Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.041898

REVIEW

Visual SLAM Based on Object Detection Network: A Review

Jiansheng Peng[1,2,*], Dunhua Chen[1], Qing Yang[1], Chengjun Yang[2], Yong Xu[2] and Yong Qin[2]

[1]College of Automation, Guangxi University of Science and Technology, Liuzhou, 545000, China

[2]Department of Artificial Intelligence and Manufacturing, Hechi University, Hechi, 547000, China

*Corresponding Author: Jiansheng Peng. Email: sheng120410@163.com

Received: 10 May 2023 Accepted: 27 October 2023 Published: 26 December 2023

ABSTRACT

Visual simultaneous localization and mapping (SLAM) is crucial in robotics and autonomous driving. However, traditional visual SLAM faces challenges in dynamic environments. To address this issue, researchers have proposed semantic SLAM, which combines object detection, semantic segmentation, instance segmentation, and visual SLAM. Despite the growing body of literature on semantic SLAM, there is currently a lack of comprehensive research on the integration of object detection and visual SLAM. Therefore, this study aims to gather information from multiple databases and review relevant literature using specific keywords. It focuses on visual SLAM based on object detection, covering different aspects. Firstly, it discusses the current research status and challenges in this field, highlighting methods for incorporating semantic information from object detection networks into mileage measurement, closed-loop detection, and map construction. It also compares the characteristics and performance of various visual SLAM object detection algorithms. Lastly, it provides an outlook on future research directions and emerging trends in visual SLAM. Research has shown that visual SLAM based on object detection has significant improvements compared to traditional SLAM in dynamic point removal, data association, point cloud segmentation, and other technologies. It can improve the robustness and accuracy of the entire SLAM system and can run in real time. With the continuous optimization of algorithms and the improvement of hardware level, object visual SLAM has great potential for development.

KEYWORDS

Object detection; visual SLAM; visual odometry; loop closure detection; semantic map

1 Introduction

SLAM (Simultaneous localization and mapping) can help robots achieve the task of obstacle avoidance and navigation in unknown environments and is an indispensable component of robot intelligence. Its localization and mapping functions are widely used in various fields such as augmented reality [1,2], driverless [3], logistics navigation [4], medical devices [5,6], and map reconstruction [7,8].

There are two mainstream SLAM techniques, namely Lidar SLAM and visual SLAM (VSLAM). The basic principle of Lidar SLAM is to acquire the depth information of the environment by Lidar sensors and then use this information for simultaneous localization and map building. Its main steps include scan matching, feature extraction, data association, state estimation, and map update. The

LiDAR sensor can calculate the accurate distance information of all objects through the laser round-trip time, so high localization and map-building accuracy can be achieved. In addition, Lidar is more stable in environments with changing lighting conditions, low light, or darkness. Lidar uses its light source to emit the laser beam and is therefore not disturbed by external light. Visual SLAM derives its information from cameras, which have low manufacturing costs and real-time access to the surrounding information, so visual SLAM has developed rapidly in recent years. Visual SLAM calculates the actual distance and direction of an object based on its position in the images. There are also many SLAM systems incorporating IMU sensors [9], which utilize the visual localization information to estimate the zero-bias of the IMU and reduce the cumulative error of the IMU caused by the zero-bias; on the contrary, the IMU can provide the vision with localization during fast motion as well as preventing the vision from tracking failures due to image occlusion, so the IMU and the camera can perfectly complement each other's strengths. Traditional visual SLAM treats the nearby environment as static, which brings him a great limitation in dynamic environments, such as pedestrians on the street side and vehicles running on the highway, and these dynamic objects can seriously affect the accuracy of localization and cause bias in environment building. In addition, in the case of sparse or occluded feature points, visual SLAM will lose tracking and have poor loop closure detection.

Since visual SLAM alone has many drawbacks, many scientists and scholars try to incorporate semantic information into visual SLAM to improve the localization accuracy and robustness greatly, and the commonly used methods based on deep learning to extract image semantic information include object detection [10–12], semantic segmentation [13,14] and instance segmentation [15], as shown in Fig. 1, all three techniques in visual SLAM have notable performance in visual SLAM. For example, Zhang et al. [16] used the YOLOv3 [17] object detection algorithm to detect potential objects by first obtaining the object region using RGB images, then using k-means clustering for the object region in the depth map to obtain the object mask, and finally using a multi-view geometric approach to determine the motion of the object. Han et al. [18] filtered dynamic objects by adding a new semantic segmentation thread. PSPNet-SLAM rejects dynamic feature points in two steps, firstly, optical flow is calculated for all feature points, and those with optical flow values greater than a certain range are set as dynamic points, the second step is to use the semantic segmentation network to obtain dynamic object masks to remove the feature points remaining in the dynamic object region, and after two screenings, complete static feature points are obtained, which improves the localization accuracy of SLAM in dynamic environments. Reference [19] used the more effective 2D instance segmentation module to extract the semantic information of potential dynamic objects, and in order to achieve the real-time requirement, the neural network is trained offline first then the real-time image is processed online, and finally the multimodal fusion module is used to further enhance the segmentation effect and remove the dynamic objects, which has excellent robustness in complex environments.

The advantage of object detection over semantic segmentation and instance segmentation is reflected in its real-time nature, which is one of the reasons why it is widely used in SLAM systems. The combination of target detection and visual SLAM also faces many difficulties, and the target detection algorithm based on deep learning is limited by external conditions and has a large upside from hardware to algorithms. The following are the challenges of deep learning-based target detection and the research difficulties of object SLAM, respectively.

Figure 1: Semantic information acquisition

The application of deep learning in the field of object detection has made significant progress, but it also faces some challenges:

(1) Requires a large amount of annotation data: Deep learning requires a large amount of annotation data for training, but obtaining annotation data requires manual participation, which is time-consuming, labor-intensive, and costly.

(2) The generalization ability of the model is limited: the deep learning model is prone to overfitting, that is, it performs well in training data, but poorly in test data.

(3) There are many types and quantities of targets: Deep learning object detection algorithms require recognition and localization of different types and quantities of targets, which puts higher demands on the complexity and computational resource requirements of the model. In complex scenes, multiple target objects may appear simultaneously, which makes target localization and recognition more difficult.

(4) The shape and scale of the target vary greatly: the same object can also change in its shape and appearance at different scales, and deep learning models need to have a certain degree of robustness to cope with such changes.

SLAM based on object detection is a method that combines object detection and SLAM technology, which faces the following challenges:

(1) Difficulty in meeting real-time requirements: target detection and SLAM often need to operate under real-time requirements, especially in applications such as robot navigation and autonomous driving. Real-time requirements mean that the algorithms need to process a large amount of sensor data and output accurate target detection and localization results in a limited amount of time, thus requiring efficient data processing and algorithm implementation.

(2) Difficulty in data association: data association is the process of matching and associating target detection results with maps in SLAM. This is a difficult task because target detection and SLAM use different data representations, time steps, and coordinate systems, and problems such as target ID matching, target identification, and trajectory inference across time steps need to be addressed.

(3) Dynamic environment processing: the combination of target detection and SLAM requires the processing of targets in dynamic environments, such as moving vehicles, pedestrians, or other obstacles. This requires effective target tracking and modeling methods to adapt to target motion and state changes. The difficulty is due to the uncertainty, occlusion, and diversity of motion patterns of dynamic targets, requiring the design of robust target-tracking algorithms and motion models.

There are numerous reviews on semantic SLAM [20–22]. Li et al. [23] made a detailed analysis and summary of semantic SLAM, reviewing the advantages of semantic SLAM from three aspects:

semantic segmentation, target detection, and instance segmentation, and analyzing the improvement of semantic information on the accuracy and robustness of SLAM systems, but the article did not make a detailed description of the application of target detection separately. In previous articles, they were put together and discussed, but no comparison was made between target detection and semantic segmentation, as well as instance segmentation, which could not clearly reflect their respective characteristics and advantages. This paper reviews the visual SLAM technique based on target detection. Chapter 2 introduces the current mainstream target detection networks and traditional visual SLAM algorithms, listing some of the latest algorithms; Chapter 3 introduces various methods for fusing target detection techniques in SLAM, discussing the problems solved by these methods as well as their advantages and disadvantages, Chapter 4 compares target detection-based visual SLAM with other popular visual SLAM techniques, Chapter 5 looks at the object SLAM development trend, and finally concludes the whole paper.

2 Overview of Visual SLAM System Based on Object Detection

2.1 Basic Framework of Visual SLAM

Visual SLAM is a key technology in the field of robotics and computer vision for realizing the ability of a mobile robot or camera to simultaneously localize its own position in an unknown environment and construct a map of the environment. The process is similar to a human walking in an unfamiliar place while simultaneously recognizing its surroundings and confirming its position. The main goal is to capture real-time image information by using a camera or other vision sensor to determine the position and orientation of the robot or camera in 3D space and to construct a map in the process, sometimes with the assistance of an IMU for localization. Visual SLAM includes techniques such as feature extraction and matching, data association, motion estimation, and loop detection. As shown in Fig. 2, in the traditional visual SLAM [24] framework, there are five important components, firstly, it relies on the sensors to obtain the environment information, which is usually an image or a video stream; in the front-end part, these data are needed to be preprocessed, and then based on the feature-point matching technique to calculate the camera's displacement with respect to the previous frame to get the robot's current position, which is also known as the position estimation; in the back-end, the filtering method is used or nonlinear optimization method for attitude correction, to get more accurate motion trajectory and position; the role of loopback detection is to reduce the estimation error accumulated over a long period of time, to determine whether the current frame has appeared in the previous one, and if it has, it means that it is back to the origin, and it can be reevaluated and optimized for the trajectory; the last step is to build a map, according to the position of each object in the image in the world coordinate system, and use the coordinate system of the transformation technology to reason out the position of the surrounding objects in our map, to construct a two-dimensional map or three-dimensional map.

The current mainstream visual SLAMs are ORB-SLAM2 [25], DM-VIO [26], Dso-SLAM [27], ORB-SLAM3 [28], and VINS-Mono [29]. ORB-SLAM3 is the first feature-based tightly coupled [30] VIO system that introduces maximum a posteriori estimation in the initialization part of the IMU. Relying on maximum a posteriori estimation and MLPnP [31] for bit-pose estimation, it is the most widely used visual SLAM framework with the highest accuracy. The classical visual SLAM development history is shown in Fig. 3. This article describes several common traditional SLAM systems as follows.

