Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.041900

REVIEW

Dynamic SLAM Visual Odometry Based on Instance Segmentation: A Comprehensive Review

Jiansheng Peng[1,2,*], Qing Yang[1], Dunhua Chen[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: 28 November 2023 Published: 30 January 2024

ABSTRACT

Dynamic Simultaneous Localization and Mapping (SLAM) in visual scenes is currently a major research area in fields such as robot navigation and autonomous driving. However, in the face of complex real-world environments, current dynamic SLAM systems struggle to achieve precise localization and map construction. With the advancement of deep learning, there has been increasing interest in the development of deep learning-based dynamic SLAM visual odometry in recent years, and more researchers are turning to deep learning techniques to address the challenges of dynamic SLAM. Compared to dynamic SLAM systems based on deep learning methods such as object detection and semantic segmentation, dynamic SLAM systems based on instance segmentation can not only detect dynamic objects in the scene but also distinguish different instances of the same type of object, thereby reducing the impact of dynamic objects on the SLAM system's positioning. This article not only introduces traditional dynamic SLAM systems based on mathematical models but also provides a comprehensive analysis of existing instance segmentation algorithms and dynamic SLAM systems based on instance segmentation, comparing and summarizing their advantages and disadvantages. Through comparisons on datasets, it is found that instance segmentation-based methods have significant advantages in accuracy and robustness in dynamic environments. However, the real-time performance of instance segmentation algorithms hinders the widespread application of dynamic SLAM systems. In recent years, the rapid development of single-stage instance segmentation methods has brought hope for the widespread application of dynamic SLAM systems based on instance segmentation. Finally, possible future research directions and improvement measures are discussed for reference by relevant professionals.

KEYWORDS

Dynamic SLAM; instance segmentation; visual odometry

1 Introduction

With the continuous transformation of the manufacturing industry, the automation industry based on high-tech has gradually replaced the traditional low-end industry and has become the backbone of the manufacturing sector. The change in modern production and lifestyle has promoted the development of mobile robot technology [1,2]. At present, this technology is widely used in our production and daily lives. Since the 1980s, with the maturity of fundamental theoretical algorithms such as the Bayesian filter, Kalman filter, and particle filter, SLAM technology has developed rapidly. Early SLAM methods mainly used sonar sensors, while modern SLAM algorithms are mainly developed based on laser radar or vision technology. Compared with laser radar, the use cost of a camera is lower, and the weight, size, and energy consumption are more suitable for modern equipment. At the same time, because visual SLAM can display more environmental information than laser radar SLAM, and other characteristics, current researchers believe that research into visual SLAM has greater application value [3–5].

Since 2007, classical visual SLAM has matured, especially the feature-based ORB-SLAM [6–8], and the optical flow-based direct method, direct sparse odometry (DSO) [9], which have shown excellent performance in terms of localization and mapping accuracy. However, both feature-based and optical flow-based classical visual SLAM methods can only maintain robustness and high accuracy in static environments. Nevertheless, in real-life work scenarios, it is impossible for everything to be static, and the presence of moving objects such as pedestrians, vehicles, and animals may lead to the mis-matching of feature points, which affects the accuracy of SLAM localization and mapping, as shown in Fig. 1a. Therefore, how to solve the effect of dynamic objects on SLAM systems has become a popular research topic in recent years [10].

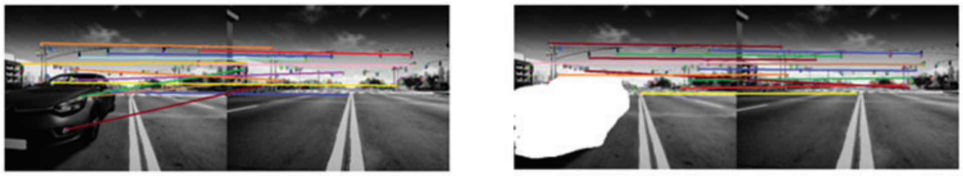

(a) (b)

Figure 1: Feature point mismatch and correct match

Currently, there are three main approaches to address the effect of dynamic objects on SLAM systems: fusion of multi-sensor information, motion estimation-based approaches, and pixel-level-based static and dynamic segmentation. Visual SLAM can enhance the robustness of the system by fusing multiple sensors, such as adding LiDAR to accurately detect and track dynamic objects in the scene and incorporating an inertial measurement unit (IMU) to provide high-frequency attitude estimation for compensating for the motion of dynamic objects. However, fusing multi-sensor approaches may lead to significant data redundancy, increasing the computational and storage overhead of the system. Additionally, the physical and electronic integration between sensors requires higher process costs and technical requirements. Motion estimation-based approaches use motion information of dynamic objects to distinguish them from static environments. Common approaches include those based on interframe differencing, optical flow estimation, and motion modeling, which model dynamic objects as additional constraints through change detection and motion estimation of sensor data to minimize their impact on the SLAM system. This method requires accurate motion modeling, which can be challenging for different types of dynamic objects. Noise, error, and uncertainty in motion estimation can also affect the estimation of the position and motion state of dynamic objects. The static and dynamic segmentation approach divides the camera data into static and dynamic parts, separating dynamic objects from the static background by segmenting the point cloud or image sequence at the pixel level. It then utilizes the static part for feature matching, as shown in Fig. 1b, to eliminate the influence of dynamic objects on the SLAM system's position estimation and map construction. Common methods for this approach mainly include segmentation algorithms based on appearance features and deep learning-based segmentation methods. The Segmentation-based approach combines visual tracking algorithms to make decisions about the target tracking state through a hierarchical decision-making tracking strategy, and the expert system updates and initializes the model online [11]. This not only greatly improves the positioning accuracy and robustness of the SLAM system but also enables real-time updates of semantic information, providing a higher level of environment understanding. This refined environment understanding helps to provide a more accurate and detailed environment description, offering more precise constraints and information for the localization and map construction of the SLAM system [12].

With the continuous progress of deep learning-based target detection, semantic segmentation, and instance segmentation methods in terms of efficiency and accuracy, more and more researchers have begun to apply these methods in SLAM visual odometry as a way to solve the problem of moving objects in dynamic scenes affecting the localization accuracy of SLAM systems [13]. Compared with target detection and semantic segmentation, instance segmentation not only realizes pixel-level segmentation but also distinguishes different individuals of the same type in a large number of dense scenes, achieving the complete separation of static and dynamic regions in the scene. Therefore, instance segmentation outperforms the methods based on target detection and semantic segmentation in dynamic SLAM. The dynamic SLAM technique based on instance segmentation can theoretically solve the influence of moving objects on the SLAM system well, but there are some related problems. The instance segmentation network can only segment a priori potential moving objects, but not non-a priori moving objects, and in practical application scenarios, there may be unknown moving objects that will lead to errors in the system localization and map construction. Compared with target detection and semantic segmentation, instance segmentation networks need to process the categories and instances to which pixels belong in the image in a more detailed way, so the reasoning process takes more time, which limits the real-time performance of dynamic SLAM systems based on instance segmentation. In addition, none of the current instance segmentation networks can handle overlapping and small-size target problems well, which also limits the application of SLAM systems to complex dynamic scenes.

The appealing factors lead to the fact that dynamic SLAM based on instance segmentation is not widely used in practical scenarios. This paper addresses these issues. Firstly, it introduces traditional SLAM methods based on mathematical models, providing researchers with a detailed understanding of SLAM systems. Then, it introduces the current instance segmentation models with excellent performance and analyzes and compares the performance of different models on the COCO dataset [14]. The aim is to enable researchers to understand the current status of instance segmentation research. Next, the paper introduces dynamic SLAM methods based on instance segmentation and analyzes and compares the performance of different methods on the TUM RGB-D and KITTI datasets [15,16]. Finally, it discusses the challenges and future development directions of dynamic SLAM based on instance segmentation, aiming to provide ideas for researchers to solve the current problems of dynamic SLAM based on instance segmentation, aiming to offer ideas for researchers to address the issues encountered by instance segmentation-based dynamic SLAM. To the best of our knowledge, there is no review on dynamic SLAM based on instance segmentation, and most of the existing reviews cover more extensive SLAM topics. Therefore, we believe it is necessary to introduce dynamic SLAM based on instance segmentation to fill this gap in the research literature.

2 Method Based on Mathematical Models

Conventional SLAM visual odometry design schemes generally perform geometric calculations on images of neighboring frames using mathematical models to obtain bit-pose estimates of camera motion. Since SLAM was first proposed in 1986 [17], traditional SLAM visual odometry based on mathematical models has become quite mature after more than 30 years of development. In some specific scene environments, this approach has a wide range of applications. Based on whether feature points need to be extracted or not, traditional SLAM visual odometry is classified as a feature point method or a direct method.

2.1 Mathematical Description of the Visual Odometer Problem

The two main tasks of SLAM systems are localization and map building. In order to provide a better initial value for localization and mapping, visual odometry uses information from adjacent frame images to estimate camera motion, and its entire process can be simply summarized as follows: During the motion of the mobile device, we need to solve for the motion from $k-1$ to time k and determine the position (position and pose) P of the mobile device. In the event that the mobile device is in motion, the continuous images captured by the camera can be expressed as $\{I_1, I_2, \cdots, I_n\}$. The transformation matrix between adjacent image frames can be calculated based on the motion of the rigid body in 3D space, denoted $T \in R^{4\times4}$, and then used to solve for the pose P. The transformation matrix $T_{k-1,\,k}$ can be expressed as follows:

$$T_{k-1,\,k} = \begin{bmatrix} R_{k-1,\,k} & t_{k-1,\,k} \\ 0 & 1 \end{bmatrix} \tag{1}$$

where $R_{k-1,\,k} \in R^{3\times3}$ denotes the rotation matrix, and $t_{k-1,\,k} \in R^{3\times1}$ denotes the translation vector. The transformation matrix of camera motion between adjacent frames at all moments can be expressed as $\{T_{1,2}, T_{2,3}, \cdots, T_{n-1,n}\}$, and this provides the camera poses at each moment in time, denoted as P_k:

$$P_k = P_{k-1} \times T_{k-1,\,k} \tag{2}$$

Therefore, the camera position during the entire motion can be obtained as $\{P_1, P_2, \cdots, P_n\}$.

2.2 Static SLAM Visual Odometry

2.2.1 Feature Point Method

An image feature point is composed of two parts: the keypoint represents the location of the feature point in the image, and the descriptor describes the information around the keypoint. In an image, feature points are usually corner points, edges, and regions that can be considered representative locations. Before 2000, researchers extracted corner points directly as features using algorithms such as Harris, Features from Accelerated Segment Test (FAST), and Good Feature Tracking (GFTT) [18–20]. However, in most cases, such pure corner points do not fully meet our feature requirements. Subsequently, researchers designed more stable local image features, such as Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Oriented Brief (ORB), and Binary Robust Invariant Scalable Keypoints (BRISK) [21–24], etc. After extracting the feature points, data association between images can be achieved based on feature matching to solve the inter-camera transformation matrix. When different types of cameras are used in mobile devices, the dimensions of the features obtained may vary. For monocular cameras, the obtained feature points only contain 2D information, and the transformation matrix can be calculated using the pair-polar geometry method. For binocular or RGB-D cameras, 3D feature points are obtained, and the transformation matrix is usually obtained by solving the Iterative Closest Point (ICP) algorithm [25]. If both 3D and 2D feature points are obtained, they can be solved using Perspective-n-Point (PnP), Efficient Perspective-n-Point (EPnP), Perspective Three Points (P3P) [26–28], etc. Since the estimation error at a certain time will inevitably accumulate with time when calculating the motion trajectory incrementally, there will be some error in estimating position and attitude based on the obtained change matrix. To minimize the accumulated error, the first positions and attitudes are usually filtered or optimized using Bundle Adjustment [29]. The basic flow of feature point-based SLAM visual odometry is shown in Fig. 2.

