

As a library, NLM provides access to scientific literature. Inclusion in an NLM database does not imply endorsement of, or agreement with, the contents by NLM or the National Institutes of Health.

Learn more: [PMC Disclaimer](#) | [PMC Copyright Notice](#)



[Front Robot AI](#), 2024; 11: 1347985.

PMCID: PMC11056647

Published online 2024 Apr 10. doi: [10.3389/frobt.2024.1347985](https://doi.org/10.3389/frobt.2024.1347985)

PMID: [38686339](#)

## A review of visual SLAM for robotics: evolution, properties, and future applications

[Basheer Al-Tawil](#)<sup>✉</sup>, [Thorsten Hempel](#), [Ahmed Abdelrahman](#), and [Ayoub Al-Hamadi](#)

### Abstract

Visual simultaneous localization and mapping (V-SLAM) plays a crucial role in the field of robotic systems, especially for interactive and collaborative mobile robots. The growing reliance on robotics has increased complexity in task execution in real-world applications. Consequently, several types of V-SLAM methods have been revealed to facilitate and streamline the functions of robots. This work aims to showcase the latest V-SLAM methodologies, offering clear selection criteria for researchers and developers to choose the right approach for their robotic applications. It chronologically presents the evolution of SLAM methods, highlighting key principles and providing comparative analyses between them. The paper focuses on the integration of the robotic ecosystem with a robot operating system (ROS) as Middleware, explores essential V-SLAM benchmark datasets, and presents demonstrative figures for each method's workflow.

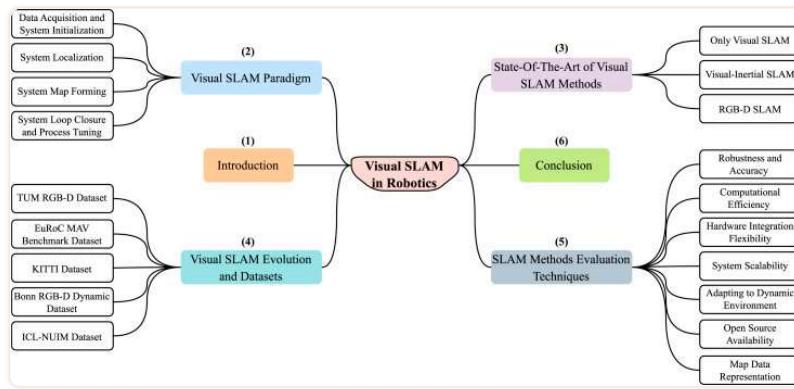
**Keywords:** V-SLAM, interactive mobile robots, ROS, benchmark, Middleware, workflow, robotic applications, robotic ecosystem

### 1 Introduction

Robotics is an interdisciplinary field that involves the creation, design, and operation of tasks using algorithms and programming ([Bongard, 2008](#); [Joo et al., 2020](#); [Awais and Henrich 2010](#); [Fong et al., 2003](#)). Its impact extends to manufacturing, automation, optimization, transportation, medical applications, and even NASA's interplanetary exploration ([Li et al., 2023b](#); [Heyer, 2010](#); [Sheridan, 2016](#); [Mazumdar et al., 2023](#)). Service robots, which interact with people, are becoming more common and useful in everyday life ([Hempel et al., 2023](#); [Lynch et al., 2023](#)). The imperative of integrating automation with human cognitive abilities becomes evident in facilitating a successful collaboration between humans and robots. This helps service robots be more effective in different situations where they interact with people ([Prati et al., 2021](#);

[Strazdas et al., 2020](#); [Zheng et al., 2023](#)). Furthermore, using multiple robots together can help them handle complex tasks better ([Zheng et al., 2022](#); [Li et al., 2023b](#); [Fiedler et al., 2021](#)). To manage and coordinate various processes, a robot operating system (ROS) plays a significant role ([Buyval et al., 2017](#)). It is an open-source framework that aids roboticists in implementing their research and projects with minimal complexity. ROS offers a multitude of features, including hardware integration, control mechanisms, and seamless device implementation into the system, thus facilitating the development and operation of robotic systems ([Altawil and Can 2023](#)).

As shown in [Figure 1](#), the paper is divided into six sections. [Section 1](#) gives the brief introduction about robotics and SLAM. [Section 2](#) presents an overview of the V-SLAM paradigm that delves into its fundamental concepts.



[FIGURE 1](#)

Article organizational chart.

[Section 3](#) presents the state-of-the-art V-SLAM methods, offering insights into the latest advancements of them. Moving forward, [section 4](#) explores the evolution of V-SLAM and discusses the most commonly used datasets. [Section 5](#) focuses on techniques for evaluating SLAM methods, aiding in the selection of appropriate methods. Finally, [Section 6](#) provides the conclusion of the article, summarizing the key points we discovered while working on our review paper.

Recently, we require robots that can move around and work well in places they have never been before. In this regard, simultaneous localization and mapping (SLAM) emerges as a fundamental approach for these robots. The primary goal of SLAM is to autonomously explore and navigate unknown environments by simultaneously creating a map and determining their own position ([Durrant-Whyte, 2012](#); [Mohamed et al., 2008](#)). Furthermore, it provides real-time capabilities, allowing robots to make decisions on-the-fly without relying on pre-existing maps. Its utility extends to the extraction, organization, and comprehension of information, thereby enhancing the robot's capacity to interpret and interact effectively with its environment ([Pal et al., 2022](#); [Lee et al., 2020](#); [Aslan et al., 2021](#)). It is crucial to enable these robots to autonomously navigate and interact in human environments, thus reducing human effort and en-

hancing overall productivity ([Arfa, 2022](#)). The construction of maps is based on the utilization of sensor data, such as visual data, laser scanning data, and data from the inertial measurement unit (IMU), followed by rapid processing ([Macario Barros et al., 2022](#)).

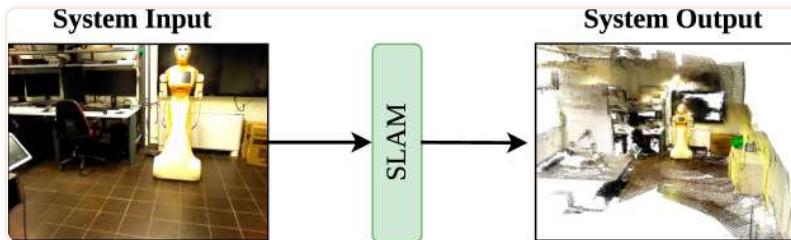
Historically, prior to the advent of SLAM technology, localization and mapping were treated as distinct entities. However, it was seen that there is a strong internal dependency between mapping and localization. Although accurate localization depends on the map, mapping depends on localization. Thus, the question is known as the “Chicken and Egg” question ([Taheri and Xia, 2021](#)). In robotics, there are different tools to help robots obtain information from surroundings and build their map. One way is to use sensors such as LiDAR, which uses light detection and ranging sensors to make a 3D map ([Huang, 2021](#); [Van Nam and Gon-Woo, 2021](#)). Another way is to use cameras, such as monocular and stereo cameras, which are applied in visual SLAM (V-SLAM). In this method, the robot uses pictures to figure out where it is and creates the required map ([Davison et al., 2007](#)). Regarding the paper’s intensive details, we provide [Table 1](#) that summarizes and includes the description of abbreviations used in the article based on SLAM principles and fundamentals.

**TABLE 1**

List of abbreviations used in this article.

Abbreviation	Explanation	Abbreviation	Explanation
V-SLAM	Visual simultaneous localization and mapping	LSD	Large-scale direct
ROS	Robot Operating System	OKVIS	Open keyframe-based visual-inertial
Lidar	Light detection and ranging	DVO	Dense visual odometry
BA	Bundle adjustment	RPGO	Robust pose-graph optimization
BoW	Bag of words	IMU	Inertial measurement unit
PTAM	Parallel tracking and mapping	GPS	Global positioning system
FAST	Features from accelerated segment test	MAV	Micro air vehicle
ROVIO	Robust visual-inertial odometry	AGV	Automated-guided vehicle
HRI	Human-robot interaction	UAV	Unmanned aerial vehicle
DTAM	Dense tracking and mapping	AR	Augmented reality
LCP	Loop closure process	VR	Virtual reality
SS	Semantic segmentation	RoLi	Range of light intensity
DSt	Dense stereo	ILR	Illumination and light robustness
DSe	Dense semantics	BRIEF	Binary Robust Independent Elementary Features
SCE	Spatial coordinate errors		

Due to the significance of visual techniques in interactive robotic applications, our research focuses on V-SLAM methodologies and their evaluation. V-SLAM can be applied to mobile robotics that utilizes cameras to create a map of their surroundings and easily locate themselves within their work space ([Li et al., 2020](#)). It uses techniques such as computer vision to extract and match visual data for localization and mapping ([Zhang et al., 2020](#); [Chung et al., 2023](#)). It allows robots to map complex environments while performing tasks such as navigation in dynamic fields ([Placed et al., 2023](#); [Khoyani and Amini 2023](#)). It places a strong emphasis on accurate tracking of camera poses and estimating past trajectories of the robot during its work ([Nguyen et al., 2022](#); [Awais and Henrich 2010](#)). [Figure 2](#) provides a basic understanding of V-SLAM. It takes an image from the environment as an input, processes it, and produces a map as an output. In V-SLAM, various types of cameras are used to capture images or videos. A commonly used camera is the monocular camera, which has a single lens, providing 2D visual information ([Civera et al., 2011](#)). However, due to its limitation of lacking depth information, researchers often turn to stereo cameras, which are equipped with two lenses set at a specific distance to capture images from different perspectives, enabling depth details ([Gao et al., 2020](#); [Meng et al., 2018](#)). Another valuable option in V-SLAM is the use of RGB-D cameras, which are capable of capturing both color information (RGB) and depth information (D) ([Meng et al., 2018](#)). Although monocular cameras are inexpensive and lightweight, they may require additional sensors in order to provide accurate data. In contrast, RGB-D and stereo cameras provide depth information. This makes RGB-D, such as Microsoft's Kinect and stereo cameras, suitable for robust and accurate SLAM systems ([Luo et al., 2021](#)).



[FIGURE 2](#)

Schematic representation of a robotic system's architecture, highlighting the incorporation of SLAM and its location within the system.

Previous research demonstrated the effectiveness of V-SLAM methods, but they are often explained with very few details and separate figures ([Khoyani and Amini, 2023](#); [Fan et al., 2020](#)), making it challenging to understand, compare, and make selections among them. As a result, our study focuses on simplifying the explanation of V-SLAM methodologies to enable readers to comprehend them easily. The main contributions of the study can be described as follows:

- Investigation into V-SLAM techniques to determine the most appropriate tools for use in robotics.
- Creation of a graphical and illustrative structural workflow for each method to enhance the comprehension of the operational processes involved in V-SLAM.
- Presentation of significant factors for the evaluation and selection criteria among the V-SLAM methods.

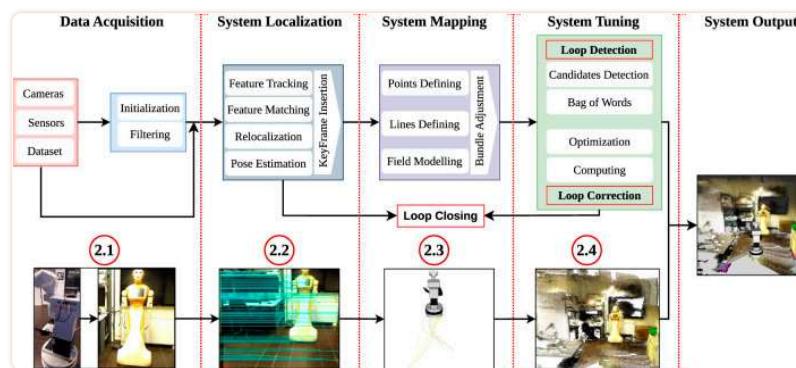
- Compilation of a comparative table that lists essential parameters and features for each V-SLAM method.
- Presentation and discussion of relevant datasets employed within the domain of robotics applications.

## 2 Visual SLAM paradigm

---

As discussed in Introduction, V-SLAM uses sensor data to provide valuable information to the system ([Khoyani and Amini, 2023](#)). Mobile robots and autonomous vehicles require the ability to understand their environment to complete their tasks and achieve their goals ([Ai et al., 2021](#)). This understanding is essential for them to be successful in their operations ([Bongard, 2008](#)).

The V-SLAM framework is composed of sequential steps that are organized to create the system and process its data; see [Figure 3](#), which explains the processes performed within V-SLAM in parallel with the demonstrated pictures. This includes the creation of a detailed map, a trajectory estimator, and the precise positioning and orientation of the cameras attached to that system ([Beghdadi and Mallem, 2022](#); [Kazerouni et al., 2022](#)). Within this framework, various scenarios can be effectively implemented and operated, such as pixel-wise motion segmentation ([Hempel and Al-Hamadi, 2020](#)), semantic segmentation ([Liu and Miura, 2021](#)), and filtering techniques ([Wang et al., 2023](#); [Grisetti et al., 2007](#)). These approaches aim to achieve a professional approach for a visual representation of the processes involved in V-SLAM. The operational framework has been systematically divided into four sections, which can be listed and explained herein.



[FIGURE 3](#)

Visual SLAM architecture: an overview of the four core components necessary for visual SLAM: data acquisition, system localization, system mapping, and system loop closure, and process tuning, enabling mobile robots to perceive, navigate, and interact with their environment.

### 2.1 Data acquisition and system initialization

In this stage of V-SLAM, we systematically prepare input data using system hardware, which includes capturing and preparing images. It involves installing cameras such as RGB-D cameras, depth cameras, or infrared sensors for collecting data and initializing the system ([Beghdadi](#)

[and Mallem, 2022](#)).

The system gathers data, with a particular emphasis on crucial filtering details aimed at effectively eliminating any noise present in the input data ([Mane et al., 2016](#); [Grisetti et al., 2007](#)). The refined data are then sent to the next stage for further processing to extract features from the input information ([Ai et al., 2021](#)). As a result, progress in SLAM methods has resulted in the creation of numerous datasets accessible to researchers to evaluate V-SLAM algorithms ([El Bouazzaoui et al., 2021](#)).

