Mech. Sci., 15, 1–16, 2024
https://doi.org/10.5194/ms-15-1-2024

Mechanical
Sciences
Open Access

Visual simultaneous localization and mapping (vSLAM) algorithm based on improved Vision Transformer semantic segmentation in dynamic scenes

Mengyuan Chen[1,2], **Hangrong Guo**[1], **Runbang Qian**[1], **Guangqiang Gong**[1], **and Hao Cheng**[1]

[1]School of Electrical Engineering, Anhui Polytechnic University, Wuhu, China
[2]Key Laboratory of Advanced Perception and Intelligent Control of High-end Equipment,
Ministry of Education, Anhui, China

Correspondence: Mengyuan Chen (mychen@ahpu.edu.cn)

Received: 16 May 2023 – Revised: 27 September 2023 – Accepted: 16 October 2023 – Published: 3 January 2024

Abstract. Identifying dynamic objects in dynamic scenes remains a challenge for traditional simultaneous localization and mapping (SLAM) algorithms. Additionally, these algorithms are not able to adequately inpaint the culling regions that result from excluding dynamic objects. In light of these challenges, this study proposes a novel visual SLAM (vSLAM) algorithm based on improved Vision Transformer semantic segmentation in dynamic scenes (VTD-SLAM), which leverages an improved Vision Transformer semantic segmentation technique to address these limitations. Specifically, VTD-SLAM utilizes a residual dual-pyramid backbone network to extract dynamic object region features and a multiclass feature transformer segmentation module to enhance the pixel weight of potential dynamic objects and to improve global semantic information for precise identification of potential dynamic objects. The method of multi-view geometry is applied to judge and remove the dynamic objects. Meanwhile, according to static information in the adjacent frames, the optimal nearest-neighbor pixel-matching method is applied to restore the static background, where the feature points are extracted for pose estimation. With validation in the public dataset TUM (The Entrepreneurial University Dataset) and real scenarios, the experimental results show that the root-mean-square error of the algorithm is reduced by 17.1 % compared with dynamic SLAM (DynaSLAM), which shows better map composition capability.

1 Introduction

Simultaneous localization and mapping (SLAM) is a mobile robot equipped with specialized sensors that enable it to model the environment while in motion, without any prior information about the environment. At the same time, it estimates its own position (Chen et al., 2022; Gao et al., 2021; Zhou et al., 2021). Depending on the sensors used by the robot, SLAM can be divided into two main categories: visual SLAM (vSLAM) and laser SLAM. However, the disadvantages of laser sensors, such as the limited amount of acquired information and the bulky size, have restricted widespread application of laser SLAM. As a result, vSLAM, which is smaller, less expensive, and capable of acquiring richer information, has become an increasingly popular research direction in the field of SLAM (Cao et al., 2021; Huang et al., 2020).

The more mature vSLAM systems are Oriented FAST and Rotated BRIEF SLAM (ORB-SLAM) and Point and Line SLAM (PL-SLAM) (Mur and Tardós, 2017; Pumarola et al., 2017). Most of the current mainstream vSLAM systems achieve high-accuracy localization and composition in static environments. When moving objects are in the environment, it is difficult for the system to perform accurate positional estimation and composition (Xu et al., 2022). To address the problem of dynamic objects in the environment, Sun et al. (2018) proposed an algorithm based on depth cameras to remove dynamic objects, which uses a hypothetical static depth map to detect planar regions and further refines the motion regions by combining the detection results with the coarse-motion regions obtained from re-

Published by Copernicus Publications.

projection errors. With the introduction of deep learning in SLAM, Mask-SLAM proposed by Kaneko et al. (2018) segmented the image using the DeepLabV2 algorithm, assigned a corresponding semantic label to each like the Kaneko element, and removed the a priori dynamic objects from it based on the acquired semantic information. However, the method did not consider the moved objects (e.g., chairs, books). Zhong et al. (2018) proposed a Detect-SLAM algorithm to propagate the motion probability of target points, detect semantic targets, update the probability value at the bit pose estimation, and retain static points below the threshold to improve the recognition of dynamic objects further. Bescos et al. (2018) employed a convolutional neural network that combines MASK-RCNN and multi-view geometry to segment potential dynamic objects (DynaSLAM). The algorithm projects both RGB and depth maps from previous key frames onto the current frame for background restoration without dynamic objects. Although the approach can effectively identify and reject dynamic objects, it is susceptible to noise. Yu et al. (2018) proposed a robust semantic vSLAM for dynamic environments (DS-SLAM), which combines the SegNet semantic segmentation network with a motion-consistency-checking method to reduce the impact of dynamic targets. Liu and Miura (2021) proposed a real-time dynamic SLAM using semantic segmentation methods (RDS-SLAM) based on ORB-SLAM3 (Campos et al., 2021), which added semantic threads and semantic-based optimization threads to achieve real-time robust tracking and mapping in dynamic environments. However, this algorithm is limited by the semantic segmentation capability, which is prone to failure due to incomplete segmentation.

In summary, existing SLAM algorithms have problems such as the existence of dynamic objects in the environment, the inability of the semantic segmentation network to accurately segment dynamic objects, and the inability to effectively repair the culling regions after the removal of dynamic objects. In this paper, we propose a VTD-SLAM algorithm, and the semantic segmentation network in this paper achieves the segmentation of potential dynamic objects by fusing multiclass feature enhancement and multiclass feature guidance (Vaswani et al., 2017; Dosovitskiy et al., 2020; Liu et al., 2021; Carion et al., 2020; Strudel et al., 2021). Adaptive geometric thresholding is used to judge the motion of dynamic objects in it (An et al., 2021; Guan et al., 2010). Static background restoration is performed on the removal region, and feature points are extracted from this region (Zhao and Lv, 2023). The algorithm of this paper is verified in the public dataset TUM (The Entrepreneurial University Dataset) and real scenes. The experimental results show that the algorithm in this paper has a greater advantage in dynamic object segmentation and static restoration compared with ORB-SLAM2, DS-SLAM, and DynaSLAM algorithms and shows good bit pose estimation and composition capability.

2 System frame

This paper introduces a novel VTD-SLAM algorithm that enhances the semantic segmentation of Vision Transformer to overcome the limitations of traditional SLAM algorithms in accurately identifying dynamic objects in dynamic scenes and efficiently repairing the culling regions (Han et al., 2022). The algorithm consists of two parts: potential dynamic object segmentation and background restoration. The potential dynamic object segmentation segment is performed by the MSNET (Multivariate feature fusion and multivariate feature estimation Semantic Segmentation Network) proposed in this paper to semantically segment the objects present in the image information and to extract the image feature points. In the dynamic object culling and background restoration phase, dynamic points are identified, and dynamic objects are excluded through adaptive geometry thresholding (Feng et al., 2021). The algorithm then utilizes the optimal nearest-neighbor pixel-matching (PatchMatch) approach for conducting similar searches, pixel matching, and feature point extraction for the region that contains the removed dynamic objects (Barnes et al., 2009). This process provides rich information for the SLAM system to perform accurate localization and map building. The frame of the VTD-SLAM algorithm is shown in Fig. 1.

3 Potential dynamic object segmentation

In order to solve the problem of blurred segmentation or segmentation errors caused by fast-moving objects in traditional semantic segmentation algorithms for dynamic object segmentation, this paper proposes a multiclass feature enhancement and multiclass feature-guided semantic segmentation network (MSNET), which contains a backbone network module for multiple extractions of dynamic object region features and the multiclass feature Transformer segmentation module (Waswani et al., 2017). The structure diagram of MSNET proposed in this paper is shown in Fig. 2.

3.1 Backbone network module

The Transformer network architecture has gained significant attention in the field of computer vision due to its exceptional performance in global feature extraction and long-range pixel relationship acquisition. As the Vision Transformer (VIT) Base 16 network directly splits the original image into small images, it will lead to the noise factors in the original image being preserved intact, such as the darker part of the image and the blurring problem caused by the moving objects, which will cause the kind of information of the generated feature maps to be weak and have poor anti-interference properties, so the backbone network module designed in this paper adds a sampling pyramid to the VIT to construct a residual dual-pyramid network (Targ et al., 2016). The scale invariance of different dimensions of the image is enhanced by

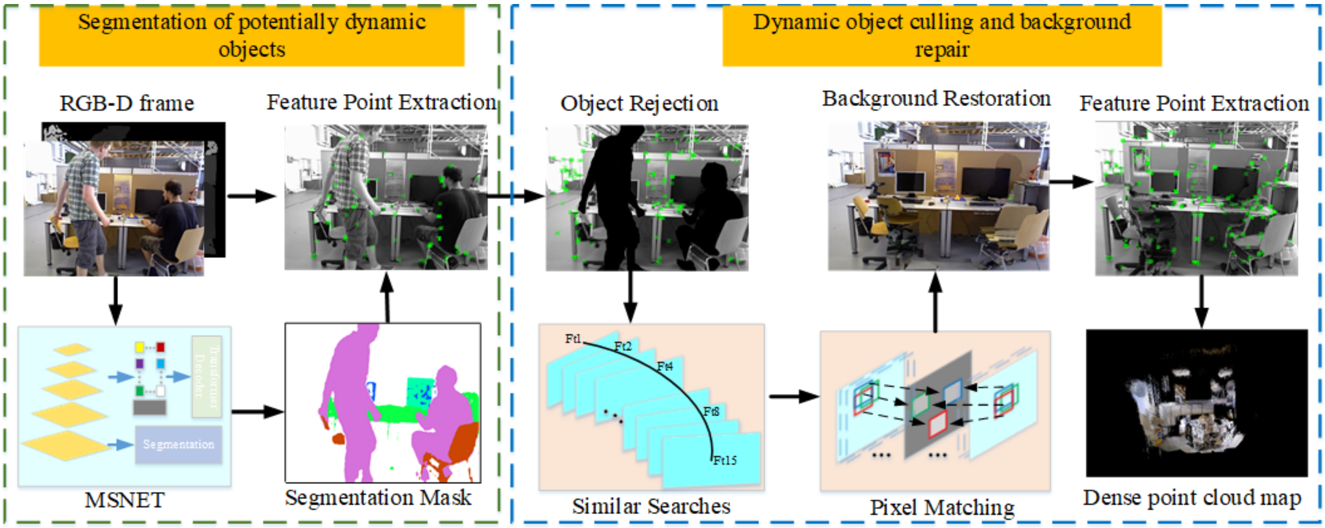


Figure 1. The framework of the VTD-SLAM algorithm.

