



Review

# An Overview on Visual SLAM: From Tradition to Semantic

Weifeng Chen <sup>1,2</sup>, Guangtao Shang <sup>2</sup>, Aihong Ji <sup>3</sup>, Chengjun Zhou <sup>2</sup>, Xiyang Wang <sup>2</sup>, Chonghui Xu <sup>2</sup>, Zhenxiong Li <sup>2</sup> and Kai Hu <sup>2,\*</sup>

- <sup>1</sup> School of Mechanical and Electronic Engineering, Quanzhou University of Information Engineering, Quanzhou 362000, China; 002021@nuist.edu.cn
- <sup>2</sup> School of Automation, Nanjing University of Information Science & Technology, Nanjing 210044, China; 20201222014@nuist.edu.cn (G.S.); 20211257010@nuist.edu.cn (C.Z.); 20211267006@nuist.edu.cn (X.W.); 20211249101@nuist.edu.cn (C.X.); 20211257005@nuist.edu.cn (Z.L.)
- <sup>3</sup> Lab of Locomotion Bioinspiration and Intelligent Robots, College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China; meeahji@nuaa.edu.cn
- \* Correspondence: 001600@nuist.edu.cn

**Abstract:** Visual SLAM (VSLAM) has been developing rapidly due to its advantages of low-cost sensors, the easy fusion of other sensors, and richer environmental information. Traditional vision-based SLAM research has made many achievements, but it may fail to achieve wished results in challenging environments. Deep learning has promoted the development of computer vision, and the combination of deep learning and SLAM has attracted more and more attention. Semantic information, as high-level environmental information, can enable robots to better understand the surrounding environment. This paper introduces the development of VSLAM technology from two aspects: traditional VSLAM and semantic VSLAM combined with deep learning. For traditional VSLAM, we summarize the advantages and disadvantages of indirect and direct methods in detail and give some classical VSLAM open-source algorithms. In addition, we focus on the development of semantic VSLAM based on deep learning. Starting with typical neural networks CNN and RNN, we summarize the improvement of neural networks for the VSLAM system in detail. Later, we focus on the help of target detection and semantic segmentation for VSLAM semantic information introduction. We believe that the development of the future intelligent era cannot be without the help of semantic technology. Introducing deep learning into the VSLAM system to provide semantic information can help robots better perceive the surrounding environment and provide people with higher-level help.

**Keywords:** SLAM; deep learning; neural networks; computer vision; semantic; intelligent era



**Citation:** Chen, W.; Shang, G.; Ji, A.; Zhou, C.; Wang, X.; Xu, C.; Li, Z.; Hu, K. An Overview on Visual SLAM: From Tradition to Semantic. *Remote Sens.* **2022**, *14*, 3010. <https://doi.org/10.3390/rs14133010>

Academic Editors: Fabio Remondino, Radosław Zimroz, Denis Guilhot and Vittorio Cannas

Received: 29 May 2022  
Accepted: 17 June 2022  
Published: 23 June 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

People need the mobile robot to perform some tasks by themselves, which needs the robot to be able to adapt to an unfamiliar environment. Therefore, SLAM [1] (Simultaneous Localization and Mapping), which enables localization and mapping in unfamiliar environments, has become a necessary capacity for autonomous mobile robots. Since it was first proposed in 1986, SLAM has attracted extensive attention from many researchers and developed rapidly in robotics, virtual reality, and other fields. SLAM refers to self-positioning based on location and map, and building incremental maps based on self-positioning. It is mainly used to solve the problem of robot localization and map construction when moving in an unknown environment [2]. SLAM, as a basic technology, has been applied to mobile robot localization and navigation in the early stage. With the development of computer technology (hardware) and artificial intelligence (software), robot research has received more and more attention and investment. Numerous researchers are committed to making robots more intelligent. SLAM is considered to be the key to promoting the real autonomy of mobile robots [3].

Some scholars divide SLAM into Laser SLAM and Visual SLAM (VSLAM) according to the different sensors adopted [4]. Compared with VSLAM, because of an early start, laser

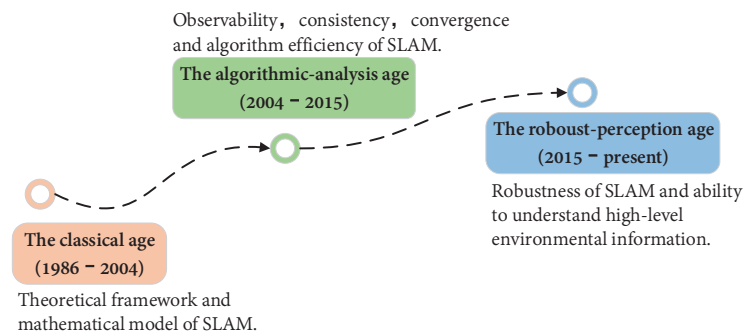
SLAM studies abroad are relatively mature and have been considered the preferred solution for mobile robots for a long time in the past. Similar to human eyes, VSLAM mainly uses images as the information source of environmental perception, which is more consistent with human understanding and has more information than laser SLAM. In recent years, camera-based VSLAM research has attracted extensive attention from researchers. Due to the advantages of cheap, easy installation, abundant environmental information, and easy fusion with other sensors, many vision-based SLAM algorithms have emerged [5]. VSLAM has the advantage of richer environmental information and is considered to be able to give mobile robots stronger perceptual ability and be applied in some specific scenarios. Therefore, this paper focuses on VSLAM and combs out the algorithms derived from it. SLAM based on all kinds of laser radar is not within the scope of discussion in this paper. Interested readers can refer to [6–8] and other sources in the literature.

As one of the solutions for autonomous robot navigation, traditional VSLAM is essentially a simple environmental understanding based on image geometric features [9]. Because traditional VSLAM only uses the geometric feature of the environment, such as points and lines, to face this low-level geometry information, it can reach a high level in real-time. Facing changes in lighting, texture, and dynamic objects are widespread, which shows the obvious shortage, in terms of position precision and robustness is flawed [10]. Although the map constructed by traditional visual SLAM includes important information in the environment and meets the positioning needs of the robot to a certain extent. It is inadequate in supporting the autonomous navigation and obstacle avoidance tasks of the robot. Furthermore, it cannot meet the interaction needs of the intelligent robot with the environment and humans [11].

People's demand for intelligent mobile robots is increasing day by day, which put forward a high need for autonomous ability and the human–computer interaction ability of robots [12]. The traditional VSLAM algorithm can meet the basic positioning and navigation requirements of the robot, but cannot complete higher-level tasks such as “help me close the bedroom door”, “go to the kitchen and get me an apple”, etc. To achieve such goals, robots need to recognize information about objects in the scene, find out their locations and build semantic maps. With the help of semantic information, the data association is upgraded from the traditional pixel level to the object level. Furthermore, the perceptual geometric environment information is assigned with semantic labels to obtain a high-level semantic map. It can help the robot to understand the autonomous environment and human–computer interaction [13]. We believe that the rapid development of deep learning provides a bridge for the introduction of semantic information into VSLAM. Especially in semantic map construction, combining it with VLAM can enable robots to gain high-level perception and understanding of the scene. It significantly improves the interaction ability between robots and the environment [14].

In 2016, Cadena et al. [15] first proposed to divide the development of SLAM into three stages. In their description, we are in a stage of robust perception, as shown in Figure 1. They describe the emphasis and contribution of SLAM in different times from three aspects: Classical, Algorithmic, and Robust. Ref. [16] summarizes the development of vision-based SLAM algorithms from 2010 to 2016 and provides a toolkit to help beginners. Yousif et al. [17] discussed the elementary framework of VSLAM and summarized several mathematical problems to help readers make the best choice. Bavle et al. [18] summarized the robot SLAM technology and pointed out the development trend of robot scene understanding. Starting from the fusion of vision and visual inertia, Servieres et al. [19] reviewed and compared important methods and summarized excellent algorithms emerging in SLAM. Azzam et al. [20] conducted a comprehensive study on feature-based methods. They classified the reviewed methods according to the visual features observed in the environment. Furthermore, they also proposed possible problems and solutions for the development of SLAM in the future. Ref. [21] introduces in detail the SLAM method based on monocular, binocular, RGB-D, and visual-inertial fusion, and gives the existing problems and future direction. Ref. [22] describes the opportunities and challenges of VSLAM from

geometry to deep learning and forecasts the development prospects of VSLAM in the future semantic era.



**Figure 1.** Overview of SLAM development era. The development of SLAM has gone through three main stages: theoretical framework, algorithm analysis, and advanced robust perception. The time points are not strictly limited, but rather represent the development of SLAM at a certain stage and the hot issues that people are interested in.

As you can see, there are some surveys and summaries of vision-based SLAM technologies. However, most of them only focus on one aspect of VSLAM, without a more comprehensive summary of the development of VSLAM. Furthermore, the above review focuses more on traditional visual SLAM algorithms, while semantic SLAM combined with deep learning is not introduced in detail. So, a comprehensive review of vision-based SLAM algorithms is necessary to help researchers and students launch their efforts at visual SLAM technologies to obtain an overview of this large field.

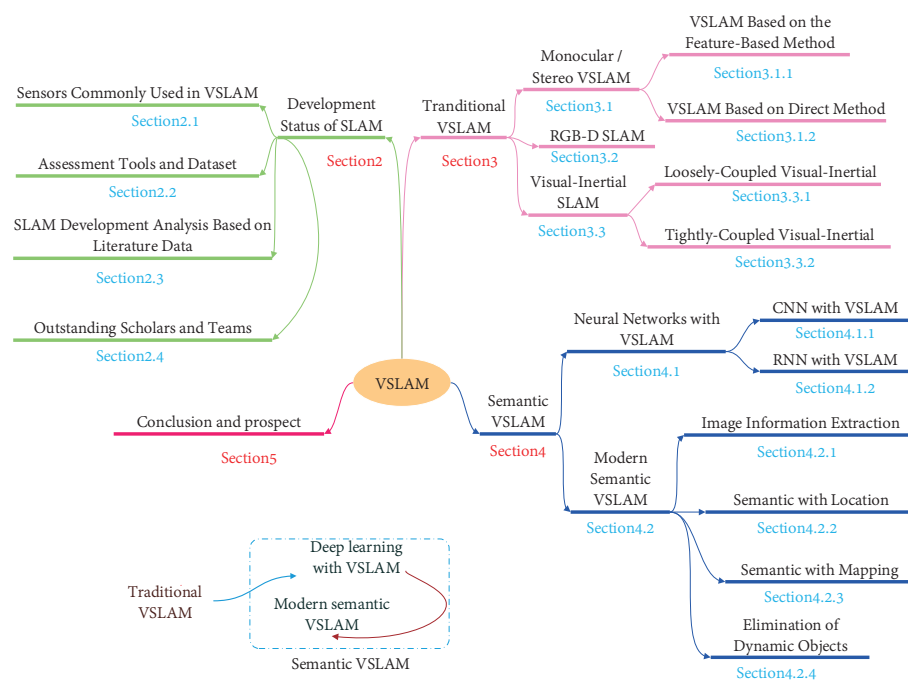
To give readers a deeper and more comprehensive understanding of the field of SLAM, we reviewed the history of general SLAM algorithms from inception to the present. In addition, we summarize the key solutions driving the technological evolution of SLAM solutions. The work of SLAM is described from the formation of point problems to the most commonly used state methods. Rather than focusing on just one aspect, we present the key main approaches to show the connections between the research that has brought the SLAM approach to its current state. In addition, we review the evolution of SLAM from traditional to semantic, a perspective that covers major, interesting, and leading design approaches throughout history. On this basis, we make a comprehensive summary of DEEP learning SLAM algorithms. Semantic VSLAM is also explained in detail to help readers better understand the characteristics of semantic VSLAM. We think our work can help readers better understand robot environment perception. Our work on semantic VSLAM can provide readers with a better idea and provide a useful reference for future SLAM research and even robot autonomous sensing. Therefore, this paper comprehensively supplements and updates the development of vision-based SLAM technology. Furthermore, this paper divides the development of vision-based SLAM into two stages: traditional VSLAM and semantic VSLAM integrating deep learning. So readers can better understand the research hot spots of VSLAM and grasp the development direction of VSLAM. We believe the traditional phase SLAM problem mainly solves the framework problem of the algorithm. In the semantic era, SLAM focuses on advanced situational awareness and system robustness in combination with deep learning.

Our review makes the following contributions to the state of the art:

- We have reviewed the development of vision-based SLAM more comprehensively, we review the recent research progress in the field of simultaneous localization and map construction based on environmental semantic information.
- Starting with a convolutional neural network (CNN) and a recurrent neural network (RNN), we describe the application of deep learning in VSLAM in detail. To our knowledge, this is the first review to introduce VSLAM from a neural network perspective.

- We describe the combination of semantic information and VSLAM in detail and point out the development direction of VSLAM in the semantic era. We mainly introduce and summarize the outstanding research achievements in the combination of semantic information and traditional visual SLAM in system localization and map construction, and make an in-depth comparison between traditional visual SLAM and semantic SLAM. Finally, the future research direction of semantic SLAM is proposed.

Specifically, in Section 1, this paper introduces the characteristics of traditional VSLAM in detail, including the direct method and the indirect method based on the front-end vision odometer, and makes a comparison between the depth camera-based VSLAM and the classical VSLAM integrated with IMU. In Section 2, this paper is divided into two parts. We firstly introduce the combination of deep learning and VSLAM from two neural networks, CNN and RNN. We believe that introducing deep learning into semantic VSLAM is the precondition for the development of semantic VSLAM. Furthermore, this stage can also be regarded as the beginning of semantic VSLAM. Then, this paper describes the process of deep learning leading semantic VSLAM to the advanced stage from the aspects of target detection and semantic segmentation. So this paper summarizes the development direction of semantic VSLAM from three aspects of localization, mapping, and elimination of dynamic objects. In Section 3, this paper introduces some mainstream SLAM data sets, and some outstanding laboratories in this area. In the end, we summarize the current research and point out the direction of VSLAM research in the future. The section table of contents for this article is shown in Figure 2.






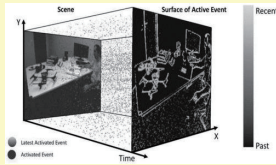
**Figure 2.** Structure diagram for the rest of this paper. This paper focuses on the second chapter of semantic VSLAM. We consider the introduction of neural networks as the beginning of semantic VSLAM. We start with a deep neural network, describe its combination with VSLAM, and then explain modern semantic VSLAM in detail from the aspects of object detection and semantic segmentation based on deep learning, and make a summary and prospect.

## 2. Development Status of SLAM

### 2.1. Sensors Commonly Used in VSLAM

The sensors used in the VSLAM typically include the monocular camera, stereo camera, and RGB-D camera. The monocular camera and the stereo camera have similar principles and can be used in a wide range of indoor and outdoor environments. As a special form of camera, the RGB-D camera can directly obtain image depth mainly by actively emitting

infrared structured light or calculating time-of-flight (TOF). It is convenient to use, but sensitive to light, and can only be used indoors in most cases [23]. Events camera appeared in recent years, a new camera sensor, a picture of a different from the traditional camera. Events camera is “events”, can be as simple as “pixel brightness change”. The change of events camera output is pixel brightness, SLAM algorithm based on the event camera is still only in the preliminary study stage [24]. In addition, as a classical SLAM system based on vision, visual-inertial fusion has achieved excellent results in many aspects. In Figure 3, we compare the main features of different cameras.

Camera	Advantage	Disadvantage
 <p>Monocular</p>	Simple structure, low cost, can be used indoor and outdoor.	Images alone cannot determine this true scale.
 <p>Stereo</p>	The farther the distance that can be measured; it can be used indoors and outdoors.	Parallax calculation is very resource-intensive.
 <p>RGB-D</p>	Can provide richer information, and also does not need to be as time-consuming or binocular depth calculation.	Narrow measuring range, large noise, small field of vision, susceptible to sunlight interference.
 <p>Event</p>	Event camera has the advantages of low delay, high dynamic range (HDR), no motion blur, very low power consumption, and low data bandwidth.	Single event has little effective information and sparse and incomplete data.

**Figure 3.** Comparison between different cameras. An event camera is not a specific type of camera, but a camera that can obtain “event information”. “Traditional cameras” work at a constant frequency and have natural drawbacks, such as lag, blurring, and overexposure when shooting high-speed objects. However, the event camera, a neuro-based method of processing information similar to the human eye, has none of these problems.

## 2.2. Assessment Tools and Dataset

SLAM problems have been around for decades. In the past few decades, many excellent algorithms have emerged, each of which has contributed to the rapid development of SLAM technology to varying degrees, despite its different focus. Each algorithm has to be compared fairly. Generally speaking, we can evaluate a SLAM algorithm from multiple perspectives such as time consumption, complexity, and accuracy. However, the most important one is that we pay the most attention to its accuracy. ATE (Absolute Trajectory Error) and RPE (Relative Pose Error) are the two most important indicators used to evaluate the accuracy of SLAM. The relative pose error is used to calculate the difference of pose changes in the same two-time stamps, which is suitable for estimating system drift. The absolute trajectory error directly calculates the difference between the real value of the camera pose and the estimated value of the SLAM system. The basic principles of ATE and RPE are as follows.

Assumptions: The given pose estimate is  $\Delta$ . The subscript represents time  $t$  (or frame), where it is assumed that the time of each frame of the estimated pose and the real pose are aligned, and the total number of frames is the same.

ATE: The absolute trajectory error is the direct difference between the estimated pose and the real pose, which can directly reflect the accuracy of the algorithm and the global trajectory consistency. It should be noted that the estimated pose and ground truth are usually not in the same coordinate system, so we need to pair them first: For stereo SLAM and RGB-D SLAM, the scale is uniform, so we need to calculate a transformation matrix from the estimated pose to the real pose by the least square method  $S \in SE(3)$ . For monocular cameras with scale uncertainties, we need to calculate a similar transformation matrix  $S \in Sim(3)$  from the estimated pose to the real pose. So the ATE of frame  $i$  is defined as follows:

$$F_i := Q_i^{-1}SP_i \quad (1)$$

Similar to RPE, RMSE is recommended for ATE statistics.

$$RMSE(F_{1:n}, \Delta) := \left( \frac{1}{m} \sum_{i=1}^m \|\text{trans}(F_i)\|^2 \right)^{\frac{1}{2}} \quad (2)$$

RPE: Relative pose error mainly describes the accuracy (compared with real pose) of two frames separated by a fixed time difference  $\Delta$ , which is equivalent to the error of the odometer directly measured. So the RPE of the frame  $I$  is defined as follows:

$$E_i := (Q_i^{-1}Q_{i+\Delta})^{-1}(P_i^{-1}P_{i+\Delta}) \quad (3)$$

Given the total number  $n$  and the interval  $\Delta$ , we can obtain  $(m = n - \Delta)$  RPE. Then we can use the root mean square error RMSE to calculate this error and obtain a population value:

$$RMSE(E_{1:n}, \Delta) = \left( \frac{1}{m} \sum_{i=1}^m \|\text{trans}(E_i)\|^2 \right)^{\frac{1}{2}} \quad (4)$$

$\text{trans}(E_i)$  represents the translation part of the relative pose error. We can evaluate the performance of the algorithm from the size of the RMSE value. However, in practice, we find that there are many choices for the selection of  $\Delta$ . To comprehensively measure the performance of the algorithm, we can calculate the average RMSE traversing all  $\Delta$ :

$$RMSE = (E_{1:n}) = \frac{1}{n} \sum_{\Delta=1}^n RMSE(E_{1:n}, \Delta) \quad (5)$$

EVO [25] is an evaluation tool for the Python version of the SLAM system that can be used with a variety of data sets. In addition to ATE and RPE, data can be obtained, it can also draw a comparison diagram of the test algorithm and real trajectory. Is a very convenient assessment kit. SLAMBench2 [26] is a publicly available software framework that evaluates current and future SLAM systems through an extensible list of data sets. It includes open and closed source code while using a comparable and specified list of performance metrics. It supports a variety of existing SLAM algorithms and datasets, such as ElasticFusion [27], ORB-SLAM2 [28], and OKVIS [29], and integrating new SLAM algorithms and datasets are straightforward.

In addition, we also need to use datasets to test specific visualization of the algorithm. Common data sets used to test various aspects of SLAM performance are illustrated in Table 1. TUM data sets mainly include multi-view data sets, 3D object recognition and segmentation, scene recognition, 3D model matching, VSALM, and other data in various directions. According to the direction applied, it can be divided into TUM RGB-D [30], TUM MonoVO [31], and TUM VI [32]. Among them, the TUM RGB-D data set mainly contains indoor images with real ground tracks. Furthermore, it provides two measures to evaluate local accuracy and global consistency of orbit, namely relative attitude error and absolute trajectory error. TUM MonoVO is used to assess monocular systems, which

contain both indoor and outdoor images. Due to the variety of scenarios, ground authenticity is not available, but rather large sequences with the same starting position are performed, allowing evaluation of cyclic drift. TUM VI is employed to the evaluation of the visual-inertial odometer. The KITTI [33] dataset is a famed outdoor environment data set jointly founded by the Karlsruhe Institute of Technology and Toyota American Institute of Technology. It is the largest computer vision algorithm evaluation data set in the world under autonomous driving scenarios, including monocular vision, binocular vision, Velodyne Lidar, POS trajectory, etc. It is the most widely used outdoor data set. The EuRoc [34] dataset A visual inertia data set developed by ETH Zurich. Cityscapes [35] is a dataset related to autonomous driving, focusing on pixel-level scene segmentation and instance annotation. In addition, many datasets are used in various scenarios, such as ICL-NUIM [36], NYU RGB-D [37], MS COCO [38], etc.

**Table 1.** Common open-source datasets for SLAM.

Dataset	Sensor	Environment	Ground-Truth	Availability	Development
Cityscapes	Stereo	Outdoor	GPS	[35]	Daimler AG R&D, Max Planck Institute for Informatics, TU Darmstadt Visual Inference Group
KITTI	Stereo/3D laser scanner	Outdoor	GPS/INS	[33]	Karlsruhe Institute of Technology and Toyota American Institute of Technology
TUM RGB-D	RGB-D	Indoor	Motion capture	[30]	
TUM MonoVO	Monocular	Indoor/Outdoor	Loop drift	[31]	Technical University of Munich
TUM VI	Stereo/IMU	Indoor/Outdoor	Motion capture	[32]	
EuRoc	Stereo/IMU	Indoor	Station/Motion capture	[34]	Eidgenössische Technische Hochschule Zürich
ICL-NUIM	RGB-D	Indoor	3D surface model SLAM estimation	[36]	Imperial College London

### 2.3. SLAM Development Analysis Based on Literature Data

Since the advent of SLAM, it has been widely used in the field of robotics. As shown in Figure 4, this paper selected about 1000 hot articles related to mobile robots in the last two decades and made this keyword heat map. The larger the circle is, the higher the frequency of the keyword appears. The circle layer shows the time from the past to the present from the inside out, and the redder the color, the more attractive it is. Connecting lines indicate that there is a connection between different keywords (data from the Web of Science Core Collection). As shown in Figure 5, the number of citations of visual SLAM and semantic SLAM-related papers is increasing rapidly. Especially around 2017, visual SLAM and semantic SLAM saw their citations skyrocket. Many advances have been made in traditional VSLAM research. To enable robots to perceive the surrounding environment from a higher level, the research of semantic VSLAM has received extensive attention. Semantic SLAM has attracted more and more attention in recent years. Furthermore, as shown in Figure 6, this paper has selected about 5000 articles from the Web of Science Core Collection. Judging from the titles of journals about SLAM published, SLAM is a topic of interest in robotics.

