan Provincial Agricultural Reclamation Group Co., Ltd. Science and Technology Innovation "Challenge and Leadership" Project: Key Technology R&D of Novel Lychee Planting Models and Automated Harvesting Robots, China. The authors would like to thank the referees for their constructive suggestions.

Data availability

Data will be made available on request.

References

Abdellali, Hichem, Frohlich, Robert, Vilagos, Viktor, & Kato, Zoltan (2021). L2d2: Learnable line detector and descriptor. In *2021 international conference on 3D vision (3DV)* (pp. 442–452). IEEE.

Agarwal, Sameer, Mierle, Keir, et al. (2012). Ceres solver: Tutorial & reference. *Google Inc*, *2*(72), 8.

Akinlar, Cuneyt, & Topal, Cihan (2011). EDLines: A real-time line segment detector with a false detection control. *Pattern Recognition Letters*, *32*(13), 1633–1642.

Alamanos, Ioannis, & Tzafestas, Costas (2023). ORB-LINE-SLAM: An open-source stereo visual SLAM system with point and line features. *Authorea Preprints*.

Almalioglu, Yasin, Saputra, Muhamad Risqi U, De Gusmao, Pedro PB, Markham, Andrew, & Trigoni, Niki (2019). Ganvo: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks. In *2019 international conference on robotics and automation* (pp. 5474–5480). IEEE.

Bailey, Tim, Nieto, Juan, Guivant, Jose, Stevens, Michael, & Nebot, Eduardo (2006). Consistency of the EKF-SLAM algorithm. In *2006 IEEE/RSJ international conference on intelligent robots and systems* (pp. 3562–3568). IEEE.

Baker, Simon, & Matthews, Iain (2004). Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, *56*, 221–255.

Ballard, Dana H. (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, *13*(2), 111–122.

Barrau, Axel, & Bonnabel, Silvere (2015). An EKF-SLAM algorithm with consistency properties. ArXiv preprint arXiv:1510.06263.

Bartoli, Adrien, & Sturm, Peter (2001). The 3D line motion matrix and alignment of line reconstructions. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition (Vol. 1)* (p. I). IEEE.

Bartoli, Adrien, & Sturm, Peter (2005). Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. *Computer Vision and Image Understanding*, *100*(3), 416–441.

Bentley, Jon Louis (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, *18*(9), 509–517.

Bloesch, Michael, Omari, Sammy, Hutter, Marco, & Siegwart, Roland (2015). Robust visual inertial odometry using a direct EKF-based approach. In *2015 IEEE/RSJ international conference on intelligent robots and systems* (pp. 298–304). IEEE.

Bonnabel, Silvere (2007). Left-invariant extended Kalman filter and attitude estimation. In *2007 46th IEEE conference on decision and control* (pp. 1027–1032). IEEE.

Bouguet, Jean-Yves, et al. (2001). Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, *5*(1–10), 4.

Brachmann, Eric, & Rother, Carsten (2019). Expert sample consensus applied to camera re-localization. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 7525–7534).

Brossard, Martin, Bonnabel, Silvere, & Barrau, Axel (2018a). Invariant Kalman filtering for visual inertial SLAM. In *2018 21st international conference on information fusion* (pp. 2021–2028). IEEE.

Brossard, Martin, Bonnabel, Silvere, & Barrau, Axel (2018b). Unscented Kalman filter on Lie groups for visual inertial odometry. In *2018 IEEE/RSJ international conference on intelligent robots and systems* (pp. 649–655). IEEE.

Bruno, Hudson Martins Silva, & Colombini, Esther Luna (2021). LIFT-SLAM: A deep-learning feature-based monocular visual SLAM method. *Neurocomputing*, *455*, 97–110.

Cadena, Cesar, Carlone, Luca, Carrillo, Henry, Latif, Yasir, Scaramuzza, Davide, Neira, José, et al. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, *32*(6), 1309–1332.

Cai, Dupeng, Li, Ruoqing, Hu, Zhuhua, Lu, Junlin, Li, Shijiang, & Zhao, Yaochi (2024). A comprehensive overview of core modules in visual SLAM framework. *Neurocomputing*, Article 127760.

Calonder, Michael, Lepetit, Vincent, Strecha, Christoph, & Fua, Pascal (2010). 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* (pp. 778–792). Springer.

Campos, Carlos, Elvira, Richard, Rodríguez, Juan J Gómez, Montiel, José MM, & Tardós, Juan D (2021). Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, *37*(6), 1874–1890.

Canny, John (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6), 679–698.

Caruso, David, Engel, Jakob, & Cremers, Daniel (2015). Large-scale direct SLAM for omnidirectional cameras. In *2015 IEEE/RSJ international conference on intelligent robots and systems* (pp. 141–148). IEEE.

Castellanos, Jose A., & Tardos, Juan D. (2012). *Mobile robot localization and map building: A multisensor fusion approach*. Springer Science & Business Media.

Chen, Zhenhang, Miao, Zhiqiang, Liu, Min, Wu, Chengzhong, & Wang, Yaonan (2024). A fast and accurate visual inertial odometry using hybrid point-line features. *IEEE Robotics and Automation Letters*.

Chen, Mingxiang, & Pan, Cihui (2023). Parsing line segments of floor plan images using graph neural networks. ArXiv preprint arXiv:2303.03851.

Cho, Nam-Gyu, Yuille, Alan, & Lee, Seong-Whan (2017). A novel linelet-based representation for line segment detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(5), 1195–1208.

Cioffi, Giovanni, & Scaramuzza, Davide (2020). Tightly-coupled fusion of global positional measurements in optimization-based visual-inertial odometry. In *2020 IEEE/RSJ international conference on intelligent robots and systems* (pp. 5089–5095). IEEE.

Clemens, Joachim, Kluth, Tobias, & Reineking, Thomas (2019). β-SLAM: Simultaneous localization and grid mapping with beta distributions. *Information Fusion*, *52*, 62–75.

Csurka, Gabriella, Dance, Christopher, Fan, Lixin, Willamowski, Jutta, & Bray, Cédric (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV (Vol. 1, No. 1–22)* (pp. 1–2). Prague.

Dai, Xili, Gong, Haigang, Wu, Shuai, Yuan, Xiaojun, & Ma, Yi (2022). Fully convolutional line parsing. *Neurocomputing*, *506*, 1–11.

Davison, Andrew J, Reid, Ian D, Molton, Nicholas D, & Stasse, Olivier (2007). MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(6), 1052–1067.

