IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 November 2023, accepted 6 December 2023, date of publication 12 December 2023,
date of current version 15 December 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3341489*

SURVEY

# A Survey of Visual SLAM Methods

**ALI RIDA SAHILI[1], SAIFELDIN HASSAN[2], SABER MUAWIYAH SAKHRIEH[2], JINANE MOUNSEF[2], (Member, IEEE), NOEL MAALOUF[1], (Member, IEEE), BILAL ARAIN[3], AND TAREK TAHA[4]**

[1]Department of Electrical and Computer Engineering, Lebanese American University, Byblos 4504, Lebanon
[2]Department of Electrical Engineering and Computing, Rochester Institute of Technology, Dubai, United Arab Emirates
[3]College of Computing and Informatics, University of Sharjah, Sharjah, United Arab Emirates
[4]Robotics Laboratory, Dubai Future Labs, Dubai, United Arab Emirates

Corresponding author: Jinane Mounsef (jmbcad@rit.edu)

**ABSTRACT** In the evolving landscape of modern robotics, Visual SLAM (V-SLAM) has emerged over the past two decades as a powerful tool, empowering robots with the ability to navigate and map their surroundings. While these methods are traditionally confined to static environments, there has been a growing interest in developing V-SLAM to handle dynamic and realistic scenes. This survey offers a comprehensive overview of the current state-of-the-art V-SLAM methods, including their strengths and weaknesses. The paper also identifies the limitations of existing techniques and proposes potential research directions for future advancements. In addition, it provides an overview of commonly used datasets to evaluate the performance of V-SLAM methods. This survey sheds valuable insights into areas that need additional research to benefit V-SLAM development, including challenges related to limited scalability for systems with multiple agents, sensitivity to lighting changes, high computational cost, and performance issues in noisy environments.

**INDEX TERMS** Visual-SLAM, semantics, dynamic environment, noisy environment, simultaneous localization and mapping, computer vision.

## I. INTRODUCTION

Simultaneous localization and mapping (SLAM) is a crucial component in autonomous robotic navigation, Augmented Reality (AR), and Virtual Reality (VR) technologies. It serves as the backbone for autonomous driving vehicles [1], as well as many other applications, including indoor robotics, such as warehouse and hospital robots [2]. Moreover, it proves valuable in the development of general service robots. SLAM involves the creation of a map of the surrounding environment and the localization of the agent within it. This process enables the identification of landmarks and the implementation of intelligent navigation solutions [3]. Visual SLAM, also known as V-SLAM, is a system that utilizes a visual-based sensor, such as an RGB camera, to implement SLAM. V-SLAM comprises several fundamental building blocks, which include map trajectory, initialization, data association, loop closure (revisiting location), relocation, and estimation algorithms [4]. V-SLAM can be classified using different methods, such as camera type categorization, as seen in [5]. Alternatively, it can be divided into two main categories: class-aware and instance-aware, as in [6]. Nonetheless, both classifications rely on the method's ability to accurately identify and categorize objects within the surrounding environment.

Over the years, there has been a significant surge in research papers on V-SLAM. According to the Web of Science database [7], the number of published research papers on V-SLAM has experienced substantial growth, increasing from around 1,000 in 2010 to more than 10,000 by mid-2023. These results were obtained by employing keywords associated with V-SLAM, encompassing the various sensors and methodologies employed in V-SLAM. Figure 1 presents a visual representation of the annual publication trend in the field of V-SLAM over the past two decades. In contrast, Semantic SLAM research has not seen the same level of proliferation as traditional V-SLAM methods. The literature currently comprises approximately

The associate editor coordinating the review of this manuscript and approving it for publication was Mu-Yen Chen.

954 research papers on Semantic SLAM, indicating the potential for further exploration and investigation in this area. Figure 2 shows the distribution of the number of publications across different fields of V-SLAM categorized by sensor type and approach type, providing insights into the research landscape. In Figure 2(a), it is noted that the majority of publications implement Monocular SLAM, while the newly emerging Event SLAM has received the least attention. When categorizing the methods by approach in Figure 2(b), Feature-based SLAM is the most researched topic, followed by Semantic SLAM and Direct SLAM. The categorization of V-SLAM algorithms based on the type of camera used and the followed approach is discussed in detail later in this paper. Notably, China, USA, and Germany are the most active countries in V-SLAM research, contributing to more than 70% of the total research conducted in the field, as reported by the Web of Science database [7]. Furthermore, Figure 3 highlights a subset of the most productive and influential institutions in the field of V-SLAM, as gauged by the number of publications they have produced in recent years. The mentioned institutions in Figure 3 collectively contribute to over 25% of the total research conducted in the field of V-SLAM. These institutions play a significant role in advancing the research and development of V-SLAM technologies.

The surge in V-SLAM methods has led to numerous survey papers covering specific topics within V-SLAM. Zhang et al. [8] compare different direct and indirect RGB-D SLAM systems, discussing their capabilities in particular tracking, mapping, and loop detection across different application scenarios. The direct camera tracking methods minimize the photometric error over corresponding pixels in two frames. These methods are robust, compared to indirect methods, in low-texture environments since they rely on the photometric value of selected pixels within a scene. Examples of the direct methods include KinectFusion [9], RGBDTAM [10], and ID-RGBDO [11], which can estimate the camera motion either by using the frame-to-model mechanisms or by using the most informative points in the camera frame. On the other hand, the indirect methods, such as ORB-SLAM2 [12], RGB-D SLAM [13], and Plane-Edge-SLAM [14], minimize the geometric errors over matching features. The indirect methods have the advantages in the situations of rapid camera motion and vigorous rotations. In these situations, the direct method is easier to encounter tracking loss, and both the accuracy and robustness are worse than indirect methods. Also, the indirect methods are found to be more effective when using rolling shutter cameras. The hybrid methods minimize a combination of direct or indirect methods to estimate the camera pose. The hybrid methods include CPA-SLAM [15], BundleFusion [16], and KDP-SLAM [17], which combine both photometric error and geometric errors for camera pose estimation. The RGB-D SLAM approaches build a map of the scene either using point-based methods or volumetric methods. Point-based methods, such as DVO-SLAM [18], ORB-SLAM2 [12], BAD-SLAM [19], and ElasticFusion [20], represent environments using graphs or surfels. The volumetric methods, such as Kintinuous [21], Voxel Hashing [22], and BundleFusion [16], represent the scene geometry by an implicit truncated signed distance (TSDF). The RGB-D SLAM methods also differ in detecting the loop closures. For example, DVO-SLAM [18] detects loop closures within a sphere of a predefined radius around the keyframe position, while Kintinuous [21] and ORB-SLAM2 [12] use bag-of words-based DBoW [23] loop detector. The survey concludes that accurate pose estimation remains a challenge in low texture and noisy environments. The relevant SLAM systems were evaluated on the publicly available RGB-D datasets. It was observed that the performance of different RGB-D SLAM methods varies in scenes prone to illumination changes. Nonetheless, the adaptability of different algorithms can be analyzed in different situations and scenarios.

The paper by Aizat et al. [24] provides an overview of navigation techniques in the context of Automated Ground Vehicle (AGV) robots operating in dynamic environments. The strategies are categorized into classical, global, and artificial intelligence-based approaches, discussing local methods like magnetic tape and global algorithms, such as A*, D* Lite, and Dijkstra [24]. Key contributions in this paper include highlighting the use of artificial intelligence techniques in AGV navigation, recognizing the trend towards algorithm combination, and emphasizing the importance of multi-AGV systems in real-world applications. In the discussion of their findings, Aizat et al. [24] highlight a gap in the practical testing of these approaches using physical hardware [24]. This presents a potential area for future work in the field of autonomous navigation. Mokssit et al. [25] offer a comprehensive examination of deep learning methods applied to the field of V-SLAM. They introduce a novel taxonomy for this subject, emphasizing the significant impact of leveraging deep learning strategies in enhancing the performance of V-SLAM [25]. By incorporating deep learning architectures, Mokssit et al. argue that robots can effectively capture intricate and challenging-to-model environmental features, mitigating uncertainties associated with visual sensory data [25]. This results in more robust solutions for real-world applications, making deep learning a compelling alternative to traditional hand-crafted approaches. The paper highlights the potential of deep learning methods to outperform classical methods in challenging scenarios, including variable illuminations, repetitive textures, occlusions, and dynamic elements, underlining the promise of deep learning in equipping robots with the ability to perceive, understand, and act effectively in real-world environments [25]. The paper proposes several research directions in deep learning with SLAM by (1) expanding the horizon of semantic scene understanding, (2) reducing computational demands, (3) improving interpretability, (4) providing more generic datasets, (5) extending probabilistic-based V-SLAM to 3D

environments, and (6) validating deep learning algorithms in real-world SLAM applications.

In [2], Barros et al. provided a comprehensive overview of three V-SLAM approaches: visual-only, visual-inertial, and RGB-D SLAM. The discussion delved into the key algorithms for each approach, employing diagrams and flowcharts for clarity. Furthermore, the authors introduced several factors influencing system accuracy and hardware implementation, encompassing the algorithm used, map density, global optimization, loop closing, and integrated systems. Furthermore, the paper discusses addressing challenges and future directions in the V-SLAM field, with a specific emphasis on deep learning algorithms, managing dynamic scenes, and exploring semantic-based Algorithms. Taketomi et al. [26] systematically categorized and summarized V-SLAM algorithms from both technical and historical perspectives. Their classification involved grouping V-SLAM algorithms into feature-based, direct-based, and RGB-D-based approaches. The survey focused particularly on algorithms proposed between 2010 and 2016, a period marked by considerable advancements in the field.

The objective of our paper is to provide a comprehensive review of V-SLAM methods, with a particular focus on modern V-SLAM methods that are more suited for dynamic and highly-dynamic environments that feature numerous moving objects. Unlike the survey papers in [8], [24], and [25], this paper does not limit itself to describing specific types of robots, sensors, or strategies but instead provides a broader and more holistic perspective on the field of V-SLAM. Moreover, compared to [2], our work provides a detailed overview of additional V-SLAM approaches, such as event-based and multimodal SLAM. Also, this paper introduces a different categorization algorithm, which focuses on how the V-SLAM system extracts information from images rather than being based on the type of camera employed. While [26] concentrates on a defined timeframe spanning 2010 to 2016, our survey delves into contemporary V-SLAM algorithms, primarily those proposed in recent years. The paper also discusses traditional V-SLAM methods that do not take into account moving objects in the scene as they form the foundation of modern V-SLAM implementations. In contrast to traditional SLAM methods, modern V-SLAM approaches are designed to deal with moving objects and changing environments, making them better equipped to handle real-world scenarios and overcome their limitations. Specifically, these methods address the issue of declining localization performance caused by the presence of dynamic objects in the scene, which can disrupt the mapping component of the system. Additionally, this paper highlights advanced V-SLAM applications that go beyond conventional setups. These advanced methods include software-based optimizations and sophisticated techniques that combine multiple sensors to improve accuracy and robustness. The review also covers novel approaches that utilize optical character recognition (OCR) for localization, enabling the agent to recognize its current location based on text in the environment [27]. Furthermore, it explores methods that localize the agent's position by recognizing objects in the scene [28]. The paper also reviews methods that incorporate additional sensors, such as the integration of Light Detection and Ranging (LiDAR) sensors into V-SLAM systems to improve robustness. For instance, the authors in [5] propose a system that combines Adaptive Monte Carlo Localization (AMCL) with an RGB-D camera and a 2D LiDAR solution. This integration aims to address AMCL's limitation in identifying and localizing the agent accurately in repetitive environments like long hallways. Additionally, some V-SLAM methods prioritize software optimization. For instance, the authors in [29] present an approach that enhances V-SLAM performance by using random sample consensus (RANSAC)-based algorithms to eliminate moving objects from the scene, thus improving localization accuracy.

The paper's contributions can be summarized as follows:

1) Summarizing key papers, algorithms, and methodologies that have been proposed in the field of V-SLAM and categorizing the modern methods, specifically introduced to handle dynamic scenes, into dynamic-aware and dynamic-inclusive.
2) Summarizing publicly available V-SLAM datasets that are commonly used to evaluate the performance of V-SLAM methods.
3) Identifying underdeveloped areas of research in V-SLAM, such as addressing challenges in noisy and unstructured environments, scalability to large settings, and real-time applicability, and highlighting potential improvements for existing methods.

This paper offers insights into the diverse sensors employed in V-SLAM (see Section II). Following that, the categorization of V-SLAM is thoroughly examined in Section III. Subsequently, Section IV delves into a review of contemporary V-SLAM methods. Additionally, the paper explores methods that prioritize software-based optimizations to enhance robustness, as discussed in Section V. Advances in scene understanding are also covered in Section VI. Commonly used datasets and evaluation tools are summarized and discussed in Section VII. Moreover, Section VIII provides a critical analysis of the current limitations of V-SLAM methods and offers recommendations for future research directions based on the current state of V-SLAM methods. Finally, a comprehensive summary of the findings is provided in Section IX, concluding the paper.

## II. TYPES OF SENSORS USED IN V-SLAM

SLAM systems have the ability to deduce the present location, approximate the path, and create a map of the surroundings using the data collected by their sensors [30]. Laser-based SLAM relies on precise distance measurements from sensors, such as LiDAR, excelling in accuracy and low-light environments. However, it can be costly and requires high processing power. Inertial-based techniques, which use accelerometers and gyroscopes, offer continuous motion tracking with low latency but are prone to drift

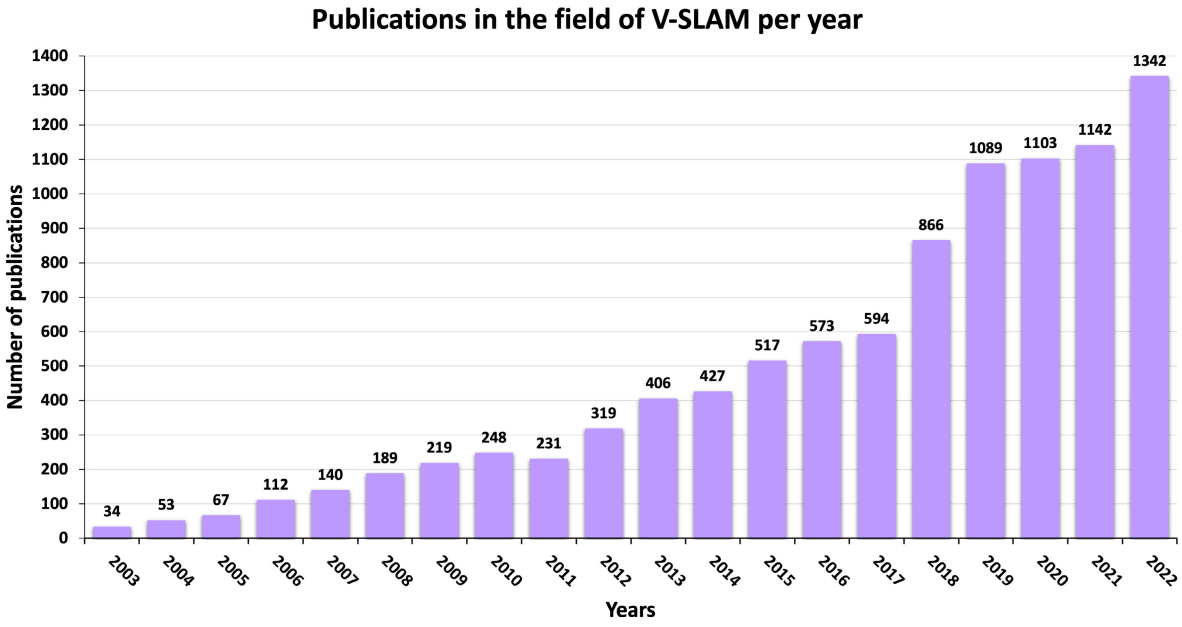
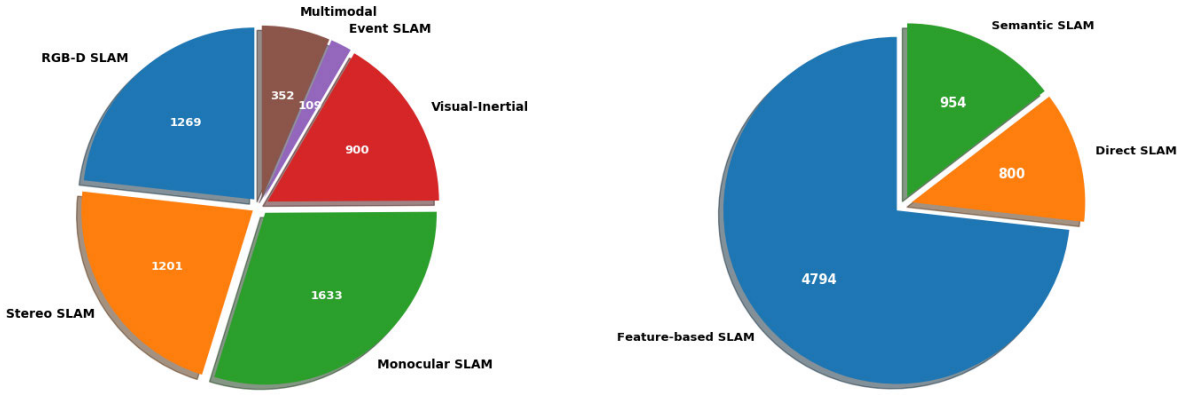



**FIGURE 1.** Annual publication trend in the field of V-SLAM over the past two decades, based on Web of Science database [7].




(a) By sensor type (b) By approach

**FIGURE 2.** Distribution of the number of publications across different fields of V-SLAM by sensor type and by approach, based on Web of Science database [7].

and lack environmental information [31]. V-SLAM relies on visual sensors as the main data input for its application, offering cost-effectiveness and rich visual information about the environment. Nevertheless, V-SLAM may encounter challenges in low-light conditions and contend with visual ambiguities. However, V-SLAM techniques can be further enhanced by incorporating additional sensors such as an Inertial Measurement Unit (IMU) or LiDAR alongside the camera, which provide valuable information about the camera's orientation and movement. Moreover, V-SLAM algorithms can benefit from utilizing a depth sensor alongside the camera aiding in motion and positioning estimation. In V-SLAM, four main types of visual sensors are commonly used: monocular, stereo, RGB-D, and event cameras [1], [32]. A comparison of these camera types is presented in Table 1. In this section, we explore six potential categories of V-SLAM systems, determined by the type of camera used and the integrated sensors within the system.

### A. MONOCULAR SLAM

Monocular cameras, while cost-effective and straightforward in design, encounter the scale ambiguity problem, making it challenging to accurately estimate landmark depth during map construction [33]. Furthermore, when a monocular camera remains stationary or undergoes only rotational movement, it lacks the ability to capture pixel distance information. These limitations restrict the accurate perception of depth and motion in the scene. Despite these challenges, monocular SLAM methods are widely used in applications where only a single camera is available, or where budget and

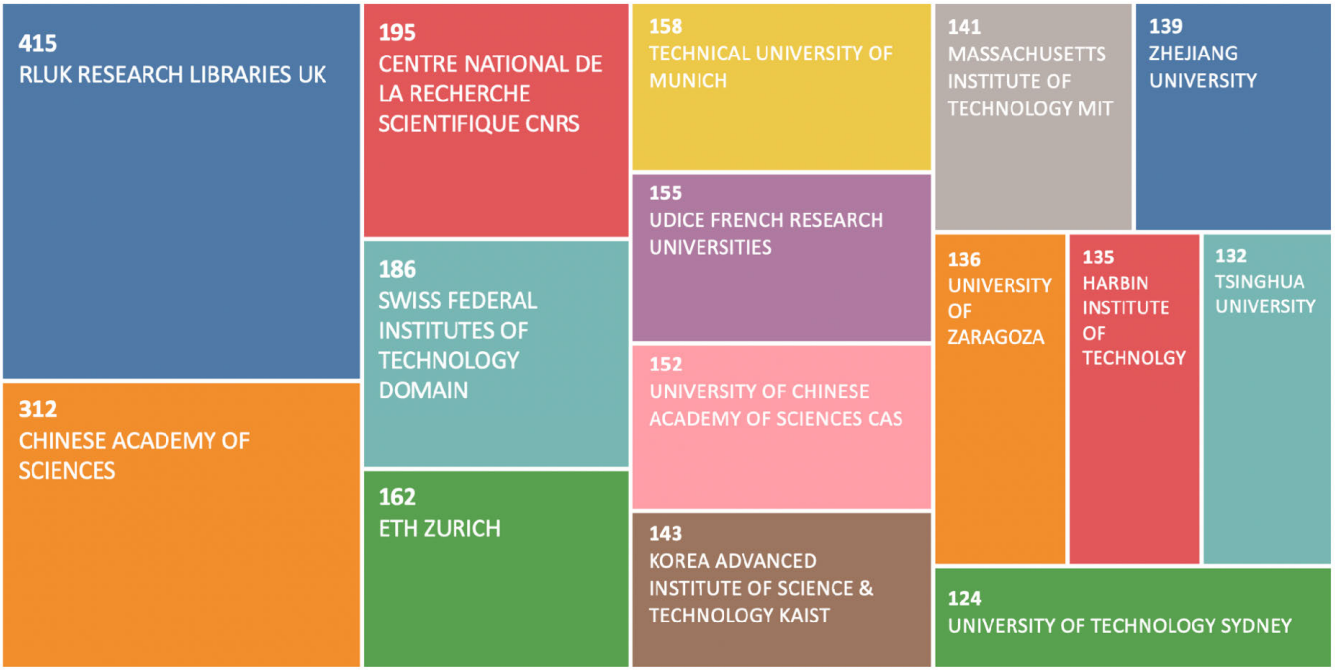



**FIGURE 3.** Subset of the most productive and influential institutions in the field of V-SLAM in recent years, based on Web of Science database [7].