As can be seen from the above data, SLAM research has always been a hot topic. With the rapid development of deep learning, the field of computer vision has made unprecedented progress. Therefore, VSLAM also ushered in a period of rapid development. Combining semantic information with VSLAM is going to be a hot topic for a long time. The development of semantic VSLAM can make robots truly autonomous.

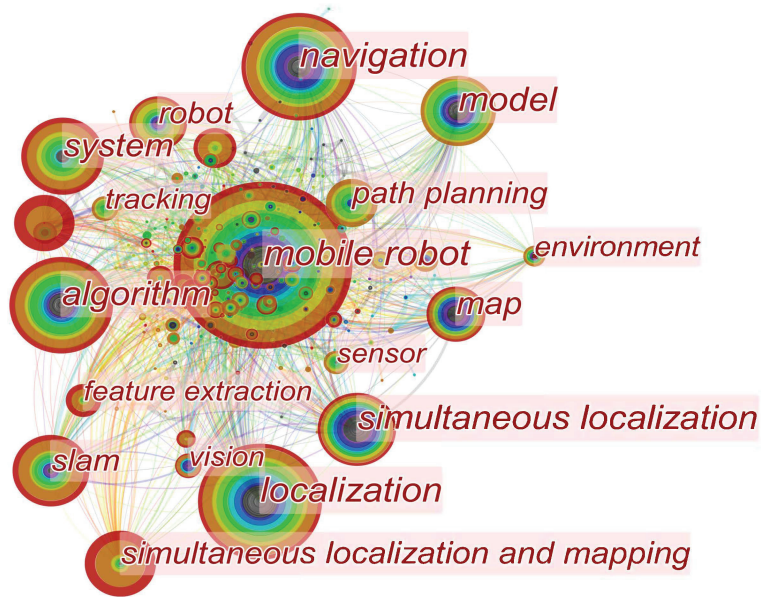


Figure 4. Hot words in mobile robot field.

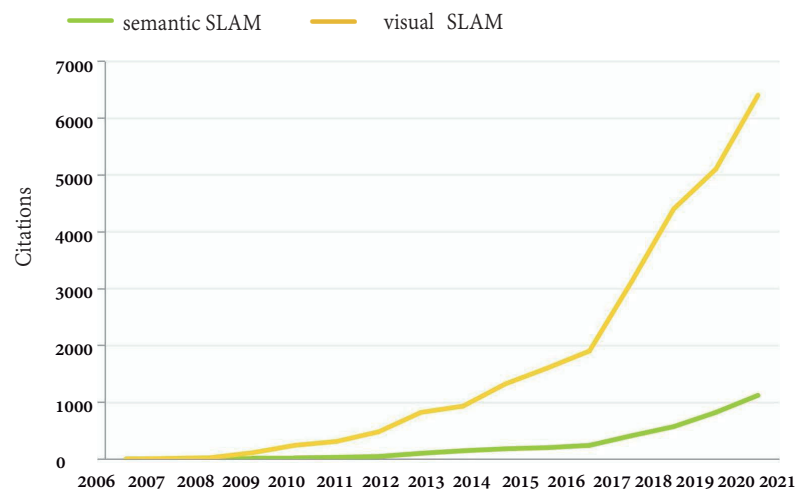


Figure 5. Citations for Web of Science articles on visual SLAM and semantic SLAM in recent years (Data are as of December 2021).

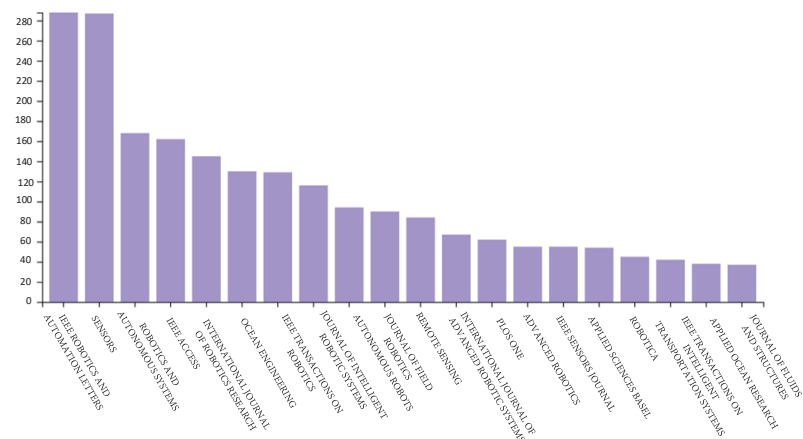


Figure 6. Publication titles about SLAM on Web of Science.



#### 2.4. Outstanding Scholars and Teams

In addition, many scholars and teams have made indelible contributions to the research of SLAM. As shown in Figure 7, we analyzed approximately 4000 articles from 2000 to 2022 (data from the Web of Science website). A larger font indicates that the author has received the most attention, and vice versa. The countries to which they belong are presented in Figure 8. The computer vision group at the Technical University of Munich in Germany is a leader in this field. The team published a variety of classic visual SLAM solutions such as DSO [39] and LSD-SLAM [40], which improved the performance of all aspects of visual SLAM. The robotics and Perception group at the University of Zurich, Switzerland, also contributed to the rapid development of SLAM technology by developing SVO and VO/VIO trajectory assessment tools. In addition, the Computer Vision and Ensemble Laboratory of ETH Zurich has also made a lot of efforts in this field. Furthermore, they have made a lot of breakthrough progress in the field of visual semantic localization in large-scale outdoor mapping. The LABORATORY of ROBOTICS, Sensing, and Real-time Group SLAM at the University of Zaragoza in Spain is one of the biggest contributors to the development of SLAM. The ORB-SLAM series launched by the laboratory is a landmark scheme in visual SLAM, which has a far-reaching influence on the research of SLAM. In addition, the efforts of many scholars and teams have promoted the rapid development of visual semantic SLAM and laid a foundation for solving various problems in the future. Table 2 shows the works of some excellent teams and their team websites for your reference, you can check the website of the team by the number quoted after its name.

Some scholars have made outstanding contributions to semantic VSLAM research. Niko Sünderhauf [41] and their team, for example, have made many advances in robot scene understanding and semantic VSLAM. The team is dedicated to making a robot understand what it sees is one of the most fascinating goals. To this end, they develop novel methods for Semantic Mapping and Semantic SLAM by combining object detection with simultaneous localization and mapping (SLAM) techniques. The team [42] of researchers is part of the Australian Centre for Robotic Vision and is based at the Queensland University of Technology in Brisbane, Australia. They work on novel approaches to SLAM (Simultaneous Localization and Mapping) that create semantically meaningful maps by combining geometric and semantic information. We believe such semantically enriched maps will help robots understand our complex world and will ultimately increase the range and sophistication of interactions that robots can have in domestic and industrial deployment scenarios.

**Table 2.** Some great teams and their contributions.

Team	Works
The Dyson Robotics Lab at Imperial College [43]	Code-SLAM [44], ElasticFusion [27], Fusion++ [45], SemanticFusion [46]
Computer Vision Group TUM Department of Informatics Technical University of Munich [47]	D3VO [48], DM-VIO [49], LSD-SLAM [40], LDSO [50], DSO [39]
Autonomous Intelligent Systems University of Freiburg [51]	Gmapping [52], RGB-D SLAMv2 [53]
HKUST Aerial Robotics Group [54]	VINS-Mono [55], VINS-Fusion [56], Event-based stereo visual odometry [57]
UW Robotics and State Estimation Lab [58]	DART [59], DA-RNN [60], RGB-D Mapping [61]
Robotics, Perception and Real Time Group UNIVERSIDAD DE ZARAGOZA [62]	ORB-SLAM2 [28], Real-time monocular objects slam [63]

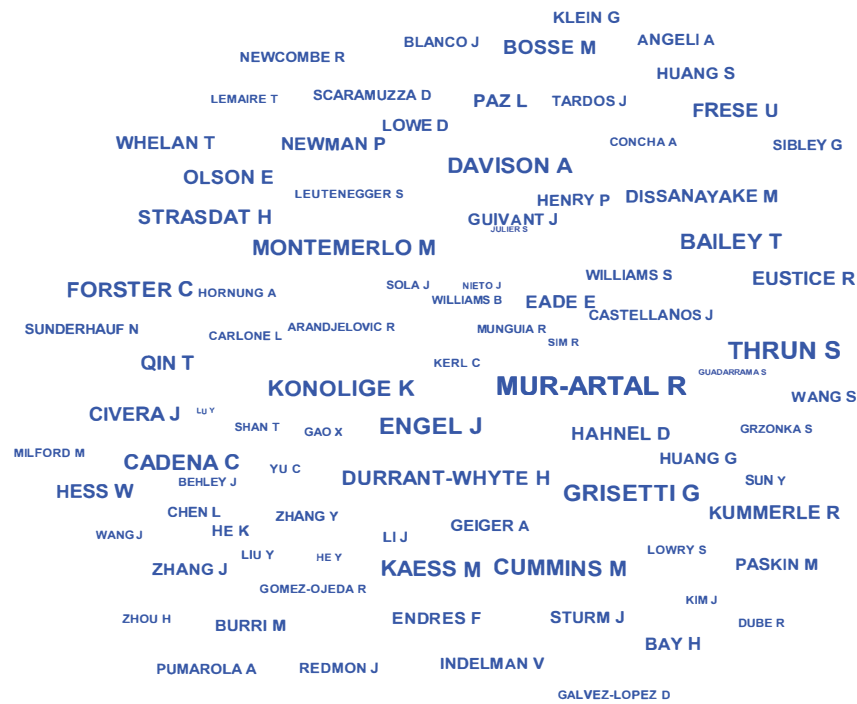


Figure 7. A distinguished scholar in the field of visual SLAM.

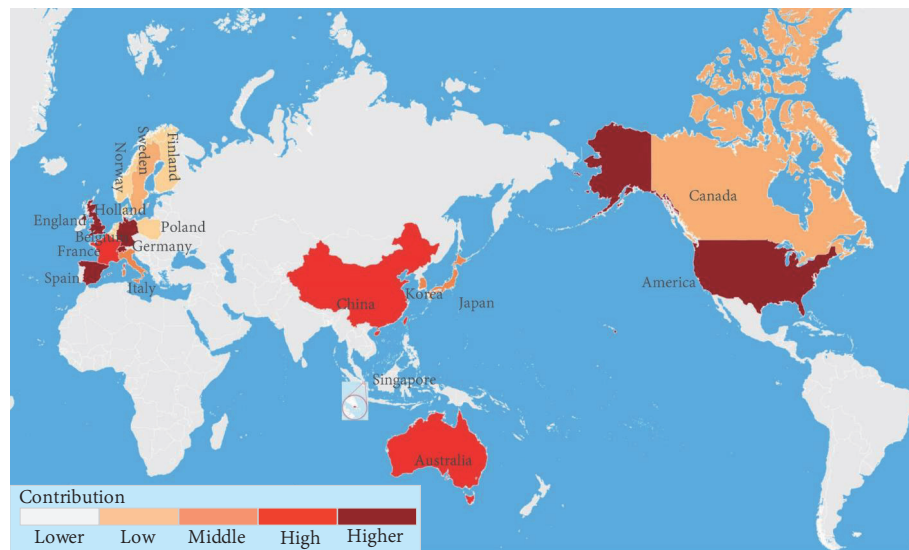
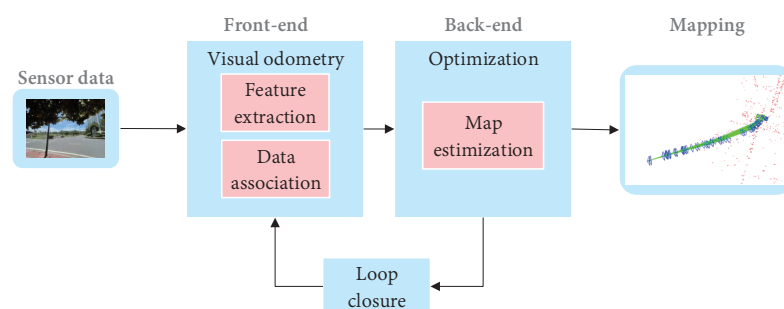


Figure 8. Contribution of different countries in the SLAM field (Colors from light to dark indicate contributions from low to high).

### 3. Traditional VSLAM

Cadena et al. [15] proposed a classical VSLAM framework, which mainly consists of two parts: front-end and back-end, as shown in Figure 9. The front end provides real-time camera pose estimation, while the back end provides map updates and optimizations. Specifically, mature visual SLAM systems include sensor data collection, front-end visual odometer, back-end optimization, loop closure detection, and map construction modules [64].



**Figure 9.** The typical visual SLAM system framework.

### 3.1. Monocular/Stereo VSLAM

In this section, we will elaborate on the VSLAM algorithm based on monocular or stereo cameras. For the VSLAM system, the visual odometer, as the front-end of SLAM, is an indispensable part [65]. Ref. [20] points out that VSLAM can be divided into the direct method and indirect method according to the different image information collected by the front-end visual odometer. The indirect method needs to select a certain number of representative points from the collected images, called key points, and detect and match them in the following images to gain the camera pose. It not only saves the key information of the image but also reduces the amount of calculation, so it is widely used. The direct method uses all the information of the image without preprocessing and directly operates on pixel intensity, which has higher robustness in an environment with sparse texture [66]. Both the indirect method and direct method have been widely concerned and developed to different degrees.

#### 3.1.1. VSLAM Based on the Feature-Based Method

The core of indirect VSLAM is to detect, extract and match geometric features (points, lines, or planes), estimate camera pose, and build an environment map while retaining important information, it can effectively reduce calculation, so it has been widely used [67]. The VSLAM method based on point feature has long been taken into account as the mainstream method of indirect VSLAM due to its simplicity and practicality [68].

Feature extraction mostly adopted corner extraction methods in the early, such as Harris [69], FAST [70], GFTT [71], etc. However, in many scenarios, simple corners cannot provide reliable features, which prompts researchers to seek more stable local image features. Nowadays, typical VSLAM methods based on point features firstly use feature detection algorithms, such as SIFT [72], SURF [73], and ORB [74], to extract key points in the image for matching. Then gain pose after minimizing reprojection error. Feature points and corresponding descriptors in the image are employed for data association. Furthermore, data association in initialization is completed through the matching of feature descriptors [75]. In Table 3, we list common traditional feature extraction algorithms and compare their main performance to help readers have a more comprehensive understanding.

**Table 3.** Comparison table of commonly used feature extraction algorithms.

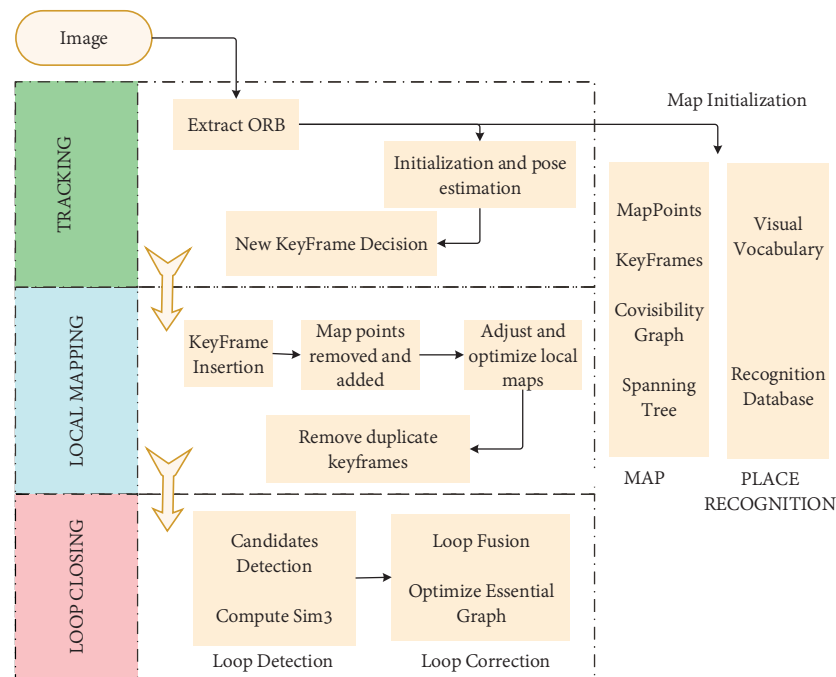
Method	Year	Type	Speed	Rotation Invariance	Scale Invariance	Illumination Invariance	Anti-Invariance
ORB [74]	2011	Point	High	Yes	Yes	Yes	Stronger
SURF [73]	2008	Point	Middle	Yes	Yes	No	Week
FAST [70]	2006	Point	High	No	Yes	No	Week
SIFT [72]	2004	Point	Low	Yes	Yes	Yes	Strong
Shi-Tomasi [71]	1994	Coner	Middle	Yes	No	Yes	Week
Harris [69]	1988	Coner	Low	Yes	No	Yes	Week
LSD [76]	2010	Line	Middle	Yes	Yes	Yes	Stronger

Davidson et al. [77] proposed MonoSLAM in 2007. This algorithm is considered to be the first real-time monocular VSLAM algorithm, which can achieve real-time drift free-motion structure recovery. The front end tracks the sparse feature points Shi-Tomasi corner point for feature point matching, and the back end uses Extended Kalman Filter (EKF) [78] for optimization, which can build the sparse environment map online in real-time. This algorithm has a milestone significance in SLAM research, but the EKF method leads to a square growth between storage and state quantity, so it is not suitable for large-scale scenarios. In the same year, the advent of PTAM [79] improved MonoSLAM's inability to work steadily for long periods in a wide range of environments. PTAM, as the first SLAM algorithm using nonlinear optimization at the back end, solves the problem of fast data growth in the filter-based method. Furthermore, it separated tracking and mapping into two different threads for the first time. The front end uses FAST corner detection to extract and estimate camera motion using image features, and the back end is responsible for nonlinear optimization and mapping. It not only ensures the real-time performance of SLAM in the calculation of camera pose but also ensures the accuracy of the whole SLAM system. However, because there is no loopback detection module, it will accumulate errors during long-running.

In 2015, Mur-Artal et al. proposed the ORB-SLAM [80]. This algorithm is regarded as the excellent successor of PTAM, and based on PTAM, added a loop closure detection module, which effectively reduces the cumulative error. As a real-time monocular visual SLAM system that uses ORB feature matching, the whole process is carried out around ORB features. As shown in Figure 10, the three threads of tracking, local mapping, and loop closure detection are used innovatively. In addition, the loop closure detection thread uses the word bag model DBoW [81] for loop closure. The loop closure method based on the BoW model can detect the loop closure quickly by detecting the image similarity. Furthermore, achieve good results in the processing speed and the accuracy of map construction. In later years, the team launched ORB-SLAM2 [28] and ORB-SLAM3 [82]. The ORB-SLAM family is one of the most widely used visual SLAM solutions due to its real-time CPU performance and robustness. However, the ORB-SLAM series relies heavily on environmental features, so it may be difficult to obtain enough feature points in an environment without texture features.

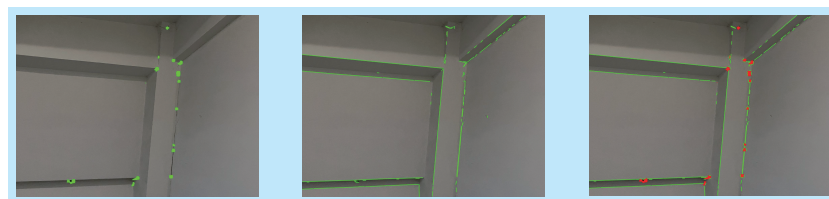
The point feature-based SLAM system relies too much on the quality and quantity of point features. It is difficult to detect enough feature points in weak texture scenes, such as corridors, windows, white walls, etc. Thus, affecting the robustness and accuracy of the system and even leading to tracking failure. In addition, due to the rapid movement of the camera, illumination changes, and other reasons, the matching quantity and quality of point features will decline seriously. To improve the feature-based SLAM algorithms, the application of line features in SLAM systems has attracted more and more attention [83]. The commonly used line feature extraction algorithm is LSD [76].

In recent years, with the improvement of computer computing capacity, VSLAM-based online features have also been developed rapidly. Smith et al. [84] proposed a monocular VSLAM algorithm-based online feature extraction in 2006. Lines are represented by two endpoints, and line features are used in the SLAM system to detect and track the two endpoints of lines in small scenes. The system can use line features alone or in combination with point-line features, which is of groundbreaking significance in VSLAM research. In 2014, Perdices et al. proposed LineSLAM, a line-based monocular SLAM algorithm [85]. For line extraction, this scheme adopts the line extraction scheme in [86]. It detects the lines every time the keyframes are acquired. Then uses the Unscented Kalman Filter (UKF) to predict the current camera state and vector probability distribution of the ground line. Then, matches the line prediction result with the detected lines. Because the scheme has no loop closure and the line segment is of infinite length instead of finite length, it is difficult to be used in practice.



**Figure 10.** Flow chart of ORB-SLAM.

As shown in Figure 11, compared with point feature or line feature alone, the combination of point feature and line feature increases the number of feature observations and data association. Furthermore, line feature is less sensitive to light changes than the point feature, which improves the positioning accuracy and robustness of the original system [76]. In 2016, Klein et al. [87] adopted the method of point-line fusion to improve the tracking failure of the SLAM system due to image blur caused by fast camera movement. In 2017, Pumarola et al. [88] published monocular PL-SLAM, and Gomez-Ojeda et al. [89] published stereo PL-SLAM in the same year. Based on ORB-SLAM, the two algorithms use the LSD detection algorithm to detect line features and then combine the point-line features in each link of SLAM. It can work even when most of the point features disappear. Furthermore, it improves the accuracy, robustness, and stability of the SLAM system, but the real-time performance is not good.



**Figure 11.** Comparison of point and line feature extraction in a weak texture environment. From left to right are ORB point feature extraction, LSD line feature extraction, and point-line combination feature extraction.

In addition, in some environments, there are some obvious surface features, which have aroused great interest of some researchers. Ref. [90] proposed a map construction method combining planes and lines. By introducing surface features into the real-time VSLAM system, the errors are reduced and the system robustness is improved by combining low-level features. In 2017, Li et al. [91] proposed a VSLAM algorithm based on point, line, and plane fusion for an artificial environment. Point features are used for the initial estimation of the robot's current pose. Lines and planes are used to describe the environment. However, most planes only exist in the artificial environment, and few suitable planes can be found in the natural environment. These limit its application range.

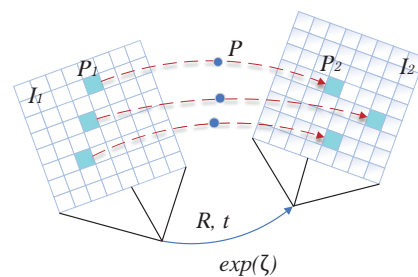
Compared with the methods that rely only on point features, SLAM systems that rely only on lines or planes can only work stably in artificial environments in most cases. The VSLAM method combining point, line, and surface features improve the localization accuracy and robustness in weak texture scenes, illumination changes, and fast camera movement. However, the introduction of line and surface features increases the time consumption of feature extraction and matching, which reduces the efficiency of the SLAM system. Therefore, the VSLAM algorithm based on the point feature still occupies the mainstream position [92]. Table 4 shows a comparison of geometric features.