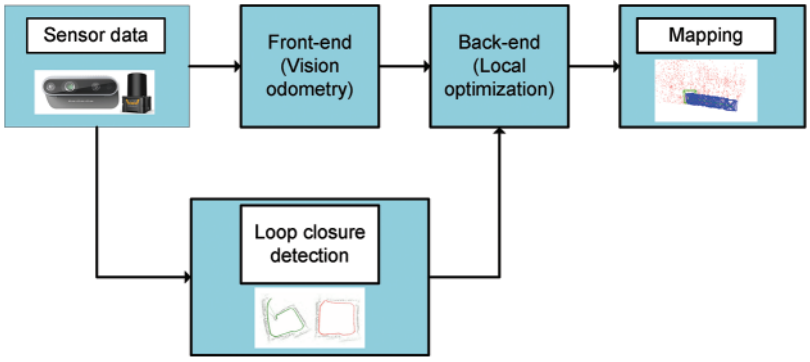


Figure 2: The framework of VSLAM

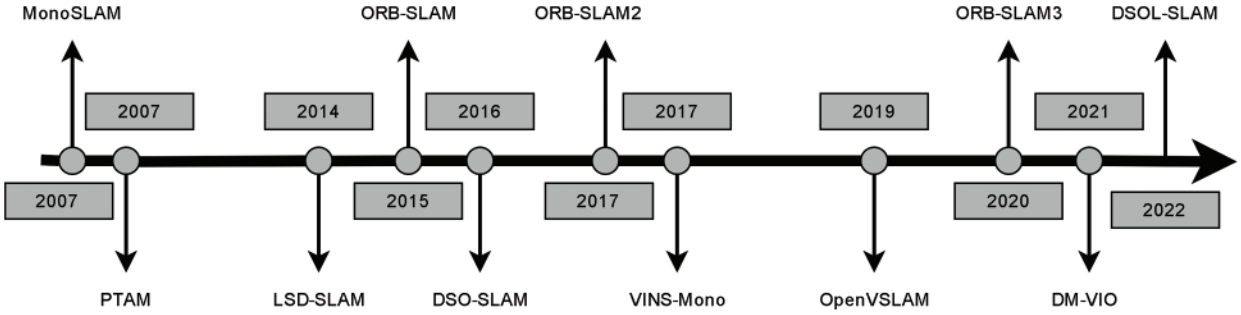


Figure 3: Classical visual SLAM development history

ORB-SLAM2 is a real-time, robust, and highly accurate visual simultaneous localization and mapping (SLAM) algorithm. It was introduced in 2016 by Mur-Artal et al. [25]. ORB-SLAM2 uses a combination of ORB (Oriented FAST and Rotated BRIEF) features and the extended Kalman filter to estimate the pose of a camera and create a 3D map of the environment. It can handle various types of camera motions, including rotation, translation, and scale changes, and can be used with both monocular and stereo cameras. One of the key features of ORB-SLAM2 is its ability to handle large-scale environments with high accuracy and robustness, even in challenging conditions such as low light or dynamic scenes. It also includes loop closure detection and relocalization capabilities, which help to correct drift and maintain accurate localization over time.

DSO-SLAM is a real-time visual SLAM algorithm that was introduced in 2016 by Engel et al. [27]. DSO-SLAM uses direct image alignment, which means it directly minimizes the photometric error between two consecutive images, without the need for feature detection or tracking. This allows DSO-SLAM to operate with high efficiency and accuracy, even in low-texture environments. One of the key features of DSO-SLAM is its ability to estimate the depth of the scene in real time. DSO-SLAM uses a depth filtering technique that combines depth estimates from multiple frames to produce a more accurate and robust depth map. It has also been extended and improved upon by researchers, such as with the introduction of Semi-Direct Visual Odometry (SVO) and Stereo DSO.

OpenVSLAM is an open-source feature point-based framework that supports multiple sensors, including monocular, binocular, and RGB-D cameras, giving it flexibility across a variety of devices and scenarios. OpenVSLAM uses incremental and global optimization algorithms to improve the accuracy of maps and camera poses. It provides camera localization, map construction, closed-loop

detection, and optimization with the capability of a complete visual SLAM system. However, when dealing with some complex scenes and fast movements, localization errors, and map drift may occur, and further algorithm improvements and optimization are needed.

ORB-SLAM3 (Fig. 4) is the latest version of the ORB-SLAM series, which has higher accuracy, better robustness, and faster operation speed than the previous versions. ORB-SLAM3 supports multiple sensors, including a monocular camera, binocular camera, RGB-D camera, and lidar. It implements multi-threaded optimization, which uses a multi-threaded approach to achieve map optimization and bit pose estimation to improve the system's operation speed. ORB-SLAM3 provides a visual interface that displays information such as maps, camera trajectories, and detected feature points in real-time, making it easy for users to observe and debug. Overall, ORB-SLAM3 is an efficient, accurate, and robust visual SLAM system that has been widely used in robot navigation, augmented reality, autonomous driving, and other fields.

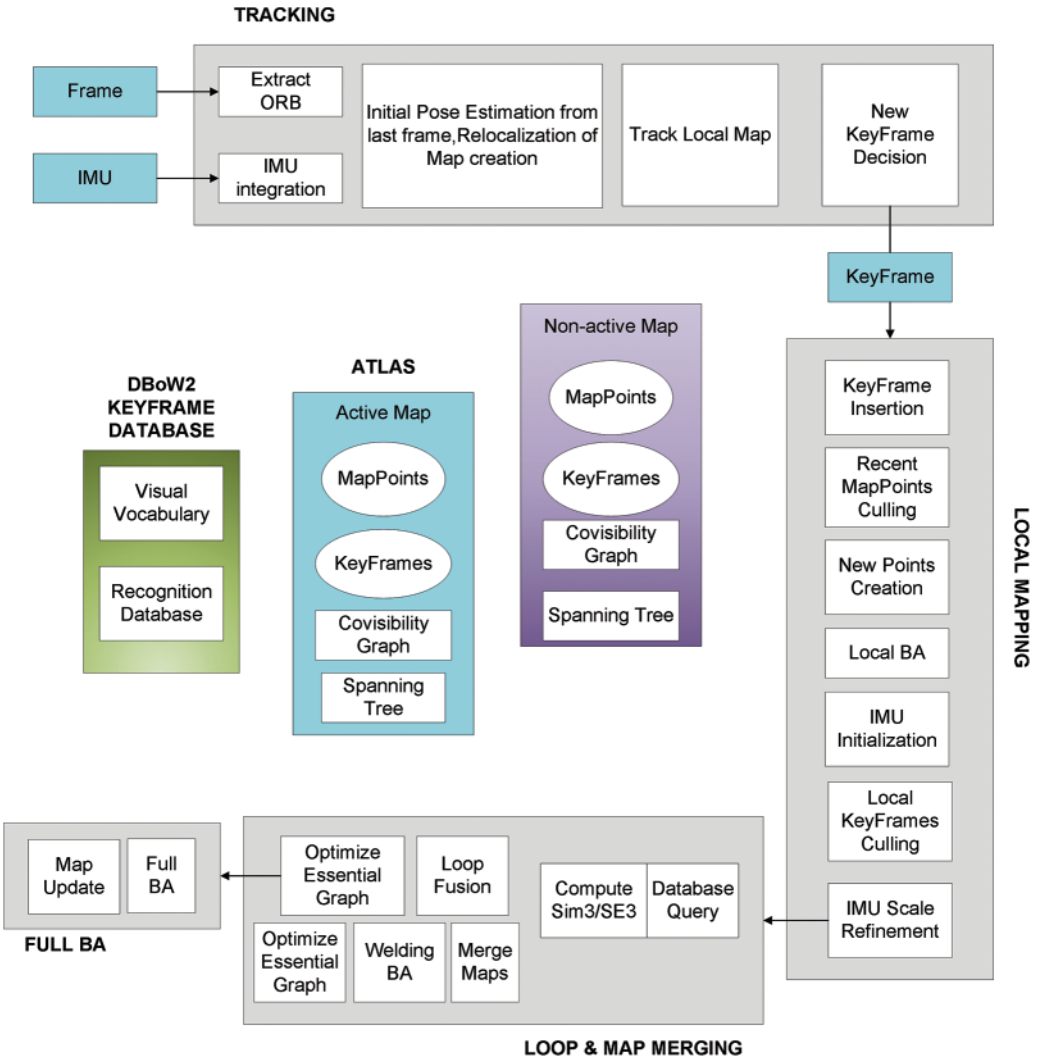


Figure 4: Main system components of ORB-SLAM3

2.2 Overview of Visual SLAM Based on Object Detection Networks

An object detection network is a model that implements modern target detection techniques, and its core task is to recognize targets in an image into specific categories while providing information about their location in the form of bounding boxes. Target detection networks are generally based on neural network techniques, specifically Convolutional Neural Networks (CNNs). CNNs are able to learn the features in an image by continuously training the dataset. The trained neural network can extract information from the image and identify and localize the targets. This paper summarizes two types of deep learning-based object detection networks, one is called a two-stage object detector, based on CNN as a feature extractor, which first determines the candidate region of the image and then detects the region, i.e., generating alternative frames in the image in advance, and then classifying the objects in the selected alternative frames, and also optimizing the position of the alternative frames, which mainly includes region proposal and detection. The classical algorithms include R-CNN, Fast R-CNN Faster R-CNN, etc. The other one is a single object detector, which directly predicts and classifies the candidate regions at each position of the image or video frame, without generating alternative recognition frames in advance. Including YOLO series [32–34], SSD [35], PP-DicoDet [36], etc. The recent YOLOv7 [37] improves the network architecture based on YOLOv5 with more advanced methods for both loss functions and label assignment mechanisms, which surpasses all previous detectors in terms of speed and accuracy. For the real-time requirements in specific scenarios, many researchers have proposed lightweight target detection networks.

As Fig. 5 shows the development history of object detection, the second stage object detection network was more popular mainly in 2014–2015. Since YOLOv1 was proposed in 2015, people recognized the speed advantage of the first-stage object detector, especially the YOLOv3 model with superior performance soon received wide attention, and the YOLO series and SSD series developed rapidly. Meanwhile, in order to meet the increasingly high-speed requirements, from 2017 onwards, researchers have proposed object detection based on improvements, replacing the backbone network with some lightweight networks, or using more lightweight convolutional modules, resulting in many lightweight object detection networks, and as of now lightweight object detection networks are still a research hotspot.

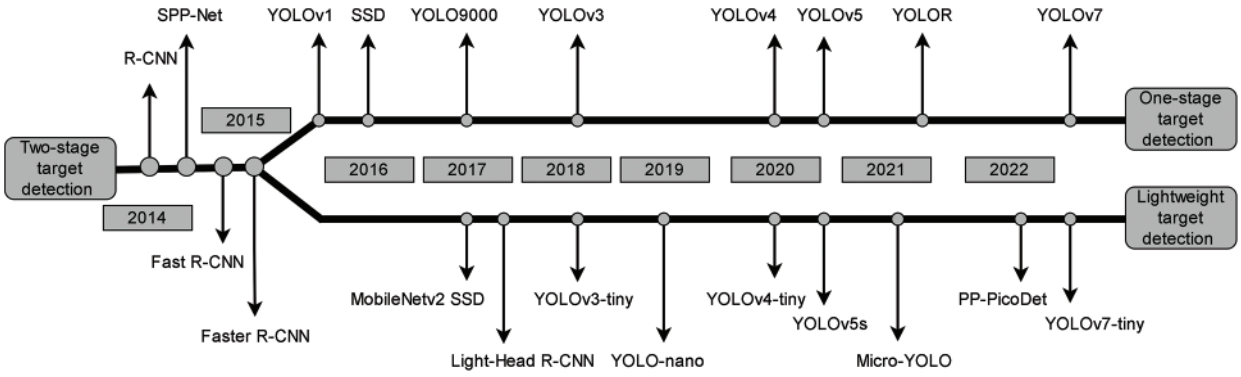


Figure 5: History of object detection network development

It is well known that traditional visual SLAM is less robust in dynamic and complex environments, so visual SLAM based on target detection network aims to improve the robustness and accuracy of visual SLAM with the help of target detection technology, as shown in Fig. 6, which is a rough block diagram of visual SLAM based on target detection network. The visual SLAM based on the

target detection network has at least one target detection network, which can be a 2D target detection network or a 3D target detection network, which is used to obtain semantic information such as bounding boxes and categories, and the semantic information can play a great role in tasks such as dynamic point culling, data correlation, and point cloud segmentation of the SLAM system, so as to improve the system accuracy.