## 2.2 System localization

In the second stage of V-SLAM, the system focuses on finding its location, which is an important part of the entire process ([Scaradozzi et al., 2018](#)). It involves the execution of various processes that are crucial for successfully determining where the robot is. Feature tracking plays a central role during this phase, with a primary focus on tasks such as feature extraction, matching, re-localization, and pose estimation ([Picard et al., 2023](#)). It aims to align and identify the frames that guide the estimation and creation of the initial keyframe for the input data ([Ai et al., 2021](#)). A keyframe is a set of video frames that includes a group of observed feature points and the camera's poses. It plays an important role for the tracking and localization process, helping in eliminating drift errors for camera poses attached to the robot ([Sheng et al., 2019](#); [Hsiao et al., 2017](#)). Subsequently, this keyframe is sent for further processing in the next stage, where it will be shaped into a preliminary map, a crucial part for the third stage of the workflow ([Aloui et al., 2022](#); [Zhang et al., 2020](#)).

## 2.3 System map formation

The third stage of the V-SLAM workflow focuses on the crucial task of building the map, an essential element in V-SLAM processes. Various types of maps can be generated using SLAM, including topological maps, volumetric (3D) maps, such as point cloud and occupancy grid maps, and feature-based or landmark maps. The choice of the map type is based on factors such as the sensors employed, application requirements, environmental assumptions, and the type of dataset used in robotic applications ([Taheri and Xia, 2021](#); [Fernández-Moral et al., 2013](#)). In robotics, a grid map is a representation of a physical environment, with each cell representing a particular location and storing data comprising obstacles, topography, and occupancy. It functions as a fundamental data structure for several robotics navigation and localization techniques ([Grisetti et al., 2007](#)). A feature-based map is a representation which captures the features of the environment, such as landmarks or objects, to facilitate localization and navigation tasks ([Li et al., 2022a](#)). A point cloud map is a representation of a physical space or object made from lots of 3D dots, showing how things are arranged in a place. It is created using special cameras or sensors and helps robots and computers understand what is around them ([Chu et al., 2018](#)).

After setting up keyframes during the localization stage, the workflow progresses to field modeling. Then, key points and feature lines are identified and detected, which is crucial for generating a map ([Schneider et al., 2018](#)). It is a process that builds and updates the map of an unknown environment and is used to continuously track the robot's location ([Chen et al., 2020](#)). It is a two-way process that works together with the localization process, where they depend on each other to achieve SLAM processes. It gathers real-time data about the surroundings, creat-

ing both a geometric and a visual model r13 (accessed on 14 November 2023). In addition, the process includes the implementation of bundle adjustments (BAs) to improve the precision of the generated map before it is moved to the final stage ([Acosta-Amaya et al., 2023](#)). BA is a tool that simultaneously refines the parameters essential for estimating and reconstructing the location of observed points in available images. It plays a crucial role in feature-based SLAM ([Bustos et al., 2019](#); [Eudes et al., 2010](#)).

## 2.4 System loop closure and process tuning

The final stage in the V-SLAM workflow involves fine-tuning the process and closing loops, resulting in the optimization of the final map. In V-SLAM, the loop closure procedure examines and maintains previously visited places, fixing any errors that might have occurred during the robot's exploration within an unknown environment. These errors typically result from the estimation processes performed in earlier stages of the SLAM workflow ([Tsintotas et al., 2022](#); [Hess et al., 2016](#)). Loop closure and process tuning can be done using different techniques, such as the extended Kalman filter SLAM (EKF-SLAM). EKF-SLAM combines loop closure and landmark observation data to adjust the map in the Kalman filter's state estimate. This tool helps address uncertainties in the surrounding world (map) and localize the robot within it ([Song et al., 2021](#); [Ullah et al., 2020](#)).

The bag-of-words (BoW) approach is another technique used to enable robots to recognize and recall previously visited locations. This is similar to how humans remember places they have been to in the past, even after a long time, due to the activities that took place there. BoW works by taking the visual features of each image and converting them into a histogram of visual words. This histogram is then used to create a fixed-size vector representation of the BoW, which is stored for use in matching and loop-closing processes ([Cui et al., 2022](#); [Tsintotas et al., 2022](#)).

Finally, graph optimization is used as a correction tool for loop closure processes. It refines the final map and robot's trajectory by optimizing the graph based on landmarks. This technique involves a graph-based representation of the SLAM issue, where vertices represent robot poses and map characteristics and edges represent constraints or measurements between the poses. It is commonly used as a correction tool in graph-based SLAM types ([Zhang et al., 2017](#); [Chou et al., 2019](#); [Meng et al., 2022](#)).

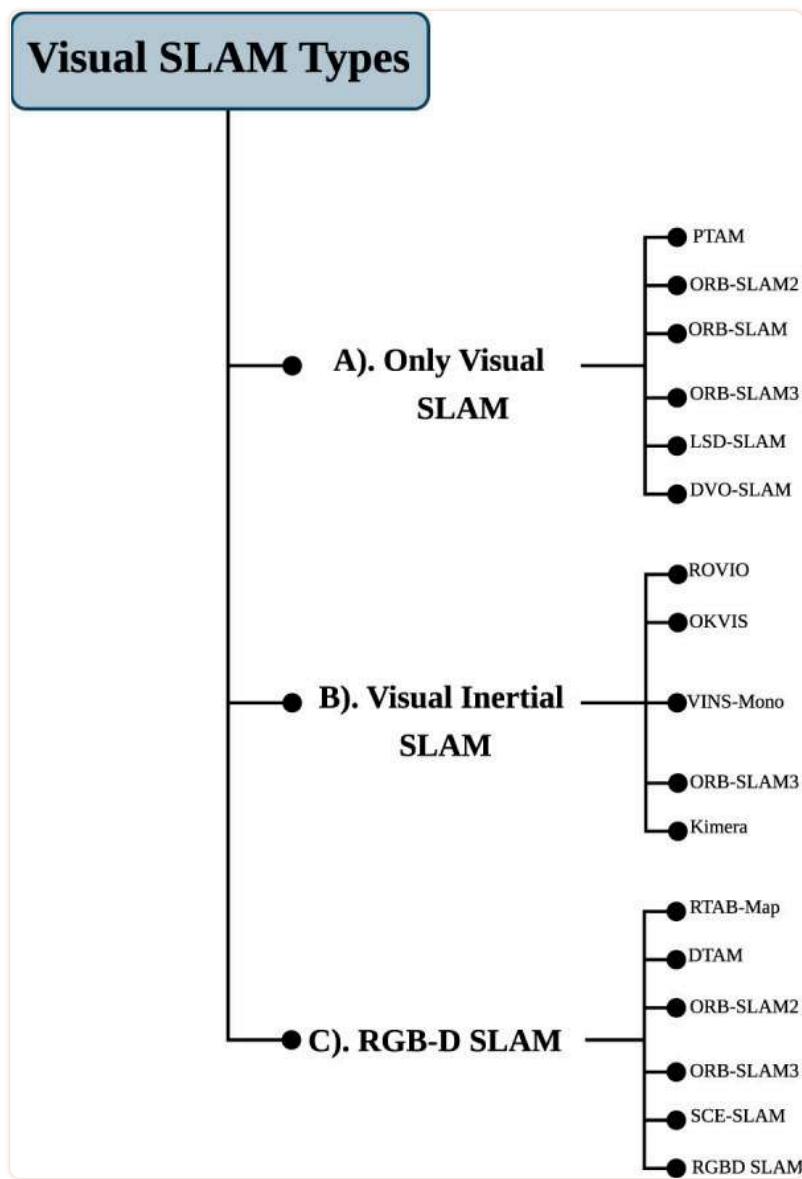
In conclusion, these comprehensive workflow processes outlined in [Sections 2.1, 2.2, 2.3](#), and [2.4](#), respectively, play an important role in V-SLAM for robotics as they facilitate the simultaneous creation of maps and real-time location tracking within the operational environment ([Li et al., 2022b](#)).

## 3 State-of-the-art of visual SLAM methods

---

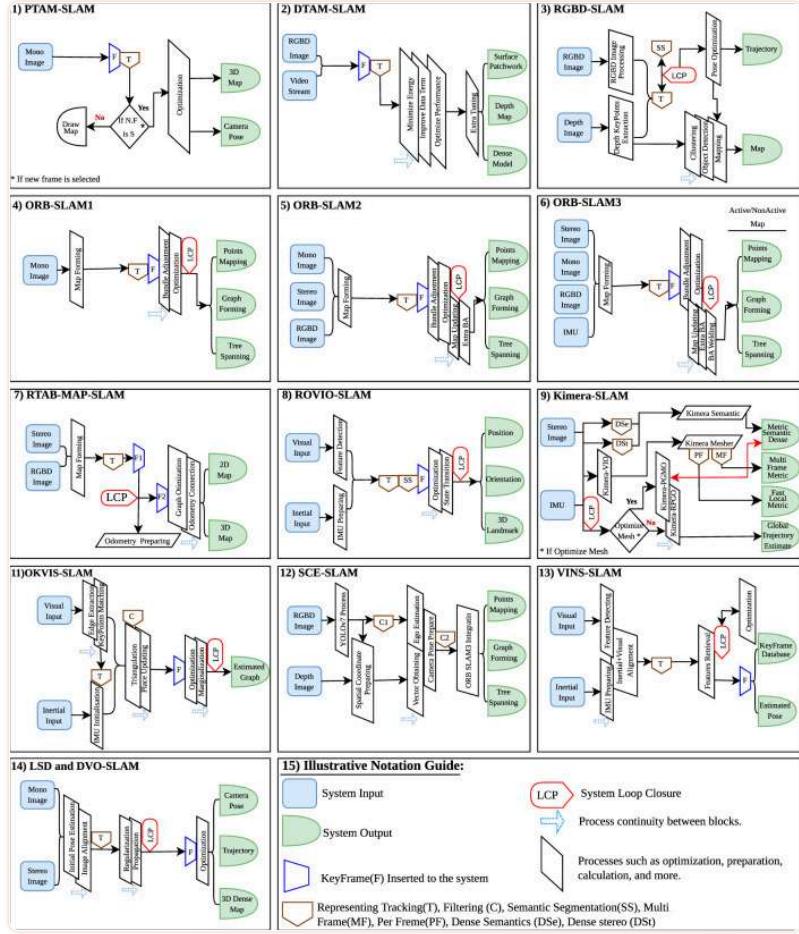
V-SLAM plays a significant role as a transformative topic within the robotics industry and research ([Khoyani and Amini, 2023](#); [Acosta-Amaya et al., 2023](#)). The progress in this field can be attributed to tools such as machine learning, computer vision, deep learning, and state-of-the-art sensor technologies, which have collectively simplified and enhanced its strategy in real-life applications ([Beghdadi and Mallem, 2022](#); [Duan et al., 2019](#)).

The landscape of V-SLAM is composed of a variety of methodologies, which can be divided into three categories, namely, only visual SLAM, visual-inertial SLAM, and RGB-D SLAM ([Macario Barros et al., 2022](#); [Theodorou et al., 2022](#)), as shown in [Figure 4](#). In this section, we provide a brief overview of the current state-of-the-art V-SLAM algorithms and techniques, including their methodology, efficiency, time requirements, and processing capacity, as well as whether they are designed to run on-board or off-board computer systems ([Tourani et al., 2022](#)). Additionally, we combine various graphical representations to create a single and comprehensive visual representation of the method workflow, as shown in [Figure 5](#).



[FIGURE 4](#)

Illustration of visual SLAM types: only-visual SLAM, visual-inertial SLAM, and RGB-D SLAM.



**FIGURE 5**

Visual SLAM methods, illustrating the state-of-the-art method and workflow for select notable SLAM methods featured in this study, presented in a simplified view.

### 3.1 Only visual SLAM

It is a SLAM system designed to map the environment around the sensors while simultaneously determining the precise location and orientation of those sensors within their surroundings. It relies entirely on visual data for estimating sensor motion and reconstructing environmental structures ([Taketomi et al., 2017](#)).

It uses monocular, RGB-D, and stereo cameras to scan the environment, helping robots map unfamiliar areas easily. This approach has attracted attention in the literature because it is cost-effective, easy to calibrate, and has low power consumption in monocular cameras while also allowing depth estimation and high accuracy in RGB-D and stereo cameras ([Macario Barros et al., 2022; Abbad et al., 2023](#)). The methods used in this part can be listed herein.

**3.1.1 PTAM-SLAM** PTAM-SLAM, which stands for parallel tracking and mapping (PTAM), is a monocular SLAM used for real-time tracking systems. It has 6-DoF camera tracking, which can be used in small scenes (K. and Mu. (2007)). This methodology demonstrates remarkable efficiency in dynamic operational settings, consistently providing high performance even in conditions of frequent and unstable lighting variations ([Soliman et al., 2023](#)); see [Table 2](#).

TABLE 2

Comparative scenarios for actively used visual SLAM methods.

SLAM method	Sensor				ILR		Output usage	Application field	RoLI	T2D
	M	S	I	O	W-S					
PTAM K. and Mu	✓	✗	✗	✗	M-H	Pose- estimation 3D mapping	Robotics, AR, and VR	+++	++++	
DTAM Ne. et al	✓	✓	✗	RGBD	S-I	Textured depth map	Robotics, AR, VR, AGV, and simulators	++	+++	
RTAB. M Labbé	✓	✓	✓	Lidar	L-H	2D and 3D mapping	Robotics, VR, AR, and 3D reconstruction	+++	+++	
ORB.S Mur-A	✓	✗	✗	✗	M-H	Tree- spanning and pose- estimating	Robotics mapping indoor navigation	++++	+++	
ORB.S2 Leut et al	✓	✓	✗	RGBD	M-H	Point- mapping and keyframe selection	Mobile mapping, robotics, VR, and UAVs	++++	++++	
ORB.S3 Ca. et al	✓	✓	✓	fish.e	L-H	2D and 3D-Map and tree- spanning	Robotics, security, and 3D reconstruction	+++++	+++++	
RGBD.S End et al	✗	✗	✓	RGBD	L-H	Maps, trajectories and 3D point cloud	3D-scanning, robotics and UAVs	+++	++++	
SCE.S Son et al	✗	✓	✗	RGBD	M-I	Camera pose and Semantic Map	Robotics, AR, and AGV	++++	+++	
OKVIS	✓	✓	✓	✗	M-H	Graph	Robotics.	++++	++++	

-ILR, illumination and light robustness—evaluates how well each SLAM method responds to varying environmental lighting.