**Table 4.** Comparison table of geometric features.

Feature	Benefits	Disbenefits
Point	Is the most popular and commonly used feature, easy to store and match, and the speed is generally faster.	It is difficult to extract sufficient features in an environment of intense light and rapid camera rotation.
Line	It has natural lighting and viewing Angle invariance, while more advanced features also improve tracking robustness and accuracy. Especially in certain artificial scenes (indoor, corridor), the interference of untextured or unreliable textures can be overcome.	The detection and matching time of the line segment is longer than that of the feature point. There is also no standard, universal SLAM optimization and loopback module on the back end. Line feature matching is also difficult, for example, line segments are easy to fracture, do not have strong geometric constraints (such as polar line geometric constraints), and do not have strong identification of texture missing places.
Plane	It has a more stable effect in artificial environments.	The range is small and can only be operated in certain artificial environments.

### 3.1.2. VSLAM Based on Direct Method

Different from feature-based methods, the direct method operates directly on pixel intensity and can retain all information about the image. Furthermore, the direct method cancels the process of feature extraction and matching, so the computational efficiency is better than the indirect method. Furthermore, it has good adaptability to the environment with complex textures. It can still keep a good effect in the environment with missing features [93]. The direct method is similar to the optical flow, and they both have a strong assumption: gray-level invariance, the principle of which is shown in Figure 12.



\$P\_1\$ and \$P\_2\$ are the pixel coordinates of point \$P\$ on the two images. Using the first camera as a reference frame, the rotation and translation of the second camera are \$R, t\$ (the corresponding lie group is \$T\$).

The brightness error of two pixels of space point \$P\$ is:

$$e = I_1(p_1) - I_2(p_2)$$

Here \$e\$ is a scalar. The optimization objective is the binary norm of the error, and the unweighted form is temporarily taken.

$$\min_T J(T) = \|e\|^2$$

If there are multiple points \$P\_i\$ in the space, the whole camera pose estimation problem becomes :

$$\min_T J(T) = \sum_{i=1}^N e_i^T e_i,$$

$$e_i = I_1(p_{1,i}) - I_2(p_{2,i}).$$

**Figure 12.** Schematic diagram of the direct method.

In 2011, Newcombe et al. [94] proposed the DTAM algorithm, which was considered the first practical direct method of VSLAM. DTAM allows tracking by comparing the input images with those created by reconstructed maps. The algorithm performs a precise and detailed reconstruction of the environment. However, it affects the computational cost of storing and processing the data, so it can only run in real-time on GPU. LSD-SLAM [40] neglects texture-free areas to improve operational efficiency and can run in real-time on CPU. LSD-SLAM, another major approach indirect method, combines featureless extraction with semi-dense reconstruction, and its core is a visual odometer using semi-dense reconstruction. The algorithm consists of three steps: tracking, depth estimation, and map optimization. Firstly, the photometric error is minimized to estimate the sensor pose. Secondly, select a keyframe for in-depth estimation. Finally, in the map optimization step, the new keyframe is merged into the map and optimized by using the posture optimization algorithm. In 2014, Forster et al. [95] proposed the semi-direct visual SLAM algorithm SVO. Since the algorithm does not need to extract features for each frame, it can run at high frame rates, which enables it to run in low-cost embedded systems [80]. SVO combines the advantages of the feature point method and direct method. The algorithm is divided into two main threads: motion estimation and mapping. Motion estimation is carried out by feature point matching, but mapping is carried out by the direct method. SVO has good results, but as a purely visual method, it only performs short-term data association, which limits its accuracy [82]. In 2018, Engel et al. [39] proposed DSO. DSO can effectively use any image pixel, which makes it robust even in featureless regions and can gain more accurate results than SVO. DSO can calculate accurate camera attitude in poor feature point detector performance, improving the robustness of low-texture areas or blurred images. In addition, the DSO uses both geometric and photometric camera calibration results for high accuracy estimation. However, DSO only considers local geometric consistency, so it inevitably produces cumulative errors. Furthermore, it is not a complete SLAM because it does not include loop closure, map reuse, etc.

Up to now, VSLAM has made many achievements in direct and indirect methods. Table 5 compares the advantages and disadvantages of the direct method and the indirect method to help readers better understand.

**Table 5.** Comparison between direct method and indirect method.

Method	Benefits	Shortcomings
Indirect	(1) The feature point itself is not sensitive to light, motion, and rotation, so it is relatively stable. (2) The camera moves faster (relatively direct method) and can track successfully, with better robustness. (3) The research time is long and the scheme is mature.	(1) It takes a long time to extract, describe and match key points. (2) The feature point loss scenario cannot be used. (3) Only sparse maps can be constructed.
Direct	(1) Fast speed, can save the calculation of feature points, and descriptors time. (2) It can be used in situations where features are missing (such as white walls), and the feature point method will deteriorate rapidly in this case. (3) Semi-dense and even dense maps can be constructed.	(1) Since the gray level is assumed to be unchanged, it is susceptible to the change in illumination. (2) Slow camera movement or high sampling frequency is required (can be improved by image pyramid). (3) The differentiation of single-pixel or pixel blocks is not strong, and the strategy of quantity instead of quality is adopted.

### 3.2. RGB-D SLAM

An RGB-D camera is a visual sensor launched in recent years. It can simultaneously collect environmental color images and depth images, and directly gain depth maps mainly by actively emitting infrared structured light or calculating time-of-flight (TOF) [96]. The RGB-D camera, as a special camera, can gain three-dimensional information in space

more conveniently. So it has been widely concerned and developed in three-dimensional reconstruction [97].

KinectFusion [98] is the first real-time 3D reconstruction system based on an RGB-D camera. It uses a point cloud created by the depth to estimate the camera pose through ICP (Iterative Closest Point). Then splices multi-frame point cloud collection based on the camera pose, and expresses reconstruction result by the TSDF (Truncated signed distance Function) model. The 3D model can be constructed in real-time with GPU acceleration. However, the system has not been optimized by loop closure. Furthermore, there will be obvious errors in long-term operation, and the RGB information of the RGB-D camera has not been fully utilized. In contrast, ElasticFusion [27] makes full use of the color and depth information of the RGB-D camera. It estimates the camera pose by the color consistency of RGB and estimates the camera pose by ICP. Then improves the estimation accuracy of the camera pose by constantly optimizing and reconstructing the map. Finally, the surfel model was used for map representation, but it could only be reconstructed in a small indoor scene. Kinitinuous [99] adds loop closure based on KinectFusion and makes non-rigid body transformation for 3d rigid body reconstruction by using a deformation graph for the first time. So it makes the results of two-loop closure reconstruction overlap, achieving good results in an indoor environment. Compared with the above algorithms, RGB-D SLAMv2 [53] is a very excellent and comprehensive system. It includes image feature detection, optimization, loop closure, and other modules, which are suitable for beginners to carry out secondary development.

Although the RGB-D camera is more convenient to use, the RGB-D camera is extremely sensitive to light. Furthermore, there are many problems with narrow, noisy, and small horizons, so most of the situation is only used in the room. In addition, the existing algorithms must be implemented using GPU. So the mainstream traditional VSLAM system still does not use the RGB-D camera as the main sensor. However, in three-dimensional reconstruction in the interior, the RGB-D camera is widely used. In addition, because of the ability to build a dense environment map, the semantic VSLAM direction, RGB-D camera is widely used. Table 6 shows the classic SLAM algorithm based on RGB-D cameras.

**Table 6.** Some SLAM algorithms for sensors with an RGB-D camera.

Method	Year	Camera Tracking	Loop Closure	Code Resource
KinectFusion [98]	2011	Direct	No	[100]
Kinitinuous [99]	2012	Direct	Yes	[101]
RGB-D SLAMv2 [53]	2013	Indirect	Yes	[102]
ElasticFusion [27]	2016	Direct	Yes	[103]
DVO-SLAM [104]	2017	Direct	Yes	[105]
BundleFusion [106]	2017	Hybrid	Yes	[107]
RGBDTAM [108]	2017	Direct	Yes	[109]

### 3.3. Visual-Inertial SLAM

The pure visual SLAM algorithm has achieved many achievements. However, it is still difficult to solve the effects of image blur caused by fast camera movement and poor illumination by using only the camera as a single sensor [110]. IMU is considered to be one of the most complementary sensors to the camera. It can obtain accurate estimation at high frequency in a short time, and reduce the impact of dynamic objects on the camera. In addition, the camera data can effectively correct the cumulative drift of IMU [111]. At the same time, due to the miniaturization and cost reduction of cameras and IMU, visual-inertial fusion has also achieved rapid development. Furthermore, it become the preferred method of sensor fusion, which is favored by many researchers [112]. Nowadays, visual-inertial fusion can be divided into loosely coupled and tightly coupled according to whether image feature information is added to the state vector [113]. Loosely coupled means the IMU and the camera estimate their motion, respectively, and then fuse their pose estimation. Tightly coupled refers to the combination of the state of IMU and the state of



the camera to jointly construct the equation of motion and observation, and then perform state estimation [114].

### 3.3.1. Loosely Coupled Visual-Inertial

The loosely coupled core is to fuse the positions and poses calculated by the vision sensor and IMU, respectively. The fusion has no impact on the results obtained by the two. Generally, the fusion is performed through EKF. Stephen Weiss [115] provided groundbreaking insights in their doctoral thesis. Ref. [116] proposed an efficient loose coupling method, and good experimental results were obtained by using an RGB-D camera and IMU. The loose-coupling implementation is relatively simple, but the fusion result is prone to error and there has been little research in this area.

### 3.3.2. Tightly Coupled Visual-Inertial

The core of the tightly coupled is to combine the states of the vision sensor and IMU through an optimized filter. It needs the image features to be added to the feature vector to jointly construct the motion equation and observation equation. Then perform state estimation to obtain the pose information. Tightly coupled needs full use of visual and inertial measurement information, which is complicated in method implementation but can achieve higher pose estimation accuracy. Therefore, it is also the mainstream method, and many breakthroughs have been made in this area.

In 2007, Mourikis et al. [117] proposed MSCKF. The core of MSCKF is to fuse IMU and visual information under the EKF in a tightly coupled way. Compared with the VO algorithm alone, MSCKF can adapt to more intense motion and texture loss, with higher robustness. Speed and accuracy can also reach a high level. MSCKF has been widely used in robot, UAV, and AR/VR fields. However, because the backend uses the Kalman filter method, global information cannot be used for optimization, and no loopback detection will cause error accumulation. Ref. [29] proposed OKVIS based on a fusion of binocular vision and IMU. However, it only outputs six degrees of freedom pose without loopback detection and map, so it is not a complete SLAM in a strict sense. Although it has good accuracy, its pose will be loose when it runs for a long time. Although these two algorithms have achieved good results, they have not been widely promoted. The lack of loop closure modules inevitably leads to cumulative errors when running for long periods of time.

The emergence of VINS-Mono [55] broke this situation. In 2018, a team from The Hong Kong University of Science and Technology (HKUST) introduced a monocular inertially tightly coupled VINS-Mono algorithm. It has since released its expanded version, Vins-Fusion, which supports multi-sensor integration, including Monocular + IMU, Stereo + IMU, and even stereo only, and also provides a version with GPS. VINS-mono is a classic fusion of vision and IMU. Its positioning accuracy is comparable to OKVIS, and it has a more complete and robust initialization and loop closure detection process than OKVIS. At the same time, VINS-Mono has set a standard for the research and application of visual SLAM, which is more monocular +IMU. In the navigation of robots, especially the autonomous navigation of UAVs, monocular cameras are not limited by RGB-D cameras (susceptible to illumination and limited depth information) and stereo cameras (occupying a large space). It can adapt to indoor, outdoor and different illumination environments with good adaptability.

As a supplement to cameras, inertial sensors can effectively solve the problem that a single camera cannot cope with. Visual inertial fusion is bound to become a long-term hot direction of SLAM research. However, the introduction of multiple sensors will lead to an increase in data, which has a high requirement on computing capacity [118]. Therefore, we believe that the next hot issue of visual-inertial fusion will be reflected in the efficient processing of sensor fusion data. How to make better use of data from different sensors will be a long-term attractive hot issue. Due to the rich information acquisition, convenient use and low price of visual sensors, the environment map constructed is closer to the real environment recognized by human beings. After decades of development, vision-based SLAM technology has achieved many excellent achievements. Table 7 summarizes some of

the best visual-based SLAM algorithms, comparing their performance in key areas, and providing open-source addresses to help readers make better choices.

**Table 7.** Best visual-based SLAM algorithms.

	Method	Sensor	Front-End	Back-End	Loop Closure	Mapping	Code Resource
Visual	MonoSLAM [77]	M	P	F	No	Sparse	[119]
	PTAM [79]	M	P	O	No	Sparse	[120]
	ORB-SLAM2 [28]	M/S/R	P	O	Yes	Sparse	[121]
	PL-SVO [122]	M	PL	O	No	Sparse	[123]
	PL-SLAM [88]	M/S	PL	O	Yes	Sparse	[124]
	DTAM [94]	M	D	O	No	Dense	[125]
	SVO [95]	M	H	O	No	Sparse	[126]
	LSD-SLAM [40]	M/S	D	O	Yes	Semi-dense	[127]
	DSO [39]	M	D	O	No	Sparse	[128]
	Method	Sensor	Coupling	Back-End	Loop Closure	Mapping	Code Resource
Visual-inertial	MSCKF [117]	M + I	T	F	No	Sparse	[129]
	OKVIS [29]	S + I	T	O	No	Sparse	[130]
	ROVIO [131]	M + I	T	F	No	Sparse	[132]
	VINS-Mono [55]	M + I	T	O	Yes	Sparse	[133]

Sensor: M represents Monocular camera; S represents Stereo camera; R represents RGB-D camera and I represents IMU. Front-end: P represents Point; PL represents Point-line; D represents Direct; H represents Hybrid. Back-end: F represents Filtering; O represents Optimization. Coupling: T represents Tightly.

In this chapter, we summarize the traditional vision-based SLAM algorithms, and summarize some excellent algorithms for your reference, hoping to give readers a more comprehensive understanding. Next, we will cover VSLAM with semantic information fusion, aiming to explore the field of SLAM more deeply.

#### 4. Semantic VSLAM

Semantic SLAM refers to a SLAM system that can not only obtain geometric information of the unknown environment and robot movement information but also detect and identify targets in the scene. It can obtain semantic information such as their functional attributes and relationship with surrounding objects, and even understand the contents of the whole environment [134]. Traditional VSLAM represents the environment in the form of point clouds and so on, which to us are a bunch of meaningless points. To perceive the world from both geometric and content levels and provide better services to humans, robots need to further abstract the features of these points and understand them [135]. With deep learning development, researchers have gradually realized its possible help to SLAM problems [136]. Semantic information can help SLAM to understand the map at a higher level. Furthermore, it lessens the dependence of the SLAM system on feature points and improves the robustness of the system [137].

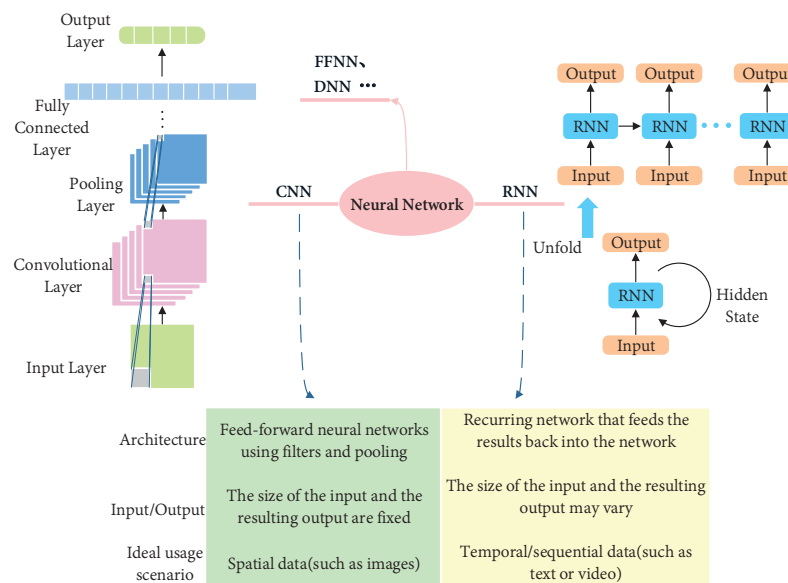
Modern semantic VSLAM systems cannot do without the help of deep learning, and feature attributes and association relations obtained through learning can be used in different tasks [138]. As an important branch of machine learning, deep learning has achieved remarkable results in image recognition [139], semantic understanding [140], image matching [141], 3D reconstruction [142], and other tasks. The application of deep learning in computer vision can greatly ease the problems encountered by traditional methods [143]. Traditional VSLAM systems have achieved commendable results in many aspects, but there are still many challenging problems to be solved [144]. Ref. [145] has summarized deep learning-based VSLAM in detail and pointed out the problems existing in traditional VSLAM. These works [146–149] suggest that deep learning should be used to replace some modules of traditional SLAM, such as loop closure and pose estimation, to improve the traditional method.

Machine learning is a subset of artificial intelligence that uses statistical techniques to provide the ability to “learn” data from a computer without complex programming. Unlike task-specific algorithms, deep learning is a subset of machine learning based on learning data. It is inspired by the function and structure of what are known as artificial neural networks. Deep learning gains great flexibility and power by learning to display the world as simpler concepts and hierarchies, and to calculate more abstract representations based on less abstract concepts. The most important difference between traditional machine learning and deep learning is the performance of data scaling. Deep learning algorithms do not work well when the data is very small, because they need big data to perfectly identify and understand it. The performance of machine learning algorithms depends on the accuracy of features identified and extracted. Deep learning algorithms, on the other hand, identify these high-level features from the data, thus reducing the effort to develop an entirely new feature extractor for each problem. Deep learning is a subset of machine learning, which has proven to be a more powerful and promising branch of the industry compared to traditional machine learning algorithms. It realizes many functions that traditional machine learning cannot achieve with its layered characteristics. SLAM systems need to collect a large amount of information in the environment, so there is a huge amount of data to calculate, and the deep learning model is just suitable for solving this problem.

This paper believes that semantic VSLAM is an evolving process. In the early stage, some researchers tried to improve the performance of VSLAM by extracting semantic information in the environment using neural networks such as CNN. In the modern stage, target detection, semantic segmentation, and other deep learning methods are powerful tools to promote the development of semantic VSLAM. Therefore, in this chapter, we will first describe the application of typical neural networks in VSLAM. We believe that this is the premise of the development of modern semantic VSLAM. The application of neural networks in VSLAM provides a model for modern semantic VSLAM. This paper believes that a neural network is a bridge to introduce semantic information into the modern semantic VSLAM system and obtain rapid development.

#### 4.1. Neural Networks with VSLAM

Figure 13 shows the typical framework of CNN and RNN. CNN can capture spatial features from the image, which help us accurately identify the object and its relationship with other objects in the image [150]. The characteristic of RNN is that it can process an image or numerical data. Because of the memory capacity of the network itself, it can learn data types with contextual correlation [151]. In addition, other types of neural networks such as DNN (Deep Neural Networks) also have some tentative work, but it is in the initial stage. This paper notes that CNN has the advantages of extracting features of things with a certain model, and then classifying, identifying, predicting, or deciding based on the features. It can be helpful to different modules of VSLAM. In addition, this paper believes that RNN has great advantages in helping to establish consistency between nearby frames. Furthermore, the high-level features have better differentiation, which can help robots to better complete data association.



**Figure 13.** Structure block diagram of CNN and RNN. CNN is suitable for extracting unmarked features from hierarchical or spatial data. RNN is suitable for temporal data and other types of sequential data.

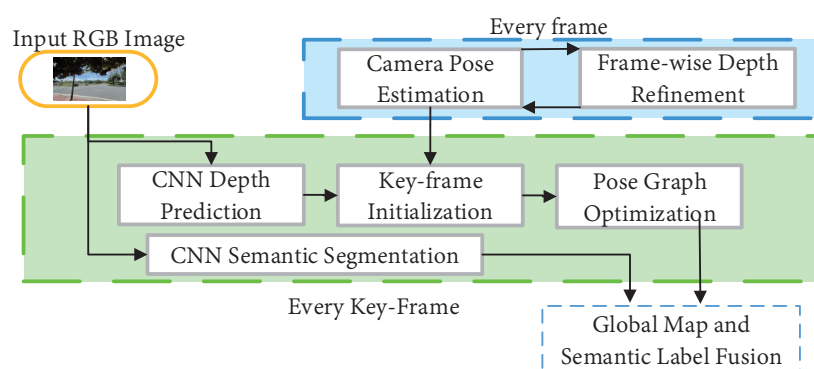
#### 4.1.1. CNN with VSLAM

Traditional inter-frame estimation methods adopt feature-based methods or direct methods to identify camera pose through multi-view geometry [152]. Features-based methods need complex feature extraction and matching. Direct methods rely on pixel intensity values, which makes it difficult for traditional methods to obtain wished results in environments such as intense illumination or sparse texture [153]. In contrast, methods based on deep learning are more intuitive and concise. That is because they do not need to extract environmental features, feature matching, and complex geometric operations [154]. As the feature detection layer of CNN learns through training data, it avoids feature extraction in display and learns implicitly from training data during use. Refs. [155,156] and other works have made a detailed summary.

CNN's advantages in image processing have been fully verified. For example, visual depth estimation improves the problem that monocular cameras cannot obtain reliable depth information [157]. In 2017, Tateno et al. [158] proposed a real-time SLAM system "CNN-SLAM" based on CNN in the framework of LSD-SLAM. As shown in Figure 14, the algorithm obtained a reliable depth map by training the depth estimation network model. CNN is used for depth prediction, which is input into subsequent modules such as traditional pose estimation to improve positioning and mapping accuracy. In addition, CNN semantic segmentation module is added to the framework, which provides help for advanced information perception of the VSLAM system. Similar work using the network to estimate depth information includes Code-SLAM [42] and DVSO [159] Based on a stereo camera. In the same year, Godard et al. [160] proposed an unsupervised image depth estimation scheme. Unsupervised learning is improved by using stereo data set, and then a single frame is used for pose estimation, which has a great improvement compared with other schemes.

CNN not only solves the problem that traditional methods cannot obtain reliable depth data by using a monocular camera but also improves the defects of traditional methods in camera pose estimation. In 2020, Yang et al. [48] proposed D3VO. In this method, deep learning is used from three aspects, including depth estimation, pose estimation, and uncertainty estimation. The prediction depth, pose and uncertainty are closely combined into a direct visual odometer to simultaneously improve the performance of front-end tracking and back-end nonlinear optimization. However, self-supervised methods are difficult to adapt to all environments. In addition, Qin et al. [161] proposed a semantic feature-based

localization method in 2020, which effectively solves the problem that traditional visual SLAM methods are prone to tracking loss. Its principle is to use CNN to detect semantic features in the narrow and crowded environment of an underground parking lot, lack of GPS signal, dim light, and sparse texture. Then use U-Net [162] to perform semantic segmentation to separate parking lines, speed bumps, and other indicators on the ground, and then use odometer information. The semantic features are mapped to the global coordinate system to build the parking lot map. Then the semantic features are matched with the previously constructed map to locate the vehicle. Finally, EKF is used to integrate visual positioning results and odometer information to ensure the system can obtain continuous and stable positioning results in the underground parking environment. Zhu et al. [163] learned rotation and translation by using CNN to focus on different quadrants of optical flow input. However, the end-to-end method to replace the visual odometer is simple and crude but without theoretical support and generalization ability.