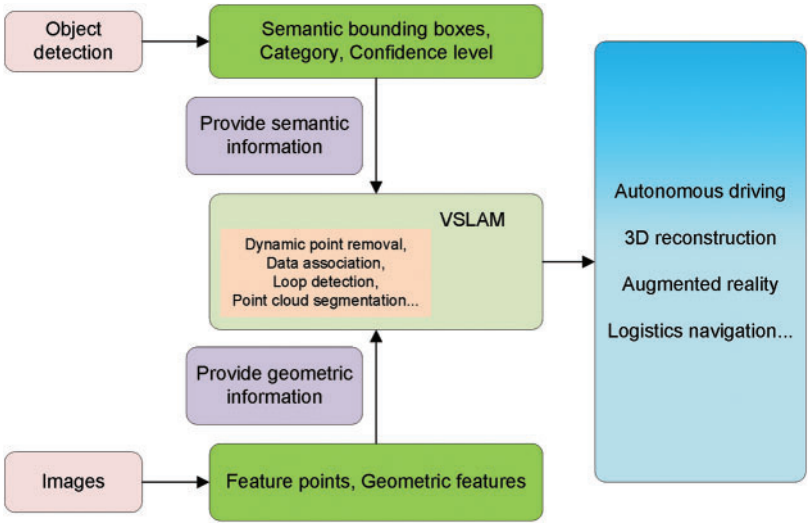


Figure 6: Visual SLAM framework based on object detection network

3 Visual SLAM Algorithm Based on Object Detection

In recent years, computer vision and deep learning have been combined with each other, leading to a huge improvement in the actual performance of vision-related tasks in terms of accuracy, execution efficiency, and robustness. Visual SLAM systems are based on computer vision, which provides a lot of room for the application of neural networks in this field. Object detection network, as one of the representatives of deep learning, can provide semantic information for traditional visual SLAM, and the semantic information can be flexibly applied to various parts of the SLAM system, its combination with visual SLAM has the following advantages: The visual SLAM system based on target detection has strong generalization ability and can work in complex environments; it is more effective in recognizing and processing dynamic targets. A data-driven approach is used to train the model, which is more in line with human-environment interaction; provides more semantic features, and improves the accuracy of closed-loop detection. The literature sources for this article include databases such as Web of Science, CNKI, Weipu, IEEE Electronic Library, and B Springer, including conference papers, journal papers, and degree papers. Firstly, a preliminary screening was conducted by searching for keywords such as Visual SLAM, Semantic SLAM, YOLO, Object detection, and autonomous driving in various databases, and obtained nearly 5907 articles. At this point, some articles have low relevance to our research, so it is necessary to quickly browse the titles and abstracts of the literature again to determine whether they are relevant to our research topic, and to exclude duplicate articles. After re-screening, there are still 965 articles left. Then, 304 articles were selected from several aspects such as language, publication year, and relevance to our research. Finally, a full-text reading of 304 literature was conducted, and 98 pieces of literature were selected for analysis from the perspectives of data integrity and algorithm innovation, including 4 review papers and 94 research papers, of which

7 were data papers introducing important datasets. During the data extraction of literature, the main information extracted was the author's name, publication year, core work, algorithm advantages and disadvantages, and applicable scenarios. Subsequently, the data was organized.

3.1 Visual Odometer Based on Object Detection

Visual odometry, also known as inter-frame motion estimation, is the process of determining the robot's pose and orientation by analyzing the multi-view geometric relationships between the associated camera images and can be used as a front-end to visual SLAM. Traditional inter-frame estimation methods based on sparse features or dense features require feature matching and complex geometric operations. Often, in practical applications, especially in outdoor dynamic environments, there are many dynamic objects, and dynamic features must be filtered out to build a complete environment map. The method based on object detection to remove dynamic objects has the characteristics of good real-time performance and high accuracy, which is also popular among researchers at present. Based on object detection to detect dynamic objects and remove outliers, this step is very important for the whole SLAM system and affects the performance of the whole system.

In dynamic scenes, some moving objects tend to interfere with object class judgment, i.e., to effectively filter out dynamic feature points while keeping the static environment as intact as possible, which is difficult to achieve in traditional SLAM. Many researchers have done a lot of experiments combining object detection networks. Detect-SLAM [38] is one of the first visual SLAM systems based on object detection networks. To eliminate the influence of dynamic objects in SLAM, the semantic bounding boxes of potential dynamic objects are obtained using SSD object detection networks, and then the features of potential object regions such as people, dogs, cats, and cars are excluded. Wang et al. [39] also combined the SSD network with ORB-SLAM2 to detect potential dynamic objects, and then calculated the basis matrix to obtain the polar line equation, according to the geometric constraint, the distance between the feature point and the polar line is greater than the threshold then it is a dynamic point and needs to be eliminated. The above SSD network is slow in detecting objects and is not suitable for application in scenarios with high real-time requirements. In contrast, the YOLO object detection network is quickly gaining attention due to its fast speed and high accuracy. Reference [40] directly used YOLOv3 to detect all semantic bounding boxes containing people and then rejects the feature points inside the boxes. Li et al. [41] used YOLOv3 to detect dynamic objects, using a priori information to reject feature points in the frame and set strict rules for keyframe selection, those with missing detection cannot be used as keyframes, and the number of feature points in keyframes must be within a certain range, this method avoids excessive computation caused by keyframe redundancy. This method of directly eliminating all feature points within the bounding box is shown in Fig. 7, and it has many drawbacks. The object detection network can only get the object category, and can not determine the dynamic objects, for example, sitting people and parked cars on the roadside are eliminated, this reduces the number of static feature points, especially when there are more objects of this category in the field of view, which greatly affects the localization accuracy. Another disadvantage is that all feature points inside the frame will be removed, then the interior points inside the frame but not on the dynamic object will also be eliminated as exterior points, which likewise reduces the number of interior points and affects the accuracy of the positional calculation.

In view of these drawbacks, some researchers have proposed to combine YOLO object detection network with optical flow or geometric methods. Chen et al. [42] proposed a method to reject dynamic points using a combination of optical flow and object detection, adding the process of rejecting dynamic feature points based on ORB-SLAM [43]. First, dynamic objects in key frames are detected using YOLOv4 network, and then, dynamic feature points in the scene are further identified and

rejected based on optical flow detection. Finally, the camera is tracked using static feature points to achieve highly robust monocular visual SLAM. To improve the speed of visual SLAM, researchers have continuously improved the YOLO object detection network, and Wu et al. [44] have improved the YOLOv5s network by replacing the backbone network of YOLOv5s with a lightweight network MobileNetV3, greatly improving the inference speed of object detection. After obtaining the semantic boundary box through YOLOv5s, the LK optical flow [45] values of all feature points within the semantic boundary box are calculated. If the optical flow value of the feature point is greater than the threshold, it is considered that the feature point is a dynamic point that needs to be removed. This method is fast and can run in real-time on the CPU, which improves the positioning accuracy by 80.16% compared to ORB-SLAM3. Others have used a combination of object detection network and geometric consistency for better detection of dynamic points, and in 2022, Ye et al. [46] added object detection threads to the VO of ORB-SLAM2, and in the paper, the YOLOv5s network was improved by using the ShuffleNetv2 network to replace the original backbone network, named YOLOv5s-L. The visual odometry improved by YOLOv5s-L is shown in Fig. 8, which first detects potential dynamic objects and then removes dynamic points accurately using limiting geometric constraints. However, both optical flow and geometric consistency are subject to errors. The effectiveness of the optical flow or geometric method is affected by the quality and characteristics of the input data, which may affect the accuracy of the optical flow or geometric method if the input data has problems such as noise, motion blur, or geometric distortion. Feature points on dynamic objects may also be judged as static points, so some people want to obtain the contours of dynamic objects and then remove them as a whole. Li et al. [47] proposed a SLAM algorithm based on deep learning and edge detection, which uses the Canny operator to calculate the contour edges of potential dynamic objects, and then calculates the optical flow of feature points inside the contour, sets a threshold value, and if the object contains dynamic feature points larger than the threshold value, it is designated as a dynamic object, and this method can remove the whole dynamic object while retaining the object contour beyond. This method can remove the whole dynamic object and keep the static points outside the object outline. However, it is difficult to extract the edges of the object with low accuracy, and it is difficult to extract the object outline completely in a complex environment, and there are errors. Therefore, it is more common to extract object contours from depth images.

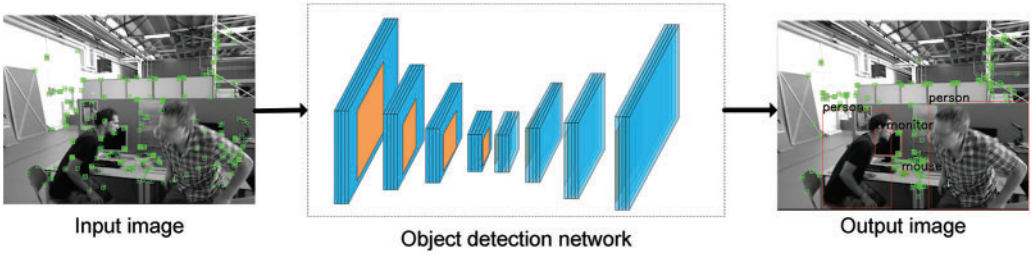


Figure 7: Eliminate dynamic points based on bounding box

Combining depth information with object detection to determine foreground objects is a more applied method, for example, Rong et al. [48] used the YOLOv4 model to detect object objects from RGB images and K-means to compute object masks from depth maps, and finally combined with multi-view geometric methods to identify moving objects in dynamic environments and segment moving foregrounds. However, the K-means algorithm alone has limited effectiveness in segmenting images. Therefore, Fang et al. [49] first obtained the object bounding box by YOLOV3 and then used K-means clustering within the corresponding depth image bounding box, and they determined the

score of each cluster by the average depth value and the number of pixels, and since the object should be used as the foreground and occupy most of the pixels in the bounding box compared with the background elements, the cluster with smaller depth value and more pixels was taken as the object segmentation result. Finally, the feature points in the segmented motion foreground are removed. The improved K-means algorithm segmented the foreground significantly better. Of course not only the k-means algorithm can achieve foreground segmentation of depth images, Hu et al. [50] proposed CFP-SLAM, the framework of CFP-SLAM is shown in Fig. 9, which uses DBSCAN clustering instead of the traditional K-means algorithm, DBSCAN clustering [51] can cluster data of arbitrary shape and can find anomalies while clustering, compared to the human pixels in-depth images clustering works better and the obtained contours are more in line with expectations. It can be seen that foreground segmentation is the focus of using depth information, and most of them use the clustering algorithm or GrabCut algorithm. The k-means algorithm does not have high segmentation accuracy, GrabCub algorithm is time-consuming, these algorithms still have a lot of room for improvement, and in practical applications, researchers will make improvements according to different scenarios for foreground segmentation.