**TABLE 1.** A comparison of camera types used in V-SLAM methods.

| Camera | Advantages | Disadvantages |
|---|---|---|
| Monocular | • Inexpensive and simple design.<br>• Versatile for indoor and outdoor environments. | • Cannot accurately estimate landmark depth.<br>• Cannot obtain pixel distance in a static state. |
| Stereo | • Can calculate pixel depth in a static state.<br>• Versatile for indoor and outdoor environments. | • Complex calibration and parameter configuration.<br>• Consumes more CPU. |
| RGB-D | • Can directly measure pixel depth by physical measurement.<br>• Consumes less computing resources. | • Narrow measurement range.<br>• Can be affected by sunlight interference. |
| Event | • High dynamic range.<br>• Consumes less power.<br>• Not affected by motion blur. | • Requires specific data processing techniques. |

power constraints limit the use of more advanced sensors. In such scenarios, monocular cameras provide a practical and viable solution for visual-based localization and mapping tasks. Figure 4 illustrates the structure of a typical Monocular SLAM approach.

### B. STEREO SLAM

These SLAM methods use a pair of cameras with known relative positions to estimate the camera trajectory and build a 3D map of the environment. Binocular or stereo cameras offer the advantage of calculating pixel depth even when stationary, providing more accurate depth information compared to monocular cameras, especially in outdoor settings [35]. Nevertheless, it is worth noting that these cameras require careful calibration of their setup, which can be a challenging task. Additionally, stereo SLAM has a higher computational cost as the system must process twice as much image information from the two cameras, which may impact real-time performance depending on the

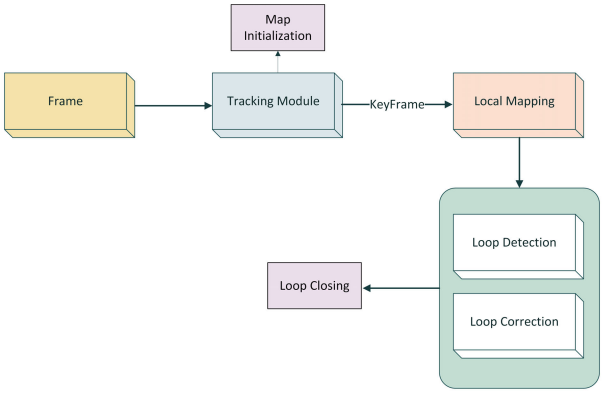



**FIGURE 4.** Monocular SLAM diagram, adapted from [34].

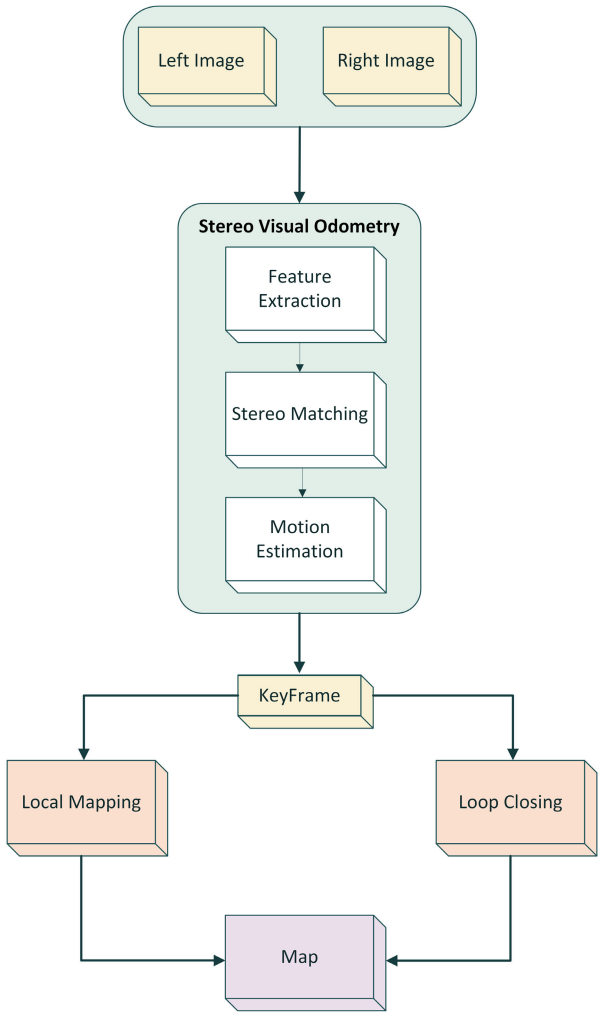



**FIGURE 5.** Stereo SLAM diagram, adapted from [12] and [36].

computational resources available [1]. Figure 5 illustrates the structure of a typical Stereo SLAM approach.

### C. RGB-D SLAM

RGB-D SLAM methods use a combination of an RGB camera and a depth sensor, such as structured-light or time-of-flight (TOF) sensor, to directly measure pixel depth,

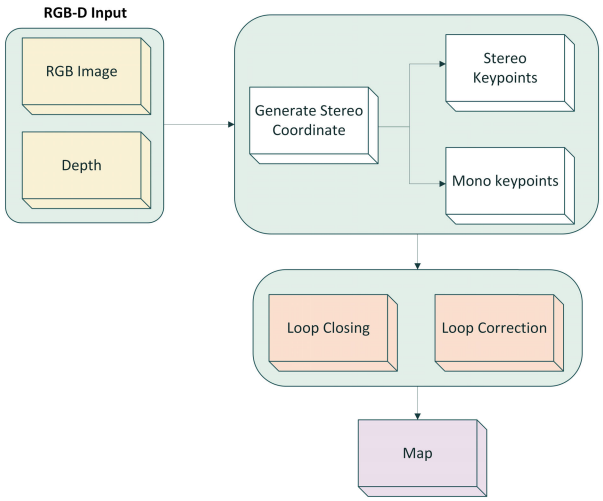



**FIGURE 6.** RGB-D SLAM diagram, adapted from [12].

simplifying the depth estimation process [37]. These methods excel in providing accurate and precise depth information, making them popular choices for well-lit indoor settings. However, RGB-D cameras have certain limitations. Their measurement range is restricted, and they are not well-suited for outdoor applications due to potential interference from sunlight, leading to unreliable mapping results [38]. Moreover, the efficiency of RGB-D SLAM in dynamic environments can depend on factors such as scene complexity, the speed of moving objects, and the quality of sensor data. To address the challenges posed by dynamic environments, researchers have proposed various techniques. These include the use of multiple sensors, integration of motion models, and the detection and removal of moving objects. By employing these strategies, the impact of dynamic elements on RGB-D SLAM performance can be mitigated, enhancing the system's overall robustness and accuracy. Figure 6 illustrates the structure of a typical RGB-D SLAM approach.

### D. EVENT-BASED SLAM

Unlike conventional cameras that capture frames at a fixed rate, event cameras, also known as dynamic vision sensors (DVSs), operate differently by reporting changes in brightness or intensity asynchronously and at a high temporal resolution, resulting in event data. Due to this unique characteristic, event cameras are well-suited for tracking fast motion and high-speed dynamics, even in challenging low-light conditions, with minimal latency [39]. The integration of event cameras as sensors for V-SLAM systems represents a novel and emerging area of research [40]. These methods use event cameras to capture the motion information from the environment, which is then used to estimate the camera's 6-degree-of-freedom (6-DoF) pose and reconstruct the 3D structure of the scene [41]. Event-based SLAM offers several advantages over traditional V-SLAM methods, including high accuracy and resilience to fast motion, low latency,

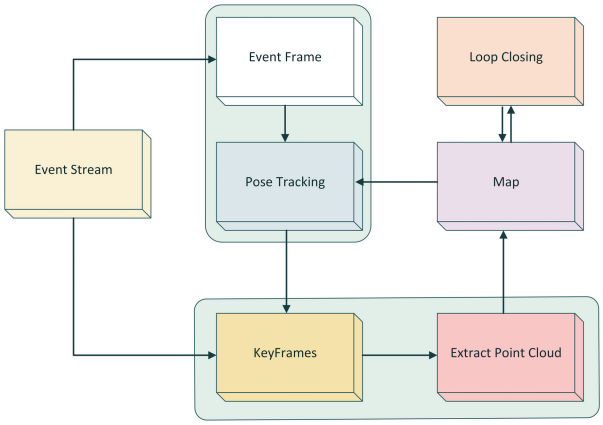



**FIGURE 7.** Event-based SLAM diagram, adapted from [42].

and low power consumption. However, event cameras mark a significant shift in how visual information is captured, necessitating the development of new techniques to process the gathered data and extract meaningful insights. One of the primary challenges is accommodating distinct space-time outputs. Event cameras produce asynchronous and spatially sparse events, in contrast to the synchronous and dense images from standard cameras. As a result, frame-based vision algorithms designed for image sequences cannot be directly applied to event data [39]. Event-based SLAM remains an active area of research, with various algorithms and methods proposed in recent years to improve the accuracy, robustness, and efficiency of event-based SLAM systems. Figure 7 illustrates the structure of a typical Event-based SLAM approach.

### E. MULTIMODAL SLAM

These SLAM methods combine two or more sensor modalities mentioned earlier to estimate camera trajectory and build a 3D map of the environment. These methods offer increased robustness and are commonly used in challenging settings where individual sensors may encounter limitations or provide insufficient information [43]. One instance of multimodal SLAM is the integration of one of the visual sensors discussed with a LiDAR, such as DVL-SLAM [44], which combines a monocular camera with a LiDAR. The system proposed in [45] showcases another multimodal SLAM approach by integrating various SLAM methods that utilize monocular vision, laser measurements, and/or inertial measurements. These multimodal approaches leverage the complementary strengths of different sensors, leading to more robust and accurate mapping results in challenging real-world scenarios. Figure 8 illustrates the structure of a typical multimodal SLAM approach.

### F. VISUAL-INERTIAL SLAM

The IMU sensor offers an effective solution for addressing tracking issues that may arise when the camera operates in challenging environments with minimal texture or occlusions. Through the fusion of visual and inertial sensors, Visual-Inertial SLAM can instantly estimate the camera's 6-DoF pose. The combination of visual and inertial measurements can enhance the robustness and accuracy of the SLAM system, particularly in challenging environments where there is a shortage of visual features due to factors like illumination change, textureless areas, or motion blur [47]. One example of Visual-Inertial SLAM is the pioneering Ultimate SLAM [48], which is a hybrid method that combines events, standard frames, and IMU measurements to deliver a resilient state estimation in challenging situations. Figure 9 illustrates the structure of a typical Visual-Inertial SLAM approach.

## III. CATEGORIZATION OF V-SLAM

SLAM algorithms have conventionally been employed to construct maps of unfamiliar environments for robots, while simultaneously determining the robot's location within the space. The process of V-SLAM can be broadly divided into two components: the front-end and the back-end. In the front-end, the visual sensor plays a crucial role in gathering data while the robot is in motion. This data is then transmitted to the visual odometer, which estimates the information from adjacent images or points, forming a local map and determining the robot's position. On the other hand, the back-end is responsible for optimizing the data collected by the front-end and generating a comprehensive map. Loop detection is an essential aspect of the back-end, as it helps determine whether the robot's previous and current positions overlap by comparing the gathered information. This step is crucial in preventing drift and ensuring accurate mapping [1].

V-SLAM approaches were initially proposed to address the navigation problem by relying solely on static features present in the surrounding environment. However, these approaches do not take into account dynamic objects in the scene. They can be broadly categorized into two main types: dense or direct methods and feature-based methods. Numerous surveys have been conducted to summarize and compare these traditional approaches, highlighting their respective strengths and limitations [2], [26]. However, these algorithms tend to focus mainly on key details, such as wall positions and orientations, while neglecting other useful information about the environment, such as furniture, door locations, and other distinctive features that could aid in accurate localization. To address this limitation and enhance mapping capabilities, the integration of deep learning applications in computer vision has advanced the field of V-SLAM, leading to the development of more robust mapping systems [51].

In this paper, V-SLAM systems are categorized into three types based on how they use information from images: (a) direct or dense methods, (b) feature-based methods, and (c) semantic scene understanding methods. Direct methods, also known as dense methods, estimate camera motion and reconstruct the environment by directly utilizing the intensity values or pixel information from camera images. Conversely, feature-based methods rely on extracting and

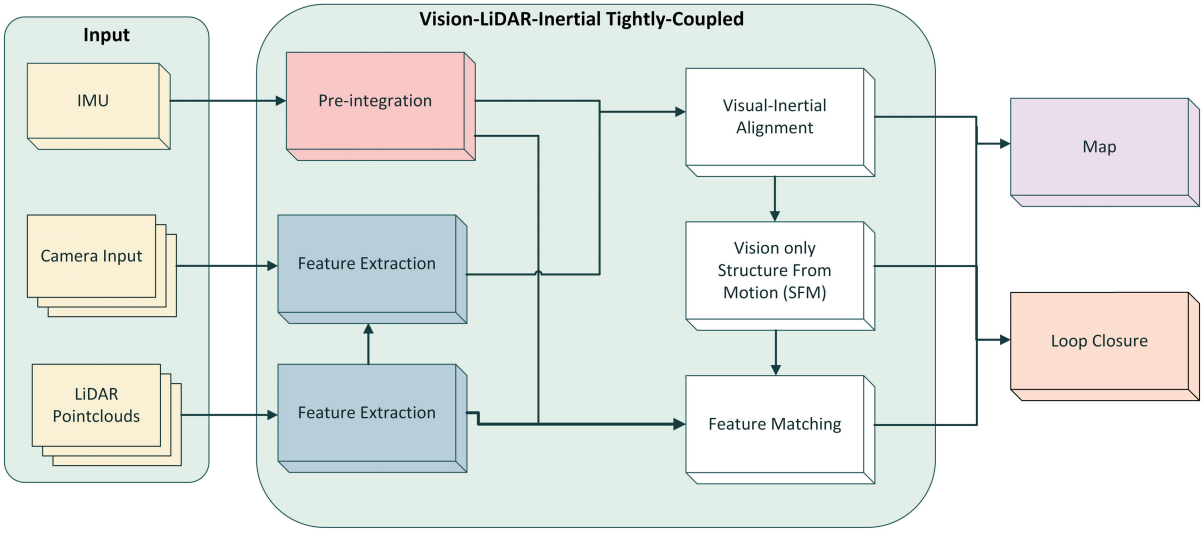



**FIGURE 8.** Multimodal SLAM diagram, adapted from [46].

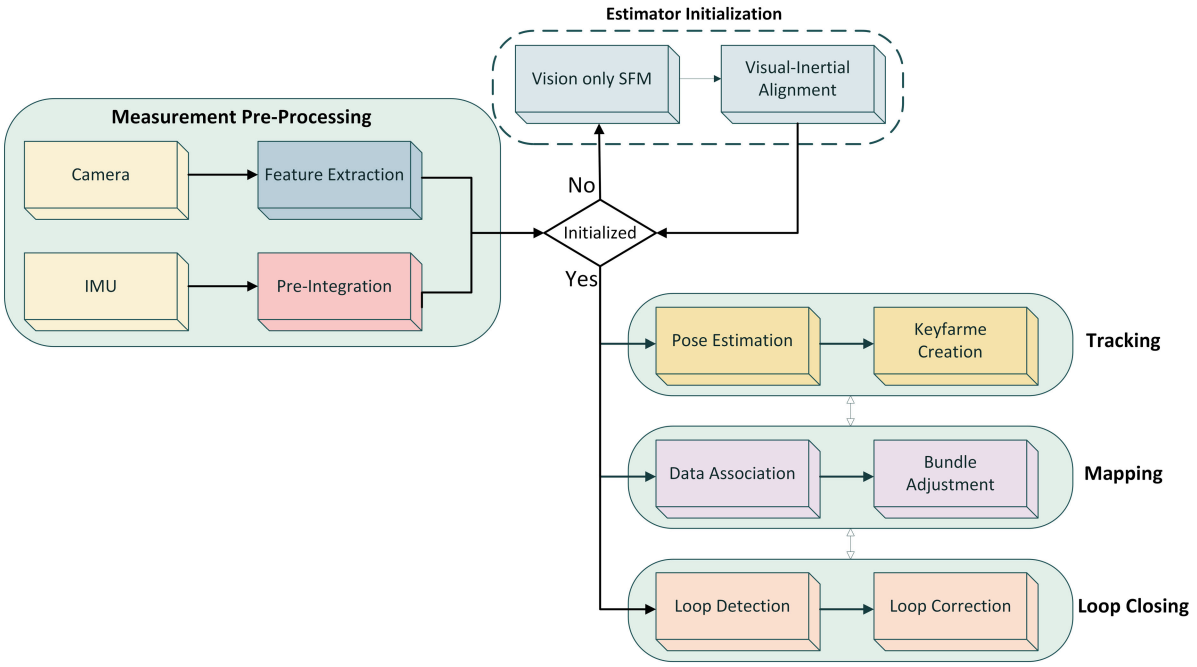



**FIGURE 9.** Visual-Inertial SLAM diagram, adapted from [49] and [50].

matching distinctive visual features from images to estimate camera motion and reconstruct the environment [52]. On the other hand, Semantic SLAM takes a different approach by incorporating machine learning (ML) techniques to utilize visual information for building a geometric map and estimating camera pose. This method incorporates a semantic understanding of the environment, allowing for the recognition and utilization of meaningful objects and structures in the scene [53].

This section provides an overview of the main and most recent state-of-the-art algorithms in V-SLAM, focusing on direct or dense methods and feature-based methods. Moreover, it provides a brief overview of modern SLAM methods that incorporate semantic information, providing better scene understanding of the environment. A more detailed discussion on these semantic methods can be found in Section IV. As the field of V-SLAM is constantly evolving, it is challenging to determine definitive top-ranked methods. However, several popular and highly regarded methods have emerged over the years. In the dense-based category, Dense Visual Odometry (DVO) techniques [18], [54] have garnered attention. In the feature-based category, ORB-SLAM and its variants [12], [34], [47] have been prominent contenders. Each of the SLAM categories has its

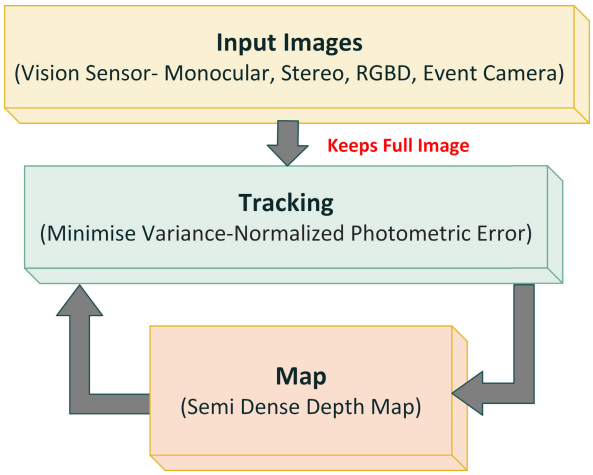



**FIGURE 10.** The general workflow of Direct/Dense SLAM, with the possibility of modifications or omissions in some of these modules.

own advantages, depending on the required level of information and scene understanding. Table 2 provides insights into the unique strengths and limitations of the different V-SLAM methodologies, assisting researchers and practitioners in choosing the most suitable approach for their specific application requirements.

### A. DIRECT/DENSE SLAM

Direct SLAM, also known as dense SLAM, is an approach that directly operates on the pixel intensities or color values of images to estimate camera motion and reconstruct the environment. Instead of relying on feature detection and tracking, direct SLAM works with the raw image data. Direct SLAM can be classified into two categories: dense and semi-dense methods. Dense methods make use of information from every single pixel in the image, taking advantage of the available data throughout the entire image. On the other hand, semi-dense methods focus on pixels where the gradient of image brightness is significant, utilizing information from these specific pixels to estimate camera motion and reconstruct the environment [55]. Figure 10 depicts the architecture of the Direct SLAM pipeline.

Dense Visual Odometry (DVO) is a state-of-the-art method in RGB-D SLAM for environments with minimal movement and serves as the foundation for upcoming dense RGB-D SLAM in high dynamic environments [18], [54]. DVO aligns consecutive RGB-D images to compute camera motion, minimizing photometric and geometry errors on both RGB and depth images. To address local drift, the authors in [18] and [54] shifted from conventional frame-to-frame alignment to frame-to-keyframe alignment. They conducted an ablation study to analyze the effects of frame-to-keyframe and graph optimization on trajectory accuracy. However, DVO relies on the assumption of photo-consistency, assuming a noiseless, non-moving scene with constant illumination. To overcome this constraint, a robust version of DVO was proposed in [54], featuring a more robust error function to handle noise and outliers in the scene.