-RoLI, range of light intensity—measures the robot's ability to operate effectively across a broad spectrum of light intensities, from very dark to very bright.

-T2D, tolerance to directionality—assesses the robot's capability to function in environments with strong directional light sources, such as spotlights and windows.

-W-S, defines the operational scale and application field of the robot (M, medium; L, large; S, small, H, hybrid, I, indoor).

-S,M, sources and materials—provides links to the source codes used in the method.

-VINS,M,S, VINS-Mono SLAM; M, monocular camera; S, stereo camera; IMU, inertial measurement unit; O, other sensors; fish,e, fish-eye camera; rgbd, RGB-D camera.

The system workflow consists of four sequential stages ([Klien and Murray, 2007](#); [Fernández-Moral et al., 2013](#)). Input preparation and system initialization involve processes such as monocular camera translation and rotation to improve image efficiency and clarity ([De Croce et al., 2019](#)). The tracking process is carried out, where tasks related to image and video processing are performed to prepare data for subsequent mapping procedures. Following that, the optimization and mapping processes are carried out to prepare the map and reveal the outputs, which include the camera pose and the 3D map used in SLAM operations ([Klien and Murray, 2007](#); [Servières et al., 2021](#)). All processes and steps are simplified and demonstrated in [Figure 5](#), part 1.

**3.1.2 ORB-SLAM** ORB-SLAM stands for oriented FAST (features from accelerated segment test) and rotated BRIEF (binary robust independent elementary features) SLAM ([Tourani et al., 2022](#)). This feature-based detector is applicable in both small and large indoor or outdoor fields ([Tourani et al., 2022](#)). Due to its real-time capabilities and high-quality map reconstruction, it is widely used in applications such as the human–robot interaction (HRI) ([Mur-Artal et al., 2015](#)), augmented reality, and autonomous navigation ([Zhu et al., 2022](#); [Yang et al., 2022](#)). ORB-SLAM is designed to handle robust and unstable motion clutter, covering essential processes such as tracking, mapping, and loop closing ([Campos et al., 2021](#)). Compared to other advanced V-SLAM methods, ORB-SLAM outperforms by enhancing the dynamic, size, and traceability of the map. It achieves real-time global localization from wide baselines, performs camera re-localization from various viewpoints, and makes better selections for frames and points in the reconstruction process ([Ragot et al., 2019](#); [Mur-A and Tars, 2014](#)); see [Table 2](#).

ORB-SLAM1 categorized to be only-visual ([Mur-Artal et al., 2015](#); [Mur-A and Tars, 2014](#)), while ORB-SLAM2 expands to both only-visual and RGB-D SLAM ([Ragot et al., 2019](#); [Mur-Artal and Tardós 2017a](#)). Furthermore, ORB-SLAM3 furthers its classification to include all three categories: only-visual, visual-inertial, and RGB-D SLAM. This expansion underscores the adaptability and versatility of ORB-SLAM in real-life applications ([Zang et al., 2023](#); [Ca et al., 2021](#); [Campos et al., 2021](#)).

The ORB-SLAM methodology process goes through four sequential phases ([Mur-Artal et al., 2015](#); [Mur-Artal and Tardós 2017a](#); [Ca et al., 2021](#)). The initial phase involves the sensor input and the tracking process ([Joo et al., 2020](#)). Across all ORB-SLAM versions, this phase shares a common approach, focusing on pose preparation and frame generation to facilitate decision-making ([Sun et al., 2017](#)). However, the difference lies in input usage; for example, ORB-SLAM1 uses one input, ORB-SLAM2 uses three, and ORB-SLAM3 uses four ([Campos et al., 2021](#)). Therefore, the quality and efficiency of the next operation depend on the input in the first stage. In the next phase, local mapping is done by adding new keyframes and creating map points with the localization process simultaneously ([Ca et al., 2021](#)). This part remains consis-

tent across all versions, but version 3 enhances its functionality by incorporating additional bundle adjustment for improved feature detection and matching ([Dai et al., 2021](#)). The subsequent phase involves loop closing, process optimization, and selecting similar candidate data in all versions. However, versions 2 and 3 include additional steps such as bundle adjustment welding and map merging ([Mur-Artal and Tardós, 2017a](#); [Zang et al., 2023](#)). The last stage is preparing the output, focusing on creating the final map that includes essential information such as graphs, lines, point mapping, and 2D and 3D maps for use in the SLAM process ([Acosta-Amaya et al., 2023](#)). [Figure 5](#) parts 4, 5, and 6 give a detailed observation about the methods of ORB-SLAM 1, 2, and 3 versions, respectively, showcasing their features and functionalities for a better understanding.

**3.1.3 LSD-SLAM** LSD-SLAM, which stands for large-scale direct monocular SLAM, is an advanced technique made for real-time mapping and positioning. It can utilize various camera setups. It is designed for large-scale mapping jobs where it can create a very accurate and detailed map of the working fields. In addition, it stays accurate even with a lower image resolution ([Engel et al., 2015](#); [Fernández-Moral et al., 2013](#)). This flexibility makes it a better choice for operating in complex, wide-ranging and dynamic environments and is used in various applications such as robotics and self-driving cars ([Mur-Artal et al., 2015](#); [Eng et al., 2014](#)); see [Table 2](#).

LSD-SLAM distinguishes itself from the DTAM-SLAM approach by focusing on areas with strong intensity changes, leaving out regions with little or no texture details. This choice comes from the challenge of figuring out how far things are in areas where there is not much texture inside images. As a result, LSD-SLAM goes beyond what DTAM can do by concentrating on places with strong changes in brightness and ignoring areas with very little texture ([Acosta-Amaya et al., 2023](#); [Khoyani and Amini, 2023](#)).

LSD and DVO-SLAM processes can function similarly, and their workflow is structured in five stages ([Macario Barros et al., 2022](#); [Luo et al., 2021](#); [Schöps et al., 2014](#); [Engel et al., 2015](#)). The first stage includes inputting mono- and stereo data and preparing them for the next processing step. The second stage is designed for tracking and estimating the initial pose by aligning images from both mono and stereo cameras. The third stage is dedicated to loop closure processes, involving keyframe preparation, regularization, and data updates to prepare frames for subsequent stages. The fourth stage carries out map optimization, including two critical phases, which are direct mapping and feature-based mapping. It also covers processes such as activation, marginalization, and direct bundle adjustment. These operations shape the necessary map and manage its pointsassesses their performance under varyin with semi-dense adjustments for use in the output stage. In the final stage, the estimated camera trajectory and pose with the dense 3D map are prepared for application in robotics' SLAM functions; see [Figure 5](#), part 14 for a detailed workflow.

**3.1.4 DVO-SLAM** DVO-SLAM, which stands for dense visual odometry SLAM, is designed to facilitate real-time motion estimation and map creation using depth-sensing devices, such as stereo and mono cameras ([Schöps et al., 2014](#)). It stands out for its ability to generate detailed and accurate environment maps while tracking the position and orientation ([Luo et al., 2021](#); [Zhu et al., 2022](#)). DVO-SLAM uses point-to-plane metrics in photo metric bundle adjustment (PBA), enhancing the navigation of robotic systems, especially in situations with less textured points. The point-to-plane metric is a cost function and optimization tool that is used to optimize the depth sensor poses and plane parameters for 3D reconstruction ([Alismail et al., 2017](#);

[Zhou et al., 2020](#); [Newcombe et al., 2011](#)). These features make DVO-SLAM suitable for more accurate applications such as in robotics and augmented reality (AR), and it is robust for operating in slightly unstable light sources ([Khoyani and Amini, 2023](#); [Kerl et al., 2013](#)); see [Table 2](#).

### 3.2 Visual-inertial SLAM

VI-SLAM is a technique that combines the capabilities of visual sensors, such as stereo cameras, and inertial measurement sensors (IMUs) to achieve its SLAM objectives and operations ([Servières et al., 2021](#); [Leut et al., 2015](#)). This hybrid approach allows a comprehensive modeling of the environment, where robots operate ([Zhang et al., 2023](#)). It can be applied to various real-world applications, such as drones and mobile robotics ([Taketomi et al., 2017](#)). The integration of IMU data enhances and augments the information available for environment modeling, resulting in improved accuracy and reduced errors within the system's functioning ([Macario Barros et al., 2022](#); [Mur-Artal and Tardós 2017b](#)). The methods and algorithms used in this approach, while implemented in real-life applications, can be listed as shown in the following section.

**3.2.1 OKVIS-SLAM** OKVIS-SLAM, which stands for open keyframe-based visual-inertial SLAM, is designed for robotics and computer vision applications that require real-time 3D reconstruction, object tracking, and position estimation ([Kasyanov et al., 2017](#)). It combines visual and inertial measurements to accurately predict the position and orientation of a robot simultaneously ([Leut et al., 2015](#)).

It accurately tracks the camera's position and orientation in real-time control during a robot's motion ([Leutenegger, 2022](#)). It uses image retrieval to connect keyframes in the SLAM pose-graph, aided by the pose estimator for locations beyond the optimization window of visual-inertial odometry ([Kasyanov et al., 2017](#); [Wang et al., 2023](#)). For portability, a lightweight semantic segmentation CNN is used to remove dynamic objects during navigation ([Leutenegger, 2022](#)). OKVIS's real-time precision and resilience make it suitable for various applications, including robotics and unmanned aerial vehicles (UAVs). It can operate effectively in complex and unstable illumination environments ([Wang et al., 2023](#)); see [Table 2](#).

We have structured the OKVIS-SLAM workflow into three key phases ([Leutenegger, 2022](#); [Kasyanov et al., 2017](#); [Wang et al., 2023](#)). The first phase focuses on receiving initial sensor inputs, including IMU and visual data. It initializes the system, conducts IMU integration, and employs tracking techniques to prepare the data for subsequent processing. The second phase is the real-time estimator and odometry filtering phase, covering various operations, such as landmark triangulation and status updating. The triangulation process is used for estimation used to generate the 3D position of visual landmarks to enhance SLAM operation ([Yousif et al., 2015](#)). In the last phase, optimization and full graph estimation are performed. This includes loop closure detection, window sliding, and marginalization. The phase selects relevant frames and optimizes the overall graph structure, ultimately providing essential outputs for the SLAM system; see [Figure 5](#), part 11.

**3.2.2 ROVIO-SLAM** ROVIO-SLAM, which stands for robust visual-inertial odometry SLAM, is a cutting-edge sensor fusion method that smoothly combines visual and inertial data. This integration significantly enhances navigation accuracy, leading to improved work efficiency in robotics systems ([Blo et al., 2015](#); [Wang et al., 2023](#)). It brings valuable attributes for robotics, ex-

celling in robust performance in challenging environments, and presents a smooth interaction between the robot and its surroundings ([Li et al., 2023a](#)). It efficiently handles extensive mapping processes, making it suitable for large-scale applications ([Kasyanov et al., 2017](#)). Moreover, it operates with low computational demands and high robustness to light, making it ideal for cost-effective robotic platforms designed for sustained, long-term operations ([Leutenegger, 2022](#)).

ROVIO-SLAM workflow is divided into three stages ([Picard et al., 2023](#); [Nguyen et al., 2020](#); [Schneider et al., 2018](#)). First, data from visual cameras and IMU are obtained and prepared for processing. In the next stage, feature detection, tracking, and semantic segmentation are done for visual data, while IMU data are prepared for integration from the other side. The processing stage involves loop closure operations, new keyframes insertion, and state transition, along with data filtering. State transitions lead to the generation of the key output, which is then transferred to the final stage, providing estimated position, orientation, and 3D landmarks; see [Figure 5](#), part 8.

**3.2.3 VINS Mono-SLAM** VINS Mono-SLAM, which stands for the visual-inertial navigation system, is an advanced sensor fusion technology that precisely tracks the motion and position of a robot or sensor in real-time. Utilizing only a single camera and an IMU, it combines visual and inertial data to enhance accuracy and ensure precise functionality of robot operations ([Mur-Artal and Tardós, 2017b](#)). Known for its efficiency in creating maps and minimizing drift errors, VINS-Mono excels in navigating challenging environments with dynamic obstacles ([Bruno and Colombini, 2021](#)). Its smooth performance in difficult lighting conditions highlights its reliability, ensuring optimal functionality for mobile robots operating in unstable lighting conditions ([Song et al., 2022](#); [Kuang et al., 2022](#)). This power-efficient, real-time monocular VIO method is suitable for visual SLAM applications in robotics, virtual reality, and augmented reality ([Gu et al., 2022](#)); see [Table 2](#).

The VINS-Mono SLAM workflow is organized into four stages ([Qin et al., 2018](#); [Xu et al., 2021](#)). In the first stage, we gathered visual and inertial data and prepared them for acquisition and measurement processing, including feature extraction, matching, and IMU data preparation, and sent them for visual and inertial alignment. The second stage handles loop closure operations and re-localization to adjust old states with additional feature retrieval for the next step. The third stage focuses on process optimization, incorporating bundle adjustments and additional propagation for efficiency. The final stage outputs the system's estimated pose and a keyframe database, applicable to SLAM; see [Figure 5](#), part 13.

**3.2.4 Kimera-SLAM** Kimera-SLAM is an open-source SLAM technique applied for real-time metric semantic purposes. Its framework is highly dependent on previous methodologies such as ORB-SLAM, VINS-Mono SLAM, OKVIS, and ROVIO-SLAM ([Ros. et al., 2020](#)). Exhibiting robustness in dynamic scenes, particularly in the presence of moving objects ([Wang et al., 2022](#)), Kimera-SLAM showcases resilience to variations in lighting conditions. It operates effectively in both indoor and outdoor settings, making it highly compatible with integration into interactive robotic systems ([Rosinol et al., 2021](#)). In summary, Kimera-SLAM provides a thorough and efficient solution for real-time metric-semantic SLAM, prioritizing accuracy, modality, and robustness in its operations ([Rosinol et al., 2021](#)); see [Table 2](#).