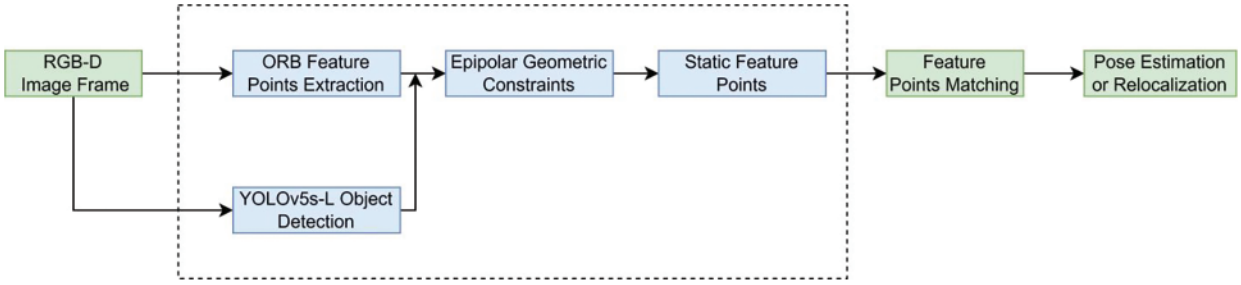


Figure 8: Visual odometer improved by YOLOv5s-L

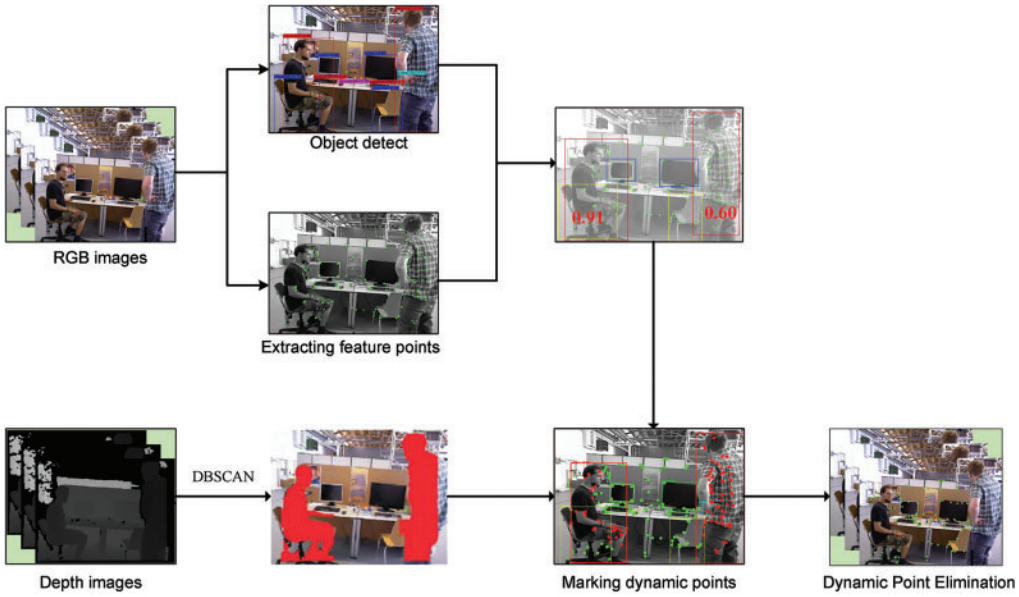


Figure 9: Dynamic point rejection method for CFP-SLAM. The image is from X. Hu's paper "CFP-SLAM: A real-time visual SLAM based on coarse-to-fine probability in dynamic environments" [50]

Also, some scholars have recently started to consider combining object detection networks with semantic segmentation networks, because semantic segmentation based on deep learning has better segmentation results than clustering. In 2021, Zheng et al. [52] proposed to add a dynamic object detection thread on ORB-SLAM2, and the steps of dynamic object detection are as follows: first, the object detection is performed on the RGB map to get the a priori information, and then the semantic segmentation is performed to segment the object, and the object is judged as a dynamic object directly based on the a priori information, and this method does not use optical flow method or geometric consistency test, so it has certain For example if the prior information of a chair is static, it will still be calculated as a static object when it is artificially moved. To avoid this situation, Xu et al. [53] proposed RDTS-SLAM, as shown in Fig. 10, which improves YOLOv5 by adding a segmentation head to the original decoder so that it has both object detection and semantic segmentation functions, and after object detection and semantic segmentation of RGB images, local optical flow is calculated for the points within the anchor frame of object detection to determine the object in the anchor frame is determined whether the object is moving or not. Table 1 lists four visual odometers that fuse object detection networks.

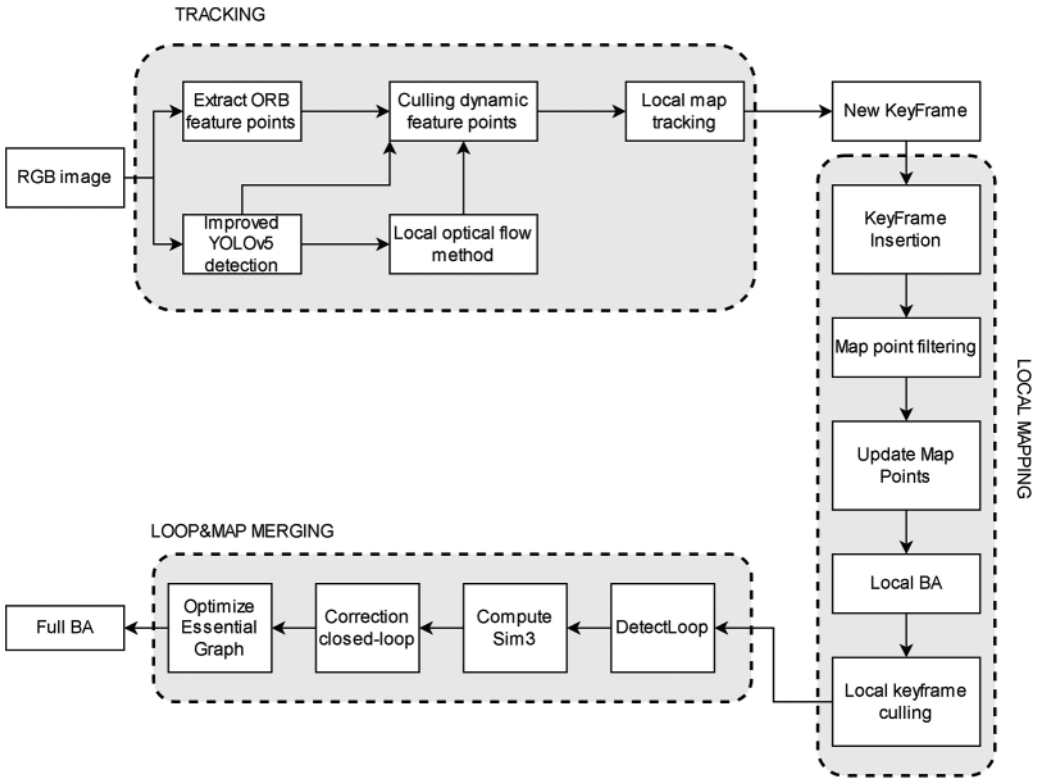


Figure 10: RDTS-SLAM system framework

Table 1: Summary of visual odometry methods based on object detection networks

	Year	Core work	Disadvantages
Detect-SLAM [38]	2018	Use the SSD network to get the semantic bounding box of potential dynamic objects, and reject the features of potential object regions such as people, dogs, cats, and cars	Slow SSD network operation
Text [41]	2019	Dynamic objects are detected using YOLOv3 and a proposed new keyframe selection strategy is presented.	Some of the inner points in the semantic framework have also been removed
Text [44]	2022	Replace YOLOv5s backbone network with a lightweight network MobileNetV3, removing feature points with optical flow values greater than a threshold	LK optical flow has errors, dynamic point rejection is incomplete
Text [49]	2021	Segmentation of motion prospects using YOLOv3 and k-means clustering	Poor splitting effect

3.2 Loop Closure Detection and Relocation Based on Object Detection

Loop closure detection is an important component of a visual SLAM system. Visual odometers generate accumulated errors during long-term positioning and navigation, and without loop closure detection, deviations will grow as the system operates, and loop closure detection is equivalent to playing a role in periodic correction. Efficient and accurate loop closure information can help SLAM systems suppress the effects of drift and eliminate the cumulative errors of long-term calculations. However, how to achieve correct loop closure detection is a major challenge at present. Traditional loopback detection uses BOW and some improved algorithms for probability, which are computationally intensive and have limited accuracy. After the rise of deep learning-based object detection networks, many researchers have combined semantic information of object detection to improve the accuracy of loop closure detection. This is also the future trend and direction of loop closure detection.

The core of loop closure detection is the matching of keyframes. Traditional loop closure detection calculates the matching score between the feature points of the current frame and the feature points of the previous frames, and the higher the score the more similar, and if the similarity exceeds the threshold, it is judged to be back to the original position. Since the manually defined feature points are easily affected by environmental changes and generate matching errors, fused object-level matching is more adaptable to various changing scenes. Zheng et al. [54] proposed a loop closure detection method incorporating semantic maps, as shown in Fig. 11, first semantic annotation of single-frame images by YOLOv3, then ORB feature points are extracted for keyframes, static scene segmentation is performed using semantic annotation information, and static ORB feature points are added to the bag-of-words model. This approach reduces the dynamic point information of loop closure detection and the score of similarity calculation is more reliable. The loop closure detection of Shi et al. [55] is divided into three steps. First is the object detection thread, in order to improve the detection speed

without affecting the accuracy, the training weights of YOLOv4 are model pruned, and in addition, in order to avoid overcomplete network training, a sparsity penalty function is added to the loss function, and then the number and class of objects obtained from the object detection thread are vectorized, so that many feature vectors are generated, and finally, the similarity is computed using a locally sensitive hash function to reduce the dimensionality of the high-dimensional vectors and calculate the cosine distance between each image frame to determine the loop, the algorithm has very good accuracy and real-time performance. Fang et al. [56] proposed a loop closure detection algorithm in May 2022, which constructs image semantic features using bounding boxes and compares them with historical frames to query similar keyframes in the loop closure detection stage, As shown in Fig. 12, which is fast and has less memory occupation compared with the visual bag-of-words method. In September of the same year, Soares et al. [57] and others classified objects based on YOLOv4 extended the Kalman filter [58] and short-term data association, and filtered static key points based on "DOC threshold" pairs. Static and change sequences are used to determine which objects need to be detected by loopback. Loopback detection based on object detection requires data association and matching, i.e., matching the same target in different frames. This may face challenges such as consistency problems of target IDs across frames, target occlusion, and deformation, which impose certain requirements on the design and implementation of the algorithm. Table 2 lists some of the loop closure detection based on object detection networks.