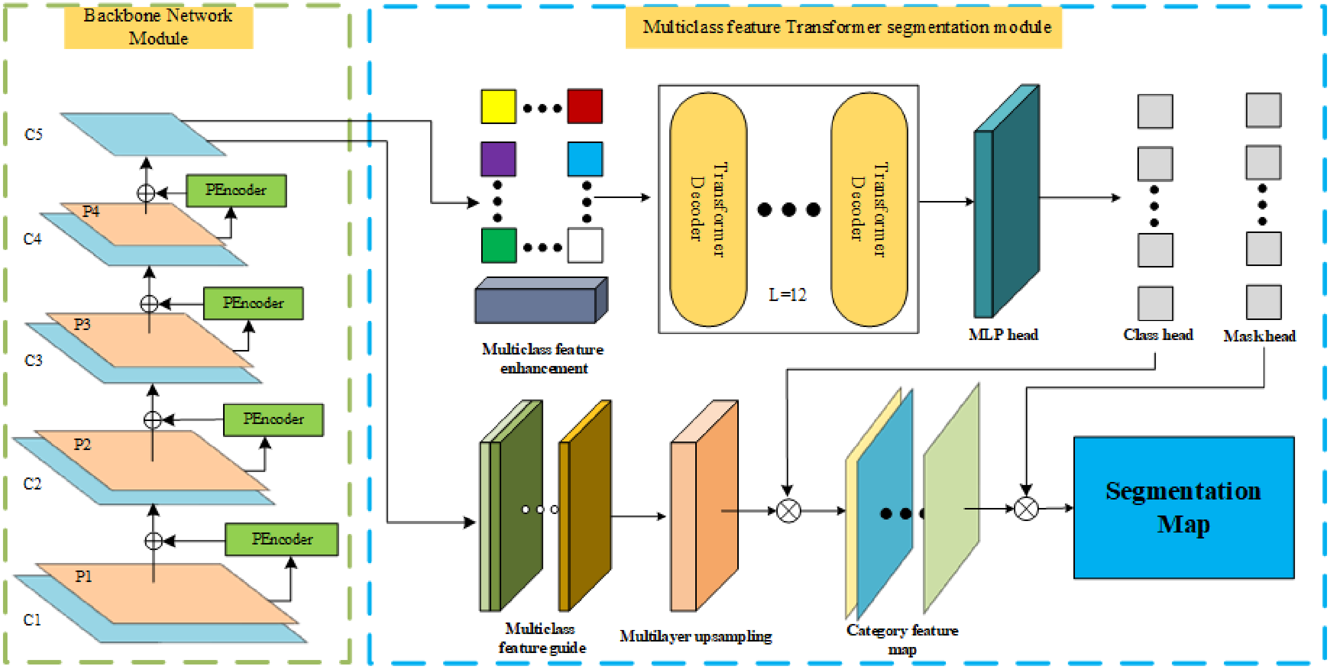


Figure 2. MSNET split network structure.

downsampling the original image C1 to P1, which helps us to extract the most useful features and reduce the influence of noise and, at the same time, in order to avoid the overfitting of the model, P1 is subjected to PEncoder, and the feature map and downsampled feature maps after the PEncoder module are concatenated and fused to obtain the initial map of the next pyramid, C2. Although C2 reduces the influence of noise, it lacks robustness because it has only one weight of feature information and is affected by dynamic objects. Additionally, to extract multiple features more effectively from the image and to enhance robustness, the feature maps C2, C3, and C4 with rich feature and boundary information are sampled to the next level of the feature pyramid dimension, producing the downsampled feature maps P2, P3, and P4. The feature maps and downsampled feature maps after the PEncoder module are concatenated and fused, resulting in the multi-feature map C5 with rich information and a strong anti-interference ability.

The proposed PEncoder module is shown in Fig. 3. The role of the PEncoder module is to extract global image feature information. The algorithm inputs the feature maps (C1, C2, C3, and C4) into the PEncoder module, performs global patch embedding and position embedding on the feature maps, and inputs them into the MSA (multi-head self-attention) and DML (dropout, multilayer perceptron and layer normalization) layers in turn to obtain the global-feature linkage information on the feature maps. The feature maps obtained by the PEncoder module are calculated as shown in Eqs. (1), (2), and (3):

$$\mathrm{MSA}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{D}}\right)V, \tag{1}$$

$$a_{i-1} = \mathrm{MSA}(\mathrm{LN}(b_{i-1})) + b_{i-1}, \tag{2}$$

$$b_i = \mathrm{MLP}(\mathrm{LN}(a_{i-1})) + a_{i-1}, \tag{3}$$

where Q denotes the query vector, K denotes the key vector, V denotes the value vector, and D denotes the dimension, $i \in \{1, \ldots, L\}$; b_{i-1} is the input tokens, a_{i-1} is the tokens obtained after the MSA layer, and b_i is the tokens obtained after the PEncoder module; LN is the layer normalization, and MLP is the multilayer perceptron.

3.2 Multiclass feature Transformer segmentation module

The visual SLAM system is vulnerable to the interference of dynamic objects in operation, and failure to accurately identify dynamic objects among them will interfere with the robustness of the SLAM system. The multiclass feature Transformer segmentation module designed in this paper consists of two subnetworks, multiclass feature enhancement, and multiclass feature guidance. Firstly, multiclass feature enhancement is performed on feature map C5, and the multi-feature map C5 with multi-feature information is input into the Multi-Class Feature Enhancement Module, filtered by m filters with single-class potential dynamic feature information to get the m single-class potential dynamic feature signatures, fused together, and finally fused into the multiclass potential dynamic feature information, which in turn enhances the potential dynamic object pixel weights and inputs the fused features into the Transformer decoder and then the MLP head to get the Class head and the Mask head. At the same time, multiclass feature guidance is performed on feature map C5, feature map C5 is input into the multiclass feature guidance module, the attention mechanism of the channel dimension and spatial dimension is operated on C5, and average pooling and maximum pooling are performed on the feature information of each dimension to form a feature map with global semantic information, respectively, and feature maps are fused to obtain feature maps with dense global semantic information, which can in turn improve the global semantic information. The dense global semantic information feature map is input into the multilayer upsampling and is multiplied by the Class head to output the category feature map, and then the category feature map is multiplied by the Mask head to get the segmentation map. The multiclass feature Transformer segmentation module leverages a combination of multiclass feature enhancement and guidance to generate pixel weights that incorporate strong class feature information and improve the extraction of feature information in the region of interest. This approach enhances the pixel weights of moving objects and improves global semantic information while mitigating the impact of noise interference (Tian et al., 2023).

The multiclass feature enhancement module in this paper is shown in Fig. 4. The multiclass feature enhancement module serves to improve single-class feature information and to remove redundant feature information. This algorithm designs M filters $\{f_1, \ldots, f_{\mathrm{M}}\}$ pre-trained with a single class of feature information to filter feature map C5, thus removing irrelevant and overlapping feature information. Specifically, M filters are feature maps with 60 common potential dynamic object feature information, and M filters are used to convolve feature map C5 to obtain M feature maps $\{t_1, \ldots, t_{\mathrm{M}}\}$ with obvious potential dynamic feature information and fuse them to further obtain a feature information map $\mathbf{F}_{\mathrm{f}}$ that retains a large number of potential dynamic features, thus removing a large amount of other irrelevant feature information, removing the interference of a large amount of other irrelevant feature information, and enhancing the potential dynamic feature information in the image. The feature map with strong single-class features is generated by the filter, and the fused feature map is obtained by concatenation fusion, which has the advantages of rich single-class information and strong anti-interference ability. The filter decision method designed in this paper is shown in Eq. (4):

$$\begin{cases} y \to \omega_j, \\ F(\omega j) = \max\limits_{k=1,\ldots,c} P(\omega k | y_1, \ldots, y_M), \end{cases} \tag{4}$$

where y denotes the feature information of feature map C5, ω_j denotes the j class of features, k is for class, $P(\omega k|y)$ denotes the posterior probability of the k class, and $F(\omega_j)$ is the j class of feature information.

The multiclass feature guidance module used in this paper is shown in Fig. 5 (Woo et al., 2018). The multiclass feature guidance component is designed to increase the channel weights of the region of interest, determine its spatial location, and direct the neural network to focus on the feature information of the dynamic region. Firstly, attention operations are performed on feature map C5 in the channel dimension and the spatial dimension, respectively, and then the obtained feature map $\mathbf{F}_{\mathrm{c}}$ is concatenated with the feature map $\mathbf{F}_{\mathrm{s}}$ to obtain the output $\mathbf{F}_{\mathrm{e}}$. Benefit from the ability of the attention mechanism to perceive global semantic information so that the generated $\mathbf{F}_{\mathrm{e}}$ is a feature map with dense global semantic information. On the one hand, the channel attention

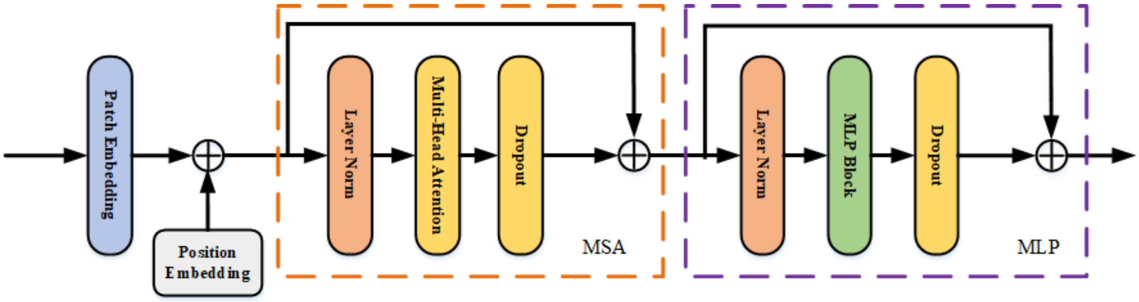


Figure 3. PEncoder module.