A novel category of DVO methods has recently emerged, leveraging edge alignment as a key component. These methods show promise as an alternative to other direct approaches since they utilize sparse representations, offer a larger convergence basin, and exhibit stability under changes in illumination. In [56], a new edge-based V-SLAM system called real-time robust edge-based SLAM (RESLAM) was introduced, specifically designed for RGB-D sensors. A comprehensive V-SLAM method was built, incorporating edge utilization across all stages, including camera pose estimation, sliding window optimization, loop closure, and relocalization. RESLAM refines initial depth information from the sensor, camera poses, and camera intrinsics within a sliding window to enhance accuracy. Moreover, a novel edge-based verification technique is introduced for loop closures, which can also be utilized for relocalization. The authors demonstrated that RESLAM performs comparably to several cutting-edge methods, such as ORB-SLAM2 [12] and DVO-SLAM [18], while operating in real-time on a CPU only, making it suitable for mobile robotics and navigation tasks.

Engel et al. proposed Direct Sparse Odometry (DSO), a visual odometry method that combines a novel sparse and direct structure from motion formulation for accurate tracking of camera motion and 3D reconstruction [57]. Unlike traditional methods that rely on keypoint detectors, DSO samples pixels evenly across all image regions, including edges and featureless walls, improving accuracy and robustness. The proposed model integrates a full photometric calibration, accounting for various factors such as exposure time, lens vignetting, and non-linear response functions. DSO was evaluated on multiple datasets, revealing its superior performance compared to state-of-the-art direct and indirect methods in terms of both tracking accuracy and robustness. However, without incorporating loop closing techniques, DSO is prone to accumulated drift in unobservable degrees-of-freedom, leading to inaccuracies in the long-term camera trajectory and map. LDSO (Loop Detection and Pose-Graph Optimization) [58] enhances DSO by introducing loop closing capabilities, improving repeatability of selected points from DSO and enabling reliable detection of potential loop closures using a bag-of-words (BoW) technique.

Bryner et al. introduced a method for tracking the 6-DOF pose of an event camera with respect to a photometric 3D map in a known environment [41]. This approach utilizes raw events directly, without intermediate features, and employs a maximum-likelihood framework for joint estimation of camera poses and velocities through nonlinear optimization. While it demonstrated accuracy with noise-free event data in controlled scenarios, real-world conditions introduced errors due to noise, camera imperfections, sensor delays, and calibration inaccuracies. Nevertheless, pose tracking remains effective in real-world conditions. The authors have made datasets with ground truth poses available to enhance reproducibility and research in event-SLAM field.

**TABLE 2. A comparative analysis of various visual-SLAM methodologies.**

| V-SLAM method | Advantages | Disadvantages |
|---|---|---|
| Direct SLAM | • Provides dense reconstruction, beneficial in poorly textured scenes.<br>• Offers a simpler implementation compared to other SLAM methods.<br>• Yields a detailed representation of the scene, unlike sparse techniques. | • Demands high computational resources due to the pixel-intensive optimization.<br>• Faces challenges in varying lighting conditions or low-light environments.<br>• Exhibits reduced robustness in dynamic scenes. |
| Feature-based SLAM | • Integrates seamlessly with existing feature-based vision techniques.<br>• Requires lower computational resources when compared to direct methods.<br>• Demonstrates robust performance in dynamic scenes. | • Yields a sparse reconstruction of the environment, limiting scene detail.<br>• Shows limited robustness in the presence of illumination changes. |
| Semantic SLAM | • Offers advanced environmental understanding.<br>• Predicts future events based on the current scene.<br>• Recognizes object relationships and functionality within the environment. | • Associated with high computational costs due to complex data processing.<br>• Requires substantial volumes of high-quality data for training ML models.<br>• Applicable in scenarios where ML models can reliably classify high-level data. |

Despite showing potential in specific applications, dense V-SLAM methods have limitations. They require significant computational resources in terms of time and memory, making them impractical for real-time applications or large environments due to increasing memory requirements. Additionally, they may struggle in changing lighting conditions or low-light environments as they rely on aligning consecutive frames. Most importantly, they exhibit weak robustness to dynamic scenes with moving objects, which may limit their applicability in such scenarios.

### B. FEATURE-BASED SLAM

Feature-based methods in V-SLAM focus on specific areas in the image with unique characteristics known as features. These features can vary in scale, ranging from low-level points, corners, and lines to middle-level blobs and planes, and even high-level objects with semantic labels. The key aspect of a feature is its repeatability, meaning it can be reliably detected across multiple frames captured from different viewpoints. V-SLAM systems may utilize a single level of features or a combination of multiple feature levels, creating a hybrid approach [59]. Figure 11 illustrates the architecture of the Feature-based SLAM pipeline.

One of the fundamental frameworks for feature-based V-SLAM methods is the ORB-SLAM, initially proposed in [34]. ORB-SLAM uses visual features to estimate the robot's position and create a 3D map of the environment. It employs oriented FAST corners [60] and Rotated BRIEF descriptors [61] for feature detection and description. The algorithm consists of three main components: tracking, mapping, and loop closure detection [12], [34], [47]. The tracking component estimates the camera's real-time position and orientation using visual features from the camera image. The mapping component creates a 3D map by combining camera motion estimates with detected visual features. Loop closure detection identifies previously visited places by detecting similar visual features in different parts of the map and optimizing the map to improve accuracy.

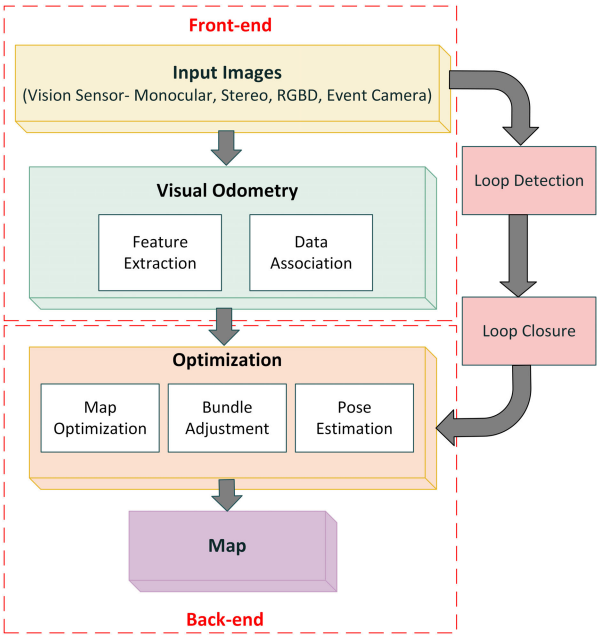



**FIGURE 11. The general workflow of Feature-based SLAM, with the possibility of modifications or omissions in some of these modules.**

The original version of ORB-SLAM, proposed in [34], utilizes a monocular camera for visual input and exhibits limited accuracy and robustness compared to stereo and RGB-D inputs. However, it is still capable of handling large-scale environments with loop-closures in real-time. Several variants of ORB-SLAM have been proposed, such as ORB-SLAM2 [12], which improved the original version by handling stereo and RGB-D inputs, a more robust loop-closure detection algorithm, and an improved map management system. Moreover, ORB-SLAM2 integrates a relocalization module that allows the algorithm to recover from tracking failures by determining its position within the map. More recently, Campos et al. proposed ORB-SLAM3, which incorporates semantic information into the

map and includes a relocalization module that enables the algorithm to recover from failures by utilizing both geometric and semantic information [47]. This new framework proves beneficial for improving V-SLAM in environments with moving objects.

EAO-SLAM [62] is a monocular object SLAM system built on ORB-SLAM2, addressing data association and pose estimation issues. The framework incorporates a semantic thread that adopts YOLOv3 for object detection. To handle data association, the authors propose an ensemble method that combines parametric and nonparametric statistical tests. This approach is integrated in the tracking thread, which merges information from bounding boxes, semantic labels, and point clouds. For object pose estimation, EAO-SLAM proposes a centroid and scale estimation procedure, along with an object pose initialization approach based on the isolation Forest (iForest) algorithm, which improves the accuracy of estimation by eliminating outliers. The joint optimization process is then used to optimize both object pose and scale, along with the camera pose, resulting in a lightweight and object-oriented map. The system generates object-oriented semantic maps using the data association and object pose estimation algorithms, along with a semi-dense mapping system. However, it is worth noting that some inaccurate estimations may arise when observing large objects that cannot be adequately captured by a fast-moving camera, as exemplified by the chair in the fire sequence within the Microsoft RGB-D dataset.

Another approach by Wei et al. uses a Dirichlet Process Mixture Model (DPMM) for data association of cuboid landmarks in monocular SLAM [63]. The method begins with object detection in the current frame and subsequently tracks the objects using keypoint matching. The DPMM is then used to cluster keypoints to the same object, and a graph-based optimization is performed to associate the detected objects with the map features. To implement this approach, the YOLOv2 object detector [64] is used, and a method similar to CubeSLAM [65] is adopted to determine the object's position, orientation, and scale. Moreover, cuboids are added as a new type of vertex in the pose graph for SLAM optimization. This inclusion helps in reducing scale drift and enhances loop closing performance. However, the method has two limitations: it assumes that most objects in the environment remain static for a few local frames, and it relies on the assumption of small noise covariance. These constraints may affect the accuracy and robustness of the approach in scenarios with dynamic or noisy environments.

In summary, feature-based SLAM algorithms provide a robust and efficient solution for V-SLAM, capable of handling various camera setups and large-scale environments, overcoming the computational challenges of dense methods. However, traditional SLAM approaches still suffer from a significant limitation: their effectiveness diminishes considerably when dynamic objects, such as moving individuals, vehicles, or other robots, are added to the environment. To tackle this challenge, modern SLAM methods, as discussed in the following section, offer more resilient solutions to address the complexities of such environments.

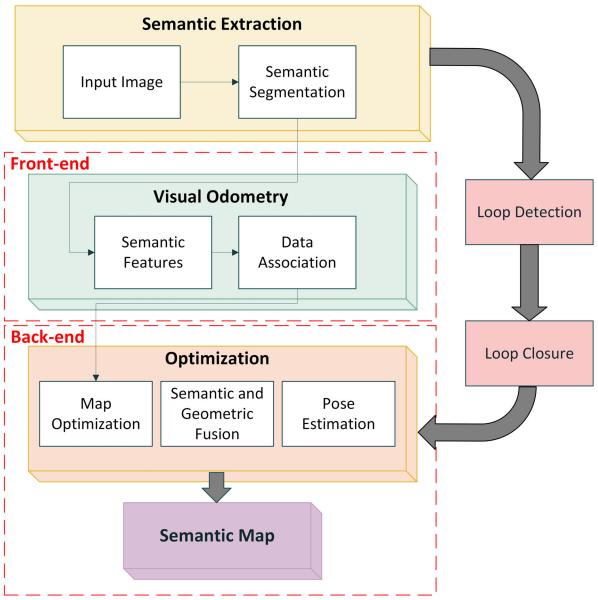



**FIGURE 12.** The general workflow of Semantic SLAM, with the possibility of modifications or omissions in some of these modules.

### C. SEMANTIC SLAM

Semantic SLAM represents a significant advancement over conventional SLAM as it integrates valuable additional information about the environment, such as the location of objects like tables and chairs, into the mapping process. This results in a more precise and informative map. Furthermore, Semantic SLAM utilizes machine learning techniques to analyze visual data and infer high-level details about the environment, leading to a more comprehensive understanding of the scene. This includes recognizing objects and their relationships, understanding object functionality, and even predicting forthcoming events based on the current scene. For instance, a scene understanding SLAM algorithm might employ object recognition and object function awareness to deduce that a chair is likely to be located close to a table. Information from images can be used in Semantic SLAM through direct methods, as seen in [66], or through feature-based approaches, such as those implemented in [27] and [67]. Figure 12 illustrates the architecture of the Semantic SLAM pipeline.

The accuracy of the semantic segmentation method used can significantly impact the performance of SLAM systems. If the semantic segmentation algorithm fails to accurately identify and distinguish between objects, it can result in errors in mapping and localization. Recently, there has been a recent shift in the focus of SLAM research towards better handling dynamic environments, reflecting real-world scenarios [68]. V-SLAM approaches, such as those discussed in [1], are better suited for environments with people and other objects

that could otherwise hinder the performance of the SLAM algorithm.

In summary, Semantic SLAM differs from traditional SLAM approaches in that it incorporates the semantic information of the environment and leverages machine learning techniques to infer high-level information about the scene, rather than solely relying on geometric features to create a map. By combining traditional SLAM techniques with semantic understanding of the environment, Semantic SLAM systems go beyond perceiving geometric information to also interpreting the semantics of the scene.

## IV. MODERN V-SLAM METHODS

To enhance the accuracy and robustness of V-SLAM in dynamically changing environments, it is crucial to mitigate the impact of moving objects, which is a key differentiator between traditional and modern V-SLAM implementations. This requires the identification and handling of dynamic characteristics in the surrounding scene. Over the years, various techniques have been proposed to address this challenge, such as using fixed features, integrating adaptable stationary characteristics, addressing occluded backgrounds, and focusing on features at the edges of moving objects. However, several challenges remain, including dealing with large moving objects that occupy significant portions of the frame, handling stationary but mobile objects like chairs, windows, and books, and ensuring real-time efficiency in SLAM approaches. One such method proposed in [69] is DMS-SLAM, a general V-SLAM system designed for dynamic environments. It works with monocular, stereo, and RGB-D cameras and integrates SLAM with Grid-based Motion Statistics (GMS) to handle dynamic scenes.

Modern V-SLAM methods can be broadly categorized into two main groups based on how they handle dynamic elements: dynamic-aware and dynamic-inclusive methods. These groups have distinct approaches for designing modern V-SLAM systems, each with its own contributions and potential areas for improvement. Table 3 provides an overview of the strengths and limitations of different modern V-SLAM categories.

### A. DYNAMIC-AWARE METHODS

These methods take into account the presence of moving objects in the environment, but they do not directly incorporate them into the SLAM process. Instead, their focus is on mitigating the influence of dynamic features on SLAM output by treating them as outliers or noise. The main goal is to detect and remove dynamic objects from the visual data before using it for camera pose estimation and mapping. The techniques within this group employ various actions to remove moving objects from the scene either by identifying them (e.g. background subtraction, foreground detection, or segmentation) [20], [70], [71], [72], [73], [74], [75], [76], [77], [78], [79], [80], [81], [82], [83] or by analyzing motion patterns (e.g. optical flow, a consistency check, or multi-view geometry) [84], [85], [86], [87]. The typical workflow of Dynamic-Aware methods is shown in Figure 13.

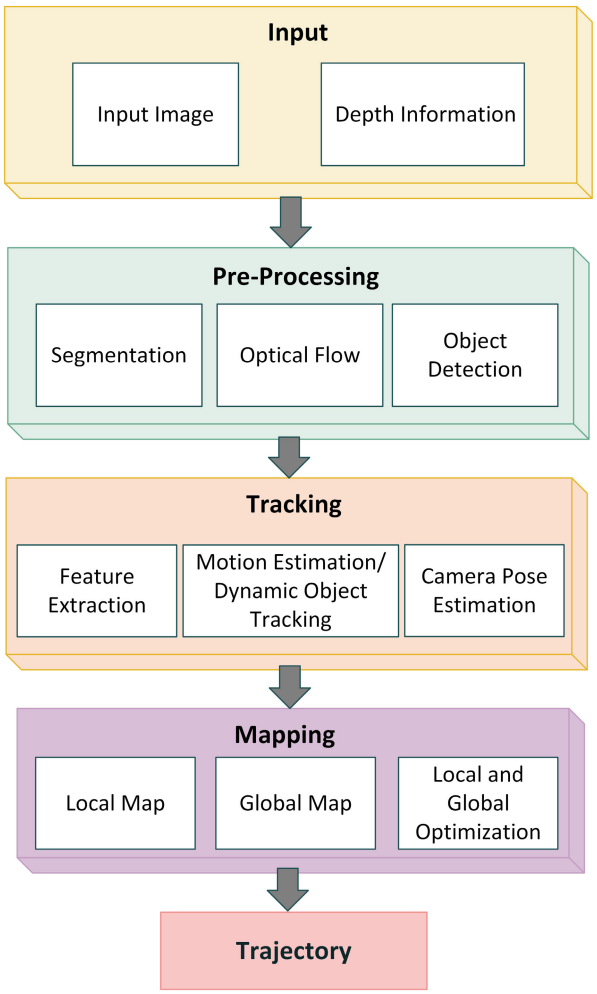



**FIGURE 13.** Dynamic-aware methods framework, with the potential for adjustments or exclusions in certain modules.

In this section, we will explore different methods based on how they detect and handle moving objects in the environment. For a summarized comparison of the dynamic-aware methods discussed in this section, refer to Table 4.

#### 1) BACKGROUND SUBTRACTION

Background subtraction methods attempt to distinguish the foreground (active scene) from the static background in the image. This can be done through various approaches, such as comparing object motion between frames at either the pixel-level or cluster-level.

A robust background model-based dense visual odometry (BaMVO) was one of the early V-SLAM techniques designed to handle dynamic scenes [71]. It uses a non-parametric model inspired by background subtraction to estimate the background in RGB-D data. Camera motion is then determined by applying a standard DVO on consecutive frames with non-moving backgrounds [54], [88]. However,

**TABLE 3.** Overview of strengths and limitations in different modern V-SLAM categories.

| Category | Advantages | Disadvantages |
|---|---|---|
| Dynamic-aware | • Can be applied atop any state-of-the-art SLAM methods after detecting and removing dynamic elements (as a pre-processing step). | • Has a limited ability to handle highly-dynamic environments.<br>• Lacks scalability for many applications (since dynamic objects are removed). |
| Dynamic-inclusive | • Adaptable to a wide variety of robotic applications (e.g. object grasping). | • Demands high computational requirements.<br>• Requires sophisticated algorithms for incorporating dynamics. |

this method has limitations. First, it requires initializing the background model assuming the camera will not move for the next N frames, which needs careful calibration. Second, the non-parametric model struggles to handle many dynamic objects, making it unsuitable for real-world scenarios with fast motion, periodic dynamic features, and noise.

Sun et al. proposed a motion removal technique as a pre-processing step for advanced RGB-D SLAM [72]. This approach involves detecting the motion of mobile objects through image differencing, tracking them using a particle filter, and determining the foreground by applying a maximum a posteriori (MAP) estimator on depths. However, this method can incorrectly classify moving objects as static and vice versa, leading to lower accuracy in low-dynamic scenarios and issues with slow camera ego-motions.

Jaimez et al. introduced VO-SF, a novel approach that estimates motion at the cluster-level rather than the pixel-level [73]. The method involves first using a visual odometry module to estimate motion and then clustering the RGB-D image into $k$ clusters using K-means. The static background is segmented from the odometry output, and moving clusters are used to estimate potential static points for background refinement. However, clustering methods like K-means can lead to inaccurate scene flow estimates, mixing static and dynamic points in the same cluster, thus confusing the distinction between points that are not moving (static) and points that are moving (dynamic). This limitation comes from the fact that K-means assumes certain characteristics about the clusters it forms, such as equal sizes and spherical shapes. In a scene with a mixture of static and dynamic points, the clusters representing these points might not adhere to these assumptions.

Building upon VO-SF, Scona et al. introduced StaticFusion (SF), which estimates motion and scene segmentation on a cluster-level using an objective function with two energy terms [74]. The first term considers photometric and geometric consistency for pixels within static clusters, while the second term acts as a regularization term to enhance dynamic point detection. However, SF's performance deteriorates when the camera slows down, as it may classify moving objects that appear static relative to the camera as part of the static environment.

While background estimation and subtraction show promising results, they have several shortcomings, particularly in highly dynamic environments. Therefore, methods based on background subtraction are prone to lighting variations, dynamic backgrounds, shadows, reflections, slow camera speed, and may struggle to detect small or slow-moving objects, introducing extra challenges to the problem.

### 2) CONSISTENCY CHECK

Consistency check methods are designed to detect and remove dynamic objects from the scene as they can negatively impact the performance of V-SLAM systems, specifically in camera pose estimation and map building components. These methods use techniques like semantic segmentation and motion estimation to filter out moving objects [79].

Li et al. proposed SWIAICP-SLAM, a VO approach that relies solely on edge-depth points for odometry estimation [75]. Their method involves dividing depth-edge points into two groups: foreground (stable) and occluded (sensitive to camera motion) edge points. To determine the likelihood of each keyframe point belonging to the static environment, they employ a static weighting method. This likelihood is then integrated into the Intensity Assisted Iterative Closest Point (IAICP) method, used for point cloud registration [89].

Zhong et al. presented Detect-SLAM in their work [76], which introduces a novel approach following the pipeline of ORB-SLAM2 [12]. Instead of using the Single Shot Multi-box Object Detector (SSD) [90] on every frame, they devised a moving probability method based on feature-matching to detect dynamic objects efficiently. The method achieves a balance between speed and accuracy. To further enhance the results, the authors employed a local map. Subsequently, the Grab-Cut algorithm [91] was applied to separate the background and obtain the segment mask of the target object, thereby facilitating the reconstruction of an instance-level semantic map.