Denis, Patrick, Elder, James H., & Estrada, Francisco J. (2008). Efficient edge-based methods for estimating manhattan frames in urban imagery. In *Computer vision–ECCV 2008: 10th European conference on computer vision, Marseille, France, October 12–18, 2008, proceedings, Part II 10* (pp. 197–210). Springer.

Desolneux, Agnes, Moisan, Lionel, & Morel, Jean-Michel (2007). *From gestalt theory to image analysis: a probabilistic approach (Vol. 34)*. Springer Science & Business Media.

DeTone, Daniel, Malisiewicz, Tomasz, & Rabinovich, Andrew (2018). Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 224–236).

Di Stefano, Luigi, Mattoccia, Stefano, & Tombari, Federico (2005). ZNCC-based template matching using bounded partial correlation. *Pattern Recognition Letters*, *26*(14), 2129–2134.

Dosovitskiy, Alexey, Ros, German, Codevilla, Felipe, Lopez, Antonio, & Koltun, Vladlen (2017). CARLA: An open urban driving simulator. In *Conference on robot learning* (pp. 1–16). PMLR.

Duda, Richard O., & Hart, Peter E. (1972). Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, *15*(1), 11–15.

Engel, Jakob, Koltun, Vladlen, & Cremers, Daniel (2017). Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(3), 611–625.

Engel, Jakob, Schöps, Thomas, & Cremers, Daniel (2014). LSD-SLAM: Large-scale direct monocular SLAM. In *European conference on computer vision* (pp. 834–849). Springer.

Engel, Jakob, Stückler, Jörg, & Cremers, Daniel (2015). Large-scale direct SLAM with stereo cameras. In *2015 IEEE/RSJ international conference on intelligent robots and systems* (pp. 1935–1942). IEEE.

Fernandes, Duarte, Silva, António, Névoa, Rafael, Simões, Cláudia, Gonzalez, Dibet, Guevara, Miguel, et al. (2021). Point-cloud based 3D object detection and classification methods for self-driving applications: A survey and taxonomy. *Information Fusion*, *68*, 161–191.

Fischler, Martin A., & Bolles, Robert C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*(6), 381–395.

Forster, Christian, Pizzoli, Matia, & Scaramuzza, Davide (2014). SVO: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation* (pp. 15–22). IEEE.

Forster, Christian, Zhang, Zichao, Gassner, Michael, Werlberger, Manuel, & Scaramuzza, Davide (2016). SVO: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, *33*(2), 249–265.

Freda, Luigi (2023). PLVS: A SLAM system with points, lines, volumetric mapping, and 3D incremental segmentation. ArXiv preprint arXiv:2309.10896.

Fu, Qiang, Wang, Jialong, Yu, Hongshan, Ali, Islam, Guo, Feng, He, Yijia, et al. (2020). PL-VINS: Real-time monocular visual-inertial SLAM with point and line features. ArXiv preprint arXiv:2009.07462.

Fu, Qiang, Yu, Hongshan, Lai, Lihai, Wang, Jingwen, Peng, Xia, Sun, Wei, et al. (2019). A robust RGB-D SLAM system with points and lines for low texture indoor environments. *IEEE Sensors Journal*, *19*(21), 9908–9920.

Furukawa, Yasutaka, & Shinagawa, Yoshihisa (2003). Accurate and robust line segment extraction by analyzing distribution around peaks in Hough space. *Computer Vision and Image Understanding*, *92*(1), 1–25.

Gálvez-López, Dorian, & Tardos, Juan D. (2012). Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, *28*(5), 1188–1197.

Gao, Shuang, Wan, Jixiang, Ping, Yishan, Zhang, Xudong, Dong, Shuzhou, Yang, Yuchen, et al. (2022). Pose refinement with joint optimization of visual points and lines. In *2022 IEEE/RSJ international conference on intelligent robots and systems* (pp. 2888–2894). IEEE.

Gao, Xiang, Wang, Rui, Demmel, Nikolaus, & Cremers, Daniel (2018). LDSO: Direct sparse odometry with loop closure. In *2018 IEEE/RSJ international conference on intelligent robots and systems* (pp. 2198–2204). IEEE.

Gee, Andrew P., & Mayol-Cuevas, Walterio (2006). Real-time model-based SLAM using line segments. In *International symposium on visual computing* (pp. 354–363). Springer.

Geneva, Patrick, Eckenhoff, Kevin, Lee, Woosik, Yang, Yulin, & Huang, Guoquan (2020). Openvins: A research platform for visual-inertial estimation. In *2020 IEEE international conference on robotics and automation* (pp. 4666–4672). IEEE.

Gomez-Ojeda, Ruben, Briales, Jesus, & Gonzalez-Jimenez, Javier (2016). PL-SVO: Semi-direct monocular visual odometry by combining points and line segments. In *2016 IEEE/RSJ international conference on intelligent robots and systems* (pp. 4211–4216). IEEE.

Gomez-Ojeda, Ruben, & Gonzalez-Jimenez, Javier (2016). Robust stereo visual odometry through a probabilistic combination of points and line segments. In *2016 IEEE international conference on robotics and automation* (pp. 2521–2526). IEEE.

Gomez-Ojeda, Ruben, Moreno, Francisco-Angel, Zuniga-Noël, David, Scaramuzza, Davide, & Gonzalez-Jimenez, Javier (2019). PL-SLAM: A stereo SLAM system through the combination of points and line segments. *IEEE Transactions on Robotics*, *35*(3), 734–746.

Gu, Geonmo, Ko, Byungsoo, Go, SeoungHyun, Lee, Sung-Hyun, Lee, Jingeun, & Shin, Minchul (2022). Towards light-weight and real-time line segment detection. In *Proceedings of the AAAI conference on artificial intelligence (Vol. 36, No. 1)* (pp. 726–734).

Guan, Robin Ping, Ristic, Branko, Wang, Liuping, & Palmer, Jennifer L (2019). KLD sampling with gmapping proposal for Monte Carlo localization of mobile robots. *Information Fusion*, *49*, 79–88.

Hartley, Richard I. (1995). A linear method for reconstruction from lines and points. In *Proceedings of IEEE international conference on computer vision* (pp. 882–887). IEEE.

Hartley, Richard, & Zisserman, Andrew (2003). *Multiple view geometry in computer vision*. Cambridge University Press.

He, Jiaming, Li, Mingrui, Wang, Yangyang, & Wang, Hongyu (2024). PLE-SLAM: A visual-inertial SLAM based on point-line features and efficient IMU initialization. ArXiv preprint arXiv:2401.01081.

He, Jiaming, Li, Mingrui, Wang, Yangyang, & Wang, Hongyu (2025). PLE-SLAM: A visual-inertial SLAM based on point-line features and efficient IMU initialization. *IEEE Sensors Journal*.