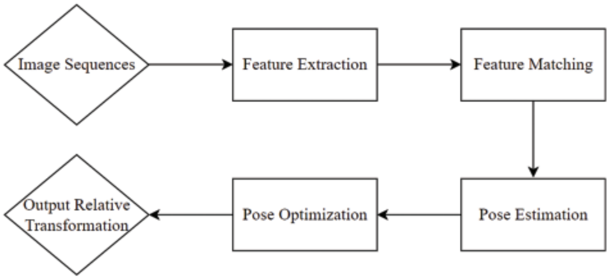


Figure 2: Flow chart of SLAM visual odometry by feature point method

At present, the ORB-SLAM series algorithms are the most popular feature point methods. In 2015, Raul Mur-Artal et al. optimized the method of keyframe selection and map construction based on the PTAM [30] algorithm. They added functions for map initialization and closed-loop detection, and proposed the ORB-SLAM algorithm, which achieved superior results in processing speed, tracking effectiveness, and map accuracy. The biggest feature of this algorithm is that the ORB features of the image are uniformly used in all steps, and the rotation invariance of the ORB features can be used to construct scale invariance using a pyramid, making feature extraction and tracking, keyframe selection, three-dimensional reconstruction, closed-loop detection, and other steps consistent. To address the scale drift that ORB-SLAM may have and the localization failure that occurs during pure rotation, ORB-SLAM2 provides threads for processing binocular or RGB-D camera image input to solve these problems. Additionally, ORB-SLAM3 adds visual inertial odometry (VIO) to ORB-SLAM2 to enable mobile devices to operate in scenes with fewer feature points during long-term operation.

2.2.2 Direct Method

Using the direct method, camera motion is estimated based on the brightness of pixels, which eliminates the need to calculate key points and descriptors, as well as the computation time required for feature extraction and the issues related to feature point scarcity. The direct method works well in scenes with light and dark changes. The direct method has evolved from optical flow and utilizes the assumption of grayscale invariance to establish data correlation from optical flow. To estimate camera motion, it minimizes photometric error, as shown in Fig. 3.

In 2011, Dense Tracking and Mapping (DTAM) [31] became the first SLAM algorithm that does not rely on feature points. It continues the idea of processing key frames, but instead of extracting feature points, it directly analyzes and calculates the information of each pixel in the key frame to optimize the camera motion based on the grayscale and depth information of the image, minimizing the photometric error, and achieving real-time computation. However, because this method requires the use of a GPU, it cannot be deployed on mobile devices for the time being. Additionally, DTAM works best in light environment invariant conditions, limiting its use scenarios. In 2014, Engel et al. proposed Large-Scale Direct Monocular SLAM (LSD-SLAM) [32]. This method generates the gradient of an image by extracting pixels with more significant gradients and tracks the gradient using the variance-normalized photometric error to create a semi-dense depth map. It matches the current frame with the current keyframe to obtain the camera pose of the current frame. One problem with this method is that the number of point clouds obtained by the semi-dense method is insufficient, and the target may be lost when the mobile device moves rapidly. Based on the advantages of both the feature point method and the direct method, Forster et al. proposed the Semi-Direct Visual Odometry (SVO) algorithm [33]. It combines the feature point method and the direct method by first extracting sparse feature points from the image and then matching these feature points with the direct method to estimate the camera motion pose. SVO significantly improves the algorithm's speed by only extracting sparse feature points without descriptor computation. However, the method does not consider the camera rotation view, limiting its applicability to some common scenes.

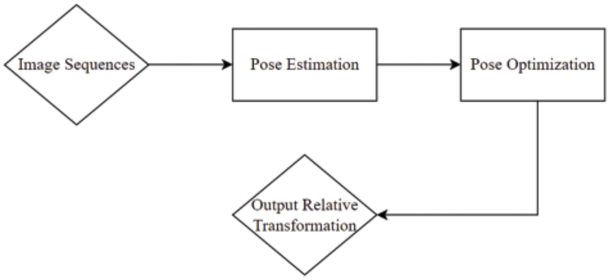


Figure 3: Flow chart of direct SLAM visual odometry

To sum up, the visual odometry method based on feature points is supported by mature mathematical models. The descriptor can stably and robustly describe the characteristics of the image, leading to strong accuracy and robustness. However, there are also some problems associated with this method. It takes time to extract key points and calculate descriptors, resulting in a delay of almost 20 milliseconds even for fast ORB features, which leads to low real-time performance. Additionally, in places with few features, such as white walls, empty areas, or pipes, there may not be sufficient matching points to accurately calculate camera motion. Furthermore, using feature points ignores the surrounding pixel information other than features, making the method vulnerable to dynamic objects and occlusions. On the other hand, the direct method, based on optical flow, addresses some of the shortcomings of the feature point-based method. It does not rely on feature points, avoiding the need for feature calculation time and mitigating the issue of feature missing. It can work as long as there are light and dark changes in the scene due to the assumption of photometric invariance. Additionally, it can use any number of global pixels for pose estimation, thereby improving the utilization of image pixels and allowing for semi-dense or dense map construction. However, the direct method has its challenges. It heavily relies on the strong assumption of photometric invariance, making the system sensitive to photometric changes. If camera exposure parameters or scene illumination changes, the assumption of photometric invariance may be invalidated, and the algorithm may not work correctly. Moreover, the camera's pose is entirely calculated by image gradient search, which may lead to local optimization issues when the image gray level is a non-convex function with respect to coordinates.

2.3 Dynamic SLAM Visual Odometry

2.3.1 Feature Geometry Constraint Method

In a static scene, the images of all 3D points in space must be located on the polar line corresponding to the previous frame's image. Based on this condition, classic visual SLAM uses feature points to find geometric mapping relationships between frames and performs camera inter-frame pose estimation and spatial pose mapping. The processing of dynamic SLAM problems based on feature point geometric constraints is to eliminate dynamic feature points using the properties of antipolar geometric constraints. As shown in Fig. 4, the line $\overrightarrow{C_1x_1}$ and $\overrightarrow{C_2x_2}$ will intersect in the 3D space at the point X, where C_1 and C_2 are camera positions. X can determine a plane called the polar plane. The lines connecting the polar to the image plane I_1 and I_2 are the intersection points of e_1 and e_2. e_1 and e_2 are called the polar point, and C_1C_2 is called the baseline. The polar plane and the two image planes I_1 and I_2 are called the polar lines of intersection of l_1 and l_2. The standard pair of polar constraints is $x_2^T F x_1 = 0$, where x_1 and x_2 are the positions of matching point pairs in consecutive inter-frame images, and F is the basis matrix. The specifics of the violation of geometric constraints in dynamic scenes are shown in Fig. 5, and it is obvious that there are several problems. Due to the motion of feature points occurring after I_2 in the frame x_2, the projection point of the feature point on the polar line in frame I_2 becomes larger, causing the inverse projection ray connected to the camera optical centers and the projection point in both frames to not intersect at a point. The point x_2 in the frame I_1 also has a large distance between the reprojected features and the observed features in the frame, which leads to estimation errors in the base matrix F. These errors affect camera pose estimation and the creation of spatial pose maps.

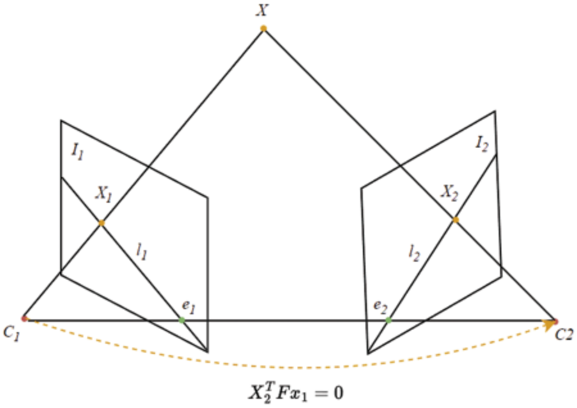


Figure 4: Standard constraints in static scenes

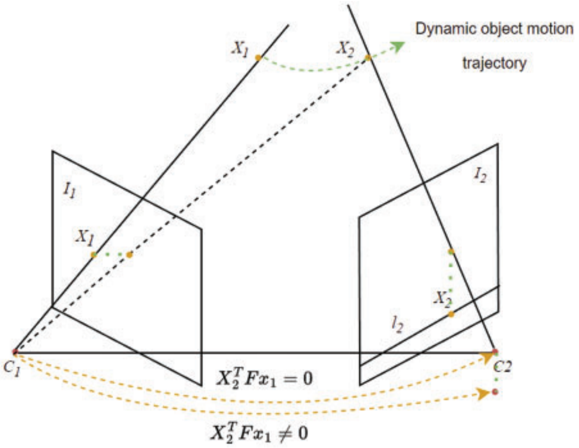


Figure 5: Constraint violation in dynamic scenes

Depending on the conditions of the pair of polar constraints, some dynamic points can be eliminated. However, in three-dimensional space, when the motion direction of an object is parallel to the camera's motion direction, meaning a point moves along a polar plane, the image of that point also moves along a polar plane, resulting in degenerate motion. Existing methods use a planar parallax constraint method to solve this problem by incorporating a third view. In the dynamic environment, it is quite challenging not only to estimate a main reference plane but also to calculate three or more views. To address this problem, Kundu et al. [34] introduced an approach that relies on a probabilistic framework of recursive Bayesian filtering using two geometric constraints to detect dynamic points. The first constraint is the aforementioned pair-polar constraint, which specifies that the static feature points in the previous image frame should be located on the corresponding polar lines in the second image frame. The second constraint uses the flow vector boundary (FVB) constraint in robotics to solve the degenerate motion problem when the motion direction of the object is parallel to the camera's motion direction. Finally, the probability framework in the recursive Bayesian filter is used to calculate the probabilities of the feature points being static or dynamic, based on the limits of the polar lines and extended focus.

The projection error method is another approach in the feature point geometric constraint method. Lin et al. [35] proposed a method based on the combination of a reprojection depth difference cumulative map and static probability to classify feature points for rejection. Because the depth map output from the depth camera may have depth missing at the close object edges due to parallax, it is necessary to use the previous depth image and inter-frame pose relationship for the current depth image to perform depth compensation. Then, the depth difference between adjacent frames is calculated and cumulatively modeled to partition the dynamic region and static region based on the overall image depth cumulative difference. Next, the Euclidean distance and variance of the static Euclidean distance point set between feature points and matching map points in the static and dynamic regions are calculated. Subsequently, the static probability of feature points in the dynamic region is calculated. Feature points are rejected by calculating the Euclidean distance between the feature points in the dynamic region and the matching map points. Keller et al. [36] argued that existing algorithms that use mobile sensors to accumulate depth measurements into a model often assume a static scene or treat dynamic content as outliers, leading to limitations such as small scene size, poor dynamic robustness, and inadequate real-time performance. To address these limitations, Keller et al. proposed a solution by building a new real-time dense reconstruction system scheme based on the findings of Izadi et al. [37]. Outliers are used as depth samples of dynamic objects during the Iterative Closest Point (ICP) process, retrieved by constructing ICP state diagrams, and then obtained unstable model contours. This method allows for stable segmentation of dynamic objects in the scene and continuous updates of the global reconstruction. Zou et al. [38] utilized multi-camera reprojection to process dynamic points by projecting the 3D position of the feature point in the nth frame and comparing the Euclidean distance with the corresponding feature point tracked in the nth frame. If the distance is within a certain threshold, the point is considered static; otherwise, it is considered uncertain. To distinguish these uncertain points, the 3D position is obtained by triangulation and comparing the Euclidean distance between the feature point and the 3D position projection. If it is less than 0, it is considered a dynamic point. Tan et al. [39] obtained the key frame through conditional judgment, projected the key frame to the current frame to detect the appearance and structure of the feature point, and then used the Adaptive Random Sample Consensus (RANSAC) algorithm to remove the abnormal points. This method can effectively and timely remove and update invalid feature points and key frames, allowing the system to maintain robustness in a gradually changing scene. However, it may have difficulty maintaining robustness when the objects in the scene are moving rapidly. Sun et al. [40]

obtained the pixel difference image by subtracting the previous distorted frame with self-motion compensation from the current frame and then separated the static and dynamic parts of the scene by vector quantization segmentation. The drawback of this method is that the static and dynamic parts may not be completely segmented when the parallax between consecutive frames is relatively large or when there are too many moving objects in the scene.