Zhang et al. extended the ORB-SLAM2 framework in their work [77] by adding the YOLO object detection module [64] to extract semantic information at the object level. They further refined the probabilities of the detected objects using conditional random fields (CRF) as an object

regularizer [92]. In addition, an octo-map was generated from the point clouds of the objects, and correspondences between existing and temporary objects were identified using a KdTree. To accelerate the mapping process, a fast line rasterization algorithm was used [93]. Despite these advancements, the main limitation of this approach lies in the object detector, which tends to detect numerous unnecessary features, including excessive background information within the bounding box. Moreover, the process of determining similarities between existing and temporary objects is prone to being influenced by the localization component.

Schorghuber et al. proposed SLAMANTIC in their work [78] as a solution to address the issue of dynamic features. This approach incorporates a dynamic factor term that considers the semantic labeling of a 3D point and its coherence, determined by Mask R-CNNN [94]. SLAMANTIC divides 3D points into three categories: static, static-dynamic, and dynamic, making it suitable for various types of data, including monocular, stereo, and RGB-D. While inspired by ORB-SLAM [34], this method does not perform well when objects that are assumed to be static are in motion, such as a building painted on a car.

In [79], the authors propose the DS-SLAM system, which extends the ORB-SLAM2 framework and operates through five parallel threads: tracking, semantic segmentation, local mapping, loop closing, and dense semantic map construction using an octo-tree. The system employs two types of features extraction: SegNet [95] and ORB extractor through the tracking thread, along with an additional moving consistency check based on epipolar constraints. ORB features detected within dynamic zones identified by SegNet are filtered out as outliers. The system then builds a semantic octo-tree map by discarding all dynamic objects. To account for the limitations of SegNet [95] in complex situations, a log-odds score is assigned to each voxel to filter out the unstable ones. However, two notable limitations are observed: the segmentation module is restricted to specific types of recognized objects, and the octo-tree map needs rebuilding when loop closure is detected, which may impact the speed of DS-SLAM.

In conclusion, consistency check methods show a promising solution by adding a straightforward consistency test that primarily relies on the likelihood of moving features. However, determining the appropriate consistency check based on the extracted features and semantics remains an unsolved challenge.

#### 3) MULTI-VIEW GEOMETRY

These methods focus on handling dynamic objects by analzing the relationships among multiple views of the scene. They leverage mathematical tools such as epipolar geometry, homography, and others to extract valuable information about camera motion, scene depth and features from different viewpoints of the same scene.

In [80], Bescos et al. proposed DynaSLAM, a cutting-edge V-SLAM technique designed for dynamic scenes. Compatible with monocular, stereo, and RGB-D data within the ORB-SLAM2 framework [12], DynaSLAM first uses Mask R-CNN [94] for semantic segmentation of RGB channels to identify dynamic objects. For RGB-D data, a lightweight version of the ORB-SLAM2 tracker and multi-view geometry are used to handle static yet movable objects not detected by Mask R-CNN. A background inpainting (BI) method is introduced to fill the background with static information after removing dynamic objects. While effective, DynaSLAM's instance segmentation module results in computational overhead. To enhance robustness and real-time applicability, a multi-object tracking system was recently integrated into DynaSLAM [96].

In [97], Cui and Ma presented SOF-SLAM, a technique based on the ORB-SLAM2 framework, which takes a different approach to remove dynamic features. Instead of using ORB features, they incorporated a semantic optical flow module that combines information from optical flow, multi-view geometry constraints, and semantic segmentation using SegNet [95] via the fundamental matrix. However, their method has two limitations: it relies on a hard decision technique to distinguish dynamic from static ones, which may not always be accurate, and it uses information from only two consecutive frames, leading to instability in highly dynamic scenes. To address these issues, Cheng et al. extended the approach by introducing a weight average approach and combining optical flow and multi-geometry constraints to detect and remove dynamic points, resulting in DM-SLAM [98]. However, like other similar techniques, DM-SLAM still does not consider dynamic features in the mapping process, which can be problematic in highly dynamic scenes with limited static information.

Dynamic Deep Learning-SLAM (DDL-SLAM) [81] adopts a methodology similar to SOF-SLAM for eliminating moving objects by combining semantic segmentation using DUNET [99] and multi-view geometry. The approach involves removing all dynamic parts, reconstructing missing parts using a basic BI algorithm, and constructing an octo-tree map. However, there are areas for improvement, such as injecting semantic information into the reconstructed map and utilizing a more efficient and robust BI method. Additionally, DDL-SLAM's high computational requirements for the inpainting method make it unsuitable for real-time applications.

In [100], Wang et al. proposed a module for detecting moving objects, serving as a pre-processing step before applying state-of-the-art RGB-D SLAM techniques. The module extracts inliers and outliers by analyzing the fundamental matrix computed on RGB data. Depth map features are clustered using K-means, and inliers and outliers are mapped to these clusters to eliminate moving features. Subsequently, pose estimation and tracking threads are applied to the remaining static features. While effective, this approach requires a minimum number of static features in each frame for accurate identification of inliers and outliers and is susceptible to errors when dealing with fast-moving objects.

In [101], Wen et al. suggest a method to separate dynamic from static features using depth error, photometric error, and re-projection error. They combine a multi-view geometry approach with an instance segmentation module, performing tracking based on Lucas Kanade (LK) optical flow estimation [102]. However, the reconstructed semantic octo-tree map is built solely from static features and is not suitable for real-time applications. Additionally, the segmentation module is limited to only four object classes, indicating a lack of generalization. PSPNet-SLAM [53] uses a similar framework but replaces the Mask R-CNN module with PSPNet [103]. PSPNet-SLAM performs better in high dynamic environments but struggles in environments with fewer moving objects. Indeed, the PSPNet-SLAM's design might lead to over-segmentation or misinterpretation of static elements due to its focus on identifying and tracking moving objects. This can result in reduced accuracy and a less stable mapping process when there are limited dynamic elements to track.

OFM-SLAM (Optical Flow combining MASK-RCNN SLAM) [104] utilizes a comparable approach to previous methods [53], [101], leveraging optical flow with multi-geometry constraints and Mask R-CNN modules to identify dynamic features. It then reconstructs a semantic octo-tree map from the extracted semantic features. However, OFM-SLAM faces limitations in detecting slow movements or static, yet mobile features and suffers from high computational costs, making it not viable for real-time applications.

In [82], Long et al. built upon DynaSLAM [80] with three key enhancements. First, they incorporated object segmentation using PSPNet [103] to improve accuracy. Second, they introduced a lightweight homography matrix-based approach to compensate for tracking errors. Finally, they developed a decision-making technique inspired by ant colony algorithms (ACP) called the reverse ant colony search strategy to distinguish dynamic from static features [105]. However, the proposed decision module presents challenges and may limit the effectiveness of dynamic parts removal.

In [83], Hu et al. enhanced the ORB-SLAM3 framework [47] with two additional steps. First, they employed an upgraded version of DeepLabv3+ [106], which uses depthwise separable convolution in the Atrous Spatial Pyramid Pooling (ASPP) module. This enhanced version provides a more powerful and versatile model compared to the standard version [107], enabling pixel-level semantic segmentation to differentiate dynamic features. Additionally, they used an ant colony strategy to reduce the time required to parse and remove dynamic points. The framework can also easily integrate IMU data into the tracking threads, enabling accurate localization due to the flexibility of the ORB-SLAM3 framework.

In conclusion, multi-view geometry and epipolar constraints methods demonstrate their significance in distinguishing static and dynamic features. However, relying solely on these methods may not be sufficient, particularly when dealing with static yet movable objects. Combining these approaches with others, as demonstrated in [82] and [104], is essential to effectively address this issue.

#### 4) OPTICAL FLOW

Optical flow methods compare the motion of pixels between consecutive frames in a video, providing valuable information about the scene and camera motion that is used to improve accuracy and robustness of V-SLAM systems. However, this approach is sensitive to visual disturbances such as lighting issues, occlusions and also incurs a high computational cost due to the nature of the approach. In [84], Zhang et al. propose FlowFusion, a method that enhances a typical visual odometry (VO) estimator by incorporating optical flow obtained from PWC-Net [108] to handle dynamic objects. They introduce a technique called ''dynamic clustering segmentation'' based on color, depth, and optical flow by applying two loss functions, namely photometric loss over RGB color and geometric loss over depth. To optimize the optical flow estimation, they utilize a GPU-based PWC-Net implementation to manage its computational intensity. Their approach demonstrates robustness to both slow and fast motions.

The authors of [85] demonstrated the advantages of incorporating dynamic articulated objects into feature-based V-SLAM systems, based on two key observations: the consistent 3D structure of each rigid part of an articulated object over time and the coherent motion exhibited by points on the same rigid part. To address this, they introduced AirDOS, a dynamic object-aware system that incorporates rigidity and motion constraints to effectively model articulated objects. In the preprocessing and tracking stages of AirDOS, ORB features are extracted, and Mask R-CNN [94] is employed for instance segmentation to identify potential moving objects. For articulated objects like humans, Alpha-Pose [109] is utilized to extract human key points, and their 3D positions are determined by triangulating the corresponding key points from stereo images. Subsequently, the motion of moving humans is tracked using optical flow generated by PWC-Net [108]. Furthermore, AirDOS employs bundle adjustment on every frame to capture the full trajectory, ensuring its robustness in densely populated urban environments.

In [86], Chen et al. used the ORB-SLAM2 framework [12] and suggested a technique for identifying dynamic features by combining object detection and optical flow. They used a modified version of YOLOv4 [110], which incorporated an attention mechanism module inspired by [111], to identify moving objects. Subsequently, they removed the moving parts from the ORB features and applied the same procedure as proposed in ORB-SLAM2 for local mapping, loop detection, and bundle adjustment optimization. This approach is suitable for monocular cameras but has certain limitations, such as difficulty in handling boundaries and the removal of some static information due to the object detector.

Li et al. have introduced a robust stereo SLAM algorithm that incorporates dynamic region rejection [87]. The

algorithm detects dynamic feature points by analyzing the fundamental matrix, which is computed using feature pairs obtained by tracking the optical flow in consecutive frames. Then, the current frame is partitioned into superpixels labeled with disparity at their boundaries. From this, dynamic regions are obtained based on the dynamic feature points and superpixel boundaries types. The proposed SLAM algorithm excludes feature points within the dynamic region and only uses information from the static region to estimate the pose. This approach effectively mitigates the negative impact of moving objects on the algorithm, resulting in improved localization and mapping accuracy.

Optical flow has shown promising results in detecting motion. However, its sensitivity to fast motions and light conditions imposes additional constraints on the V-SLAM problem when relying solely on optical flow for motion detection. As a result, some methods, such as those presented in [53], [104], and [101], combine optical flow with multi-view geometry or epipolar constraints to mitigate these limitations.

In summary, advanced methods that are aware of environments with moving objects exhibit precise performance by effectively detecting and eliminating dynamic features throughout the trajectory. However, these methods often face challenges due to their intensive computational requirements, particularly when incorporating semantic features. Furthermore, they may discard a significant part of valuable scene information. Although dynamic features can influence the overall localization process, they retain valuable environmental information that holds potential benefits for various applications, including robotic grasping [112], [113].

### B. DYNAMIC-INCLUSIVE METHODS

Dynamic-inclusive methods take a slightly different approach compared to dynamic-aware methods by integrating information obtained from objects in motion into the SLAM process rather than explicitly excluding them. Instead of treating dynamic objects as outliers or noise, these methods enhance the SLAM process by incorporating the moving objects into the map and estimation processes. This may involve utilizing motion models to approximate the movement of such objects and leveraging this data to enhance the precision of the SLAM estimate. These methods belong to one of two main categories. The first category includes direct-based SLAM methods, which directly estimate the 3D structure of the environment from the input images. Specifically, these methods use a sparse depth map to estimate the camera pose and map the 3D environment. The second category involves fusion-based SLAM methods that combine depth measurements from multiple sensors to estimate the camera pose and 3D environment. Specifically, they fuse the depth measurements from an RGB-D camera and a LiDAR sensor to achieve robust localization and mapping. Some of these methods use volumetric fusion with a dense 3D representation of the environment to estimate the camera pose and map the 3D environment. Specifically, they use a dense signed distance function (SDF) representation to model the environment and a continuous-time fusion approach to update the map in real-time. The typical workflow of dynamic-inclusive methods is shown in Figure 14. Moreover, for a summarized comparison of the dynamic-inclusive methods discussed in this section, refer to Table 5.

#### 1) DIRECT-BASED

These appraoches are mostly with monocular camera where they employ a sparse depth map to deduce the camera's position and map the 3D surroundings. Yuan et al. proposed a method in [115] based on the framework of ORB-SLAM2 that categorizes features into three groups: static, dynamic, and static but movable. Dynamic and static objects are detected using Mask R-CNN [94], while nearby features (within 5-10 pixels) are checked using epipolar constraints between consecutive frames to identify static but movable features. A moving consistency test is then performed over the last n frames (with n = 5) to address pose estimation and random error issues. Mapping and bundle adjustment follow the same approach as ORB-SLAM2, but with modifications to meet SaD-SLAM requirements. However, the local map still incorporates points from moving objects and verifies their moving consistency. Although the method demonstrates robustness, the tuning of the number of frames and static features from movable objects must be carefully considered for accurate localization.

CubeSLAM [65] is an innovative method for monocular 3D object detection and SLAM. This approach combines semantic object detection and geometric SLAM into a unified framework, showcasing its significant benefits. The monocular 3D object detection utilizes a new method that generates high-quality cuboid proposals from 2D bounding boxes using vanishing points. The SLAM part is built on the ORB SLAM2 framework [12], with modifications to the bundle adjustment to include objects, points, and camera poses together. The system performs exceptionally well in scenarios with wide baseline matching, repetitive objects, and occlusions. It includes moving objects in dynamic environments in the tightly-coupled optimization process to improve camera pose estimation. For 2D object detection, the YOLO detector [64] is used for indoor scenarios, while MS-CNN [116] is used for outdoor scenarios. CubeSLAM was evaluated using diverse indoor and outdoor datasets, demonstrating exceptional accuracy in 3D object detection. It achieved a 3D recall rate of 90% with an Intersection over Union (IoU) score of 0.6, as proved by its performance on both the SUN RGBD dataset [117] and the KITII dataset. Nonetheless, one limitation of this approach is its assumption that the model for dynamic scenarios involves rigid objects adhering to physically feasible motion models. This assumption may not hold true for all real-world scenarios.

Recently, Gonzalez et al. [118] introduced a novel approach called TwistSLAM, which is based on the

**TABLE 4.** Comparative overview of dynamic-aware methods.

| Method | Sensors | Dense / Feature | Data Association | Map Representation | GPU | Real-time* | Code Available |
|---|---|---|---|---|---|---|---|
| BaMVO [71] | RGB-D | Dense | - | - | X | ✓ | BitBucket |
| MR-RGBD-SLAM [72] | RGB-D | Dense | Keyframe-to-Keyframe | Pose graph | X | X | Github |
| VO-SF [73] | RGB-D | Dense | - | - | X | X | Github |
| StaticFusion [74] | RGB-D | Dense | - | - | ✓ | X | Github |
| SWIAICP-SLAM [75] | RGB-D | Dense | IAICP | Pose graph | X | ✓ | Github |
| Detect-SLAM [76] | RGB-D | Feature | ICP with object ID | Semantic object map | ✓ | ✓ | Github |
| Zhang *et al*. [77] | RGB-D | Feature | KdTree | Octree | ✓ | ✓ | X |
| SLAMANTIC [78] | RGB-D Stereo Monocular | Feature | Keyframes | Semantic 3D map | ✓ | X | Github |
| DS-SLAM [79] | RGB-D | Feature | Bundle Adjustment | Octree | ✓ | ✓ | Github |
| DynaSLAM [80] | RGB-D Stereo Monocular | Feature | Bundle Adjustment | Local mapping | X | X | Github |
| SoF-SLAM [97] | RGB-D | Feature | Bundle Adjustment | Local mapping | X | ✓ | X |
| DM-SLAM [98] | RGB-D | Feature | Keyframes | Dense map | ✓ | X | X |
| DDL-SLAM [81] | RGB-D | Feature | Bundle Adjustment | Octree | ✓ | X | X |
| Wang *et al*. [100] | RGB-D | Feature | - | - | X | ✓ | X |
| Wen et. al [101] | RGB-D | Feature | Occupation probability | Semantic Octree + Local map | X | X | X |
| PSPNet-SLAM [53] | RGB-D | Feature | Occupation probability | Semantic Octree + Local map | ✓ | X | X |
| OFM-SLAM [104] | RGB-D | Feature | Occupation probability | Octree | ✓ | X | X |
| OCMulti-view PSPNet-SLAM [82] | RGB-D Stereo Monocular | Feature | Occupation probability | Semantic Octree + Local map | ✓ | X | X |
| DeepLabv3+SLAM [83] | RGB-D | Feature | Keyframes | Local mapping | ✓ | X | X |
| FlowFusion (FF) [84] | RGB-D | Feature | - | - | ✓ | X | X |
| AirDOS [85] | Stereo | Feature | Bundle Adjustment | - | X | X | Github |
| Chen *et al*. [86] | RGB-D | Dense | - | - | ✓ | X | X |
| Li *et al*. [87] | Stereo | Feature | Bundle Adjustment | Local mapping | X | ✓ | X |
| DORB-SLAM [114] | Monocular | Feature | Keyframes | - | X | X | X |

* If a method is indicated as lacking real-time functionality, this is either because the authors have noted it or because it has been evaluated through simulations.

ORB-SLAM2 framework. The method first creates a set of semantic clusters using panoptic segmentation [119], where each cluster contains the 3D points of one object (either static or dynamic). Next, the static clusters are used to estimate the camera's pose, while the dynamic clusters are used to track and update the poses of objects over the map. This is achieved by estimating their twists, which represent their linear and angular velocities. The

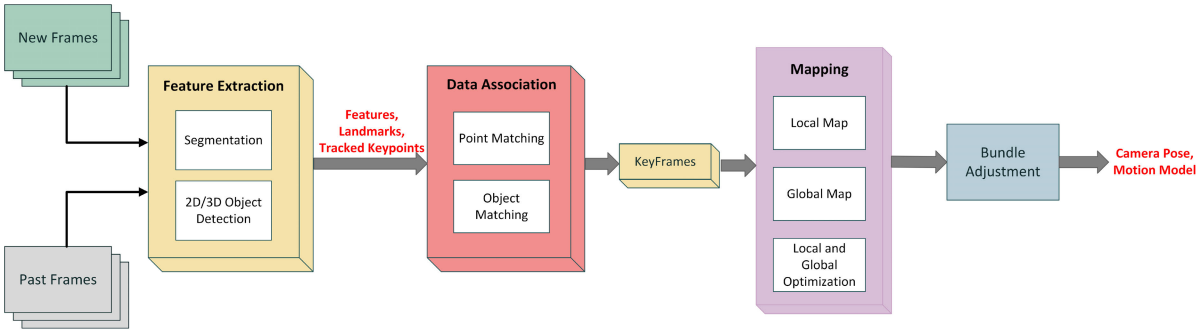



**FIGURE 14.** Dynamic-inclusive methods framework, with the potential for adjustments or exclusions in certain modules.

method considers that these dynamic objects are linked through mechanical joints, which impose constraints (inter-cluster constraints) and limit certain degrees of freedom. For example, a car cannot move vertically. This novel approach helps reduce the amount of noise in the pose estimation phase. To handle data association and keypoint estimation, an optical flow algorithm is used to overcome the problem of large displacements in the matching process [120]. By incorporating these techniques, TwistSLAM enhances the accuracy and robustness of object tracking and camera pose estimation within the ORB-SLAM2 framework.

### 2) FUSION-BASED

These methods combine depth measurements from multiple sensors to estimate the camera pose and 3D environment. Specifically, they fuse the depth measurements from an RGB-D camera and a LiDAR sensor to achieve robust localization and mapping. The Co-Fusion method, as described in [121], leverages the depth measurements provided by an RGB-D camera and a LiDAR sensor. It involves three steps applied to each frame after background extraction. First, tracking is performed by minimizing an objective function that combines two losses: an iterative closest point (ICP) alignment loss and a photometric loss. In the second step, segmentation is carried out, which is divided into two parts: motion segmentation based on Conditional Random Fields (CRFs), where labeling is done over Simple Linear Iterative Clustering (SLIC) superpixels, and an instance segmentation module. Finally, fusion is performed using the same tracking and fusion modules as in ElasticFusion [20]. Co-Fusion utilizes SharpMask [122], a refined version of DeepMask, for instance segmentation, making it suitable for real-time applications compared to Mask R-CNN. However, the method has a significant disadvantage, as it only incorporates static information into the reconstructed map. Therefore, it may fail in scenes with a high level of motion due to the lack of static features. Furthermore, experiments have shown that it struggles with fast motions and non-rigid dynamic objects, as motion detection strongly relies on the segmentation results.