The procedural workflow of this technique can be summarized in five stages (Ros et al. (2020)). First, the input pre-processing includes dense 2D semantics, dense stereo, and Kimera-VIO. It also includes front-end and back-end operations such as tracking, feature extraction, and matching, which yield an accurate state estimation. The second stage involves robust pose graph optimization (Kimera-RPGO), tasked with optimization and the formulation of a global trajectory. Subsequently, the third stage features the per-frame and multi-frame 3D mesh generator (Kimera-Mesher), responsible for the execution and generation of 3D meshes representing the environment. The fourth stage introduces semantically annotated 3D meshes (Kimera-Semantics), dedicated to generating 3D meshes with semantic annotations. This stage sets the groundwork for the subsequent and final stage, where the generated 3D meshes are utilized for output visualization, ultimately serving SLAM purposes, as illustrated in [Figure 5](#), part 9.

### 3.3 RGB-D SLAM

RGB-D is an innovative approach that integrates RGB-D cameras with depth sensors to estimate and build models of the environment ([Ji et al., 2021](#); [Macario Barros et al., 2022](#)). This technique has found applications in various domains, including robotic navigation and perception ([Luo et al., 2021](#)). It demonstrates efficient performance, particularly in well-lit indoor environments, providing valuable insights into the spatial landscape ([Dai et al., 2021](#)).

The incorporation of RGB-D cameras and depth sensors enables the system to capture both color and depth information simultaneously. This capability is advantageous in indoor applications, addressing the challenge of dense reconstruction in areas with low-textured surfaces ([Zhang et al., 2021b](#)). The objective of RGB-D SLAM is to generate a precise 3D reconstruction for the system surroundings, with a focus on the acquisition of geometric data to build a comprehensive 3D model ([Chang et al., 2023](#)). The methods used in this section are listed as follows:

#### 3.3.1 RTAB-Map SLAM

RTAB-Map SLAM, which stands for real-time appearance-based mapping, is a visual SLAM technique that works with RGB-D and stereo cameras ([Ragot et al., 2019](#)). It is a versatile algorithm that can handle 2D and 3D mapping tasks depending on the sensor and data that are given ([Peter et al., 2023](#); [Acosta-Amaya et al., 2023](#)). It integrates RGB-D and stereo data for 3D mapping, enabling the detection of static and dynamic 3D objects in the robot's environment ([Ragot et al., 2019](#)). It is applicable in large outdoor environments where LiDAR rays cannot reflect and manage the field around the robot ([Gurel, 2018](#)). Variable lighting and environmental interactions can cause robotic localization and mapping errors. Therefore, RTAB's robustness and adaptability to changing illumination and scenes enable accurate operation in challenging environments. It can handle large, complex environments and is quickly adaptable to work with multiple cameras or laser rangefinders ([Li et al., 2018](#); [Peter et al., 2023](#)). Additionally, the integration of T265 (Intel RealSense Camera) and implementation of ultra-wideband (UWB) ([Lin and Yeh, 2022](#)) address robot wheel slippage with drifting error handling, enhancing system efficiency with precise tracking and 3D point cloud generation, as done in [Persson et al. \(2023\)](#); see [Table 2](#).

The RTAB-MAP SLAM method involves a series of steps that enable it to function ([Gurel, 2018](#); [Labbé and Michaud, 2019](#)). Initially, the hardware and front-end stage is responsible for tasks such as obtaining data from stereo and RGB-D cameras, generating frames, and integrating sensors. This stage prepares the frames that will be used in the subsequent stage. After the frames have been processed simultaneously with the tracking process, the loop closure is activated to generate the necessary odometry. Subsequently, the keyframes equalization and optimization processes are initiated to improve the quality of the 2D and 3D maps generated for SLAM applications, as shown in [Figure 5](#), part 7.

**3.3.2 DTAM-SLAM** DTAM-SLAM, which stands for dense tracking and mapping, is a V-SLAM algorithm specified for real-time camera tracking. It provides robust six degrees of freedom (6 DoF) tracking and facilitates efficient environmental modeling for robotic systems ([Ne et al., 2011](#); [Macario Barros et al., 2022](#)). This approach plays a fundamental role in advancing applications such as robotics, augmented reality, and autonomous navigation, delivering precise tracking and high-quality map reconstruction. Furthermore, it is slightly dynamic with light; thus, it is accurate to operate in high and strong illumination fields ([Zhu et al., 2022](#); [Yang et al., 2022](#)); see [Table 2](#).

The DTAM-SLAM workflow is divided into a series of steps, each with its own purpose ([Ne et al., 2011](#); [Macario Barros et al., 2022](#)). It begins with the input such as the RGB-D camera, which helps initialize the system work. In the camera tracking and reconstruction stage, the system selects frames and estimates textures on the image. It then accurately tracks the 6DoF camera motion, determining its exact position and orientation. Furthermore, the optimization framework is activated and uses techniques such as spatially regularized energy minimization to enhance data terms, thereby improving the image quality that is captured from video streaming. As a result, the advanced process tuning carries out operations that improve the method's performance and producing precise outputs such as dense models, surface patchwork, and texture depth maps (see [Figure 5](#), part 2).

**3.3.3 RGBD-SLAM** RGDB-SLAM, which stands for simultaneous localization and mapping using red–green–blue and depth data, is an important method that creates a comprehensive 3D map containing both static and dynamic elements ([Ji et al., 2021](#)). This method involves the tracking of trajectories and mapping of points associated with moving objects ([Steinbrücker et al., 2011](#); [Niu et al., 2019](#)). Using these data types enhances and provides precise SLAM results ([End et al., 2012](#); [Li Q. et al., 2022a](#)). It has the ability to create registered point clouds or OctoMaps for the purpose that can be used for robotic systems ([Zhang and Li 2023](#); [Ren et al., 2022](#)). In robotics applications, RGB-D SLAM, specifically V-SLAM, excels in both robustness and accuracy. It effectively addresses challenges such as working in a dynamic environment ([Steinbrücker et al., 2011](#); [Niu et al., 2019](#)). The implementation of RGB-D SLAM faced a challenge in balancing segmentation accuracy, system load, and the number of detected classes from images. This challenge was tackled using TensorRT, optimized by YOLOX for high-precision real-time object recognition ([Chang et al., 2023](#); [Martínez-Otzeta et al., 2022](#)). It has versatile applications in real-world robotics scenarios, including autonomous driving cars, mobile robotics, and augmented reality ([Zhang and Li, 2023](#); [Bahraini et al., 2018](#)); see [Table 2](#).

The RGB-D SLAM workflow can be organized into five essential stages, each playing a crucial role in the SLAM process ([Ji et al., 2021](#); [Hastürk and Erkmen, 2021](#); [End et al., 2012](#)). The initial stage involves data acquisition, where RGB-D and depth camera data are collected as the foundational input for subsequent stages. Moving on to the second stage, processing of RGB-D

details was activated. During this phase, tasks include feature extraction and pairwise matching while simultaneously addressing depth-related activities, such as storing point clouds, and aligning lines or shapes. In the third stage, activities such as noise removal and semantic segmentation (SS), in addition to loop closure detection, are performed to lay the groundwork for map construction. The fourth stage is dedicated to focus on pose estimation and optimization techniques, leading to improvement in the accuracy of the system output. The final stage involves generating trajectory estimation and maps, refining the outputs for use in SLAM applications in robotic systems; see [Figure 5](#), part 3.

**3.3.4 SCE-SLAM** SCE-SLAM, which stands for spatial coordinate errors SLAM, represents an innovative real-time semantic RGB-D SLAM technique. It has been developed to tackle the constraints posed by traditional SLAM systems when operating in dynamic environments ([Li et al., 2020](#)). The method was improved to increase the performance of existing V-SLAM methods such as ORB-SLAM3 and makes it useful with greater accuracy and robustness in dynamic situations with the help of merging semantic and geometric data and leveraging YOLOv7 for quick object recognition ([Wu et al., 2022](#)). Thanks to these improvements, the SLAM algorithms can be well-suited for dynamic scenarios which allows in greater adaptability and comprehension of system surroundings. This enables robotic systems to operate in more complex circumstances with the fewer mistakes or slippage errors ([Liu and Miura, 2021](#)). Moreover, robots equipped with SCE-SLAM are empowered to operate in a more flexible and error-reduced manner, and it can operate in challenging light environments ([Son et al., 2023; Ren et al., 2022](#)); see [Table 2](#).

The SCE-SLAM workflow is divided into three key stages ([Son et al., 2023](#)). The first stage involves the semantic module. This module processes camera input data and employs Yolov2 to remove noise from the input. The second stage is the geometry module, where depth image analysis and spatial coordinate recovery are performed, preparing the system for integration with ORB-SLAM3. The final stage is dedicated to the integration of ORB-SLAM3. This integration facilitates the execution of processes within ORB-SLAM3. The process works in parallel with the loop closure technique, which results in a more accurate and precise system output; see [Figure 5](#), Part 12.

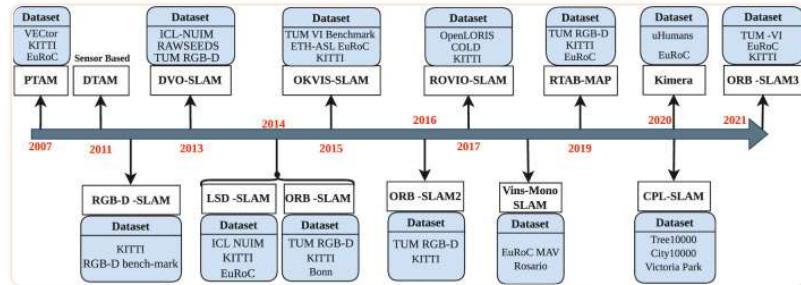
## 4 Visual SLAM evolution and datasets

---

The roots of SLAM can be traced back to nearly three decades ago, when it was first introduced by Smith et al. ([Picard et al. \(2023\); Khoyani and Amini \(2023\)](#)). Recently, visual SLAM has changed a lot and made a big impact on robotics and computer vision ([Khoyani and Amini, 2023](#)). Along this journey, different V-SLAM methods have been created to tackle specific challenges in robot navigation, mapping, and understanding the surroundings ([Aloui et al., 2022; Sun et al., 2017](#)). To verify and compare these V-SLAM methods, important datasets have been created which played a crucial role in the field ([Pal et al., 2022; Tian et al., 2023a](#)). In this section, we explore the evolution of V-SLAM methods over time and how they have advanced with the help of using the suitable datasets.

To offer a more comprehensible perspective, we provide an illustrative timeline depicting the evolution of the most well-known V-SLAM methods, as shown in [Figure 6](#). This graphical representation illustrates the development of the V-SLAM methodologies from 2007 to 2021. These methods have been applied in various fields, including agriculture, healthcare, and industrial

sectors, with a specific focus on interactive mobile robots. Additionally, we highlight several significant and widely recognized benchmark datasets crucial to V-SLAM, as shown in the following section.



**FIGURE 6**

Timeline illustrates the evolutionary journey of SLAM techniques, accompanied by the datasets that have played a pivotal role in their development. It showcases the dynamic progression of SLAM technologies over time, reflecting the symbiotic relationship between innovative methods and the rich variety of datasets they have been tested and refined with.

#### 4.1 TUM RGB-D dataset

The TUM RGB-D dataset is a widely used resource in the field of V-SLAM, which helps demonstrate the effectiveness and practicality of V-SLAM techniques. This dataset provides both RGB images and depth maps, with the RGB images saved in a  $640 \times 480$  8-bit format and the depth maps in a  $640 \times 480$  16-bit monochrome ([Chu et al., 2018](#)). It offers RGB-D data, making it appropriate for both depth-based and V-SLAM techniques. Its usefulness extends to essential tasks such as mapping and odometry, providing researchers with a considerable volume of data for testing SLAM algorithms across diverse robotic applications ([Ji et al., 2021](#); [End et al., 2012](#)). The adaptability of these datasets is remarkable, as they find application in mobile robotics and handheld platforms, demonstrating effectiveness in both indoor and outdoor environments ([Martínez-Otzeta et al., 2022](#); [Son et al., 2023](#)).

Some of the recent studies used TUM datasets, such as in [Li et al. \(2023c\)](#). They have leveraged the TUM RGB-D dataset to establish benchmarks customized to their specific research objectives. The study initiated its investigations with RGB-D images and ground truth poses provided by the TUM datasets, utilizing them to construct 3D scenes characterized with real space features. The integrative role assumed by the TUM RGB-D dataset in this context attains profound significance as a fundamental resource within the domain of V-SLAM research. For more details, refer to the TUM RGB-D SLAM dataset.

#### 4.2 EuRoC MAV benchmark dataset

The EuRoC MAV benchmark dataset is specifically designed for micro aerial vehicles (MAVs) and contributes a valuable resource in the domain of MAV-SLAM research since it includes sensor data such as IMU and visual data such as stereo images. These datasets, published in early 2016, are made accessible for research purposes and offer a diverse usability in indoor and

outdoor applications. Consequently, it serves as a relevant choice for evaluating MAV navigation and mapping algorithms, particularly in conjunction with various visual V-SLAM methodologies ([Sharafutdinov et al., 2023](#); [Leutenegger, 2022](#); [Burri et al., 2016](#)).

The EuRoC MAV benchmark dataset, of notable benefits to robotics, is particularly valuable for researchers working on visual-inertial localization algorithms like OpenVINS ([Geneva et al., 2020](#); [Sumikura et al., 2019](#)) and ORB-SLAM2 ([Mur-Artal and Tardós, 2017a](#)). This dataset incorporates synchronized stereo images, IMU measurements, and precise ground truth data, providing comprehensive resources for algorithm development. Its comprehensive data structure makes it highly suitable for thoroughly testing and validating algorithms tailored for MAV purposes ([Burri et al., 2016](#)). For more details, refer to the EuRoC MAV dataset.