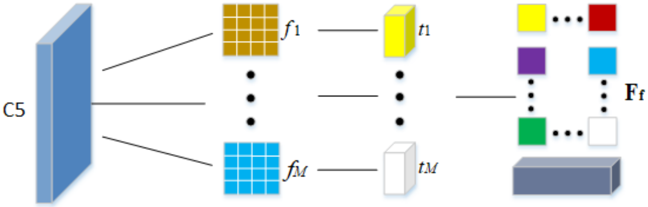


Figure 4. Multiclass feature enhancement module.

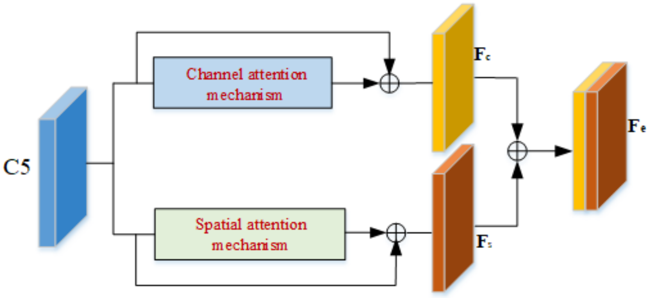


Figure 5. Multiclass feature bootstrap module.

mechanism assigns weights to different channel features in the motivation part, and the focus of our work is the recognition of potential dynamic objects, so we use the sigmoid function as the activation function when assigning weights to strengthen the weights of the potential dynamic feature information; on the other hand, the spatial attention mechanism can automatically capture the important areas in the image, and the combination of the spatial and channel attention mechanisms makes the multiclass feature guidance module pay more attention to the important areas in the image when it works, so that the multiclass feature guidance module pays more attention to the potential dynamic feature information in the important areas of the image and then guides the model to pay attention to the area where the potential dynamic object features are located.

The attention mechanism is shown in Fig. 6. The channel attention mechanism acts to assign the corresponding weights to each layer of channels on the feature map, the average pooling and maximum pooling operations are performed on the feature map to obtain information about each channel of the feature map, the features $\mathbf{F}_{\mathrm{avg}}$ and $\mathbf{F}_{\mathrm{max}}$ are obtained by the average pooling and maximum pooling, and then the features are concatenated and fused to obtain the feature information with enhanced channel correlation $\mathbf{f}_{\mathrm{v}}$ (Hu et al., 2018). The spatial attention mechanism serves to capture important areas of the image. The feature map is input into the spatial attention mechanism, and the concatenation fusion is performed to form the feature map $\mathbf{f}_{\mathrm{c}}$ through average pooling and maximum pooling, and then the $3 \times 3 \times 1$ convolutional layer and the sigmoid function are used to obtain the weight assignment feature map $\mathbf{f}_{\mathrm{u}}$, which leads the network to focus on the regions where dynamic objects are located. The outputs $\mathbf{f}_{\mathrm{v}}$ and $\mathbf{f}_{\mathrm{u}}$ obtained by the channel attention mechanism and the spatial attention mechanism are calculated as shown in Eqs. (5) and (6).

$$\mathbf{f}_{\mathrm{v}} = \sigma\left(\eta\left(\mathbf{F}_{\mathrm{avg}} + \mathbf{F}_{\mathrm{max}}\right)\right) \tag{5}$$

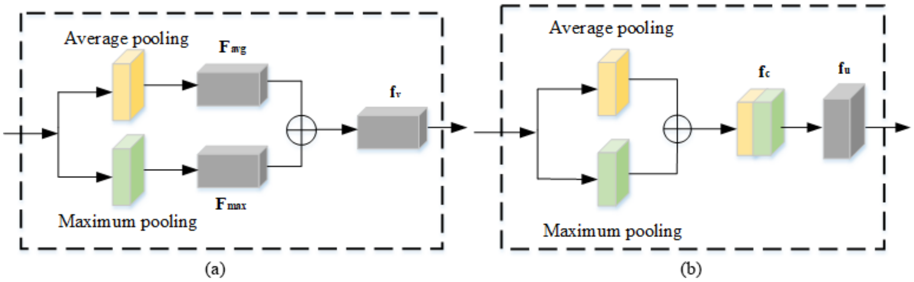
$$\mathbf{f}_{\mathrm{u}} = \sigma\left(c(\mathbf{f}_{\mathrm{c}})\right) \tag{6}$$

σ denotes the sigmoid function, η is the tanh function, and c is the $3 \times 3 \times 1$ convolutional layer.

4 Dynamic object culling and background restoration

4.1 Dynamic object culling

In the potential dynamic object segmentation environment, using a semantic segmentation network can only obtain the potential dynamic objects in the image but cannot specifically determine the dynamic objects (Liu et al., 2023). To address this problem, this paper proposes the use of adaptive geometric thresholding for dynamic object detection. Firstly, the image frames are extracted with ORB features and matched with neighboring feature points to obtain n matching pairs, and then the basis matrix $\mathbf{F}$ is obtained by the eight-point method. Then the fundamental matrix is used



(a)

(b)

Figure 6. Attention mechanism.

to calculate the polar lines of the current frame. In a concrete approach, the pixel position of the feature point in the previous framework is known as $L_1 = [x1, y1, 1]$, and the corresponding position of this point in the current frame is $L_2 = [x2, y2, 1]$. Then the polar line I_1 projected by the point L_1 into the current frame can be found, and its calculation formula is shown in Eq. (7).

$$I_1 = \begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \\ \boldsymbol{Z} \end{pmatrix} = \mathbf{F} L_1 = \mathbf{F} \begin{pmatrix} x1 \\ y1 \\ 1 \end{pmatrix} \tag{7}$$

$\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$ represent the line vector in the foundation matrix. Ultimately, the distance d of the matched feature point to its corresponding polar line is calculated, and if this distance exceeds a threshold, it is considered a moving point and vice versa a static point. Its calculation formula is shown in Eq. (8).

$$d = \left(\|\boldsymbol{X}\|^2 + \|\boldsymbol{Y}\|^2 \right)^{-\frac{1}{2}} \left| L_2^{\mathrm{T}} L_1 \right| \tag{8}$$

Due to the influence of noise in the feature-tracking process, the dynamic points cannot be judged effectively by the basis matrix only, so this paper proposes an adaptive method in the nonlinear pose optimization stage. Using the Gauss–Newton iteration method can get the uncertainty error of object motion estimation as $\sum x = \Sigma \left(J^{\mathrm{T}} J \right)$ sets the uncertainty error to satisfy the ω-dimensional Gaussian distribution, and then its differential motion entropy $F(x_{\mathrm{o}})$ is calculated as shown in Eq. (9).

$$F\left(x_{\mathrm{o}}\right) = \frac{1}{2} \log \left| \sum x_o \right| + \frac{\omega}{2} \log(\omega \pi e) \tag{9}$$

Differential motion entropy can be considered the level of pose uncertainty obtained by minimizing photometric residuals. Specifically, a three-dimensional motion observation with high entropy will result in a larger shift of an object in the image, while an observation with low entropy will produce a smaller image discrepancy. Based on this, the object's dynamic deviation is compared to a dynamic threshold $\Delta d = f\left(F(x_{\mathrm{o}})\right)$ guided by the differential motion entropy and slowly increases with entropy, and, if $d > \Delta d$, the object is judged to be dynamic and removes the feature points on this object.

4.2 Background restoration

Conventional dynamic SLAM algorithms often suffer from reduced feature information for background restoration and pose estimation after the removal of dynamic objects. This limitation results in a low number of extracted feature points and inaccurate pose estimation, which can negatively impact loop-back detection and static map construction (Wang et al., 2023). As the quality of image restoration heavily relies on the richness of image information, it can be challenging to efficiently obtain adequate image information by solely projecting previous frames onto the target frame. Therefore, this paper proposes a background restoration algorithm based on PatchMatch to complete the restoration of static backgrounds of RGB images and depth images in the current frame after dynamic region rejection with the help of static information in adjacent frames. The background restoration algorithm of optimal nearest-neighbor pixel matching includes similarity search and pixel matching. Among them, a similar search contains two-step information estimation and reference framework selection. Firstly, we find the key frames to be selected through information estimation, get the judgment threshold displacement and rotation change amount through the motion connection between adjacent frames, get the reference frames through the judgment thresholds, and finally use the reference frames combined with the approximate nearest-neighbor matching algorithm for the background restoration. Its specific steps are as follows.

4.2.1 Similarity search step 1: information estimation

Specifically, a 15-frame image frame is moved in the direction of an arrow, starting from the target frame, to serve as the starting point for the restoration window. The ORB features are extracted from each frame in the window, and the number of feature points is $\sum M_i$. Each reference frame to be selected is matched with the target frame, and the number of feature points matched is recorded as $\sum N_i$. If $\sum M_i$ and $\sum N_i$ satisfy the condition of Eq. (10), the reference frame to be selected is considered to have rich image information.

$$\left| \sum M_i - \sum N_i \right| < 60 \tag{10}$$

Also, by estimating motion links between neighboring frames, $\boldsymbol{P} = [X, Y, Z]^{\mathrm{T}}$ is a point in space with projections p_1 and p_2 in the reference and target frames, respectively, and the pixel positions of pixel points p_1 and p_2 are obtained from the pinhole camera model as

$$\begin{aligned} s_1 p_1 &= \mathbf{K}\boldsymbol{P}, \\ s_2 p_2 &= \mathbf{K}(\mathbf{R}_{21}\boldsymbol{P} + \boldsymbol{t}_{21}). \end{aligned} \tag{11}$$

s_1 and s_2 are the scale factors, $\mathbf{K}$ is the camera internal reference matrix, and $\mathbf{R}_{21}$ and $\mathbf{t}_{21}$ are the rotation and translation matrices of the target map with respect to the reference map, which can be obtained by solving the base matrix or the essential matrix to recover the camera motion and then projecting the three-dimensional vector onto the two-dimensional plane to derive the displacement change Δp and rotation change $\Delta\theta$ between any two frames.