PoseFusion, a method proposed in [123], is built similarly to Co-Fusion. However, it has a limitation that restricts its application to scenes where humans are the only dynamic objects present. The process begins by detecting human poses using OpenPose [124], which are then eliminated by applying Min-Cut segmentation techniques like the one proposed in [125]. By removing these dynamic parts, any state-of-the-art RGB-D SLAM technique, such as ElasticFusion [20], can be applied to the remaining static features. Nevertheless, it should be noted that the key drawback of this approach is its limited capability to handle only humans as dynamic objects, as it solely detects human poses as dynamic objects in the scene.

MaskFusion, proposed by Runz et al. in [126], is a real-time dense RGB-D SLAM method that incorporates object-level tracking and can maintain multiple object models. The system consists of three main modules: tracking, segmentation, and fusion. In the tracking module, geometric and photometric losses are minimized to track object poses. Geometric loss is calculated using ICP, while photometric loss is between the current frame and stored object models represented as sets of surfels. To distinguish between dynamic and static features, a moving consistency test is introduced, similar to StaticFusion [74]. The segmentation module utilizes a combined approach between instance segmentation by Mask R-CNN and a geometric-based segmentation. While Mask R-CNN provides object masks with imperfect boundaries, the geometric-based segmentation generates an edginess map based on depth discontinuity and concavity, providing good boundaries in real-time but suffers from over-segmentation issues. In the fusion module, surfels are associated to create a dense 3D map, similar to the fusion step in ElasticFusion [20]. However, MaskFusion has limitations for rigid objects and struggles to track small objects due to misclassification by Mask R-CNN. Some improvements to the system follow a similar methodology but represent the 3D map using volumetric signed distance fields (SDFs) with octo-trees, as seen in MidFusion [127] and Em-Fusion [128].

Brasch et al. proposed a monocular SLAM approach tailored for highly dynamic environments [129]. This

approach incorporates a probabilistic outlier model based on semantic prior information predicted by a Convolutional Neural Network (CNN). To achieve robustness in challenging conditions, the approach uses a combination of feature-based and direct approaches. The proposed SLAM system builds upon the ORB-SLAM framework [34]. For pose estimation, descriptive features are used whenever possible. In cases where an insufficient number of features can be found, direct features are used in addition. The estimation of probabilistic outlier rejection involves calculating an inlier ratio for each map point, which indicates the level of reliability and stability associated with that specific map point. Additionally, the inclusion of semantic information provides an independent source of information about the likelihood that the map points are dynamic. This integration of semantic priors enhances the system's ability to handle highly dynamic environments effectively.

Dynamic-inclusive methods are better suited for applications that require a comprehensive map of the environment compared to dynamic-aware methods, which exclude dynamic features and lose a significant amount of information. However, implementing dynamic-inclusive methods can be challenging due to potential computational complexity and the need for sophisticated algorithms to detect and track moving objects effectively.

Modern SLAM methods have become more complex compared to traditional methods due to the incorporation of additional processing layers. As a result, these advanced methods require higher computational resources. One way to address this challenge is by adopting a server-based or cluster-based approach, as proposed in [130], to distribute the processing load effectively. Nevertheless, the high cost and complexity of modern SLAM methods can limit their applicability in scenarios with limited computational resources. To overcome this, hybrid methods that combine elements of traditional and modern SLAM can be used. These methods can employ dynamic-inclusive techniques in areas with moving objects and switch to traditional methods in areas without any moving objects, optimizing computational efficiency. Additionally, existing algorithms can be optimized to reduce computational requirements. The choice of motion detection algorithm plays a crucial role in the accuracy and computational needs of V-SLAM methods. For instance, optical-flow-based methods [12], [84] are more accurate, while semantic-based methods [79], [80], [101] require higher computational resources but offer greater capabilities. Selecting the most suitable motion detection algorithm can significantly impact the overall performance of the SLAM system.

## V. ROBUSTNESS

In this section, we explore SLAM techniques that target enhancing accuracy by addressing input-related challenges, such as the errors introduced by sensor measurements, which can accumulate and affect the accuracy of the device's positioning over time. Specifically, we focus on methods that implement software-based optimizations to improve the overall performance of V-SLAM. These approaches include the utilization of deep learning algorithms and the integration of multiple sensors to bolster the robustness of V-SLAM in challenging environments. Furthermore, they address challenges, such as front-end ambiguities and reflective surface detection. Additionally, they encompass the implementation of edge computing, which mitigates the computational burden of V-SLAM on mobile devices. Table 6 summarizes the methods discussed in this section.

### A. APPLYING DEEP LEARNING TECHNIQUES

Integrating deep learning into V-SLAM processes has proven to be a powerful solution for enhancing V-SLAM robustness in real-world applications, where conditions like lighting, weather, and other factors can vary. Deep learning excels in extracting rich, high-level features from images, surpassing the limitations of traditional geometric features. This not only enhances the robustness of V-SLAM, especially in keyframe feature matching, but also addresses issues like frame estimation errors and pose solution failures [131]. One promising approach is demonstrated in [4], where the potential of deep learning in boosting SLAM resilience is showcased. This approach aims to reduce trajectory estimation errors by using deep neural networks fine-tuned using an Adaptive Moment Estimation (Adam) optimizer. By leveraging deep learning to reduce noise patterns, the method has been evaluated in simulation and real-world scenarios, using a Pioneer 3AT robot. The deep neural network is trained on a dataset of 6751 samples, including the robot's 2D position, orientation, and corresponding ground truth.

In [132], another innovative method combining monocular V-SLAM and deep learning based on ORB-SLAM2 is presented. This method incorporates a single-shot multibox detector to improve object detection performance in monocular SLAM. By implementing the selection tracking algorithm, dynamic objects in the scene are effectively eliminated, and a missed detection compensation algorithm is employed to improve the recall rate during object tracking. The primary focus of this method is to improve the detection of dynamic content, thereby enhancing the robustness and stability in terms of absolute trajectory error (ATE), while building upon the foundation of ORB-SLAM2 [67].

In [133], the authors propose DeepFusion, a novel 3D reconstruction system that addresses the limitations of sparse monocular SLAM and depth-based reconstruction methods. DeepFusion overcomes these challenges by generating dense depth maps in real-time from RGB images and scale-ambiguous poses obtained from a monocular SLAM system, specifically ORB-SLAM2 [12] in their implementation. DeepFusion employs a CNN to produce fully dense depth maps for keyframes with metric scale, representing the observed geometry. The system then combines the output of a semi-dense multi-view stereo algorithm with the depth

**TABLE 5.** Comparative overview of dynamic-inclusive methods.

| Method | Sensors | Dense / Feature | Data Assosciation | Map Representation | GPU | Real-time* | Code Available |
|---|---|---|---|---|---|---|---|
| SaD-SLAM [115] | RGB-D | Feature | Bundle Adjustment | Local mapping | X | X | X |
| CubeSLAM [65] | RGB-D Monocular | Feature | Bundle Adjustment | Local mapping | X | ✓ | Github |
| TwistSLAM [118] | Stereo | Feature | Bundle Adjustment | Local mapping | X | X | X |
| Co-Fusion [121] | RGB-D | Dense | ICP | 3D Map + 3D models | X | ✓ | Github |
| PoseFusion (PF) [123] | RGB-D | Feature | ICP | Point clouds | ✓ | ✓ | X |
| MaskFusion [126] | RGB-D | Dense | Surfels | Semantic 3D Map + 3D models | ✓ | ✓ | Github |
| MID-Fusion [127] | RGB-D | Dense | - | Volumetric | X | ✓ | X |
| EM-Fusion [128] | RGB-D | Dense | Probabilistic | Volumetric | ✓ | X | X |
| Brasch *et al.* [129] | Monocular | Feature | Bundle Adjustment | - | ✓ | ✓ | X |

* If a method is indicated as lacking real-time functionality, this is either because the authors have noted it or because it has been evaluated through simulations.

and gradient predictions from the CNN in a probabilistic manner. This fusion process is optimized with each new frame by incorporating new geometric constraints. To achieve real-time dense 3D reconstructions, DeepFusion formulates a cost function that combines per-pixel losses based on network depth predictions, sparse semi-dense depth estimates, and pairwise constraints from network depth gradient predictions. The system also estimates the shape of the observed scene and its absolute scale while predicting per-pixel mean and variance to obtain uncertainties for all network outputs. These uncertainties are then probabilistically fused with geometric constraints. Although DeepFusion has shown promising results, the authors suggest that further investigating into design choices, such as training data selection, and finding ways to handle extreme outliers produced by the network could lead to further improvements in the system.

### B. DEALING WITH FRONT-END AMBIGUITIES

Achieving robustness in V-SLAM systems pose challenges, particularly when the front-end and back-end modules interact. Many frameworks assume that the back-end optimizer receives accurate and unbiased information from the front-end, providing a single solution for each unknown variable. However, this approach becomes problematic when ambiguities arise. For example, when a feature point is detected as similar to multiple landmarks or when two loop closure candidates contradict each other, the front-end struggles to determine the correct information. Consequently, incorrect data can be incorporated into the back-end optimization, compromising the integrity of the V-SLAM system and potentially leading to the failure of the entire robotic system.

To address the challenges posed by front-end ambiguities, it is crucial for the back-end solver to explicitly consider and account for these unsolvable cases while producing multiple probable solutions. Hsiao et al. developed MH-iSAM2 [134], a novel online nonlinear incremental optimizer designed to enhance the robustness of robotic systems in the face of such ambiguities. MH-iSAM2 builds upon the incremental smoothing and mapping using Bayes tree (iSAM2) algorithm [135]. The key innovation lies in its incorporation of multi-mode measurements to model the ambiguities as inputs and generate multi-hypothesis outputs, thereby representing multiple possible solutions for the most likely results. The algorithm relies on two essential data structures: an extension of the original Bayes tree, facilitating efficient multi-hypothesis inference, and a Hypo-tree that explicitly tracks and associates the hypotheses of each variable while facilitating all the necessary inference processes for optimization. By adopting MH-iSAM2, robotic systems can effectively recognize and navigate through temporarily unsolvable ambiguities, significantly improving overall system robustness and ensuring reliable V-SLAM performance in challenging environments.

### C. DEALING WITH REFLECTIVE SURFACES

V-SLAM algorithms often encounter challenges when detecting reflective surfaces, such as mirrors and glasses, due to their unique optical properties, which can cause issues for traditional RGB sensors. These surfaces are highly reflective and can lead to distorted measurements or even the failure in object detection. To address this issue, Park et al. proposed a novel solution that uses 3D depth information to identify virtual images reflected in real-time within indoor

environments [136]. Their technique involves comparing the spatial information of detected objects with their surrounding environment to determine their geometric relationship. By using semantic segmentation and plane detection, they analyze the layout of the indoor space surrounding the object. The method effectively differentiates between real and reflected images of detected object candidates by leveraging 3D depth information. The authors evaluated the performance of the proposed algorithm using a large indoor dataset acquired from a Living Lab environment. Comparing the results of conventional detectors, such as Faster R-CNN [137] and RetinaNet [138], they observed a significant improvement in precision, with over 30% enhancement in the Living Lab dataset.

In [139], the authors presented a simple yet effective method for detecting glass panels by analyzing the specular reflection of laser beams from the glass surfaces. Their approach involves analyzing the intensity profile of the reflected light around the normal angle incident to the glass panel. They proposed that integrating this method with an existing SLAM algorithm could enable the resulting SLAM system to promptly detect and localize glass obstacles. To evaluate the efficacy of the proposed method, the authors conducted experiments in office buildings using a PR2 robot. The experimental results showed that the proposed method achieved an accuracy rate of approximately 95% for all glass panels with no false positive detections. Nevertheless, to capture specularly reflected light, this approach requires the robot to follow pathways that enable it to scan objects from their surface normals.

In [140], a technique that uses the fusion of polarization camera and laser sensor to detect glass obstacles across a broad range was introduced. The polarization camera has proven to be more effective in glass detection than laser range-finders (LRFs) at certain angles. LRFs can only detect glass at small incident angles, whereas the degree of polarization of reflected light on the glass surface is significant at larger incident angles. The system uses the straight line of the glass obtained from LRF measurements and the degree of polarization to determine whether an obstacle is glass or vacant space. Experimental results demonstrate the effectiveness of the method in successfully detecting glass obstacles across a wide range of scenarios. However, the cost associated with this method is relatively high when compared to other approaches.

### D. INTEGRATING MULTIPLE SENSORS

As mentioned in Section II, some SLAM techniques integrate IMU or LiDAR sensors alongside cameras to enhance system robustness in challenging scenarios. Relying solely on visual sensors may lead to failure or inadequate information [50]. DynaVINS [141] stands as an innovative visual-inertial SLAM framework, specifically designed to handle challenges arising from dynamic objects and temporarily static objects. Temporarily static objects are objects that appear stationary while within the field of view but can move when they are no longer observable. To ensure robustness, DynaVINS leverages bundle adjustment, which uses pose priors estimated from IMU preintegration to identify and reject features originating from dynamic objects. Moreover, the framework introduces a robust global optimization approach that organizes constraints into multiple hypotheses, effectively dealing with persistent loop closures caused by temporarily static objects.

In [142], Chou and Chou introduced an advanced integration approach named TVL-SLAM, which seamlessly combines visual and LiDAR data for simultaneous localization and mapping. In TVL-SLAM, the visual and LiDAR front-ends work independently, while the back-end optimization combines measurements from both modalities. The system builds upon ORB-SLAM2 [12] and a LiDAR SLAM method with average performance. To tackle challenges faced by individual visual or LiDAR methods, TVL-SLAM employs motion estimation and loop closing techniques that leverage information from both sources, resulting in higher accuracy and resilience in scenarios where either modality alone may fail. To improve reliability, the system uses cross-validation of visual and LiDAR motion estimation to identify and discard outlier features. Additionally, to enhance computational efficiency, the authors propose a general LiDAR factor (GLF) that compresses multiple LiDAR residuals into a concise 6-dimensional form. This optimization contributes to the overall efficiency of the TVL-SLAM approach. Despite the remarkable performance of TVL-SLAM, surpassing several existing visual/LiDAR SLAM approaches, some challenges, such as dealing with crowded highway scenes with limited shape features and numerous moving objects, remain unaddressed. Future research could focus on developing effective solutions to handle these specific challenges.

In [143], Schneider et al. presented an open framework called maplab designed for visual-inertial mapping. What sets maplab apart from other visual-inertial SLAM systems is its comprehensive approach. It not only facilitates the creation and localization of feature-based maps but also offers a suite of map maintenance and processing capabilities. The framework provides these capabilities to the research community in the form of a collection of tools, accessible through a user-friendly console. The toolset includes functionalities such as multi-session merging, sparsification, loop closing, and dense reconstruction of maps. This feature-rich workflow proves highly efficient for algorithm prototyping and parameter tuning. Additionally, maplab incorporates ROVIOLI (ROVIO with Localization Integration), an online mapping and localization frontend based on ROVIO [144]. ROVIOLI employs image intensity within patches, rather than relying solely on point features, making it robust even in the presence of motion blur. ROVIOLI can generate new maps from raw visual and inertial sensor data while enabling real-time tracking of a global drift-free pose when provided with a localization map.

### E. EDGE COMPUTING

The continuous operation of the modern V-SLAM techniques on mobile devices is often hindered due to its high computational power needs. Researchers have recently introduced the concept of offloading resource-intensive V-SLAM processing steps from mobile robots to edge computing as a promising solution to address this limitation [145]. In [146], Edge-SLAM is introduced as a solution to address the computationally demanding nature of V-SLAM. The authors propose a split architecture, distributing the computational load between a mobile device and an edge device. ORB-SLAM2 [12] is used as a prototype V-SLAM system, keeping the tracking computation on the mobile device and moving local mapping and loop closing to the edge. The results of this method indicate that this split architecture allows V-SLAM to function long-term with limited resources without compromising accuracy. It also maintains a constant computation and memory cost on the mobile device, enabling the deployment of other applications that rely on V-SLAM.

Cao et al. introduced edgeSLAM [147], an innovative solution to address the computational challenges of V-SLAM on mobile devices, offering both accuracy and real-time performance through efficient edge computing. To enhance the system accuracy, semantic segmentation is employed in edgeSLAM, while the high computational power needed for localization, mapping, and the semantic segmentation process is reduced through computation offloading. This method brings forth a range of notable innovations, including effective computation offloading, opportunistic data sharing, adaptive task scheduling, and the support for multiple users. The evaluation results illustrate that edgeSLAM can achieve real-time performance, with an average frame rate of 35 fps and a localization accuracy of 5 cm, surpassing many existing V-SLAM approaches. Additionally, two case studies on pedestrian localization and robot navigation are provided by the authors to highlight the practical usability of edgeSLAM.

Existing edge offloading methods mainly employ static offloading, which permanently transfers computation tasks from mobile robots to edge computing over wireless networks. Nonetheless, this approach can be challenging as wireless networks are inherently dynamic, and network quality may vary due to factors, such as fading or temporary obstructions, resulting in delays between the mobile device and the edge. To address this limitation, a novel network offloading approach called DynNetSLAM is proposed as a system that dynamically adjusts V-SLAM processing, based on changing network conditions [148]. DynNetSLAM introduces dynamic adaptation of V-SLAM computation offloading based on measured wireless network latency. It does this by setting an offloading latency threshold, a safe zone around this threshold, and a hysteresis mechanism to control the dynamic offloading. The evaluation results indicate that DynNetSLAM significantly reduces the probability of track loss events compared to ORB-SLAM2 [12], which processes V-SLAM statically on the mobile device. Moreover, with its dynamic offloading strategies, DynNetSLAM significantly reduces the adverse effects of network latency on Edge-SLAM [146], achieving a low track loss ratio while maintaining accuracy.

The importance of robustness in SLAM algorithms is often underestimated and requires further investigation. There is a need to improve the stability of SLAM systems, especially during extended operational periods, in order to address the issue of accumulated errors over time, as observed in [4]. Additionally, research efforts can be directed towards enhancing the quality of sensor data input prior to processing, as this can significantly boost system performance across various applications. Another crucial, yet often overlooked aspect is the filtering of environmental factors that can impact the performance of SLAM algorithms, such as lighting issues. Addressing these factors presents opportunities for optimization and holds the potential for significant advancements in the field of SLAM.

## VI. SCENE UNDERSTANDING

This section focuses on the issue of limited generalization in SLAM applications, as highlighted in [59]. The authors surveyed various V-SLAM methods that utilize monocular, RGB-D, and stereo cameras. However, each SLAM implementation has its inherent limitations. For example, monocular cameras suffer from lack of depth information and may require additional sensors to compensate for this deficiency. On the other hand, RGB-D cameras have a limited depth range, making them less suitable for outdoor environments. Although stereo cameras offer the best overall accuracy, they are also the most expensive and complex, and they can be sensitive to sudden changes in brightness. In light of these challenges, this section will explore semantic methods that utilize spatial relationships to establish connections between information in the scene, aiming to address the limitations posed by different camera configurations.

The integration of newly emerging methods, such as Optical Character Recognition (OCR) and object detection in V-SLAM system, is discussed in this section. These methods can provide valuable information for scene understanding and enhance the capabilities of V-SLAM systems. An overview of Scene Understanding SLAM methods is shown in Figure 15. Moreover, Table 7 provides a summary of the approaches outlined in this section.

### A. USING OCR

Optical Character Recognition (OCR) techniques are capable of extracting the semantic information from texts within optical frames. Balaban et al. proposed a method that incorporates semantic markers into a SLAM map by using natural text markers [149]. It interprets door placards to label office locations, utilizing YOLO for sign detection and EAST [150] for text recognition. Placards are accurately positioned by analyzing a point cloud within an RGB-D camera frame, localized with a modified ORB-SLAM2 [12]. Semantic mapping is executed as a subsequent step following

**TABLE 6.** Comparative overview of enhanced robustness approaches.

| Method | Sensors | Dense / Feature | Data Association | Map Representation | GPU | Real-time* | Code Available |
|---|---|---|---|---|---|---|---|
| Azzam *et al.* [4] | RGB-D | Feature | - | Local mapping | X | ✓ | X |
| Xiao *et al.* [132] | Monocular | Feature | Bundle Adjustment | Local mapping | ✓ | ✓ | X |
| DeepFusion [133] | Monocular | Feature | Keyframes | Dense depth map | ✓ | ✓ | X |
| MH-iSAM2 [134] | - | - | - | - | X | X | BitBucket |
| Park *et al.* [136] | RGB-D | Feature | - | - | ✓ | ✓ | X |
| Wang *et al.* [139] | LRF | Feature | Particle filter | Grid map | X | ✓ | Github |
| Yamaguchi *et al.* [140] | LRF Polarization camera | - | - | - | X | X | X |
| VINS-Mono [50] | Monocular IMU | Feature | Keyframes | - | X | ✓ | Github |
| DynaVINS [141] | Stereo Monocular IMU | Feature | Bundle Adjustment | 3D feature map | X | X | Github |
| TVL-SLAM [142] | Stereo Monocular LiDAR | Feature | Bundle Adjustment | Visual map + LiDAR voxel map | X | ✓ | X |
| Maplab [143] | Stereo IMU | Feature | Keyframes | Dense reconstruction | X | ✓ | Github |
| Edge-SLAM [146] | RGB-D Stereo Monocular | Feature | Bundle Adjustment | Local mapping | ✓ | ✓ | Github |
| edgeSLAM [147] | Monocular | Feature | Keyframes | Local mapping | ✓ | ✓ | Github |
| DynNetSLAM [148] | RGB-D Stereo Monocular | Feature | Keyframes | Local mapping | X | ✓ | X |

* If the method is indicated as not having real-time functionality, it is either because the authors have pointed it out or it has been evaluated through simulations.

robot exploration, resulting in a comprehensive mapping solution.