2.3.2 Optical Flow Direct Method

Optical flow is the estimation of pixel motion in successive frames of an image under the assumption of photometric invariance [41]. The motion field in the image corresponds to this estimation, which can be used to segment moving objects. Scene flow in a 3D point cloud and optical flow in a 2D image describe various moving objects.

In 2009, Klappstein et al. [42] were the first to propose a scheme to distinguish moving objects based on motion metrics, where motion metrics refer to the degree to which the image points in the scene motion field obtained by optical flow analysis violate the optical flow motion. They used a Kalman filter to resolve some noise contained in direct optical flow analysis and then used an image segmentation algorithm to distinguish moving objects based on motion metrics. However, this method cannot distinguish different objects because it cannot distinguish different motion directions. Kerl et al. [43] constructed a photometric error function between pixel points with fused motion prior weights based on the assumption of photometric consistency. They then optimized the camera's pose by minimizing the error function. However, this method can only reject a part of dynamic features, and the system is less robust. Derome et al. [44] proposed a fast and dense optical flow method to detect moving objects based on Lucas-Kanade optical flow. This method is calculated from the residuals between the predicted image and the binocular camera image. It has the capability to process large images rapidly and has high compatibility. Ochs et al. [45] demonstrated that considering motion over a period of time, motion will be more effectively utilized. Contrary to the optical flow of adjacent frames, the trajectory across a segment of frame points is not susceptible to short-term changes that impede the separation of different objects. They emphasized that dynamic research based on motion vectors is better than pixel-based research. They proposed a method of segmenting moving objects based on a priori free models for the analysis of long-term point trajectory data. This algorithm can process any length of point trajectory. By using spectral clustering and model selection, the algorithm can process noisy data with outliers and achieve better segmentation results in scenes that contain multiple moving objects. Wang et al. [46] proposed a practical dense motion object segmentation method based on Ochs' research. This method addresses the problems of motion ambiguity and poor robustness when there are no feature regions in dense SLAM. It sequentially completes motion object segmentation through optical flow computation, sparse point trajectory clustering, and densification operations. The main feature of this method is to SLAM before separating different moving objects and then processing rigid and non-rigid moving objects in a unified manner. In 2020, Zhang et al. [47] used optical flow residuals to highlight dynamic semantics in RGB-D point clouds. The scheme first inputs two consecutive frames of images with depth information and then splits them into two simultaneous threads. One thread enters the PWC-Net network to compute optical flow, and the second thread uses image luminosity and depth to obtain the camera's bit pose. The obtained estimated optical flow and camera bit pose can be calculated to get 2D scene flow, and finally, dynamic object segmentation is performed in 2D scene flow using a dynamic clustering algorithm, which takes the average of the sum of the three light intensity residuals, depth residuals, and optical flow residuals as clusters of the clustering algorithm. This scheme can extract different kinds of motion objects compared with other schemes, but it is not good at perceiving slight motion. Additionally, due to the time-consuming computation of dense optical flow, the use of GPU acceleration leads to limited device scenarios.

As a conclusion, both feature geometry constraints and optical flow direct methods are used in dynamic SLAM visual odometry. They aim to first remove the dynamic part of the objects and then use the remaining static part to estimate the camera's position and pose. In feature geometry constraints, static and dynamic points are distinguished based on whether they meet the geometric constraints. Static feature points meet the constraints, while dynamic feature points do not. This method does not add a significant computational burden as feature point matching is already a standard part of dynamic SLAM processing, resulting in higher real-time performance. Additionally, it is supported by mature mathematical models and is not vulnerable to external interference, making it highly robust. However, when the number of feature points is small, the camera pose cannot be accurately obtained because a single geometric constraint may not distinguish whether the dynamic points obtained are due to object motion or incorrect feature point matching. The optical flow direct method has clear advantages over other methods in dealing with non-rigid objects. Optical flow provides both the motion information of moving objects and rich 3-dimensional structure information of the scene, which is beneficial for dense mapping. Moreover, it performs well in scenes with a large number of moving objects. However, this method is sensitive to lighting and camera conditions, and it may struggle to maintain stability in scenes with complex lighting changes. Additionally, setting an appropriate threshold for the segmentation boundary of moving objects can be challenging.

3 Methods Based on Instance Segmentation

The model-based approach has been developed for many years, with perfect mathematical model support. It does not depend on the data in the training set, is less computationally intensive, and can run in real-time in unknown scenes. However, it has various problems, as mentioned above. In the past few years, as deep learning has rapidly developed, more and more researchers have started to incorporate techniques, particularly target detection, semantic segmentation, and instance segmentation, into dynamic SLAM visual odometry to solve the interference problem of dynamic objects in SLAM systems. Unlike model-based methods, deep learning-based methods use semantic tagging or target detection to preprocess potential dynamic objects. This can efficiently remove feature points from dynamic objects, greatly improving localization accuracy, and achieving higher dynamic robustness. Target detection aims to find all the objects to be detected in the image and mark the position of the object with a bounding box. This method has the advantage of fast detection speed but can only mark the approximate position of the object with a box, which cannot achieve accurate segmentation. In SLAM visual odometry, if all the feature points within the box are directly removed, too many static points will be removed, which affects positioning accuracy. Although semantic segmentation achieves pixel-level segmentation, it cannot distinguish different instances of the same category. In a large number of dynamic dense scenes, when different instances of the same category exist, some static and some dynamic, it cannot achieve accurate segmentation. This leads to the wrong removal of static points or wrong utilization of dynamic points, which affects localization accuracy. The method based on instance segmentation overcomes the shortcomings of these two methods and is capable of achieving complete dynamic feature point rejection in complex scenes. As an instance-level segmentation technology that can be implemented, instance segmentation can not only segment the objects to be detected in the image but also distinguish different instances of the same object. In highly dynamic and complex scenes, applying instance segmentation technology as an auxiliary thread to the SLAM visual odometry can achieve a complete separation of dynamic and static regions. This reduces the impact of dynamic objects on the positioning accuracy of the system. Fig. 6 shows the differences between target detection, semantic segmentation, and instance segmentation.

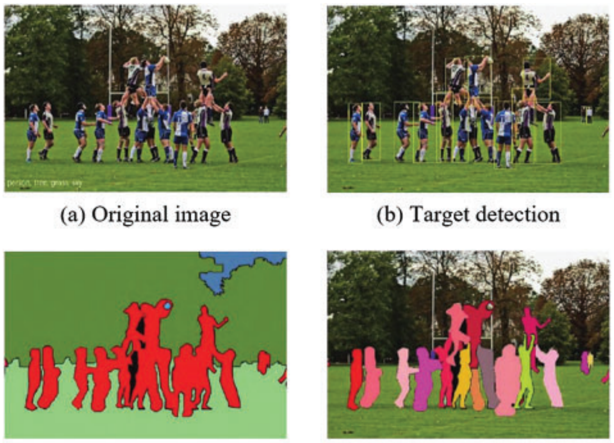

(a) Original image
(b) Target detection
(c) Semantic segmentation
(d) Instance segmentation

Figure 6: Three different types of image segmentation

3.1 Comparison of Instance Segmentation Algorithms

Instance segmentation is a high-level task that combines target detection with semantic segmentation. There are two types of methods: two-stage and one-stage. For a long time, two-stage methods have followed two routes. The first scheme is a top-down approach based on target detection. This scheme uses target detection to locate the box in which each instance is located. Semantic segmentation is then performed within the box to determine the mask for each instance. Additionally, there is a bottom-up scheme based on semantic segmentation. In this scheme, pixels are first identified by semantic segmentation, and then clustering or other metric learning methods are used to differentiate between instances of the same kind. Single-stage instance segmentation algorithms can roughly be divided into two categories: methods based on global image and methods based on local image [48]. The method based on global image does not need to go through the process of clipping and ROI Align. Instead, it first forms the feature map of the entire instance and then uses some operations to combine the features to obtain the final mask of each instance. On the other hand, the method based on local information outputs segmentation results directly. In a sense, a bounding box is a rough mask, using a small boundary rectangle to approximate the outline of the mask. Fig. 7 introduces representative instance segmentation methods on each timeline.

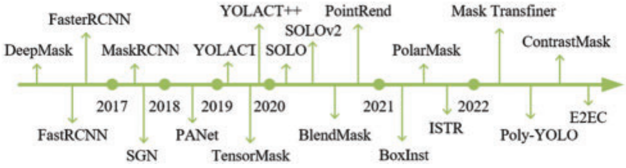


Figure 7: A representative instance segmentation algorithm

3.1.1 Two-Stage Instance Segmentation

In 2015, DeepMask [49] was proposed. The algorithm predicts a mask proposal for each spatial region using the sliding window method. However, this method goes through the fully connected network to extract the mask, leading to the loss of connection between the mask and the features, as well as redundant extractions. Moreover, the use of downsampling methods also results in the loss of location information. The masterpiece of instance segmentation, Mask R-CNN [50], was introduced in 2017. Mask R-CNN adds a mask branch for semantic segmentation on top of the target detection network Faster R-CNN [51], which includes classification and regression branches. The Mask R-CNN framework is shown in Fig. 8. With the mature development in the field of target detection, the Faster R-CNN in Mask R-CNN can be simply replaced with a better detector to steadily improve the instance segmentation results. However, the use of ROI Align, which unifies ROI of different sizes into the same size for batch operation, may lead to information loss of spatial features, especially for large targets, resulting in poor predictions, particularly for those with complex contours. Cascade Mask R-CNN [52] follows a similar approach and designs mask branches based on Cascade R-CNN [53]. It gradually increases the Intersection-Over-Union (IOU) threshold in different layers to improve the prediction accuracy of the bounding boxes and alleviate the mismatch problem. Although Cascade R-CNN has improved segmentation accuracy to some extent, it still does not fully solve the problem of rough edge prediction for large targets in Mask R-CNN.