He, Yijia, Zhao, Ji, Guo, Yue, He, Wenhao, & Yuan, Kui (2018). PL-VIO: Tightly-coupled monocular visual–inertial odometry using point and line features. *Sensors*, *18*(4), 1159.

Henein, Mina, Zhang, Jun, Mahony, Robert, & Ila, Viorela (2020). Dynamic SLAM: The need for speed. In *2020 IEEE international conference on robotics and automation* (pp. 2123–2129). IEEE.

Heo, Sejong, Jung, Jae Hyung, & Park, Chan Gook (2018). Consistent EKF-based visual-inertial navigation using points and lines. *IEEE Sensors Journal*, *18*(18), 7638–7649.

Hough, Paul V. C. (1962). *Method and means for recognizing complex patterns*. Google Patents, US Patent 3, 069, 654.

Hu, Zhuhua, Qi, Wenlu, Ding, Kunkun, Liu, Guangfeng, & Zhao, Yaochi (2024). An adaptive lighting indoor vSLAM with limited on-device resources. *IEEE Internet of Things Journal*.

Hua, Tong, Li, Tao, Pang, Liang, Liu, Guoqing, Xuanyuan, Wencheng, Shu, Chang, et al. (2023). PLV-IEKF: Consistent visual-inertial odometry using points, lines, and vanishing points. In *2023 IEEE international conference on robotics and biomimetics* (pp. 1–7). IEEE.

Hua, Tong, Li, Tao, & Pei, Ling (2023). PIEKF-VIWO: Visual-inertial-wheel odometry using partial invariant extended Kalman filter. In *2023 IEEE international conference on robotics and automation* (pp. 2083–2090). IEEE.

Huang, Shoudong, & Dissanayake, Gamini (2007). Convergence and consistency analysis for extended Kalman filter based SLAM. *IEEE Transactions on Robotics*, *23*(5), 1036–1049.

Huang, Guoquan P, Mourikis, Anastasios I, & Roumeliotis, Stergios I (2008). Analysis and improvement of the consistency of extended Kalman filter based SLAM. In *2008 IEEE international conference on robotics and automation* (pp. 473–479). IEEE.

Huang, Guoquan P, Mourikis, Anastasios I, & Roumeliotis, Stergios I (2010). Observability-based rules for designing consistent EKF SLAM estimators. *The International Journal of Robotics Research*, *29*(5), 502–528.

Huang, Siyu, Qin, Fangbo, Xiong, Pengfei, Ding, Ning, He, Yijia, & Liu, Xiao (2020). TP-LSD: Tri-points based line segment detector. In *European conference on computer vision* (pp. 770–785). Springer.

Huang, Kun, Wang, Yifan, Zhou, Zihan, Ding, Tianjiao, Gao, Shenghua, & Ma, Yi (2018). Learning to parse wireframes in images of man-made environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 626–635).

Jeon, Jinwoo, Lim, Hyunjun, Seo, Dong-Uk, & Myung, Hyun (2022). Struct-MDC: Mesh-refined unsupervised depth completion leveraging structural regularities from visual SLAM. *IEEE Robotics and Automation Letters*, *7*(3), 6391–6398.

Joswig, Michael, & Theobald, Thorsten (2013). Plücker coordinates and lines in space. *Polyhedral and Algebraic Methods in Computational Geometry*, 193–207.

Jung, KwangYik, Kim, YeEun, Lim, HyunJun, & Myung, Hyun (2021). ALVIO: Adaptive line and point feature-based visual inertial odometry for robust localization in indoor environments. In *RiTA 2020: proceedings of the 8th international conference on robot intelligence technology and applications* (pp. 171–184). Springer.

Kannapiran, Shenbagaraj, Bendapudi, Nalin, Yu, Ming-Yuan, Parikh, Devarth, Berman, Spring, Vora, Ankit, et al. (2023). Stereo visual odometry with deep learning-based point and line feature matching using an attention graph neural network. In *2023 IEEE/RSJ international conference on intelligent robots and systems* (pp. 3491–3498). IEEE.

Kannapiran, Shenbagaraj, Van Baar, Jeroen, & Berman, Spring (2021). A visual inertial odometry framework for 3d points, lines and planes. In *2021 IEEE/RSJ international conference on intelligent robots and systems* (pp. 9206–9211). IEEE.

Kazerouni, Iman Abaspur, Fitzgerald, Luke, Dooly, Gerard, & Toal, Daniel (2022). A survey of state-of-the-art on visual SLAM. *Expert Systems with Applications*, *205*, Article 117734.

Klein, Georg, & Murray, David (2007). Parallel tracking and mapping for small AR workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality* (pp. 225–234). IEEE.

Klein, Georg, & Murray, David (2008). Improving the agility of keyframe-based SLAM. In *Computer vision–ECCV 2008: 10th European conference on computer vision, Marseille, France, October 12–18, 2008, proceedings, Part II 10* (pp. 802–815). Springer.

Klein, Georg, & Murray, David (2009). Parallel tracking and mapping on a camera phone. In *2009 8th IEEE international symposium on mixed and augmented reality* (pp. 83–86). IEEE.

Ko, Jinwon, Jin, Dongkwon, & Kim, Chang-Su (2024). Semantic line combination detector. ArXiv preprint arXiv:2404.18399.

Kümmerle, Rainer, Grisetti, Giorgio, Strasdat, Hauke, Konolige, Kurt, & Burgard, Wolfram (2011). G 2 o: A general framework for graph optimization. In *2011 IEEE international conference on robotics and automation* (pp. 3607–3613). IEEE.

Lange, Manuel, Schweinfurth, Fabian, & Schilling, Andreas (2019). Dld: A deep learning based line descriptor for line feature matching. In *2019 IEEE/RSJ international conference on intelligent robots and systems* (pp. 5910–5915). IEEE.

Lee, Seong Hun, & Civera, Javier (2018). Loosely-coupled semi-direct monocular slam. *IEEE Robotics and Automation Letters*, *4*(2), 399–406.

Lee, Sang Jun, & Hwang, Sung Soo (2019). Elaborate monocular point and line slam with robust initialization. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1121–1129).

Lee, Junesuk, & Park, Soon-Yong (2021). PLF-VINS: Real-time monocular visual-inertial SLAM with point-line fusion and parallel-line fusion. *IEEE Robotics and Automation Letters*, *6*(4), 7033–7040.

Leonard, John J., & Durrant-Whyte, Hugh F. (1991). Simultaneous map building and localization for an autonomous mobile robot. In *IROS (Vol. 3)* (pp. 1442–1447).