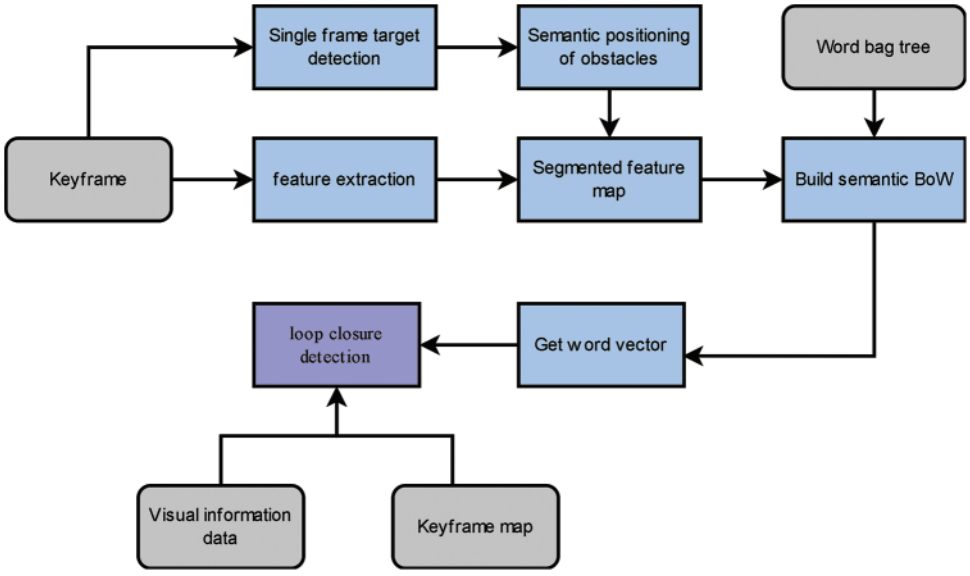


Figure 11: Semantic loop closure detection block diagram

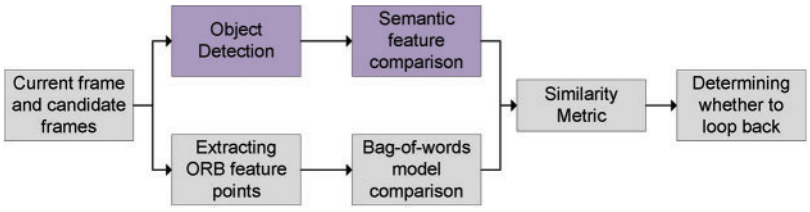


Figure 12: Loopback detection based on semantic bounding boxes

Table 2: Summary of loop closure detection methods based on object detection network

	Year	Core work	Disadvantages
Text [54]	2020	Improving the dictionary method based on semantic annotation information of keyframes, combined with the idea of motion feature point removal	Not very real-time
Text [55]	2021	The YOLO4 model optimized by loss function is used to perform object detection on the images acquired by the camera; then, the high-dimensional data is downscaled using a locally sensitive hash function, and the loop is determined based on the cosine distance	Poor loop closure effect in complex scenes
DyOD-SLAM [56]	2022	In the loop closure detection stage, image semantic features are constructed using bounding boxes and compared with historical frames to query similar keyframes, which is faster and less memory intensive than the visual bag-of-words method	Distinguish moving objects only based on semantic information, there is the case of missing dynamic feature points
Changing-SLAM [57]	2022	Filtering static key points based on "DOC threshold" pairs, using static and change sequences to determine which objects need to be loop closure detected	High calculation volume

Relocalization refers to the ability to reacquire the robot's positional position after tracking loss. Xiang et al. [59] proposed a robot relocalization enhancement method, combining SSD network and particle filtering algorithm, dividing the environment into multiple subregions, using SSD to obtain the semantic information of the environment, matching the semantic information with the subregions based on the great likelihood estimation to complete the coarse positioning of the robot, constraining the particle filtering to the subregion most likely to be located in the subregion, and then determine the specific location of the robot in that subregion, which can effectively re-localize the robot but runs slowly. Mahattansin et al. [60] addressed the drawback that the number of candidate frames to be matched is too many for traditional re-localization, add semantic information detected by YOLO to the bag-of-words model, use semantic features to quickly filter out similar candidate frames, and calculate the current bit pose by the feature points in the candidate frames and the basis matrix, which greatly improves the execution speed compared with the traditional relocation method, but it does not support object-level map reconstruction. In the recent OA-SLAM [61], a fully automated SLAM system is constructed that can automatically relocate and supports semantic map construction, which combines object and feature points to calculate the poses and can achieve positional recovery accurately in real-time.

With the development of deep learning, the speed and accuracy of object detection have been improved, and semantic features are more stable than feature points under strong illumination changes, making image matching no longer limited to matching using only traditional feature points, and both

loop closure detection and relocation need to match image frames, so fusion with object detection is an effective way to improve performance, both for loop closure detection and relocation.

3.3 Object Detection-Based Mapping

Maps are the basis for the robot to achieve navigation, and the robot should have the ability to understand its environment centered on itself; it needs to distinguish between rooms and corridors or record the state of different objects [62]. Traditional visual SLAM systems assume that the environment is static and extract only geometric feature information, and the settings of these features are artificially designed by experts, like ORB features and SURF features, which are difficult to represent in the map in dynamic scenes or in case of lighting changes. To overcome the drawbacks of traditional point cloud maps with much noise, some researchers have used target detection networks to build object-level semantic maps [63]. Galindo et al. [64] were the first to point out in their paper that incorporating semantic information in maps to help robots obtain information such as attributes and categories of objects has great scope for tasks such as navigation and obstacle avoidance.

Semantic maps can understand deep information in the environment [65], and usually, semantic maps are divided into scene-oriented building and object-oriented building, as shown in Fig. 13. Scene-oriented building refers to pixel-level building of 2D images, extracting semantic information of object detection and fusing with point clouds to build a full-scene map. Object-oriented building means that the map contains only some of the required objects and uses object detection to find the required objects, such as tables and walls while ignoring other irrelevant objects, which improves the building's efficiency. Reference [66] proposed a scene-oriented semantic map-building algorithm based on RTABMAP [67], using YOLOv2 for object detection, and then combining the Canny operator and region-growing algorithm to achieve accurate segmentation of objects. Finally, static semantic information is fused with 3D point clouds to construct a full-scene map. The method can build static backgrounds completely but slowly, while the object-oriented building algorithm has a natural advantage in terms of speed and can be well applied to scenarios with high real-time requirements. Reference [68] used integrated data association with a combination of nonparametric and parametric tests, fused bounding boxes and labels, and improved the association success rate using nonparametric tests for non-Gaussian distributed objects. The experimental results show that the constructed point cloud maps are clearer and the point locations are more accurate compared to ORB-SLAM2. Maolanon et al. [69] in order to enable the service robot to navigate well indoors, object-oriented map building is required, using a lightweight YOLOv3 network to classify and localize various furniture in the house, and dividing the map into multiple grids, keeping only those of interest containing furniture such as sofas, and then annotating semantic information on the map, which improves the speed of map building.

The scene-oriented building method based on object detection can eliminate dynamic objects and build a comprehensive static scene map, while object-oriented building builds only maps of objects of interest, such as tables, computers, sofas, etc. The object-oriented building method is able to accurately locate and track objects because it focuses on tracking and localizing objects. This can provide more reliable information in robot navigation and path planning. In contrast to the object-oriented building method, the scene-oriented building method can handle unknown objects or objects that have not been detected. It can build and update the structure and features of the scene, rather than relying solely on object detection results. Object-oriented and scene-oriented mapping each have some advantages and disadvantages. In practical applications, these factors can be weighed as needed to select the most appropriate building method.

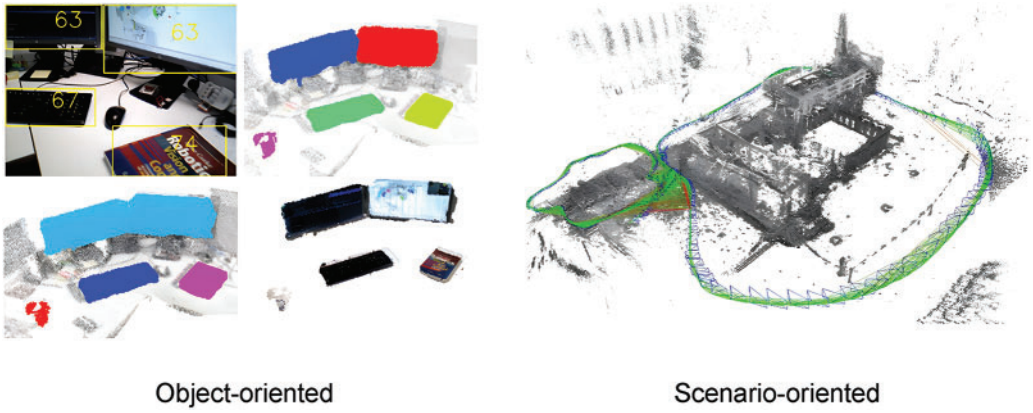
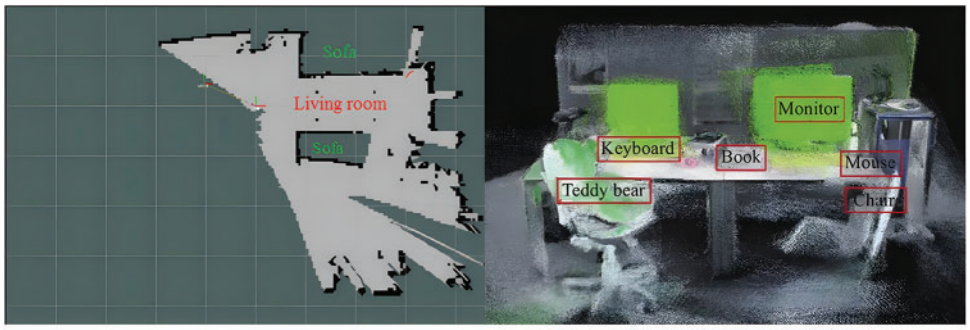


Figure 13: Object-oriented building (left) and scenario-oriented building (right). The left image is from Niko Süderhauf's paper "Meaningful maps with object-oriented semantic mapping" [70]. Right image from Jakob Engel's paper "LSD-SLAM: large-scale direct monocular SLAM" [71]

The maps are divided into two-dimensional and three-dimensional maps according to the number of dimensions, as shown in Fig. 14. The two-dimensional map is generally used to represent tangent information, commonly with a top view, and is also widely used in the navigation tasks of various service robots, especially for robots such as sweeping robots that only move in this plane on the ground. Three-dimensional maps, on the other hand, can be applied to navigation tasks that move in three dimensions and are more suitable for UAVs and industrial robots. Qiu et al. [72] constructed a semantic map in three steps, first using the gmapping [73] algorithm to construct a 2D raster map, then using YOLOv3 to obtain semantic information to get the location and category of the objects, and finally mapping the semantic information to the 2D raster map to form a 2D semantic map. Semantic labels can also be mapped to 3D maps, and Hu [74] constructed a 3D dense point cloud map, which did object regularization to get more accurate physical labels based on YOLOv3 detection, and fused it with the dense point cloud obtained by ORB-SLAM2 to get a point cloud semantic map. Ju et al. [75] added semantic information to the point cloud map of ORB-SLAM2 by first detecting the objects in the RGB map with YOLOv5, then completing the semantic labeling on the point cloud map, and finally using VCCS clustering to achieve point cloud segmentation, where different objects are represented by different colors and VCCS clustering increases the accuracy of segmentation and constructs an efficient 3D point cloud map. The above several articles fully demonstrate the role of object detection in SLAM map building, whether it is scene or object-oriented, and whether it is a 2D or 3D map, integrating semantic information of object detection can make the accuracy of the map improve.

The 2D raster map is fast and can quickly determine the object position by target detection, but the vertical information in the map is lost, and the height information and stereo sense cannot be well represented in the 2D raster map. Compared with a 2D raster map, a 3D point cloud map uses the detected target object information and adds their position and geometric features in a 3D point cloud to the established map, it can capture more detailed information and can represent the three-dimensional structure of the environment more accurately, which makes it more advantageous in dealing with the environment with complex geometric features and identifying objects. The computational complexity of a 3D point cloud is high, and the sparsification or quadtree methods are generally adopted to reduce the computational effort. Table 3 shows the comparison of several algorithms for building graphs based on target detection.