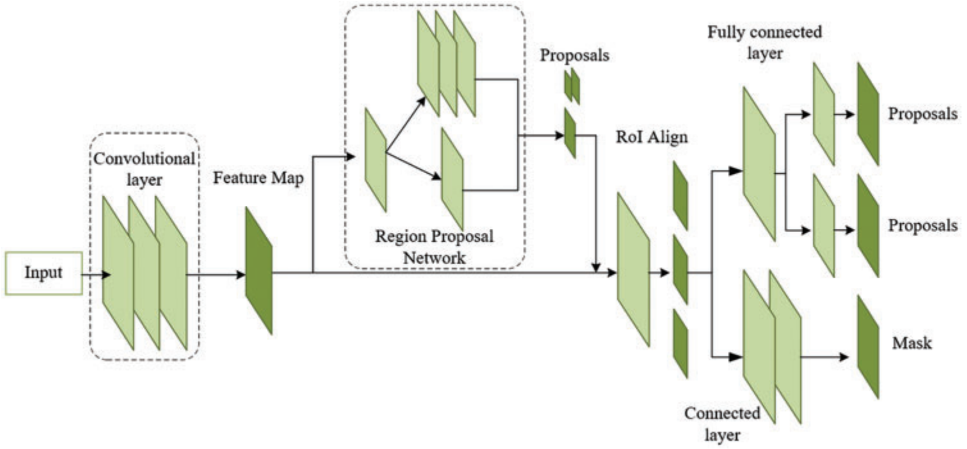


Figure 8: Mask R-CNN framework

HTC [54] is currently the best solution for balancing accuracy and speed in the second stage. It proposes two main innovations. Firstly, it designs a cascade of mask branches to pass mask information from the previous stage to the next stage. Secondly, it adds semantic branching and semantic segmentation supervision to enhance contextual semantic features. While these innovations are not particularly novel, they effectively stabilize the accuracy of instance segmentation at a high level. PANet [55] improves on Mask RCNN by introducing bottom-up path enhancement, dynamic feature pooling, and fully connected layer fusion, resulting in improved instance segmentation performance, the model structure is shown in Fig. 9. Other two-stage top-down methods, such as Mask Scoring R-CNN [56], also suffer from the problems of information loss for large targets and poor edge prediction due to the use of ROI Align for batch operations. Another category is the bottom-up scheme based on semantic segmentation, which first performs pixel-by-pixel classification through semantic segmentation and then distinguishes different instances of the same kind using clustering or other metric learning methods. For example, Brabandere et al. [57] proposed a proposal-independent instance segmentation scheme based on the Pixel Embedded semantic segmentation framework and late mean-shift clustering, which effectively handles occluded objects and distinguishes highly similar objects. Fathi et al. [58] distinguished different instances by calculating pixel similarity using metric learning, while Kong et al. [59] improved the recursive grouping model using the Gaussian fuzzy mean shift (GBMS) algorithm to distinguish different instance pixels. These bottom-up approaches maintain better low-level features, but the subsequent steps, such as clustering or metric calculations, can be cumbersome, and the models may have poor generalization ability. They also require high-quality dense segmentation and may struggle to cope with complex scenes containing many categories.

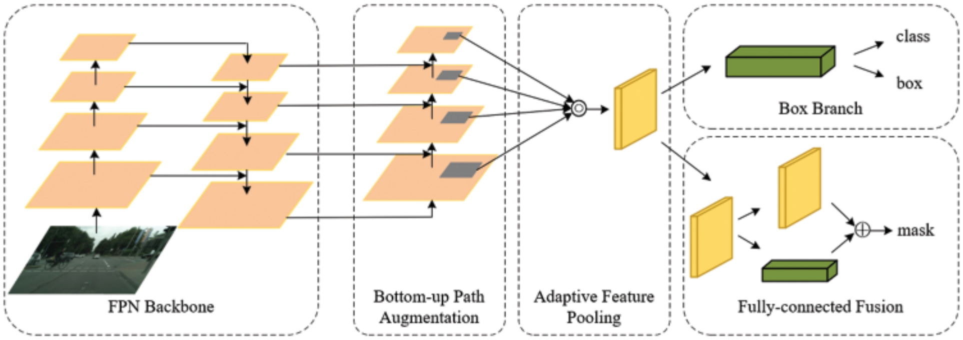


Figure 9: PANet framework

In summary, two-stage instance segmentation uses a local mask to crop regions within the bounding box and align them to the same size. This method offers the advantages of a simple and easy-to-learn mask branch, providing better results for small targets with clear details. However, until 2018, both top-down and bottom-up approaches faced development bottlenecks. Top-down heavily relied on target detection effects, while bottom-up addressed limitations using supervisory frames but struggled with end-to-end training and performance due to clustering algorithms. Neither of these methods solved issues like poor segmentation of large targets, loss of feature information, and bias in box prediction. Consequently, researchers shifted their focus to single-stage instance segmentation methods. For more details on the classification of some two-stage instance segmentation algorithms, please see Table 1.

Table 1: Summary of classification of some two-stage example segmentation algorithms

Algorithm	Year	Innovation points	Advantages	Disadvantages
DeepMask [49]	2015	Generate mask candidates with high regression rates.	Independent of edges, hyperpixels, etc.	Rough boundary.
Mask R-CNN [50]	2017	Replace ROI pooling with ROI align.	Both object detection and instance segmentation.	Dependent target detection.

(Continued)

Table 1 (continued)

Algorithm	Year	Innovation points	Advantages	Disadvantages
Cascade mask R-CNN [52]	2018	Gradually increase the IOU threshold at different layers.	/	Poor edge prediction for complex and large targets.
HTC [54]	2019	Cascade structure.	Multitasking capability.	High parameter calculation.
PANet [55]	2018	Top-down feature path, adaptive fusion ROI pooling.	Enhance information fusion and feature utilization between different scales.	/
Text [57]	2017	Mean-shift clustering.	Detection of obscured objects, good at distinguishing highly similar objects.	Not good at objects of different shapes and sizes.
Text [58]	2017	Metric learning calculates the similarity between pixels.	/	Poor model generalization.

3.1.2 Single-Stage Instance Segmentation

Instance-sensitive FCN [60] was the first to explore the field of single-stage instance segmentation in 2016. Its idea is that, relative to FCN which outputs semantic labels on each pixel, Instance-sensitive FCN needs to output whether it is at the relative position of a certain instance, using the encoded position information to distinguish the same semantics. The effect of this approach was not outstanding at that time, but it provided a good idea for the subsequent single-stage segmentation algorithms. In response to the fact that Instance-sensitive FCN only has a single object output without category information and requires a separate downstream network to complete the category information, FCIS [61] in 2017 outputs both instance masks and category information by computing position-sensitive inside. It also made category detection based on ROI instead of introducing another branch to do this task. YOLACT [62] was proposed in 2019 as a prototype for a series of algorithms such as BlendMask [63], EmbedMask [64], Condinst [65], etc. These are a series of methods that predict a set of instance-specific parameters with a shared 1/4 or 1/8 global feature. YOLACT first generates k prototype masks for each image through a Protonet network, then predicts k linear combination coefficients for each instance, and finally generates instance masks by linear combination. The network structure is shown in Fig. 10. In the subsequent YOLACT++ [66], mainly the concept of mask rescoring is added and BlendMask replaces the YOLACT target detection network from RetinaNet to FCOS [67], and adds an instance-related parameter prediction branch in the detection head part. EmbedMask predicts instance-specific parameters with the global shared feature map. The instance mask is obtained by clustering, learning a set of vectors for each instance, and then calculating the Euclidean distance between the instance vector and the vector of each pixel. When the distance is less than a given threshold, the current pixel is considered to belong to the current instance. CondInst also uses FCOS as the target detection network and introduces relative position features after the P3 layer in FCOS. SOLO [68] series is more prominent in the current instance segmentation algorithms in terms of accuracy and speed. SOLO proposes an idea to transform the instance segmentation problem into a classification problem of pixel instance labels. Inspired by the target recognition network YOLO [69], SOLO performs instance segmentation on a grid. SOLO first divides the feature maps of different layers of FPN equally into S $*$ S grids. Different positive and negative sample assignment strategies are given for the same and different layers of FPN. The category branch is responsible for predicting object categories, while the mask branch predicts the probability that S2 pixels belong to each grid instance. In the test phase, NMS [70] is used to get the final mask results. The framework of SOLO and SOLOv2 is shown in Fig. 11. SOLOv2 [71] adopts a dynamic convolution approach similar to CondInst, which predicts a set of convolution parameters for each grid representative instance and dynamically convolves its FPN feature map after fusion of each resolution to get the instance mask. It also proposes a faster method of non-maximal value suppression, Matrix NMS, achieving a balance of precision and speed, but there is a significant contrast in the performance of small and large targets, with poor results for small targets and good results for large targets. In 2020, the He group proposed PointRend [72]. This method optimizes image segmentation of object edges, making it perform better in parts of object edges that are difficult to segment. It also solves the problem of how to quickly calculate masks, reduces resource consumption, improves segmentation accuracy, and the model has the advantage of scalability. It has an excellent segmentation effect for moving video with dense occlusion. Tencent's RCG Research Center has also proposed a method that is effective for video instance segmentation. QueryInst [73] is an end-to-end instance segmentation method based on Query, consisting of a query object detector and six parallel monitoring-driven dynamic masks. It is the first query-based instance segmentation method, achieving the best performance among all online VIS methods, and achieving an ideal balance between speed and accuracy.

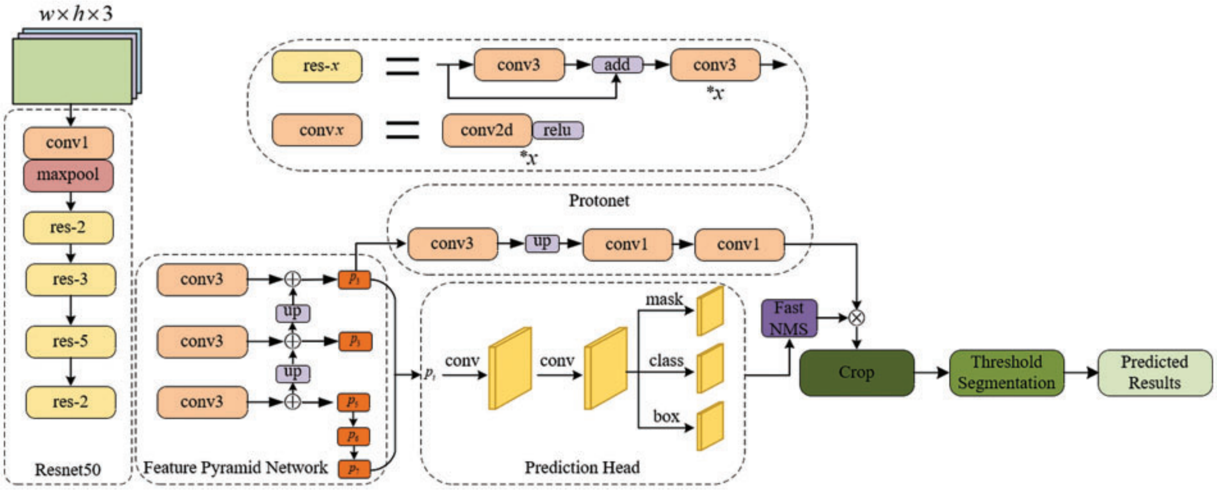
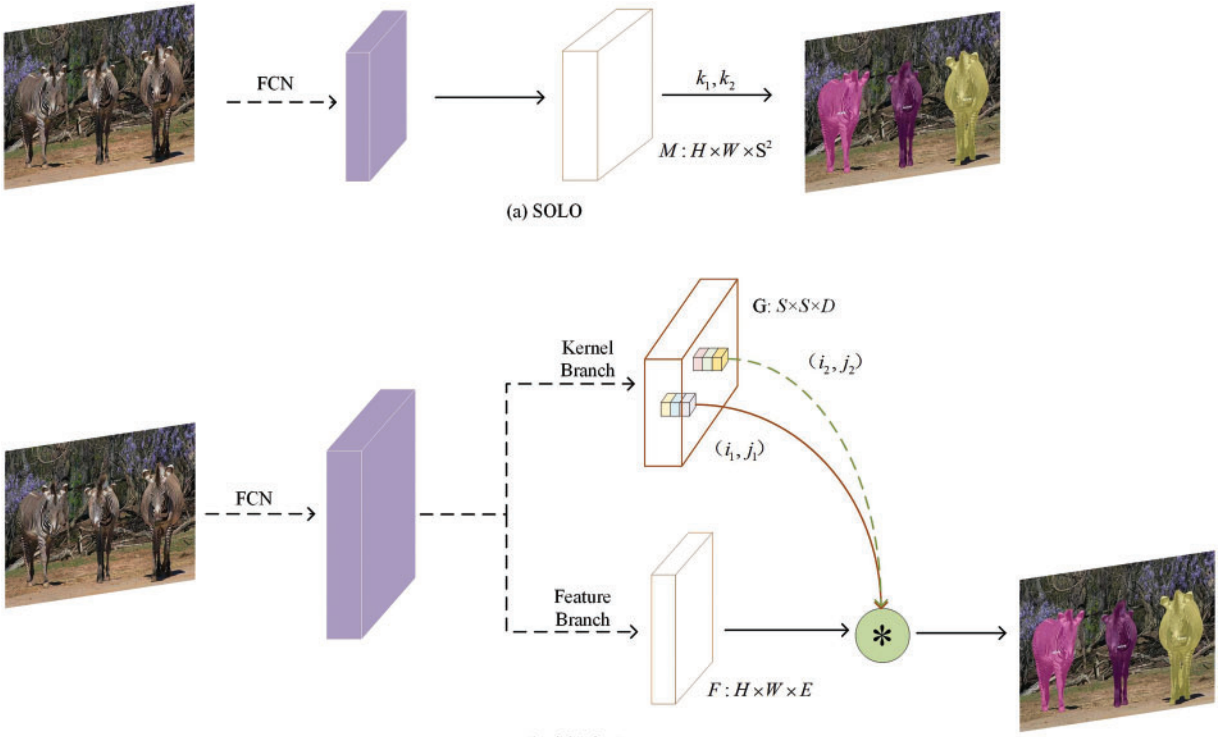