Lepetit, Vincent, Moreno-Noguer, Francesc, & Fua, Pascal (2009). EP n P: An accurate O (n) solution to the P n P problem. *International Journal of Computer Vision*, *81*, 155–166.

Leutenegger, Stefan, Lynen, Simon, Bosse, Michael, Siegwart, Roland, & Furgale, Paul (2015). Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, *34*(3), 314–334.

Li, Yanyan, Brasch, Nikolas, Wang, Yida, Navab, Nassir, & Tombari, Federico (2020). Structure-slam: Low-drift monocular slam in indoor environments. *IEEE Robotics and Automation Letters*, *5*(4), 6583–6590.

Li, Yang, Chen, Chao Ping, Maitlo, Nizamuddin, Mi, Lantian, Zhang, Wenbo, & Chen, Jie (2020). Deep neural network–based loop detection for visual simultaneous localization and mapping featuring both points and lines. *Advanced Intelligent Systems*, *2*(1), Article 1900107.

Li, Ziqi, Gao, Wei, Chen, Haoyao, & Zhang, Shiwu (2023). USP-SLAM: Deep learning based visual SLAM with robust feature extraction under dynamic environments. In *2023 IEEE international conference on robotics and biomimetics* (pp. 1–6). IEEE.

Li, Xin, He, Yijia, Lin, Jinlong, & Liu, Xiao (2020). Leveraging planar regularities for point line visual-inertial odometry. In *2020 IEEE/RSJ international conference on intelligent robots and systems* (pp. 5120–5127). IEEE.

Li, Mingyang, & Mourikis, Anastasios I. (2013). High-precision, consistent EKF-based visual-inertial odometry. *The International Journal of Robotics Research*, *32*(6), 690–711.

Li, Wanting, Shao, Yu, Wang, Yongcai, Wang, Shuo, Bai, Xuewei, & Li, Deying (2023). IDLL: Inverse depth line based visual localization in challenging environments. ArXiv preprint arXiv:2304.11748.

Li, Kai, Yao, Jian, Lu, Xiaohu, Li, Li, & Zhang, Zhichao (2016). Hierarchical line matching based on line–junction–line structure descriptor and local homography estimation. *Neurocomputing*, *184*, 207–220.

Li, Yanyan, Yunus, Raza, Brasch, Nikolas, Navab, Nassir, & Tombari, Federico (2021). RGB-D SLAM with structural regularities. In *2021 IEEE international conference on robotics and automation* (pp. 11581–11587). IEEE.

Lim, Hyunjun, Jeon, Jinwoo, & Myung, Hyun (2022). UV-SLAM: Unconstrained line-based SLAM using vanishing points for structural mapping. *IEEE Robotics and Automation Letters*, *7*(2), 1518–1525.

Lim, Hyunjun, Kim, Yeeun, Jung, Kwangik, Hu, Sumin, & Myung, Hyun (2021). Avoiding degeneracy for monocular visual SLAM with point and line features. In *2021 IEEE international conference on robotics and automation* (pp. 11675–11681). IEEE.

Lin, Xiao, & Wang, Chen (2023). AirLine: Efficient learnable line detection with local edge voting. In *2023 IEEE/RSJ international conference on intelligent robots and systems* (pp. 3270–3277). IEEE.

Lin, Shuyue, Zhang, Xuetao, Liu, Yisha, Wang, Hanzhang, Zhang, Xuebo, & Zhuang, Yan (2024). FLM PL-VIO: A robust monocular point-line visual-inertial odometry based on fast line matching. *IEEE Transactions on Industrial Electronics*.

Lin, Zhihao, Zhang, Qi, Tian, Zhen, Yu, Peizhuo, & Lan, Jianglin (2024). DPL-SLAM: enhancing dynamic point-line SLAM through dense semantic methods. *IEEE Sensors Journal*.

Lin, Xinyu, Zhou, Yingjie, Liu, Yipeng, & Zhu, Ce (2023a). Effective and efficient line segment detection for visual measurement guided by level lines. *IEEE Transactions on Instrumentation and Measurement*.

Lin, Xinyu, Zhou, Yingjie, Liu, Yipeng, & Zhu, Ce (2023b). Level-line guided edge drawing for robust line segment detection. In *ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing* (pp. 1–5). IEEE.

Lin, Xinyu, Zhou, Yingjie, Liu, Yipeng, & Zhu, Ce (2024). A comprehensive review of image line segment detection and description: Taxonomies, comparisons, and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liu, Hongmin, Cao, Chengyang, Ye, Hanqiao, Cui, Hainan, Gao, Wei, Wang, Xing, et al. (2024). Lightweight structured line map based visual localization. *IEEE Robotics and Automation Letters*.

Liu, Guangfeng, Hu, Zhuhua, Zhao, Yaochi, Li, Ruoqing, Ding, Kunkun, & Qi, Wenlu (2024). A key frame selection and local ba optimization method for vslam. *International Journal of Robotics & Automation*.

Liu, Xin, Wen, Shuhuan, & Zhang, Hong (2023). A real-time stereo visual-inertial SLAM system based on point-and-line features. *IEEE Transactions on Vehicular Technology*, *72*(5), 5747–5758.

Lu, Yan, & Song, Dezhen (2015). Robust RGB-d odometry using point and line features. In *Proceedings of the IEEE international conference on computer vision* (pp. 3934–3942).

Lucas, Bruce D., & Kanade, Takeo (1981). An iterative image registration technique with an application to stereo vision. In *IJCAI'81: 7th international joint conference on artificial intelligence (Vol. 2)* (pp. 674–679).

Ma, QuanMeng, Jiang, Guang, & Lai, DianZhi (2020). Robust line segments matching via graph convolution networks. ArXiv preprint arXiv:2004.04993.

Ma, Quanmeng, Jiang, Guang, Wu, Jiajie, Cai, Changshuai, Lai, Dianzhi, Bai, Zixuan, et al. (2021). WGLSM: An end-to-end line matching network based on graph convolution. *Neurocomputing*, *453*, 195–208.

Ma, Xin, & Liang, Xinwu (2021). Point-line-based RGB-D SLAM and bundle adjustment uncertainty analysis. ArXiv preprint arXiv:2102.07110.

Ma, Xiaolin, & Ning, Shiwen (2020). Real-time visual-inertial SLAM with point-line feature using improved edlines algorithm. In *2020 IEEE 5th information technology and mechatronics engineering conference* (pp. 1323–1327). IEEE.