4.2.2 Similarity search step 2: reference frame selection

The use of a suitable reference frame is crucial for performing approximate nearest-neighbor matching. Hence, the process of combining image information and the bit pose is executed as follows: in step (1), a group of forward and reverse 15 frames is selected for feature extraction and matching. The reference frame that meets the criteria is then identified and added to the reference frame library for future selection. When Δp exceeds the threshold τ and $\Delta\theta$ is greater than the threshold γ, this frame will be added to the reference frame library to be selected when the conditions of step 1 are met. When the condition of Eq. (12) is satisfied, the reference frame is added to the reference frame library. The mathematical expression of the reference frame selection method is shown in Eq. (12).

$$\mathrm{rf} = \left\{ f_i \,|\, \left(\left[\Delta p > \tau\right] \cup \left[\Delta\theta > \gamma\right] \cup \left[|M_i - N_i| < 100\right] \right) \right\} \tag{12}$$

4.2.3 Pixel matching

The reference frame library is obtained by a similar search, and pixel matching is performed by improving the traditional approximate nearest-neighbor matching algorithm for background restoration, which consists of three processes: initialization, propagation, and search. Firstly, the target image to be restored is the A image and the reference frame is the B image, a pixel block in the A image is randomly selected as a matching block and randomly assigned an offset, and a matching block is found in the B image to correspond to it. Secondly, the propagation step calculates the offset difference between the matching block in the A image and the matching block in the B image and finds the value with the smallest offset. Thirdly, the search step finds a better match for each pixel point in the B image within a concentric circle centered on the present, and the radius of the search starts at the size of the image and is iteratively reduced by a factor of 0.5 until the end. Propagation and search are repeated until the most suitable and accurate offset is found for each pixel block, and the pixel value corresponding to this offset is assigned to the corresponding pixel block in the A image. The offset calculation formula is shown in Eq. (13).

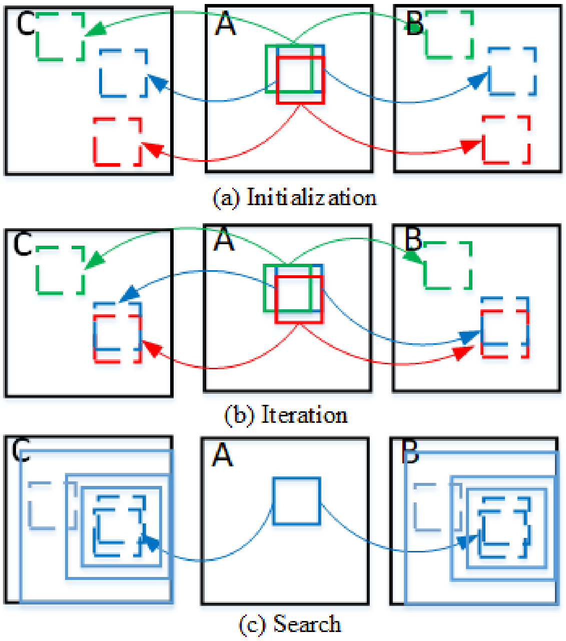


Figure 7. Preferred approximate nearest match.

$$R = \frac{\sum(\mathrm{A}-\mathrm{B})^2 + \sum\left(\overline{\mathrm{A}}-\overline{\mathrm{B}}\right)^2}{n^2} \tag{13}$$

A, B, $\overline{\mathrm{A}}$, $\overline{\mathrm{B}}$, and n^2 are the size of the original matrix block in the A image, the original matrix block in the B image, the offset matrix block in the A image, and the offset matrix block in the B image, respectively. Experimental results indicate that the reference frame library typically contains two to four image frames, with the typical number of iterations being three. Taking two frames of images as an example, the schematic diagram of optimal nearest-neighbor pixel matching is shown in Fig. 7.

5 Experimental results and analysis

This is the hardware and software configuration of the platform used for the experiments in this paper: the CPU is an Inter i9-12900K processor at 3.2 GHZ, the memory is 16 GB, the GPU is an RTX3090 graphics card with 24 GB video memory, and the system is Ubuntu 18.04.

5.1 Potential dynamic object segmentation experiments

This paper uses the COCO2017 (Lin et al., 2014) dataset as training, the experimental software environment is Python 3.8.6, pytorch 1.11.1, and cuda 11.3, the compiler is pycharm 2020.1, the model training parameters are learn rate = 0.001, epochs = 64, and batch size = 4, and the dropout probability is 0.56. In order to verify the semantic segmentation performance of the proposed MSNET algorithm, the desired scenarios are selected in the TUM dataset. The results of the semantic segmentation of W_xyz, W_rpy, W_halfsphere, and W_static sequences by a fully convolutional network (FCN), DeepLabV3Plus, and MSNET in the TUM dataset are compared (Schubert et al., 2018; Dai et al., 2016; Chen et al., 2018). For example, Fig. 8 shows that the green boxes are the auxiliary marks made in this paper. Figure 8a–l illustrate the segmentation results of the FCN, DeepLabV3Plus, and the proposed algorithm, respectively, for four different sequences from the TUM dataset. A comparison of the semantic segmentation performance across the four figures reveals that the FCN and DeepLabV3Plus, which utilize convolutional network structures, are prone to being influenced by the size of the convolutional kernel and the perceptual field. This limitation renders the convolutional network effective only for local features and less accurate in segmenting human torsos. The traditional FCN algorithm uses a fully convolutional network structure, which is very prone to segmentation errors for category-rich images, as shown in Fig. 8b and c. The proposed MSNET algorithm leverages the improved Vision Transformer backbone structure to directly extract feature textures from images. Instead of relying solely on convolutional networks, MSNET adopts a combination of the Transformer decoder, the multiclass feature enhancement module, and multiclass feature guidance for pixel-level segmentation. As Transformer networks can obtain global interpixel feature texture dependencies and improve the accuracy of object recognition and segmentation, MSNET is capable of performing effective and accurate segmentation.

Figure 9 shows the comparison of the evaluation metrics of the three semantic segmentation algorithms, Fig. 9a shows the mean intersection ratio (MIoU) of the three algorithms, and Fig. 9b shows the mean pixel accuracy (MPA) of the categories of the three algorithms. It is one of the most commonly used criteria to evaluate the effectiveness of semantic segmentation, and the larger the value, the better the segmentation effect. The MSNET algorithm proposed in this paper achieves an average intersection ratio of more than 80 % for all four sequences and an average pixel accuracy of more than 70 % for all the categories, which is higher than the other two algorithms.

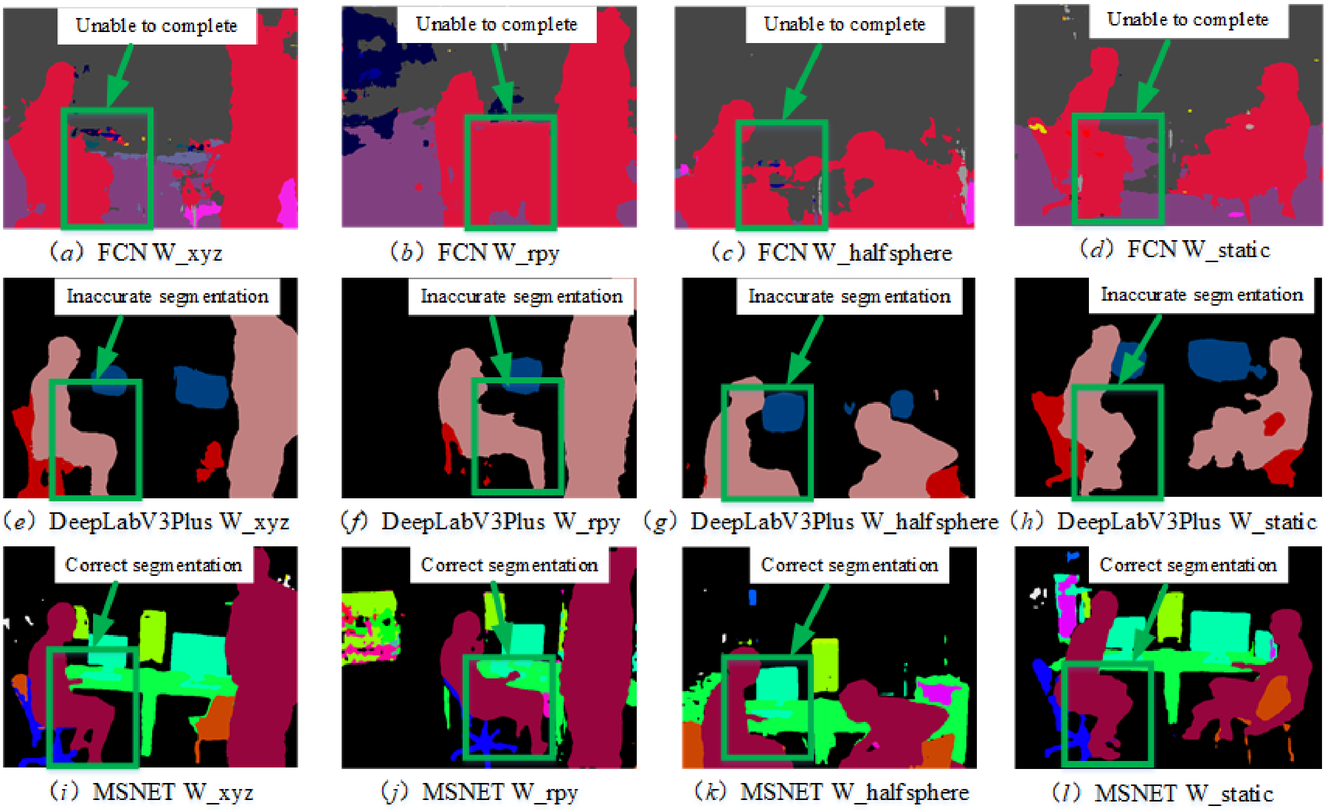
5.2 Dynamic object culling and background restoration

The dynamic objects are judged by multi-view geometry, and the dynamic objects in the image are rejected. In order to get the complete static scene for pose estimation and composition, the optimal nearest-neighbor pixel matching is used to complement the rejected area in this paper. The effect of the background restoration process is shown in Fig. 10, where Fig. 10a–c show the system input containing the original RGB map and the original depth map of the dynamic figure. Figure 10d–f show the RGB map and depth map of dynamic object rejection. The proposed algorithm in this paper utilizes adaptive geometric thresholding to reject dynamic objects and perform restoration of missing image areas through optimal nearest-neighbor pixel matching. This approach results in the creation of a complete, static image. Figure 10g–i show the RGB maps and depth maps after background restoration, and the restored images contain only the static background in the original scene.