Figure 10: YOLACT network structure



(a) SOLO

(b) SOLOv2

Figure 11: SOLO and SOLOv2 frameworks

Local region-based approaches output segmentation results of instances directly based on local information. The bounding box is, in a sense, a rough mask that uses the smallest bounding rectangle to approximate the contour of the mask. ExtremeNet [74] detects using four polar points, and this rich parameterization can be naturally extended by using polar points on its corresponding edges in both directions to extend to a segment of 1/4 of the entire edge length to form an octagonal mask. Based on the inspiration of ExtremeNet, a series of subsequent researchers have attempted to encode the contours of the instance masks as fixed-length coefficients given different decomposition bases. Ese-Seg [75] designed an inner radius shape signature for each instance and fitted it to a Chebyshev polynomial fit; PolarMask [76] uses rays spaced at a constant angle from the center to describe the contour; FourierNet [77] introduces a contour shape decoder using the Fourier transform and implements smoother bounds than PolarMask. These methods based on contour prediction masks typically use 20 to 40 coefficients to parameterize the masked contour. This method is fast in inference and easy to optimize, but it cannot accurately depict the mask and cannot depict objects with holes in the center. Poly-YOLO [78] is a polygonal instance segmentation scheme based on the idea of YOLOv3. Instance segmentation is performed through limited polygons, which not only reduces the amount of training parameters for YOLOv3 but also greatly improves the average accuracy.

ContrastMask [79] is a new instance segmentation method that makes full use of training data, allowing data from new categories to contribute to the optimization process of the segmentation model. Through a unified pixel-level contrast learning framework, the segmentation ability of basic categories is transferred to new categories. This improves the ability to distinguish between the foreground and background of both basic and new categories. However, the correctness of the new category foreground and background partitions cannot be guaranteed. In 2022, Zhang et al. proposed

E2EC [80] as a multi-stage and efficient end-to-end contour-based instance segmentation model, which can be used for high-quality instance segmentation. The modular idea proposed by E2EC can easily be applied to other contour-based instance segmentation methods.

In 2021, Cheng et al. integrated the advantages of the DETR model [81] and the Transformer architecture [82] and proposed MaskFormer [83], an image segmentation model that treats semantic segmentation and instance segmentation as the same task. MaskFormer extracts features through a backbone network and passes the features to a pixel decoder. The decoder gradually upsamples the image features, generating pixel-by-pixel embeddings that capture local and global contextual information for each pixel in the image. Simultaneously, the Transformer Decoder attends to the image features and produces a set of "N" segment embeddings, assigning weights to different parts of the image using attention mechanisms. By extending the prediction mechanism of the DETR model, MaskFormer creates class-specific masks for each detected object and generates high-quality segmentation masks. This approach overcomes the limitations of traditional pixel-by-pixel and mask classification methods. It addresses the challenge faced by classical methods like Faster R-CNN and Mask R-CNN, which struggle to handle the overlapping of objects of the same class, resulting in inconsistent classifications. Mask2Former [84] improves upon MaskFormer's overall structure by exchanging the order of self-attention and cross-attention in the Transformer Decoder, thus accelerating the model's convergence. It leverages multi-scale high-resolution features by feeding different layers of the pixel decoder as a feature pyramid into different layers of the Transformer Decoder, enhancing the model's segmentation accuracy for small targets. Furthermore, Mask2Former employs sampling of random points to calculate the mask loss, significantly improving the model's training efficiency without compromising performance. In 2023, Facebook introduced the Segment Anything Model (SAM) [85], which is a large-scale model in the field of image segmentation. Traditional image segmentation relies on fixed labels for prediction and can only accomplish specific scene-related tasks. SAM, however, enables zero-shot transfer to new image distributions and tasks, allowing it to predict segmentation for any target that requires segmentation. The model structure of SAM is not overly complex. It utilizes the Vision Transformer (ViT) model [86] to encode the image and requires a cueing encoder to embed cueing features, which can include coordinates, bounding boxes, and text descriptions. The encoded image features and cueing features are then combined in a lightweight mask decoder based on the DETR design to predict segmentation masks. The success of SAM can be attributed not only to its model structure but also to the vast amount of training data. Facebook has established a "data engine" divided into three stages: assisted-manual, semi-automatic, and fully automatic. This data engine generates data using an initial basic Segment Anything model, which is then used to train the Segment Anything model iteratively. This process enables the accumulation of more data and the improvement of the model's performance.

In summary, the development of single-stage instance segmentation is mainly attributed to advancements in single-stage target detectors like RetinaNet, CenterNet, and FCOS. These detectors have inspired several good ideas for instance segmentation, resulting in many fast processing methods capable of achieving real-time or near real-time performance (30+ FPS). The primary limitation for instance segmentation speed is often Non-Maximum Suppression (NMS). To achieve real-time performance, YOLACT utilizes Fast NMS, and SOLOv2 uses Matrix NMS, both experimentally proven to enhance instance segmentation speed. The single-stage instance segmentation algorithm offers various complex and open solutions, making it currently the most effective approach in terms of accuracy and speed. Moreover, it overcomes the limitation of detection frames, signifying the future trend in instance segmentation research. Table 2 presents a summary of the classification of some single-stage instance segmentation algorithms, while Table 3 shows the accuracy and processing speed of these algorithms on the COCO dataset. The algorithms are evaluated using ResNet-101-FPN as the base network, with the best-performing learning rule from the original paper and IOU criterion to measure Average Precision (AP). Params represent the network parameters. Instance segmentation techniques aim to achieve algorithmic real-time performance and high accuracy. Single-stage instance segmentation is on par with the performance of two-stage methods but offers simpler, more efficient, and easier-to-train network architecture. Despite the progress, there is still room for improvement in the existing algorithms' performance. Therefore, the desired direction of development should prioritize achieving fast real-time instance segmentation while pursuing accuracy improvement, making it better suited for practical applications.

Table 2: Summary of classification of some single-stage example segmentation algorithms

Algorithm	Year	Innovation points	Advantages	Disadvantages
Instance-sensitive FCN [60]	2016	Location sensitive map.	/	Very low precision.
FCIS [61]	2017	Internal/external position sensitivity map.	/	Low precision.
YOLACT/YOLACT++ [62,66]	2019	Fusion of prototype and inspection maps.	Good real-time.	Lower precision.
BlendMask [63]	2020	New attention mechanism for computing global features.	Good segmentation effect.	Slow processing speed.
CondInst [65]	2020	Dynamic network direct output mask.	High speed and high precision.	Big target details do not work well.
SOLO/SOLOv2 [68,71]	2020	Matrix NMS, Point features.	High speed and high precision.	Long training time.
PolarMask [76]	2020	Polar coordinate modeling mask.	Fast speed.	Blurred edge information.
QueryInst [73]	2021	New upsampling	Good video effect.	/
Poly-YOLO [78]	2022	Limit polygon execution.	High accuracy.	Slow processing speed.
ContrastMask [79]	2022	Shared pixel level contrast loss.	Easy to distinguish between foreground and background.	Low segmentation accuracy.
E2EC [80]	2022	Multi stage, efficient end-to-end.	Easy to apply to other contour-based methods.	/

Table 3: Performance comparison of different algorithms in COCO datasets

Algorithm	Basic network	AP	AP_{50}	AP_{75}	AP_S	AP_m	AP_l	Fps	GPU
Mask R-CNN [50]	ResNet-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4	5.1	V100
Cascade mask R-CNN [52]	ResNet-101-FPN	38.4	60.2	41.4	20.2	41.0	50.6	8.1	V100
PANet [55]	ResNeXt-101-FPN	40.0	62.8	43.1	18.8	42.3	57.2	23.8	V100
YOLACT++ [66]	ResNet-101-FPN	29.8	48.5	31.2	9.9	31.3	47.7	33.3	V100
BlendMask [63]	ResNet-101-FPN	38.4	60.7	41.3	18.2	41.5	53.3	9.8	1080Ti
CondInst [65]	ResNet-101-FPN	39.1	60.9	42.0	21.5	41.7	50.9	12.0	1080Ti
PolarMask [76]	ResNet-101-FPN	30.4	51.9	31.0	13.4	32.4	42.8	12.3	V100
SOLO [68]	ResNet-101-FPN	37.8	59.5	40.4	16.4	40.6	54.2	22.8	V100
SOLOv2 [71]	ResNet-101-FPN	39.7	60.7	42.9	17.3	42.9	57.4	31.4	V100
PointRend [72]	ResNet-50-FPN	36.3	/	39.7	/	/	/	/	V100
QueryInst [73]	ResNet-50-FPN	36.2	56.7	39.7	/	/	/	32.3	V100
Poly-YOLO [78]	SE-Darknet-53	39.2	62.3	41.2	/	/	/	58.6	V100
ContrastMask [79]	ResNet-101-FPN	36.6	62.2	37.7	17.5	36.5	50.1	/	V100
E2EC [80]	DLA-34	33.8	52.9	35.9	/	/	/	30.1	V100

3.2 Comparison of Dynamic SLAM Algorithms Based on Instance Segmentation

The dynamic confirmation design method based on instance segmentation usually chooses a two-threaded design. It combines the potential moving objects segmented by the instance segmentation network with traditional dynamic SLAM methods, such as multi-view geometry, in order to eliminate all dynamic features, as shown in Fig. 12. This is a typical two-threaded design dynamic removal scheme. Yu et al. [87] proposed a DS-SLAM system for dynamic environments in ORB-SLAM2 visual odometry, combining SegNet [88] semantic segmentation networks with optical flow-based object movement consistency checking methods. This system filters out dynamic elements of the scene by removing all dynamic feature points, reducing the impact of dynamic objects on the whole SLAM system. In the later dense map creation process, it matches semantic labels and establishes a dense 3D semantic octree map, greatly enhancing the robot's external perception capability, and significantly improving the accuracy of system localization in dynamic environments. The SegNet network used in the semantic segmentation thread in DS-SLAM is a semantic segmentation depth network for images released in 2017. It can segment 20 objects, including people, as potentially moving objects. At the same time, another thread obtains feature points by calculating the optical flow pyramid, uses the RANSAC algorithm to find the basic matrix, and calculates the position of the polar lines in each frame. If the distance between the feature points and the polar lines is greater than the threshold value, it is considered as a potential dynamic point. Finally, the number of potential dynamic points on different segmented objects is counted. If the threshold value is exceeded, the object is considered to be a moving object, and all feature points on the object are removed. The author tested the dynamic sequence of the TUM RGB-D dataset, and the results showed that the performance of the DS-SLAM system on most high dynamic sequences was improved by an order of magnitude compared to ORB-SLAM2. However, due to the use of a semantic segmentation network rather than an instance segmentation network in the image segmentation thread, the accuracy was insufficient in dense and complex dynamic scenes. In the same year, Bescos et al. [89] proposed DynaSLAM with the same design idea as DS-SLAM, both of which add image segmentation threads to the ORB-SLAM2 system. With monocular or binocular cameras, DynaSLAM extracts the a priori dynamic objects from the image using Mask R-CNN, then extracts the remainder from the feature points. This brute force approach of treating all a priori objects as dynamic objects and all non-a priori objects as static objects is not an adequate approach to reflect the real motion of the scene, which also leads to the poor performance of DynaSLAM when using monocular or binocular cameras. When using an RGB-D camera, Mask-R-CNN performs dynamic segmentation while using a multi-view geometry method to detect dynamic feature points. Then, a region growth algorithm is used to further obtain non-priority dynamic objects in the scene. By combining these two threads, moving objects in the scene can be extracted to the maximum extent. Finally, static information from previous frames is used to repair the parts occluded by dynamic objects. DynaSLAM's performance in the TUM dataset is generally comparable to that of DS-SLAM, but its performance in the walking rpy sequence is significantly better by nearly an order of magnitude.