Raster diagram Point cloud map

Figure 14: 2D raster map (left) and 3D point cloud map (right). The left image is from P. Maolanon's paper "Indoor room identify and mapping with virtual-based SLAM using furniture and household objects relationship based on CNNs" [69]. The right image is from X. Hu's paper "Semantic SLAM based on visual SLAM and object detection" [50]

Table 3: Summary of object detection network-based map-building methods

	Year	Core work	Disadvantages
Text [68]	2020	It is based on ORB-SLAM2 and YOLOv3, combining bounding boxes and semantic labels to construct mappings.	Difficult to estimate accurately for large objects
Text [69]	2019	After learning the CNN detectors for furniture and objects, they are combined with SLAM algorithms via ROS. SLAM maps and simultaneous room detection can be automatically generated in unknown environments.	Less training data
Text [74]	2021	Generate dense point cloud labels with semantic information to obtain point cloud semantic maps.	Cannot run in real time
Text [75]	2021	Semantic point cloud segmentation is performed using VCCS algorithm to construct 3D point cloud maps.	The use of YOLOv5s detection effect is general

4 Datasets and Object Visual SLAM

4.1 Comparison of SLAM Datasets

Due to the need for high-precision measurement equipment to test the performance of SLAM systems in the real world, the operation is complex and the cost is high. Therefore, in order to detect the performance of visual SLAM systems, open-source datasets are usually used for evaluation. Choosing a suitable dataset is an important task that requires reasonable selection based on the applicable scenario of the system and the sensors used. This article lists 7 commonly used datasets for SLAM systems. References [76–82] provided a detailed introduction to each dataset, and Table 4 lists the main

features of each dataset. The sensor type is an important indicator for selecting a dataset. For depth cameras, the TUM dataset can be selected, and for binocular vision SLAM, KITTI, EuRoC, etc., can be selected. For certain visual SLAM systems that require IMU information fusion, EuRoc, and TUK Campus datasets can be selected. In addition, for different experimental scenarios, different indoor and outdoor lighting conditions and whether it is a dynamic scene are also important considerations. For datasets in outdoor environments such as streets and Railways, the scene is a dynamic scene with a large number of moving people and vehicles. TUM RGB-D has both dynamic and static sequences, and different experimental scenario data needs to be selected based on the experimental purpose. For SLAM systems that require testing of loop detection accuracy, New College and Nordland annotate loop information and annotate all loop frames, which can be used to separately test the accuracy of loop detection in the system. The KITTI and TUK Campus datasets are relatively friendly to visual SLAM based on object detection. These two datasets have label information for object detection. KITTI not only has 2D boxes labeled, but also annotates 3D boxes, which is also practical for performance detection of 3D object SLAM. Of course, there are still many open-source datasets that have not been listed here. In short, researchers need to choose datasets based on actual application scenarios, sensor types, and task requirements.

Table 4: Comparison of common SLAM datasets

Dataset	Year	Camera	IMU	LiDAR	Environment	Mark loop detection	Object detection label
New College [76]	2009	Stereo	—	√	Campus, park	√	—
KITTI [77]	2012	Stereo	—	√	Street, road	—	2D & 3D boxes 8 categories
TUM RGB-D [78]	2012	RGB-D	—	—	Indoors	—	—
Nordland [79]	2013	Mono	—	—	Railway, outdoors	√	—
EuRoC [80]	2016	Stereo	√	—	Indoors	—	—
Oxford RobotCar [81]	2017	Stereo fisheye	—	√	Street, highway	—	—
TUK Campus [82]	2021	Stereo	√	√	Campus	—	2D boxes 4 categories

4.2 Comparison of Visual SLAM Based on Object Detection Networks

Sixteen SLAM algorithms based on object detection are compared in Table 5, including their main methods and applicability to dynamic scenes. Most of the inputs to the visual SLAM system in the table are RGB-D cameras, while reference [83] utilized IMU to overcome the drawbacks of scale ambiguity and environmental sensitivity of monocular cameras. All SLAMs based on object detection networks use SSD or YOLO series networks. Some systems replace the backbone network with MobileNetV3 or directly use lightweight versions of models such as YOLOv5s and YOLOv3 tiny to enable real-time object detection, such as [44,69]. If the system needs to adapt to dynamic environments, it can be combined with the optical flow method to detect dynamic points and remove them, such as [44,83], or dynamic objects can be segmented using segmentation methods. Reference [49] used k-means to cluster the depth map and segment the parts belonging to humans on the depth map. Reference [84] directly used DeeplabV3+ to semantically segment the RGB image, which can accurately obtain the masks of

humans and cars, thereby removing dynamic objects. In references [68,69,74,75], these systems do not process dynamic targets in visual odometry, and they cannot adapt to dynamic scenes. They mainly use object detection to construct semantic maps, which provide important information for navigation. The dataset generally uses TUM and KITTI the most, and if experimental conditions are available, it can also be run in actual indoor and outdoor scenes. The real-time performance mainly depends on two aspects: the speed of image construction and the speed of object detection. Dense point cloud images and semantic point cloud images take a long time. If the system does not require high map accuracy, only semi-dense or sparse maps need to be constructed, such as [68,85,86]. Object detection often uses lightweight methods to improve detection speed.

Table 5: Comparison of visual SLAM based on object detection networks

Algorithm	Input	Method	Dynamic scene	Map	Dataset	Real-time
Detect-SLAM [38]	RGB-D	SSD + Grab−Cut	√	Object map	TUM RGB-D + real indoor	√
Text [41]	RGB-D	Yolov3	√	Point cloud maps	TUM RGB-D	√
Text [44]	RGB-D	MobileNetV3-YOLOv5s + Optical flow	√	Sparse map	TUM RGB-D	√
Text [49]	RGB-D	YOLOv3 + k-means	√	Semantic point cloud maps	TUM RGB-D	√
Text [54]	RGB-D	YOLOv3 + semantic Bag-of-words	√	Semantic point cloud maps	TUM RGB-D + real indoor	—
Text [55]	RGB-D	YOLOv4 + Locality Sensitive Hash function	—	Dense point cloud maps	TUM RGB-D + real indoor + New College	—
DyOD-SLAM [56]	RGB-D	YOLOv4 + Otsu	√	Sparse map	TUM RGB-D	√
Changing-SLAM [57]	RGB-D	YOLOv4 + Kalman Filter	√	Semantic point cloud maps	TUM RGB-D + PUC-USP	√
Text [68]	RGB-D	Yolov3	—	Semi-dense semantic point cloud maps	TUM RGB-D + real indoor	√
Text [69]	RGB-D +Laser	YOLOv3 tiny	—	Semantic raster map	Real indoor	√
Text [74]	RGB-D	YOLOv3	—	Semantic point cloud map	TUM RGB-D	√
Text [75]	RGB-D	YOLOv5s	—	Semantic point cloud map	TUM RGB-D	√
Mod-SLAM [83]	MONO +IMU	YOLOv2 + Optical flow	√	Sparse map	Self-record	√
Text [84]	RGB-D	YOLO v5x + DeeplabV3+	√	Semantic point cloud map	KITTI	—

(Continued)

Table 5 (continued)

Algorithm	Input	Method	Dynamic scene	Map	Dataset	Real-time
Text [85]	Stereo	YOLOv5s + Optical flow	√	Sparse map	KITTI	√
Text [86]	Stereo	YOLOv5 + k-best assignment enumeration	√	Sparse map	KITTI	√

5 Future Outlook

5.1 Higher Performance Lightweight Object Detection Network

The use of lighter and more accurate backbone networks is one of the trends in lightweight object detection networks. Backbone networks are the basic feature extractors for object detection tasks and largely affect the speed and accuracy of object detection. Therefore, many people have researched various efficient backbone networks, and the mainstream ones are currently ShuffleNetV2 [87], MobileNetV2 [88], MobileNetV3 [89], and GhostNet [90], these lightweight networks make it possible for object detection to run in real-time on the CPU. Qian et al. [91] then used the MobileNetV2 lightweight network instead of the traditional VGG network for feature extraction, which greatly improves the detection speed of the SSD object detection network. Qiu et al. [92] improved YOLOv5 by replacing the original residual block with the GhostBottleneck module, reducing the number of parameters. It can be seen that in order to meet the real-time requirements, more and more researchers choose to use lightweight feature extraction networks to replace the original deep networks. In addition, in order to enhance the small object detection capability, Li et al. [93] added a channel attention mechanism to YOLOv5 and changed the loss function to the CIoU loss function. The attention mechanism increases the local feature extraction capability of YOLOv5, so it has a better detection effect on small objects. In practical situations, the object detection network can enhance the feature extraction effect of edge or small objects by incorporating the attention mechanism. Nowadays, more and more efficient networks are proposed, and high accuracy and efficiency will be the continued development of object detection networks direction.

5.2 Data Association Based on Object Detection

Accurate data association is still a key concern for visual SLAM systems, and traditional visual SLAM systems use only geometric feature association, which is poorly robust and cannot distinguish similar features well. Wang et al. [94] used ORBSLAM2 as the basis and added semantic features using YOLOv3 in order to obtain better data association results, and the process of fusing ORB features and semantic features is to use semantic bounding boxes as constraints for ORB feature matching, which eliminates the mismatching of similar features between different categories and improves the data association accuracy. However, this method is difficult to judge when there are a large number of objects of the same category, and it also increases the error in the case of occlusion. The current semantic data association all face similar problems, therefore, it is necessary to study in depth how to obtain more semantic information in complex environments and use potential semantic information to improve data association.

5.3 Autonomous Robot Navigation

The application of semantic information in the direction of autonomous navigation is also currently a popular one. Visual simultaneous localization and map building have intrinsic limitations that come from a purely environmental understanding based on the geometric features of images. However, semantic SLAM features high-level environment perception, thus opening a new door, and thus a new development ecology for autonomous robot navigation. Chen et al. expressed their vision of the future of autonomous navigation robots [95] based on the semantic ORB-SLAM2 algorithm for autonomous navigation of mobile robots and their extensive work to apply semantic SLAM to autonomous navigation with good results. Li [96] used YOLOv4-tiny to color-code different objects in the point cloud map to build a clear semantic map, which helped the UAV to plan the path better and enhanced the obstacle avoidance effect of the UAV in autonomous navigation. This shows that object SLAM will definitely play an important role in the field of autonomous robot navigation in the future.

5.4 3D Object SLAM

Many researchers found that 2D target detection is difficult to solve the problem of repeated target occlusion and to obtain a more robust visual SLAM system, more and more researchers are trying to increase the system's perception of 3D objects in the environment using 3D target detection. 3D object SLAM requires the integration of information from individual vertices or edges of a 3D object into the map to ensure that the object's position and appearance in the map is accurate, and 3D semantic information presents better generalization performance in 3D map building and data association. Yang et al. [97] associated feature points with objects in different viewpoints up with optical flow tracking for common feature points, while triangulating 3D objects to improve the accuracy of bit-pose calculation for outdoor SLAM. smSLAM+LCD [98] added 3D object detection to loopback detection, obtained 3D models of objects by improved YOLOv3, and compared edges and vertices of 3D semantic information during loopback detection, with similar candidate frames for better differentiation effect. Many experiments show that relying on 3D road signs to establish VSLAM with 3D object constraints can yield higher odometry accuracy and can build more detailed semantic maps of the environment. However, the current accuracy of 3D object detection is limited and requires high computational performance of the machine, so the algorithm needs to be continuously optimized.

6 Conclusion

This article covers the fundamentals, methods, and challenges of VSLAM using object detection. It discusses the current development status and difficulties faced, outlines the VSLAM framework, and analyzes the pros and cons of each algorithm for visual odometer, loop detection, positioning, and map construction. Besides, various visual SLAMs based on object detection networks are compared with their characteristics analyzed. Finally, based on the current research status, I draw the following conclusion:

1) The running speed of visual SLAM systems based on object detection networks is greatly affected by the speed of object detection. Real-time goals can be achieved by replacing the backbone network of the object detection network with a lighter network, such as MobilNets and GhostNet.