In order to further verify the performance of the background restoration algorithm, four dynamic sequences in the TUM dataset are selected to evaluate the localization and mapping capabilities of the algorithm. Compared with ORB-SLAM2, DynaSLAM, NR-Ours (no background repair our algorithm), and our algorithm, the absolute pose error (APE) is used in the experiment. The APE is the statistical information that compares the estimated trajectory with the reference trajectory and calculates the entire trajectory suitable for global consistency of test trajectories. At the same time, in order to eliminate accidental errors, this paper conducted five experiments on each sequence and took the root mean square error (RMSE), mean value (Mean), and standard deviation (SD) as its measurement units. The experimental results of the absolute pose errors of the four algorithms are compared in Table 1, where boldface represents the optimal value. It can be seen from the table that our algorithm performs better than the other three algorithms in the four sequence scenes, which can effectively reduce the influence of dynamic objects on pose. As can be seen from the figure, some experimental data of NR-Ours are the same as those of DynaSLAM, but most of them are better than those of DynaSLAM. This shows that the performance of our algorithm is already higher than that of DynaSLAM without background restoration. However, the addition of the background restoration algorithm further improves the performance of our algorithm. This shows that the background repair algorithm can effectively supplement the missing feature information due to the removal of dynamic objects and then provide a good basis for the localization and mapping of the SLAM system.

5.3 Feature extraction

In this paper, the TUM dataset is selected to verify the feature extraction effect of this paper's algorithm VTD-SLAM in dynamic scenes. As shown in Fig. 11, the feature extraction results of three algorithms, i.e., ORB-SLAM2, DynaSLAM, and VTD-SLAM, are compared to the TUM dataset. Figure 11a–d show the feature extraction results for ORB-SLAM2. This algorithm cannot reject the feature points on dynamic objects, which will cause the SLAM system to produce serious interference in the front-end alignment and poor bit pose estimation and composition. Figure 11e–h show the feature extraction effect of DynaSLAM, which only rejects feature points on dynamic objects but does not perform back-

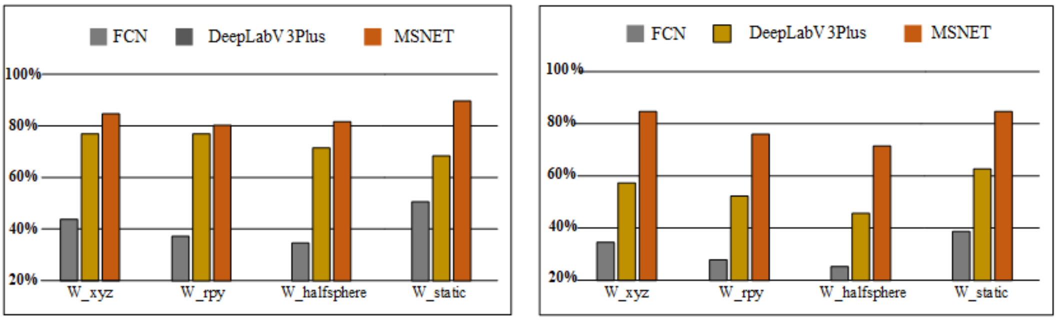


(a) FCN W_xyz (b) FCN W_rpy (c) FCN W_halfsphere (d) FCN W_static

(e) DeepLabV3Plus W_xyz (f) DeepLabV3Plus W_rpy (g) DeepLabV3Plus W_halfsphere (h) DeepLabV3Plus W_static

(i) MSNET W_xyz (j) MSNET W_rpy (k) MSNET W_halfsphere (l) MSNET W_static

Figure 8. Comparison of semantic segmentation of three algorithms in the TUM dataset.



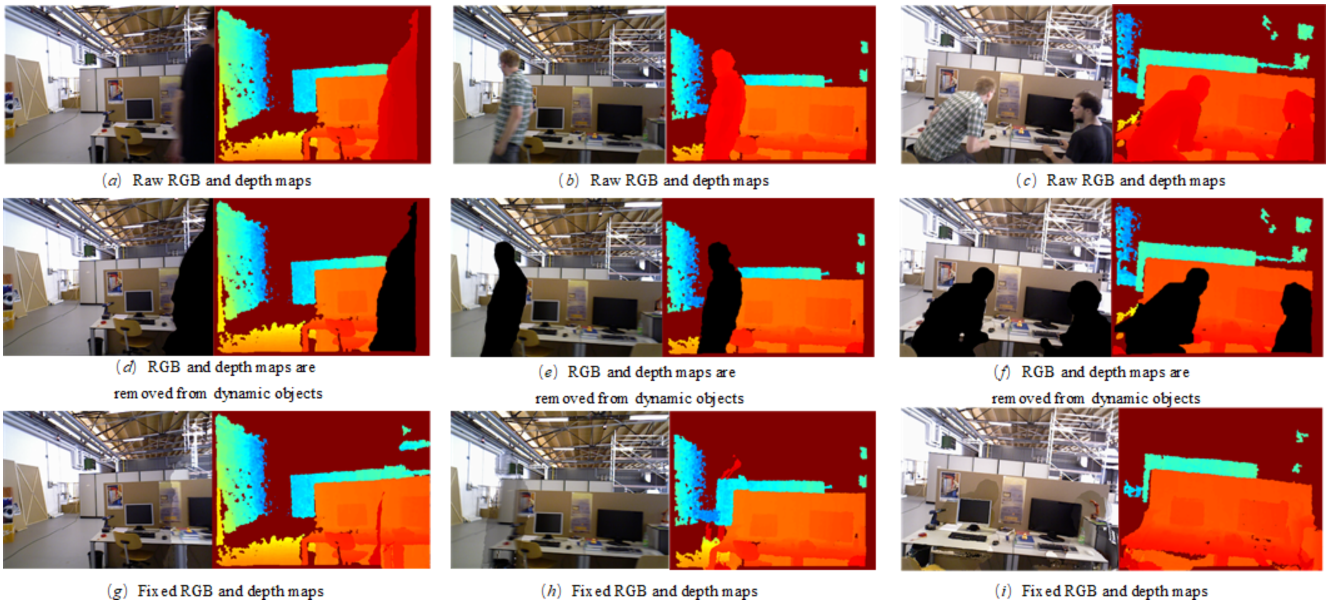
(a) Comparison of MIoU values of 3 algorithms (b) Comparison of MPA values of 3 algorithms

Figure 9. Evaluation index comparison.

ground restoration, and can use fewer static features. Figure 11i–l show the feature extraction effect of this algorithm. By eliminating dynamic regions and repairing static backgrounds in the eliminated regions, this algorithm extracts richer static feature points and builds a more accurate trajectory map.

5.4 SLAM system evaluation

In this paper, the W_halfsphere, W_rpy, W_xyz, and W_static sequences containing dynamic scenes in the TUM dataset are selected to verify the effectiveness of the algorithms. Figure 12 shows the trajectory plots constructed by ORB-SLAM2, DS-SLAM, DynaSLAM, and the four algorithms of VTD-SLAM proposed in this paper under different scenarios. The black curve in the figure indicates the real trajectory of camera motion, the blue curve is the estimated camera motion trajectory by the SLAM algorithm, and the red curve is the trajectory error. From Fig. 12, we can see that, since ORB-SLAM2 is only for static environments and cannot recognize dynamic objects, large trajectory errors when operating in a dynamic environment will


(a) Raw RGB and depth maps

(b) Raw RGB and depth maps

(c) Raw RGB and depth maps

(d) RGB and depth maps are removed from dynamic objects

(e) RGB and depth maps are removed from dynamic objects

(f) RGB and depth maps are removed from dynamic objects

(g) Fixed RGB and depth maps

(h) Fixed RGB and depth maps

(i) Fixed RGB and depth maps

Figure 10. Effect picture of background restoration.

Table 1. Comparison of absolute pose errors of three algorithms in the TUM dataset (m). Bold numbers indicate optimal values, with smaller values representing better algorithm performance.

TUM sequence	ORB-SLAM2			Dyna-SLAM			NR-Ours			Ours		
	RMSE	Mean	SD	RMSE	Mean	SD	RMSE	Mean	SD	RMSE	Mean	SD
fr3_walking_xyz	0.75	0.58	0.46	0.07	0.05	0.06	0.07	0.05	0.04	0.05	0.04	0.03
fr3_walking_static	0.41	0.31	0.26	0.06	0.05	0.04	0.05	0.04	0.03	0.03	0.03	0.01
fr3_ walking_ rpy	0.98	0.83	0.52	0.34	0.44	0.21	0.24	0.33	0.11	0.23	0.20	0.11
fr3_walking_halfsphere	0.28	0.27	0.17	0.26	0.20	0.19	0.13	0.11	0.10	0.13	0.09	0.09
Mean	0.60	0.50	0.35	0.18	0.19	0.13	0.12	0.13	0.07	0.11	0.09	0.06

have a greater impact on the back-end composition. The DS-SLAM algorithm recognizes and rejects dynamic objects in the environment through the SegNet instance segmentation network and motion feature point detection, but the SegNet instance segmentation network can only recognize 20 kinds of objects, and it is easy to omit segmenting dynamic objects beyond these 20 kinds when running in indoor environments with rich kinds of information, which causes its trajectory error to be larger than our algorithm's trajectory error. DynaSLAM also uses an instance segmentation network to segment the region where the dynamic objects are located and eliminates them to reduce the influence of dynamic objects on the composition. However, the DynaSLAM algorithm uses a MASK-RCNN network, which is limited by the sensory field of the convolutional network and the size of the convolutional kernel, making the MASK-RCNN network unable to obtain global semantic information, and it is prone to segmentation errors and incomplete segmentation when segmenting large objects, so the trajectory error generated by it for the SLAM task is still larger than that of our algorithm. However, these two algorithms suffer from limitations in accurately and clearly segmenting dynamic objects. Additionally, neither approach performs background restoration after dynamic object rejection, and both fail to extract feature points from the rejected areas. As a result, the number of static feature points for pose estimation and map building is reduced, which can significantly impact the positioning accuracy of the algorithm. The VTD-SLAM algorithm proposed in this paper incorporates the MSNET semantic segmentation algorithm, and thanks to the attention mechanism, it is able to perceive the global semantic information so as to accurately and clearly recognize the dynamic objects in the environment and eliminate them. The optimal nearest-neighbor pixel-matching method proposed by us effectively complements the static background of the eliminated region, and the high-quality feature points are filtered out to be used for the bit position estimation and composition. In addition, we filter out the high-quality feature points for position estimation and composition, which reduces the influence of dynamic objects on the composition. Therefore, the algorithm in this pa-