Ma, Jiayi, Wang, Xinya, He, Yijia, Mei, Xiaoguang, & Zhao, Ji (2019). Line-based stereo SLAM by junction matching and vanishing point alignment. *IEEE Access*, *7*, 181800–181811.

Maire, Michael, Arbelaez, Pablo, Fowlkes, Charless, & Malik, Jitendra (2008). Using contours to detect and localize junctions in natural images. In *2008 IEEE conference on computer vision and pattern recognition* (pp. 1–8). IEEE.

Matsumoto, Yuya, Nakano, Gaku, & Ogura, Kazumine (2024). Indoor visual localization using point and line correspondences in dense colored point cloud. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 3616–3625).

Moezzi, Reza, Krcmarik, David, Hlava, Jaroslav, & Cýrus, Jindřich (2020). Hybrid SLAM modelling of autonomous robot with augmented reality device. *Materials Today: Proceedings*, *32*, 103–107.

Montemerlo, Michael (2002). FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proc. of AAAI02*.

Montemerlo, Michael, & Thrun, Sebastian (2003). Simultaneous localization and mapping with unknown data association using FastSLAM. *2*, In *2003 IEEE international conference on robotics and automation (cat. no. 03CH37422)* (pp. 1985–1991). IEEE.

Montiel, J. M. M., Tardós, Juan D., & Montano, Luis (2000). Structure and motion from straight line segments. *Pattern Recognition*, *33*(8), 1295–1307.

Mouragnon, Etienne, Lhuillier, Maxime, Dhome, Michel, Dekeyser, Fabien, & Sayd, Patrick (2009). Generic and real-time structure from motion using local bundle adjustment. *Image and Vision Computing*, *27*(8), 1178–1193.

Mourikis, Anastasios I., & Roumeliotis, Stergios I. (2007). A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Proceedings 2007 IEEE international conference on robotics and automation* (pp. 3565–3572). IEEE.

Mur-Artal, Raul, Montiel, Jose Maria Martinez, & Tardos, Juan D. (2015). ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, *31*(5), 1147–1163.

Mur-Artal, Raúl, & Tardós, Juan D. (2014). Fast relocalisation and loop closing in keyframe-based SLAM. In *2014 IEEE international conference on robotics and automation* (pp. 846–853). IEEE.

Mur-Artal, Raul, & Tardós, Juan D. (2017a). Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, *33*(5), 1255–1262.

Mur-Artal, Raúl, & Tardós, Juan D. (2017b). Visual-inertial monocular SLAM with map reuse. *IEEE Robotics and Automation Letters*, *2*(2), 796–803.

Ok, Ali Özgün, Wegner, Jan Dirk, Heipke, Christian, Rottensteiner, Franz, Sörgel, Uwe, & Toprak, Vedat (2012). Accurate reconstruction of near-epipolar line segments from stereo aerial images. *Photogrammetrie Fernerkundung Geoinformation*, *2012*, 345–358.

Pătrăucean, Viorica, Gurdjos, Pierre, & von Gioi, Rafael Grompone (2016). Joint a contrario ellipse and line detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(4), 788–802.

Pătrăucean, Viorica, Gurdjos, Pierre, & Von Gioi, Rafael Grompone (2012). A parameterless line segment and elliptical arc detector with enhanced ellipse fitting. In *Computer vision–ECCV 2012: 12th European conference on computer vision, Florence, Italy, October 7–13, 2012, proceedings, Part II 12* (pp. 572–585). Springer.

Pautrat, Rémi, Barath, Daniel, Larsson, Viktor, Oswald, Martin R, & Pollefeys, Marc (2023). Deeplsd: Line segment detection and refinement with deep image gradients. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 17327–17336).

Pautrat, Rémi, Lin, Juan-Ting, Larsson, Viktor, Oswald, Martin R, & Pollefeys, Marc (2021). SOLD2: Self-supervised occlusion-aware line description and detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11368–11378).

Pautrat, Rémi, Suárez, Iago, Yu, Yifan, Pollefeys, Marc, & Larsson, Viktor (2023). Gluestick: Robust image matching by sticking points and lines together. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9706–9716).

Peng, Xiongfeng, Liu, Zhihua, Li, Weiming, Tan, Ping, Cho, Soon Yong, & Wang, Qiang (2024). Dvi-slam: A dual visual inertial slam network. In *2024 IEEE international conference on robotics and automation* (pp. 12020–12026). IEEE.

Pumarola, Albert, Vakhitov, Alexander, Agudo, Antonio, Sanfeliu, Alberto, & Moreno-Noguer, Francese (2017). PL-SLAM: Real-time monocular visual SLAM with points and lines. In *2017 IEEE international conference on robotics and automation* (pp. 4503–4508). IEEE.

Qian, Kun, Zhao, Wei, Li, Kai, Ma, Xudong, & Yu, Hai (2019). Visual SLAM with BoPLW pairs using egocentric stereo camera for wearable-assisted substation inspection. *IEEE Sensors Journal*, *20*(3), 1630–1641.

Qiao, Chengyu, Bai, Tingming, Xiang, Zhiyu, Qian, Qi, & Bi, Yunfeng (2021). Superline: A robust line segment feature for visual slam. In *2021 IEEE/RSJ international conference on intelligent robots and systems* (pp. 5664–5670). IEEE.

Qin, Tong, Li, Peiliang, & Shen, Shaojie (2018). Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, *34*(4), 1004–1020.

Quan, Long, & Lan, Zhongdan (1999). Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *21*(8), 774–780.

Rohacz, Anke, Weißenfels, Stefan, & Strassburger, Steffen (2020). Concept for the comparison of intralogistics designs with real factory layout using augmented reality, SLAM and marker-based tracking. *Procedia CIRP*, *93*, 341–346.

Rong, Hanxiao, Gao, Yanbin, Guan, Lianwu, Ramirez-Serrano, Alex, Xu, Xu, & Zhu, Yunyu (2021). Point-line visual stereo slam using edlines and pl-bow. *Remote Sensing*, *13*(18), 3591.

Ronneberger, Olaf, Fischer, Philipp, & Brox, Thomas (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–mICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, Part III 18* (pp. 234–241). Springer.

Rosinol, Antoni, Abate, Marcus, Chang, Yun, & Carlone, Luca (2020). Kimera: an open-source library for real-time metric-semantic localization and mapping. In *2020 IEEE international conference on robotics and automation* (pp. 1689–1696). IEEE.

Rosten, Edward, & Drummond, Tom (2006). Machine learning for high-speed corner detection. In *Computer vision–ECCV 2006: 9th European conference on computer vision, Graz, Austria, May 7–13, 2006. Proceedings, Part I 9* (pp. 430–443). Springer.