2) Combining object detection with optical flow or geometric consistency methods is beneficial to remove dynamic feature points. Additionally, a segmentation method can effectively remove all

feature points on an entire object. This approach allows for obtaining a mask of the object within the bounding box and subsequently removing all feature points within that mask. Considering the specific circumstances, the combination can effectively accomplish the desired task.

3) To obtain better data association results, there are several methods: the target detection network assigns a unique ID to each target, which can be used to match targets between different time steps; Using a Kalman filter to infer the state of the target between different time steps and perform data correlation; Combining depth image information for target matching.

4) When performing loop detection in complex scenes, calculating the cosine distance of semantic bounding boxes for different image frames as part of similarity calculation can greatly improve the robustness of loop detection.

5) In the mapping task, to obtain an accurate and complete static background map, after detecting the object boundary box, combining parameter testing or clustering methods to segment point clouds can improve the accuracy of the map.

The main difficulty of the current object detection SLAM development is that the limited computational resources cannot meet the increasing computational resource needs of the algorithm, which will lead to the constraint of the SLAM system real-time, and deep learning requires high hardware, so the computational cost increases, which is a major reason why the object detection SLAM has not been widely used in the industry yet. In addition, the accuracy and robustness of the semantic SLAM system depend on the accuracy of environmental semantic information extraction, and with the continuous optimization of the target detection network and further expansion of the data set, the level of semantic information extraction will gradually rise, which will further promote the integration of target detection and visual SLAM.

Acknowledgement: All authors sincerely thank all organizations and institutions that have provided data and resources. Thank you to Hechi University for providing research equipment. Thank you to Mr. Guo Jinsong for helping to check the grammar and format of this article. Thank you to Mr. Xu Hengming for providing the research ideas. Thank you to all colleagues in our laboratory, whose professional insights and suggestions have had a profound impact on our research work.

Funding Statement: The authors are highly thankful to the National Natural Science Foundation of China (No. 62063006), to the Natural Science Foundation of Guangxi Province (No. 2023GXNSFAA026025), to the Innovation Fund of Chinese Universities Industry-University-Research (ID: 2021RYC06005), to the Research Project for Young and Middle-aged Teachers in Guangxi Universities (ID: 2020KY15013), and to the Special Research Project of Hechi University (ID: 2021GCC028). This research was financially supported by the Project of Outstanding Thousand Young Teachers' Training in Higher Education Institutions of Guangxi, Guangxi Colleges and Universities Key Laboratory of AI and Information Processing (Hechi University), Education Department of Guangxi Zhuang Autonomous Region.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: J. Peng, D. Chen; data collection: Q. Yang; analysis and interpretation of results: D. Chen, C. Yang; draft manuscript preparation: Y. Xu, Y. Qin. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] H. C. Chi, T. H. Tsai and S. Y. Chen, "SLAM-based augmented reality system in interactive exhibition," in *2020 IEEE Conf. on IOT, Communication and Engineering*, Yunlin, Taiwan, pp. 258–262, 2020.

[2] J. C. Piao and S. D. Kim, "Real-time visual–Inertial SLAM based on adaptive keyframe selection for mobile AR applications," *IEEE Transactions on Multimedia*, vol. 21, no. 11, pp. 2827–2836, 2019.

[3] P. Li, T. Qin and S. J. Shen, "Stereo vision-based semantic 3D object and ego-motion tracking for autonomous driving," in *Proc. of ECCV*, Munich, Bavaria, Germany, pp. 646–661, 2018.

[4] Y. Liu, J. Shang and L. Yu, "An intelligent logistics navigation device based on SLAM dual radar," in *2021 IEEE Conf. on Data Science and Computer Application (ICDSCA)*, Dalian, Liaoning, China, pp. 830–834, 2021.

[5] Ó. G. Grasa, E. Bernal, S. Casado, I. Gil and J. M. M. Montiel, "Visual SLAM for handheld monocular endoscope," *IEEE Transactions on Medical Imaging*, vol. 33, no. 1, pp. 135–146, 2014.

[6] M. Turan, Y. Almalioglu, H. Gilbert, H. Araujo, E. Konukoglu *et al.,* "Magnetic-visual sensor fusion based medical SLAM for endoscopic capsule robot," 2017. [Online]. Available: https://arxiv.org/abs/1705.06196 (accessed on 05/04/2023).

[7] Y. Liu, M. Xu, G. Jiang, X. Tong, J. Yun *et al.,* "Target localization in local densemapping using RGBD SLAM and object detection," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 4, pp. 1–11, 2022.

[8] N. S. Pai, W. Z. Huang, P. Y. Chen and S. A. Chen, "Optimization and path planning of simultaneous localization and mapping construction based on binocular stereo vision," *Sensors and Materials*, vol. 34, no. 3, pp. 1091–1104, 2022.

[9] G. Nützi, S. Weiss, D. Scaramuzza and R. Siegwart, "Fusion of IMU and vision for absolute scale estimation in monocular SLAM," *Journal of Intelligent and Robotic Systems*, vol. 61, no. 1–4, pp. 287–299, 2011.

[10] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conf. on Computer Vision and Pattern Recognition*, Columbus, OH, USA, pp. 580–587, 2014.

[11] R. Girshick, "Fast R-CNN," in *Proc. of IEEE/ICCV*, Santiago, De, Chile, pp. 1440–1448, 2015.

[12] S. Q. Ren, K. M. He, R. Girshick and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1137–1149, 2017.

[13] O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. of MICCA*, Munich, Bavaria, Germany, pp. 234–241, 2015.

[14] A. Kendal, V. Badrinarayanan and R. Cipolla, "Bayesian SegNet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," 2015. [Online]. Available: https://arxiv.org/abs/1511.02680 (accessed on 05/04/2023).

[15] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," in *Proc. of IEEE/ICCV*, Venice, Vinato Province, Italy, pp. 2980–2988, 2017.

[16] X. Zhang, Y. Peng, M. Yang, G. Cao and C. Wu, "Moving object detection for camera pose estimation in dynamic environments," in *Proc. of CYBER*, Xi'an, Shaanxi, China, pp. 93–98, 2020.

[17] A. Farhadi and J. Redmon, "YOLOv3: An incremental improvement," *Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 1804–2767, 2018.

[18] S. Han and Z. Xi, "Dynamic scene semantics SLAM based on semantic segmentation," *IEEE Access*, vol. 8, no. 1, pp. 43563–43570, 2020.

[19] J. Y. Ko, H. Wang and L. Xie, "Multi-modal semantic SLAM for complex dynamic environments," 2022. [Online]. Available: https://arxiv.org/abs/2205.04300 (accessed on 05/04/2023).

[20] R. J. Liu, S. Wang, C. Zhang and B. H. Zhang, "A survey on visual SLAM based on deep learning," *Journal of System Simulation*, vol. 32, no. 7, pp. 1244–1256, 2020.
[21] K. Chen, J. Zhang, J. Liu, Q. Tong, R. Liu *et al.,* "Semantic visual simultaneous localization and mapping: A survey," 2022. [Online]. Available: https://arxiv.org/abs/2209.06428 (accessed on 05/04/2023).
[22] L. Xia, J. Cui, R. Shen, X. Xu, Y. Gao *et al.,* "A survey of image semantics-based visual simultaneous localization and mapping: Application-oriented solutions to autonomous navigation of mobile robots," *International Journal of Advanced Robotic Systems*, vol. 17, no. 3, 2020.
[23] X. Q. Li, W. He, S. Q. Zhu, Y. H. Li and T. Xie, "Survey of simultaneous localization and mapping based on environmental semantic information," *Chinese Journal of Engineering*, vol. 43, no. 6, pp. 754–767, 2021.
[24] X. Wang and Y. F. Zuo, "Advances in visual SLAM research," *CAAI Transactions on Intelligent Systems*, vol. 15, no. 5, pp. 825–834, 2020.
[25] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.
[26] L. V. Stumberg and D. Cremers, "DM-VIO: Delayed marginalization visual-inertial odometry," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1408–1415, 2022.
[27] J. Engel, V. Koltun and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2017.
[28] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.
[29] T. Qin, P. Li and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
[30] Y. F. Cai, Z. H. Lu, Y. C. Li, L. Chen and H. Wang, "Tightly coupled SLAM system based on multi-sensor fusion," *Automotive Engineering*, vol. 44, no. 3, pp. 350–361, 2022.
[31] S. Urban, J. Leitloff and S. Hinz, "Mlpnp-a real-time maximum likelihood solution to the perspective-n-point problem," 2016. [Online]. Available: https://arxiv.org/abs/1607.08112 (accessed on 05/04/2023).
[32] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. of IEEE/CVPR*, Las Vegas, NV, USA, pp. 779–788, 2016.
[33] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. of IEEE/CVPR*, Honolulu, HI, USA, pp. 6517–6525, 2017.
[34] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng *et al.,* "YOLOv6: A single-stage object detection framework for industrial applications," 2022. [Online]. Available: https://arxiv.org/abs/2209.02976 (accessed on 10/04/2023).
[35] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed *et al.,* "SSD: Single shot multibox detector," in *Proc. of ECCV*, Amsterdam, North Netherlands Province, Netherlands, pp. 21–37, 2016.
[36] H. Li, S. Zhang, X. Li, L. Su, H. Huang *et al.,* "DetectorNet: Transformer-enhanced spatial temporal graph neural network for traffic prediction," in *Proc. ICAGIS*, Xi'an, Shaanxi, China, pp. 133–136, 2021.
[37] C. Wang, A. Bochkovskiy and H. Liao, "YOLOv7: Trainable bag-offreebies sets new state-of-the-art for real-time object detectors," 2022. [Online]. Available: https://arxiv.org/abs/2207.02696 (accessed on 10/04/2023).
[38] F. Zhong, S. Wang, Z. Zhang, C. Chen and Y. Wang, "Detect-SLAM: Making object detection and SLAM mutually beneficial," in *2018 IEEE Conf. on Applications of Computer Vision*, Lake Tahoe, NV, USA, pp. 1001–1010, 2018.
[39] E. Wang, Y. Zhou and Q. Zhang, "Improved visual odometry based on SSD algorithm in dynamic environment," in *Proc. of CCC*, Shenyang, Liaoning, China, pp. 7475–7480, 2020.
[40] B. Gökcen and E. Uslu, "Object aware RGBD SLAM in dynamic environments," in *Proc. of INISTA*, Biarritz, Pyrenees Atlantic, France, pp. 1–6, 2022.
[41] P. Li, G. Zhang, J. Zhou, R. Yao, X. Zhang *et al.,* "Study on SLAM algorithm based on object detection in dynamic scene," in *Proc. of ICAMechS*, Kusatsu, Shiga Prefecture, Japan, pp. 363–367, 2019.