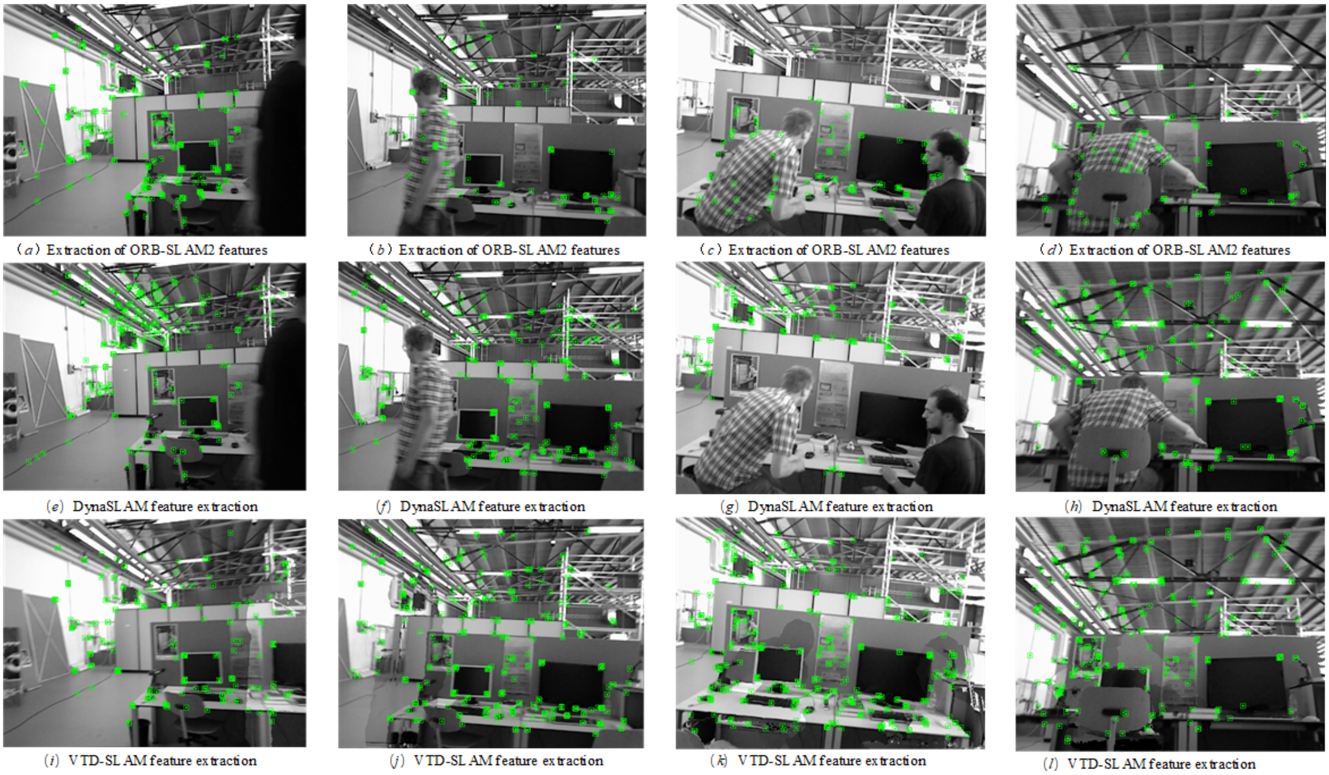

(a) Extraction of ORB-SLAM2 features
(b) Extraction of ORB-SLAM2 features
(c) Extraction of ORB-SLAM2 features
(d) Extraction of ORB-SLAM2 features
(e) DynaSLAM feature extraction
(f) DynaSLAM feature extraction
(g) DynaSLAM feature extraction
(h) DynaSLAM feature extraction
(i) VTD-SLAM feature extraction
(j) VTD-SLAM feature extraction
(k) VTD-SLAM feature extraction
(l) VTD-SLAM feature extraction

Figure 11. Three different algorithm feature extraction results.

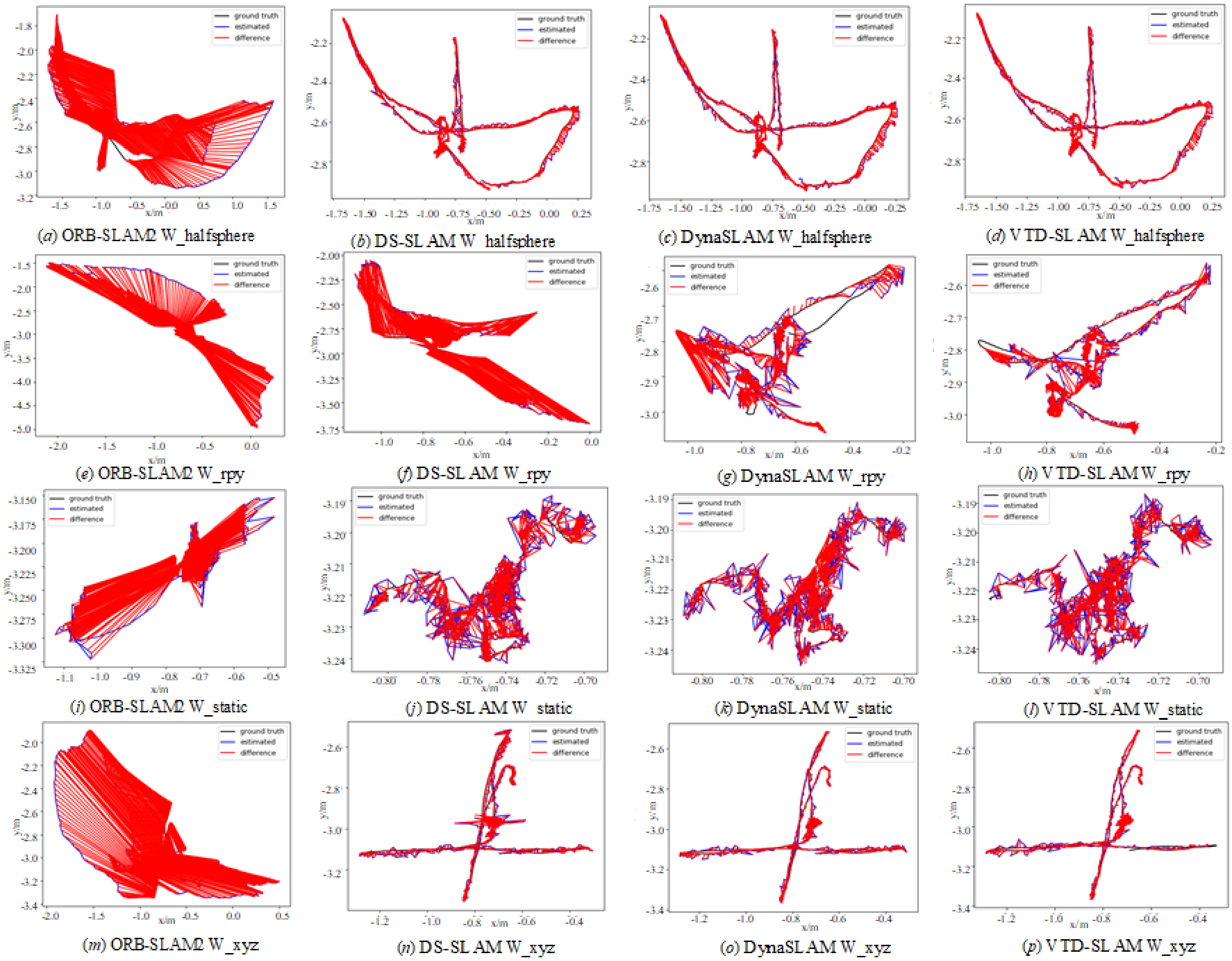
per has high accuracy, the trajectory error is smaller than the other three algorithms, and the constructed trajectory map is closer to the real trajectory (Shi et al., 2023a, b).

Table 2 shows the RMSE of the comparison of absolute trajectory errors of four algorithms in the TUM dataset, and the lower value of the RMSE represents the higher robustness of the system. From Table 1, we can see that the algorithms in this paper achieve lower error values in all four sequences, of which the RMSEs of the W_halfsphere and W_static sequences are smaller than ORB-SLAM2 and DS-SLAM and are equal to that of the DynaSLAM algorithm. The W_rpy sequence is 95.6 %, 93.5 %, and 95.5 % less than ORB-SLAM2, DS-SLAM, and DynaSLAM by 95.6 %, 93.5 %, and 17.1 %, respectively, and the W_xyz sequence is less by 97.2 %, 48.0 %, and 13.3 % compared to ORB-SLAM2, DS-SLAM, and DynaSLAM, respectively. Our mean absolute trajectory error is still lower than the other three algorithms, which shows that our algorithm can effectively reduce the influence of dynamic objects on the SLAM system.

5.5 Real scenario testing

The effectiveness of the algorithm in this paper is verified in a real scenario where the semantic segmentation algorithm as the input of VTD-SLAM is generated offline from the PC. The experimental platform is a Husky wheeled robot with the following hardware configuration: the CPU is an i7-10875H processor, the memory is 8 GB, the GPU is GTX1080, and the OS is Ubuntu 18.04. The robot hardware appearance is shown in Fig. 13a, and the main parameter settings are shown in Table 3. The real environment is shown in Fig. 13b, with a size of 12 m × 5 m. Figure 13c shows the layout of the real scene, where the yellow part is the lab bench, sections A–B form the robot motion route, and sections C–D are the pedestrian round-trip motion route. Table 2 shows the main parameter settings for the Husky wheeled mobile robot.