#### 4.3 KITTI dataset

The KITTI dataset is a widely utilized resource in robotics navigation and SLAM, with a particular emphasis on V-SLAM. Designed for outdoor SLAM applications in urban environments, KITTI integrates data from multiple sensors, including depth cameras, lidar, GPS, and inertial measurement unit (IMU), contributing to the delivery of precise results for robotic applications ([Geiger et al., 2013](#)). Its versatility extends to supporting diverse research objectives such as 3D object detection, semantic segmentation, moving object detection, visual odometry, and road-detection algorithms ([Wang et al., 2023](#); [Raikwar et al., 2023](#)).

As a valuable asset, researchers routinely rely on the KITTI dataset to evaluate the effectiveness of V-SLAM techniques in real-time tracking scenarios. In addition, it serves as an essential tool for researchers and developers engaged in the domains of self-driving cars and mobile robotics ([Geiger et al., 2012](#); [Ortega-Gomez et al., 2023](#)). Furthermore, its adaptability facilitates the evaluation of sensor configurations, thereby contributing to the refinement and assessment of algorithms crucial to these fields ([Geiger et al. \(2013\)](#)). For more details, refer to the KITTI Vision Benchmark Suite.

#### 4.4 Bonn RGB-D dynamic dataset

The Bonn dataset is purposefully designed for RGB-D SLAM, containing dynamic sequences of objects. It showcases RGB-D data accompanied by a 3D point cloud representing the dynamic environment, which has the same format as TUM RGB-D datasets ([Palazzolo et al., 2019](#)). It covers both indoor and outdoor scenarios, extending beyond the boundaries of controlled environments. It proves valuable for developing and evaluating algorithms related to tasks such as robot navigation, object recognition, and scene understanding. Significantly, this dataset is versatile enough to address the complexities of applications used in light-challenging areas ([Soares et al., 2021](#); [Ji et al., 2021](#)). In addition, it proves to be an important resource for evaluating V-SLAM techniques characterized by high dynamism and crowds where the robot might face the challenge of object detection and interaction with the surrounding environment ([Dai et al., 2021](#); [Yan et al., 2022](#)). For more details, refer to the Bonn RGB-D dynamic dataset.

#### 4.5 ICL-NUIM dataset

It is a benchmark dataset which is designed for RGB-D applications, serving as a valuable tool for evaluating RGB-D, visual odometry, and V-SLAM algorithms, particularly in indoor situations ([Handa et al., 2014](#)). It includes 3D sensor data and ground truth poses, facilitating the benchmarking of techniques related to mapping, localization, and object detection in the domain of robotic systems. Its pre-rendered sequences, scripts for generating test data, and standardized data formats are beneficial for researchers in evaluating and improving their SLAM algorithms ([Chen et al., 2020](#)). A unique aspect of the ICL-NUIM dataset is its inclusion of a three-dimensional model. This feature empowers researchers to explore and devise new scenarios for robotic systems, which operates in unknown environments. Moreover, it promotes improvements in V-SLAM, which makes it possible to generate semantic maps that improve robots' flexibility and adaptability to integration into that environment easily and flexibly ([Zhang et al., 2021a](#)). For more details, refer to the ICL-NUIM dataset.

## 5 Guidelines for evaluating and selecting visual SLAM methods

---

Choosing the right visual SLAM algorithm is crucial for building an effective SLAM system. With the continuous advancements in V-SLAM methodologies responding to diverse challenges, it is essential to navigate structured criteria to deploy and implement precise solutions ([Placed et al., 2023](#); [Sousa et al., 2023](#)). In the context of robotic systems, we provide important parameters. We outline them by offering concise explanations of the selection criteria that guide how to choose suitable SLAM methods for field applications. These parameters are listed below.

### 5.1 Robustness and accuracy

When choosing among V-SLAM methods, a key consideration is the robustness and accuracy of the method ([Zhu et al., 2022](#)). In particular, a robust algorithm can handle sensor noise, obstacles, and changing environments to ensure continuous and reliable operation ([Bongard, 2008](#)). Additionally, accuracy is equally important for creating precise maps and localization, allowing the robot to make informed decisions and move through the environment without errors ([Kucner et al., 2023](#); [Nakamura et al., 2023](#)). These qualities collectively enhance the algorithm's reliability in challenging real-world situations, making them crucial factors for successful mobile robotic applications.

### 5.2 Computational efficiency and real-time requirements

In the application of mobile robotics, the selection of the SLAM algorithm is extremely important, focusing on the efficiency of the process happening inside the robot's computational architecture ([Macario Barros et al., 2022](#)). Therefore, the chosen V-SLAM algorithm must be carefully tailored to meet the computational demands imposed by the real-time constraints of the robot. This entails a delicate balancing act as the selected algorithm should be seamlessly integrated with the available processing power and hardware resources, all while satisfying the stringent real-time requirements of the application. The critical consideration for this step is the quality of the sensors, the professors, and/or computers so that they can generate a quick response and accurate localization in a very limited time ([Henein et al., 2020](#)).

### 5.3 Flexible hardware integration

In robotic applications, it is important for researchers to choose a SLAM algorithm that works well with the robot's sensors. Integrating suitable hardware improves speed and performance in SLAM systems through accelerators, method optimization, and energy-efficient designs ([Eyyazpour et al., 2023](#)). Various V-SLAM algorithms are designed for specific sensor types such as RGB-D, lidar, and stereo cameras. This facilitates seamless integration into the SLAM system, enhancing the functionality of utilizing integrated hardware ([Wang et al., 2022](#)). Moreover, the availability of ROS packages and open-source software for sensors and cameras provides increased modality and flexibility during system installation. This, in turn, enhances adaptability and makes integration easy and free of challenges ([Sharafutdinov et al., 2023](#); [Roch et al., 2023](#)). For example, the OAK-D Camera, also known as the OpenCV AI Kit, is a smart camera that is great for indoor use. It can automatically process data files and use neural reasoning right inside the camera, without needing extra computer power from the robot. This means it can run neural network models without making the robot's operating system work harder ([Han et al., 2023](#)).

#### 5.4 System scalability

In SLAM algorithms for robotics, scalability is a vital factor to keep in mind during the design of the system Middleware architecture. It enables rapid situational awareness over large areas, supports flexible dense metric-semantic SLAM in multi-robot systems, and facilitates fast map learning in unknown environments ([Castro, 2021](#)). This parameter needs to evaluate the algorithm's capability to adjust to different mapping sizes and environmental conditions, particularly considering light emission, video, and/or image clarity. It should also provide versatility for various application needs, applicable to both indoor and outdoor scenarios ([Laidlow et al., 2019](#); [Zhang et al., 2023](#)).

#### 5.5 Adapting to dynamic environments

The ability of a SLAM algorithm to handle dynamic objects in the environment is an important consideration for robotics systems. This parameter assesses the algorithm's ability to detect, track, and incorporate dynamic objects and moving obstacles into the mapping process ([Lopez et al., 2020](#)). It focuses on the algorithm's capability to enable the robot to handle these objects effectively and respond quickly during the ongoing SLAM process ([Wu et al., 2022](#)). A robust dynamic environment should ensure the algorithm's ability to adapt and respond in real-time applications. This is crucial for systems operating in environments where changes occur instantaneously, such as in interactive robotics applications ([Li et al., 2018](#)).

#### 5.6 Open-source availability and community support

When choosing a SLAM algorithm for our project, it is important to observe whether it is open-source and has a community of active users. It is important because it makes it easier to customize and adapt the system according to our needs, benefiting from the experiences of the user community ([Khoyani and Amini 2023](#); [Xiao et al., 2019](#)). Additionally, having community support ensures that the algorithm receives updates, bug fixes, and improvements. This enhances the reliability and longevity of the algorithm, making it better equipped to handle challenges during system implementation ([Persson et al., 2023](#)).

## 5.7 Map data representation and storage

This parameter focuses on how a SLAM algorithm is represented and manages maps, allowing the researcher to determine its suitability for system hardware implementation. The evaluation includes the chosen method's map representation, whether it is grid-based, feature-based, or point cloud, helping in assessing the efficiency of storing map information in the robotic system without encountering challenges ([Persson et al., 2023](#); [Acosta-Amaya et al., 2023](#)). The selection of map representation influences memory usage and computational demands. It is a critical factor for robotic applications, especially those based on CNN and deep learning approaches ([Duan et al., 2019](#)).

In conclusion, we have summarized the preceding details in [Table 2](#), offering a comprehensive overview of various V-SLAM algorithms. This table serves as a valuable resource for informed algorithm selection with comparative details for each method. It offers insights into the sensor capabilities, examining the types of sensors most effectively used by each algorithm and their role in facilitating algorithmic functionality. Moreover, the table underscores the potential application domains of the methods, empowering researchers to align their research objectives with suitable V-SLAM methodologies. The table also classifies algorithms based on their mapping scale distinguishing between small-scale (up to 100 m), medium-scale (up to 500 m), and large-scale (1 km and beyond) mapping capabilities ([Tian et al., 2023b](#); [Hong et al., 2021](#)).

It also assesses their performance under varying illumination conditions, classifying algorithms based on their robustness, with categories ranging from the lowest, which represents **(+)** and to the highest which represents **(+++++)**. Additionally, the table categorizes the algorithms based on their range of light intensity (RoLI), which reflects the robot's ability to operate effectively in diverse lighting conditions, spanning from very dim to extremely bright. Moreover, the tolerance to directionality (T2D) category assesses the algorithm's ability to function in environments with strong directional light sources, such as spotlights and windows. Collectively, these criteria collectively furnish a valuable resource for researchers seeking to pick the most fitting SLAM approach for their specific research endeavors.

## 6 Conclusion

---

The study simplifies the evaluation of V-SLAM methods, making it easy to understand their behavior and suitability for robotics applications. It covers various active V-SLAM methods, each with unique strengths, limitations, specialized use cases, and special workflows. It has served as a solid foundation for the proposed research methodology for selection among V-SLAM methods. Throughout the research, it becomes evident that V-SLAM's evolution is importantly linked to the availability of benchmark datasets, serving as a ground base for method validation. Consequently, the work has laid a strong foundation for understanding the system behavior of the working V-SLAM methods. It explores SLAM techniques that operate in the ROS environment, offering flexibility in simplifying the architecture of robotic systems. The study includes the identification of suitable algorithms and sensor fusion approaches relevant to researchers' work.

By examining previous studies, we identified the potential benefits of incorporating V-SLAM software tools into the system architecture. Additionally, the integration of hardware tools such as the T265 camera and OAK-D camera emerged as a valuable strategy. This integration has a

significant potential in reducing errors during robot navigation, thereby enhancing overall system robustness.

## Funding Statement

---

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This work is funded and supported by the Federal Ministry of Education and Research of Germany (BMBF) (AutoKoWAT-3DMAT under grant No. 13N16336) and German Research Foundation (DFG) under grants Al 638/15-1.

## Author contributions

---

BA: investigation, software, supervision, and writing–review and editing. TH: data curation, methodology, conceptualization, validation, investigation, resources, visualization, writing–review and editing. AA: methodology, formal analysis, validation, investigation, visualization, software, writing–review and editing. AA–H: methodology, supervision, project administration, validation, funding acquisition, resources, writing–review and editing.

## Conflict of interest

---

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

---

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

---

1. Abbad A. M., Haouala I., Raisov A., Benkredda R. (2023). Low cost mobile navigation using 2d-slam in complex environments [[Google Scholar](#)]
2. Acosta-Amaya G. A., Cadavid-Jimenez J. M., Jimenez-Builes J. A. (2023). Three-dimensional location and mapping analysis in mobile robotics based on visual slam methods. *J. Robotics* 2023, 1–15. 10.1155/2023/6630038 [[CrossRef](#)] [[Google Scholar](#)]
3. Ai Y.-b., Rui T., Yang X.-q., He J.-l., Fu L., Li J.-b., et al. (2021). Visual slam in dynamic environments based on object detection. *Def. Technol.* 17, 1712–1721. 10.1016/j.dt.2020.09.012 [[CrossRef](#)] [[Google Scholar](#)]
4. Alismail H., Browning B., Lucey S. (2017). “Photometric bundle adjustment for vision-based slam,” in Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20–24, 2016 (Springer; ), 324–341. Revised Selected Papers, Part IV. [[Google Scholar](#)]