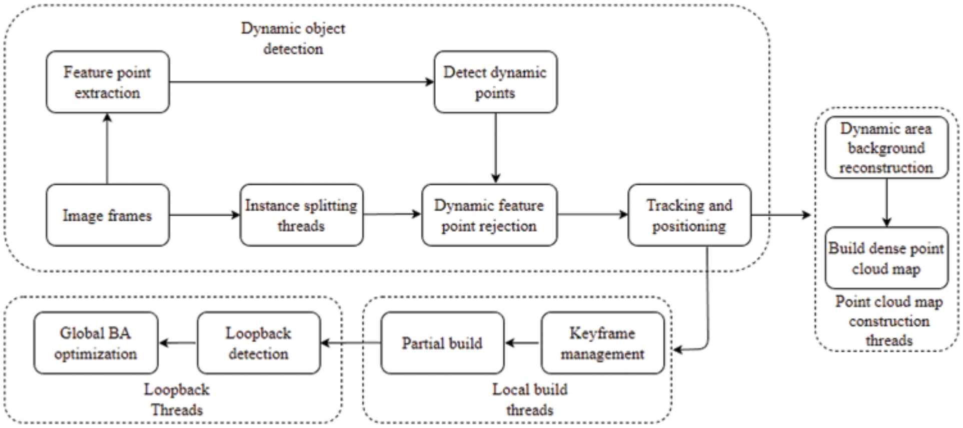


Figure 12: Typical dynamic removal framework

DS-SLAM and DynaSLAM pioneered the application of instance segmentation techniques to dynamic SLAM, significantly improving visual SLAM localization accuracy in dynamic environments. Their design ideas have inspired many subsequent researchers. After generating the motion mask, Li et al. [90] used semantic information to assist global dense optical flow constraints based on ORB-SLAM3 to eliminate dynamic objects. First, the method uses Mask R-CNN instance segmentation network to obtain the semantic mask of dynamic objects a priori. The dense optical flow is then calculated to obtain the optical flow mask generated by motion. The method determines whether each pixel point has dynamic information in the optical flow mask by traversing the pixels of the semantic mask. This approach is significantly better relative to DS-SLAM in terms of improving the impact of dynamic objects, and it also exhibits better real-time performance than DynaSLAM. Due to the difficulty of running neural networks in real-time in SLAM systems, some researchers have considered detecting only key frames and propagating the results to other frames later. RDS-SLAM [91] adds semantic threads to the ORB-SLAM3 system and uses Mask R-CNN for instance segmentation only in key frames. The results of segmented semantic information are fused using Bayesian theory to map points, updating the movement probability of map points matching the keyframe features. Martin et al. [92] proposed MaskFusion, an instance segmentation method that integrates semantic and geometric information, and applied it to SALM visual odometry. Based on image depth discontinuity and normal vectors, geometric segmentation can provide object boundaries in real-time and accurately, making up for the shortcomings of Mask R-CNN. In a complete system, geometric segmentation is used for frames without semantic object masks, combined with frames with semantic masks. MaskFusion is a real-time SLAM system that can represent scenes at the object level, but because too small objects cannot provide sufficient geometric information, it is not suitable for small dynamic object scenes. When robots operate in complex dynamic environments, simplifying the dynamic SLAM problem by deleting moving objects or tracking moving objects alone can improve the accuracy of the system. However, this strong assumption limits the development of autonomous mobile robot systems in designing a series of important realistic scenarios, such as highly dynamic and unstructured environments. To achieve the utilization of dynamic objects, Zhang et al. [93] proposed the dynamic target perception system VDO-SLAM, which makes full use of instance-level semantic information and can estimate the motion of rigid objects in a scene without having a priori information about the target shape or motion model. The system first segments and identifies potentially movable poses in the scene through an instance segmentation network. It then achieves dynamic confirmation by optical flow estimation of all points within the semantic mask using dense optical flow. Finally, the system tracks the dynamic target and correlates the dynamic target by successive frames to obtain motion estimation of the target.

With the continuous development of instance segmentation algorithms, more instance segmentation algorithms with high accuracy and good real-time performance are being used in dynamic SLAM visual odometry. Chang et al. [94] used YOLACT to segment keyframes, causing feature points belonging to dynamic objects to be discarded. To supplement missing dynamic targets, optical flow was introduced to detect artificially driven dynamic targets. Geometric constraints were also introduced to further filter missing dynamic targets. Xu et al. [95] chose YOLACT++ as the instance segmentation module and Mask Flownet-S [96] as the optical flow prediction module to propose a high real-time dynamic region mask detection algorithm. This was done to address the inconsistency of the background optical flow direction of dynamic and static objects. The acquired images are simultaneously entered into the instance segmentation network and optical flow network to generate dynamic region masks based on optical flow direction. Dynamic region mask detection is then performed based on the optical flow field vector direction information to filter the feature points in the dynamic region masks. The localization accuracy is improved by 44% compared to DS-SLAM localization accuracy in high dynamic scenes in the TUM dataset and 20% compared to DynaSLAM in half of the high dynamic scenes. Wang et al. [97] proposed a dynamic RGB-D SLAM method based on augmented segmentation to address the problem of incomplete dynamic point rejection due to missed segmentation in instance segmentation. They used the K-means [98] algorithm to cluster the depth image and then combined it with the YOLACT instance segmentation network to determine whether there is a missed segmentation in the current frame. If so, the segmentation result is based on the multi-frame information. Finally, the motion of each instance object in the scene is determined based on the repaired instance segmentation results. Zhang et al. [99] used SOLOv2, a current instance segmentation network with better comprehensive performance, to solve the problem of unreliable RANSAC algorithm in large dynamic scenes in the ORB-SLAM2 system, achieving real-time processing but with poor performance in low dynamic scenes. Sun et al. [100] also used SOLOv2 to segment out potentially dynamic regions, combined with the use of regional dynamic attitude and geometric constraints to further enhance the filtering effect of dynamic points. They designed SOLO-SLAM to add new semantic constraints based on the semantic properties of map points, which to some extent solves the problem of fewer optimization constraints brought by dynamic information filtering and significantly improves the accuracy of ORB-SLAM3. Table 4 shows the summary of the partial dynamic SLAM scheme, and Table 5 shows the absolute trajectory error (ATE) comparison of the partial dynamic SLAM scheme with ORB-SLAM2 on the dynamic sequences of the TUM dataset. Table 6 shows the performance of selected SLAM algorithms on the KITTI dataset. The KITTI dataset is an open dataset widely used in the fields of computer vision and autonomous driving. It was collaboratively created by the Karlsruhe Institute of Technology in Germany and the Toyota Research Institute in the United States with the aim of supporting various computer vision tasks and research in autonomous driving.

Table 4: Performance of some single-stage instance segmentation algorithms on the COCO dataset

Program name	Year of publication	Camera type	Methods	Real-time
DS-SLAM [87]	2018	RGB-D.	SegNet+Motion consistency detection.	√
DynaSLAM [89]	2018	Monocular, binocular, RGB-D.	Mask R-CNN+Multiview geometry.	×
Text [90]	2021	RGB-D.	Mask R-CNN+ Dense optical flow.	×
Text [95]	2022	RGB-D.	YOLACT+++ MaskFlownet-S optical flow.	√
Text [97]	2022	RGB-D.	YOLACT+K-means+symmetric transfer error.	√
Text [99]	2020	Monocular, binocular, RGB-D.	SOLOv2+RANSAC.	√

Table 5: Performance of partial dynamic SLAM schemes compared to ORB-SLAM2 in TUM dataset

Sequences	DS-SLAM [87]	DynaSLAM [89]	Text [95]	Text [97]	Text [99]
fr3_sitting_static	25.94%	/	39.90%	34.10%	/
fr3_sitting_xyz	/	−66.67%	7.70%	3.26%	−28.57%
fr3_walking_half	93.76%	92.88%	96.80%	/	/
fr3_walking_static	97.91%	93.33%	98.00%	98.00%	79.70%
fr3_walking_xyz	96.71%	96.73%	97.90%	97.90%	91.78%
fr3_walking_rpy	48.97%	94.71%	96.20%	95.49%	/

Table 6: Compare the ATE [m] performance of some algorithms on the KITTI dataset

Sequence	ORB-SLAM2 [7]	DS-SLAM [87]	DynaSLAM [89]	Text [93]	Text [94]
00	1.3	1.4	1.4	1.2	1.3
01	10.4	9.2	9.4	9.0	8.1
02	5.7	10.3	6.7	5.4	6.8

(Continued)

Table 6 (continued)

Sequence	ORB-SLAM2 [7]	DS-SLAM [87]	DynaSLAM [89]	Text [93]	Text [94]
03	0.6	0.6	0.6	0.6	0.8
04	0.2	0.2	0.2	0.2	0.1
05	0.8	1.5	0.8	0.8	0.7
06	0.8	0.8	0.8	0.7	0.8
07	0.5	0.5	0.5	0.5	0.5
08	3.6	5.7	3.5	3.2	3.5
09	3.2	6.5	1.6	1.6	1.5
10	1.0	1.2	1.2	1.0	0.9

In summary, dynamic SLAM visual odometry based on instance segmentation combines the semantic information at the instance level obtained from the instance segmentation network with the results of the tracking thread to identify the dynamic and static regions in the scene. It selects feature points in the static regions for pose estimation, resulting in much better performance than traditional SLAM algorithms on the TUM dataset. However, the high time consumption of instance segmentation poses challenges for achieving real-time SLAM systems. Additionally, instance segmentation can only identify a priori objects, limiting its effective application to non-a priori objects in the environment.

4 Currently Challenges and Future Research Directions

4.1 Efficient Instance Segmentation Algorithms

A large number of experimental studies have demonstrated that semantic information can be used to address the interference of dynamic objects in SLAM systems. However, existing instance segmentation algorithms are difficult to apply widely in practical robots, both in terms of speed and accuracy. To address this issue, more efficient instance segmentation algorithms can be designed by making improvements in the following areas.

Firstly, existing instance segmentation algorithm models have a large number of parameters, which leads to high computational complexity. In the future, lightweight model structures and faster inference algorithms can be designed, such as those based on lightweight models like MobileNet [101] or EfficientNet [102], as well as inference acceleration using neural network accelerators like GPUs and TPUs, to improve the running speed and real-time performance of the algorithms.