**Figure 14.** The structure of CNN-SLAM.

Loop closure detection can eliminate cumulative trajectory errors and map errors, and determines the accuracy of the whole system, which is essentially a scene identification problem [164]. Traditional methods are matched by artificially designed sparse features or pixel-level dense features. Deep learning can learn high-level features in images through neural networks. Furthermore, its recognition rate can reach a higher level by using the powerful recognition ability of deep learning to extract higher-level robust features of images. In this way, the system can have stronger adaptability to image changes such as perspective and illumination and improve the loop closure image recognition ability [165]. Therefore, scene identification based on deep learning can improve the accuracy of loop closure detection, and CNN has also obtained many reliable effects for loop closure detection. Memon et al. [166] proposed a dictionary-based deep learning method, which is different from the traditional Bow dictionary and uses higher-level and more abstract deep learning features. This method does not need to create vocabulary, has higher memory efficiency, and has a faster running speed than similar methods. However, this paper is only based on the likeness score detection cycle, so it is not widely representative. Li et al. [167] proposed a learning feature-based visual SLAM system named DXSLAM, which solved the limitations of the above methods. Local and global features are extracted from each frame using CNN, and these features are then fed into modern SLAM pipelines for posture tracking, local mapping, and repositioning. Compared with traditional BOW-based methods, it achieves higher efficiency and lower computational cost. In addition, Qin et al. [168] used CNN to extract environmental semantic information and modeled the visual scene as a semantic subgraph. It can effectively improve the efficiency of loopback detection by using semantic information. Refs. [169,170] and others describe in detail the achievements of deep learning in many aspects. However, with the introduction of more complex and better models, how to ensure the real-time performance of model calculation? How to better set in the loop closure detection model in resource-constrained platforms, and the lightweight of the model is also a major problem [171].

CNN has achieved good results in replacing some modules of the traditional VSLAM algorithm, such as depth estimation and loop closure detection. Its stability is still not as good as the traditional VSLAM algorithm [172]. In contrast, the semantic information extraction of the CNN system has brought better effects. The process of traditional VSLAM is optimized by using CNN to extract the semantic information of the environment with higher-level features, making the traditional VSLAM achieve better results. Using a neural network to extract semantic information and combining it with VSLAM will be an area of great interest. With the help of semantic information, the data association is upgraded from the traditional pixel level to the object level. The perceptual geometric environment information is assigned with semantic labels to obtain a high-level semantic map. It can help the robot to understand the autonomous environment and human–computer interaction. Table 8 shows some main application links of the CNN network in VSLAM. Some are involved in many aspects, only the main contributions are listed here.

**Table 8.** CNN used for VSLAM.

Part	Method	Contribution
Image Depth Estimation	CNN-SLAM [158]	The depth estimation is performed only on the keyframe, which improves the computing efficiency.
	UnDeepVo [173]	Real-scale monocular vision odometer is realized in an unsupervised way.
	Code-SLAM [44]	A real-time monocular SLAM system is implemented that allows simultaneous optimization of camera motion and maps.
	DVSO [159]	Design a novel deep network that refines predicted depth from a single image in a two-stage process.
Pose estimation	DeTone et al. [174]	It uses only the location of points, not the descriptor of local points.
	VINet [175]	The ability to combine the information in a specific area naturally and cleverly can significantly reduce drift.
	D3VO [48]	The proposed monocular visual odometer framework utilizes deep learning networks at three levels.
	Zhu et al. [163]	Present a novel four-branch network to learn the rotation and translation by leveraging Convolutional Neural Networks (CNNs) to focus on different quadrants of optical flow input.
Loop closure	Memon et al. [166]	Two deep neural networks are used together to speed up the loop closure detection and to ignore the effect of mobile objects on loop closure detection.
	Li et al. [167]	Train a visual vocabulary of local features with a Bag of Words (BoW) method. Based on the local features, global features, and vocabulary, a highly reliable loop closure detection method is built.
	Qin et al. [168]	Models the visual scene as a semantic sub-graph by only preserving the semantic and geometric information from object detection.
Semantic information	CNN-SLAM [158]	By integrating Geometry and semantic information, a map with semantic information is generated.
	Naseer et al. [176]	To achieve real-time semantic segmentation and maintain a good efficiency of differentiation.
	SemanticFusion [46]	The semantic prediction of CNN’s multiple views can be probabilistically integrated into the map.
	Qin et al. [161]	A novel semantic feature used in the visual SLAM framework is proposed.
	Bowman et al. [177]	An optimization problem for sensor state and semantic landmark location is proposed.

#### 4.1.2. RNN with VSLAM

The research of RNN (recurrent neural network) began in the 1980s and 1990s and developed into one of the classical deep learning algorithms in the early 21st century. Long short-term Memory Networks (LSTM) are one of the most common recurrent neural networks [178]. LSTM is a variant of RNN, which remembers a controllable amount of previous training data or forgets it more properly [179]. As shown in Figure 15, the structure of LSTM and the equations of state of its different modules are given. LSTM with special implicit units can preserve input for a long time. LSTM inherits most characteristics of the RNN model and solves the Vanishing Gradient problem caused by the gradual reduction of

the Gradient back transmission process. As another variant of RNN, GRU (Gated Recurrent Unit) is easier to train and can improve training efficiency [180]. RNN has some advantages in learning nonlinear features of sequences because of its memorization and parameter sharing. RNN constructed by introducing a convolutional neural network CNN can deal with computer vision problems involving sequence input [181].

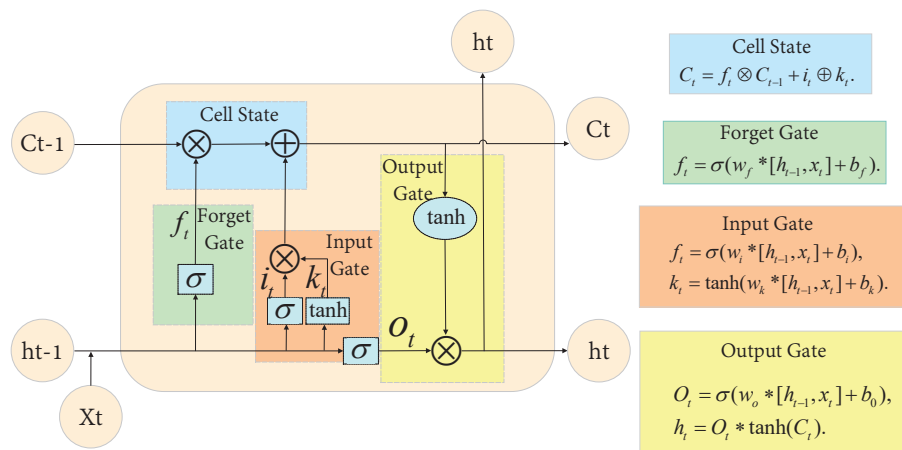


Figure 15. The basic framework of LSTM.

In pose estimation, the end-to-end deep learning method is introduced to solve pose parameters between frames of visual images without feature matching and complex geometric operations. It can quickly obtain the relative pose parameters between frames by directly inputting nearby frames [182]. Xue et al. [183] use deep learning to learn the process of feature selection and realize pose estimation based on RNN. In pose estimation, rotation and displacement are trained separately, which has better adaptability compared with traditional methods. In 2021, Teed et al. [184] introduced DROID-SLAM, whose core is a learnable update operator. As shown in Figure 16, the update operator is a  $3 \times 3$  convolutional GRU with a hidden state of H. The iterative application of the update operator creates a series of attitudes and depths that converge to a fixed point that reflects a real reconstruction. The algorithm is an end-to-end neural network architecture for visual SLAM, which has great advantages over previous work in challenging environments.

Most existing methods adopt to combine CNN with RNN to improve the overall performance of VSLAM. CNN and RNN can be combined using a separate layer, with the output of CNN as the input of RNN. On the one hand, it can automatically learn the effective feature representation of the VO problem through CNN. On the other hand, it can implicitly model the timing model (motion model) and data association model (image sequence) through RNN [185]. In 2017, Yu et al. [60] combined RNN with KinectFusion to carry out semantic annotation on RGB-D collected images to reconstruct a 3D semantic map. They introduced a new loop closure unit into RNN to solve the problem of GPU computing resource consumption. This method makes full use of the advantages of RNN to realize the annotation of semantic information. High-level features have better discrimination and help the robot to better complete the data association. Due to the use of RGB-D cameras, they can only be operated in indoor environments. DeepSeqSLAM [186] solved this problem well. In this scheme, a trainable CNN+RNN architecture is used to jointly learn visual and location representations from a single monocular image sequence. An RNN is used to integrate temporal information on short image sequences. At the same time, using the dynamic information processing functions of these networks, end-to-end position and sequence position learning are realized for the first time. Furthermore, the ability to learn meaningful temporal relationships from single image sequences of large driving datasets. In running time, accuracy, and calculation needs, sequence-based methods are significantly superior to traditional methods and can operate stably in outdoor environments.

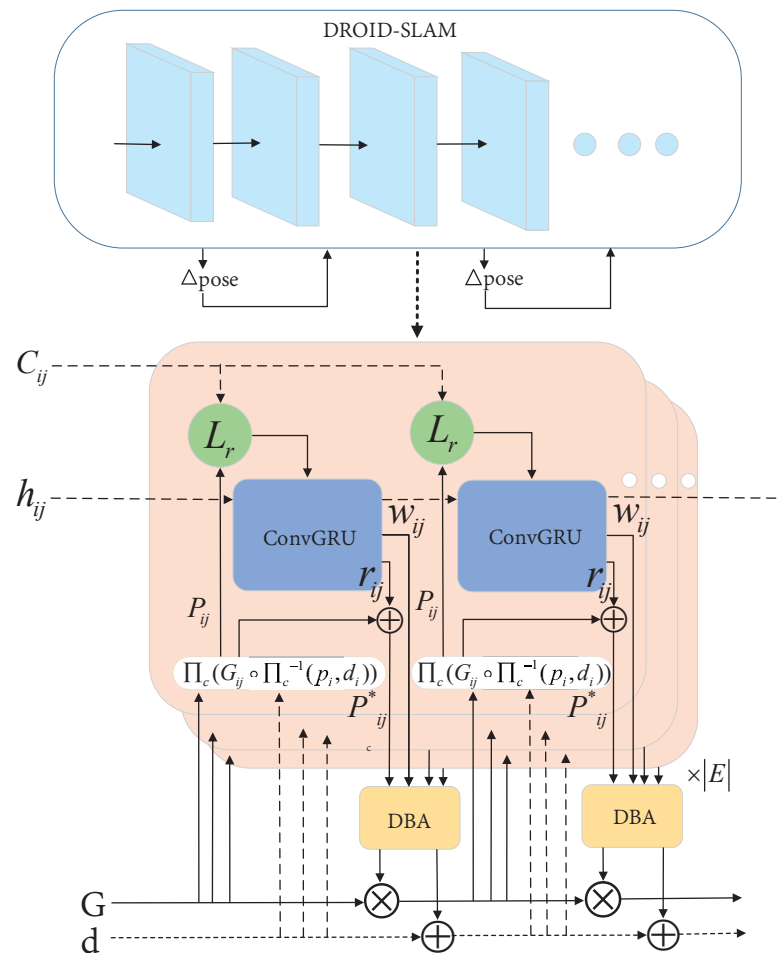


Figure 16. The core framework of DROID-SLAM.

CNN can be combined with many links of VLSAM, such as feature extraction and matching, depth estimation, and pose estimation, and has achieved good results in these aspects. RNN, by contrast, has a smaller scope of application, but it has a great advantage in helping to establish consistency between nearby frames. RNN is a common method for data-driven timing modeling in deep learning. Inertial data such as high frame rate angular velocity and acceleration output by IMU have strict dependence on timing, which is especially suitable for RNN models. Based on this, Clark et al. [175] proposed to use a conventional small LSTM network to process the original data of IMU and obtain the motion characteristics under IMU data. Finally, they combined visual motion features with IMU motion features, and sent it into a core LSTM network for feature fusion and pose estimation. Its principle of it is shown in Figure 17.

Compared with pose estimation, we believe that RNN is more attractive for its contribution to visual-inertial data fusion. This method can effectively fuse visual-inertial data and is more convenient than traditional methods. Similar work, such as [187,188], proves the effectiveness of the fusion strategy, which provides better performance compared with direct fusion. This paper gives the contribution of RNN to partial VSLAM in Table 9.

This paper introduces the combination of deep learning and traditional VSLAM from the classical neural networks CNN and RNN in this section. Table 10 shows some excellent algorithms combining neural networks with VSLAM.



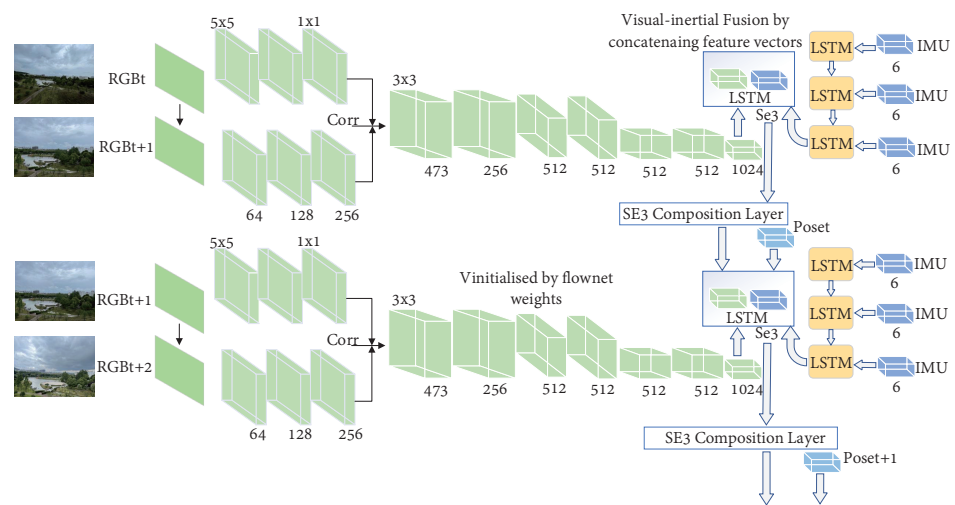


Figure 17. Clark et al. proposed a framework for visual inertia fusion using LSTM.

Table 9. RNN used for VSLAM.

Part	Method	Contribution
VO	Xue et al. [183]	Proposing a dual-branch recurrent network to learn the rotation and translation separately by leveraging current CNN for feature representation and RNN for image sequence reasoning.
	Teed et al. [184]	It consists of recurrent iterative updates of camera pose and pixel-wise depth through a Dense Bundle Adjustment layer.
	DA-RNN [60]	A novel framework for joint 3D scene mapping and semantic labeling.
	DeepSeqSLAM [186]	A trainable CNN+RNN architecture for jointly learning visual and positional representations from a single monocular image sequence of a route.
VIO	Clark et al. [175]	It is the first end-to-end trainable method for visual-inertial odometry which performs a fusion of the data at an intermediate feature-representation level.
	DeepVIO [187]	It reduces the impacts of inaccurate Camera-IMU calibrations and unsynchronized and missing data.
	Chen et al. [188]	It proposes a novel end-to-end selective sensor fusion framework for monocular VIO.
	Yasin et al. [189]	Using adversarial training and self-adaptive visual-inertial sensor fusion.
	Wong et al. [190]	The fusion method of visual inertia + depth data set is proposed for the first time to further enhance the complementary advantages of visual and inertial sensors.

Table 10. An excellent algorithm combining neural networks with VSLAM.

	Method	Year	Sensor	Neural Network	Supervision
VO	CNN-SLAM [158]	2017	Monocular	CNN	Supervised
	DeepVo [191]	2017	Monocular	R-CNN	Supervised
	Code-SLAM [44]	2018	Monocular	U-Net	Supervised
	DVSO [159]	2018	Stereo	DispNet	Semi-supervised
	UnDeepVo [173]	2018	Monocular	VGG encoder-decoder	Unsupervised
	CNN-SVO [192]	2019	Monocular	CNN	Hybrid
	GANVO [193]	2019	Monocular	GAN	Unsupervised
	Li et al. [194]	2019	Monocular	CNN	Supervised
	D3VO [48]	2020	Monocular	CNN	Hybrid
	DeepSeqSLAM [186]	2020	Monocular	CNN+RNN	Supervised
	DeepSLAM [145]	2021	Monocular	R-CNN	Unsupervised
	LIFT-SLAM [195]	2021	Monocular	DNN	Supervised
	Zhang et al. [196]	2021	Stereo	U-Net encoder-decoder	Unsupervised
	VIO	VINet [175]	2017	Monocular + IMU	CNN + LSTM
VIOlearner [197]		2020	Monocular + IMU	CNN	Unsupervised
DeepVIO [187]		2019	Stereo + IMU	CNN + LSTM	Supervised
Chen et al. [188]		2019	Monocular + IMU	FlowNet + LSTM	Supervised
Kim et al. [198]		2021	Monocular + IMU	CNN + LSTM	Unsupervised
Gurturk et al. [199]		2021	Monocular + IMU	CNN + LSTM	Supervised

#### 4.2. Modern Semantic VSLAM

Deep learning has made many achievements in pose estimation, depth estimation, and loop closure detection. However, in VSLAM, deep learning is currently unable to shake the dominance of traditional methods. However, applying deep learning to semantic VSLAM research can obtain more valuable discoveries, which can quickly promote to development of semantic VSLAM. Refs. [60,158,168] used CNN or RNN to extract semantic information in the environment to improve the performance of different modules in traditional VSLAM. The semantic information was used for pose estimation and loopback detection. It significantly improved the performance of traditional methods and proved the effectiveness of semantic information for the VSLAM system. This paper believes that this provides technical support for the development of modern semantic VSLAM and is the beginning of modern semantic VSLAM. Using deep learning methods such as target detection and semantic segmentation to create a semantic map, which is an important representative period of semantic SLAM development. Refs. [135,200] points out that semantic SLAM can be divided into two types according to different target detection methods. One is to detect targets using traditional methods. Real-time monocular object SLAM is the most common one, using a large number of binary words and a database of object models to provide real-time detection. However, it's very limited because there are many types of 3D object entities for semantic classes such as "cars." Another approach to SLAM is object recognition using deep learning methods, such as those proposed in [46].

Semantics and SLAM may seem to be separate modules, but they are not. In many applications, the two go hand in hand. On the one hand, semantic information can help SLAM to improve the accuracy of mapping and localization, especially for complex dynamic scenes [201]. The mapping and localization of traditional SLAM are mostly based on pixel-level geometric matching. With semantic information, we can upgrade the data association from the traditional pixel level to the object level, improving the accuracy of complex scenes [202]. On the other hand, by using SLAM technology to calculate the position constraints between objects, the consistency constraints can be applied to the recognition results of the same object at different angles and at different times, thus improving the accuracy of semantic understanding. The integration of semantic and SLAM not only contributes greatly to the improvement of the accuracy of both but also promotes the application of SLAM in robotics, such as robot path planning and navigation, carrying objects according to human instructions, doing housework, and accompanying human movement, etc.

For example, We want a robot to walk from the bedroom to the kitchen to get an apple. How does that work? Relying on traditional SLAM, the robot calculates its location (automatically) and Apple's location (manually) and then does path planning and navigation. If the apple is in the refrigerator, you also need to manually set the relationship between the refrigerator and the apple. However, now with our semantic SLAM technology, it's much more natural for a human to send a robot, "Please go to the kitchen and get me an apple", and the robot will do the rest automatically. If there is a contaminated ground in front of the robot during an operation, traditional path planning algorithms need to manually mark the contaminated area so the robot can bypass it [203].

Semantic information can help robots better understand their surroundings. Integrating semantic information into VSLAM is a growing field that has received more and more attention in recent years. This paper will elaborate on our understanding of semantic VSLAM from two aspects of localization, mapping, and dynamic object removal in this section. We believe the biggest contribution of deep learning for VSLAM is the introduction of semantic information. It can improve the performance of different modules of traditional methods to varying degrees. Especially in the construction of the semantic map, which promotes the innovation of the whole intelligent robot field.

#### 4.2.1. Image Information Extraction

The core difference between modern semantic VSLAM and traditional VSLAM lies in the integration of the object detection module. It can obtain the attributes and semantic information of objects in the environment [204]. The first step of semantic VSLAM is to extract semantic information from the images gained by the camera. Furthermore, semantic information based on image information can be achieved through classifying image information [205]. Traditional target detection relies on interpretable machine learning classifiers, such as decision trees and SVM, to classify and realize target features. However the detection process is slow, the accuracy is low and the generalization ability is weak [206]. Image classification based on deep learning can be divided into Object detection, Semantic segmentation, and Instance segmentation, as shown in Figure 18.



**Figure 18.** From left to right are the test renderers of YOLOv5, Deeplabv3, and Mask R-CNN.

How to better extract semantic information from images is a hot research issue in computer vision, whose essence is to extract object character information from scenes [207]. We believe that although neural networks such as CNN also contribute to semantic information extraction, modern semantic VSLAM relies more on semantic extraction modules such as target detection. Object detection and image semantic segmentation are both methods of extracting semantic information from images. Semantic segmentation of images is to understand images at the pixel level to obtain deep-level information in the image, including space, category, and edge. Semantic segmentation technology based on a deep neural network breaks through the bottleneck of traditional semantic segmentation [208]. Compared with semantic segmentation, target detection only obtains the object information and spatial information of the image. Furthermore, it identifies the category of each object by drawing the candidate box of the object, so target detection is faster than semantic segmentation [209]. Compared with object detection, semantic segmentation technology has higher accuracy, but its speed is much lower [210].

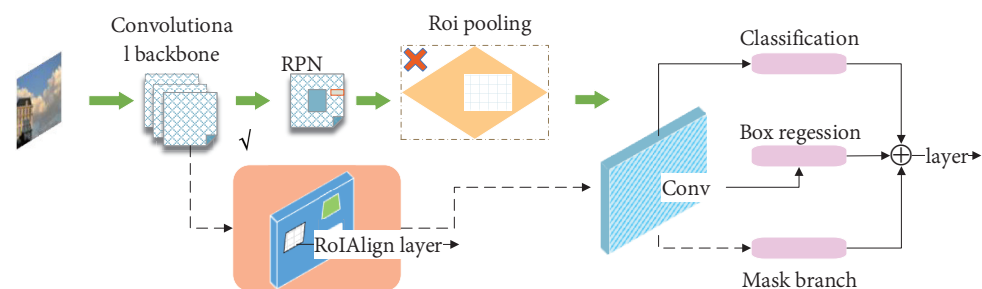
Target detection is divided into one-stage and two-stage structures [211]. Early target detection algorithms use two-stage architecture. After creating a series of candidate boxes as samples, sample classification is carried out through a convolutional neural network. Common algorithms include R-CNN [212], Fast R-CNN [213], Faster R-CNN [214], and so on. Later, YOLO [215] creatively proposed the one-stage structure. It directly carried out the Two steps of the two-stage in One step, completed the classification and positioning of objects in one step, and directly output the candidate box and its category obtained by regression. One-stage reduces the steps of the target detection algorithm and directly converts the problem of target frame positioning into regression problem theory without the need to create candidate boxes, which are superior in speed. Common algorithms include YOLO and SSD [216].