5. Aloui K., Guizani A., Hammadi M., Haddar M., Soriano T. (2022). "Systematic literature review of collaborative slam applied to autonomous mobile robots," in 2022 IEEE Information Technologies and Smart Industrial Systems (ITSIS), 1–5. [[Google Scholar](#)]
6. Altawil B., Can F. C. (2023). Design and analysis of a four dof robotic arm with two grippers used in agricultural operations. *Int. J. Appl. Math. Electron. Comput.* 11, 79–87. 10.18100/ijamec.1217072 [[CrossRef](#)] [[Google Scholar](#)]
7. Arfa E. (2022). Study and implementation of LiDAR-based SLAM algorithm and map-based autonomous navigation for a telepresence robot to be used as a chaperon for smart laboratory requirements. *Master's thesis*, [[Google Scholar](#)]
8. Aslan M. F., Durdu A., Yusefi A., Sabanci K., Sungur C. (2021). A tutorial: mobile robotics, slam, bayesian filter, keyframe bundle adjustment and ros applications. *Robot Operating Syst. (ROS) Complete Reference* 6, 227–269. [[Google Scholar](#)]
9. Awais M., Henrich D. (2010). Human-robot collaboration by intention recognition using probabilistic state machines , 75–80. [[Google Scholar](#)]
10. Bahraini M. S., Bozorg M., Rad A. B. (2018). Slam in dynamic environments via ml-ransac. *Mechatronics* 49, 105–118. 10.1016/j.mechatronics.2017.12.002 [[CrossRef](#)] [[Google Scholar](#)]
11. Beghdadi A., Mallem M. (2022). A comprehensive overview of dynamic visual slam and deep learning: concepts, methods and challenges. *Mach. Vis. Appl.* 33, 54. 10.1007/s00138-022-01306-w [[CrossRef](#)] [[Google Scholar](#)]
12. Blo M., Omari S., Hutter M., Siegwart R. (2015). "Robust visual inertial odometry using a direct ekf-based approach," in 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS) (IEEE; ), 298–304. [[Google Scholar](#)]
13. Bongard J. (2008). *Probabilistic robotics. sebastian thrun, wolfram burgard, and dieter fox*. Cambridge, MA, United States: MIT press, 647. 2005. [[Google Scholar](#)]
14. Bruno H. M. S., Colombini E. L. (2021). Lift-slam: a deep-learning feature-based monocular visual slam method. *Neurocomputing* 455, 97–110. 10.1016/j.neucom.2021.05.027 [[CrossRef](#)] [[Google Scholar](#)]
15. Burri M., Nikolic J., Gohl P., Schneider T., Rehder J., Omari S., et al. (2016). The euroc micro aerial vehicle datasets. *Int. J. Robotics Res.* 35, 1157–1163. 10.1177/0278364915620033 [[CrossRef](#)] [[Google Scholar](#)]
16. Bustos A. P., Chin T.-J., Eriksson A., Reid I. (2019). "Visual slam: why bundle adjust?," in 2019 international conference on robotics and automation (ICRA) (IEEE; ), 2385–2391. [[Google Scholar](#)]
17. Buyval A., Afanasyev I., Magid E. (2017). "Comparative analysis of ros-based monocular slam methods for indoor navigation," in Ninth International Conference on Machine Vision (ICMV 2016) (SPIE; ), 305–310. [[Google Scholar](#)]
18. Ca C., Elvira R., Rodríguez J. J. G., Montiel J. M., Tardós J. D. (2021). Orb-slam3: an accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Trans. Robotics* 37, 1874–1890. 10.1109/tro.2021.3075644 [[CrossRef](#)] [[Google Scholar](#)]
19. Campos C., Elvira R., Rodríguez J. J. G., Montiel J. M., Tardós J. D. (2021). Orb-slam3: an accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Trans. Robotics* 37, 1874–1890. 10.1109/tro.2021.3075644 [[CrossRef](#)] [[Google Scholar](#)]
20. Castro G. I. (2021). *Scalability and consistency improvements in SLAM systems with applications in active multi-robot exploration*. Ph.D. thesis (FACULTY OF EXACT AND NATURAL SCIENCES DEPARTMENT OF COMPUTATIONÓN Improvements; ). [[Google Scholar](#)]
21. Chang Z., Wu H., Li C. (2023). Yolov4-tiny-based robust rgb-d slam approach with point and surface feature fusion in complex indoor environments. *J. Field Robotics* 40, 521–534. 10.1002/rob.22145 [[CrossRef](#)] [[Google Scholar](#)]

22. Chen H., Yang Z., Zhao X., Weng G., Wan H., Luo J., et al. (2020). Advanced mapping robot and high-resolution dataset. *Robotics Aut. Syst.* 131, 103559. 10.1016/j.robot.2020.103559 [CrossRef] [Google Scholar]
23. Chou C., Wang D., Song D., Davis T. A. (2019). “On the tunable sparse graph solver for pose graph optimization in visual slam problems,” in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE; ), 1300–1306. [Google Scholar]
24. Chu P. M., Sung Y., Cho K. (2018). Generative adversarial network-based method for transforming single rgb image into 3d point cloud. *IEEE Access* 7, 1021–1029. 10.1109/access.2018.2886213 [CrossRef] [Google Scholar]
25. Chung C.-M., Tseng Y.-C., Hsu Y.-C., Shi X.-Q., Hua Y.-H., Yeh J.-F., et al. (2023). “Orbeez-slam: a real-time monocular visual slam with orb features and nerf-realized mapping,” in 2023 IEEE International Conference on Robotics and Automation (ICRA) (IEEE; ), 9400–9406. [Google Scholar]
26. Civera J., Gálvez-López D., Riazuelo L., Tardós J. D., Montiel J. M. M. (2011). “Towards semantic slam using a monocular camera,” in 2011 IEEE/RSJ international conference on intelligent robots and systems (IEEE; ), 1277–1284. [Google Scholar]
27. Cui Y., Chen X., Zhang Y., Dong J., Wu Q., Zhu F. (2022). Bow3d: bag of words for real-time loop closing in 3d lidar slam, *IEEE Robotics Automation Lett.* 8, 2828–2835. 10.1109/lra.2022.3221336 [CrossRef] [Google Scholar]
28. CVG, T. U. o. M. (2023). LSD-SLAM: large-scale direct monocular SLAM. Available at: <https://cvg.cit.tum.de/research/vslam/lسدslam?redirect>.
29. Dai W., Zhang Y., Zheng Y., Sun D., Li P. (2021). Rgb-d slam with moving object tracking in dynamic environments. *IET Cyber-Systems Robotics* 3, 281–291. 10.1049/csy2.12019 [CrossRef] [Google Scholar]
30. [Dataset] uz.slaml (2023). ORB-SLAM3. Available at: [https://github.com/UZ-SLAMLab/ORB\\_SLAM3](https://github.com/UZ-SLAMLab/ORB_SLAM3).
31. Davison A. J., Reid I. D., Molton N. D., Stasse O. (2007). Monoslam: real-time single camera slam. *IEEE Trans. pattern analysis Mach. Intell.* 29, 1052–1067. 10.1109/tpami.2007.1049 [PubMed] [CrossRef] [Google Scholar]
32. De Croce M., Pire T., Bergero F. (2019). Ds-ptam: distributed stereo parallel tracking and mapping slam system. *J. Intelligent Robotic Syst.* 95, 365–377. 10.1007/s10846-018-0913-6 [CrossRef] [Google Scholar]
33. Duan C., Junginger S., Huang J., Jin K., Thurow K. (2019). Deep learning for visual slam in transportation robotics: a review. *Transp. Saf. Environ.* 1, 177–184. 10.1093/tse/tdz019 [CrossRef] [Google Scholar]
34. Durrant-Whyte H. F. (2012). *Integration, coordination and control of multi-sensor robot systems*, 36. Springer Science and Business Media. [Google Scholar]
35. El Bouazzaoui I., Rodriguez S., Vincke B., El Ouardi A. (2021). Indoor visual slam dataset with various acquisition modalities. *Data Brief* 39, 107496. 10.1016/j.dib.2021.107496 [PMC free article] [PubMed] [CrossRef] [Google Scholar]
36. End F., Hess J., Engelhard N., Sturm J., Cremers D., Burgard W. (2012). “An evaluation of the rgb-d slam system,” in 2012 IEEE international conference on robotics and automation (IEEE; ), 1691–1696. [Google Scholar]
37. Eng J., Schöps T., Cremers D. (2014). “Lsd-slam: large-scale direct monocular slam,” in European conference on computer vision (Springer; ), 834–849. [Google Scholar]
38. Engel J., Stückler J., Cremers D. (2015). “Large-scale direct slam with stereo cameras,” in 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS) (IEEE; ), 1935–1942. [Google Scholar]
39. eth.a (2023a). OKVIS: open keyframe-based visual-inertial SLAM. Available at: <https://github.com/ethz-asl/okvis>.
40. eth.a (2023b). Rovio: robust visual inertial odometry. Available at: <https://github.com/ethz-asl/rovio>.

41. Eudes A., Lhuillier M., Naudet-Collette S., Dhome M. (2010). "Fast odometry integration in local bundle adjustment-based visual slam," in 2010 20th International Conference on Pattern Recognition (IEEE; ), 290–293. [[Google Scholar](#)]
42. Eyvazpour R., Shoaran M., Karimian G. (2023). Hardware implementation of slam algorithms: a survey on implementation approaches and platforms. *Artif. Intell. Rev.* 56, 6187–6239. 10.1007/s10462-022-10310-5 [[CrossRef](#)] [[Google Scholar](#)]
43. Fan T., Wang H., Rubenstein M., Murphey T. (2020). Cpl-slam: efficient and certifiably correct planar graph-based slam using the complex number representation. *IEEE Trans. Robotics* 36, 1719–1737. 10.1109/tro.2020.3006717 [[CrossRef](#)] [[Google Scholar](#)]
44. felix (2023). RGB-D SLAM v2. Available at: [https://github.com/felixendres/rgbdslam\\_v2](https://github.com/felixendres/rgbdslam_v2).
45. Fernández-Moral E., Jiménez J. G., Arévalo V. (2013). Creating metric-topological maps for large-scale monocular slam. *ICINCO* (2), 39–47. [[Google Scholar](#)]
46. Fiedler M.-A., Werner P., Khalifa A., Al-Hamadi A. (2021). Sfpd: simultaneous face and person detection in real-time for human–robot interaction. *Sensors* 21, 5918. 10.3390/s21175918 [[PMC free article](#)] [[PubMed](#)] [[CrossRef](#)] [[Google Scholar](#)]
47. Fong T., Nourbakhsh I., Dautenhahn K. (2003). A survey of socially interactive robots. *Robotics Aut. Syst.* 42, 143–166. 10.1016/s0921-8890(02)00372-x [[CrossRef](#)] [[Google Scholar](#)]
48. Gao B., Lang H., Ren J. (2020). "Stereo visual slam for autonomous vehicles: a review," in 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (IEEE; ), 1316–1322. [[Google Scholar](#)]
49. Geiger A., Lenz P., Stiller C., Urtasun R. (2013). Vision meets robotics: the kitti dataset. *Int. J. Robotics Res.* 32, 1231–1237. 10.1177/0278364913491297 [[CrossRef](#)] [[Google Scholar](#)]
50. Geiger A., Lenz P., Urtasun R. (2012). "Are we ready for autonomous driving? the kitti vision benchmark suite," in 2012 IEEE conference on computer vision and pattern recognition (IEEE; ), 3354–3361. [[Google Scholar](#)]
51. Geneva P., Eckenhoff K., Lee W., Yang Y., Huang G. (2020). "OpenVINS: a research platform for visual-inertial estimation," in Proc. of the IEEE International Conference on Robotics and Automation, Paris, France. [[Google Scholar](#)]
52. GPL (2023). Available at: <https://github.com/Oxford-PTAM/PTAM-GPL>.
53. Grisetti G., Stachniss C., Burgard W. (2007). Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Trans. Robotics* 23, 34–46. 10.1109/tro.2006.889486 [[CrossRef](#)] [[Google Scholar](#)]
54. Gu P., Meng Z., Zhou P. (2022). Real-time visual inertial odometry with a resource-efficient harris corner detection accelerator on fpga platform , 10542–10548. [[Google Scholar](#)]
55. Gurel C. S. (2018). *Real-time 2d and 3d slam using rtab-map, gmapping, and cartographer packages*. University of Maryland. [[Google Scholar](#)]
56. Han Y., Mokhtarzadeh A. A., Xiao S. (2023). Novel cartographer using an oak-d smart camera for indoor robots location and navigation. *J. Phys. Conf. Ser.* 2467, 012029. 10.1088/1742-6596/2467/1/012029 [[CrossRef](#)] [[Google Scholar](#)]
57. Handa A., Whelan T., McDonald J., Davison A. J. (2014). "A benchmark for rgbd visual odometry, 3d reconstruction and slam," in 2014 IEEE international conference on Robotics and automation (ICRA) (IEEE; ), 1524–1531. [[Google Scholar](#)]
58. Hastürk Ö., Erkmen A. M. (2021). Dudmap: 3d rgbd mapping for dense, unstructured, and dynamic environment. *Int. J. Adv. Robotic Syst.* 18, 172988142110161. 10.1177/17298814211016178 [[CrossRef](#)] [[Google Scholar](#)]

59. Hempel T., Al-Hamadi A. (2020). Pixel-wise motion segmentation for slam in dynamic environments. *IEEE Access* 8, 164521–164528. 10.1109/access.2020.3022506 [[CrossRef](#)] [[Google Scholar](#)]
60. Hempel T., Dinges L., Al-Hamadi A. (2023). “Sentiment-based engagement strategies for intuitive human-robot interaction,” in Proceedings of the 18th International Joint Conference on Computer Vision, 680–686. *Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 4: VISAPP*. INSTICC (SciTePress). 10.5220/0011772900003417 [[CrossRef](#)] [[Google Scholar](#)]
61. Henein M., Zhang J., Mahony R., Ila V. (2020). *Dynamic slam: the need for speed*, 2123–2129. [[Google Scholar](#)]
62. Hess W., Kohler D., Rapp H., Andor D. (2016). “Real-time loop closure in 2d lidar slam,” in 2016 IEEE international conference on robotics and automation (ICRA) (IEEE; ), 1271–1278. [[Google Scholar](#)]
63. Heyer C. (2010). “Human-robot interaction and future industrial robotics applications,” in 2010 ieee/rsj international conference on intelligent robots and systems (IEEE; ), 4749–4754. [[Google Scholar](#)]
64. hkust.a (2023). VINS-Mono. Available at: <https://github.com/HKUST-Aerial-Robotics/VINS-Mono>.
65. Hong S., Bangunharcana A., Park J.-M., Choi M., Shin H.-S. (2021). Visual slam-based robotic mapping method for planetary construction. *Sensors* 21, 7715. 10.3390/s21227715 [[PMC free article](#)] [[PubMed](#)] [[CrossRef](#)] [[Google Scholar](#)]
66. Hsiao M., Westman E., Zhang G., Kaess M. (2017). “Keyframe-based dense planar slam,” in 2017 IEEE International Conference on Robotics and Automation (ICRA) (ieee; ), 5110–5117. [[Google Scholar](#)]
67. Huang L. (2021). “Review on lidar-based slam techniques,” in 2021 International Conference on Signal Processing and Machine Learning (CONF-SPML) (IEEE; ), 163–168. [[Google Scholar](#)]
68. Introlab (2023). RTAB-Map. Available at: <http://introlab.github.io/rtabmap/>.
69. Ji T., Wang C., Xie L. (2021). “Towards real-time semantic rgb-d slam in dynamic environments,” in 2021 IEEE International Conference on Robotics and Automation (ICRA) (IEEE; ), 11175–11181. [[Google Scholar](#)]
70. Joo S.-H., Manzoor S., Rocha Y. G., Bae S.-H., Lee K.-H., Kuc T.-Y., et al. (2020). Autonomous navigation framework for intelligent robots based on a semantic environment modeling. *Appl. Sci.* 10, 3219. 10.3390/app10093219 [[CrossRef](#)] [[Google Scholar](#)]
71. Kasyanov A., Engelmann F., Stückler J., Leibe B. (2017). Keyframe-based visual-inertial online slam with relocalization , 6662–6669. [[Google Scholar](#)]
72. Kazerouni I. A., Fitzgerald L., Dooly G., Toal D. (2022). A survey of state-of-the-art on visual slam. *Expert Syst. Appl.* 205, 117734. 10.1016/j.eswa.2022.117734 [[CrossRef](#)] [[Google Scholar](#)]
73. Kerl C., Sturm J., Cremers D. (2013). “Dense visual slam for rgb-d cameras,” in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IEEE; ), 2100–2106. [[Google Scholar](#)]
74. Khoyani A., Amini M. (2023). A survey on visual slam algorithms compatible for 3d space reconstruction and navigation , 01–06. [[Google Scholar](#)]
75. Klein G., Murray D. (2007). “Parallel tracking and mapping for small ar workspaces,” in 2007 6th IEEE and ACM international symposium on mixed and augmented reality (IEEE; ), 225–234. [[Google Scholar](#)]
76. Kuang Z., Wei W., Yan Y., Li J., Lu G., Peng Y., et al. (2022). A real-time and robust monocular visual inertial slam system based on point and line features for mobile robots of smart cities toward 6g. *IEEE Open J. Commun. Soc.* 3, 1950–1962. 10.1109/ojcoms.2022.3217147 [[CrossRef](#)] [[Google Scholar](#)]
77. Kucner T. P., Magnusson M., Mghames S., Palmieri L., Verdoja F., Swaminathan C. S., et al. (2023). Survey of maps of dynamics for mobile robots. *Int. J. Robotics Res.*, 02783649231190428. [[Google Scholar](#)]