A frame of image information acquired during the operation of the selected mobile robot is shown in Fig. 14. Figure 14a is the original image frame in the dynamic scene, and Fig. 14b is the semantic segmentation result in the dynamic scene, demonstrating the effectiveness of the MSNET algorithm in accurately segmenting potential dynamic objects. The background restoration result of the VTD-SLAM algorithm is shown in Fig. 14c, where the red box serves as an auxiliary marker and the box represents the restored dynamic object area. As the number of static feature points that can be extracted after the dynamic objects are removed is limited, this may lead to poor bit pose estimation and map construction abilities of the SLAM system. By performing background restoration on the rejected regions of dynamic objects and screening the restored regions for high-quality feature points, this algorithm improves the accuracy of robot pose estimation and composition, as shown in Fig. 14d of the algorithm with the ORB-SLAM2 and DynaSLAM algo-



(*a*) ORB-SLAM2 W_halfsphere (*b*) DS-SLAM W_halfsphere (*c*) DynaSLAM W_halfsphere (*d*) VTD-SLAM W_halfsphere

(*e*) ORB-SLAM2 W_rpy (*f*) DS-SLAM W_rpy (*g*) DynaSLAM W_rpy (*h*) VTD-SLAM W_rpy

(*i*) ORB-SLAM2 W_static (*j*) DS-SLAM W_static (*k*) DynaSLAM W_static (*l*) VTD-SLAM W_static

(*m*) ORB-SLAM2 W_xyz (*n*) DS-SLAM W_xyz (*o*) DynaSLAM W_xyz (*p*) VTD-SLAM W_xyz

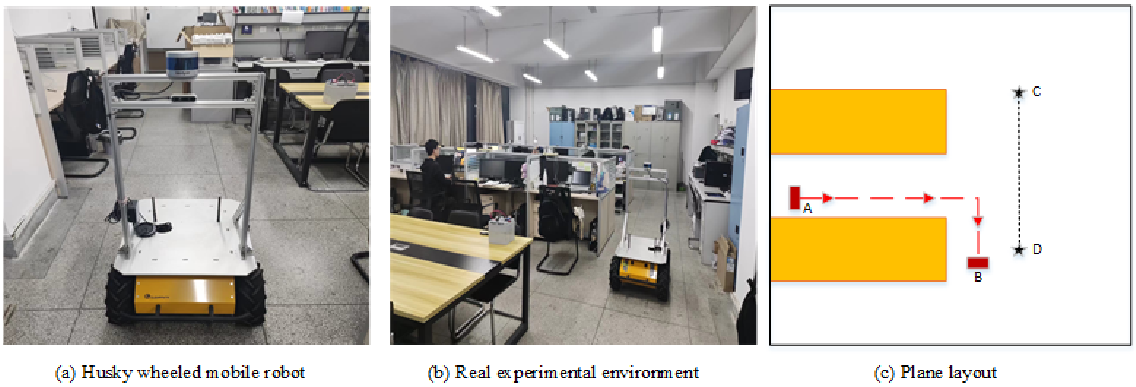
Figure 12. Trajectories of the four different algorithms.

Table 2. Comparison of absolute trajectory errors of four algorithms in the TUM dataset (m). Bold numbers indicate optimal values, with smaller values representing better algorithm performance.

Test sequence	ORB-SLAM2	DS-SLAM	DynaSLAM	Ours
W_halfsphere	0.351	0.025	0.025	0.025
W_rpy	0.662	0.444	0.035	0.029
W_static	0.090	0.008	0.006	0.006
W_xyz	0.459	0.025	0.015	0.013
Mean	0.391	0.126	0.034	0.018

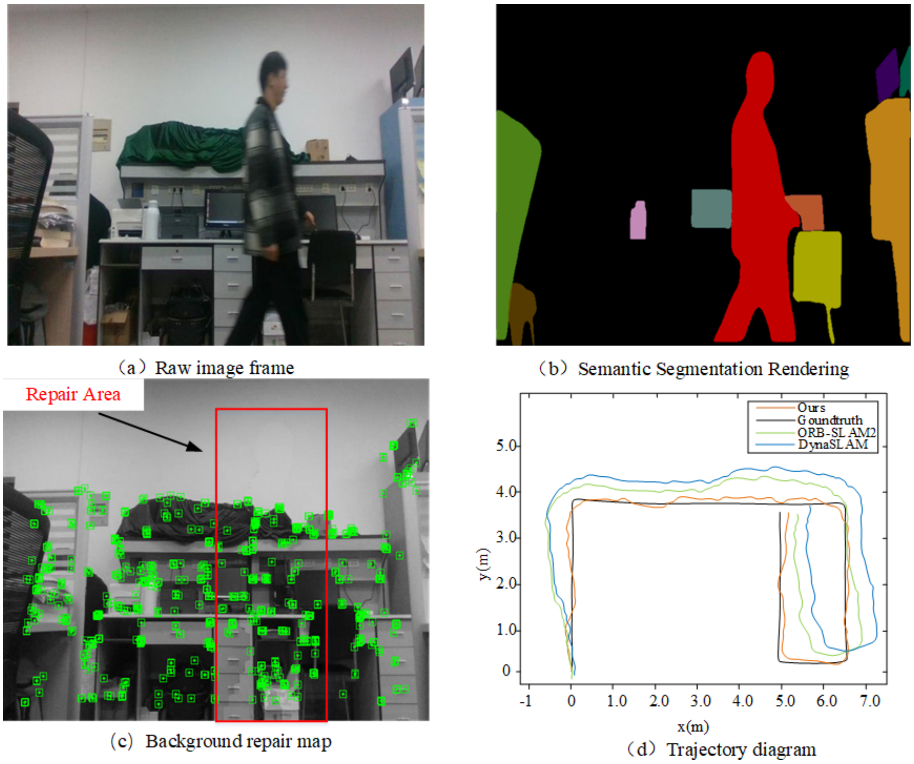
rithms and real trajectories of contrast. Because the ORB-SLAM2 algorithm cannot identify the dynamic target, its trajectory is quite different from the actual trajectory. The DynaSLAM algorithm adds the MASK-RCNN algorithm to identify dynamic objects, so its trajectory error is reduced compared with the ORB-SLAM2 algorithm, but the trajectory error is still larger than ours. However, the algorithm in this paper increases the influence of semantic segmentation, eliminates the dynamic target, and carries out background repair, so that the trajectory error of the VTD-SLAM algorithm is minimized and is very close to the real trajectory, which shows the reliability of our algorithm in the real scene.

Figure 15 shows the comparison of the running time of the three algorithms ORB-SLAM2, DynaSLAM, and VTD-SLAM2 in four TUM dataset sequences and in real scenes. It can be seen from the figure that the running time of the



(a) Husky wheeled mobile robot (b) Real experimental environment (c) Plane layout

Figure 13. Experimental platform and real experimental environment.



(a) Raw image frame (b) Semantic Segmentation Rendering (c) Background repair map (d) Trajectory diagram

Figure 14. Real scene experiment.

Table 3. Main parameter setting.

Variable name	Parameter	Numerical value
Operating speed	V_s	$1\,\mathrm{m\,s^{-1}}$
Rotational speed	V_θ	$0.48\,\mathrm{rad\,s^{-1}}$
Range of direction change	Θ	$[0, 2\pi]$
Camera sampling frequency	H	30 fps

algorithm in this paper is reduced by about 20 % compared with that of the DynaSLAM algorithm. Due to the addition of the semantic segmentation network and the elimination of dynamic objects, the running time of this algorithm is greatly increased compared with ORB-SLAM2, but it can still effectively complete the recognition of dynamic objects and the repair of the elimination area.

6 Conclusion

To improve the robustness of mobile robots in dynamic scenes, a VTD-SLAM algorithm is proposed in this paper, which has the following advantages. (1) To address the issues of segmentation blurring or errors that may arise from object movement during the semantic segmentation of dy-

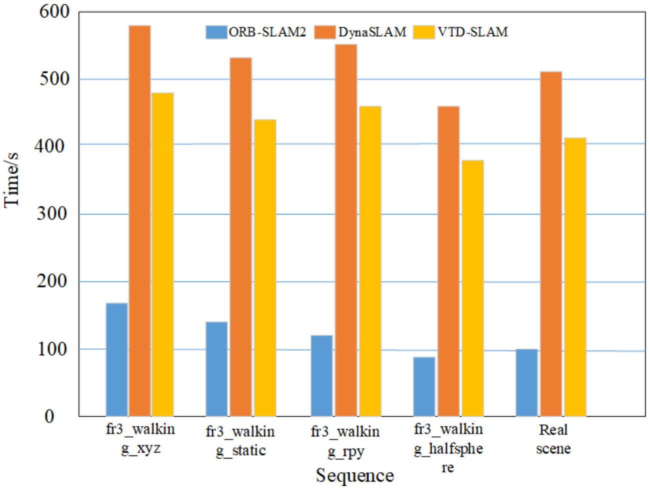


Figure 15. Comparison of the running times of the three algorithms.

namic objects using existing networks, this paper introduces the MSNET algorithm. The proposed approach leverages a multiclass feature enhancement and multiclass feature guidance technique to improve the semantic segmentation of dynamic objects. The problems of segmentation ambiguity and segmentation errors are reduced effectively, and the semantic segmentation ability of dynamic objects is improved. (2) Adopt optimal nearest-neighbor pixel matching to repair the images in the complementary rejection region and extract the high-quality feature points in the region to provide more feature information for the system bit pose estimation and the construction of dense point cloud maps. To demonstrate the effectiveness of the algorithm proposed in this paper, it was validated using the publicly available TUM dataset. The results show that the VTD-SLAM algorithm outperforms three other algorithms, i.e., ORB-SLAM2, DS-SLAM, and DynaSLAM, in terms of localization accuracy and composition capability. Our approach introduces the use of Transformer's large models in the SLAM domain, which will allow workers within the SLAM field to focus more on more accurate large model networks, and can introduce deep-learning frameworks into SLAM with more confidence, contributing to the field application of SLAM engineering.

Code availability. The code is available at https://github.com/Oneghr/VTD-SLAM (Chen et al., 2023).

Data availability. All data are included in this study are available upon request by contacting the corresponding author.

Video supplement. See our demo video https://github.com/Oneghr/VTD-SLAM (Chen et al., 2023) for the videos.

Author contributions. MC, HG, RQ, GG, and HC contributed significantly to the conception and design of the work, data acquisition, analysis, and interpretation. HG and RQ contributed to the acquisition of simulation experimental data and data collation. HG helped with the writing and the language.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher' note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors.