In 2014, the appearance of R-CNN subverted the traditional object detection scheme, improved the detection accuracy, and promoted the rapid development of object detection technology. Its core is to extract candidate regions, then obtain feature vectors through Alexnet, and finally use SVM classification and frame correction. However, the speed of feature extraction is limited due to the serial feature extraction method used by R-CNN. Ross proposed Fast R-CNN in 2015 to solve this problem well. Region of Interest Pooling (ROI Pooling) operation is used in Fast R-CNN to improve the efficiency of feature

extraction, and Region generation network (RPN) is used for coordinate correction. Many candidate frames (anchor) are set in RPN. Then the dependency relation of the anchor to the background is judged, to work out the coverage area of the anchor and determine whether the target is covered. In addition, YOLO improves the accuracy of prediction, speeds up the processing speed and increases the types of identified objects, and proposes a joint training method for target classification and detection. YOLO is one of the most widely used target detection algorithms, offering real-time detection and a series of improved versions since then.

Different from object detection, semantic segmentation not only predicts the position and category of objects in the image but also accurately describes the boundary between different kinds of objects. However, in semantic segmentation technology, an ordinary convolutional neural network cannot obtain enough information. To solve this problem, Long et al. proposed a fully convolutional neural network FCN [217]. Compared with CNN, FCN does not have a fully connected layer. The new FCN obtains the spatial position of the feature map and fuses the output of different depth layers with the hierarchical structure. This method combines local information with global information and improves the accuracy of semantic segmentation. In Segnet network proposed by Badriarayanan et al. [218], the encoder-decoder structure was proposed, which combined two independent networks to improve the accuracy of segmentation. However, the combination of two independent networks severely reduced the detection speed. Zhao et al. proposed PSPNet [219] and a pyramid module, which fuses the features of each level, such as a pyramid, and finally fuses the output to further improve the segmentation effect.

In recent years, the continuous improvement of computer performance promotes the rapid development of instance segmentation in vision. Instance segmentation not only has the classification on the pixel level (semantic segmentation) but also has the location information of different objects (target detection), even the same object can be detected. In 2017, He et al. proposed the Mask R-CNN [220]. This algorithm is the pioneering work of instance segmentation. As shown in Figure 19, its main idea is to add a branch for semantic segmentation based on Faster R-CNN.



**Figure 19.** The framework of MASK-RCNN.

Although the target detection and segmentation technology based on a neural network have been perfect, it needs to rely on powerful computing capacity to achieve real-time processing. VSLAM has a high requirement for real-time operation, so how efficiently separating the needed object and its semantic information from the environment will be a long-term and hard task. As the basis of semantic VSLAM, after processing semantic segmentation, we will pay attention to the influence of semantic information on different aspects of VSLAM. We will elaborate on three aspects of localization, mapping, and dynamic object removal. Object detection and semantic segmentation are both a means of extracting semantic information from images. Table 11 shows the contribution of some algorithms. Object detection is faster than semantic segmentation. However, semantic segmentation is better in precision. Instance segmentation integrates object detection and semantic segmentation, and has outstanding performance in precision, but can not guarantee the running speed. For some schemes that cannot provide the original paper, we provide the open-source code, such as YOLOV5.

**Table 11.** Part of the classical image detection algorithm.

Field	Model	Year	Contribution
Object detection	R-CNN [212]	2014	The first algorithm that successfully applied deep learning to target detection.
	Fast R-CNN [213]	2015	Image feature extraction is performed only once.
	Faster R-CNN [214]	2017	Integrated into a network, the comprehensive performance has been greatly improved.
	SSD [216]	2016	SSD was an early incarnation of the single-phase model.
	YOLO [215]	2016	Think of detection as a regression problem, using a network to output positions and categories.
	YOLOv5 [221]	2020	The environment is easy to configure and model training is very fast.
Semantic segmentation	FCN [217]	2015	It opens the first application of a convolutional neural network in semantic segmentation.
	SegNet [218]	2017	A completely symmetrical structure is adopted.
	DeepLabv1 [222]	2014	Atrous convolution.
	DeepLabv3+ [223]	2018	Greatly reduce the number of parameters.
	PSPnet [219]	2017	A Pyramid Pooling Module can aggregate contextual information from different regions.
Instance segmentation	Mask R-CNN [220]	2017	It can not only detect the target in the image but also give a high-quality segmentation result for each target.
	YOLACT [224]	2019	Based on the one-stage target detection algorithm, the overall architecture design is very lightweight and achieves good results in speed and effect.

#### 4.2.2. Semantic with Location

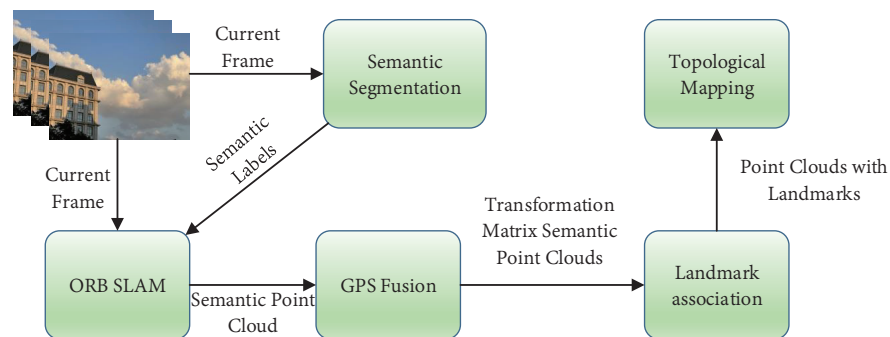
Location accuracy is one of the most basic assessment standards in the SLAM system and is a precondition for mobile robots to perform many tasks [225]. Introducing environmental semantic information can effectively improve the scale uncertainty and cumulative drift in visual SLAM localization, thus improving the localization accuracy to varying degrees [226].

Bowman et al. [177] proposed a sensor state estimation and semantic landmark location optimization problem, which integrates metric information, semantic information, and data association. After obtaining semantic information from target detection, they introduced the Expectation-Maximization (EM) and calculated the probability of data association according to the result of semantic classification. They successfully converted semantic SLAM into a probability problem and improved the localization accuracy of the SLAM system. However, there are many strong assumptions in this paper. Such as the projection of the three-dimensional center of the object should be close to the center of the detection network, which is not easy to meet in practice.

In 2020, Zhao et al. [227] of Xi'an Jiaotong University proposed a landmark visual semantic SLAM system for a large-scale outdoor environment. Its core is to combine a 3D point cloud in ORB-SLAM with semantic segmentation information in the convolutional neural network model PSPNET-101. It can build a 3D semantic map of a large-scale environment. They proposed a method to associate real landmarks with a point cloud map. It associates architectural landmarks with the semantic point cloud and associates landmarks obtained from Google Maps with a semantic 3D map for urban area navigation. With the help of a semantic point cloud, the system realizes landmark-based relocation in a wide range of outdoor environments without GPS information. Its process is shown in Figure 20. In 2018, ETH Zurich proposed VSO [228] based on semantic information for autonomous driving scenarios. This scheme solves the problem of visual SLAM localization in the environment of outdoor lighting changes. It establishes constraints between semantic information with images and takes advantage of the advantage that semantic information is not affected by Angle of view, scale, and illumination. Similarly, Stenborg et al. [229] also proposed solutions to such problems.

In the aspect of trajectory estimation, geometric features can only provide short-term constraints for camera pose, which will produce large deviations in a wide range

of environments. In contrast, objects, as higher-level features, can keep their semantic information unchanged when light intensity, observation distance, and Angle change. For example, a table is still a table under any light and Angle, and its more stable performance can provide long-term constraints for the camera posture. In addition, semantic SLAM can effectively solve the problems that traditional visual SLAM is sensitive to illumination changes and interferes with the robustness of system positioning. We believe that VSLAM localization is essentially camera pose estimation. Semantic information can improve the positioning accuracy of traditional VSLAM systems under strong illumination and high camera rotation. However, in practice, the introduction of semantic information will inevitably slow down the operation of the whole system, which is an urgent problem to be solved in VSLAM. We believe that in most cases, traditional VSLAM still performs well in localization accuracy. However, semantic help for VSLAM systems to improve localization accuracy is also worthy of research. Table 12 compares the differences between traditional methods and semantic methods for VSLAM localization.



**Figure 20.** Zhao et al. proposed a large-scale outdoor positioning process using semantic information.

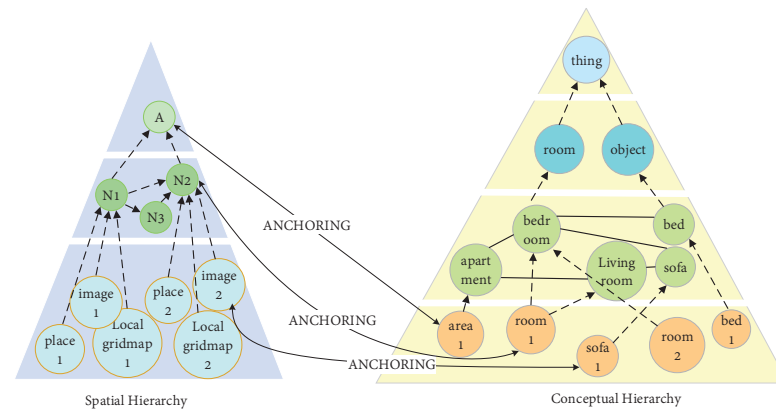
**Table 12.** Comparison between traditional methods and semantic methods for VSLAM localization.

	Method	Characteristic
Traditional	Epipolar Geometry, Perspective-n-Point, Iterative Closest Point, Optical Flow ...	Geometric features can only provide short-term constraints for camera pose and may fail in environments with strong light and fast motion.
Semantic	semantic label, data association	The semantic information can remain constant when the light intensity, observation distance, and angle change.

#### 4.2.3. Semantic with Mapping

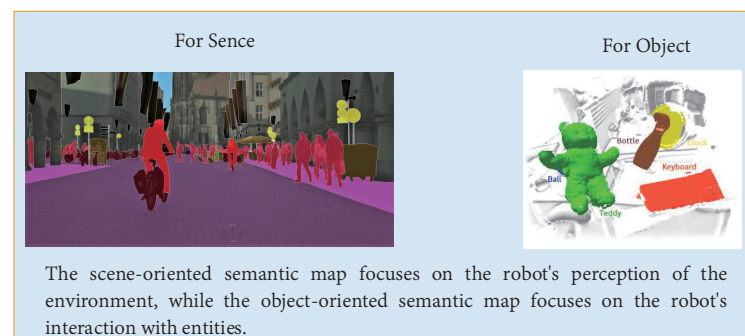
Another key juncture of VSLAM and deep learning is the semantic map construction of SLAM, and most semantic VSLAM systems are based on this idea [230]. For a robot to understand the environment as well as a human and perform different tasks from one place to another requires a different skill than a geometric map can provide [231]. Robots should have the ability to have a human-centered understanding of their environment. It needs to distinguish between a room and a hallway, or the different functions of a kitchen and a living room in the future [232]. Therefore, semantic attributes involving human concepts (such as room types, objects, and their spatial layout), which is considered a necessary attribute of future robots [233]. In recent years, with the rapid development of deep learning, a semantic map containing semantic information has gradually come into people's view [234]. The semantic map in the semantic SLAM system enables robots to obtain geometric information such as feature points of the environment. Furthermore, it also identifies objects in the environment and obtains semantic information such as location, attribute, and category. Compared with the map constructed by traditional VSLAM, the robot can be equipped with perceptual ability. It is significant for the robot to deal with

a complex environment and complete human–computer interaction [235]. Semantic map construction is one of the hot topics in SLAM research [236]. In 2005, Galindo et al. [237] proposed the concept of a semantic map. As shown in Figure 21, it is represented by two parallel layers: spatial representation and semantic representation. It provides robots with an inference ability similar to humans to the environment (for example, a bedroom is a room containing a bed). Later, Vasudevan et al. [238] further strengthened people’s understanding of semantic maps.



**Figure 21.** The semantic map concept mentioned in Galindo’s article.

In recent years, deep learning technology has developed rapidly. More and more researchers combine deep learning with SLAM technology. They use target detection, semantic segmentation, and other algorithms to obtain semantic information about the environment. Furthermore, integrate it into the environment map to construct the environment semantic map [239]. As shown in Figure 22, the research on semantic map construction is mainly divided into two directions: scene-oriented semantic map construction and object-oriented semantic map construction.



**Figure 22.** Different types of semantic maps.

Most scenario-oriented semantic maps are based on deep learning methods, which map 2D semantic information to 3D point clouds. Scenario-oriented semantic maps can help robots better understand their environment [240]. In 2020, MIT proposed Kimera [241]. This is a mature scenario-oriented semantic SLAM algorithm. Ref. [242] proposed an algorithm of semantic map construction oriented to the scene. Based on RTABMAP [243], YOLO is used for target detection. After roughly estimating the position of the object, they used the Canny operator to detect the edge of the target object in the depth image. Then they achieved accurate segmentation of the object by processing edge based on the region growth algorithm. Through the non-deep learning semantic segmentation algorithm, they solved the problem of large computing resources in traditional semantic map construction, and constructed the scene-oriented semantic map in real-time. The scene-oriented semantic map will help the robot better understand the environment, and build a more expressive

environment map. However, this method cannot provide more help for a robot to know the environment, preventing the robot and the environment of the individual to interact, to a certain extent restricting the intellectualized degree of the robot [244]. In addition, such algorithms need to carry out pixel-level semantic segmentation of objects in the scene, which leads to much system calculation and low real-time performance. Therefore, some scholars turn to object-oriented semantic map construction algorithms [245].

An object-oriented semantic map refers to a map that contains only partial instance semantic information, and the semantic information exists independently in the method of clustering [246]. This type of map allows robots to operate and maintain the semantic information of each entity on the map. So it is more conducive for robots to understand the environment and interact with entities in the environment, improving the practicality of the map [247]. Reference [45] proposed a voxel-based semantic visual SLAM system based on mask-RCNN and KinectFusion algorithm. After object detection by the Mask-RCNN algorithm, object detection results are fused with the TSDF model based on voxel foreground theory to construct an object-oriented semantic map. Although the accuracy of detection is guaranteed, it still cannot solve the problem of the poor real-time performance of the algorithm. Ref. [248] proposed a lightweight object-oriented SLAM system, which effectively solves the problems of data association and attitude estimation, and solves the problem of the poor real-time performance of the above methods. The core framework is developed based on ORB-SLAM2 and uses YOLOv3 as an object detector to fuse semantic thread. In the tracer thread, boundary box, semantic label, and point cloud information are fused, and the object-oriented semi-dense semantic map is constructed. Experimental results show that compared with ORB-SLAM2, the scheme can deal with multiple classes of objects with different scales and directions in a complex environment, and can better express the environment. However, for some large objects, accurate pose estimation is not possible. Similarly, University College London proposed DSP-SLAM [249].

At present, most semantic map construction methods need to deal with both instance segmentation and semantic segmentation at the same time, which leads to poor real-time performance of the system [250]. Table 13 lists some semantic map construction work. In addition, when dealing with dynamic objects, most algorithms realize system robustness by eliminating dynamic objects, which will make the system lose much useful information. Therefore, SLAM oriented to dynamic scenes is an urgent problem to be solved [251].

**Table 13.** Part of the excellent semantic mapping algorithms.

Reference	Year	Sensor	Semantic labeling	Map	Contribution
Vineet et al. [252]	2015	S	Random Forest	Voxel	The first system can perform dense, large-scale, outdoor semantic reconstruction of a scene in real-time.
Zhao et al. [253]	2016	D	SVM	Voxel	Use temporal information and higher-order cliques to enforce the labeling consistency for each image labeling result.
Li et al. [254]	2016	D	Deeplabv2	Voxel	There is no need to obtain a semantic segmentation for each frame in a sequence.
SemanticFusion [46]	2016	D	CNN with CRF	Surfel	Allows the CNN's semantic predictions from multiple viewpoints to be probabilistically fused into a dense semantically annotated map.
Yang et al. [255]	2017	S	CNN with CRF	Grid	Further, optimize 3D grid labels through a novel CRF model.
Panopticfusion [256]	2020	D	PSPNET with CRF Mask R-CNN with CRF	Voxel	A novel online volumetric semantic mapping system at the level of stuff and things.
Kimera [241]	2020	S + I	Pixel-wise	Mesh	It is modular and allows replacing each module or executing them in isolation.
AVP-SLAM [161]	2020	M + I + E	U-Net	Voxel	Autonomous parking.
RoadMap [257]	2021	R + M + I + E	CNN	Voxel	A framework of on-vehicle mapping, on-cloud maintenance, and user-end localization.

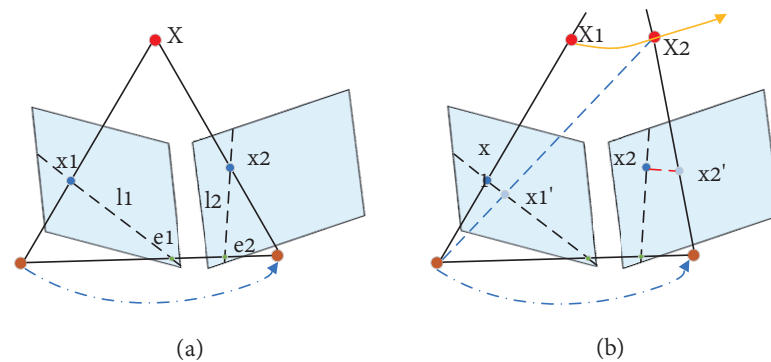
Sensor: S represents Stereo camera; M represents Monocular camera; I represents IMU; E represents encoder; R represents RTK-GPS and D represents RGB-D camera.



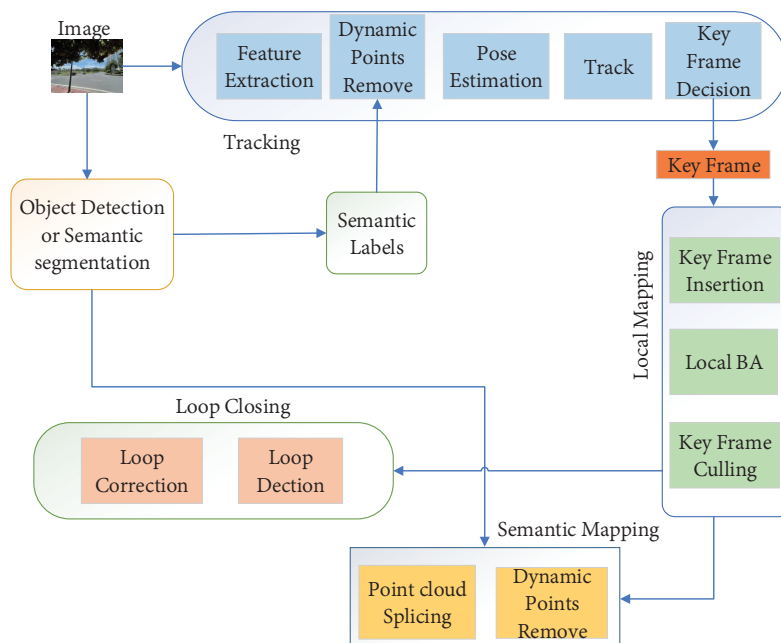
#### 4.2.4. Elimination of Dynamic Objects

Traditional VSLAM algorithms assume that objects in the environment are static or low-motion, which affects the applicability of the VSLAM system in actual scenes [258]. When dynamic objects exist in the environment (such as people, vehicles and pets), they will bring wrong observation data to the system and reduce the accuracy and robustness of the system [259]. Traditional methods solve the influence of some outliers on the system through the RANSAC algorithm. However, if dynamic objects occupy most of the image area or moving objects are fast, reliable observation data still cannot be obtained [260]. As shown in Figure 23, the camera cannot accurately capture data due to dynamic objects. So how to solve the impact of dynamic objects on the SLAM system has become the goal of many researchers.

Now, the solutions to the problem of disturbance brought by dynamic objects to the SLAM system are consistent. That is, before the visual odometer, using target detection and image segmentation algorithm to filter out the dynamic areas in the image. Then use static environment points to calculate the nearby positions of the camera and construct a map containing semantic information [261]. Figure 24 shows a typical structure. Although the influence of dynamic objects cannot be completely solved, the robustness of the system is greatly improved.



**Figure 23.** Traditional methods use geometric constraints to judge whether an object is moving or not. For example, in (a), X is a static point in space, so the spatial transformation relation can be obtained smoothly. In (b), the motion of space point X1 will bring systematic error after it moves to X2.



**Figure 24.** Typical dynamic object removal frame.

In 2018, Bescos et al. [262] proposed the DynaSLAM algorithm for visual SLAM for dynamic scenarios based on ORB-SLAM2. The system provides interfaces for monocular, stereo, and RGB-D cameras. For monocular and stereo cameras, MASK-RCNN is used to segment dynamic objects in each frame to avoid feature extraction of dynamic objects in the SLAM system. If an RGB-D camera is used, the method of multi-view geometry is used for more accurate motion segmentation. Dynamic segments are removed from the current frame and map. However, this method chooses to remove all potentially moving objects, such as parked cars. This may lead to too few remaining stationary feature points and affect camera pose estimation. In the same year, the Tsinghua University team proposed a complete SLAM system DS-SLAM [263] based on ORB-SLAM2. Its core is the ORB-SLAM2 added a semantic network segmentation, and as a separate thread running in real-time. It can remove objects in the scene dynamic segmentation and create a separate thread to build a dense semantic octree map to help the robot to achieve a higher level of the task.

Some methods use semantic information to hide objects that are considered to be dynamic. Although such methods improve the influence of dynamic objects on the system to a certain extent, the one-size-fits-all approach may cause the system to lose many useful feature points. For example, a car parked on the roadside may be regarded as a dynamic object and all feature points carried by it are filtered out [264]. However, a car stationary on the side of the road can be used as a reliable feature point in the system. However, it can even be a major source of high-quality feature points. Reference [265] proposed the integration of semantic information into traditional VSLAM methods. This method does not need to motion detection. The introduction of confidence, gives each object a different possible movement probability, to judge whether an object is in motion. Furthermore, the semantic label distribution is combined with map point observation consistency, to estimate the reliability of each 3D point measurement. Then use it in the map of pose estimation and optimization steps. This method can handle objects that are considered dynamic but are stationary, such as cars parked on the side of the road. Reference [266] is based on the optical flow method to remove dynamic objects. Its core idea is based on ORB-SLAM2. In its front end, four CNN neural networks are used to simultaneously predict the depth, posture, optical flow, and semantic mask of each frame. By calculating the rigid optical flow synthesized by depth and posture and comparing the estimated optical flow, the initial motion region is obtained. The algorithm can distinguish the moving object from the current scene and retain the feature points of the static object. Avoiding the removal of the moving object based on the category attribute only, which leads to the tracking failure of the SLAM system. The article [267] has presented a visual SLAM system that is built on ORB-SLAM2 and performs robustly and accurately in dynamic environments through discarding the moving feature points with the help of semantic information obtained by Mask-RCNN and depth information provided by RGB-D camera. This method tries to exploit more reliable feature points for camera pose estimation by finding out the static feature points extracted from movable objects, which would benefit a lot when static objects could not provide enough feature points in the scene.