78. Labb   F., Michaud M. (2019). Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *J. field robotics* 36, 416–446. 10.1002/rob.21831 [[CrossRef](#)] [[Google Scholar](#)]
79. Laidlow T., Czarnowski J., Leutenegger S. (2019). Deepfusion: real-time dense 3d reconstruction for monocular slam using single-view depth and gradient predictions , 4068–4074. [[Google Scholar](#)]
80. Lee G., Moon B.-C., Lee S., Han D. (2020). Fusion of the slam with wi-fi-based positioning methods for mobile robot-based learning data collection, localization, and tracking in indoor spaces. *Sensors* 20, 5182. 10.3390/s20185182 [[PMC free article](#)] [[PubMed](#)] [[CrossRef](#)] [[Google Scholar](#)]
81. Leut S., Lynen S., Bosse M., Siegwart R., Furgale P. (2015). Keyframe-based visual–inertial odometry using nonlinear optimization. *Int. J. Robotics Res.* 34, 314–334. 10.1177/0278364914554813 [[CrossRef](#)] [[Google Scholar](#)]
82. Leutenegger S. (2022). *Okvis2: realtime scalable visual-inertial slam with loop closure*. arXiv preprint arXiv:2202.09199 . [[Google Scholar](#)]
83. Li D., Shi X., Long Q., Liu S., Yang W., Wang F., et al. (2020). “Dxslam: a robust and efficient visual slam system with deep features,” in 2020 IEEE/RSJ International conference on intelligent robots and systems (IROS) (IEEE; ), 4958–4965. [[Google Scholar](#)]
84. Li G., Hou J., Chen Z., Yu L., Fei S. (2023a). Robust stereo inertial odometry based on self-supervised feature points. *Appl. Intell.* 53, 7093–7107. 10.1007/s10489-022-03278-w [[CrossRef](#)] [[Google Scholar](#)]
85. Li P., Qin T., Shen S. (2018). “Stereo vision-based semantic 3d object and ego-motion tracking for autonomous driving,” in Proceedings of the European Conference on Computer Vision (ECCV; ), 646–661. [[Google Scholar](#)]
86. Li Q., Wang X., Wu T., Yang H. (2022a). Point-line feature fusion based field real-time rgb-d slam. *Comput. Graph.* 107, 10–19. 10.1016/j.cag.2022.06.013 [[CrossRef](#)] [[Google Scholar](#)]
87. Li S., Zhang D., Xian Y., Li B., Zhang T., Zhong C. (2022b). *Overview of deep learning application on visual slam*, 102298. *Displays*. [[Google Scholar](#)]
88. Li S., Zheng P., Liu S., Wang Z., Wang X. V., Zheng L., et al. (2023b). Proactive human–robot collaboration: mutual-cognitive, predictable, and self-organising perspectives. *Robotics Computer-Integrated Manuf.* 81, 102510. 10.1016/j.rcim.2022.102510 [[CrossRef](#)] [[Google Scholar](#)]
89. Li Y., Guo Z., Yang Z., Sun Y., Zhao L., Tombari F. (2023c). *Open-structure: a structural benchmark dataset for slam algorithms*. arXiv preprint arXiv:2310.10931 . [[Google Scholar](#)]
90. Lin H.-Y., Yeh M.-C. (2022). Drift-free visual slam for mobile robot localization by integrating uwb technology. *IEEE Access* 10, 93636–93645. 10.1109/access.2022.3203438 [[CrossRef](#)] [[Google Scholar](#)]
91. Liu Y., Miura J. (2021). Rds-slam: real-time dynamic slam using semantic segmentation methods, *Ieee Access* 9, 23772–23785. 10.1109/access.2021.3050617 [[CrossRef](#)] [[Google Scholar](#)]
92. Lopez J., Sanchez-Vilarino P., Cacho M. D., Guill  n E. L. (2020). Obstacle avoidance in dynamic environments based on velocity space optimization. *Robotics Aut. Syst.* 131, 103569. 10.1016/j.robot.2020.103569 [[CrossRef](#)] [[Google Scholar](#)]
93. Luo H., Pape C., Reithmeier E. (2021). Robust rgbd visual odometry using windowed direct bundle adjustment and slanted support plane. *IEEE Robotics Automation Lett.* 7, 350–357. 10.1109/lra.2021.3126347 [[CrossRef](#)] [[Google Scholar](#)]
94. Lynch C., Wahid A., Tompson J., Ding T., Betker J., Baruch R., et al. (2023). Interactive language: talking to robots in real time. *IEEE Robotics Automation Lett.*, 1–8. 10.1109/lra.2023.3295255 [[CrossRef](#)] [[Google Scholar](#)]

95. Macario Barros A., Michel M., Moline Y., Corre G., Carrel F. (2022). A comprehensive survey of visual slam algorithms. *Robotics* 11, 24. 10.3390/robotics11010024 [[CrossRef](#)] [[Google Scholar](#)]
96. Mane A. A., Parihar M. N., Jadhav S. P., Gadre R. (2016). "Data acquisition analysis in slam applications," in 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT) (IEEE; ), 339–343. [[Google Scholar](#)]
97. Martínez-Otzeta J. M., Rodríguez-Moreno I., Mendialdua I., Sierra B. (2022). Ransac for robotic applications: a survey. *Sensors* 23, 327. 10.3390/s23010327 [[PMC free article](#)] [[PubMed](#)] [[CrossRef](#)] [[Google Scholar](#)]
98. Mazumdar H., Chakraborty C., Sathvik M., Jayakumar P., Kaushik A. (2023). Optimizing pix2pix gan with attention mechanisms for ai-driven polyp segmentation in iomt-enabled smart healthcare. *IEEE J. Biomed. Health Inf.*, 1–8. 10.1109/jbhi.2023.3328962 [[PubMed](#)] [[CrossRef](#)] [[Google Scholar](#)]
99. Meng X., Gao W., Hu Z. (2018). Dense rgb-d slam with multiple cameras. *Sensors* 18, 2118. 10.3390/s18072118 [[PMC free article](#)] [[PubMed](#)] [[CrossRef](#)] [[Google Scholar](#)]
100. Meng X., Li B., Li B., Li B., Li B. (2022). "Prob-slam: real-time visual slam based on probabilistic graph optimization," in Proceedings of the 8th International Conference on Robotics and Artificial Intelligence, 39–45. 10.1145/3573910.3573920 [[CrossRef](#)] [[Google Scholar](#)]
101. MIT.S (2023). Kimera: an open-source library for real-time metric-semantic localization and mapping. Available at: <https://github.com/MIT-SPARK/Kimera>.
102. Mohamed N., Al-Jaroodi J., Jawhar I. (2008). "Middleware for robotics: a survey," in 2008 IEEE Conference on Robotics, Automation and Mechatronics (Ieee; ), 736–742. [[Google Scholar](#)]
103. Mur-A J. D., Tars R. (2014). "Orb-slam: tracking and mapping recognizable," in Proceedings of the Workshop on Multi View Geometry in Robotics (MVIGRO)-RSS. [[Google Scholar](#)]
104. Mur-Artal R., Montiel J. M. M., Tardos J. D. (2015). Orb-slam: a versatile and accurate monocular slam system. *IEEE Trans. robotics* 31, 1147–1163. 10.1109/tro.2015.2463671 [[CrossRef](#)] [[Google Scholar](#)]
105. Mur-Artal R., Tardós J. D. (2017a). Orb-slam2: an open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. robotics* 33, 1255–1262. 10.1109/tro.2017.2705103 [[CrossRef](#)] [[Google Scholar](#)]
106. Mur-Artal R., Tardós J. D. (2017b). Visual-inertial monocular slam with map reuse. *IEEE Robotics Automation Lett.* 2, 796–803. 10.1109/lra.2017.2653359 [[CrossRef](#)] [[Google Scholar](#)]
107. Nakamura T., Kobayashi M., Motoi N. (2023). Path planning for mobile robot considering turnabouts on narrow road by deep q-network. *IEEE Access* 11, 19111–19121. 10.1109/access.2023.3247730 [[CrossRef](#)] [[Google Scholar](#)]
108. Navvis (2023). Map forming. Available at: <https://www.navvis.com/technology/slam> (Accessed on November 14, 2023).
109. Ne R. A., Lovegrove S. J., Davison A. J. (2011). "Dtam: dense tracking and mapping in real-time," in 2011 international conference on computer vision (IEEE; ), 2320–2327. [[Google Scholar](#)]
110. Newcombe R. A., Izadi S., Hilliges O., Molyneaux D., Kim D., Davison A. J., et al. (2011). "Kinectfusion: real-time dense surface mapping and tracking," in 2011 10th IEEE international symposium on mixed and augmented reality (Ieee; ), 127–136. [[Google Scholar](#)]
111. Nguyen Q. H., Johnson P., Latham D. (2022). Performance evaluation of ros-based slam algorithms for handheld indoor mapping and tracking systems. *IEEE Sensors J.* 23, 706–714. 10.1109/jsen.2022.3224224 [[CrossRef](#)] [[Google Scholar](#)]

112. Nguyen T., Mann G. K., Vardy A., Gosine R. G. (2020). Ckf-based visual inertial odometry for long-term trajectory operations. *J. Robotics* 2020, 1–14. 10.1155/2020/7362952 [[CrossRef](#)] [[Google Scholar](#)]
113. Niu X., Liu H., Yuan J. (2019). “Rbg-d indoor simultaneous location and mapping based on inliers tracking statistics,” in Journal of Physics: Conference Series (IOP Publishing; ), 1176, 062023. [[Google Scholar](#)]
114. Ortega-Gomez J. I., Morales-Hernandez L. A., Cruz-Albaran I. A. (2023). A specialized database for autonomous vehicles based on the kitti vision benchmark. *Electronics* 12, 3165. 10.3390/electronics12143165 [[CrossRef](#)] [[Google Scholar](#)]
115. Pal S., Gupta S., Das N., Ghosh K. (2022). Evolution of simultaneous localization and mapping framework for autonomous robotics—a comprehensive review. *J. Aut. Veh. Syst.* 2, 020801. 10.1115/1.4055161 [[CrossRef](#)] [[Google Scholar](#)]
116. Palazzolo E., Behley J., Lottes P., Giguere P., Stachniss C. (2019). “Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals,” in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE; ), 7855–7862. [[Google Scholar](#)]
117. Persson N., Ekström M. C., Ekström M., Papadopoulos A. V. (2023). “On the initialization problem for timed-elastic bands,” in Proceedings of the 22nd IFAC World Congress (IFAC WC). [[Google Scholar](#)]
118. Peter J., Thomas M. J., Mohan S. (2023). Development of an autonomous ground robot using a real-time appearance based (rtab) algorithm for enhanced spatial mapping [[Google Scholar](#)]
119. Picard Q., Chevobbe S., Darouich M., Didier J.-Y. (2023). *A survey on real-time 3d scene reconstruction with slam methods in embedded systems*. arXiv preprint arXiv:2309.05349 . [[Google Scholar](#)]
120. Placed J. A., Strader J., Carrillo H., Atanasov N., Indelman V., Carbone L., et al. (2023). A survey on active simultaneous localization and mapping: state of the art and new frontiers. *IEEE Trans. Robotics* 39, 1686–1705. 10.1109/tro.2023.3248510 [[CrossRef](#)] [[Google Scholar](#)]
121. Prati E., Villani V., Grandi F., Peruzzini M., Sabattini L. (2021). Use of interaction design methodologies for human–robot collaboration in industrial scenarios. *IEEE Trans. Automation Sci. Eng.* 19, 3126–3138. 10.1109/tase.2021.3107583 [[CrossRef](#)] [[Google Scholar](#)]
122. Qin T., Li P., Shen S. (2018). Vins-mono: a robust and versatile monocular visual-inertial state estimator. *IEEE Trans. Robotics* 34, 1004–1020. 10.1109/tro.2018.2853729 [[CrossRef](#)] [[Google Scholar](#)]
123. Ragot N., Khemmar R., Pokala A., Rossi R., Ertaud J.-Y. (2019). “Benchmark of visual slam algorithms: orb-slam2 vs rtab-map,” in 2019 Eighth International Conference on Emerging Security Technologies (EST) (IEEE; ), 1–6. [[Google Scholar](#)]
124. Raikwar S., Yu H., Herlitzius T. (2023). 2d lidar slam localization system for a mobile robotic platform in gps denied environment. *J. Biosyst. Eng.* 48, 123–135. 10.1007/s42853-023-00176-y [[CrossRef](#)] [[Google Scholar](#)]
125. raulmur (2023a). ORB-SLAM. Available at: [https://github.com/raulmur/ORB\\_SLAM](https://github.com/raulmur/ORB_SLAM).
126. raulmur (2023b). ORB-SLAM2. Available at: [https://github.com/raulmur/ORB\\_SLAM2](https://github.com/raulmur/ORB_SLAM2).
127. Ren G., Cao Z., Liu X., Tan M., Yu J. (2022). Plj-slam: monocular visual slam with points, lines, and junctions of coplanar lines. *IEEE Sensors J.* 22, 15465–15476. 10.1109/jsen.2022.3185122 [[CrossRef](#)] [[Google Scholar](#)]
128. Rintar (2023). dtam-1. Available at: <https://github.com/Rintarooo/dtam-1>.
129. Roch J., Fayyad J., Najjaran H. (2023). Dopeslam: high-precision ros-based semantic 3d slam in a dynamic environment. *Sensors* 23, 4364. 10.3390/s23094364 [[PMC free article](#)] [[PubMed](#)] [[CrossRef](#)] [[Google Scholar](#)]