Secondly, the comprehensive performance of current single-stage instance segmentation algorithms is better than that of two-stage algorithms. However, the one-stage method without anchor boxes cannot handle small object segmentation well. This is reflected in poor far-field image segmentation effects when applied to robots. In the future, research can be conducted on how to effectively fuse features of multiple scales to improve the accuracy of instance segmentation algorithms.

Currently, instance segmentation in training mostly uses datasets such as COCO or Cityscapes. To enable robots to better operate in complex dynamic scenes, larger, higher quality, and more challenging 3D image datasets are required. To address the challenge of a small number of samples in the training set, transfer learning and data augmentation are effective solutions currently available. Optimizing instance segmentation tasks in a targeted manner will increase the robustness of the network when applied in SLAM systems.

4.2 Object Overlap and Occlusion

In dynamic environments, objects may overlap and occlude each other, posing a huge challenge to the perception and recognition tasks of robots. Due to occlusion, the object can be divided into multiple parts, resulting in instance fragmentation. The current method of clip merging is expensive, complex, and time-consuming. Although the use of structured 4D tensors [103] to represent masked objects in the spatial domain and the use of spatial transformer networks can partially solve the problem, both single-stage and two-stage instance segmentation algorithms cannot fully solve the segmentation problem caused by object overlap and occlusion. Therefore, research on how to improve the network's handling of instance fragments has further value.

To improve the segmentation of instances in overlapping and occluded scenes, new approaches can be explored, such as designing algorithms based on semantic information, incorporating depth or motion cues, and incorporating temporal information.

4.3 Improving Robot Autonomous Decision-Making Ability

It is no longer sufficient for modern robots to simply recognize dynamic objects. They must also be able to infer the next actions of those objects based on their movements, which requires a higher level of autonomous decision-making ability. This ability can be enhanced through several methods. Combining multiple sensors can improve the robot's perception and allow it to more accurately perceive its surroundings. Building more precise environmental maps can help the robot better understand its environment and make more informed decisions. Path planning methods based on heuristic algorithms like A$*$ or Dijkstra can enable robots to plan efficient action paths based on both environmental and perceptual information. Finally, reinforcement learning algorithms can be applied to robot decision-making, allowing them to make more rational decisions based on the dynamic and uncertain nature of their surroundings.

4.4 Time-Varying Semantic Map

The current semantic map built by SLAM system is still not suitable for long-term positioning. If a time-varying semantic map can be constructed to account for temporal changes, it will greatly enhance the long-term and dynamic positioning of robots. There is a lack of research in this area in current semantic SLAM, and it can be considered to apply recurrent neural networks (RNNs) to the mapping of dynamic SLAM systems to achieve long-term autonomous localization of robots.

4.5 Unknown Environment

Since the objects handled by instance segmentation are a priori objects, not all dynamic objects can be completely extracted in the face of an unknown environment. This limitation is a key factor in restricting the upper limit of the development of instance-level semantic SLAM. To address this problem, we need to design auxiliary threads to help the instance segmentation network achieve the extraction of non-a priori objects. However, increasing the number of auxiliary threads will certainly affect the real-time performance of the system and the number of parameters. Therefore, it is necessary to design a better solution to achieve the segmentation of non-a priori objects.

4.6 Information Loss

The majority of current algorithms achieve robustness by eliminating dynamic objects, but this will result in the system losing a great deal of useful information. Therefore, it is necessary to study how to make better use of the dynamic objects extracted from the instance segmentation network instead of discarding them directly. The segmented dynamic objects are also very helpful for building real-time environmental maps.

5 Conclusion

Instance segmentation is an important branch in computer vision that aims to separate each object in an image and assign a unique identifier. This is crucial for robots to perceive and recognize objects in different scenes. Simultaneous Localization and Mapping (SLAM) is a technology that allows robots to autonomously navigate in unknown environments by simultaneously locating and building maps. Dynamic SLAM based on instance segmentation is a rapidly developing field that aims to solve the challenges of perception and navigation of robots in complex dynamic environments. However, due to the overlap and occlusion of dynamic objects in real environments, susceptibility of cameras to external interference, and limitations of instance segmentation network segmentation effect and real-time performance, the current dynamic SLAM technology based on instance segmentation cannot be widely applied in robots.

This article first introduces the traditional design methods based on mathematical models and concludes that they have low accuracy and limited applicability in complex environments. It then comprehensively introduces instance segmentation networks and the dynamic SLAM method based on instance segmentation. By comparing the design methods and performance of two-stage instance segmentation networks and single-stage instance segmentation networks on datasets, we conclude that single-stage instance segmentation has much higher real-time performance than two-stage instance segmentation networks, and the accuracy is gradually catching up with the two-stage instance networks. We analyze and compare various dynamic SLAM systems that combine instance segmentation networks and summarize their specific performance on the dataset. Although the accuracy has been improved compared to traditional methods, the real-time performance still cannot meet the requirements. Finally, we discuss the current challenges and future development directions of dynamic SLAM based on instance segmentation. We believe that a more efficient single-stage instance segmentation network and the improvement of robot autonomous decision-making ability, as well as how to handle non-prior dynamic objects in the environment, are the keys to solving the problem that current dynamic SLAM systems cannot be widely applied. We hope that this article can provide an introduction and inspiration for relevant practitioners.

Acknowledgement: The authors are highly thankful to the National Natural Science Foundation of China, to the Innovation Fund of Chinese Universities Industry-University-Research, to the Research Project for Young and Middle-Aged Teachers in Guangxi Universities, and to the Special Research Project of Hechi University. 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.

Funding Statement: The authors are highly thankful to the National Natural Science Foundation of China (No. 62063006), the Natural Science Foundation of Guangxi Province (No. 2023GXNS-FAA026025), 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, Q. Yang; data collection: D. Chen; analysis and interpretation of results: C. Yang, Y. Xu; draft manuscript preparation: 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] A. M. Barros, M. Michel, Y. Moline, G. Corre and F. Carrel, "A comprehensive survey of visual SLAM algorithms," *Robotics*, vol. 11, no. 1, pp. 24–50, 2022.

[2] W. Chen, G. Shang, A. Ji, C. Zhou, X. Wang *et al.,* "An overview on visual SLAM: From tradition to semantic," *Remote Sensing*, vol. 14, no. 13, pp. 3010–3056, 2022.

[3] T. Taketomi, H. Uchiyama and S. Ikeda, "Visual SLAM algorithms: A survey from 2010 to 2016," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, pp. 1–11, 2017.

[4] J. Fuentes-Pacheco, J. Ruiz-Ascencio and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: A survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 55–81, 2015.

[5] G. Younes, D. Asmar, E. Shammas and Z. John, "Keyframe-based monocular SLAM: Design, survey, and future directions," *Robotics and Autonomous Systems*, vol. 98, pp. 67–88, 2017.

[6] 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.

[7] 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.

[8] 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.

[9] 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, 2018.

[10] M. R. U. Saputra, A. Markham and N. Trigoni, "Visual SLAM and structure from motion in dynamic environments: A survey," *ACM Computing Surveys*, vol. 51, no. 2, pp. 1–36, 2018.

[11] D. Zhang, Z. Zheng, R. Jia and M. Li, "Visual tracking via hierarchical deep reinforcement learning," in *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 35, no. 4, pp. 3315–3323, 2021.

[12] K. Wang, Y. Lin, L. Wang, L. Han, M. Hua *et al.,* "A unified framework for mutual improvement of SLAM and semantic segmentation," in *2019 Int. Conf. on Robotics and Automation (ICRA)*, Montreal, QC, Canada, IEEE, pp. 5224–5230, 2019.

[13] A. Beghdadi and M. Mallem, "A comprehensive overview of dynamic visual SLAM and deep learning: Concepts, methods and challenges," *Machine Vision and Applications*, vol. 33, no. 4, pp. 54–75, 2022.

[14] T. Lin, M. Maire and S. Belongie, "Microsoft COCO: Common objects in context," in *Lecture Notes in Computer Science*, vol. 8693. Berlin, Germany: Springer, pp. 740–755, 2014.

[15] J. Sturm, N. Engelhard, F. Endres, B. Wolfram and C. Daniel, "A benchmark for the evaluation of RGB-D SLAM systems," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal, pp. 573–580, 2012.

[16] A. Geiger, P. Lenz, C. Stiller and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[17] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *The International Journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.

[18] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey Vision Conf.*, vol. 15, no. 50, pp. 147–152, 1988.

[19] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conf. on Computer Vision*, Beijing, China, pp. 1508–1515, 2006.

[20] J. Shi and Tomasi, "Good features to track," in *1994 Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, USA, pp. 593–600, 1994.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[22] L. Juan and O. Gwun, "A comparison of SIFT, PCA-SIFT and SURF," *International Journal of Image Processing (IJIP)*, vol. 3, no. 4, pp. 143–152, 2009.

[23] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 Int. Conf. on Computer Vision*, Barcelona, Spain, pp. 2564–2571, 2011.

[24] S. Leutenegger, M. Chli and R. Y. Siegwart, "BRISK: Binary Robust invariant scalable keypoints," in *2011 Int. Conf. on Computer Vision*, Barcelona, Spain, pp. 2548–2555, 2011.

[25] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. of Third Int. Conf. on 3-D Digital Imaging and Modeling*, Quebec City, QC, Canada, pp. 145–152, 2001.

[26] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[27] V. Lepetit, F. Moreno-Noguer and P. Fua, "EP*n*P: An accurate *O(n)* solution to the P*n*P problem," *International Journal of Computer Vision*, vol. 81, no. 1, pp. 155–166, 2009.

[28] X. Gao, X. Hou, J. Tang and H. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 930–943, 2003.

[29] X. Wei, Y. Zhang, Z. Li and X. Xue, "DeepSFM: Structure from motion via deep bundle adjustment," in *Computer Vision-ECCV 2020: 16th European Conf.*, Glasgow, UK, pp. 230–247, 2020.

[30] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *2007 6th IEEE and ACM Int. Symp. on Mixed and Augmented Reality*, Nara, Japan, pp. 225–234, 2007.

[31] R. A. Newcombe, S. J. Lovegrove and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *2011 Int. Conf. on Computer Vision*, Barcelona, Spain, pp. 2320–2327, 2011.

[32] J. Engel, T. Schöps and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Computer Vision-ECCV 2014: 13th European Conf.*, Zurich, Switzerland, pp. 834–849, 2014.

[33] C. Forster, M. Pizzoli and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *2014 IEEE Int. Conf. on Robotics and Automation (ICRA)*, Hong Kong, China, pp. 5–22, 2014.

[34] A. Kundu, K. M. Krishna and J. Sivaswamy, "Moving object detection by multi-view geometric techniques from a single camera mounted robot," in *2009 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, St. Louis, MO, USA, pp. 4306–4312, 2009.

[35] K. Lin, X. Liang and J. Cai, "Dynamic RGB-D SLAM algorithm based on reprojection depth difference cumulative map and static probability," *Journal of Zhejiang University (Engineering Science)*, vol. 56, no. 6, pp. 1062–1070, 2022.

[36] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich *et al.,* "Real-time 3D reconstruction in dynamic scenes using point-based fusion," in *2013 Int. Conf. on 3D Vision—3DV 2013*, Seattle, WA, USA, pp. 1–8, 2013.

[37] S. Izadi, D. Kim, O. Hilliges, O. Hilliges, D. Molyneaux *et al.,* "KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera," in *Proc. of the 24th Annual ACM Symp. on User Interface Software and Technology*, New York, NY, USA, pp. 559–568, 2011.

[38] D. Zou and P. Tan, "CoSLAM: Collaborative visual SLAM in dynamic environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 354–366, 2013.