Semantic information can better help the system to solve the interference brought by dynamic objects, due to the high consumption of computing resources. However, the existing schemes are generally not real-time enough to be widely promoted to practical robots, and the application scenarios are greatly limited [268]. In addition, semantic information may not be available at the camera frame rate, or may not always provide accurate data [269]. Assigning an image region to the wrong semantic class may unnecessarily exclude it from posture estimation, which can be critical in a sparsely textured environment [270]. Current solutions to this problem focus on using methods such as optical flow to detect objects that are moving in the scene [271]. Although the existing algorithms have achieved good results in data sets, they have not achieved very reliable results in practical engineering. Table 14 shows the VSLAM algorithms using a deep neural network to improve the dynamic environment in recent years.

**Table 14.** Some excellent VSLAM algorithms for dynamic scenarios in recent years.

Model	Year	Sensor	Scene	Dynamic Detection	Dataset	Code Resource
Reddy et al. [272]	2016	Stereo	Outdoor	[273]	KITTI	0
DynaSLAM [262]	2018	Monocular/Stereo/ RGB-D	Outdoor/Indoor	Mask R-CNN	KITTI/TUM RGB-D	[274]
DS-SLAM [263]	2018	RGB-D	Indoor	SegNet	TUM RGB-D	[275]
Detect-SLAM [276]	2018	RGB-D	Indoor	SSD	TUM RGB-D	[277]
Wang et al. [278]	2019	RGB-D	Indoor	YOLOv3	NYU Depth Dataset V2	0
SLAMANTIC [265]	2019	Monocular/Stereo	Outdoor	Mask R-CNN	TUM RGB-D/ VKITTI	[279]
DynSLAM [280]	2018	Stereo	Outdoor	Cascades [281]	KITTI	[282]
STDyn-SLAM [283]	2022	Stereo	Outdoor	SegNet	KITTI	[284]
PoseFusion [285]	2018	RGB-D	Indoor	OpenPose	Freiburg RGB-D SLAM	[286]
RDS-SLAM [287]	2021	RGB-D	Indoor	SegNet/Mask R-CNN	TUM RGB-D	[288]
YO-SLAM [289]	2021	RGB-D	Indoor	Yolact	TUM RGB-D	0
Zhang et al. [290]	2021	Panoramic	Data	Yolact	[291]	0
DOE-SLAM [292]	2021	Monocular	Indoor	self-initiated *	TUM RGB-D	0
DRSO-SLAM [293]	2021	RGB-D	Indoor	Mask R-CNN	TUM RGB-D	0
DDL-SLAM [294]	2020	RGB-D	Indoor	DUNet	TUM RGB-D	0
RDMO-SLAM [295]	2021	RGB-D	Indoor	Mask R-CNN	TUM RGB-D	0

Code resource: 0 represents no code resource. Dynamic detection: [242] represent please refer to this paper; self-initiated \* represent refer to the method proposed in this paper.

## 5. Conclusions and Prospect

Simultaneous localization and mapping is a major research problem in the robotics community, where a great deal of effort has been devoted to developing new methods to maximize their robustness and reliability. Vision-based SLAM technology has experienced many years of development, and many excellent algorithms have emerged, which have been successfully applied in various fields such as robotics and UAV. The rapid development of deep learning has promoted the innovation of the computer field, and the combination of the two has become an active research field. Therefore, the research on VSLAM has received more and more attention. In addition, with the advent of the intelligent era, higher requirements are put forward for the autonomy of mobile robots. In order to realize advanced environment perception of robots, semantic VSLAM has been proposed and developed rapidly. Traditional VSLAM only restores the geometric features of the environment when constructing the environment map, which cannot meet the requirements of robot navigation, human–computer interaction, autonomous exploration, and other applications. However, the early semantic map construction method generally adopts the model library matching method, which requires the construction of an object model library in advance, which has great limitations and is not conducive to popularization and application. With the improvement of computer performance and the rapid development of deep learning technology, VSLAM technology is combined with deep learning technology to fill the deficiency of the traditional VSLAM system. In recent years, as the most promising and advantageous computer vision processing method, deep learning technology has been widely concerned by SLAM researchers. In the semantic SLAM system, environmental semantic information can be directly learned from pre-trained image sets and real-time perceived image sets by deep learning techniques. It can also be used to make better use of large data sets, giving the system greater generalization capability. When constructing a semantic map, the semantic SLAM system can use the deep learning method to detect and classify objects in the environment and construct a map with richer information, which has better practicality.

In this article, we investigate most of the most advanced visual SLAM solutions that use features to locate robots and map their surroundings. We classify them according to the feature types relied on by feature-based visual SLAM methods; Traditional VSLAM and VSLAM combined with deep learning. The strengths and weaknesses of each category are thoroughly investigated and, where applicable, the challenges that each solution overcomes

are highlighted. This work demonstrates the importance of using vision as the only external perceptual sensor to solve SLAM problems. This is mainly because the camera is an ideal sensor because it is light, passive, low-power, and capable of capturing rich and unique information about a scene. However, the use of vision requires reliable algorithms with good performance and consistency under variable lighting conditions, due to moving people or objects, phantoms of featureless areas, transitions between day and night, or any other unforeseen circumstances. Therefore, SLAM systems using vision as the only sensor remain a challenging and promising research area. Image matching and data association are still open research fields in computer vision and robot vision, respectively. The choice of detectors and descriptors directly affects the performance of the system to track salient features, identify previously seen areas, build a consistent environmental model, and work in real-time. Data correlation in particular requires long-term navigation, despite a growing database and a constantly changing and complex environment. Accepting bad associations will cause serious errors in the entire SLAM system, meaning that location calculations and map construction will be inconsistent.

In addition, we highlight the development of VSLAM that fuses semantic information. The VSLAM system combined with semantic information achieves better results in terms of robustness, precision, and high-level perception. More attention will be paid to the research of semantic VLSAM. Semantic VSLAM will fundamentally improve the autonomous interaction ability of robots.

Combined with other studies, we make the following prospects for the future development of VSLAM:

(1) Engineering application. After decades of development, VSLAM has been widely used in many fields such as robotics. However, SLAM is sensitive to environmental illumination, high-speed motion, motion interference and other problems, so how to improve the robustness of the system and build large-scale maps for a long time are all worthy of challenges. The two main scenarios used in SLAM are based on embedded platforms such as smart phones or drones, and 3D reconstruction, scene understanding and deep learning. How to balance real-time and accuracy is an important open question. Solutions for dynamic, unstructured, complex, uncertain and large-scale environments remain to be explored.

(2) Theoretical support. The information features learned through deep learning still lack intuitive meaning and clear theoretical guidance. At present, deep learning is mainly applied to local sub-modules of SLAM, such as depth estimation and closed-loop detection. However, how to apply deep learning to the entire SLAM system remains a big challenge. Traditional VSLAM still has advantages in positioning and navigation. Although some modules of traditional methods are improved by deep learning, the scope of deep learning is generally not wide, and it may achieve good results in some data sets, but it may be unstable in another scene. The positioning and mapping process involves a lot of mathematical formulas, and deep learning has drawbacks in dealing with mathematical problems while using deep learning has fewer data to carry out relevant training, and this method is more traditional. The SLAM framework does not present significant advantages and is not yet available. The main algorithms of SLAM technology. In the future, SLAM will gradually absorb deep learning methods and improve training numbers data sets are used to improve the accuracy and robustness of positioning and mapping.

(3) High-level environmental information perception, and human–computer interaction. With the further development of deep learning, the research and application of semantic VSLAM will have a huge space for development. In the future intelligent era, people’s demand for intelligent autonomous mobile robots will increase rapidly. How to use semantic VSLAM technology to better improve the autonomous ability of robots will be a long-term and difficult task. Although there have been some excellent achievements in recent years, compared with the classical VSLAM algorithm, semantic VSLAM is still in the development stage. Currently, there are not many open source solutions for semantic SLAM, and the application of semantic SLAM is still in the initial stage, mainly because the construction of an accurate semantic map requires a lot of computing resources. This

severely interferes with the real-time performance of SLAM. With the continuous improvement of hardware level in the future, the problem of the poor real-time performance of SLAM systems may be greatly improved.

(4) Establish a sound evaluation system. Semantic VSLAM technology has developed rapidly in recent years. However, compared with traditional VSLAM, there are no perfect evaluation criteria for the time being. In SLAM system research, ATE or RPE is generally used to evaluate the system performance. However, both of these evaluation criteria are based on the pose estimation results of the SLAM system, and there is no universally recognized reliable evaluation criterion for the effect of map construction. For a semantic SLAM system, how to evaluate the accuracy of semantic information acquisition and how to evaluate the effect of semantic map construction are the issues that should be considered in the evaluation criteria of the semantic SLAM system. Furthermore, it is not a long-term solution to evaluate only by subjective indicators. In the future, it will be a hot topic how to establish systematic evaluation indicators for semantic VSLAM.

**Author Contributions:** Conceptualization, W.C., K.H. and G.S.; methodology, W.C., K.H. and G.S.; software, C.Z. and X.W.; formal analysis, W.C., K.H. and G.S.; investigation, W.C., K.H. and A.J.; writing—original draft preparation, G.S.; writing—review W.C., K.H. and G.S.; editing, W.C., K.H. and G.S.; visualization, A.J., X.W., C.X., and Z.L.; supervision, W.C., K.H. and A.J.; project administration, W.C., K.H. and A.J.; funding acquisition, A.J. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the National Key R&D programme of China (Grant Nos. 2019YFB1309600), National Natural Science Foundation of China (Grant Nos. 51875281 and 51861135306). These funds come from A.J.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** Research in this article was supported by the National Key R&D programme of China (Grant No. 2019YFB1309600), National Natural Science Foundation of China (Grant Nos. 51875281 and 51861135306) are deeply appreciated. The authors would like to express heartfelt thanks to the reviewers and editors who submitted valuable revisions to this article.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

SLAM	Stimulation Location and Mapping
VSLAM	Visual Stimulation Location and Mapping
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
IMU	Inertial Measurement Unit
EVO	Python package for the evaluation of odometry and SLAM
ATE	Absolute Trajectory Error
RPE	Relative Pose Error
TUM	Technical University of Munich
TOF	Time-Of-Flight
CPU	Central Processing Unit
GPU	Graphics Processing Unit
BoW	Bags of Binary Words
UKF	Unscented Kalman Filter
ICP	Iterative Closest Point
TSDF	Truncated Signed Distance Function
VO	Visual Odometry
VIO	Visual-Inertial Odometry
DNN	Deep Neural Networks
LSTM	Long Short-Term Memory Networks
GRU	Gated Recurrent Unit

3D	Three-Dimensional
EM	Expectation-Maximization
MIT	Massachusetts Institute of Technology
UAV	Unmanned Aerial Vehicle

## References

- Smith, R.C.; Cheeseman, P. On the Representation and Estimation of Spatial Uncertainty. *Int. J. Robot. Res.* **1986**, *5*, 56–68. [\[CrossRef\]](#)
- Deng, G.; Li, J.; Li, W.; Wang, H. SLAM: Depth image information for mapping and inertial navigation system for localization. In Proceedings of the 2016 Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), Tokyo, Japan, 20–22 July 2016; pp. 187–191.
- Cui, L.; Ma, C. SOF-SLAM: A Semantic Visual SLAM for Dynamic Environments. *IEEE Access* **2019**, *7*, 166528–166539 [\[CrossRef\]](#)
- Bresson, G.; Alsayed, Z.; Yu, L.; Glaser, S. Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving. *IEEE Trans. Intell. Veh.* **2017**, *2*, 194–220. [\[CrossRef\]](#)
- Karlsson, N.; Bernardo, E.d.; Ostrowski, J.; Goncalves, L.; Pirjanian, P.; Munich, M.E. The vSLAM Algorithm for Robust Localization and Mapping. In Proceedings of the Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; pp. 24–29.
- Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-time loop closure in 2D LIDAR SLAM. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278.
- Grisetti, G.; Stachniss, C.; Burgard, W. Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling. In Proceedings of the Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; pp. 2432–2437.
- Khan, M.U.; Zaidi, S.A.A.; Ishtiaq, A.; Bukhari, S.U.R.; Samer, S.; Farman, A. A Comparative Survey of LiDAR-SLAM and LiDAR based Sensor Technologies. In Proceedings of the 2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC), Karachi, Pakistan, 15–17 July 2021; pp. 1–8.
- Gupta, A.; Fernando, X. Simultaneous Localization and Mapping (SLAM) and Data Fusion in Unmanned Aerial Vehicles: Recent Advances and Challenges. *Drones* **2022**, *6*, 85. [\[CrossRef\]](#)
- Arun, A.; Nirmaladevi, P. A Survey on Current Semantic level Algorithms for improving Performance in CBIR. In Proceedings of the Materials Science and Engineering Conference Series, Chennai, India, 1 February 2021; p. 012118.
- Burguera, A.; Bonin-Font, F.; Font, E.G.; Torres, A.M. Combining Deep Learning and Robust Estimation for Outlier-Resilient Underwater Visual Graph SLAM. *J. Mar. Sci. Eng.* **2022**, *10*, 511. [\[CrossRef\]](#)
- Alatise, M.B.; Hancke, G.P. A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods. *IEEE Access* **2020**, *8*, 39830–39846. [\[CrossRef\]](#)
- Wang, P.; Cheng, J.; Feng, W. Efficient construction of topological semantic map with 3D information. *J. Intell. Fuzzy Syst.* **2018**, *35*, 3011–3020. [\[CrossRef\]](#)
- Wang, S.; Clark, R.; Wen, H.; Trigoni, N. End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. *Int. J. Robot. Res.* **2017**, *37*, 513–542. [\[CrossRef\]](#)
- Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [\[CrossRef\]](#)
- Taketomi, T.; Uchiyama, H.; Ikeda, S. Visual SLAM algorithms: A survey from 2010 to 2016. *IPSJ Trans. Comput. Vis. Appl.* **2017**, *9*, 16. [\[CrossRef\]](#)
- Yousif, K.; Bab-Hadiashar, A.; Hoseinnezhad, R. An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics. *Intell. Ind. Syst.* **2015**, *1*, 289–311. [\[CrossRef\]](#)
- Bavle, H.; Sánchez-López, J.L.; Schmidt, E.F.; Voos, H. From SLAM to Situational Awareness: Challenges and Survey. *arXiv* **2021**, arXiv:2110.00273.
- Servières, M.; Renaudin, V.; Dupuis, A.; Antigny, N. Visual and Visual-Inertial SLAM: State of the Art, Classification, and Experimental Benchmarking. *J. Sens.* **2021**, *2021*, 2054828. [\[CrossRef\]](#)
- Azzam, R.; Taha, T.; Huang, S.; Zweiri, Y. Feature-based visual simultaneous localization and mapping: A survey. *SN Appl. Sci.* **2020**, *2*, 224. [\[CrossRef\]](#)
- Macario Barros, A.; Michel, M.; Moline, Y.; Corre, G.; Carrel, F. A Comprehensive Survey of Visual SLAM Algorithms. *Robotics* **2022**, *11*, 24. [\[CrossRef\]](#)
- Li, R.; Wang, S.; Gu, D. Ongoing Evolution of Visual SLAM from Geometry to Deep Learning: Challenges and Opportunities. *Cogn. Comput.* **2018**, *10*, 875–889. [\[CrossRef\]](#)
- Medeiros Esper, I.; Smolkin, O.; Manko, M.; Popov, A.; From, P.J.; Mason, A. Evaluation of RGB-D Multi-Camera Pose Estimation for 3D Reconstruction. *Appl. Sci.* **2022**, *12*, 4134. [\[CrossRef\]](#)
- Zuo, Y.; Yang, J.; Chen, J.; Wang, X.; Wang, Y.; Kneip, L. DEVO: Depth-Event Camera Visual Odometry in Challenging Conditions. *arXiv* **2022**, arXiv:2202.02556.
- EVO. Python. Available online: <https://github.com/MichaelGrupp/evo> (accessed on 25 April 2022).

26. Bodin, B.; Wagstaff, H.; Saecdi, S.; Nardi, L.; Vespa, E.; Mawer, J.; Nisbet, A.; Lujan, M.; Furber, S.; Davison, A.J.; et al. SLAMBench2: Multi-Objective Head-to-Head Benchmarking for Visual SLAM. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 3637–3644.
27. Whelan, T.; Salas-Moreno, R.F.; Glocker, B.; Davison, A.J.; Leutenegger, S. ElasticFusion: Real-time dense SLAM and light source estimation. *Int. J. Robot. Res.* **2016**, *35*, 1697–1716. [CrossRef]
28. Mur-Artal, R.; Tardós, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
29. Leutenegger, S.; Lynen, S.; Bosse, M.; Siegwart, R.; Furgale, P. Keyframe-based visual-inertial odometry using nonlinear optimization. *Int. J. Robot. Res.* **2014**, *34*, 314–334. [CrossRef]
30. TUM RGB-D. Available online: <https://vision.in.tum.de/data/datasets/rgbd-dataset> (accessed on 25 April 2022).
31. TUM MonoVo. Available online: <http://vision.in.tum.de/mono-dataset> (accessed on 25 April 2022).
32. TUM VI. Available online: <https://vision.in.tum.de/data/datasets/visual-inertial-dataset> (accessed on 25 April 2022).
33. KITTI. Available online: <http://www.cvlibs.net/datasets/kitti/> (accessed on 22 May 2022).
34. EuRoc. Available online: <https://projects.asl.ethz.ch/datasets/doku.php?id=kmavvisualinertialdatasets> (accessed on 25 April 2022).
35. Cityscapes. Available online: <https://www.cityscapes-dataset.com/> (accessed on 25 April 2022).
36. ICL-NUIM. Available online: <https://www.doc.ic.ac.uk/~ahanda/VaFRIC/iclnuim.html> (accessed on 25 April 2022).
37. NYU RGB-D. Available online: <https://cs.nyu.edu/silberman/datasets/> (accessed on 25 April 2022).
38. MS COCO. Available online: <https://paperswithcode.com/dataset/coco> (accessed on 25 April 2022).
39. Engel, J.; Koltun, V.; Cremers, D. Direct Sparse Odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 611–625. [CrossRef] [PubMed]
40. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-Scale Direct Monocular SLAM. In Proceedings of the Computer Vision—ECCV 2014, Zurich, Switzerland, 6–12 September 2014; pp. 834–849.
41. Niko Sünderhauf. Available online: <https://nikosunderhauf.github.io/projects/sceneunderstanding/> (accessed on 12 June 2022).
42. SemanticSLAM.ai. Available online: <http://www.semanticslam.ai/> (accessed on 12 June 2022).
43. The Dyson Robotics Lab at Imperial College. Available online: <http://www.imperial.ac.uk/dyson-robotics-lab> (accessed on 25 April 2022).
44. Bloesch, M.; Czarnowski, J.; Clark, R.; Leutenegger, S.; Davison, A.J. CodeSLAM - Learning a Compact, Optimisable Representation for Dense Visual SLAM. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2560–2568.
45. McCormac, J.; Clark, R.; Bloesch, M.; Davison, A.; Leutenegger, S. Fusion++: Volumetric Object-Level SLAM. In Proceedings of the 2018 International Conference on 3D Vision (3DV), Verona, Italy, 5–8 September 2018; pp. 32–41.
46. McCormac, J.; Handa, A.; Davison, A.; Leutenegger, S. SemanticFusion: Dense 3D semantic mapping with convolutional neural networks. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 4628–4635.
47. Computer Vision Group TUM Department of Informatics Technical University of Munich. Available online: <https://vision.in.tum.de/research> (accessed on 25 April 2022).
48. Yang, N.; Stumberg, L.v.; Wang, R.; Cremers, D. D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1278–1289.
49. Stumberg, L.v.; Cremers, D. DM-VIO: Delayed Marginalization Visual-Inertial Odometry. *IEEE Robot. Autom. Lett.* **2022**, *7*, 1408–1415. [CrossRef]
50. Gao, X.; Wang, R.; Demmel, N.; Cremers, D. LDSO: Direct Sparse Odometry with Loop Closure. In Proceedings of the 2018 IEEE/RISJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 2198–2204.
51. Autonomous Intelligent Systems University of Freiburg. Available online: [http://ais.informatik.uni-freiburg.de/index\\_en.php](http://ais.informatik.uni-freiburg.de/index_en.php) (accessed on 23 May 2022).
52. Grisetti, G.; Stachniss, C.; Burgard, W. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. *IEEE Trans. Robot.* **2007**, *23*, 34–46. [CrossRef]
53. Endres, F.; Hess, J.; Sturm, J.; Cremers, D.; Burgard, W. 3-D mapping with an RGB-D camera. *IEEE Trans. Robot.* **2013**, *30*, 177–187. [CrossRef]
54. HKUST Aerial Robotics Group. Available online: <https://uav.hkust.edu.hk/> (accessed on 25 April 2022).
55. Qin, T.; Li, P.; Shen, S. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [CrossRef]
56. Qin, T.; Pan, J.; Cao, S.; Shen, S. A General Optimization-based Framework for Local Odometry Estimation with Multiple Sensors. *arXiv* **2019**, arXiv:1901.03638.
57. Zhou, Y.; Gallego, G.; Shen, S. Event-Based Stereo Visual Odometry. *IEEE Trans. Robot.* **2021**, *37*, 1433–1450. [CrossRef]
58. UW Robotics and State Estimation Lab. Available online: <http://rse-lab.cs.washington.edu/projects/> (accessed on 25 April 2022).
59. Schmidt, T.; Newcombe, R.; Fox, D. DART: Dense articulated real-time tracking with consumer depth cameras. *Auton. Robot.* **2015**, *39*, 239–258. [CrossRef]
60. Xiang, Y.; Fox, D. DA-RNN: Semantic Mapping with Data Associated Recurrent Neural Networks. *arXiv* **2017**, arXiv:1703.03098.