In [27], an innovative approach is presented to enhance localization in SLAM by leveraging text information within the local environment. This technique is particularly useful in man-made structures with repeated geometric patterns, such as identical floors, rooms, and corridors, where the kidnapped robot problem may arise. The proposed method emphasizes the role of text in the scene, including room numbers and signs in the environment. By combining text-level information with distinctive visual features, this approach improves localization accuracy.

### B. OBJECT SLAM

Object SLAM is a branch of semantic SLAM that prioritizes mapping with objects as the primary elements and often utilizes instance-level segmentation or object detection in its semantic network [151]. Martins et al. propose an approach that utilizes object-level information obtained from depth cameras (including RGB-D cameras or stereo camera rigs such as ZED 3D stereo cameras) to improve robot situational awareness [28]. This method allows the recognition of both dynamic and static objects in the environment. The approach involves collecting visual and depth information using RGB-D or stereo cameras and constructing a map with object-level information. Additionally, Kalman filters are implemented on sensor readings to improve accuracy. This approach enhances the robot's ability to perceive and understand its environment effectively, which is crucial for autonomous robotic applications.

In [66], Bao et al. proposed a semantic direct mono-SLAM algorithm to enhance localization performance in urban environments. The algorithm incorporates a point group movement consistency (PGMC) check and a point reselection

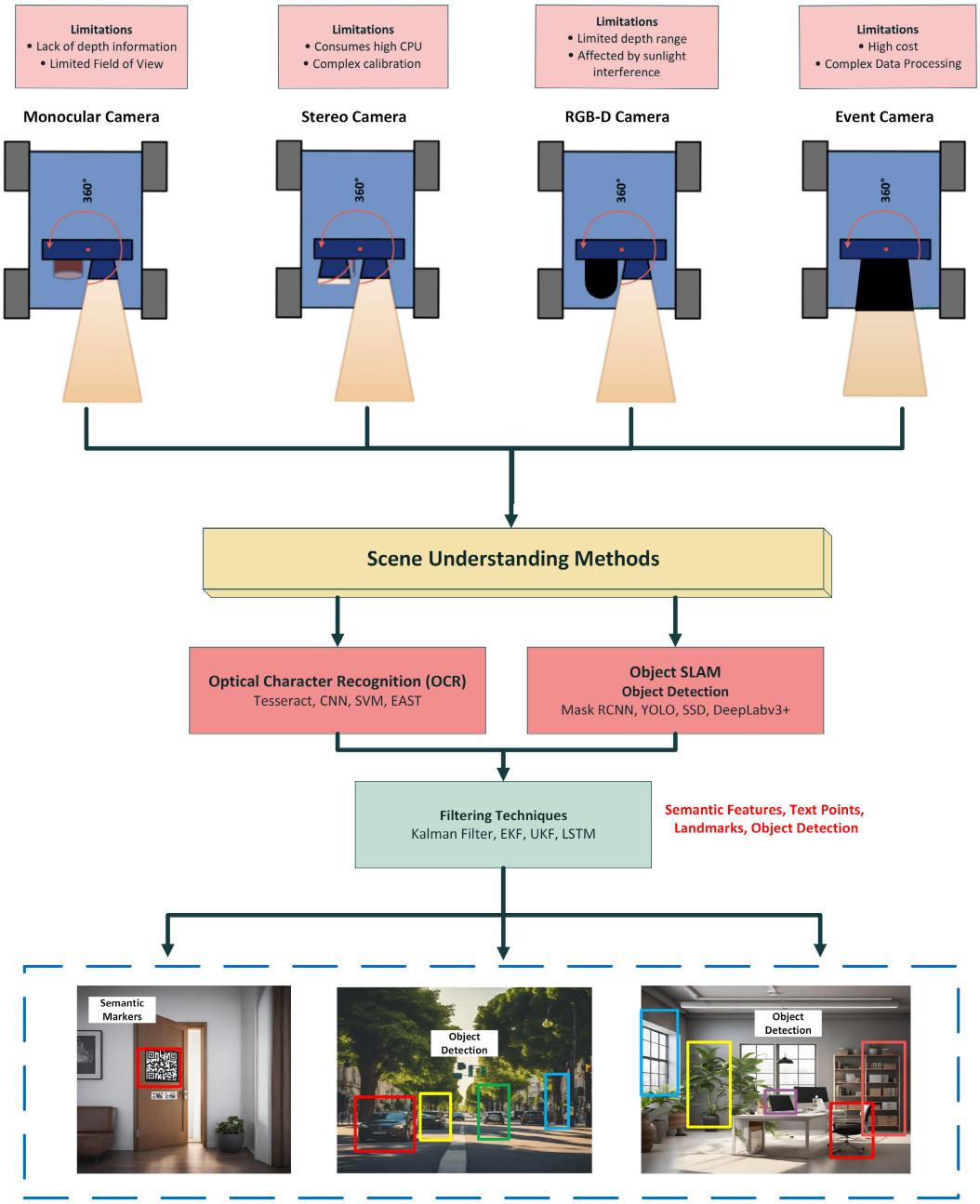



**FIGURE 15.** Overview of scene understanding SLAM system: Incorporating OCR and object detection (outdoor and indoor environments) into the typical V-SLAM methodologies has the potential to elevate scene comprehension and bolster the capabilities of V-SLAM systems.

strategy based on coarse semantic plane (CSP) constraints into direct sparse odometry with loop closure (LDSO). The system is capable of building a dense semantic map using a monocular camera by modeling numerous small semantic planes from sparse point clouds, which reduces hardware requirements. Semantic information is extracted from the environment using DeepLabv3+ [106]. To handle moving objects and improve robustness, the algorithm employs a point group movement consistency check to distinguish between moving and still dynamic points. This is done by using epipolar geometry to determine the motion status of a dynamic point, thereby avoiding the filtering out of all dynamic points. Furthermore, the algorithm uses the CSP priori to improve tracking accuracy. This is achieved by discarding surficial and building points that disagree with nearby points of the same semantic class. The authors evaluated their method using the KITTI dataset [152]. The evaluation involved running each sequence in KITTI ten times to obtain the average absolute trajectory errors (ATE). The performance of the proposed method was compared with

ORB-SLAM2 [12], LDSO [58], LDSO with PGMC, and LDSO with CSP-based point reselection strategy. The results indicated that the proposed method outperformed LDSO and achieved comparable performance to ORB-SLAM2 while maintaining the robustness of LDSO. Moreover, the proposed method produced a dense semantic map of the complex urban environment with superior quality and clearer main structures compared to the maps generated by ORB-SLAM2 and LDSO.

Recently proposed, SO-SLAM is a monocular object SLAM system designed to model objects using quadrics and build an object-level map representing the environment [153]. This system proposes two new modules, single frame initialization and orientation fine optimization, which rely on three spatial structure constraints: scale proportional constraints, symmetrical texture constraints, and supporting plane constraints. The inclusion of these modules significantly reduce the object SLAM systems' dependence on the number and variability of observations, improving its robustness. One of the key contributions of SO-SLAM is the ability to extract constraints from a single frame observation, allowing the set up of a complete ellipsoid from scratch. Moreover, a new flexible object scale prior called Scale Proportional Constraint (SPC) is proposed, which constrains an object's proportional scale rather than its precise scale. To further enhance the system's ability to constrain an object's orientation based on its symmetry property, an Improved-DT descriptor is introduced. This descriptor combines the advantages of both pixel closest distance and point descriptor's nearest edge distance. The authors conducted experiments that show the effectiveness of the proposed single-frame initialization and texture orientation optimization modules. However, it is worth noting that the proposed method has a limitation in accurately estimating the center and scale of small objects such as books and keyboards. This limitation can negatively affect the accuracy of the three-dimensional symmetry point in the texture constraint. Despite this drawback, SO-SLAM shows promising advancements in the domain of monocular object SLAM.

This review highlights that the discussed SLAM methods primarily focus on spatial relationships between objects in the scene. However, it is evident that there is still ample potential in incorporating temporal relationships between objects in SLAM applications, which could lead to significant improved performance. Combining both spatial and temporal information can further enhance the robustness of SLAM systems. Furthermore, recent advancements in visual recognition techniques offer an opportunity to positively impact SLAM systems by improving feature detection, object tracking, and scene reconstruction. As a result, future research in this field could effectively focus on incorporating both spatial and temporal information, as well as leveraging state-of-the-art visual recognition techniques, to develop more accurate and efficient SLAM systems. Such advancements have the potential to revolutionize the field and enable SLAM systems to perform even more effectively in complex real-world scenarios.

## VII. DATASETS AND EVALUATION TOOLS

This section presents an overview of various evaluation modes, metrics, and comparisons used in both qualitative and quantitative assessments of SLAM algorithms. It also highlights the accessibility of public datasets tailored for various sensor types and the availability of open-source codes for research and development purposes.

### A. DATASETS

Researchers often evaluate the effectiveness of a proposed SLAM method by testing it on publicly available datasets and comparing its performance with state-of-the-art SLAM algorithms. Table 8 provides a comprehensive overview of commonly used public datasets that facilitate the evaluation of various aspects of SLAM performance. These datasets include evaluation modes, metrics, qualitative and quantitative comparisons, as well as availability of open-source codes.

#### 1) OUTDOOR ENVIRONMENTS

In the domain of outdoor environments, several datasets are available, focusing on urban scenes with diverse driving scenarios and environmental conditions. Notably, the KITTI, VKITTI, and Cityscapes datasets are commonly used in research. The KITTI dataset [152], developed collaboratively by the Karlsruhe Institute of Technology and Toyota American Institute of Technology, is a renowned outdoor environment dataset for autonomous driving scenarios. It is the largest dataset for evaluating computer vision algorithms in such scenarios, covering monocular and binocular vision, Velodyne LiDAR, and POS trajectory data. KITTI is widely used and includes 389 stereo and optical flow image pairs, stereo visual odometry sequences spanning 39.2 km, and over 200,000 3D object annotations captured in cluttered scenarios. Based on KITTI dataset, another dataset called Virtual KITTI (VKITTI) was introduced in [154]. VKITTI is a fully annotated photorealistic synthetic video dataset, created using advanced computer graphics technology and a novel cloning method. This dataset facilitates the scientific evaluation of the impact of various lighting and weather conditions on the recognition performance of statistical computer vision models. The Cityscapes dataset [155] consists of street scenes captured with high-resolution stereo cameras and semantic segmentation labels, making it suitable for evaluating Semantic SLAM algorithms. This dataset features a large and diverse set of stereo video sequences filmed on the streets of 50 different cities. Among these images, 5000 possess pixel-level annotations of high quality, while an additional 20,000 have coarse annotations, offering valuable resources for SLAM methods that rely on large volumes of weakly-labeled data.

**TABLE 7.** Comparative overview of scene understanding approaches.

| Method | Sensors | Dense / Feature | Data Assosciation | Map Representation | GPU | Real-time* | Code Available |
|---|---|---|---|---|---|---|---|
| Balaban *et al.* [149] | RGB-D | Feature | ICP | Octree | X | X | X |
| Text-MCL [27] | LRF Monocular | Feature | AMCL | Grid map | X | X | X |
| Martins *et al.* [28] | RGB-D Stereo LiDAR | Feature | Kalman filter | Semantic object map | ✓ | ✓ | Github |
| Bao *et al.* [66] | Monocular | Dense | Bundle Adjustment | Octree | X | X | X |
| SO-SLAM [153] | Monocular | Dense | Manually annotated | Object-level map | ✓ | ✓ | Github |

* If the method is indicated as not having real-time functionality, it is either because the authors have pointed it out or it has been evaluated through simulations.

#### 2) INDOOR ENVIRONMENTS

Several datasets are specifically tailored to indoor environments, such as EuRoC, HRPSlam, ICL-NUIM, and TUM-RGBD datasets. The EuRoC datasets [156] comprise visual-inertial sequences recorded using a stereo camera and an IMU on a Micro Aerial Vehicle (MAV). EuRoC consists of eleven datasets that cover a range of scenarios, from slow flights in good visual conditions to fast flights with motion blur and low illumination. These datasets are divided into two groups: the first batch was collected in an industrial environment with millimeter-precise ground truth obtained from a laser tracking system, while the second batch was recorded in a room equipped with a motion capture system. The HRPSLAM datasets [158], proposed by the Humanoid Research Group at the National Institute of Advanced Industrial Science and Technology (AIST) in Japan, serve as a benchmark for V-SLAM algorithms. These datasets focus on assessing the performance of indoor visual odometry and V-SLAM techniques in dynamic environments. They were developed using an on-board RGB-D camera mounted on the HRP-4 humanoid robot, and include challenging scenarios such as shaking, full occlusion, and falling down to evaluate humanoid visual sensing capabilities. The ICL-NUIM dataset [159] provided by the Imperial College London, consists of RGB-D sequences and ground truth poses of indoor scenes with various textures, lighting conditions, and occlusions. The TUM-RGBD dataset [13] is made available by the Technical University of Munich (TUM). It contains RGB-D indoor sequences recorded using a Microsoft Kinect sensor, along with ground truth poses, and is used to evaluate SLAM algorithms that rely on RGB-D sensors. These datasets collectively offer valuable resources for evaluating SLAM algorithms in indoor environments with diverse challenges.

#### 3) HYBRID ENVIRONMENTS

Some datasets provide scenes that encompass both indoor and outdoor environments, one of which is the ETH3D dataset [161], curated by the Computer Vision and Geometry Lab at ETH Zurich. This dataset features high-quality 3D scans of real-world environments, including RGB-D sequences, stereo images, LiDAR scans, ground truth poses, and surface reconstructions. Another dataset, TUM VI [162], is provided by the Technical University of Munich and includes visual-inertial sequences captured using a stereo camera and an IMU, along with ground truth poses. TUM VI serves as a benchmark to evaluate visual-inertial SLAM algorithms that use both visual and inertial measurements to estimate the robot's pose and map features. This dataset contains scenes from both indoor and outdoor environments.

#### 4) SYNTHETIC ENVIRONMENTS

Synthetic datasets have become increasingly popular in recent years. One such dataset is TartanAir [163], introduced by the Tartan Robotics Group at Carnegie Mellon University. TartanAir offers a large-scale dataset of high-fidelity photorealistic 3D environments, including dynamic objects, diverse lighting, and weather conditions in both indoor and outdoor scenes. Another notable dataset is InteriorNet [164], developed by Stanford University's AI research group. InteriorNet consists of over 10,000 real-world indoor scenes featuring different layouts, styles, and object configurations. These synthetic datasets provide valuable resources for evaluating and developing SLAM algorithms in controlled and diverse environments.

While high-quality datasets are available for evaluating SLAM algorithms, it is crucial to acknowledge that uncertainties persist regarding their real-world applicability. Relying exclusively on testing these algorithms with such datasets might constrain the evaluation to specific geographic regions, potentially underestimating their effectiveness in different locations. Furthermore, the limited implementation of these algorithms in real-world scenarios can be attributed to the high computational demands of V-SLAM algorithms. The online implementation becomes challenging without dedicated parallel processing hardware, as most mobile computers lack the computing capabilities of desktop GPUs. These computational constraints pose significant challenges for widespread deployment and real-time applications of V-SLAM algorithms.

### B. METRICS

When assessing SLAM algorithms, various aspects, such as power and time consumption, complexity, and accuracy,

**TABLE 8.** Common public datasets used to evaluate V-SLAM performance.

| Dataset Name | Type of sensors | Provided data | Static / Dynamic | Indoor / Outdoor | Evaluation metrics | Ground truth | VSLAM Methods |
|---|---|---|---|---|---|---|---|
| KITTI [152] | GPS / IMU<br>Laser scanner<br>Monocular Camera<br>Stereo camera | Monocular images<br>Stereo images<br>IMU measurements | Static / Dynamic | Outdoor | Average Precision (AP)<br>Average Orientation Similarity (AOS) | Semi-dense (50%) | [63], [65], [80], [98], [129]<br>[66], [85], [87], [118], [142] |
| VKITTI [154] | Monocular camera | RGB images<br>Depth maps | Static / Dynamic<br>(different weather conditions) | Outdoor | Multiple Object Tracking Accuracy (MOTA) | Automatically generated<br>ground truth annotations | [78], [121], [129] |
| Cityscapes [155] | GPS<br>Stereo cameras<br>Vehicle odometry sensors | Stereo images<br>GPS measurements<br>Odometry measurements | Static / Dynamic<br>(different weather conditions) | Outdoor | Intersection over Union (IoU)<br>Average Precision (AP) | Fully convolutional networks<br>(FCN-8s) | [78], [129] |
| EuRoC [156] | Stereo camera<br>IMU<br>Laser scanner | Stereo images<br>IMU measurements | Static | Indoor | Absolute Trajectory Error (ATE) | Station/Motion capture systems | [47], [50], [143], [157] |
| HRPSlam [158] | RGB-D camera<br>IMU | RGB images with depth maps<br>IMU measurements | Static / Dynamic | Indoor | Absolute Trajectory Error (ATE)<br>Relative Pose Error (RPE) | Motion capture system | [84] |
| ICL-NUIM [159] | RGB-D Camera | RGBD data<br>Depth maps | Static | Indoor | Cloud/mesh distance metric<br>Root Mean Square Error (RMSE) | 3D surface ground truth | [56], [65], [128], [133], [153] |
| TUM-RGBD [13] | RGB-D Camera<br>(Microsoft Kinect) | RGBD data<br>Accelerometer data | Static / Dynamic | Indoor | Absolute Trajectory Error (ATE)<br>Relative Pose Error (RPE) | Motion capture system | [18], [62], [71], [72], [75], [77], [79], [80]<br>[76], [78], [81], [97], [98], [115], [160]<br>[53], [82]–[84], [86], [100], [101], [104]<br>[29], [73], [74], [126], [127], [132], [157]<br>[56], [65], [114], [123], [133], [153] |
| ETH3D [161] | Stereo camera<br>Laser scanner | RGB images<br>Stereo images<br>Camera poses | Static / Dynamic | Indoor / Outdoor | Absolute Trajectory Error (ATE) | 3D point clouds | [157] |
| TUM VI [162] | Stereo camera<br>IMU<br>Light sensor | Stereo images (high dynamic range)<br>IMU measurements | Static / Dynamic | Indoor / Outdoor | Absolute Trajectory Error (ATE) | Motion capture system | [47] |
| TartanAir [163] | Photo-realistic simulation<br>environments(AirSim) | Stereo RGB images<br>Depth images<br>Segmentation and optical flow<br>Camera poses<br>LiDAR point cloud | Static / Dynamic<br>(different lighting &<br>weather conditions) | Indoor / Outdoor | Absolute Trajectory Error (ATE)<br>Relative Pose Error (RPE)<br>Success Rate (SR) | Grid maps/ Optical flow/<br>Stereo disparity/ Simulated<br>LiDAR measurements | [85], [157] |
| InteriorNet [164] | ViSim<br>(Synthetic data) | RGB images<br>Stereo images<br>Depth maps<br>Camera poses<br>IMU measurements | Static / Dynamic<br>(different random lightings) | Indoor | Absolute Trajectory Error (ATE) | Synthesize related ground truth | [127] |

can be taken into account [32]. However, the most crucial criterion is accuracy, which is usually evaluated by comparing estimates with ground-truth data. The primary accuracy metrics commonly used for SLAM are RPE (Relative Pose Error) and ATE (Absolute Trajectory Error). RPE measures the local accuracy of the estimated trajectory, whereas ATE measures the global consistency of the estimated trajectory. These metrics are fundamental in determining the performance and reliability of SLAM algorithms.

#### 1) RELATIVE POSE ERROR (RPE)

To estimate the system drift, the RPE is utilized to measure the difference in pose changes between two identical timestamps. At a specific time step $i$, the RPE is defined as:

$$E_i = (Q_i^{-1} Q_{i+\Delta})^{-1} (P_i^{-1} P_{i+\Delta}), \tag{1}$$

where $P_i$ is the estimated pose, $Q_i$ is the ground truth, and $\Delta$ is a fixed time interval.