Rublee, Ethan, Rabaud, Vincent, Konolige, Kurt, & Bradski, Gary (2011). ORB: An efficient alternative to SIFT or SURF. In *2011 international conference on computer vision* (pp. 2564–2571). IEEE.

Salaün, Yohann, Marlet, Renaud, & Monasse, Pascal (2016). Multiscale line segment detector for robust and accurate SfM. In *2016 23rd international conference on pattern recognition* (pp. 2000–2005). IEEE.

Sarlin, Paul-Edouard, DeTone, Daniel, Malisiewicz, Tomasz, & Rabinovich, Andrew (2020). Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4938–4947).

Seo, Dong-Uk, Lim, Hyungtae, Lee, Eungchang Mason, Lim, Hyunjun, & Myung, Hyun (2023). Enhancing robustness of line tracking through semi-dense epipolar search in line-based SLAM. In *2023 IEEE/RSJ international conference on intelligent robots and systems* (pp. 3483–3490). IEEE.

Shu, Fangwen, Wang, Jiaxuan, Pagani, Alain, & Stricker, Didier (2023). Structure plp-slam: Efficient sparse mapping and localization using point, line and plane for monocular, rgb-d and stereo cameras. In *2023 IEEE international conference on robotics and automation* (pp. 2105–2112). IEEE.

Si, Hanqian, Yu, Huai, Chen, Kuangyi, & Yang, Wen (2024). Point-line visual-inertial odometry with optimized line feature processing. *IEEE Transactions on Instrumentation and Measurement*.

Sinkhorn, Richard (1964). A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics*, *35*(2), 876–879.

Smith, Randall C., & Cheeseman, Peter (1986). On the representation and estimation of spatial uncertainty. *The International Journal of Robotics Research*, *5*(4), 56–68.

Strasdat, Hauke, Montiel, J. M. M., & Davison, Andrew J. (2010). Scale drift-aware large scale monocular SLAM. In *Robotics: science and systems (Vol. 2, No. 3)* (p. 5).

Sturm, Jürgen, Engelhard, Nikolas, Endres, Felix, Burgard, Wolfram, & Cremers, Daniel (2012). A benchmark for the evaluation of RGB-d SLAM systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems* (pp. 573–580). IEEE.

Suárez, Iago, Buenaposada, José M., & Baumela, Luis (2022). ELSED: Enhanced line segment drawing. *Pattern Recognition*, *127*, Article 108619.

Suárez, Iago, Muñoz, Enrique, Buenaposada, José M, & Baumela, Luis (2018). FSG: A statistical approach to line detection via fast segments grouping. In *2018 IEEE/RSJ international conference on intelligent robots and systems* (pp. 97–102). IEEE.

Sugiura, Takayuki, Torii, Akihiko, & Okutomi, Masatoshi (2015). 3D surface reconstruction from point-and-line cloud. In *2015 international conference on 3D vision* (pp. 264–272). IEEE.

Sun, Ke, Mohta, Kartik, Pfrommer, Bernd, Watterson, Michael, Liu, Sikang, Mulgaonkar, Yash, et al. (2018). Robust stereo visual inertial odometry for fast autonomous flight. *IEEE Robotics and Automation Letters*, *3*(2), 965–972.

Tateno, Keisuke, Tombari, Federico, Laina, Iro, & Navab, Nassir (2017). Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6243–6252).

Teng, Zhaoyu, Han, Bin, Cao, Jie, Hao, Qun, Tang, Xin, & Li, Zhaoyang (2023). PLI-SLAM: A tightly-coupled stereo visual-inertial SLAM system with point and line features. *Remote Sensing*, *15*(19), 4678.

Tomasi, Carlo, & Kanade, Takeo (1991). Detection and tracking of point. *International Journal of Computer Vision*, *9*(137–154), 3.

Vakhitov, Alexander, Funke, Jan, & Moreno-Noguer, Francesc (2016). Accurate and linear time pose estimation from points and lines. In *European conference on computer vision* (pp. 583–599). Springer.

Vakhitov, Alexander, & Lempitsky, Victor (2019). Learnable line segment descriptor for visual slam. *IEEE Access*, *7*, 39923–39934.

Von Gioi, Rafael Grompone, Jakubowicz, Jeremie, Morel, Jean-Michel, & Randall, Gregory (2008). LSD: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(4), 722–732.

Von Gioi, Rafael Grompone, Jakubowicz, Jérémie, Morel, Jean-Michel, & Randall, Gregory (2012). LSD: A line segment detector. *Image Processing on Line*, *2*, 35–55.

Von Stumberg, Lukas, Usenko, Vladyslav, & Cremers, Daniel (2018). Direct sparse visual-inertial odometry using dynamic marginalization. In *2018 IEEE international conference on robotics and automation* (pp. 2510–2517). IEEE.

Wan, Eric A., & Van Der Merwe, Rudolph (2000). The unscented Kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 adaptive systems for signal processing, communications, and control symposium (cat. no. 00EX373)* (pp. 153–158). Ieee.

Wang, Sen, Clark, Ronald, Wen, Hongkai, & Trigoni, Niki (2017). Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *2017 IEEE international conference on robotics and automation* (pp. 2043–2050). IEEE.

Wang, Haobo, Guan, Lianwu, Yu, Xilin, & Zhang, Zibin (2022). PL-ISLAM: An accurate monocular visual-inertial SLAM with point and line features. In *2022 IEEE international conference on mechatronics and automation* (pp. 1141–1146). IEEE.

Wang, Zhiheng, Wu, Fuchao, & Hu, Zhanyi (2009). MSLD: A robust descriptor for line matching. *Pattern Recognition*, *42*(5), 941–953.

Wang, Qiuyuan, Yan, Zike, Wang, Junqiu, Xue, Fei, Ma, Wei, & Zha, Hongbin (2021). Line flow based simultaneous localization and mapping. *IEEE Transactions on Robotics*, *37*(5), 1416–1432.

Wang, Ze, Yang, Kailun, Shi, Hao, Li, Peng, Gao, Fei, Bai, Jian, et al. (2023). LF-VISLAM: A SLAM framework for large field-of-view cameras with negative imaging plane on mobile agents. *IEEE Transactions on Automation Science and Engineering*, *21*(4), 6321–6335.

Wang, Ze, Yang, Kailun, Shi, Hao, Li, Peng, Gao, Fei, & Wang, Kaiwei (2022). LF-VIO: A visual-inertial-odometry framework for large field-of-view cameras with negative plane. In *2022 IEEE/RSJ international conference on intelligent robots and systems* (pp. 4423–4430). IEEE.