61. Henry, P.; Krainin, M.; Herbst, E.; Ren, X.; Fox, D. RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments. In *Experimental Robotics: The 12th International Symposium on Experimental Robotics*, Khatib, O., Kumar, V., Sukhatme, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; pp. 477–491.
62. Robotics, Perception and Real Time Group UNIVERSIDAD DE ZARAGOZA. Available online: <http://robots.unizar.es/slamlab/> (accessed on 25 April 2022).
63. Gálvez-López, D.; Salas, M.; Tardós, J.D.; Montiel, J.M.M. Real-time monocular object SLAM. *Robot. Auton. Syst.* **2016**, *75*, 435–449. [[CrossRef](#)]
64. Taheri, H.; Xia, Z.C. SLAM; definition and evolution. *Eng. Appl. Artif. Intell.* **2021**, *97*, 104032. [[CrossRef](#)]
65. Lin, L.; Wang, W.; Luo, W.; Song, L.; Zhou, W. Unsupervised monocular visual odometry with decoupled camera pose estimation. *Digit. Signal Process.* **2021**, *114*, 103052. [[CrossRef](#)]
66. Zhu, K.; Jiang, X.; Fang, Z.; Gao, Y.; Fujita, H.; Hwang, J.-N. Photometric transfer for direct visual odometry. *Knowl.-Based Syst.* **2021**, *213*, 106671. [[CrossRef](#)]
67. Guclu, O.; Can, A.B. k-SLAM: A fast RGB-D SLAM approach for large indoor environments. *Comput. Vis. Image Underst.* **2019**, *184*, 31–44. [[CrossRef](#)]
68. Cai, L.; Ye, Y.; Gao, X.; Li, Z.; Zhang, C. An improved visual SLAM based on affine transformation for ORB feature extraction. *Optik* **2021**, *227*, 165421. [[CrossRef](#)]
69. Harris, C.; Stephens, M. A Combined Corner and Edge Detector. In Proceedings of the Alvey Vision Conference, Manchester, UK, 31 August–2 September 1988.
70. Rosten, E.; Drummond, T. Machine Learning for High-Speed Corner Detection. In Proceedings of the Computer Vision—ECCV 2006, Graz, Austria, 7–13 May 2006; pp. 430–443.
71. Jianbo, S.; Tomasi. Good features to track. In Proceedings of the 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 21–23 June 1994; pp. 593–600.
72. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [[CrossRef](#)]
73. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [[CrossRef](#)]
74. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
75. Ali, A.M.; Nordin, M.J. SIFT based monocular SLAM with multi-clouds features for indoor navigation. In Proceedings of the TENCON 2010—2010 IEEE Region 10 Conference, Fukuoka, Japan, 21–24 November 2010; pp. 2326–2331.
76. Gioi, R.G.v.; Jakubowicz, J.; Morel, J.M.; Randall, G. LSD: A Fast Line Segment Detector with a False Detection Control. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 722–732. [[CrossRef](#)]
77. Davison, A.J.; Reid, I.D.; Molton, N.D.; Stasse, O. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1052–1067. [[CrossRef](#)]
78. Hu, K.; Wu, J.; Weng, L.; Zhang, Y.; Zheng, F.; Pang, Z.; Xia, M. A novel federated learning approach based on the confidence of federated Kalman filters. *Int. J. Mach. Learn. Cybern.* **2021**, *12*, 3607–3627. [[CrossRef](#)]
79. Klein, G.; Murray, D. Parallel Tracking and Mapping for Small AR Workspaces. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13–16 November 2007; pp. 225–234.
80. Mur-Artal, R.; Montiel, J.M.M.; Tardós, J.D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [[CrossRef](#)]
81. Galvez-López, D.; Tardós, J.D. Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Trans. Robot.* **2012**, *28*, 1188–1197. [[CrossRef](#)]
82. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.M.; Tardós, J.D. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [[CrossRef](#)]
83. Vakhitov, A.; Funke, J.; Moreno-Noguer, F. Accurate and Linear Time Pose Estimation from Points and Lines. In *Computer Vision—ECCV 2016*; Springer: Cham, Switzerland, 2016; pp. 583–599.
84. Smith, P.; Reid, I.D.; Davison, A.J. Real-Time Monocular SLAM with Straight Lines. *BMVC* **2006**, *6*, 17–26.
85. Perdices, E.; López, L.M.; Cañas, J.M. LineSLAM: Visual Real Time Localization Using Lines and UKF. In *ROBOT2013: First Iberian Robotics Conference: Advances in Robotics*; Armada, M.A., Sanfeliu, A., Ferre, M., Eds.; Springer International Publishing: Cham, Switzerland, 2014; Volume 1, pp. 663–678.
86. Montero, A.S.; Nayak, A.; Stojmenovic, M.; Zaguia, N. Robust line extraction based on repeated segment directions on image contours. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–7.
87. Klein, G.; Murray, D. Improving the Agility of Keyframe-Based SLAM. In Proceedings of the Computer Vision—ECCV 2008, Marseille, France, 12–18 October 2008; pp. 802–815.
88. Pumarola, A.; Vakhitov, A.; Agudo, A.; Sanfeliu, A.; Moreno-Noguer, F. PL-SLAM: Real-time monocular visual SLAM with points and lines. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 4503–4508.
89. Gomez-Ojeda, R.; Moreno, F.A.; Zuñiga-Noël, D.; Scaramuzza, D.; Gonzalez-Jimenez, J. PL-SLAM: A Stereo SLAM System Through the Combination of Points and Line Segments. *IEEE Trans. Robot.* **2019**, *35*, 734–746. [[CrossRef](#)]



90. Gee, A.P.; Chekhlov, D.; Calway, A.; Mayol-Cuevas, W. Discovering Higher Level Structure in Visual SLAM. *IEEE Trans. Robot.* **2008**, *24*, 980–990. [[CrossRef](#)]
91. Li, H.; Hu, Z.; Chen, X. PLP-SLAM: A Visual SLAM Method Based on Point-Line-Plane Feature Fusion. *ROBOT* **2017**, *39*, 214–220.
92. Zhang, N.; Zhao, Y. Fast and Robust Monocular Visual-Inertial Odometry Using Points and Lines. *Sensors* **2019**, *19*, 4545. [[CrossRef](#)]
93. He, X.; Gao, W.; Sheng, C.; Zhang, Z.; Pan, S.; Duan, L.; Zhang, H.; Lu, X. LiDAR-Visual-Inertial Odometry Based on Optimized Visual Point-Line Features. *Remote Sens.* **2022**, *14*, 622. [[CrossRef](#)]
94. Newcombe, R.A.; Lovegrove, S.J.; Davison, A.J. DTAM: Dense tracking and mapping in real-time. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2320–2327.
95. Forster, C.; Pizzoli, M.; Scaramuzza, D. SVO: Fast semi-direct monocular visual odometry. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 15–22.
96. Zhang, J.; Ganesh, P.; Volle, K.; Willis, A.; Brink, K. Low-Bandwidth and Compute-Bound RGB-D Planar Semantic SLAM. *Sensors* **2021**, *21*, 5400. [[CrossRef](#)] [[PubMed](#)]
97. Filatov, A.; Zaslavskiy, M.; Krinkin, K. Multi-Drone 3D Building Reconstruction Method. *Mathematics* **2021**, *9*, 3033. [[CrossRef](#)]
98. Newcombe, R.A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A.J.; Kohi, P.; Shotton, J.; Hodges, S.; Fitzgibbon, A. KinectFusion: Real-time dense surface mapping and tracking. In Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality, Basel, Switzerland, 26–29 October 2011; pp. 127–136.
99. Kaess, M.; Fallon, M.; Johannsson, H.; Leonard, J. Kintinuous: Spatially extended kinectfusion. In Proceedings of the RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras, Sydney, Australia, 9–10 July 2012; p. 9.
100. KinectFusion. Available online: <https://github.com/ParikaGoel/KinectFusion> (accessed on 21 April 2022).
101. Kintinuous. Available online: <https://github.com/mp3guy/Kintinuous> (accessed on 21 April 2022).
102. RGB-DSLAMv2. Available online: [https://github.com/felixendres/rgbdslam\\_v2](https://github.com/felixendres/rgbdslam_v2) (accessed on 21 April 2022).
103. ElasticFusion. Available online: <https://github.com/mp3guy/ElasticFusion> (accessed on 21 April 2022).
104. Yan, Z.; Ye, M.; Ren, L. Dense Visual SLAM with Probabilistic Surfel Map. *IEEE Trans. Vis. Comput. Graph.* **2017**, *23*, 2389–2398. [[CrossRef](#)]
105. DVO-SLAM. Available online: [https://github.com/tum-vision/dvo\\_slam](https://github.com/tum-vision/dvo_slam) (accessed on 21 April 2022).
106. Dai, A.; Nießner, M.; Zollhöfer, M.; Izadi, S.; Theobalt, C. BundleFusion: Real-Time Globally Consistent 3D Reconstruction Using On-the-Fly Surface Reintegration. *ACM Trans. Graph.* **2017**, *36*, 1. [[CrossRef](#)]
107. BundleFusion. Available online: <https://github.com/niessner/BundleFusion> (accessed on 21 April 2022).
108. Concha, A.; Civera, J. RGBDTAM: A cost-effective and accurate RGB-D tracking and mapping system. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), British, CO, Canada, 24–28 September 2017; pp. 6756–6763.
109. RGBDTAM. Available online: <https://github.com/alejocb/rgbdtam> (accessed on 21 April 2022).
110. Liu, Y.; Zhao, C.; Ren, M. An Enhanced Hybrid Visual-Inertial Odometry System for Indoor Mobile Robot. *Sensors* **2022**, *22*, 2930. [[CrossRef](#)]
111. Xie, H.; Chen, W.; Wang, J. Hierarchical forest based fast online loop closure for low-latency consistent visual-inertial SLAM. *Robot. Auton. Syst.* **2022**, *151*, 104035. [[CrossRef](#)]
112. Lee, W.; Eckenhoff, K.; Yang, Y.; Geneva, P.; Huang, G. Visual-Inertial-Wheel Odometry with Online Calibration. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA 24 October–24 January 2020; pp. 4559–4566.
113. Cheng, J.; Zhang, L.; Chen, Q. An Improved Initialization Method for Monocular Visual-Inertial SLAM. *Electronics* **2021**, *10*, 3063. [[CrossRef](#)]
114. Jung, J.H.; Cha, J.; Chung, J.Y.; Kim, T.I.; Seo, M.H.; Park, S.Y.; Yeo, J.Y.; Park, C.G. Monocular Visual-Inertial-Wheel Odometry Using Low-Grade IMU in Urban Areas. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 925–938. [[CrossRef](#)]
115. Weiss, S. Vision Based Navigation for Micro Helicopters. Ph.D. Thesis, ETH Zürich, Zürich, Switzerland, 2012.
116. Falquez, J.M.; Kasper, M.; Sibley, G. Inertial aided dense&semi-dense methods for robust direct visual odometry. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 3601–3607.
117. Mourikis, A.I.; Roumeliotis, S.I. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. In Proceedings of the Proceedings 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 3565–3572.
118. Wisht, D.; Camurri, M.; Das, S.; Fallon, M. Unified Multi-Modal Landmark Tracking for Tightly Coupled Lidar-Visual-Inertial Odometry. *IEEE Robot. Autom. Lett.* **2021**, *6*, 1004–1011. [[CrossRef](#)]
119. MonoSLAM. Available online: <https://github.com/rrg-polito/mono-slam> (accessed on 22 April 2022).
120. PTAM. Available online: <https://github.com/Oxford-PTAM/PTAM-GPL> (accessed on 22 April 2022).
121. ORB-SLAM2. Available online: [https://github.com/raulmur/ORB\\_SLAM2](https://github.com/raulmur/ORB_SLAM2) (accessed on 22 April 2022).
122. Gomez-Ojeda, R.; Briales, J.; Gonzalez-Jimenez, J. PL-SVO: Semi-direct Monocular Visual Odometry by combining points and line segments. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 4211–4216.
123. PL-SVO. Available online: <https://github.com/rubengooj/pl-svo> (accessed on 22 April 2022).
124. PL-SLAM. Available online: <https://github.com/rubengooj/pl-slam> (accessed on 22 April 2022).
125. DTAM. Available online: <https://github.com/anuranbaka/OpenDTAM> (accessed on 22 April 2022).

126. SVO. Available online: [https://github.com/uzh-rpg/rpg\\_svo](https://github.com/uzh-rpg/rpg_svo) (accessed on 22 April 2022).
127. LSD-SLAM. Available online: [https://github.com/tum-vision/llds\\_lam](https://github.com/tum-vision/llds_lam) (accessed on 21 April 2022).
128. DSO. Available online: <https://github.com/JakobEngel/dso> (accessed on 22 April 2022).
129. MSCKF-MONO. Available online: [https://github.com/daniilidis-group/msckf\\_mono](https://github.com/daniilidis-group/msckf_mono) (accessed on 22 April 2022).
130. OKVIS. Available online: <https://github.com/ethz-asl/okvis> (accessed on 21 April 2022).
131. Bloesch, M.; Omari, S.; Hutter, M.; Siegwart, R. Robust visual inertial odometry using a direct EKF-based approach. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 298–304.
132. ROVIO. Available online: <https://github.com/ethz-asl/rovio> (accessed on 22 April 2022).
133. VINS-Mono. Available online: <https://github.com/HKUST-Aerial-Robotics/VINS-Mono> (accessed on 22 April 2022).
134. Sualeh, M.; Kim, G.-W. Semantics Aware Dynamic SLAM Based on 3D MODT. *Sensors* **2021**, *21*, 6355. [[CrossRef](#)]
135. Wang, S.; Gou, G.; Sui, H.; Zhou, Y.; Zhang, H.; Li, J. CDSFusion: Dense Semantic SLAM for Indoor Environment Using CPU Computing. *Remote Sens.* **2022**, *14*, 979. [[CrossRef](#)]
136. Vishnyakov, B.; Sgibnev, I.; Sheverdin, V.; Sorokin, A.; Masalov, P.; Kazakhmedov, K.; Arseev, S. Real-time semantic slam with dcnn-based feature point detection, matching and dense point cloud aggregation. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **2021**, *XLIII-B2-2021*, 399–404. [[CrossRef](#)]
137. Li, P.; Zhang, G.; Zhou, J.; Yao, R.; Zhang, X.; Zhou, J. Study on Slam Algorithm Based on Object Detection in Dynamic Scene. In Proceedings of the 2019 International Conference on Advanced Mechatronic Systems (ICAMechS), Shiga, Japan, 26–28 August 2019; pp. 363–367.
138. Xu, D.; Vedaldi, A.; Henriques, J.F. Moving SLAM: Fully Unsupervised Deep Learning in Non-Rigid Scenes. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 4611–4617.
139. Hu, K.; Wu, J.; Li, Y.; Lu, M.; Weng, L.; Xia, M. FedGCN: Federated Learning-Based Graph Convolutional Networks for Non-Euclidean Spatial Data. *Mathematics*. **2022**, *10*, 1000. [[CrossRef](#)]
140. Hu, K.; Weng, C.; Zhang, Y.; Jin, J.; Xia, Q. An Overview of Underwater Vision Enhancement: From Traditional Methods to Recent Deep Learning. *J. Mar. Sci. Eng.* **2022**, *10*, 241. [[CrossRef](#)]
141. Hu, K.; Li, M.; Xia, M.; Lin, H. Multi-Scale Feature Aggregation Network for Water Area Segmentation. *Sensors* **2022**, *14*, 206. [[CrossRef](#)]
142. Lechelek, L.; Horna, S.; Zrour, R.; Naudin, M.; Guillevin, C. A Hybrid Method for 3D Reconstruction of MR Images. *J. Imaging* **2022**, *8*, 103. [[CrossRef](#)]
143. Hu, K.; Ding, Y.; Jin, J.; Weng, L.; Xia, M. Skeleton Motion Recognition Based on Multi-Scale Deep Spatio-Temporal Features. *Appl. Sci.* **2022**, *12*, 1028. [[CrossRef](#)]
144. Michael, E.; Summers, T.H.; Wood, T.A.; Manzie, C.; Shames, I. Probabilistic Data Association for Semantic SLAM at Scale. *arXiv* **2022**, arXiv:2202.12802.
145. Li, R.; Wang, S.; Gu, D. DeepSLAM: A Robust Monocular SLAM System With Unsupervised Deep Learning. *IEEE Trans. Ind. Electron.* **2021**, *68*, 3577–3587. [[CrossRef](#)]
146. Garg, R.; Bg, V.K.; Carneiro, G.; Reid, I. Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue. In Proceedings of the Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 11–14 October 2016; pp. 740–756.
147. Mukherjee, A.; Chakraborty, S.; Saha, S.K. Learning Deep Representation for Place Recognition in SLAM. In Proceedings of the Pattern Recognition and Machine Intelligence, Kolkata, India, 5–8 December 2017; pp. 557–564.
148. Gao, X.; Zhang, T. Unsupervised learning to detect loops using deep neural networks for visual SLAM system. *Auton. Robot.* **2017**, *41*, 1–18. [[CrossRef](#)]
149. Oh, J.; Eoh, G. Variational Bayesian Approach to Condition-Invariant Feature Extraction for Visual Place Recognition. *Appl. Sci.* **2021**, *11*, 8976. [[CrossRef](#)]
150. Mumuni, A.; Mumuni, F. CNN Architectures for Geometric Transformation-Invariant Feature Representation in Computer Vision: A Review. *SN Comput. Sci.* **2021**, *2*, 340. [[CrossRef](#)]
151. Ma, R.; Wang, R.; Zhang, Y.; Pizer, S.; McGill, S.K.; Rosenman, J.; Frahm, J.-M. RNN-SLAM: Reconstructing the 3D colon to visualize missing regions during a colonoscopy. *Med. Image Anal.* **2021**, *72*, 102100. [[CrossRef](#)] [[PubMed](#)]
152. Wang, K.; Ma, S.; Chen, J.; Ren, F.; Lu, J. Approaches, Challenges, and Applications for Deep Visual Odometry: Toward Complicated and Emerging Areas. *IEEE Trans. Cogn. Dev. Syst.* **2022**, *14*, 35–49. [[CrossRef](#)]
153. Hong, S.; Bangunharcana, A.; Park, J.-M.; Choi, M.; Shin, H.-S. Visual SLAM-Based Robotic Mapping Method for Planetary Construction. *Sensors* **2021**, *21*, 7715. [[CrossRef](#)]
154. Duan, C.; Junginger, S.; Huang, J.; Jin, K.; Thurow, K. Deep Learning for Visual SLAM in Transportation Robotics: A review. *Transp. Saf. Environ.* **2020**, *1*, 177–184. [[CrossRef](#)]
155. Loo, S.Y.; Shakeri, M.; Tang, S.H.; Mashohor, S.; Zhang, H. Online Mutual Adaptation of Deep Depth Prediction and Visual SLAM. *arXiv* **2021**, arXiv:2111.04096.
156. Kim, J.J.Y.; Urschler, M.; Riddle, P.J.; Wicker, J.S. SymbioLCD: Ensemble-Based Loop Closure Detection using CNN-Extracted Objects and Visual Bag-of-Words. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; p. 5425.

157. Steenbeek, A.; Nex, F. CNN-Based Dense Monocular Visual SLAM for Real-Time UAV Exploration in Emergency Conditions. *Drones* **2022**, *6*, 79. [[CrossRef](#)]
158. Tateno, K.; Tombari, F.; Laina, I.; Navab, N. CNN-SLAM: Real-Time Dense Monocular SLAM with Learned Depth Prediction. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6565–6574.
159. Yang, N.; Wang, R.; Stückler, J.; Cremers, D. Deep Virtual Stereo Odometry: Leveraging Deep Depth Prediction for Monocular Direct Sparse Odometry. In Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; pp. 835–852.
160. Godard, C.; Aodha, O.M.; Brostow, G.J. Unsupervised Monocular Depth Estimation with Left-Right Consistency. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6602–6611.
161. Qin, T.; Chen, T.; Chen, Y.; Su, Q. AVP-SLAM: Semantic Visual Mapping and Localization for Autonomous Vehicles in the Parking Lot. In Proceedings of the IEEE/RISJ International Conference on Intelligent Robots Systems, Las Vegas, NV, USA, 24 October 2020–24 January 2020; pp. 5939–5945.
162. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, Munich, Germany, 5–9 October 2015; pp. 234–241.
163. Zhu, R.; Yang, M.; Liu, W.; Song, R.; Yan, B.; Xiao, Z. DeepAVO: Efficient pose refining with feature distilling for deep Visual Odometry. *Neurocomputing* **2022**, *467*, 22–35. [[CrossRef](#)]
164. Luo, Y.; Xiao, Y.; Zhang, Y.; Zeng, N. Detection of loop closure in visual SLAM: A stacked assorted auto-encoder based approach. *Optoelectron. Lett.* **2021**, *17*, 354–360. [[CrossRef](#)]
165. Wen, S.; Zhao, Y.; Yuan, X.; Wang, Z.; Zhang, D.; Manfredi, L. Path planning for active SLAM based on deep reinforcement learning under unknown environments. *Intell. Serv. Robot.* **2020**, *13*, 263–272. [[CrossRef](#)]
166. Memon, A.R.; Wang, H.; Hussain, A. Loop closure detection using supervised and unsupervised deep neural networks for monocular SLAM systems. *Robot. Auton. Syst.* **2020**, *126*, 103470. [[CrossRef](#)]
167. Li, D.; Shi, X.; Long, Q.; Liu, S.; Yang, W.; Wang, F.; Wei, Q.; Qiao, F. DXSLAM: A robust and efficient visual SLAM system with deep features. In Proceedings of the 2020 IEEE/RISJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2020; pp. 4958–4965.
168. Qin, C.; Zhang, Y.; Liu, Y.; Lv, G. Semantic loop closure detection based on graph matching in multi-objects scenes. *J. Vis. Commun. Image Represent.* **2021**, *76*, 103072. [[CrossRef](#)]
169. Chen, C.; Wang, B.; Lu, C.X.; Trigoni, A.; Markham, A. A Survey on Deep Learning for Localization and Mapping: Towards the Age of Spatial Machine Intelligence. *arXiv* **2020**, arXiv:2006.12567.
170. Ye, X.; Ji, X.; Sun, B.; Chen, S.; Wang, Z.; Li, H. DRM-SLAM: Towards dense reconstruction of monocular SLAM with scene depth fusion. *Neurocomputing* **2020**, *396*, 76–91. [[CrossRef](#)]
171. Cao, J.; Zeng, B.; Liu, J.; Zhao, Z.; Su, Y. A novel relocation method for simultaneous localization and mapping based on deep learning algorithm. *Comput. Electr. Eng.* **2017**, *63*, 79–90. [[CrossRef](#)]
172. Arshad, S.; Kim, G.-W. Role of Deep Learning in Loop Closure Detection for Visual and Lidar SLAM: A Survey. *Sensors* **2021**, *21*, 1243. [[CrossRef](#)]
173. Li, R.; Wang, S.; Long, Z.; Gu, D. UnDeepVO: Monocular Visual Odometry Through Unsupervised Deep Learning. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 7286–7291.
174. DeTone, D.; Malisiewicz, T.; Rabinovich, A.J.A. Toward Geometric Deep SLAM. *arXiv* **2017**, arXiv:1707.07410.
175. Clark, R.; Wang, S.; Wen, H.; Markham, A.; Trigoni, A. VINet: Visual-Inertial Odometry as a Sequence-to-Sequence Learning Problem. In Proceedings of the AAAI, Québec City, QC, Canada, 23–26 October 2017.
176. Naseer, T.; Oliveira, G.L.; Brox, T.; Burgard, W. Semantics-aware visual localization under challenging perceptual conditions. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 2614–2620.
177. Bowman, S.L.; Atanasov, N.; Daniilidis, K.; Pappas, G.J. Probabilistic data association for semantic SLAM. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 1722–1729.
178. Hu, K.; Zheng, F.; Weng, L.; Ding, Y.; Jin, J. Action Recognition Algorithm of Spatio-Temporal Differential LSTM Based on Feature Enhancement. *Appl. Sci.* **2021**, *11*, 7876. [[CrossRef](#)]
179. Chen, W.; Zheng, F.; Gao, S.; Hu, K. An LSTM with Differential Structure and Its Application in Action Recognition. *Math. Probl. Eng.* **2022**, *2022*, 7316396. [[CrossRef](#)]
180. Cao, B.; Li, C.; Song, Y.; Qin, Y.; Chen, C. Network Intrusion Detection Model Based on CNN and GRU. *Appl. Sci.* **2022**, *12*, 4184. [[CrossRef](#)]
181. Chen, E.Z.; Wang, P.; Chen, X.; Chen, T.; Sun, S. Pyramid Convolutional RNN for MRI Image Reconstruction. *IEEE Trans. Med. Imaging* **2022**. [[CrossRef](#)] [[PubMed](#)]
182. Sang, H.; Jiang, R.; Wang, Z.; Zhou, Y.; He, B. A Novel Neural Multi-Store Memory Network for Autonomous Visual Navigation in Unknown Environment. *IEEE Robot. Autom. Lett.* **2022**, *7*, 2039–2046. [[CrossRef](#)]