By calculating the individual relative pose errors along the sequence from a set of $n$ camera poses (where $m = n - \Delta$), the SLAM algorithm's performance can be evaluated. The root mean square error (RMSE) is then used to determine the overall error and assess the algorithm's performance:

$$RMSE(E_{1:n}, \Delta) = (\frac{1}{m} \sum_{i=1}^{m} ||trans(E_i)||^2)^{1/2}, \tag{2}$$

where $trans(E_i)$ is the translational component of the relative pose error. In practical scenarios, there exist numerous options available for selecting the value of the time interval. To achieve a comprehensive evaluation of the algorithm's performance, the average RMSE across all possible time interval values can be computed:

$$RMSE(E_{1:n}) = \frac{1}{n} \sum_{\Delta=1}^{n} RMSE(E_{1:n}, \Delta) \tag{3}$$

#### 2) ABSOLUTE TRAJECTORY ERROR (ATE)

The ATE computes the difference between the true value of the camera pose and the estimated value provided by the SLAM algorithm. At a particular time step $i$, the ATE can be computed as follows:

$$F_i = Q_i^{-1} S P_i, \tag{4}$$

where $P_i$ represents the estimated trajectory, $Q_i$ denotes the ground truth trajectory, and $S$ represents the rigid-body transformation corresponding to the least-squares solution [165].

Similar to the RPE, the RMSE is calculated across all time indices for the translational components as follows:

$$RMSE(F_{1:n}) = (\frac{1}{n} \sum_{i=1}^{n} ||trans(F_i)||^2)^{1/2} \tag{5}$$

It is worth noting that some researchers prefer to evaluate the mean error instead of the root mean squared error, as the former method is less sensitive to outliers and provides a more robust measure of the performance.

In addition to the previously mentioned metrics, several other evaluation metrics are commonly used to assess the accuracy of SLAM algorithms, including:

- Average Precision (AP): A measure of object detection accuracy that considers both precision (fraction of true positives among detected positives) and recall (fraction of true positives detected). It is calculated by computing the area under the precision-recall curve, which plots precision against recall for different confidence thresholds.
- Average Orientation Similarity (AOS): A measure of object pose estimation accuracy that considers both translation and rotation errors. It is computed as the average cosine similarity between the estimated and ground-truth orientation vectors of each object instance.
- Multiple Object Tracking Accuracy (MOTA): A measure of tracking accuracy that considers both detection and tracking errors. It is computed as the percentage of missed detections, false positives, and identity switches relative to the total number of ground-truth objects.
- Intersection over Union (IoU): A measure of object segmentation accuracy that calculates the overlap between the predicted and ground-truth bounding boxes or masks. It is computed as the ratio of the intersection area to the union area.
- Success Rate (SR): A measure of visual odometry accuracy that indicates the percentage of frames in which the estimated camera pose error is below a certain threshold. It is commonly used to evaluate the performance of SLAM systems in outdoor environments, where a GPS signal is not available.

## VIII. DISCUSSION AND RECOMMENDATIONS

In recent years, significant developments have been made in various aspects of V-SLAM systems, resulting in reliable solutions and notable improvements. However, there are still unresolved issues and limitations that require further investigation to make V-SLAM techniques more robust. These limitations include limited scalability, sensitivity to lighting variations, performance issues in unstructured or noisy environments, and computational power requirements. Our comprehensive review has identified these open research fields as key challenges that need to be addressed.

### A. LIMITED SCALABILITY

One of the main challenges faced by most V-SLAM algorithms is their limited scalability. These algorithms may struggle to handle large environments, especially those with complex geometries or highly dynamic non-rigid objects. As the environment's scale increases, the computational requirements of the algorithm also grow, making real-time data processing difficult. To address the scalability issue, researchers have proposed several solutions. This problem can be approached using sub-mapping techniques to divide the map into smaller, more manageable segments. This allows

the algorithm to focus on processing localized areas, reducing the overall computational burden. Additionally, exploring hardware acceleration and parallel computing techniques can significantly improve the computational efficiency of V-SLAM algorithms, enabling them to handle larger and more complex environments effectively. By leveraging the full potential of modern hardware, V-SLAM algorithms can make substantial strides towards overcoming their scalability limitations.

### B. ROBUSTNESS IN NOISY ENVIRONMENTS

The presence of noise from various sources in the V-SLAM pipeline can hinder the accuracy of the estimation algorithm, producing inaccurate maps and trajectory estimates. V-SLAM algorithms face challenges when occlusion occurs, causing objects or features to be obscured from the camera's view. To address this issue, researchers propose exploiting temporal information to predict the trajectory of moving objects. By incorporating information about the object's previous position and velocity into the current estimate, the algorithm can make more accurate predictions even when the object is occluded. However, this technique presents new challenges, such as correctly identifying moving objects in the environment and accurately estimating their trajectories. Moreover, the existing literature lacks extensive research on handling reflections and high presence of glass, which can pose significant challenges for V-SLAM algorithms.

Despite some progress in improving SLAM algorithms to handle accuracy issues arising from input errors, as discussed in Section V, there are still challenges to be addressed. To overcome these challenges, incorporating complementary scene understanding methods in V-SLAM approaches can lead to significant improvements in dealing with noisy and challenging environments. For instance, using deep learning techniques like neural networks to estimate camera pose can enhance the algorithm's ability to handle noisy data and improve estimation accuracy. Moreover, as stated earlier in Section II, using event cameras instead of conventional cameras has the potential to bolster the robustness of V-SLAM algorithms in high-speed scenarios and HDR environments. Another approach is to combine data from multiple sensors like cameras, LiDARs, and IMUs, to improve SLAM accuracy and robustness in noisy environments. By leveraging data from multiple sensors, the system can achieve a comprehensive understanding of the environment, leading to better mapping and estimation results. By exploring these techniques and further investigating the optimality of V-SLAM estimation, researchers can enhance the robustness of V-SLAM algorithms and ensure reliable performance in noisy and challenging real-world scenarios.

### C. HANDLING DIVERSE LIGHTING CONDITIONS

V-SLAM algorithms can be sensitive to changes in lighting conditions, leading to potential accuracies in pose estimation and feature tracking. Significant changes in illumination can affect the camera's ability to detect and track features in the environment. To address the impact of lighting variations, researchers have proposed various techniques. One approach is adaptive thresholding [166], where the algorithm dynamically adjusts the threshold for feature detection based on the current illumination level. Another technique involves camera exposure control [167], where the camera settings, such as shutter speed, are modified to optimize image capture in different lighting scenarios. By adjusting the camera's exposure settings, the algorithm can capture images with optimal brightness levels, enhancing its performance under varying lighting conditions. High Dynamic Range (HDR) imaging [168] is another approach utilized to address lighting challenges. HDR imaging involves capturing multiple images taken at varying exposure levels and combining them to create a composite image with a broader range of luminance values. This helps mitigate the effects of high contrast lighting situations, providing more reliable feature detection and tracking. While these techniques have shown promise in improving SLAM performance under varying lighting conditions, further research is needed to explore and address the intricacies of V-SLAM in changing lighting environments. Ensuring reliable performance and accuracy across a wide range of lighting conditions remains an important focus for advancing V-SLAM algorithms.

### D. ROBUSTNESS IN UNSTRUCTURED ENVIRONMENTS

The efficacy of V-SLAM algorithms heavily relies on the detection and tracking of features in the environment, such as edges, corners, and textured regions. However, in unstructured environments with a lack of distinctive texture, such as blank walls or surfaces with little features, these algorithms may encounter difficulties in identifying suitable features to track, leading to a decline in performance. Moreover, in textureless environments with few salient feature points, drift errors in robot position and orientation can cause system failures. While Semantic SLAM has been viewed as a significant improvement to solve this issue, there is still room for improvement in the field of semantic segmentation. Further research and development in semantic segmentation can lead to more robust SLAM implementations, enabling these algorithms to perform more effectively in unstructured environments with limited texture and features. A semantic scene analysis reduces the reliance on environmental features while also utilizing available information from other objects in the scene.

### E. VARIOUS FEATURE PROCESSING

One of the critical issues with current V-SLAM solutions is their lack of adaptability to different environments. These solutions heavily rely on specific types of features, and their failure to detect them can lead to a significant degradation in the accuracy. This issue may occur due to intermittent feature presence in challenging environments, sudden movements, or the vision system's inability to detect

them if the SLAM system solely depends on a limited set of features, neglecting other image elements or new objects that the system was not trained to detect. To address these challenges, the vision system should be flexible enough to accommodate various types of features that are relevant to the robot's environment. For instance, after a transition from indoor to outdoor environments, the system should be adaptable and use different types of features that are more relevant to the new surroundings. Developing robust feature processing techniques that can handle diverse environments and adapt to changing conditions will significantly improve the performance and reliability of SLAM systems in real-world scenarios.

### F. SCENE UNDERSTANDING

Recent advances in deep learning have paved the way for the widespread use of object detectors in V-SLAM. However, these object detectors often lack the ability to consider the spatial and temporal relationships between detected objects. For example, detecting a chair in one frame and a table in the next frame may not capture their spatial relationship or that they belong to the same room or it can spot a person holding a cup in one frame and a water dispenser in the next frame but may not link these frames to understand that the person is filling the cup. Another case in object grasping, V-SLAM methods may not inherently grasp the orientation of objects like a door in the open or closed state. As discussed in Section VI, integrating spatial and temporal relationships into object detection algorithms remains an area that requires further research in V-SLAM. Techniques such as 3D geometry, object tracking, and scene understanding techniques can be explored to address this limitation. Enhancing scene understanding by incorporating these relationships will contribute to a more comprehensive understanding of the environment, enabling V-SLAM systems to produce more accurate and robust results.

### G. QUALITY OF INFORMATION AND COMPUTATIONAL COST

V-SLAM systems often demand substantial computational resources, making real-time implementation challenging, especially on low-power devices like drones or mobile robots. Achieving a balance between information retrieval and computational cost is one of the major challenges in V-SLAM. Dense maps can provide high-dimensional, complete scene information, but processing this data in real-time can be computationally demanding. On the other hand, sparse representations are less computationally intensive but may not capture all the necessary information. In addition to balancing information retrieval and computational cost, V-SLAM systems also need to deal with real-time performance issues, such as frame losses during peak processing periods, which can adversely impact the V-SLAM system's performance.

To address these issues, researchers are exploring various methods. One approach involves using parallel processing techniques to distribute the computational load across multiple processors, thus improving overall efficiency. Another method focuses on recovering a dense semantic map from sparse point clouds, reducing the hardware requirements for generating a detailed map. Also, researchers have explored hardware acceleration using Field-Programmable Gate Arrays (FPGAs) to improve the computational efficiency of V-SLAM algorithms. Another approach involves developing V-SLAM algorithms that can operate on compressed or low-resolution images, which reduces the computational burden without compromising the overall performance significantly. Additionally, as stated previously in Section V, an innovative technique has been recently proposed to reduce the computational load on low-power devices by offloading resource-intensive V-SLAM processing steps from mobile robots to edge computing. By leveraging edge computing, SLAM algorithms can process data more efficiently and handle larger datasets, though they may face limitations, such as increased latency that must be carefully managed. Finding a balance between computational efficiency and information richness remains an ongoing challenge in V-SLAM research. By exploring parallel processing techniques, hardware acceleration, image compression methods, and cloud-based solutions, researchers can develop more efficient V-SLAM implementations that can achieve better real-time performance and cater to a broader range of devices and applications.

## IX. CONCLUSION

This paper presents a comprehensive overview of the current state of traditional and modern V-SLAM approaches. We have discussed various V-SLAM methods, including monocular, RGB-D, stereo, and event-based methods. These approaches utilize visual and inertial data, and some even integrate LiDAR information for enhanced performance in different environments.

Throughout the survey, we highlighted the strengths and limitations of each approach, and identified key challenges in the field of V-SLAM. These challenges include limited scalability, sensitivity to lighting variations, performance issues in unstructured or noisy environments, and the requirement for substantial computational power. In the field of scene understanding, we discussed the need for algorithms that can effectively consider spatial and temporal relationships between detected objects, enabling more precise and meaningful data in mapping and localization. Additionally, we emphasized the importance of balancing information retrieval and computational cost in V-SLAM systems, as dense maps can provide valuable scene information but come with higher computational demands, while sparse representations may lack crucial details. The survey also included an overview of commonly used datasets for V-SLAM evaluation. There is potential for the development of outdoor datasets that include more dynamic scenes, reflecting real-world scenarios more accurately. Furthermore, we discussed the evaluation metrics used to assess the

accuracy of V-SLAM algorithms, focusing on metrics such as RPE and ATE. These metrics provide a basis for evaluating the algorithms' performance in terms of pose estimation and trajectory accuracy.

To conclude, the field of V-SLAM is rapidly evolving with an increasing number of publications each year. There are several areas of potential improvement for future research, including multi-agent approaches, multimodal methods that combine multiple sensor inputs, and enhancing the robustness of V-SLAM algorithms to handle sensor and environment noise. Continued research and development in these areas will undoubtedly lead to more advanced and reliable V-SLAM solutions for a wide range of applications.

## REFERENCES


[1] J. Cheng, L. Zhang, Q. Chen, X. Hu, and J. Cai, "A review of visual SLAM methods for autonomous driving vehicles," *Eng. Appl. Artif. Intell.*, vol. 114, Sep. 2022, Art. no. 104992.

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

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

[4] R. Azzam, T. Taha, S. Huang, and Y. Zweiri, "A deep learning framework for robust semantic SLAM," in *Proc. Adv. Sci. Eng. Technol. Int. Conf. (ASET)*, Feb. 2020, pp. 1–7.

[5] H. Kuang, Y. Li, Y. Zhang, Y. Wan, and G. Ge, "Research on rapid location method of mobile robot based on semantic grid map in large scene similar environment," *Robotica*, vol. 40, no. 11, pp. 4011–4030, Nov. 2022.

[6] S. R. Bista, D. Hall, B. Talbot, H. Zhang, F. Dayoub, and N. Sünderhauf, "Evaluating the impact of semantic segmentation and pose estimation on dense semantic SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 5328–5335.

[7] C. Analytics. *Web of Science—All Databases*. [Online]. Available: https://www.webofscience.com/wos/

[8] S. Zhang, L. Zheng, and W. Tao, "Survey and evaluation of RGB-D SLAM," *IEEE Access*, vol. 9, pp. 21367–21387, 2021.

[9] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Reality*, Oct. 2011, pp. 127–136.

[10] A. Concha and J. Civera, "RGBDTAM: A cost-effective and accurate RGB-D tracking and mapping system," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 6756–6763.

[11] A. Fontán, J. Civera, and R. Triebel, "Information-driven direct RGB-D odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4928–4936.

[12] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[13] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580.

[14] Q. Sun, J. Yuan, X. Zhang, and F. Duan, "Plane-edge-SLAM: Seamless fusion of planes and edges for SLAM in indoor environments," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 4, pp. 2061–2075, Oct. 2021.

[15] L. Ma, C. Kerl, J. Stückler, and D. Cremers, "CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1285–1291.

[16] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration," *ACM Trans. Graph.*, vol. 36, no. 4, p. 1, Aug. 2017.

[17] M. Hsiao, E. Westman, G. Zhang, and M. Kaess, "Keyframe-based dense planar SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5110–5117.

[18] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 2100–2106.

[19] T. Schöps, T. Sattler, and M. Pollefeys, "BAD SLAM: Bundle adjusted direct RGB-D SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 134–144.

[20] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison, "Elasticfusion: Dense slam without a pose graph," in *Proc. Robot., Sci. Syst.*, 2015.

[21] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially extended kinectfusion," Tech. Rep., 2012.

[22] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3D reconstruction at scale using voxel hashing," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 1–11, Nov. 2013.

[23] D. Galvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012.

[24] M. Aizat, A. Azmin, and W. Rahiman, "A survey on navigation approaches for automated guided vehicle robots in dynamic surrounding," *IEEE Access*, vol. 11, pp. 33934–33955, 2023.

[25] S. Mokssit, D. B. Licea, B. Guermah, and M. Ghogho, "Deep learning techniques for visual SLAM: A survey," *IEEE Access*, vol. 11, pp. 20026–20050, 2023.

[26] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual SLAM algorithms: A survey from 2010 to 2016," *IPSJ Trans. Comput. Vis. Appl.*, vol. 9, no. 1, pp. 1–11, Dec. 2017.

[27] G. Ge, Y. Zhang, W. Wang, Q. Jiang, L. Hu, and Y. Wang, "Text-MCL: Autonomous mobile robot localization in similar environment using text-level semantic information," *Machines*, vol. 10, no. 3, p. 169, Feb. 2022.

[28] R. Martins, D. Bersan, M. F. M. Campos, and E. R. Nascimento, "Extending maps with semantic and contextual object information for robot navigation: A learning-based framework using visual and depth cues," *J. Intell. Robot. Syst.*, vol. 99, nos. 3–4, pp. 555–569, Sep. 2020.

[29] Y. Sun, M. Liu, and M. Q.-H. Meng, "Motion removal for reliable RGB-D SLAM in dynamic environments," *Robot. Auton. Syst.*, vol. 108, pp. 115–128, Oct. 2018.

[30] C. Xu, S. Zhang, and K. Jiang, "A multi-source information fusion method for mobile robot visual-inertial navigation," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2023, pp. 296–301.

[31] K. Cao, R. Liu, Z. Wang, K. Peng, J. Zhang, J. Zheng, Z. Teng, K. Yang, and R. Stiefelhagen, "Tightly-coupled LiDAR-visual SLAM based on geometric features for mobile agents," 2023, *arXiv:2307.07763*.

[32] W. Chen, G. Shang, A. Ji, C. Zhou, X. Wang, C. Xu, Z. Li, and K. Hu, "An overview on visual SLAM: From tradition to semantic," *Remote Sens.*, vol. 14, no. 13, p. 3010, Jun. 2022.

[33] H. Liu, G. Zhang, and H. Bao, "Robust keyframe-based monocular SLAM for augmented reality," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Sep. 2016, pp. 1–10.

[34] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[35] J. Engel, J. Stückler, and D. Cremers, "Large-scale direct SLAM with stereo cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 1935–1942.

[36] R. Gomez-Ojeda, F.-A. Moreno, D. Zuñiga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez, "PL-SLAM: A stereo SLAM system through the combination of points and line segments," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 734–746, Jun. 2019.

[37] G. Alenyà, S. Foix, and C. Torras, "Using ToF and RGBD cameras for 3D robot perception and manipulation in human environments," *Intell. Service Robot.*, vol. 7, no. 4, pp. 211–220, Oct. 2014.

[38] Q. Jin, Y. Liu, Y. Man, and F. Li, "Visual SLAM with RGB-D cameras," in *Proc. Chin. Control Conf. (CCC)*, Jul. 2019, pp. 4072–4077.

[39] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, Jan. 2022.

[40] Y. Zuo, J. Yang, J. Chen, X. Wang, Y. Wang, and L. Kneip, "DEVO: Depth-event camera visual odometry in challenging conditions," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 2179–2185.

[41] S. Bryner, G. Gallego, H. Rebecq, and D. Scaramuzza, "Event-based, direct camera tracking from a photometric 3D map using nonlinear optimization," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 325–331.

[42] H. Rebecq, T. Horstschaefer, G. Gallego, and D. Scaramuzza, "EVO: A geometric approach to event-based 6-DOF parallel tracking and mapping in real time," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 593–600, Apr. 2017.

[43] M. Chghaf, S. Rodriguez, and A. E. Ouardi, "Camera, LiDAR and multi-modal SLAM systems for autonomous ground vehicles: A survey," *J. Intell. Robot. Syst.*, vol. 105, no. 1, p. 2, Apr. 2022, doi: 10.1007/s10846-022-01582-8.

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

[45] E. López, S. García, R. Barea, L. Bergasa, E. Molinos, R. Arroyo, E. Romera, and S. Pardo, "A multi-sensorial simultaneous localization and mapping (SLAM) system for low-cost micro aerial vehicles in GPS-denied environments," *Sensors*, vol. 17, no. 4, p. 802, Apr. 2017. [Online]. Available: https://www.mdpi.com/1424-8220/17/4/802

[46] X. He, W. Gao, C. Sheng, Z. Zhang, S. Pan, L. Duan, H. Zhang, and X. Lu, "LiDAR-visual-inertial odometry based on optimized visual point-line features," *Remote Sens.*, vol. 14, no. 3, p. 622, Jan. 2022. [Online]. Available: https://www.mdpi.com/2072-4292/14/3/622

[47] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[48] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 994–1001, Apr. 2018.

[49] A. Tourani, H. Bavle, J. L. Sanchez-Lopez, and H. Voos, "Visual SLAM: What are the current trends and what to expect?" *Sensors*, vol. 22, no. 23, p. 9297, Nov. 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/23/9297

[50] T. Qin, P. Li, and S. Shen, "VINS-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[51] B. Vishnyakov, I. Sgibnev, V. Sheverdin, A. Sorokin, P. Masalov, K. Kazakhmedov, and S. Arseev, "Real-time semantic SLAM with DCNN-based feature point detection, matching and dense point cloud aggregation," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 43, pp. 399–404, Jun. 2021. [Online]. Available: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLIII-B2-2021/399/2021/

[52] N. Yang, R. Wang, X. Gao, and D. Cremers, "Challenges in monocular visual odometry: Photometric calibration, motion bias and rolling shutter effect," 2017, pp. 1–12, *arXiv:1705.04300*.

[53] S. Han and Z. Xi, "Dynamic scene semantics SLAM based on semantic segmentation," *IEEE Access*, vol. 8, pp. 43563–43570, 2020.