Wang, Ze, Yang, Kailun, Shi, Hao, Zhang, Yufan, Xu, Zhijie, Gao, Fei, et al. (2024). LF-PGVIO: a visual-inertial-odometry framework for large field-of-view cameras using points and geodesic segments. *IEEE Transactions on Intelligent Vehicles*.

Wang, Zikai, Zhong, Baojiang, Chen, Xueyuan, & Zheng, Hangjia (2024). MPG-LSD: A high-quality line segment detector based on multi-scale perceptual grouping. *Pattern Recognition*, *149*, Article 110286.

Wei, Hao, Tang, Fulin, Xu, Zewen, Zhang, Chaofan, & Wu, Yihong (2021). A point-line vio system with novel feature hybrids and with novel line predicting-matching. *IEEE Robotics and Automation Letters*, *6*(4), 8681–8688.

Wei, Dong, Zhang, Yongjun, Liu, Xinyi, Li, Chang, & Li, Zhuofan (2021). Robust line segment matching across views via ranking the line-point graph. *ISPRS Journal of Photogrammetry and Remote Sensing*, *171*, 49–62.

Wen, Huanyu, Tian, Jindong, & Li, Dong (2020). PLS-VIO: Stereo vision-inertial odometry based on point and line features. In *2020 international conference on high performance big data and intelligent systems* (pp. 1–7). IEEE.

Xia, Linlin, Meng, Deang, Zhang, Jingjing, Zhang, Daochang, & Hu, Zhiqi (2022). Visual-inertial simultaneous localization and mapping: Dynamically fused point-line feature extraction and engineered robotic applications. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1–11.

Xu, Bo, Chen, Yu, Zhang, Shoujian, & Wang, Jingrong (2020). Improved point–line visual–inertial odometry system using helmert variance component estimation. *Remote Sensing*, *12*(18), 2901.

Xu, Kuan, Hao, Yuefan, Yuan, Shenghai, Wang, Chen, & Xie, Lihua (2023). Airvo: An illumination-robust point-line visual odometry. In *2023 IEEE/RSJ international conference on intelligent robots and systems* (pp. 3429–3436). IEEE.

Xu, Kuan, Hao, Yuefan, Yuan, Shenghai, Wang, Chen, & Xie, Lihua (2025). Airslam: An efficient and illumination-robust point-line visual slam system. *IEEE Transactions on Robotics*.

Xu, Zezhong, Shin, Bok-Suk, & Klette, Reinhard (2015). Closed form line-segment extraction using the Hough transform. *Pattern Recognition*, *48*(12), 4012–4023.

Xu, Bo, Wang, Peng, He, Yijia, Chen, Yu, Chen, Yongnan, & Zhou, Ming (2022). Leveraging structural information to improve point line visual-inertial odometry. *IEEE Robotics and Automation Letters*, *7*(2), 3483–3490.

Xu, Lei, Yin, Hesheng, Shi, Tong, Jiang, Di, & Huang, Bo (2022). EPLF-vins: Real-time monocular visual-inertial SLAM with efficient point-line flow features. *IEEE Robotics and Automation Letters*, *8*(2), 752–759.

Xue, Nan, Bai, Song, Wang, Fudong, Xia, Gui-Song, Wu, Tianfu, & Zhang, Liangpei (2019). Learning attraction field representation for robust line segment detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1595–1603).

Xue, Nan, Bai, Song, Wang, Fu-Dong, Xia, Gui-Song, Wu, Tianfu, Zhang, Liangpei, et al. (2019). Learning regional attraction for line segment detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(6), 1998–2013.

Xue, Nan, Wu, Tianfu, Bai, Song, Wang, Fu-Dong, Xia, Gui-Song, Zhang, Liangpei, et al. (2023). Holistically-attracted wireframe parsing: From supervised to self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(12), 14727–14744.

Yan, Jinjin, Zheng, Youbing, Yang, Jinquan, Mihaylova, Lyudmila, Yuan, Weijie, & Gu, Fuqiang (2024). PLPF-VSLAM: An indoor visual SLAM with adaptive fusion of point-line-plane features. *Journal of Field Robotics*, *41*(1), 50–67.

Yang, Yulin, Geneva, Patrick, Eckenhoff, Kevin, & Huang, Guoquan (2019). Visual-inertial odometry with point and line features. In *2019 IEEE/RSJ international conference on intelligent robots and systems* (pp. 2447–2454). IEEE.

Yang, Shichao, Song, Yu, Kaess, Michael, & Scherer, Sebastian (2016). Pop-up slam: Semantic monocular plane slam for low-texture environments. In *2016 IEEE/RSJ international conference on intelligent robots and systems* (pp. 1222–1229). IEEE.

Yang, Haozhi, Yuan, Jing, Gao, Yuanxi, Sun, Xingyu, & Zhang, Xuebo (2023). UPLP-SLAM: Unified point-line-plane feature fusion for RGB-D visual SLAM. *Information Fusion*, *96*, 51–65.

Yao, Hexiong, Ma, Yuexin, Li, Peijing, Zhai, Chunlei, Song, Jiangbo, Ouyang, Mingjun, et al. (2024). SG-VIO: Monocular visual-inertial odometry with tightly coupled structural lines and gravity to avoid degeneracy. *IEEE Internet of Things Journal*.

Yi, Kwang Moo, Trulls, Eduard, Lepetit, Vincent, & Fua, Pascal (2016). Lift: Learned invariant feature transform. In *Computer vision–ECCV 2016: 14th European conference, Amsterdam, the Netherlands, October 11–14, 2016, proceedings, Part VI 14* (pp. 467–483). Springer.

Yin, Hesheng, Li, Shaomiao, Tao, Yu, Guo, Junlong, & Huang, Bo (2022). Dynam-SLAM: An accurate, robust stereo visual-inertial SLAM method in dynamic environments. *IEEE Transactions on Robotics*, *39*(1), 289–308.

Yoon, Sungho, & Kim, Ayoung (2021). Line as a visual sentence: Context-aware line descriptor for visual localization. *IEEE Robotics and Automation Letters*, *6*(4), 8726–8733.

Yu, Hongshan, Fu, Qiang, Yang, Zhengeng, Tan, Lei, Sun, Wei, & Sun, Mingui (2018). Robust robot pose estimation for challenging scenes with an RGB-D camera. *IEEE Sensors Journal*, *19*(6), 2217–2229.

Yu, Xilin, Gan, Huiqi, Guan, Lianwu, Yang, Zaizhu, Zhang, Zibin, & Lin, Kaihong (2023). DPL-SLAM: A SLAM system by deep learning point features and line features integration. In *2023 IEEE international conference on mechatronics and automation* (pp. 2372–2377). IEEE.