[42] B. Chen, G. Peng, D. He, C. Zhou and B. Hu, "Visual SLAM based on dynamic object detection," in *Proc. of CCDC*, Kunming, Yunnan, China, pp. 5966–5971, 2021.
[43] R. Mur-Artal, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
[44] Z. J. Wu, H. Chen, Y. Peng and W. Song, "Visual SLAM with lightweight YOLOv5s in dynamic environment," *Computer Engineering*, vol. 48, no. 8, pp. 187–195, 2022.
[45] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.
[46] Q. Ye, C. Dong, X. Liu, L. Gao, K. Zhang *et al.,* "A visual odometry algorithm in dynamic scenes based on object detection," in *Proc. of PRAI*, Chengdu, Sichuan, China, pp. 465–470, 2022.
[47] L. Li and C. Cai, "Robust SLAM in dynamic scenes based on deep learning and edge detection," *Journal of Sensing and Actuators*, vol. 34, no. 1, pp. 80–88, 2021.
[48] H. Rong, A. Ramirez-Serrano, L. Guan and X. Cong, "Robust RGB-D SLAM for dynamic environments based on YOLOv4,"in *2020 IEEE Conf. on VTC2020-Fall*, Victoria, BC, Canada, pp. 1–6, 2020.
[49] B. F. Fang, X. Zhang and H. Wang, "Pixel-level segmentation algorithm combining depth map clustering and object detection," *Pattern Recognition and Artificial Intelligence*, vol. 35, no. 2, pp. 130–140, 2022.
[50] X. Hu, Y. Zhang, Z. Cao, R. Ma, Y. Wu *et al.,* "CFP-SLAM: A real-time visual SLAM based on coarse-to-fine probability in dynamic environments," in *2022 IEEE/RSJ Conf. on Intelligent Robots and Systems (IROS)*, Kyoto, Japan, pp. 4399–4406, 2022.
[51] E. Schubert, J. Sander, M. Ester, H. P. Kriegel and X. Xu, "DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017.
[52] S. Zheng, L. Kong, T. You and D. Yi, "Semantic SLAM algorithm based on deep learning in dynamic environment," *Journal of Computer Applications*, vol. 41, no. 10, pp. 2945–2951, 2021.
[53] C. Xu, J. Yan, H. Yang, B. Wang and H. Wu, "Visual SLAM algorithm based on target detection and semantic segmentation," *Computer Engineering*, vol. 49, no. 8, pp. 199–206, 2023.
[54] B. Zheng, Q. Liu, F. Zhao, X. Zhang and Q. Wang, "Loop detection and semantic mapping algorithm fusing with semantic information," *Journal of Chinese Inertial Technology*, vol. 28, no. 5, pp. 629–637, 2020.
[55] X. Shi and L. Li, "Loop closure detection for visual SLAM systems based on convolutional netural network," in *2021 IEEE Conf. on Computational Science and Engineering*, Shenyang, Liaoning, China, pp. 123–129, 2021.
[56] J. Fang and Z. H. Fang, "Visual SLAM optimization in based on object detection network," *Journal of Beijing University of Technology*, vol. 48, no. 5, pp. 466–475, 2022.
[57] J. C. V. Soares, V. S. Medeiros, G. F. Abati, M. Becker, G. Caurin *et al.,* "Visual localization and mapping in dynamic and changing environments," 2022. [Online]. Available: https://arxiv.org/abs/2209.10710 (accessed on 13/04/2023).
[58] L. H. Lin, J. C. Zheng, G. H. Huang and G. W. Cai, "Monocular visual inertial odometer based on convolutional neural network with extended Kalman filter," *Journal of Instrumentation*, vol. 42, no. 10, pp. 188–198, 2021.
[59] C. Xiang, L. Jiang, B. Lei and J. Y. Zhu, "Enhancement of mobile robot relocalization based on environmental semantic information," *Journal of Wuhan University of Science and Technology*, vol. 43, no. 3, pp. 224–229, 2020.
[60] N. Mahattansin, K. Sukvichai, P. Bunnun and T. Isshiki, "Improving relocalization in visual SLAM by using object detection," in *Proc. of ECTI-CON*, Banwuli, Prachuap Khiri Khan, Thailand, pp. 1–4, 2022.
[61] M. Zins, G. Simon and M. O. Berger, "OA-SLAM: Leveraging objects for camera relocalization in visual SLAM," in *2022 IEEE Conf. on ISMAR*, Singapore, pp. 720–728, 2022.
[62] Z. Liao, Y. Hu, J. Zhang, X. Qi, X. Zhang *et al.,* "SO-SLAM: Semantic object SLAM with scale proportional and symmetrical texture constraints," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4008–4015, 2022.

[63] R. Yin, Y. Cheng, H. Wu, Y. Song, B. Yu *et al.,* "FusionLane: Multi-sensor fusion for lane marking semantic segmentation using deep neural networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1543–1553, 2020.
[64] C. Galindo, A. Saffifiotti, S. Coradeschi, P. Buschka, J. A. Fernandez-Madrigal *et al.,* "Multi-hierarchical semantic maps for mobile robotics," in *2005 IEEE Conf. on Intelligent Robots and Systems*, Edmonton, AB, Canada, pp. 2278–2283, 2005.
[65] R. Ashour, M. Abdelkader, J. Dias, N. I. Almoosa and T. Taha, "Semantic hazard labelling and risk assessment mapping during robot exploration," *IEEE Access*, vol. 10, pp. 16337–16349, 2022.
[66] M. Mao, H. Zhang, S. Li and B. Zhang, "SEMANTIC-RTAB-MAP (SRM): A semantic SLAM system with CNNs on depth images," *Mathematical Foundations of Computing*, vol. 2, no. 1, pp. 29–41, 2019.
[67] M. Labbé and F. Michaud, "Appearance-based loop closure detection for online large-scale and long-term operation," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 734–745, 2013.
[68] Y. Wu, Y. Zhang, D. Zhu, Y. Feng, S. Coleman *et al.,* "EAO-SLAM: Monocular semi-dense object SLAM based on ensemble data association," in *Proc. of IROS*, Las Vegas, NV, USA, pp. 4966–4973, 2020.
[69] P. Maolanon, K. Sukvichai, N. Chayopitak and A. Takahashi, "Indoor room identify and mapping with virtual based SLAM using furnitures and household objects relationship based on CNNs," in *Proc. of IC-ICTES*, Bangkok, Thailand, pp. 1–6, 2019.
[70] N. Sünderhauf, T. T. Pham, Y. Latif, M. Milford and I. Reid, "Meaningful maps with object-oriented semantic mapping," in *2017 IEEE/RSJ Conf. on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, pp. 5079–5085, 2017.
[71] J. Engel, T. Schöps and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. of ECCV*, Zurich, Switzerland, pp. 834–849, 2014.
[72] H. Qiu, Z. Lin and J. Li, "Semantic map construction via multi-sensor fusion," in *Proc. of YAC*, Nanchang, Jiangxi, China, pp. 495–500, 2021.
[73] G. Grisetti, C. Stachniss and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.
[74] X. Y. Hu, "Semantic SLAM based on visual SLAM and object detection," *Journal of Applied Optics*, vol. 42, no. 1, pp. 57–64, 2021.
[75] Q. Ju, F. F. Liu, G. C. Li and X. N. Wang, "Semantic map generation algorithm combined with YOLOv5," in *Proc. of ICCEA*, Kunming, Yunnan, China, pp. 7–10, 2021.
[76] M. Smith, I. Baldwin, W. Churchill, R. Paul and P. Newman, "The new college vision and laser data set," *The International Journal of Robotics Research*, vol. 28, no. 5, pp. 595–599, 2009.
[77] A. Geiger, P. Lenz and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conf. on Computer Vision and Pattern Recognition*, Providence, USA, pp. 3354–3361, 2012.
[78] J. Sturm, N. Engelhard, F. Endres, W. Burgard and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal, pp. 573–580, 2012.
[79] N. Sünderhauf, P. Neubert and P. Protzel, "Are we there yet? Challenging SeqSLAM on a 3000 km journey across all four seasons," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Karlsruhe, Baden Württemberg, Germany, pp. 1–3, 2013.
[80] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder *et al.,* "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.
[81] W. Maddern, G. Pascoe, C. Linegar and P. Newman, "1 year, 1000 km: The Oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.
[82] H. E. Keen, Q. H. Jan and K. Berns, "Drive on pedestrian walk. tuk campus dataset," in *2021 IEEE Conf. on Intelligent Robots and Systems (IROS)*, Prague, Central Bohemian Region, Czech Republic, pp. 3822–3828, 2021.
[83] J. Hu, H. Fang, Q. Yang and W. Zha, "MOD-SLAM: Visual SLAM with moving object detection in dynamic environments," in *Proc. of CCC*, Shanghai, China, pp. 4302–4307, 2021.

[84] M. Li, Y. Jia, J. Zhu and W. Shen, "A dense dynamic SLAM system based on motion element filtering," in *Proc. of ICCC*, Chengdu, Sichuan, China, pp. 554–559, 2021.
[85] Q. Zang, K. Zhang, L. Wang and L. Wu, "An adaptive ORB-SLAM3 system for outdoor dynamic environments," *Sensors*, vol. 23, no. 3, pp. 1359, 2023.
[86] E. Michael, T. Summers, T. A. Wood, C. Manzie and I. Shames, "Probabilistic data association for semantic SLAM at scale," in *2022 IEEE Conf. on Intelligent Robots and Systems (IROS)*, Kyoto, Japan, pp. 4359–4364, 2022.
[87] N. Ma, X. Zhang, H. T. Zheng and J. Sun, "ShuffleNet V2: Practical guidelines for efficient cnn architecture design," in *Proc. of ECCV*, Munich, Bavaria, Germany, pp. 116–131, 2018.
[88] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. C. Chen, "MobileNetV2 Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 4510–4520, 2018.
[89] A. Howard, M. Sandler, G. Chu, L. C. Chen, B. Chen *et al.,* "Searching for MobileNetV3," in *2019 IEEE/CVF Conf. on Computer Vision (ICCV)*, Seoul, South Korea, pp. 1314–1324, 2019.
[90] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu *et al.,* "GhostNet: More features from cheap operations," in *2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, USA, pp. 1577–1586, 2020.
[91] W. Q. Qian and J. Wang, "Pedestrian object detection based on lightweight SSD algorithm," *Computer Simulation*, vol. 39, no. 9, pp. 487–491, 2022.
[92] T. H. Qiu, L. Wang, P. Wang and Y. E. Bai, "Research on object detection algorithm based on improved YOLOv5," *Computer Engineering and Applications*, vol. 58, no. 13, pp. 63–73, 2022.
[93] Y. Li, J. Zhang, Y. Hu, Y. Zhao and Y. Cao, "Real-time safety helmet-wearing detection based on improved YOLOv5," *Computer Systems Science and Engineering*, vol. 43, no. 3, pp. 1219–1230, 2022.
[94] Y. Wang and A. Zell, "Improving feature-based visual SLAM by semantics," in *2018 IEEE Conf. on Image Processing, Applications and Systems*, Sophia Antipolis, France, pp. 7–12, 2018.
[95] G. Chen, W. Chen, H. Yu and H. Wang, "Research on autonomous navigation method of mobile robot based on semantic ORB-SLAM2 algorithm," *Machine Tools and Hydraulics*, vol. 48, no. 9, pp. 16–20, 2020.
[96] Z. H. Li, "Design of UAV autonomous navigation system based on semantic VSLAM," China National Knowledge Infrastructure, 2022. [Online]. Available: https://kns.cnki.net/kcms2/article/abstract?v=2F6201taHdfnfj4OhP6V362lrQXTcSoBu1q89BFighGztndCWyODbAmAEvPoC8cdsnPwb9oKdvAImmrQs8Hwh2DHQ-GhGkE4Y8HBuYZIxurivBIwriJhiJ7_Dg9zb72UmE_lbTE71=&uniplatform=NZKPT&language=CHS (accessed on 13/04/2023).
[97] S. Yang and S. Scherer, "CubeSLAM: Monocular 3-D object SLAM," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 925–938, 2019.
[98] Z. Qian, J. Fu and J. Xiao, "Towards accurate loop closure detection in semantic SLAM with 3D semantic covisibility graphs," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2455–2462, 2022.