[54] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for RGB-D cameras," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 3748–3754.

[55] G. Younes, D. Asmar, E. Shammas, and J. Zelek, "Keyframe-based monocular SLAM: Design, survey, and future directions," *Robot. Auton. Syst.*, vol. 98, pp. 67–88, Dec. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889017300647

[56] F. Schenk and F. Fraundorfer, "RESLAM: A real-time robust edge-based SLAM system," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 154–160.

[57] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.

[58] X. Gao, R. Wang, N. Demmel, and D. Cremers, "LDSO: Direct sparse odometry with loop closure," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 2198–2204.

[59] R. Azzam, T. Taha, S. Huang, and Y. Zweiri, "Feature-based visual simultaneous localization and mapping: A survey," *Social Netw. Appl. Sci.*, vol. 2, no. 2, pp. 1–24, Feb. 2020.

[60] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision—ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Germany: Springer, 2006, pp. 430–443.

[61] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Computer Vision—ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Germany: Springer, 2010, pp. 778–792.

[62] Y. Wu, Y. Zhang, D. Zhu, Y. Feng, S. Coleman, and D. Kerr, "EAO-SLAM: Monocular semi-dense object SLAM based on ensemble data association," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 4966–4973.

[63] S. Wei, G. Chen, W. Chi, Z. Wang, and L. Sun, "Object clustering with Dirichlet process mixture model for data association in monocular SLAM," *IEEE Trans. Ind. Electron.*, vol. 70, no. 1, pp. 594–603, Jan. 2023.

[64] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.

[65] S. Yang and S. Scherer, "CubeSLAM: Monocular 3-D object SLAM," *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 925–938, Aug. 2019.

[66] Y. Bao, Y. Pan, Z. Yang, and R. Huan, "Utilization of semantic planes: Improved localization and dense semantic map for monocular SLAM in urban environment," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 6108–6115, Jul. 2021.

[67] T. Ran, L. Yuan, J. Zhang, D. Tang, and L. He, "RS-SLAM: A robust semantic SLAM in dynamic environments based on RGB-D sensor," *IEEE Sensors J.*, vol. 21, no. 18, pp. 20657–20664, Sep. 2021.

[68] M. R. U. Saputra, A. Markham, and N. Trigoni, "Visual SLAM and structure from motion in dynamic environments: A survey," *ACM Comput. Surveys*, vol. 51, no. 2, pp. 1–36, Mar. 2019, doi: 10.1145/3177853.

[69] G. Liu, W. Zeng, B. Feng, and F. Xu, "DMS-SLAM: A general visual SLAM system for dynamic scenes with multiple sensors," *Sensors*, vol. 19, no. 17, p. 3714, Aug. 2019.

[70] R. A. Newcombe, D. Fox, and S. M. Seitz, "DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 343–352.

[71] D.-H. Kim and J.-H. Kim, "Effective background model-based RGB-D dense visual odometry in a dynamic environment," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1565–1573, Dec. 2016.

[72] Y. Sun, M. Liu, and M. Q.-H. Meng, "Improving RGB-D SLAM in dynamic environments: A motion removal approach," *Robot. Auto. Syst.*, vol. 89, pp. 110–122, Mar. 2017.

[73] M. Jaimez, C. Kerl, J. Gonzalez-Jimenez, and D. Cremers, "Fast odometry and scene flow from RGB-D cameras based on geometric clustering," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3992–3999.

[74] R. Scona, M. Jaimez, Y. R. Petillot, M. Fallon, and D. Cremers, "StaticFusion: Background reconstruction for dense RGB-D SLAM in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 3849–3856.

[75] S. Li and D. Lee, "RGB-D SLAM in dynamic environments using static point weighting," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2263–2270, Oct. 2017.

[76] F. Zhong, S. Wang, Z. Zhang, C. Chen, and Y. Wang, "Detect-SLAM: Making object detection and SLAM mutually beneficial," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1001–1010.

[77] L. Zhang, L. Wei, P. Shen, W. Wei, G. Zhu, and J. Song, "Semantic SLAM based on object detection and improved octomap," *IEEE Access*, vol. 6, pp. 75545–75559, 2018.

[78] M. Schörghuber, D. Steininger, Y. Cabon, M. Humenberger, and M. Gelautz, "SLAMANTIC–Leveraging semantics to improve VSLAM in dynamic environments," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 3759–3768.

[79] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "DS-SLAM: A semantic visual SLAM towards dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1168–1174.

[80] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4076–4083, Oct. 2018.

[81] Y. Ai, T. Rui, M. Lu, L. Fu, S. Liu, and S. Wang, "DDL-SLAM: A robust RGB-D SLAM in dynamic environments combined with deep learning," *IEEE Access*, vol. 8, pp. 162335–162342, 2020.

[82] X. Long, W. Zhang, and B. Zhao, "PSPNet-SLAM: A semantic SLAM detect dynamic object by pyramid scene parsing network," *IEEE Access*, vol. 8, pp. 214685–214695, 2020.

[83] Z. Hu, J. Zhao, Y. Luo, and J. Ou, "Semantic SLAM based on improved deeplabv3+ in dynamic scenarios," *IEEE Access*, vol. 10, pp. 21160–21168, 2022.

[84] T. Zhang, H. Zhang, Y. Li, Y. Nakamura, and L. Zhang, "FlowFusion: Dynamic dense RGB-D SLAM based on optical flow," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 7322–7328.

[85] Y. Qiu, C. Wang, W. Wang, M. Henein, and S. Scherer, "AirDOS: Dynamic SLAM benefits from articulated objects," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 8047–8053.

[86] B. Chen, G. Peng, D. He, C. Zhou, and B. Hu, "Visual SLAM based on dynamic object detection," in *Proc. 33rd Chin. Control Decis. Conf. (CCDC)*, May 2021, pp. 5966–5971.

[87] G. Li, X. Liao, H. Huang, S. Song, B. Liu, and Y. Zeng, "Robust stereo visual SLAM for dynamic environments with moving object," *IEEE Access*, vol. 9, pp. 32310–32320, 2021.

[88] F. Steinbrücker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense RGB-D images," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Nov. 2011, pp. 719–722.

[89] S. Li and D. Lee, "Fast visual odometry using intensity-assisted iterative closest point," *IEEE Robot. Autom. Lett.*, vol. 1, no. 2, pp. 992–999, Jul. 2016.

[90] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 21–37.

[91] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut' interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[92] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 24, 2011.

[93] L. Niu and Z. Shao, "A fast line rasterization algorithm based on pattern decomposition," *J. Comput.-Aided Design Comput. Graph.*, vol. 22, no. 8, pp. 1286–1292, Sep. 2010.

[94] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[95] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[96] B. Bescos, C. Campos, J. D. Tardos, and J. Neira, "DynaSLAM II: Tightly-coupled multi-object tracking and SLAM," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5191–5198, Jul. 2021.

[97] L. Cui and C. Ma, "SOF-SLAM: A semantic visual SLAM for dynamic environments," *IEEE Access*, vol. 7, pp. 166528–166539, 2019.

[98] J. Cheng, Z. Wang, H. Zhou, L. Li, and J. Yao, "DM-SLAM: A feature-based SLAM system for rigid dynamic scenes," *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 4, p. 202, Mar. 2020.

[99] Q. Jin, Z. Meng, T. D. Pham, Q. Chen, L. Wei, and R. Su, "DUNet: A deformable network for retinal vessel segmentation," *Knowl.-Based Syst.*, vol. 178, pp. 149–162, Aug. 2019.

[100] R. Wang, W. Wan, Y. Wang, and K. Di, "A new RGB-D SLAM method with moving object detection for dynamic indoor scenes," *Remote Sens.*, vol. 11, no. 10, p. 1143, May 2019.

[101] S. Wen, P. Li, Y. Zhao, H. Zhang, F. Sun, and Z. Wang, "Semantic visual SLAM in dynamic environment," *Auton. Robots*, vol. 45, no. 4, pp. 493–504, 2021.

[102] J.-Y. Bouguet et al., "Pyramidal implementation of the affine Lucas Kanade feature tracker description of the algorithm," Intel Corp., Tech. Rep., 2001, p. 4, vol. 5, nos. 1–10.

[103] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6230–6239.

[104] X. Zhao, T. Zuo, and X. Hu, "OFM-SLAM: A visual semantic SLAM for dynamic indoor environments," *Math. Problems Eng.*, vol. 2021, pp. 1–16, Apr. 2021.

[105] X. Dai, S. Long, Z. Zhang, and D. Gong, "Mobile robot path planning based on ant colony algorithm with A* heuristic method," *Frontiers Neurorobotics*, vol. 13, p. 15, Apr. 2019.

[106] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.

[107] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

[108] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8934–8943.

[109] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "RMPE: Regional multi-person pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2353–2362.

[110] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[111] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[112] P. Gritsenko, I. Gritsenko, A. Seidakhmet, and B. Kwolek, "Plane-based humanoid robot navigation and object model construction for grasping," in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, 2018, p. 0.

[113] Y. Wu, Y. Zhang, D. Zhu, X. Chen, S. Coleman, W. Sun, X. Hu, and Z. Deng, "Object SLAM-based active mapping and robotic grasping," in *Proc. Int. Conf. 3D Vis. (3DV)*, Dec. 2021, pp. 1372–1381.

[114] L. Cui and F. Wen, "A monocular ORB-SLAM in dynamic environments," *J. Phys., Conf. Ser.*, vol. 1168, Feb. 2019, Art. no. 052037.

[115] X. Yuan and S. Chen, "SaD-SLAM: A visual SLAM based on semantic and depth information," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 4930–4935.

[116] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands, vol. 9908, Oct. 2016, pp. 354–370.

[117] J. Xiao, A. Owens, and A. Torralba, "SUN3D: A database of big spaces reconstructed using SfM and object labels," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1625–1632.

[118] M. Gonzalez, E. Marchand, A. Kacete, and J. Royan, "TwistSLAM: Constrained SLAM in dynamic environment," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 6846–6853, Jul. 2022.

[119] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. (2019). *Detectron2*. [Online]. Available: https://github.com/facebookresearch/detectron2

[120] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *Proc. 16th Eur. Conf. Comput. Vis. (ECCV)*, Glasgow, U.K. Springer, 2020, pp. 402–419.

[121] M. Rünz and L. Agapito, "Co-fusion: Real-time segmentation, tracking and fusion of multiple objects," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 4471–4478.

[122] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 75–91.

[123] T. Zhang and Y. Nakamura, "Posefusion: Dense rgb-d slam in dynamic human environments," in *Proc. Int. Symp. Exp. Robot.* Springer, 2018, pp. 772–780.

[124] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1302–1310.

[125] A. Golovinskiy and T. Funkhouser, "Min-cut based segmentation of point clouds," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops, ICCV Workshops*, Sep. 2009, pp. 39–46.

[126] M. Runz, M. Buffier, and L. Agapito, "MaskFusion: Real-time recognition, tracking and reconstruction of multiple moving objects," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Oct. 2018, pp. 10–20.

[127] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, "MID-fusion: Octree-based object-level multi-instance dynamic SLAM," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 5231–5237.

[128] M. Strecke and J. Stueckler, "EM-fusion: Dynamic object-level SLAM with probabilistic data association," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5864–5873.

[129] N. Brasch, A. Bozic, J. Lallemand, and F. Tombari, "Semantic monocular SLAM for highly dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 393–400.

[130] D. Zou and P. Tan, "CoSLAM: Collaborative visual SLAM in dynamic environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 354–366, Feb. 2013.

[131] H. Pu, J. Luo, G. Wang, T. Huang, H. Liu, and J. Luo, "Visual SLAM integration with semantic segmentation and deep learning: A review," *IEEE Sensors J.*, vol. 23, no. 19, pp. 22119–22138, Oct. 2023.

[132] L. Xiao, J. Wang, X. Qiu, Z. Rong, and X. Zou, "Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment," *Robot. Auton. Syst.*, vol. 117, pp. 1–16, Jul. 2019.

[133] T. Laidlow, J. Czarnowski, and S. Leutenegger, "DeepFusion: Real-time dense 3D reconstruction for monocular SLAM using single-view depth and gradient predictions," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 4068–4074.

[134] M. Hsiao and M. Kaess, "MH-iSAM2: Multi-hypothesis iSAM using Bayes tree and hypo-tree," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 1274–1280.

[135] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "ISAM2: Incremental smoothing and mapping using the Bayes tree," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 216–235, Feb. 2012.

[136] D. Park and Y.-H. Park, "Identifying reflected images from object detector in indoor environment utilizing depth information," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 635–642, Apr. 2021.

[137] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[138] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[139] X. Wang and J. Wang, "Detecting glass in simultaneous localisation and mapping," *Robot. Auton. Syst.*, vol. 88, pp. 97–103, Feb. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889015302670

[140] E. Yamaguchi, H. Higuchi, A. Yamashita, and H. Asama, "Glass detection using polarization camera and LRF for SLAM in environment with glass," in *Proc. 21st Int. Conf. Res. Educ. Mechatronics (REM)*, Dec. 2020, pp. 1–6.

[141] S. Song, H. Lim, A. J. Lee, and H. Myung, "DynaVINS: A visual-inertial SLAM for dynamic environments," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 11523–11530, Oct. 2022.

[142] C.-C. Chou and C.-F. Chou, "Efficient and accurate tightly-coupled visual-lidar SLAM," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 14509–14523, Sep. 2022.

[143] T. Schneider, M. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart, "Maplab: An open framework for research in visual-inertial mapping and localization," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1418–1425, Jul. 2018.

[144] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 298–304.

[145] P. Sossalla, J. Rischke, J. Hofer, and F. H. P. Fitzek, "Evaluating the advantages of remote SLAM on an edge cloud," in *Proc. 26th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2021, pp. 1–4.

[146] A. J. Ben Ali, M. Kouroshli, S. Semenova, Z. S. Hashemifar, S. Y. Ko, and K. Dantu, "Edge-SLAM: Edge-assisted visual simultaneous localization and mapping," *ACM Trans. Embedded Comput. Syst.*, vol. 22, no. 1, pp. 1–31, Oct. 2022, doi: 10.1145/3561972.

[147] H. Cao, J. Xu, D. Li, L. Shangguan, Y. Liu, and Z. Yang, "Edge assisted mobile semantic visual SLAM," *IEEE Trans. Mobile Comput.*, pp. 1–15, 2022.

[148] P. Sossalla, J. Hofer, J. Rischke, C. Vielhaus, G. T. Nguyen, M. Reisslein, and F. H. P. Fitzek, "DynNetSLAM: Dynamic visual SLAM network offloading," *IEEE Access*, vol. 10, pp. 116014–116030, 2022.

[149] D. Balaban and J. Hart, "Automatic sign reading and localization for semantic mapping with an office robot," 2022, *arXiv:2209.11432*.

[150] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "EAST: An efficient and accurate scene text detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2642–2651.

[151] Y. Wu, Y. Zhang, D. Zhu, Z. Deng, W. Sun, X. Chen, and J. Zhang, "An object SLAM framework for association, mapping, and high-level tasks," *IEEE Trans. Robot.*, vol. 39, no. 4, pp. 2912–2932, 2023.

[152] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[153] Z. Liao, Y. Hu, J. Zhang, X. Qi, X. Zhang, and W. Wang, "SO-SLAM: Semantic object SLAM with scale proportional and symmetrical texture constraints," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4008–4015, Apr. 2022.

[154] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "VirtualWorlds as proxy for multi-object tracking analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4340–4349.

[155] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.

[156] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, Sep. 2016.

[157] Z. Teed and J. Deng, "DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 16558–16569.

[158] T. Zhang and Y. Nakamura, "HRPSlam: A benchmark for RGB-D dynamic SLAM and humanoid vision," in *Proc. 3rd IEEE Int. Conf. Robot. Comput. (IRC)*, Feb. 2019, pp. 110–116.

[159] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 1524–1531.

[160] W. Dai, Y. Zhang, P. Li, Z. Fang, and S. Scherer, "RGB-D SLAM in dynamic environments using point correlations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 373–389, Jan. 2022.

[161] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2538–2547.

[162] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers, "The TUM VI benchmark for evaluating visual-inertial odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1680–1687.

[163] W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor, and S. Scherer, "TartanAir: A dataset to push the limits of visual SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 4909–4916.

[164] W. Li, S. Saeedi, J. McCormac, R. Clark, D. Tzoumanikas, Q. Ye, Y. Huang, R. Tang, and S. Leutenegger, "InteriorNet: Mega-scale multi-sensor photo-realistic indoor scenes dataset," 2018, *arXiv:1809.00716*.

[165] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 5, no. 7, p. 1127, Jul. 1988.

[166] L. Yu, E. Yang, and B. Yang, "AFE-ORB-SLAM: Robust monocular VSLAM based on adaptive FAST threshold and image enhancement for complex lighting environments," *J. Intell. Robot. Syst.*, vol. 105, no. 2, Jun. 2022.

[167] Z. Zhang, C. Forster, and D. Scaramuzza, "Active exposure control for robust visual odometry in HDR environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3894–3901.

[168] C.-H. Yeh and M.-H. Lin, "Robust 3D reconstruction using HDR-based SLAM," *IEEE Access*, vol. 9, pp. 16568–16581, 2021.

**ALI RIDA SAHILI** received the M.Sc. degree in mathematics, learning and vision from École Normale Supérieure (ENS), Paris, France, in 2021, and the master's degree in science and executive engineering in applied mathematics for robotics, control systems, and vision from École des Mines de Paris, France, in 2021. In 2022, he joined Lebanese American University (LAU), Beirut, Lebanon, as a part-time Researcher, where he is working on computer vision and machine learning focusing on robot navigation in indoor/outdoor environments.

**SAIFELDIN HASSAN** received the bachelor's degree in electrical engineering from the Rochester Institute of Technology (RIT), Dubai. Since 2020, he has been responsible for the AI and Robotics Center, RIT Dubai, where he is a Software Engineer working on research in computer vision and robotics applications.

**SABER MUAWIYAH SAKHRIEH** received the bachelor's degree in electrical engineering from the Jordan University of Science and Technology (JUST), Irbid, Jordan, in 2017. He is currently pursuing the master's degree in electrical engineering with the Rochester Institute of Technology (RIT), Dubai, United Arab Emirates. He is a Creative Design and Innovation Instructor with the Emirates Schools Establishment, Dubai. His research interests include AI, 3D printing, computer vision, and robotics.

**JINANE MOUNSEF** (Member, IEEE) received the Ph.D. degree in electrical engineering from Arizona State University, Tempe, AZ, USA. She is currently an Assistant Professor with the Department of Electrical Engineering, Rochester Institute of Technology (RIT), Dubai, United Arab Emirates. She is also leading the AI/Robotics Laboratory, RIT Dubai, an Advisory Board Member with the New York Institute and Laboratory for Artificial Intelligence (NYILAI), and the Head of Research and Education for Women in AI (WAI), UAE Section. She has been in the research field for more than 15 years in machine learning, computer vision, and image processing, during which she was with multidisciplinary teams on a long track of research articles. Her methodological and theoretical research as well as a considerable portion of applied and collaborative work addresses novel techniques in computer vision, in addition to designing and implementing smart systems.

**NOEL MAALOUF** (Member, IEEE) received the B.E. and Ph.D. degrees in electrical and computer engineering from the American University of Beirut, in 2012 and 2018, respectively. He was a Postdoctoral Fellow with the Poznan University of Technology, in 2019. He is currently an Assistant Professor with Lebanese American University. His research interests include legged robots, biomimetics, biomechatronics, and environment perception.

**BILAL ARAIN** received the Ph.D. degree in electrical engineering from the UNSW Canberra at the Australian Defence Force Academy, in 2009. From 2009 to 2021, he was a Postdoctoral Fellow with the Commonwealth Scientific and Industrial Research Organization and the Queensland University of Technology, Brisbane, Australia. During this period, he also worked in the mining and precision agriculture industries in Australia. He is currently an Assistant Professor with the University of Sharjah, United Arab Emirates. His research interest includes addressing practical problems in robotics, with a specific emphasis on visual robot perception, unmanned system navigation, and control.

**TAREK TAHA** received the Ph.D. degree in robotics and mechatronics from the Centre of Excellence for Autonomous Systems (CAS), University of Technology Sydney, Australia. He was with Sydney, Australia, in the Advanced Research and Development Sector, before joining Khalifa University. Then, he founded and led the Autonomous Aerial Laboratory, Algorythma, Abu Dhabi, before leading the Robotics Laboratory, Dubai Future Labs. He has more than 15 years of experience in robotics, autonomous systems, and artificial intelligence while working for industry and academia. He led productive research and development teams to advance research outcomes and develop practical engineering solutions and products for deployment in real-world applications. He supervised/co-supervised various research projects and activities focusing on robot autonomy, perception, navigation, connected autonomous vehicles, and artificial intelligence. He has peer-reviewed papers on autonomous aerial systems, autonomous exploration, navigation and mapping, machine vision, human–robot interaction, assistive robotics, and reinforcement learning.