130. Ros A., Abate M., Chang Y., Carlone L. (2020). "Kimera: an open-source library for real-time metric-semantic localization and mapping," in 2020 IEEE International Conference on Robotics and Automation (ICRA) (IEEE; ), 1689–1696. [[Google Scholar](#)]
131. Rosinol A., Violette A., Abate M., Hughes N., Chang Y., Shi J., et al. (2021). Kimera: from slam to spatial perception with 3d dynamic scene graphs. *Int. J. Robotics Res.* 40, 1510–1546. 10.1177/02783649211056674 [[CrossRef](#)] [[Google Scholar](#)]
132. Scaradozzi D., Zingaretti S., Ferrari A. (2018). Simultaneous localization and mapping (slam) robotics techniques: a possible application in surgery. *Shanghai Chest* 2, 5. 10.21037/shc.2018.01.01 [[CrossRef](#)] [[Google Scholar](#)]
133. Schneider T., Dymczyk M., Fehr M., Egger K., Lynen S., Gilitschenski I., et al. (2018). maplab: an open framework for research in visual-inertial mapping and localization. *IEEE Robotics Automation Lett.* 3, 1418–1425. 10.1109/lra.2018.2800113 [[CrossRef](#)] [[Google Scholar](#)]
134. Schöps T., Engel J., Cremers D. (2014). "Semi-dense visual odometry for ar on a smartphone," in 2014 IEEE international symposium on mixed and augmented reality (ISMAR) (IEEE; ), 145–150. [[Google Scholar](#)]
135. Servières M., Renaudin V., Dupuis A., Antigny N. (2021). Visual and visual-inertial slam: state of the art, classification, and experimental benchmarking. *J. Sensors* 2021, 1–26. 10.1155/2021/2054828 [[CrossRef](#)] [[Google Scholar](#)]
136. Sharafutdinov D., Griguletskii M., Kopanov P., Kurenkov M., Ferrer G., Burkov A., et al. (2023). Comparison of modern open-source visual slam approaches. *J. Intelligent Robotic Syst.* 107, 43. 10.1007/s10846-023-01812-7 [[CrossRef](#)] [[Google Scholar](#)]
137. Sheng L., Xu D., Ouyang W., Wang X. (2019). "Unsupervised collaborative learning of keyframe detection and visual odometry towards monocular deep slam," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 4302–4311. [[Google Scholar](#)]
138. Sheridan T. B. (2016). Human–robot interaction: status and challenges. *Hum. factors* 58, 525–532. 10.1177/0018720816644364 [[PubMed](#)] [[CrossRef](#)] [[Google Scholar](#)]
139. Soares J. C. V., Gattass M., Meggiolaro M. A. (2021). Crowd-slam: visual slam towards crowded environments using object detection. *J. Intelligent Robotic Syst.* 102, 50. 10.1007/s10846-021-01414-1 [[CrossRef](#)] [[Google Scholar](#)]
140. Soliman A., Bonardi F., Sidibé D., Bouchafa S. (2023). Dh-ptam: a deep hybrid stereo events-frames parallel tracking and mapping system. *arXiv preprint arXiv:2306.01891* [[Google Scholar](#)]
141. Son S., Chen J., Zhong Y., Zhang W., Hou W., Zhang L. (2023). Sce-slam: a real-time semantic rgbd slam system in dynamic scenes based on spatial coordinate error. *Meas. Sci. Technol.* 34, 125006. 10.1088/1361-6501/aceb7e [[CrossRef](#)] [[Google Scholar](#)]
142. Song K., Li J., Qiu R., Yang G. (2022). Monocular visual-inertial odometry for agricultural environments. *IEEE Access* 10, 103975–103986. 10.1109/access.2022.3209186 [[CrossRef](#)] [[Google Scholar](#)]
143. Song Y., Zhang Z., Wu J., Wang Y., Zhao L., Huang S. (2021). A right invariant extended kalman filter for object based slam. *IEEE Robotics Automation Lett.* 7, 1316–1323. 10.1109/lra.2021.3139370 [[CrossRef](#)] [[Google Scholar](#)]
144. Sousa R. B., Sobreira H. M., Moreira A. P. (2023). A systematic literature review on long-term localization and mapping for mobile robots. *J. Field Robotics* 40, 1245–1322. 10.1002/rob.22170 [[CrossRef](#)] [[Google Scholar](#)]
145. Steinbrücker F., Sturm J., Cremers D. (2011). "Real-time visual odometry from dense rgb-d images," in 2011 IEEE international conference on computer vision workshops (ICCV Workshops) (IEEE; ), 719–722. [[Google Scholar](#)]
146. Strazdas D., Hintz J., Felßberg A.-M., Al-Hamadi A. (2020). Robots and wizards: an investigation into natural human–robot interaction. *IEEE Access* 8, 207635–207642. 10.1109/access.2020.3037724 [[CrossRef](#)] [[Google Scholar](#)]

147. Sumikura S., Shibuya M., Sakurada K. (2019). “Openvslam: a versatile visual slam framework,” in Proceedings of the 27th ACM International Conference on Multimedia, 2292–2295. [[Google Scholar](#)]
148. Sun Y., Liu M., Meng M. Q.-H. (2017). Improving rgb-d slam in dynamic environments: a motion removal approach. *Robotics Aut. Syst.* 89, 110–122. 10.1016/j.robot.2016.11.012 [[CrossRef](#)] [[Google Scholar](#)]
149. Taheri H., Xia Z. C. (2021). Slam; definition and evolution. *Eng. Appl. Artif. Intell.* 97, 104032. 10.1016/j.engappai.2020.104032 [[CrossRef](#)] [[Google Scholar](#)]
150. Taketomi T., Uchiyama H., Ikeda S. (2017). Visual slam algorithms: a survey from 2010 to 2016. *IPSJ Trans. Comput. Vis. Appl.* 9, 16–11. 10.1186/s41074-017-0027-2 [[CrossRef](#)] [[Google Scholar](#)]
151. Theodorou C., Velisavljevic V., Dyo V., Nonyelu F. (2022). Visual slam algorithms and their application for ar, mapping, localization and wayfinding. *Array* 15, 100222. 10.1016/j.array.2022.100222 [[CrossRef](#)] [[Google Scholar](#)]
152. Tian Y., Chang Y., Quang L., Schang A., Nieto-Granda C., How J. P., et al. (2023a). *Resilient and distributed multi-robot visual slam: datasets, experiments, and lessons learned*. *arXiv preprint arXiv:2304.04362* . [[Google Scholar](#)]
153. Tian Y., Chang Y., Quang L., Schang A., Nieto-Granda C., How J. P., et al. (2023b). *Resilient and distributed multi-robot visual slam: datasets, experiments, and lessons learned*. *arXiv preprint arXiv:2304.04362* . [[Google Scholar](#)]
154. Tourani A., Bavle H., Sanchez-Lopez J. L., Voos H. (2022). Visual slam: what are the current trends and what to expect? *Sensors* 22, 9297. 10.3390/s22239297 [[PMC free article](#)] [[PubMed](#)] [[CrossRef](#)] [[Google Scholar](#)]
155. Tsintotas K. A., Bampis L., Gasteratos A. (2022). The revisiting problem in simultaneous localization and mapping: a survey on visual loop closure detection. *IEEE Trans. Intelligent Transp. Syst.* 23, 19929–19953. 10.1109/tits.2022.3175656 [[CrossRef](#)] [[Google Scholar](#)]
156. tum.v (2023). DVO-SLAM: direct visual odometry for monocular cameras. Available at: [https://github.com/tum-vision/dvo\\_slam](https://github.com/tum-vision/dvo_slam).
157. Ullah I., Su X., Zhang X., Choi D. (2020). Simultaneous localization and mapping based on kalman filter and extended kalman filter. *Wirel. Commun. Mob. Comput.* 2020, 1–12. 10.1155/2020/2138643 [[CrossRef](#)] [[Google Scholar](#)]
158. Van Nam D., Gon-Woo K. (2021). “Solid-state lidar based-slam: a concise review and application,” in 2021 IEEE International Conference on Big Data and Smart Computing (BigComp) (IEEE; ), 302–305. [[Google Scholar](#)]
159. Wang H., Ko J. Y., Xie L. (2022). *Multi-modal semantic slam for complex dynamic environments*. *arXiv preprint arXiv:2205.04300* . [[Google Scholar](#)]
160. Wang Z., Pang B., Song Y., Yuan X., Xu Q., Li Y. (2023). Robust visual-inertial odometry based on a kalman filter and factor graph. *IEEE Trans. Intelligent Transp. Syst.* 24, 7048–7060. 10.1109/tits.2023.3258526 [[CrossRef](#)] [[Google Scholar](#)]
161. Wu W., Guo L., Gao H., You Z., Liu Y., Chen Z. (2022). Yolo-slam: a semantic slam system towards dynamic environment with geometric constraint. *Neural Comput. Appl.* 34, 6011–6026. 10.1007/s00521-021-06764-3 [[CrossRef](#)] [[Google Scholar](#)]
162. Xiao L., Wang J., Qiu X., Rong Z., Zou X. (2019). Dynamic-slam: semantic monocular visual localization and mapping based on deep learning in dynamic environment. *Robotics Aut. Syst.* 117, 1–16. 10.1016/j.robot.2019.03.012 [[CrossRef](#)] [[Google Scholar](#)]
163. Xu C., Liu Z., Li Z. (2021). Robust visual-inertial navigation system for low precision sensors under indoor and outdoor environments. *Remote Sens.* 13, 772. 10.3390/rs13040772 [[CrossRef](#)] [[Google Scholar](#)]

164. Yan L., Hu X., Zhao L., Chen Y., Wei P., Xie H. (2022). Dgs-slam: a fast and robust rgbd slam in dynamic environments combined by geometric and semantic information. *Remote Sens.* 14, 795. 10.3390/rs14030795 [[CrossRef](#)] [[Google Scholar](#)]
165. Yang X., Li H., Zhai H., Ming Y., Liu Y., Zhang G. (2022). “Vox-fusion: dense tracking and mapping with voxel-based neural implicit representation,” in 2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR) (IEEE; ), 499–507. [[Google Scholar](#)]
166. Yousif K., Bab-Hadiashar A., Hoseinnezhad R. (2015). An overview to visual odometry and visual slam: applications to mobile robotics. *Intell. Ind. Syst.* 1, 289–311. 10.1007/s40903-015-0032-7 [[CrossRef](#)] [[Google Scholar](#)]
167. Zang Q., Zhang K., Wang L., Wu L. (2023). An adaptive orb-slam3 system for outdoor dynamic environments. *Sensors* 23, 1359. 10.3390/s23031359 [[PMC free article](#)] [[PubMed](#)] [[CrossRef](#)] [[Google Scholar](#)]
168. Zhang J., Zhu C., Zheng L., Xu K. (2021a). Rosefusion: random optimization for online dense reconstruction under fast camera motion. *ACM Trans. Graph. (TOG)* 40, 1–17. 10.1145/3476576.3476604 [[CrossRef](#)] [[Google Scholar](#)]
169. Zhang Q., Li C. (2023). Semantic slam for mobile robots in dynamic environments based on visual camera sensors. *Meas. Sci. Technol.* 34, 085202. 10.1088/1361-6501/acd1a4 [[CrossRef](#)] [[Google Scholar](#)]
170. Zhang S., Zheng L., Tao W. (2021b). Survey and evaluation of rgb-d slam. *IEEE Access* 9, 21367–21387. 10.1109/access.2021.3053188 [[CrossRef](#)] [[Google Scholar](#)]
171. Zhang W., Wang S., Dong X., Guo R., Haala N. (2023). Bamf-slam: bundle adjusted multi-fisheye visual-inertial slam using recurrent field transforms. *arXiv preprint arXiv:2306.01173* [[Google Scholar](#)]
172. Zhang X., Liu Q., Zheng B., Wang H., Wang Q. (2020). A visual simultaneous localization and mapping approach based on scene segmentation and incremental optimization. *Int. J. Adv. Robotic Syst.* 17, 172988142097766. 10.1177/1729881420977669 [[CrossRef](#)] [[Google Scholar](#)]
173. Zhang X., Su Y., Zhu X. (2017). “Loop closure detection for visual slam systems using convolutional neural network,” in 2017 23rd International Conference on Automation and Computing (ICAC) (IEEE; ), 1–6. [[Google Scholar](#)]
174. Zheng P., Li S., Xia L., Wang L., Nassehi A. (2022). A visual reasoning-based approach for mutual-cognitive human-robot collaboration. *CIRP Ann.* 71, 377–380. 10.1016/j.cirp.2022.04.016 [[CrossRef](#)] [[Google Scholar](#)]
175. Zheng S., Wang J., Rizos C., Ding W., El-Mowafy A. (2023). Simultaneous localization and mapping (slam) for autonomous driving: concept and analysis. *Remote Sens.* 15, 1156. 10.3390/rs15041156 [[CrossRef](#)] [[Google Scholar](#)]
176. Zhou L., Koppel D., Ju H., Steinbruecker F., Kaess M. (2020). “An efficient planar bundle adjustment algorithm,” in 2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR) (IEEE; ), 136–145. [[Google Scholar](#)]
177. Zhu Z., Peng S., Larsson V., Xu W., Bao H., Cui Z., et al. (2022). Nice-slam: neural implicit scalable encoding for slam , 12786–12796. [[Google Scholar](#)]