Acknowledgements. First of all, I would like to thank all the people who helped me in the thesis writing process, especially all the participants of this team. Secondly, I would like to thank every editor and teacher for their great support.

Financial support. This work was supported by the National Nature Science Foundation of China (grant no. 61903002), the Anhui University Collaborative Innovation Project (grant no. GXXT-2021-050), the Anhui University Outstanding Youth Research Project (grant no. 2022AH020065), funding for scientific research activities of potential candidates of academic and technical leaders in Anhui Province (grant no. 2022H292), and the Anhui Polytechnic University young and middle-aged top-talent program.

Review statement. This paper was edited by Zi Bin and reviewed by Hang Su and two anonymous referees.

References

An, L., Pan, X., Li, T., and Wang, M.: A visual dynamic-SLAM method based semantic segmentation and multi-view geometry, in: Proceedings of the International Conference on High Performance Computing and Communication, Xiamen, China, 3–5 December 2021, 255–263, https://doi.org/10.1117/12.2628175, 2022.

Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. B.: PatchMatch: A randomized correspondence algorithm for structural image editing, ACM T. Graphic., 28, 10 pp., 2009.

Bescos, B., Fácil, J. M., Civera, J., and Neira, J.: DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes, IEEE Robotics and Automation Letters, 3, 4076–4083, https://doi.org/10.1109/LRA.2018.2860039, 2018.

Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J. M. M., and Tardós J. D.: Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam, IEEE T. Robot., 37, 1874–1890, https://doi.org/10.1109/TRO.2021.3075644, 2021.

Cao, J., Yu, J., Pan, S., Gao, F., Yu, C., Xu, Z., Huang, Z., and Wang, Y.: SLAM pose graph optimization method using dual visual odometry, Journal of Computer Aided Design and Graphics, 33, 1264–1272, 2021.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S.: End-to-end object detection with transformers, in: Proceedings of the 2020 European conference on computer vision, 2020 European conference on computer vision workshops, Glasgow, 23–28 August 2020, Springer, 213–229, 2020.

Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European conference on computer vision, European conference on computer vision workshops, Munich, Germany, 8–14 September 2018, Springer, 801–818, 2018.

Chen, M., Ding, L., and Zhang, Y.: Fast PL-SLAM Algorithm Based on Improved Keyframe Selection Strategy, Acta Electronica Sinica, 50, 608–618, https://doi.org/10.12263/DZXB.20210567, 2022.

Chen, M., Guo, H., Qian, R., Gong, G., and Cheng, H.: VSLAM Algorithm Based on Improved Vision Transformer Semantic Segmentation in Dynamic Scenes, GitHub [code/video], https://github.com/Oneghr/VTD-SLAM (last access: 19 December 2023), 2023.

Dai, J., Li, Y., He, K., and Sun, J.: R-fcn: Object detection via region-based fully convolutional networks, GitHub [code], https://github.com/daijifeng001/r-fcn (last access: 19 December 2023), 2016.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale, arXiv[preprint], https://doi.org/10.48550/arXiv.2010.11929, 3 June 2021.

Feng, X., Pei, W., Jia, Z., Chen, F., Zhang, D., and Lu, G.: Deep-masking generative network: A unified framework for background restoration from superimposed images, IEEE T. Image Process., 30, 4867–4882, https://doi.org/10.1109/TIP.2021.3076589, 2021.

Gao, X., Shi, X., Ge, Q., and Chen, K.: A review of visual SLAM for dynamic object scenes, Robotics, 43, 733–750, https://doi.org/10.13973/j.cnki.robot.200323, 2021.

Guan, L., Franco, J. S., and Pollefeys, M.: Multi-view occlusion reasoning for probabilistic silhouette-based dynamic scene reconstruction, Int. J. Comput. Vision, 90, 283–303, https://doi.org/10.1007/s11263-010-0341-y, 2010.

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., Yang, Z., Zhang, Y., and Tao, D.: A survey on vision transformer, IEEE T. Pattern Anal., 45, 87–110, https://doi.org/10.1109/TPAMI.2022.3152247, 2022.

Hu, J., Shen, L., and Sun, G.: Squeeze-and-excitation networks, in: Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, IEEE conference on computer vision and pattern recognition workshops, Salt Lake City, UT, USA, 18–22 June 2018, IEEE, 7132–7141, 2018.

Huang, B., Zhao, J., and Liu, J.: A survey of simultaneous localization and mapping with an Envision in 6G Wireless Networks, arXiv [preprint], https://doi.org/10.48550/arXiv.1909.05214, 14 February, 2020.

Kaneko, M., Iwami, K., Ogawa, T., Yamasaki, T., and Aizawa, K.: Mask-slam: Robust feature-based monocular slam by masking using semantic segmentation, in: Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, USA, 18–22 June 2018, IEEE, 258–266, 2018.

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L.: Microsoft coco: Common objects in context, in: 13th European Conference on Computer Vision, European Conference on Computer Vision workshops, Zurich, Switzerland, 6–12 September 2014, Springer, 8693, 740–755, 2014.

Liu, Y. and Miura, J.: RDS-SLAM: real-time dynamic SLAM using semantic segmentation methods, IEEE Access, 9, 23772–23785, https://doi.org/10.1109/ACCESS.2021.3050617, 2021.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Virtual, 11–17 October 2021, IEEE, 10012—10022, 2021.

Liu, Z., Yang, D., Wang, Y., Liu, M., and Li, R.: EGNN: Graph structure learning based on evolutionary computation helps more in graph neural networks, Appl. Soft Comput., 135, 110040, https://doi.org/10.1016/j.asoc.2023.110040, 2023.

Mur, A. R. and Tardós, J. D.: Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras, IEEE T. Robot., 33, 1255–1262, https://doi.org/10.1109/TRO.2017.2705103, 2017.

Pumarola, A., Vakhitov, A., Agudo, A., Sanfeliu, A., and Moreno, Nr. F.: PL-SLAM: Real-time monocular visual SLAM with points and lines, in: Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017, IEEE, 4503–4508, https://doi.org/10.1109/ICRA.2017.7989522, 2017.

Schubert, D., Goll, T., Demmel, N., Usenko, V., Stückler, J., and Cremers, D.: The TUM VI benchmark for evaluating visual-inertial odometry, in: Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018, IEEE, 1680–1687, https://doi.org/10.1109/IROS.2018.8593419, 2018.

Shi, Y., Li, H., Fu, X., Luan, R., Wang, Y., Sun, Z., Niu, Y., Wang, C., Zhang, C., and Wang, Z. L.: Self-powered difunctional sensors based on sliding contact-electrification and tribovoltaic effects for pneumatic monitoring and controlling, Nano Energy, 110, 108339, https://doi.org/10.1016/j.nanoen.2023.108339, 2023a.

Shi, Y., Li, L., Yang, J., Wang, Y., and Hao, S.: Center-based transfer feature learning with classifier adaptation for surface defect recognition, Mech. Syst. Signal Pr., 188, 110001, https://doi.org/10.1016/j.ymssp.2022.110001, 2023b.

Strudel, R., Garcia, R., Laptev, I., and Schmid, C.: Segmenter: Transformer for semantic segmentation, in: Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Virtual, 11–17 October 2021, IEEE, 597–606, 2021

Sun, Y., Liu, M., and Meng, M.: Motion removal for reliable RGB-D SLAM in dynamic environments, Robot. Auton. Syst., 108, 115–128, https://doi.org/10.1016/j.robot.2018.07.002, 2018.

Targ, S., Almeida, D., and Lyman, K.: Resnet in resnet: Generalizing residual architectures, arXiv [preprint], https://doi.org/10.48550/arXiv.1603.08029, 25 March 2016.

Tian, C., Xu, Z., Wang, L., and Liu, Y.: Arc fault detection using artificial intelligence: Challenges and benefits, Math. Biosci. Eng., 20, 12404–12432, https://doi.org/10.3934/mbe.2023552, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I.: Attention is all you need, Adv. Neur. In., 30, 11 pp., 2017.

Wang, Y., Liu, Z., Xu, J., and Yan, W.: Heterogeneous network representation learning approach for ethereum identity identification, IEEE Transactions on Computational Social Systems, 10, 890 –899, https://doi.org/10.1109/TCSS.2022.3164719, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I.: Attention is all you need, in: Proceedings of the Annual Conference on Neural Information Processing Systems, Conference and Workshop on Neural Information Processing Systems workshops, Long Beach, Los Angeles area, USA, 4–9 December 2017, MIT Press, Los Angeles, ISBN 9781510860964, 2017.

Woo, S., Park, J., Lee, J. Y., and Kweon, I. S.: Cbam: Convolutional block attention module, in: Proceedings of the European conference on computer vision, European conference on computer vision workshops, Munich, Germany, 8–14 September 2018, 3–19, 2018.

Xu, C., Zhou, Y., and Luo, C.: Visual SLAM Method Based on Optical Flow and Instance Segmentation in Dynamic Scenes, Acta Optica Sinica, 42, 147–159, https://doi.org/10.3788/AOS202242.1415002, 2022.

Yu, C., Liu, Z., Liu, X. J., Xie, F., Yang, Y., Wei, Q., and Fei, Q.: DS-SLAM: A semantic visual SLAM towards dynamic environments, in: Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE/RSJ International Conference on Intelligent Robots and Systems workshops, Madrid, Spain, 1–5 October 2018, IEEE, 1168–1174, https://doi.org/10.1109/IROS.2018.8593691, 2018.

Zhao, J. and Lv, Y.: Output-feedback Robust Tracking Control of Uncertain Systems via Adaptive Learning, Int. J. Control. Autom., 21, 1108–1118, https://doi.org/10.1007/s12555-021-0882-6, 2023.

Zhong, F., Wang, S., Zhang, Z., Chen, C., and Wang, Y.: Detect-SLAM: Making object detection and SLAM mutually beneficial, in: Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision, Harrah's and Harvey's Lake Tahoe, NV, USA, 11–15 March 2018, IEEE, 1001–1010, https://doi.org/10.1109/WACV.2018.00115, 2018.

Zhou, Z., Cao, J., and Di, S.: A review of SLAM algorithm for 3D lidar, Journal of Instrument and Instrument, 42, 13–27, https://doi.org/10.19650/j.cnki.cjsi.J2107897, 2021.