[39] W. Tan, H. Liu, Z. Dong, G. Zhang and H. Bao, "Robust monocular SLAM in dynamic environments," in *2013 IEEE Int. Symp. on Mixed and Augmented Reality (ISMAR)*, Adelaide, SA, Australia, pp. 209–218, 2013.

[40] Y. Sun, M. Liu and M. Meng, "Improving RGB-D SLAM in dynamic environments: A motion removal approach," *Robotics and Autonomous Systems*, vol. 89, no. 1, pp. 110–122, 2017.

[41] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 3, pp. 185–203, 1981.

[42] J. Klappstein, T. Vaudrey, C. Rabe, A. Wedel and R. Klette, "Moving object segmentation using optical flow and depth information," in *Advances in Image and Video Technology: Third Pacific Rim Symp.*, Tokyo, Japan, pp. 611–623, 2009.

[43] C. Kerl, J. Sturm and D. Cremers, "Robust odometry estimation for RGB-D cameras," in *2013 IEEE Int. Conf. on Robotics and Automation*, Karlsruhe, Germany, pp. 3748–3754, 2013.

[44] M. Derome, A. Plyer, M. Sanfourche and G. L. Besnerais, "Moving object detection in real-time using stereo from a mobile platform," *Unmanned Systems*, vol. 3, no. 4, pp. 253–266, 2015.

[45] P. Ochs, J. Malik and T. Brox, "Segmentation of moving objects by long term video analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1187–1200, 2014.

[46] Y. Wang and S. Huang, "Towards dense moving object segmentation based robust dense RGB-D SLAM in dynamic scenarios," in *2014 13th Int. Conf. on Control Automation Robotics & Vision (ICARCV)*, Singapore, pp. 1841–1846, 2014.

[47] T. Zhang, H. Zhang, Y. Li, Y. Nakamura and L. Zhang, "FlowFusion: Dynamic dense RGB-D SLAM based on optical flow," in *2020 IEEE Int. Conf. on Robotics and Automation (ICRA)*, Paris, France, pp. 7322–7328, 2020.

[48] S. A. Taghanaki, K. Abhishek, J. P. Cohen, J. Cohen-Adad and G. Hamarneh, "Deep semantic segmentation of natural and medical images: A review," *Artificial Intelligence Review*, vol. 54, no. 1, pp. 137–178, 2021.

[49] P. O. O. Pinheiro, R. Collobert and P. Dollár, "Learning to segment object candidates," *Advances in Neural Information Processing Systems*, vol. 28, no. 1, pp. 1990–1998, 2015.

[50] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," in *Proc. of the IEEE Int. Conf. on Computer Vision*, New York, NY, USA, pp. 2961–2969, 2017.

[51] S. Ren, K. 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. 39, no. 6, pp. 1137–1149, 2017.

[52] Z. Cai and N. Vasconcelos, "Cascade R-CNN: High quality object detection and instance segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 5, pp. 1483–1498, 2021.

[53] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, New York, NY, USA, pp. 6154–6162, 2018.

[54] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li *et al.,* "Hybrid task cascade for instance segmentation," in *2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, New York, USA, pp. 4969–4978, 2019.

[55] S. Liu, L. Qi, H. Qin, J. Shi and J. Jia, "Path aggregation network for instance segmentation," in *2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, New York, USA, pp. 8759–8768, 2018.

[56] Z. Huang, L. Huang, Y. Gong, C. Huang and X. Wang, "Mask scoring RCNN," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, New York, USA, pp. 6409–6418, 2019.

[57] B. D. Brabandere, D. Neven and L. V. Gool, "Semantic instance segmentation with a discriminative loss function," arXiv:1708.02551, 2017.

[58] A. Fathi, Z. Wojna, V. Rathod, P. Wang and H. Song, "Semantic instance segmentation via deep metric learning," arXiv:1703.10277, 2017.

[59] S. Kong and C. Fowlkes, "Recurrent pixel embedding for instance grouping," in *2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 9018–9028, 2018.

[60] J. Dai, K. He, Y. Li, S. Ren and J. Sun, "Instance-sensitive fully convolutional networks," in *Computer Vision-ECCV 2016: 14th European Conf.*, Amsterdam, The Netherlands, pp. 534–549, 2016.

[61] Y. Li, H. Qi, J. Dai, X. Ji and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *2017 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 4438–4446, 2017.

[62] D. Bolya, C. Zhou, F. Xiao and Y. Lee, "Yolact: Real-time instance segmentation," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision*, New York, USA, pp. 9157–9166, 2019.

[63] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang *et al.,* "BlendMask: Top-down meets bottom-up for instance segmentation," in *2020 IEEE/CVF Conf. on Computer Vision*, New York, USA, pp. 8570–8578, 2020.

[64] H. Ying, Z. Huang, S. Liu, T. Shao and K. Zhou, "Embedmask: Embedding coupling for one-stage instance segmentation," arXiv:1912.01954, 2019.

[65] Z. Tian, C. Shen and H. Chen, "Conditional convolutions for instance segmentation," in *Computer Vision-ECCV 2020: 16th European Conf.*, Glasgow, UK, pp. 282–298, 2020.

[66] D. Bolya, C. Zhou, F. Xiao and Y. Lee, "YOLACT++ better real-time instance segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 1108–1121, 2022.

[67] Z. Tian, C. Shen, H. Chen and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision*, New York, USA, pp. 9627–9636, 2019.

[68] X. Wang, T. Kong, C. Shen, Y. Jing and L. Li, "SOLO: Segmenting objects by locations," in *Computer Vision-ECCV 2020: 16th European Conf.*, Glasgow, UK, pp. 649–665, 2020.

[69] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 779–788, 2016.

[70] P. Arbelaez, J. Pont-Tuset, J. T. Barron, F. Marques and J. Malik, "Multiscale combinatorial grouping," in *2014 IEEE Conf. on Computer Vision and Pattern Recognition*, Columbus, USA, pp. 328–335, 2014.

[71] X. Wang, R. Zhang, T. Kong, L. Li and C. Shen, "SOLOv2: Dynamic and fast instance segmentation," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17721–17732, 2020.

[72] A. Kirillov, Y. Wu, K. He and R. Girshick, "PointRend: Image segmentation as rendering," in *2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, pp. 9796–9805, 2020.

[73] Y. Fang, S. Yang, X. Wang, Y. Li, C. Fang *et al.,* "QueryInst: Parallelly supervised mask query for instance segmentation," arXiv:2105.01928, 2021.

[74] X. Zhou, J. Zhuo and P. Krähenbühl, "Bottom-up object detection by grouping extreme and center points," in *2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, pp. 850–859, 2019.

[75] W. Xu, H. Wang, F. Qi and C. Lu, "Explicit shape encoding for real-time instance segmentation," in *2019 IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Seoul, Korea (South), pp. 5167–5176, 2019.

[76] E. Xie, P. Sun, X. Song, W. Wang, X. Liu *et al.,* "PolarMask: Single shot instance segmentation with polar representation," in *2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, pp. 12190–12199, 2020.

[77] H. U. M. Riaz, N. Benbarka and A. Zell, "FourierNet: Compact mask representation for instance segmentation using differentiable shape decoders," in *2020 25th Int. Conf. on Pattern Recognition (ICPR)*, Milan, Italy, pp. 7833–7840, 2021.

[78] P. Hurtik, V. Molek, J. Hula, M. Vajgl, P. Vlasanek *et al.,* "Poly-YOLO: Higher speed, more precise detection and instance segmentation for YOLOv3," *Neural Computing and Applications*, vol. 34, no. 10, pp. 8275–8290, 2022.

[79] X. Wang, K. Zhao, R. Zhang, S. Ding, Y. Wang *et al.,* "ContrastMask: Contrastive learning to segment every thing," in *2022 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, pp. 11594–11603, 2022.

[80] T. Zhang, S. Wei and S. Ji, "E2EC: An end-to-end contour-based method for high-quality high-speed instance segmentation," in *2022 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, pp. 4433–4442, 2022.

[81] N. Carion, F. Massa and G. Synnaeve, "End-to-end object detection with transformers," in *European Conf. on Computer Vision*, Cham, Springer International Publishing, pp. 213–229, 2020.

[82] M. Jaderberg, K. Simonyan and A. Zisserman, "Spatial transformer networks," arXiv:1506.02025, 2015.

[83] B. Cheng, A. Schwing and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," *Advances in Neural Information Processing Systems*, vol. 43, pp. 17864–17875, 2021.

[84] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *2022 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, pp. 1280–1289, 2022.

[85] A. Kirillov, E. Mintun and N. Ravi, "Segment anything," arXiv:2304.02643, 2023.

[86] Y. Li, H. Mao and R. Girshick, "Exploring plain vision transformer backbones for object detection," in *European Conf. on Computer Vision*, Cham, Springer Nature Switzerland, pp. 280–296, 2022.

[87] C. Yu, Z. Liu, X. Liu, F. Xie, Y. Yang *et al.,* "DS-SLAM: A semantic visual SLAM towards dynamic environments," in *2018 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Madrid, Spain, pp. 1168–1174, 2018.

[88] V. Badrinarayanan, A. Kendall and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[89] B. Bescos, J. M. Facil, J. Civera and J. Neira, "DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4076–4083, 2018.

[90] J. Li, R. Zhang, Y. Liu, Z. Zhang, R. Fan *et al.,* "The method of static semantic map construction based on instance segmentation and dynamic point elimination," *Electronics*, vol. 10, no. 16, pp. 1883–1895, 2021.

[91] Y. Liu and J. Miura, "RDS-SLAM: Real-time dynamic SLAM using semantic segmentation methods," *IEEE Access*, vol. 9, pp. 23772–23785, 2021.

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

[93] J. Zhang, M. Henein, R. Mahony and V. Ila, "VDO-SLAM: A visual dynamic object-aware SLAM system," arXiv:2005.11052, 2020.

[94] J. Chang, N. Dong and D. Li, "A real-time dynamic object segmentation framework for SLAM system in dynamic scenes," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–9, 2021.

[95] X. Chen, Y. Zhou and L. Chen, "A visual SLAM approach based on optical flow and instance segmentation in dynamic scenes," *Journal of Optics*, vol. 42, no. 14, pp. 147–159, 2022.

[96] S. Zhao, Y. Sheng, Y. Dong, E. I. C. Chang and Y. Xu, "MaskFlownet: Asymmetric feature matching with learnable occlusion mask," in *2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, pp. 6277–6286, 2020.

[97] H. Wang, D. Lu and B. Fang, "An enhanced segmentation-based RGB-D SLAM method for dynamic environments," *Robotics*, vol. 44, no. 4, pp. 418–430, 2022.

[98] G. J. Shi, B. K. Gao and L. Zhang, "The optimized K-means algorithms for improving randomly-initialed midpoints," in *Proc. of 2013 2nd Int. Conf. on Measurement, Information and Control*, Harbin, China, pp. 1212–1216, 2013.

[99] Y. Zhang and F. Zhang, "VSLAM based on instance segmentation," in *2020 5th Int. Conf. on Mechanical, Control and Computer Engineering (ICMCCE)*, Harbin, China, pp. 2072–2075, 2020.

[100] L. Sun, J. Wei, S. Su and P. Wu, "SOLO-SLAM: A parallel semantic SLAM algorithm for dynamic scenes," *Sensors*, vol. 22, no. 18, pp. 6977–7002, 2022.

[101] A. G. Howard, M. L. Zhu, B. Chen, D. Kalenichenko, W. J. Wang *et al.,* "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv:1704.04861, 2017.

[102] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Int. Conf. on Machine Learning*, Long Beach, California, USA, pp. 6105–6114, 2019.

[103] X. Chen, R. Girshick, K. He and P. Dollar, "TensorMask: A foundation for dense object segmentation," in *2019 IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Seoul, Korea (South), pp. 2061–2069, 2019.