Yu, Huai, Li, Hao, Yang, Wen, Yu, Lei, & Xia, Gui-Song (2023). Detecting line segments in motion-blurred images with events. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yu, Qida, Xu, Guili, Cheng, Yuehua, & Zhu, Zheng H. (2020). PLSD: A perceptually accurate line segment detection approach. *IEEE Access*, *8*, 42595–42607.

Yuan, Wang, Li, Zhijun, & Su, Chun-Yi (2019). Multisensor-based navigation and control of a mobile service robot. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *51*(4), 2624–2634.

Yuan, Chaofeng, Xu, Yuelei, & Zhou, Qing (2023). PLDS-SLAM: point and line features SLAM in dynamic environment. *Remote Sensing*, *15*(7), 1893.

Yunus, Raza, Li, Yanyan, & Tombari, Federico (2021). Manhattanslam: Robust planar tracking and mapping leveraging mixture of manhattan frames. In *2021 IEEE international conference on robotics and automation* (pp. 6687–6693). IEEE.

Zeng, Duofeng, Liu, Xiaotao, Huang, Kangjin, & Liu, Jing (2024). EPL-VINS: Efficient point-line fusion visual-inertial SLAM with LK-RG line tracking method and 2-dof line optimization. *IEEE Robotics and Automation Letters*.

Zhang, BaoSheng (2022). PLD-SLAM: A real-time visual SLAM using points and line segments in dynamic scenes. ArXiv preprint arXiv:2207.10916.

Zhang, Baosheng, Dong, Yanpeng, Zhao, Yufei, & Qi, Xianyu (2024). Dynpl-slam: A robust stereo visual slam system for dynamic scenes using points and lines. *IEEE Transactions on Intelligent Vehicles*.

Zhang, Chengran, Fang, Zheng, Luo, Xingjian, & Liu, Wei (2023). Accurate and robust visual SLAM with a novel ray-to-ray line measurement model. *Image and Vision Computing*, *140*, Article 104837.

Zhang, Lilian, & Koch, Reinhard (2013). An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency. *Journal of Visual Communication and Image Representation*, *24*(7), 794–805.

Zhang, Guoxuan, Lee, Jin Han, Lim, Jongwoo, & Suh, Il Hong (2015). Building a 3-D line-based map using stereo SLAM. *IEEE Transactions on Robotics*, *31*(6), 1364–1377.

Zhang, Shichao, Li, Xuelong, Zong, Ming, Zhu, Xiaofeng, & Wang, Ruili (2017). Efficient kNN classification with different numbers of nearest neighbors. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(5), 1774–1785.

Zhang, Tong, Liu, Chunjiang, Li, Jiaqi, Pang, Minghui, & Wang, Mingang (2022). A new visual inertial simultaneous localization and mapping (SLAM) algorithm based on point and line features. *Drones*, *6*(1), 23.

Zhang, Haotian, Luo, Yicheng, Qin, Fangbo, He, Yijia, & Liu, Xiao (2021). Elsd: Efficient line segment detector and descriptor. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 2969–2978).

Zhang, Jiyuan, Zeng, Gang, & Zha, Hongbin (2019). Structure-aware SLAM with planes and lines in man-made environment. *Pattern Recognition Letters*, *127*, 181–190.

Zhao, Zhangzhen, Song, Tao, Xing, Bin, Lei, Yu, & Wang, Ziqin (2022). PLI-VINS: Visual-inertial SLAM based on point-line feature fusion in indoor environment. *Sensors*, *22*(14), 5457.

Zhao, Weiqiang, Sun, Hang, Zhang, Xinyu, & Xiong, Yijin (2023). Visual SLAM combining lines and structural regularities: Towards robust localization. *IEEE Transactions on Intelligent Vehicles*.

Zhao, Yipu, & Vela, Patricio A. (2018). Good line cutting: Towards accurate pose tracking of line-assisted VO/VSLAM. In *Proceedings of the European conference on computer vision* (pp. 516–531).

Zheng, Feng, Tsai, Grace, Zhang, Zhe, Liu, Shaoshan, Chu, Chen-Chi, & Hu, Hongbing (2018). Trifo-VIO: Robust and efficient stereo visual inertial odometry using points and lines. In *2018 IEEE/RSJ international conference on intelligent robots and systems* (pp. 3686–3693). IEEE.

Zhi, Meiyu (2022). Robust key-frame stereo visual SLAM with low-threshold point and line features. ArXiv preprint arXiv:2207.05244.

Zhou, Lipu, Huang, Guoquan, Mao, Yinian, Wang, Shengze, & Kaess, Michael (2022). Edplvo: Efficient direct point-line visual odometry. In *2022 international conference on robotics and automation* (pp. 7559–7565). IEEE.

Zhou, Yichao, Qi, Haozhi, & Ma, Yi (2019). End-to-end wireframe parsing. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 962–971).

Zhou, Lipu, Wang, Shengze, & Kaess, Michael (2021). DPLVO: Direct point-line monocular visual odometry. *IEEE Robotics and Automation Letters*, *6*(4), 7113–7120.

Zhou, Fei, Zhang, Limin, Deng, Chaolong, & Fan, Xinyue (2021). Improved point-line feature based visual SLAM method for complex environments. *Sensors*, *21*(13), 4604.

Zhou, Chao, Zhang, Hong, Shen, Xiaoyong, & Jia, Jiaya (2017). Unsupervised learning of stereo matching. In *Proceedings of the IEEE international conference on computer vision* (pp. 1567–1575).

Zhu, Yeqing, Jin, Rui, Lou, Tai-shan, & Zhao, Liangyu (2021). PLD-VINS: RGBD visual-inertial SLAM with point and line features. *Aerospace Science and Technology*, *119*, Article 107185.

Zou, Danping, Wu, Yuanxin, Pei, Ling, Ling, Haibin, & Yu, Wenxian (2019). Structvio: Visual-inertial odometry with structural regularity of man-made environments. *IEEE Transactions on Robotics*, *35*(4), 999–1013.

Zubizarreta, Jon, Aguinaga, Iker, & Montiel, Jose Maria Martinez (2020). Direct sparse mapping. *IEEE Transactions on Robotics*, *36*(4), 1363–1370.

Zuo, Xingxing, Xie, Xiaojia, Liu, Yong, & Huang, Guoquan (2017). Robust visual SLAM with point and line features. In *2017 IEEE/RSJ international conference on intelligent robots and systems* (pp. 1775–1782). IEEE.