183. Xue, F.; Wang, Q.; Wang, X.; Dong, W.; Wang, J.; Zha, H. Guided Feature Selection for Deep Visual Odometry. In Proceedings of the Computer Vision—ACCV 2018, Perth, Australia, 2–6 December 2018; pp. 293–308.
184. Teed, Z.; Deng, J. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *arXiv* **2021**, arXiv:2108.10869.
185. Turan, M.; Almalioglu, Y.; Araujo, H.; Konukoglu, E.; Sitti, M. Deep EndoVO: A recurrent convolutional neural network (RCNN) based visual odometry approach for endoscopic capsule robots. *Neurocomputing* **2018**, *275*, 1861–1870. [[CrossRef](#)]
186. Chancán, M.; Milford, M. DeepSeqSLAM: A Trainable CNN+RNN for Joint Global Description and Sequence-based Place Recognition. *arXiv* **2020**, arXiv:2011.08518.
187. Han, L.; Lin, Y.; Du, G.; Lian, S. DeepVIO: Self-supervised Deep Learning of Monocular Visual Inertial Odometry using 3D Geometric Constraints. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Venetian Macao, Macau, China, 3–8 November 2019; pp. 6906–6913.
188. Chen, C.; Rosa, S.; Miao, Y.; Lu, C.X.; Wu, W.; Markham, A.; Trigoni, N. Selective Sensor Fusion for Neural Visual-Inertial Odometry. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 10534–10543.
189. Almalioglu, Y.; Turan, M.; Saputra, M.R.U.; de Gusmão, P.P.B.; Markham, A.; Trigoni, N. SelfVIO: Self-supervised deep monocular Visual-Inertial Odometry and depth estimation. *Neural Netw.* **2022**, *150*, 119–136. [[CrossRef](#)]
190. Wong, A.; Fei, X.; Tsuei, S.; Soatto, S. Unsupervised Depth Completion From Visual Inertial Odometry. *IEEE Robot. Autom. Lett.* **2020**, *5*, 1899–1906. [[CrossRef](#)]
191. Wang, S.; Clark, R.; Wen, H.; Trigoni, N. DeepVO: Towards end-to-end visual odometry with deep Recurrent Convolutional Neural Networks. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 2043–2050.
192. Loo, S.Y.; Amiri, A.J.; Mashohor, S.; Tang, S.H.; Zhang, H. CNN-SVO: Improving the Mapping in Semi-Direct Visual Odometry Using Single-Image Depth Prediction. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 5218–5223.
193. Almalioglu, Y.; Saputra, M.R.U.; Gusmão, P.P.B.; Markham, A.; Trigoni, N. GANVO: Unsupervised Deep Monocular Visual Odometry and Depth Estimation with Generative Adversarial Networks. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 5474–5480.
194. Li, Y.; Ushiku, Y.; Harada, T. Pose Graph optimization for Unsupervised Monocular Visual Odometry. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 5439–5445.
195. Bruno, H.M.S.; Colombari, E.L. LIFT-SLAM: A deep-learning feature-based monocular visual SLAM method. *Neurocomputing* **2021**, *455*, 97–110. [[CrossRef](#)]
196. Zhang, S.; Lu, S.; He, R.; Bao, Z. Stereo Visual Odometry Pose Correction through Unsupervised Deep Learning. *Sensors* **2021**, *21*, 4735. [[CrossRef](#)]
197. Shamwell, E.J.; Lindgren, K.; Leung, S.; Nothwang, W.D. Unsupervised Deep Visual-Inertial Odometry with Online Error Correction for RGB-D Imagery. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2478–2493. [[CrossRef](#)] [[PubMed](#)]
198. Kim, Y.; Yoon, S.; Kim, S.; Kim, A. Unsupervised Balanced Covariance Learning for Visual-Inertial Sensor Fusion. *IEEE Robot. Autom. Lett.* **2021**, *6*, 819–826. [[CrossRef](#)]
199. Gurturk, M.; Yusefi, A.; Aslan, M.F.; Soykan, M.; Durdu, A.; Masiero, A. The YTU dataset and recurrent neural network based visual-inertial odometry. *Measurement* **2021**, *184*, 109878. [[CrossRef](#)]
200. Guan, P.; Cao, Z.; Chen, E.; Liang, S.; Tan, M.; Yu, J. A real-time semantic visual SLAM approach with points and objects. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 1729881420905443. [[CrossRef](#)]
201. Hempel, T.; Al-Hamadi, A. An online semantic mapping system for extending and enhancing visual SLAM. *Eng. Appl. Artif. Intell.* **2022**, *111*, 104830. [[CrossRef](#)]
202. Qian, Z.; Patath, K.; Fu, J.; Xiao, J. Semantic SLAM with Autonomous Object-Level Data Association. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 11203–11209.
203. Dr Pablo F Alcantarilla. Available online: <https://blog.slamcore.com/age-of-perception> (accessed on 12 June 2022).
204. Zhang, L.; Wei, L.; Shen, P.; Wei, W.; Zhu, G.; Song, J. Semantic SLAM Based on Object Detection and Improved Octomap. *IEEE Access* **2018**, *6*, 75545–75559. [[CrossRef](#)]
205. Hu, K.; Zhang, D.; Xia, M. CDUNet: Cloud Detection UNet for Remote Sensing Imagery. *Remote Sens.* **2021**, *13*, 4533. [[CrossRef](#)]
206. Hu, K.; Zhang, Y.; Weng, C.; Wang, P.; Deng, Z.; Liu, Y. An Underwater Image Enhancement Algorithm Based on Generative Adversarial Network and Natural Image Quality Evaluation Index. *J. Mar. Sci. Eng.* **2021**, *9*, 691. [[CrossRef](#)]
207. Pazhani, A.A.J.; Vasanthayaki, C. Object detection in satellite images by faster R-CNN incorporated with enhanced ROI pooling (FrRNet-ERoI) framework. *Earth Sci. Inform.* **2022**, *15*, 553–561. [[CrossRef](#)]
208. Hoang, T.M.; Zhou, J.; Fan, Y. Image Compression with Encoder-Decoder Matched Semantic Segmentation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 619–623.
209. Hu, K.; Lu, F.; Lu, M.; Deng, Z.; Liu, Y. A Marine Object Detection Algorithm Based on SSD and Feature Enhancement. *Complexity* **2020**, *2020*, 5476142. [[CrossRef](#)]
210. Shao, F.; Chen, L.; Shao, J.; Ji, W.; Xiao, S.; Ye, L.; Zhuang, Y.; Xiao, J. Deep Learning for Weakly Supervised Object Detection and Localization: A Survey. *Neurocomputing* **2022**, *496*, 192–207. [[CrossRef](#)]

211. Liang, W.; Xu, P.; Guo, L.; Bai, H.; Zhou, Y.; Chen, F. A survey of 3D object detection. *Multimed. Tools Appl.* **2021**, *80*, 29617–29641. [[CrossRef](#)]
212. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
213. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
214. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [[CrossRef](#)] [[PubMed](#)]
215. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
216. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
217. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
218. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [[CrossRef](#)] [[PubMed](#)]
219. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid Scene Parsing Network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6230–6239.
220. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R.B. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
221. YOLOV5. Available online: <https://github.com/ultralytics/yolov5> (accessed on 23 April 2022).
222. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv* **2014**, arXiv:1412.7062.
223. DeepLabv3+. Available online: <https://github.com/Tramac/awesome-semantic-segmentation-pytorch> (accessed on 23 April 2022).
224. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. YOLACT: Real-Time Instance Segmentation. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 9156–9165.
225. Dang, X.; Rong, Z.; Liang, X. Sensor Fusion-Based Approach to Eliminating Moving Objects for SLAM in Dynamic Environments. *Sensors* **2021**, *21*, 230. [[CrossRef](#)]
226. Tschopp, F.; Nieto, J.I.; Siegwart, R.Y.; Cadena, C. Superquadric Object Representation for Optimization-based Semantic SLAM. *arXiv* **2021**, arXiv:2109.09627.
227. Zhao, Z.; Mao, Y.; Ding, Y.; Ren, P.; Zheng, N. Visual-Based Semantic SLAM with Landmarks for Large-Scale Outdoor Environment. In Proceedings of the 2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI), Xi’an, China, 21–22 September 2019; pp. 149–154.
228. Lianos, K.-N.; Schönberger, J.L.; Pollefeys, M.; Sattler, T. VSO: Visual Semantic Odometry. In Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; pp. 246–263.
229. Stenborg, E.; Toft, C.; Hammarstrand, L. Long-Term Visual Localization Using Semantically Segmented Images. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 6484–6490.
230. Dai, W.; Zhang, Y.; Li, P.; Fang, Z.; Scherer, S. RGB-D SLAM in Dynamic Environments Using Point Correlations. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 373–389. [[CrossRef](#)]
231. Han, X.; Li, S.; Wang, X.; Zhou, W. Semantic Mapping for Mobile Robots in Indoor Scenes: A Survey. *Information* **2021**, *12*, 92. [[CrossRef](#)]
232. Liao, Z.; Hu, Y.; Zhang, J.; Qi, X.; Zhang, X.; Wang, W. SO-SLAM: Semantic Object SLAM With Scale Proportional and Symmetrical Texture Constraints. *IEEE Robot. Autom. Lett.* **2022**, *7*, 4008–4015. [[CrossRef](#)]
233. Ran, T.; Yuan, L.; Zhang, J.; He, L.; Huang, R.; Mei, J. Not Only Look However, Infer: Multiple Hypothesis Clustering of Data Association Inference for Semantic SLAM. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–9. [[CrossRef](#)]
234. Yin, R.; Cheng, Y.; Wu, H.; Song, Y.; Yu, B.; Niu, R. FusionLane: Multi-Sensor Fusion for Lane Marking Semantic Segmentation Using Deep Neural Networks. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 1543–1553. [[CrossRef](#)]
235. Han, B.; Xu, L. A Monocular SLAM System with Mask Loop Closing. In Proceedings of the 2020 Chinese Control and Decision Conference (CCDC), Hefei, China, 22–24 August 2020; pp. 4762–4768.
236. Yang, S.; Fan, G.; Bai, L.; Zhao, C.; Li, D. SGC-VSLAM: A Semantic and Geometric Constraints VSLAM for Dynamic Indoor Environments. *Sensors* **2020**, *20*, 2432. [[CrossRef](#)] [[PubMed](#)]
237. Galindo, C.; Saffiotti, A.; Coradeschi, S.; Buschka, P.; Fernandez-Madrigal, J.A.; Gonzalez, J. Multi-hierarchical semantic maps for mobile robotics. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, AB, Canada, 2–6 August 2005; pp. 2278–2283.
238. Vasudevan, S.; Gächter, S.; Nguyen, V.; Siegwart, R. Cognitive maps for mobile robots—an object based approach. *Robot. Auton. Syst.* **2007**, *55*, 359–371. [[CrossRef](#)]
239. Yue, Y.; Wen, M.; Zhao, C.; Wang, Y.; Wang, D. COSEM: Collaborative Semantic Map Matching Framework for Autonomous Robots. *IEEE Trans. Ind. Electron.* **2022**, *69*, 3843–3853. [[CrossRef](#)]

240. Ashour, R.; Abdelkader, M.; Dias, J.; Almoosa, N.I.; Taha, T. Semantic Hazard Labelling and Risk Assessment Mapping During Robot Exploration. *IEEE Access* **2022**, *10*, 16337–16349. [[CrossRef](#)]
241. Rosinol, A.; Abate, M.; Chang, Y.; Carlone, L. Kimera: An Open-Source Library for Real-Time Metric-Semantic Localization and Mapping. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), virtually, 31 May–31 August 2020; pp. 1689–1696.
242. Mingyuan, M.; Hewei, Z.; Simeng, L.; Baochang, Z. SEMANTIC-RTAB-MAP (SRM): A semantic SLAM system with CNNs on depth images. *Math. Found. Comput.* **2019**, *2*, 29–41.
243. Labbé, M.; Michaud, F. Appearance-Based Loop Closure Detection for Online Large-Scale and Long-Term Operation. *IEEE Trans. Robot.* **2013**, *29*, 734–745. [[CrossRef](#)]
244. Menini, D.; Kumar, S.; Oswald, M.R.; Sandström, E.; Sminchisescu, C.; Gool, L.V. A Real-Time Online Learning Framework for Joint 3D Reconstruction and Semantic Segmentation of Indoor Scenes. *IEEE Robot. Autom. Lett.* **2022**, *7*, 1332–1339. [[CrossRef](#)]
245. Zhuang, C.; Wang, Z.; Zhao, H.; Ding, H. Semantic part segmentation method based 3D object pose estimation with RGB-D images for bin-picking. *Robot. Comput.-Integr. Manuf.* **2021**, *68*, 102086. [[CrossRef](#)]
246. Sousa, Y.C.; Bassani, H.d.F. Topological Semantic Mapping by Consolidation of Deep Visual Features. *IEEE Robot. Autom. Lett.* **2022**, *7*, 4110–4117. [[CrossRef](#)]
247. Wang, F.; Zhang, C.; Zhang, W.; Fang, C.; Xia, Y.; Liu, Y.; Dong, H. Object-Based Reliable Visual Navigation for Mobile Robot. *Sensors* **2022**, *22*, 2387. [[CrossRef](#)] [[PubMed](#)]
248. Wu, Y.; Zhang, Y.; Zhu, D.; Feng, Y.; Coleman, S.; Kerr, D. EAO-SLAM: Monocular Semi-Dense Object SLAM Based on Ensemble Data Association. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October–24 January 2020; pp. 4966–4973.
249. Wang, J.; Rünz, M.; Agapito, L. DSP-SLAM: Object Oriented SLAM with Deep Shape Priors. In Proceedings of the 2021 International Conference on 3D Vision (3DV), London, UK, 1–3 December 2021; pp. 1362–1371.
250. Fu, J.; Huang, Q.; Doherty, K.; Wang, Y.; Leonard, J.J. A Multi-Hypothesis Approach to Pose Ambiguity in Object-Based SLAM. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 7639–7646.
251. Zhai, R.; Yuan, Y. A Method of Vision Aided GNSS Positioning Using Semantic Information in Complex Urban Environment. *Remote Sens.* **2022**, *14*, 869. [[CrossRef](#)]
252. Vineet, V.; Miksik, O.; Lidegaard, M.; Nießner, M.; Golodetz, S.; Prisacariu, V.A.; Kähler, O.; Murray, D.W.; Izadi, S.; Pérez, P.; et al. Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 75–82.
253. Zhao, Z.; Chen, X. Building 3D semantic maps for mobile robots using RGB-D camera. *Intell. Serv. Robot.* **2016**, *9*, 297–309. [[CrossRef](#)]
254. Li, X.; Belaroussi, R. Semi-Dense 3D Semantic Mapping from Monocular SLAM. *arXiv* **2016**, arXiv:1611.04144.
255. Yang, S.; Huang, Y.; Scherer, S. Semantic 3D occupancy mapping through efficient high order CRFs. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 590–597.
256. Narita, G.; Seno, T.; Ishikawa, T.; Kaji, Y. PanopticFusion: Online Volumetric Semantic Mapping at the Level of Stuff and Things. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Venetian Macao, Macau, China, 3–8 November 2019; pp. 4205–4212.
257. Qin, T.; Zheng, Y.; Chen, T.; Chen, Y.; Su, Q. A Light-Weight Semantic Map for Visual Localization towards Autonomous Driving. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi’an, China, 30 May–5 June 2021; pp. 11248–11254.
258. Yan, L.; Hu, X.; Zhao, L.; Chen, Y.; Wei, P.; Xie, H. DGS-SLAM: A Fast and Robust RGBD SLAM in Dynamic Environments Combined by Geometric and Semantic Information. *Remote Sens.* **2022**, *14*, 795. [[CrossRef](#)]
259. Hu, Z.; Zhao, J.; Luo, Y.; Ou, J. Semantic SLAM Based on Improved DeepLabv3+ in Dynamic Scenarios. *IEEE Access* **2022**, *10*, 21160–21168. [[CrossRef](#)]
260. Liu, X.; Nardari, G.V.; Ojeda, F.C.; Tao, Y.; Zhou, A.; Donnelly, T.; Qu, C.; Chen, S.W.; Romero, R.A.F.; Taylor, C.J.; et al. Large-Scale Autonomous Flight With Real-Time Semantic SLAM Under Dense Forest Canopy. *IEEE Robot. Autom. Lett.* **2022**, *7*, 5512–5519. [[CrossRef](#)]
261. Chen, B.; Peng, G.; He, D.; Zhou, C.; Hu, B. Visual SLAM Based on Dynamic Object Detection. In Proceedings of the 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 22–24 May 2021; pp. 5966–5971.
262. Bescos, B.; Fàcil, J.M.; Civera, J.; Neira, J. DynaSLAM: Tracking, Mapping, and Inpainting in Dynamic Scenes. *IEEE Robot. Autom. Lett.* **2018**, *3*, 4076–4083. [[CrossRef](#)]
263. Yu, C.; Liu, Z.; Liu, X.J.; Xie, F.; Yang, Y.; Wei, Q.; Fei, Q. DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1168–1174.
264. Kaneko, M.; Iwami, K.; Ogawa, T.; Yamasaki, T.; Aizawa, K. Mask-SLAM: Robust Feature-Based Monocular SLAM by Masking Using Semantic Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 371–3718.

265. Schörghuber, M.; Steininger, D.; Cabon, Y.; Humenberger, M.; Gelautz, M. SLAMANTIC—Leveraging Semantics to Improve VSLAM in Dynamic Environments. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea, 27–28 October 2019; pp. 3759–3768.
266. Lv, X.; Wang, B.; Ye, D.; Wang, S.J.A. Semantic Flow-guided Motion Removal Method for Robust Mapping. *arXiv* **2020**, arXiv:2010.06876.
267. Yuan, X.; Chen, S. SaD-SLAM: A Visual SLAM Based on Semantic and Depth Information. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October–24 January 2020; pp. 4930–4935.
268. Wen, S.; Li, P.; Zhao, Y.; Zhang, H.; Sun, F.; Wang, Z. Semantic visual SLAM in dynamic environment. *Auton. Robot.* **2021**, *45*, 493–504. [CrossRef]
269. Wu, J.; Xiong, J.; Guo, H. Improving robustness of line features for VIO in dynamic scene. *Meas. Sci. Technol.* **2022**, *33*, 065204. [CrossRef]
270. Wang, M.; Wang, H.; Wang, Z.; Li, Y. A UAV Visual Relocalization Method Using Semantic Object Features Based on Internet of Things. In Proceedings of the Wireless Communications Mobile Computing, Dubrovnik, Croatia, 30 May–3 June 2022.
271. Lu, Z.; Hu, Z.; Uchimura, K. SLAM Estimation in Dynamic Outdoor Environments: A Review. In Proceedings of the Intelligent Robotics and Applications, Singapore, 16–18 December 2009; pp. 255–267.
272. Reddy, N.D.; Abbasnejad, I.; Reddy, S.; Mondal, A.K.; Devalla, V. Incremental real-time multibody VSLAM with trajectory optimization using stereo camera. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 4505–4510.
273. Lenz, P.; Ziegler, J.; Geiger, A.; Roser, M. Sparse scene flow segmentation for moving object detection in urban environments. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 5–9 June 2011; pp. 926–932.
274. DynaSLAM. Available online: <https://github.com/BertaBescos/DynaSLAM> (accessed on 24 April 2022).
275. DS-SLAM. Available online: <https://github.com/ivipsourcecode/DS-SLAM> (accessed on 24 April 2022).
276. Zhong, F.; Wang, S.; Zhang, Z.; Chen, C.; Wang, Y. Detect-SLAM: Making Object Detection and SLAM Mutually Beneficial. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 1001–1010.
277. Detect-SLAM. Available online: <https://github.com/liadbiz/detect-slam> (accessed on 24 April 2022).
278. Wang, Z.; Zhang, Q.; Li, J.; Zhang, S.; Liu, J. A Computationally Efficient Semantic SLAM Solution for Dynamic Scenes. *Remote Sens.* **2019**, *11*, 1363. [CrossRef]
279. SLAMANTIC. Available online: <https://github.com/mthz/slamantic> (accessed on 25 April 2022).
280. Barsan, I.A.; Liu, P.; Pollefeys, M.; Geiger, A. Robust Dense Mapping for Large-Scale Dynamic Environments. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 7510–7517.
281. Dai, J.; He, K.; Sun, J. Instance-Aware Semantic Segmentation via Multi-task Network Cascades. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3150–3158.
282. DynSLAM. Available online: <https://github.com/AndreiBarsan/DynSLAM> (accessed on 25 April 2022).
283. Esparza, D.; Flores, G. The STDyn-SLAM: A Stereo Vision and Semantic Segmentation Approach for VSLAM in Dynamic Outdoor Environments. *IEEE Access* **2022**, *10*, 18201–18209. [CrossRef]
284. STDyn-SLAM. Available online: <https://github.com/DanielaEsparza/STDyn-SLAM> (accessed on 25 April 2022).
285. Zhang, T.; Nakamura, Y. Posefusion: Dense rgb-d slam in dynamic human environments. In Proceedings of the International Symposium on Experimental Robotics, Buenos Aires, Argentina, 5–8 November 2018; pp. 772–780.
286. PoseFusion. Available online: <https://github.com/conix-center/posefusion> (accessed on 25 April 2022).
287. Liu, Y.; Miura, J. RDS-SLAM: Real-Time Dynamic SLAM Using Semantic Segmentation Methods. *IEEE Access* **2021**, *9*, 23772–23785. [CrossRef]
288. RDS-SLAM. Available online: <https://github.com/yubaoliu/RDS-SLAM> (accessed on 25 April 2022).
289. Lai, D.; Li, C.; He, B. YO-SLAM: A Robust Visual SLAM towards Dynamic Environments. In Proceedings of the 2021 International Conference on Communications, Information System and Computer Engineering (CISCE), Beijing, China, 14–16 May 2021; pp. 720–725.
290. Zhang, Y.; Xu, X.; Zhang, N.; Lv, Y. A Semantic SLAM System for Catadioptric Panoramic Cameras in Dynamic Environments. *Sensors* **2021**, *21*, 5889. [CrossRef] [PubMed]
291. Schönbein, M.; Geiger, A. Omnidirectional 3D reconstruction in augmented Manhattan worlds. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 716–723.
292. Hu, X.; Lang, J. DOE-SLAM: Dynamic Object Enhanced Visual SLAM. *Sensors* **2021**, *21*, 3091. [CrossRef]
293. Yu, N.; Gan, M.; Yu, H.; Yang, K. DRSO-SLAM: A Dynamic RGB-D SLAM Algorithm for Indoor Dynamic Scenes. In Proceedings of the 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 22–24 May 2021; pp. 1052–1058.
294. Ai, Y.; Rui, T.; Lu, M.; Fu, L.; Liu, S.; Wang, S. DDL-SLAM: A Robust RGB-D SLAM in Dynamic Environments Combined With Deep Learning. *IEEE Access* **2020**, *8*, 162335–162342. [CrossRef]
295. Liu, Y.; Miura, J. RDMO-SLAM: Real-Time Visual SLAM for Dynamic Environments Using Semantic Label Prediction With Optical Flow. *IEEE Access* **2021**, *9*, 106981–106997. [CrossRef]